KMeans_Clustering_Cricket_Match_Score

In [32]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import warnings
```

In [22]:

```python
data = pd.read_csv('Sample scores.csv')
```
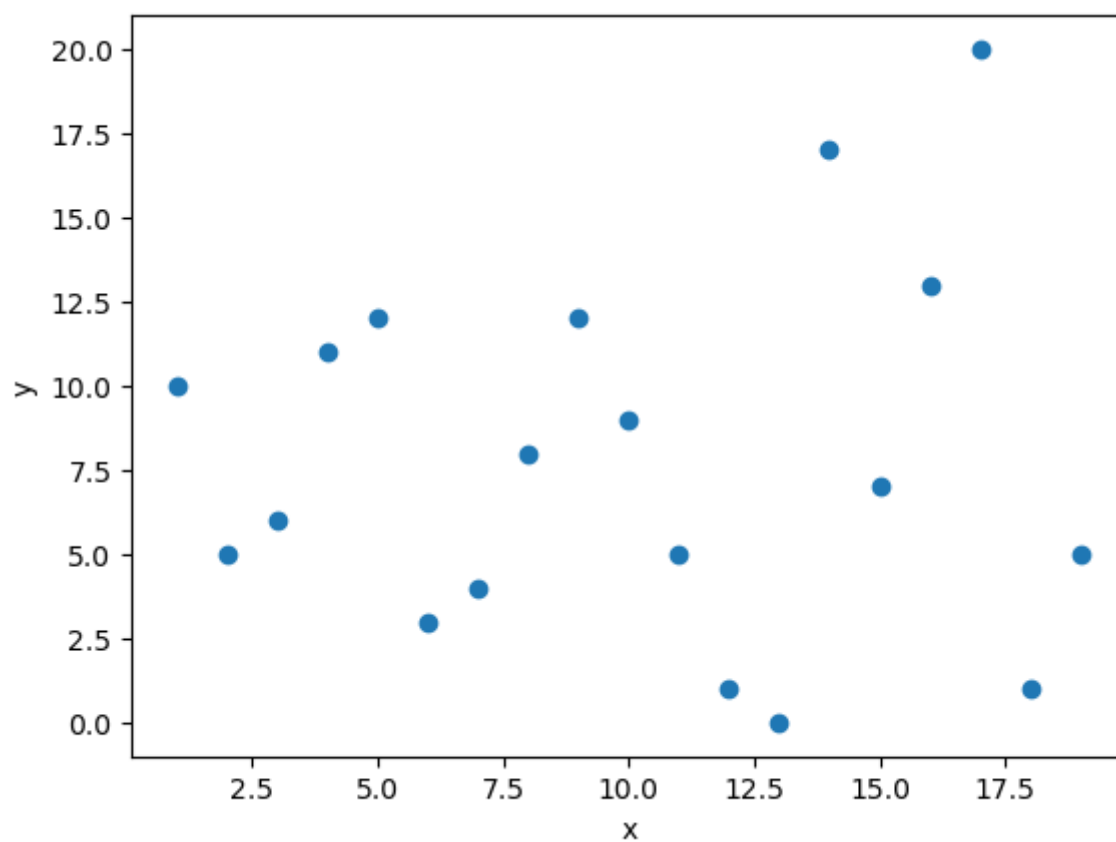
In [23]:

```python
data
```

Out[23]:

	Overs	Scores
0	1	10
1	2	5
2	3	6
3	4	11
4	5	12
5	6	3
6	7	4
7	8	8
8	9	12
9	10	9
10	11	5
11	12	1
12	13	0
13	14	17
14	15	7
15	16	13
16	17	20
17	18	1
18	19	5

```
plt.scatter(data['Overs'], data['Scores'])
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

```
df = pd.DataFrame(data, columns = ['Scores', 'Overs'])
```

```
df
```

	Scores	Overs
0	10	1
1	5	2
2	6	3
3	11	4
4	12	5
5	3	6
6	4	7
7	8	8
8	12	9
9	9	10
10	5	11
11	1	12
12	0	13
13	17	14
14	7	15
15	13	16
16	20	17
17	1	18
18	5	19

```python
kmeans = KMeans(n_clusters=3).fit(df)
```

```
C:\Users\baps\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\baps\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, wh
en there are less chunks than available threads. You can avoid it by setti
ng the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

```python
centroids = kmeans.cluster_centers_
print(centroids)
```

```
[[ 3.16666667 14.66666667]
 [ 8.          5.5       ]
 [16.66666667 15.66666667]]
```

```
plt.scatter(df['Overs'], df['Scores'], c=kmeans.labels_.astype(float), s=50, alpha=1)
plt.scatter(centroids[:,0], centroids[:,1], c='red', s=50)
plt.xlabel('Overs')
plt.ylabel('Scores')
plt.show()
```