k-Means Clustering

In [1]:

```
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
```

In [2]:

```
X = np.array([[1, 2], [1, 4], [1, 0],
              [2, 1], [3, 1],
              [4, 2], [4, 4], [4, 0]])
print(X)
```

```
[[1 2]
 [1 4]
 [1 0]
 [2 1]
 [3 1]
 [4 2]
 [4 4]
 [4 0]]
```

In [3]:

```
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(X)
```

Out[3]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=0, tol=0.0001, verbose=0)
```

In [4]:

```
print(kmeans.labels_)
```

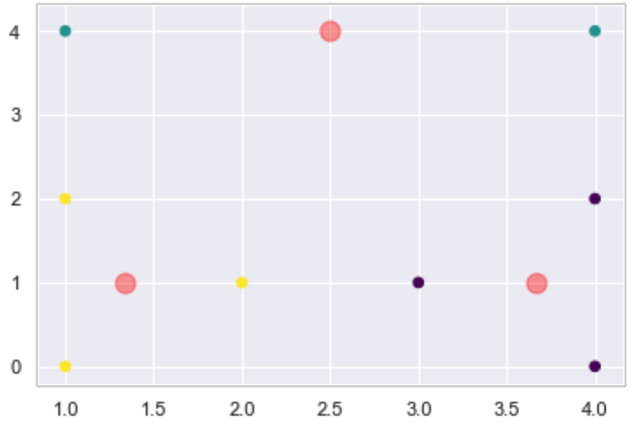
```
[2 1 2 2 0 0 1 0]
```

In [5]:

```
centers = kmeans.cluster_centers_
print(centers)
```

```
[[3.66666667 1.        ]
 [2.5        4.        ]
 [1.33333333 1.        ]]
```

In [6]:
```
plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_, s=32, cmap="viridis")
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=128, alpha=0.4);
```



In [7]:
```
kmeans.predict([[0, 1], [2, 2], [3, 2], [3, 3], [4, 3]])
```

Out[7]:
```
array([2, 2, 0, 1, 1], dtype=int32)
```

In [8]:
```
X
```

Out[8]:
```
array([[1, 2],
       [1, 4],
       [1, 0],
       [2, 1],
       [3, 1],
       [4, 2],
       [4, 4],
       [4, 0]])
```

In [9]:
```
kmeans.fit_predict(X)
```

Out[9]:
```
array([2, 1, 2, 2, 0, 0, 1, 0], dtype=int32)
```

```
In [10]: kmeans.fit_transform(X)
```

```
Out[10]: array([[2.84800125, 2.5       , 1.05409255],
                [4.01386486, 1.5       , 3.01846171],
                [2.84800125, 4.27200187, 1.05409255],
                [1.66666667, 3.04138127, 0.66666667],
                [0.66666667, 3.04138127, 1.66666667],
                [1.05409255, 2.5       , 2.84800125],
                [3.01846171, 1.5       , 4.01386486],
                [1.05409255, 4.27200187, 2.84800125]])
```

```
In [ ]:
```

```
In [11]: from sklearn.datasets.samples_generator import make_blobs
```

```
In [12]: X, y_true = make_blobs(n_samples=300, centers=4, cluster_std=0.6, random_state=0)
```

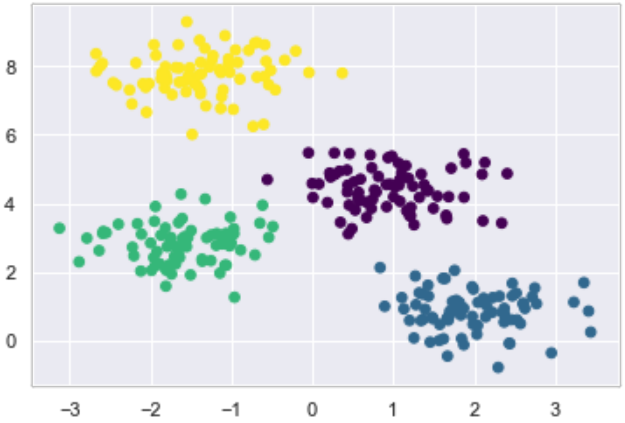
```
In [13]: print(X[:8])
```

```
[[ 0.83685684  2.13635938]
 [-1.4136581   7.40962324]
 [ 1.15521298  5.09961887]
 [-1.01861632  7.81491465]
 [ 1.27135141  1.89254207]
 [ 3.43761754  0.26165417]
 [-1.80822253  1.59701749]
 [ 1.41372442  4.38117707]]
```

```
In [14]: print(y_true[:8])
```

```
[1 3 0 3 1 1 2 0]
```

In [15]:
```
plt.scatter(X[:, 0], X[:, 1], c=y_true, s = 32, cmap='viridis');
```



In [16]:
```
kmeans = KMeans(n_clusters=4)

kmeans.fit(X)

y_kmeans = kmeans.predict(X)
```

In [17]:
```
kmeans.labels_
```

Out[17]:
```
array([0, 2, 1, 2, 0, 0, 3, 1, 2, 2, 3, 2, 1, 2, 0, 1, 1, 0, 3, 3, 0, 0,
       1, 3, 3, 1, 0, 1, 3, 1, 2, 2, 1, 2, 2, 2, 2, 2, 3, 0, 1, 3, 1, 1,
       3, 3, 2, 3, 2, 0, 3, 0, 2, 0, 0, 3, 2, 3, 2, 0, 2, 1, 2, 3, 3, 3,
       2, 0, 2, 3, 1, 3, 2, 3, 3, 2, 3, 1, 0, 2, 0, 1, 0, 0, 2, 1, 0, 1,
       2, 2, 1, 0, 2, 3, 3, 1, 0, 0, 1, 3, 2, 0, 2, 0, 1, 0, 0, 1, 2, 1,
       3, 3, 0, 2, 0, 1, 2, 0, 0, 1, 3, 0, 3, 0, 0, 0, 0, 3, 0, 3, 2, 3,
       3, 0, 2, 3, 3, 2, 1, 2, 2, 3, 1, 3, 1, 3, 2, 1, 2, 2, 2, 1, 2, 1,
       0, 3, 2, 3, 0, 1, 2, 1, 1, 0, 1, 3, 3, 1, 0, 1, 1, 2, 0, 1, 3, 2,
       0, 0, 1, 3, 0, 1, 3, 3, 1, 1, 1, 1, 0, 2, 1, 3, 1, 1, 3, 3, 3, 1,
       3, 2, 1, 3, 0, 3, 1, 2, 3, 2, 1, 2, 1, 3, 1, 1, 2, 3, 3, 0, 0, 1,
       2, 0, 0, 3, 0, 3, 1, 2, 2, 1, 1, 2, 1, 0, 3, 1, 0, 3, 2, 3, 0, 1,
       0, 2, 2, 2, 2, 3, 3, 2, 1, 3, 0, 1, 3, 3, 3, 0, 0, 2, 1, 1, 3, 0,
       2, 3, 1, 2, 1, 0, 0, 3, 3, 1, 0, 0, 0, 1, 2, 2, 0, 0, 1, 0, 0, 0,
       2, 3, 2, 1, 0, 0, 2, 2, 2, 0, 0, 1, 2, 3], dtype=int32)
```

In [18]:
```
print(y_kmeans[:8])
```

```
[0 2 1 2 0 0 3 1]
```

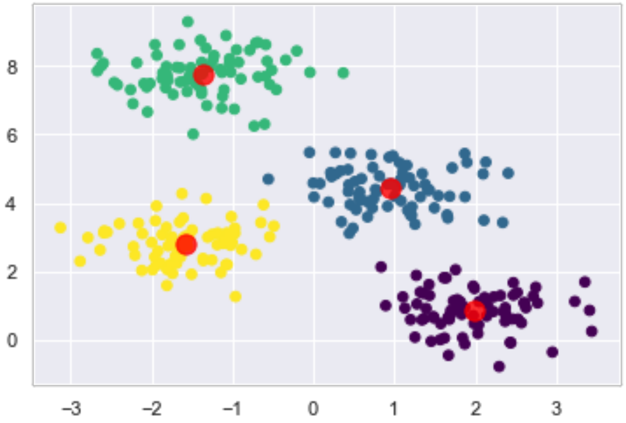
In [19]:

```
centers = kmeans.cluster_centers_
print(centers)
```

```
[[ 1.98258281  0.86771314]
 [ 0.94973532  4.41906906]
 [-1.37324398  7.75368871]
 [-1.58438467  2.83081263]]
```

In [20]:

```
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=32, cmap="viridis")
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=128, alpha=0.8);
```



In [ ]: