In [82]:
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import scipy.cluster.hierarchy as sch
import seaborn as sns
```

In [83]:
```
d = pd.read_csv("epl_1819.csv")
```

Searching Data

In [84]:
```
d.head()
```

Out[84]:

	Team	category	general_league_position	finance _live_games_televised	finance _tv_revenue	general_ma
0	Manchester City	Champions League	1	26	150986355	
1	Liverpool	Champions League	2	29	152425146	
2	Chelsea	Champions League Qualification	3	25	146030216	
3	Tottenham	Champions League Qualification	4	26	145230801	
4	Arsenal	Europa League	5	25	142193180	

5 rows × 44 columns

In [6]:

```
d.isna().sum()
```

Out[6]:

```
Team                             0
category                         0
general_league_position          0
finance _live_games_televised    0
finance _tv_revenue              0
general_matches_played           0
general_won                      0
general_draw                     0
general_lost                     0
attack_scored                    0
defence_goals_conceeded          0
general_goal_difference          0
general_points                   0
general_squad_size               0
general_squad_average_age        0
general_squad_foreigners         0
finance _team_market             0
finance _market_average          0
attack_passes                    0
attack_passes_through            0
attack_passes_long               0
attack_passes_back               0
attack_crosses                   0
attack_corners_taken             0
attack_shots                     0
attack_shots_on_target           0
attack_goals_headed              0
attack_goals_penalty             0
attack_goals_box                 0
attack_goals_outsidebox          0
general_card_yellow              0
general_card_red                 0
attack_goals_counter             0
attack_goals_freekick            0
defence_saves                    0
defence_blocks                   0
defence_interceptions            0
defence_tackles                  0
defence_tackles_last_man         0
defence_clearances               0
defence_clearances_headed        0
defence_penalty_conceeded        0
attack_posession                 0
attack_pass_accuracy             0
dtype: int64
```

In [91]:

```
d[['general_won','general_draw','general_lost','attack_posession','attack_pass_ac
```

Out[91]:

	general_won	general_draw	general_lost	attack_posession	attack_pass_accu
general_won	1.000000	-0.355612	-0.941341	0.849123	0.74
general_draw	-0.355612	1.000000	0.019354	-0.320816	-0.20
general_lost	-0.941341	0.019354	1.000000	-0.792505	-0.71
attack_posession	0.849123	-0.320816	-0.792505	1.000000	0.95
attack_pass_accuracy	0.741568	-0.204273	-0.719529	0.951080	1.00

Take 2 col to make clustering

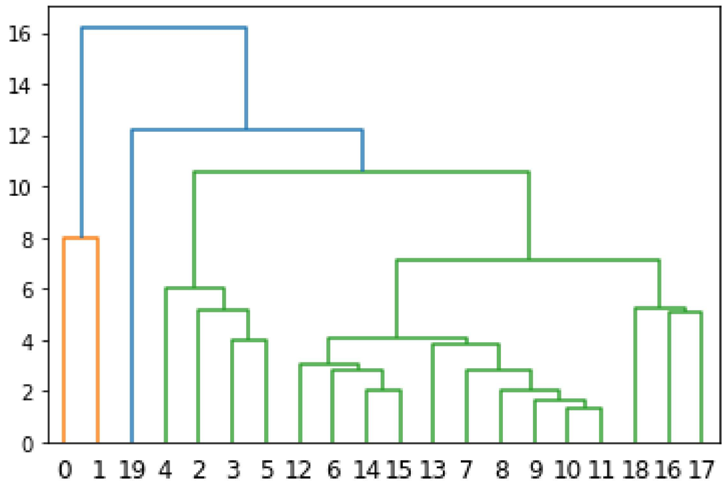
In [110]:

```
du=d.iloc[:, lambda d: [9,42]].values
```

create hierarchy dendrogram by single linkage

In [125]:

```
%%time
dendrogram = shc.dendrogram(shc.linkage(du,"single"))
plt.title="dendrogram"
plt.show()
```



```
Wall time: 588 ms
```

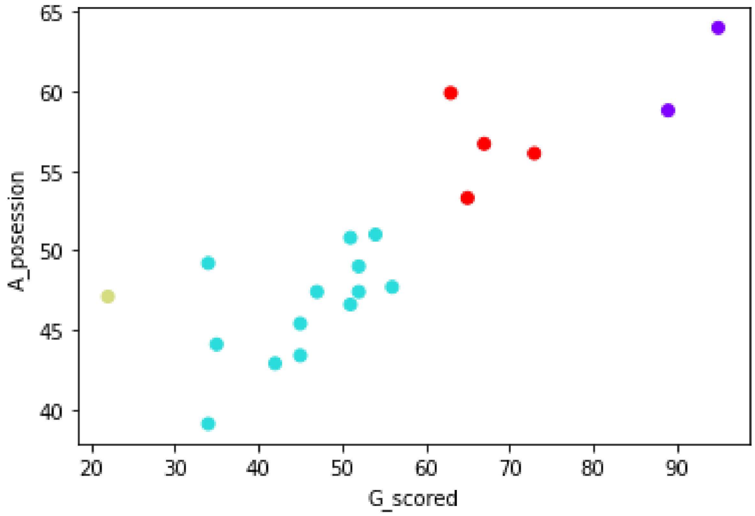
In [126]:
```
%%time
from sklearn.cluster import AgglomerativeClustering
ncluster = AgglomerativeClustering(n_clusters=4,affinity="euclidean",linkage="si
ncluster.fit_predict(du)
```

```
Wall time: 1.01 ms
```

Out[126]:
```
array([0, 0, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2],
      dtype=int64)
```

In [160]:
```
plt.scatter( du[:,0] , du[:,1],c=ncluster.labels_,cmap="rainbow")

plt.xlabel("G_scored")
plt.ylabel("A_posession")
plt.show()
```



In [114]:
```
pip install scikit-learn-extra
```

```
Requirement already satisfied: scikit-learn-extra in c:\users\owner\anaconda3\l
ib\site-packages (0.2.0)Note: you may need to restart the kernel to use updated
packages.
Requirement already satisfied: scipy>=0.19.1 in c:\users\owner\anaconda3\lib\si
te-packages (from scikit-learn-extra) (1.5.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\owner\anaconda3\lib\si
te-packages (from scikit-learn-extra) (1.18.5)

Requirement already satisfied: scikit-learn>=0.23.0 in c:\users\owner\anaconda3
\lib\site-packages (from scikit-learn-extra) (0.23.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\owner\anaconda3
\lib\site-packages (from scikit-learn>=0.23.0->scikit-learn-extra) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\users\owner\anaconda3\lib\sit
e-packages (from scikit-learn>=0.23.0->scikit-learn-extra) (0.16.0)
```

In [148]:
```
from sklearn_extra.cluster import KMedoids
```

In [149]:
```
%%time
kmc=KMedoids(n_clusters=4 , metric="manhattan",init="random")
```

```
Wall time: 0 ns
```

In [150]:

```
%time
kmc.fit_predict(du)
```

```
Wall time: 0 ns
```

Out[150]:

```
array([3, 3, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0],
      dtype=int64)
```

In [151]:

```
kmc.medoid_indices_
```

Out[151]:

```
array([16, 10,  3,  1])
```

In [152]:

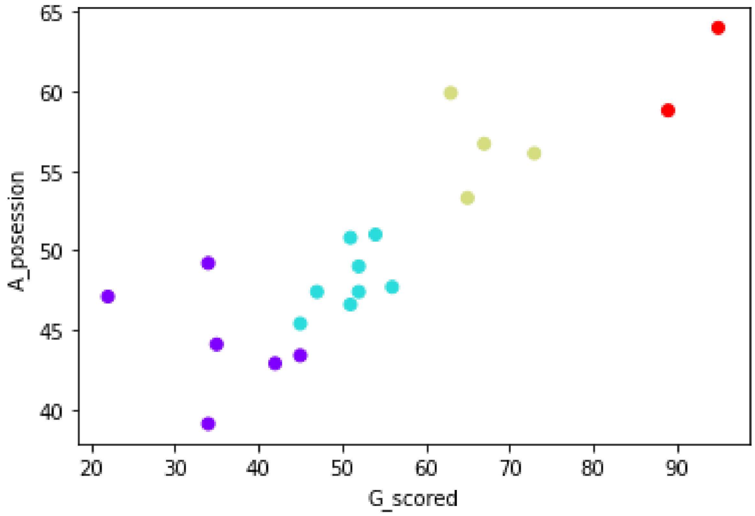
```
kmc.cluster_centers_
```

Out[152]:

```
array([[35. , 44.1],
       [52. , 47.4],
       [67. , 56.7],
       [89. , 58.8]])
```

In [161]:

```
plt.scatter(du[:,0],du[:,1],c=kmc.labels_,cmap="rainbow")
plt.xlabel("G_scored")
plt.ylabel("A_posession")
plt.show()
```



In [ ]:

```
```