africa_cup_nations

January 14, 2024

1 Africa cup of nations

It's happening !!!!

With data, we can explore this event considering the most important in the african continent

[1]:
```
import pandas as pd
import os
import sys
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.io as pio
import plotly
import numpy as np
```

[2]:
```
pio.renderers.keys()
pio.renderers.default = 'notebook'
```

[3]:
```
def read_csv_files(dir_name):
    """
    function that read files within a dir
    """
    curr_dir=os.getcwd()
    path=os.path.join(curr_dir,dir_name)
    files = [f for f in os.listdir(path) if os.path.isfile(os.path.
 ↪join(path,f))]
    dic_df={}
    i=0
    for file in files:
        dic_df[i]=pd.read_csv(os.path.join(path,file))
        i=i+1
    return dic_df,files
```

[4]:
```
dic_dfs,csv_names=read_csv_files("archive_2")
```

1.1 Data

We thought scrapping data from different websites covering the event

Hey :) Wonderful idea, this could be another **data engineering** project

This time, the data are already present in kaggle.

[5]:
```
csv_names
```

[5]:
```
['Africa Cup of Nations Matches.csv',
 'Africa Cup of Nations Players.csv',
 'Participated Teams General Statistics.csv',
 'Tournaments General Statistics.csv']
```

The data are covering the matches, the players involved, the team participated in this competition and general statistics

[6]:
```
match_df=dic_dfs[0]
match_df.head(20)
```

[6]:

	Year	Date	Time	Home Team Name	Away Team Name	\
0	1957	10 February 1957	NaN	Sudan	Egypt	
1	1957	10 February 1957	NaN	Ethiopia	South Africa	
2	1957	16 February 1957	NaN	Egypt	Ethiopia	
3	1959	22-May-59	NaN	Egypt	Ethiopia	
4	1959	25-May-59	NaN	Sudan	Ethiopia	
5	1959	29-May-59	NaN	Egypt	Sudan	
6	1962	14-Jan-62	NaN	Ethiopia	Tunisia	
7	1962	18-Jan-62	NaN	Egypt	Uganda	
8	1962	20-Jan-62	NaN	Tunisia	Uganda	
9	1962	21-Jan-62	NaN	Ethiopia	Egypt	
10	1963	24-Nov-63	NaN	Ghana	Tunisia	
11	1963	26-Nov-63	NaN	Ghana	Ethiopia	
12	1963	28-Nov-63	NaN	Ethiopia	Tunisia	
13	1963	24-Nov-63	NaN	Egypt	Nigeria	
14	1963	26-Nov-63	NaN	Egypt	Sudan	
15	1963	28-Nov-63	NaN	Sudan	Nigeria	
16	1963	30-Nov-63	NaN	Egypt	Ethiopia	
17	1963	1-Dec-63	NaN	Ghana	Sudan	
18	1965	12 November 1965	NaN	Tunisia	Ethiopia	
19	1965	14 November 1965	NaN	Senegal	Tunisia	

	Home Team Goals	Away Team Goals	Stage	\
0	1.0	2.0	Semifinals	
1	NaN	NaN	Semifinals	
2	4.0	0.0	Final	
3	4.0	0.0	Final Tournament	
4	1.0	0.0	Final Tournament	
5	2.0	1.0	Final Tournament	
6	4.0	2.0	Semifinals	
7	2.0	1.0	Semifinals	
8	3.0	0.0	Third place match	

9	4.0	2.0	Final
10	1.0	1.0	Group A
11	2.0	0.0	Group A
12	4.0	2.0	Group A
13	6.0	3.0	Group B
14	2.0	2.0	Group B
15	4.0	0.0	Group B
16	3.0	0.0	Third place match
17	3.0	0.0	Final
18	4.0	0.0	Group A
19	0.0	0.0	Group A

	Win Conditions	Stadium
0	NaN	Municipal Stadium
1	Ethiopia wins due to disqualification of othe…	NaN
2	NaN	Municipal Stadium
3	NaN	Prince Farouk Stadium
4	NaN	Prince Farouk Stadium
5	NaN	Prince Farouk Stadium
6	NaN	Hailé Sélassié Stadium
7	NaN	Hailé Sélassié Stadium
8	NaN	Hailé Sélassié Stadium
9	win after extra time	Hailé Sélassié Stadium
10	NaN	Accra Sports Stadium
11	NaN	Accra Sports Stadium
12	NaN	Accra Sports Stadium
13	NaN	Kumasi Sports Stadium
14	NaN	Kumasi Sports Stadium
15	NaN	Kumasi Sports Stadium
16	NaN	Accra Sports Stadium
17	NaN	Accra Sports Stadium
18	NaN	Stade Chedli Zouiten
19	NaN	Stade Chedli Zouiten

	City	Attendance
0	Khartoum	30000.0
1	NaN	NaN
2	Khartoum	30000.0
3	Cairo	30000.0
4	Cairo	20000.0
5	Cairo	30000.0
6	Addis Ababa	30000.0
7	Addis Ababa	30000.0
8	Addis Ababa	NaN
9	Addis Ababa	NaN
10	Accra	NaN
11	Accra	NaN

```
12         Accra           NaN
13        Kumasi           NaN
14        Kumasi           NaN
15        Kumasi           NaN
16         Accra           NaN
17         Accra           NaN
18         Tunis       16000.0
19         Tunis           NaN
```

[7]:
```
match_df.shape
```

[7]:
```
(622, 12)
```

[8]:
```
match_df.isnull().sum()
```

[8]:
```
Year                  0
Date                  0
Time                289
Home Team Name        0
Away Team Name        0
Home Team Goals       4
Away Team Goals       4
Stage                 0
Win Conditions      566
Stadium               4
City                  4
Attendance          100
dtype: int64
```

[9]:
```
match_df.dtypes
```

[9]:
```
Year                  int64
Date                 object
Time                 object
Home Team Name       object
Away Team Name       object
Home Team Goals     float64
Away Team Goals     float64
Stage                object
Win Conditions       object
Stadium              object
City                 object
Attendance          float64
dtype: object
```

[10]:
```
match_df.columns
```

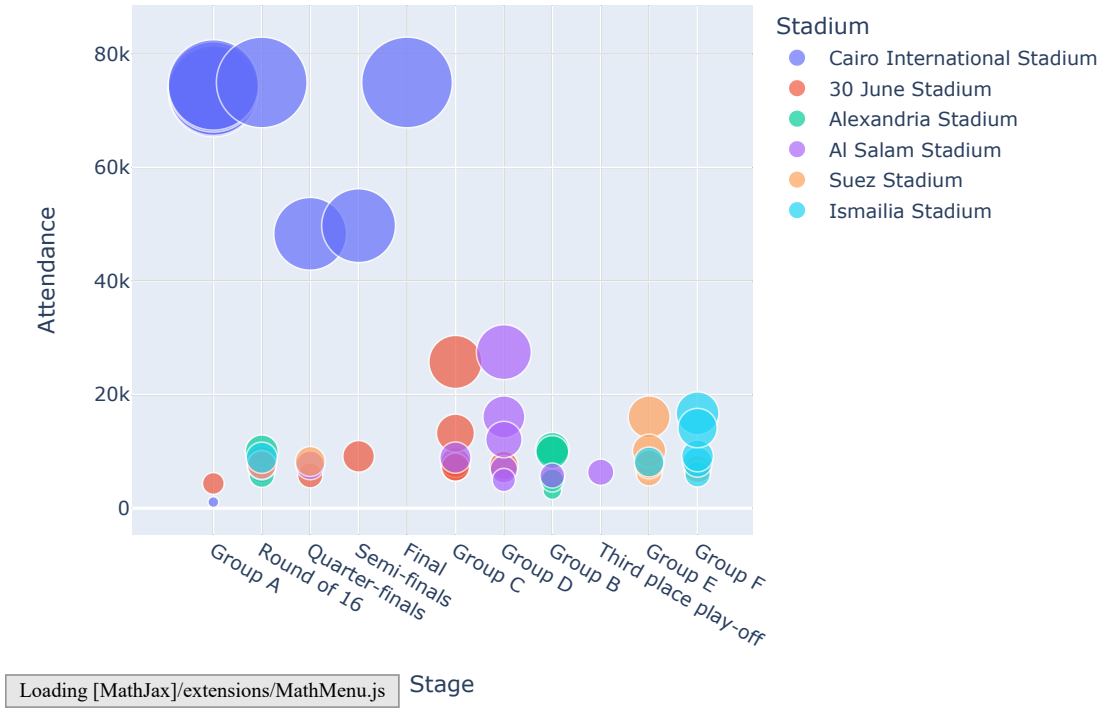
[10]:
```
Index([' Year ', ' Date ', ' Time ', ' Home Team Name ', ' Away Team Name ',
       ' Home Team Goals ', ' Away Team Goals ', ' Stage ', ' Win Conditions ',
       ' Stadium ', ' City ', ' Attendance '],
      dtype='object')
```

[11]:
```
match_grp= match_df.groupby(" Stage ")[[" Year "," Attendance "]]
match_grp= match_df.reset_index()
match_grp.head()
```

[11]:

	index	Year	Date	Time	Home Team Name	Away Team Name	Home Team Goals	Away Team Goals	Stage	Win Conditions	Stadium	City	Attendance
0	0	1957	10 February 1957	NaN	Sudan	Egypt	1.0	2.0	Semifinals	NaN	Municipal Stadium	Khartoum	30000.0
1	1	1957	10 February 1957	NaN	Ethiopia	South Africa	NaN	NaN	Semifinals	Ethiopia wins due to disqualification of othe…	NaN	NaN	NaN
2	2	1957	16 February 1957	NaN	Egypt	Ethiopia	4.0	0.0	Final	NaN	Municipal Stadium	Khartoum	30000.0
3	3	1959	22-May-59	NaN	Egypt	Ethiopia	4.0	0.0	Final Tournament	NaN	Prince Farouk Stadium	Cairo	30000.0
4	4	1959	25-May-59	NaN	Sudan	Ethiopia	1.0	0.0	Final Tournament	NaN	Prince Farouk Stadium	Cairo	20000.0

[12]:
```
plotly.offline.init_notebook_mode()
fig=px.scatter(match_df[match_df[" Year "] == 2019],y=" Attendance ",color="␣
 ↪Stadium ",x=" Stage ",
              hover_name=" City ",size=" Attendance ", size_max=40)
fig.show(renderer="notebook+pdf")
```



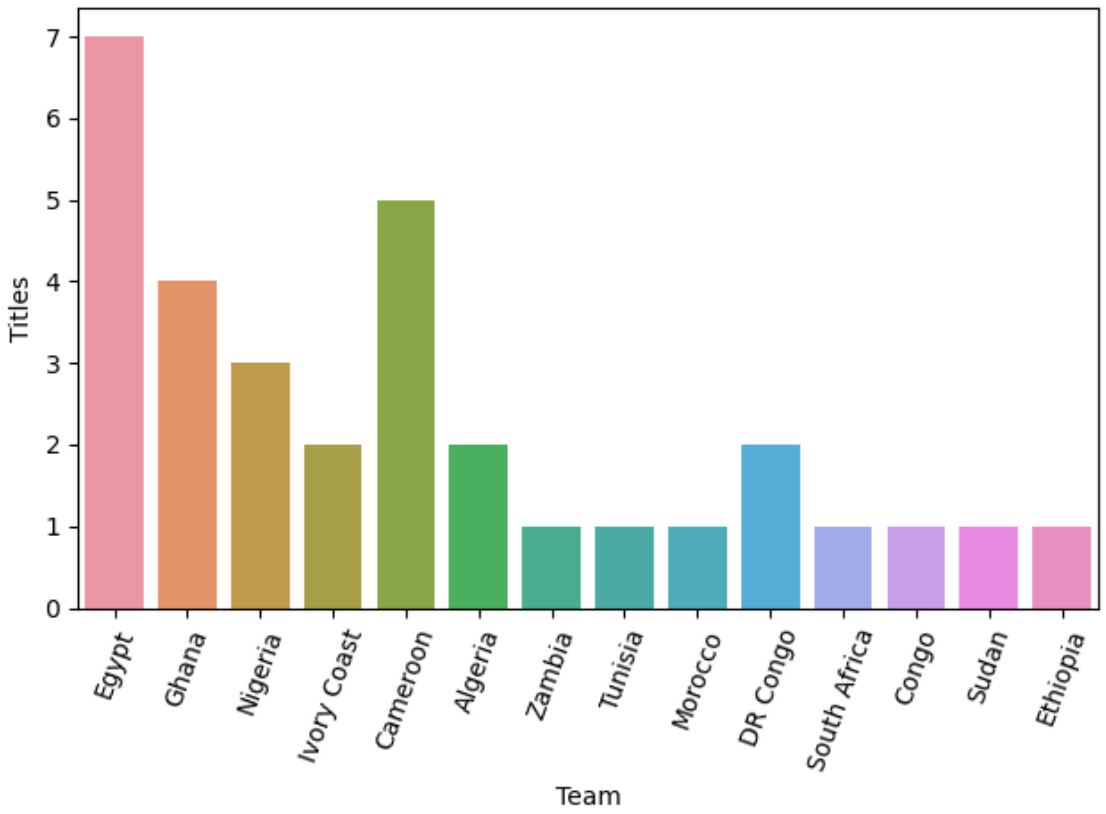
Loading [MathJax]/extensions/MathMenu.js

Having an idea where matches have taken place in **Egypt for the can 2019** with the number of attendance during the different competitions and the hosting stadium

1.2 The teams statistics won championship titles

[13]:
```
tournament_df=dic_dfs[2]
tournament_df.dropna(inplace=True)
```

[14]:
```
sns.barplot(x="Team",y="Titles",data=tournament_df)
plt.xticks(rotation=70)
plt.tight_layout()
```



[15]:
```
players_df=dic_dfs[1]
players_df.head()
```

[15]:

	Year	Shirt Number	Player Position	Player Name	Date of birth (age)	Caps	Goals	Club	Country
0	1957	NaN	GK	Ali Bakr	NaN	NaN	NaN	Zamalek	Egypt
1	1957	NaN	GK	Paraskos "Brascos" Trimeritis	NaN	NaN	NaN	El-Qanah	Egypt
2	1957	NaN	GK	Abdel-Galil Hameida	NaN	NaN	NaN	Al-Ahly	Egypt
3	1957	NaN	DF	Mosaad Daoud	NaN	NaN	NaN	El-Olympi	Egypt
4	1957	NaN	DF	El-Sayed El-Arabi	NaN	NaN	NaN	Teram	Egypt

[16]:
```
players_df.isnull().sum()
```

[16]:
```
Year                    0
Shirt Number         2466
```

```
 Player Position        389
 Player Name             42
Date of birth (age)    1620
Caps                   5445
Goals                  7142
Club                    805
Country                   0
dtype: int64
```

[17]:
```
players_df.dtypes
```

[17]:
```
 Year                   int64
 Shirt Number          object
 Player Position       object
 Player Name           object
Date of birth (age)    object
Caps                   object
Goals                  object
Club                   object
Country                object
dtype: object
```

1.2.1 Data cleaning

[18]:
```
players_df['Goals'] = players_df['Goals'].str.replace(r'\D', '', regex=True)
```

[19]:
```
players_df.shape
```

[19]:
```
(7534, 9)
```

[20]:
```
scored_df=players_df[~players_df['Goals'].isnull() ]
scored_df['Goals'].replace('', np.nan, inplace=True)
scored_df=scored_df.dropna(subset=['Goals'])
scored_df['Goals'] = scored_df['Goals'].apply(pd.to_numeric)
scored_df.head()
```

```
C:\Users\dabanti\AppData\Local\Temp\ipykernel_13224\1267407201.py:2:
SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[20]:
```
        Year   Shirt Number   Player Position       Player Name   \
5878    2013              1                GK  Wayne Sandilands
```

```
5879    2013              2                DF     Siboniso Gaxa
5880    2013              3                DF    Tsepo Masilela
5881    2013              4                DF     Thabo Nthethe
5882    2013              5                DF    Anele Ngcongca

                        Date of birth (age) Caps  Goals                Club  \
5878    (1983-08-23)23 August 1983 (aged 29)    2      0    Mamelodi Sundowns
5879      (1984-04-06)6 April 1984 (aged 28)   54      0        Kaizer Chiefs
5880        (1985-05-05)5 May 1985 (aged 27)   45      0        Kaizer Chiefs
5881    (1984-10-03)3 October 1984 (aged 28)    7      0  Bloemfontein Celtic
5882   (1987-10-20)20 October 1987 (aged 25)   22      0          Racing Genk

           Country
5878  South Africa
5879  South Africa
5880  South Africa
5881  South Africa
5882  South Africa
```

[21]:
```
scored_df=scored_df.sort_values(by='Goals', ascending=False).reset_index(drop =
↪True)

scored_df.head(10)
```

[21]:
```
    Year   Shirt Number   Player Position                    Player Name   \
0    2019             10                FW                   Mohamed Salah
1    2019             17                FW  Knowledge Musona  (  captain  )
2    2019              7                FW                   Emmanuel Okwi
3    2019             17                MF                     Farouk Miya
4    2019             10                FW    Mbwana Samatta  (  captain  )
5    2019             10                FW                      Sadio Mané
6    2019              7                MF                      Ahmed Musa
7    2019              8                MF                    Trésor Mputu
8    2019             14                FW                   Raphael Bocco
9    2019             11                MF                   Khama Billiat

                        Date of birth (age) Caps  Goals              Club  \
0        (1992-06-15)15 June 1992 (aged 27)   63    390         Liverpool
1        (1990-06-21)21 June 1990 (aged 29)   36    210  Sporting Lokeren
2    (1992-12-25)25 December 1992 (aged 26)   63    200             Simba
3    (1997-11-26)26 November 1997 (aged 21)   52    190            Gorica
4    (1992-12-23)23 December 1992 (aged 26)   47    170              Genk
5       (1992-04-10)10 April 1992 (aged 27)   56    150         Liverpool
6     (1992-10-14)14 October 1992 (aged 26)   80    150          Al Nassr
7   (1985-09-17)17 September 1985 (aged 33)   47    140        TP Mazembe
8       (1989-08-05)5 August 1989 (aged 29)   61    140             Simba
9      (1990-08-19)19 August 1990 (aged 28)   37    130     Kaizer Chiefs
```

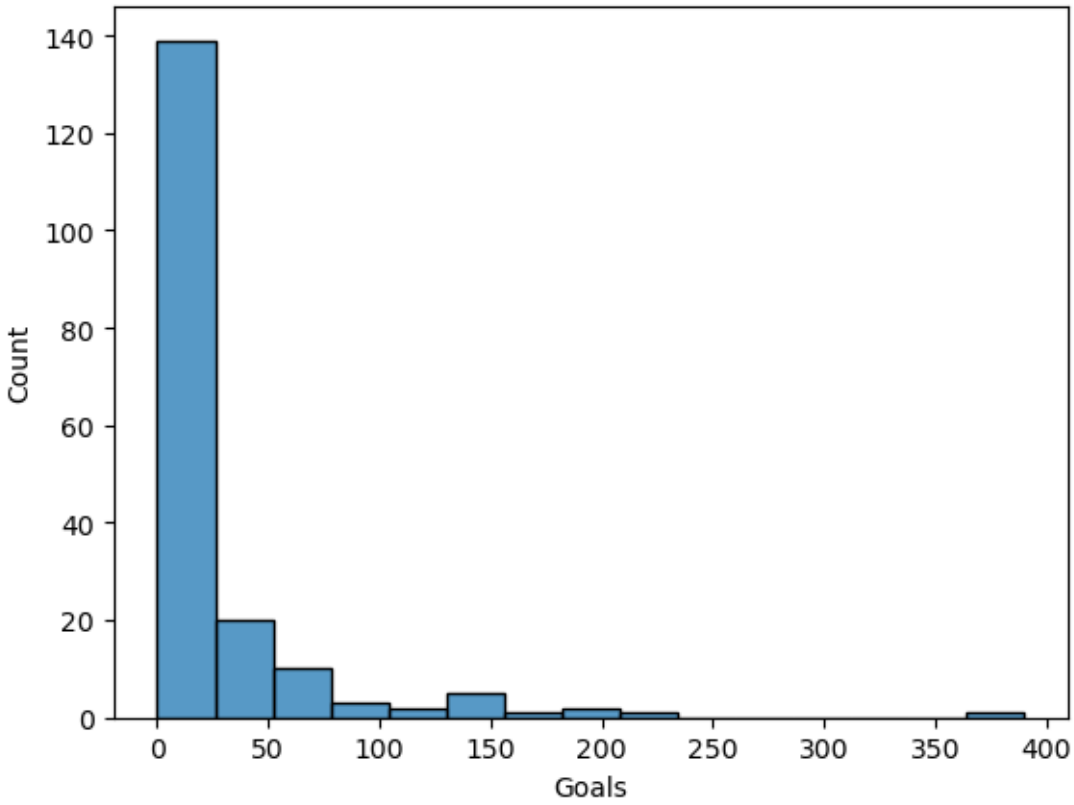
```
    Country
0     Egypt
1  Zimbabwe
2    Uganda
3    Uganda
4  Tanzania
5   Senegal
6   Nigeria
7  DR Congo
8  Tanzania
9  Zimbabwe
```

[22]:
```
scored_df.isnull().sum()
```

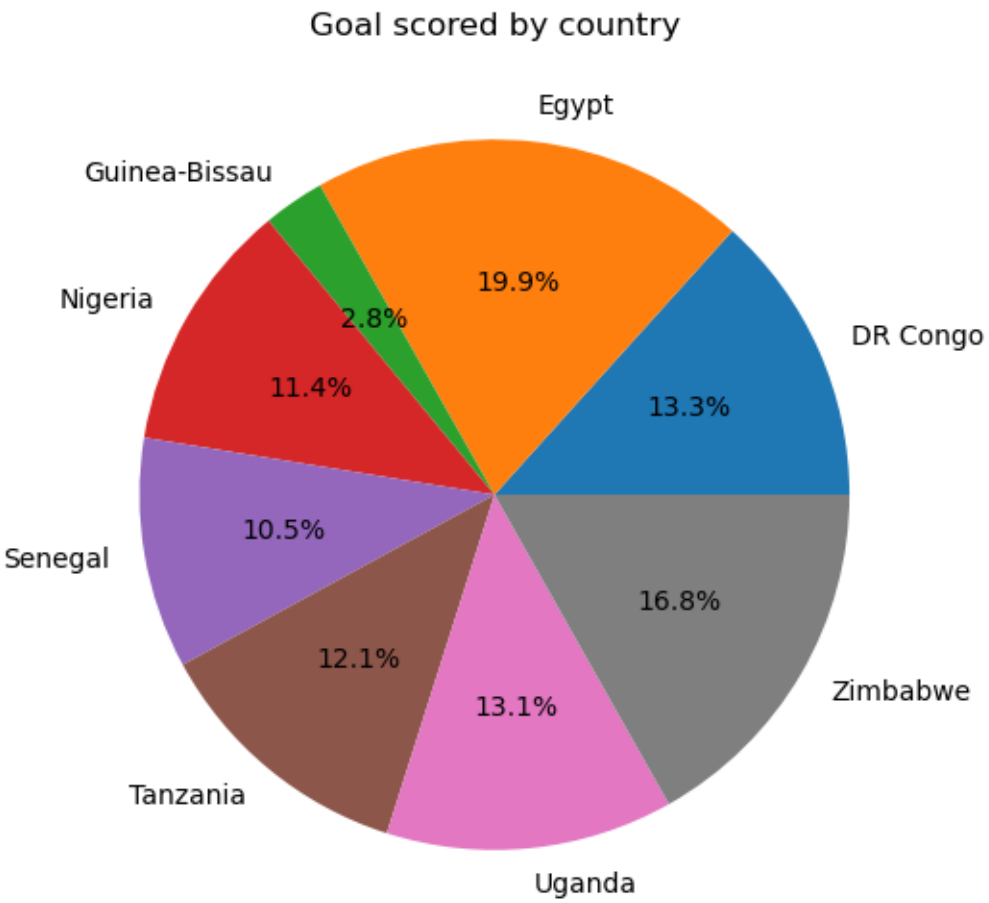
[22]:
```
 Year                   0
 Shirt Number           0
 Player Position        0
 Player Name            0
Date of birth (age)    92
Caps                    0
Goals                   0
Club                    0
Country                 0
dtype: int64
```

[23]:
```
sns.histplot(x="Goals",data=scored_df[scored_df[' Year '] == 2019],bins = 15)
```

[23]:
```
<AxesSubplot:xlabel='Goals', ylabel='Count'>
```



[24]:
```
plt.figure(figsize=(6,6))
df_pie=scored_df[scored_df[' Year '] == 2019].groupby("Country")['Goals'].
 ↪agg('sum')
plt.pie(df_pie, labels=df_pie.keys(),autopct='%1.1f%%')
plt.title('Goal scored by country')
plt.show()
```



[25]:
```
teams_df=dic_dfs[2]
teams_df.head()
```

[25]:

	Rank	Team	Titles	Part's	Pld	W	D	L	GF	GA	GD	Pts	Pts%
0	1	Egypt	7.0	24	100	57	17	26	164	88	76	188	62.7
1	2	Ghana	4.0	22	99	54	20	25	130	82	48	182	61.3
2	3	Nigeria	3.0	18	93	51	21	21	132	89	43	174	62.4
3	4	Ivory Coast	2.0	23	95	42	25	28	138	100	38	151	53.0
4	5	Cameroon	5.0	19	84	41	27	16	123	76	47	150	59.5

[26]:
```
teams_df.describe()
```

[26]:

	Rank	Titles	Part's	Pld	W	D	\
count	14.000000	14.000000	14.000000	14.000000	14.000000	14.000000	
mean	8.642857	2.285714	16.500000	67.428571	28.785714	18.285714	
std	5.904459	1.857565	5.543534	27.410594	17.493641	8.099654	
min	1.000000	1.000000	7.000000	24.000000	7.000000	3.000000	

25%	4.250000	1.000000	11.750000	47.750000	17.000000	14.000000
50%	7.500000	1.500000	18.000000	73.500000	25.000000	20.500000
75%	12.250000	2.750000	19.000000	90.750000	41.750000	23.750000
max	20.000000	7.000000	24.000000	100.000000	57.000000	29.000000

	L	GF	GA	GD	Pts	Pts%
count	14.000000	14.000000	14.000000	14.000000	14.000000	14.000000
mean	20.428571	89.214286	73.142857	25.928571	104.642857	48.350000
std	5.879747	44.625240	21.647018	21.734284	57.423671	10.623685
min	11.000000	27.000000	38.000000	3.000000	24.000000	29.600000
25%	16.250000	54.500000	58.750000	10.500000	66.750000	39.600000
50%	21.000000	90.500000	79.000000	15.000000	98.000000	48.550000
75%	25.000000	128.250000	88.750000	41.750000	150.750000	57.875000
max	29.000000	164.000000	102.000000	76.000000	188.000000	62.700000

[27]:
```
teams_scored_df = scored_df.merge(teams_df,left_on="Country", right_on="Team")
```

[28]:
```
teams_scored_df["Titles"].corr(teams_scored_df["Goals"])
```

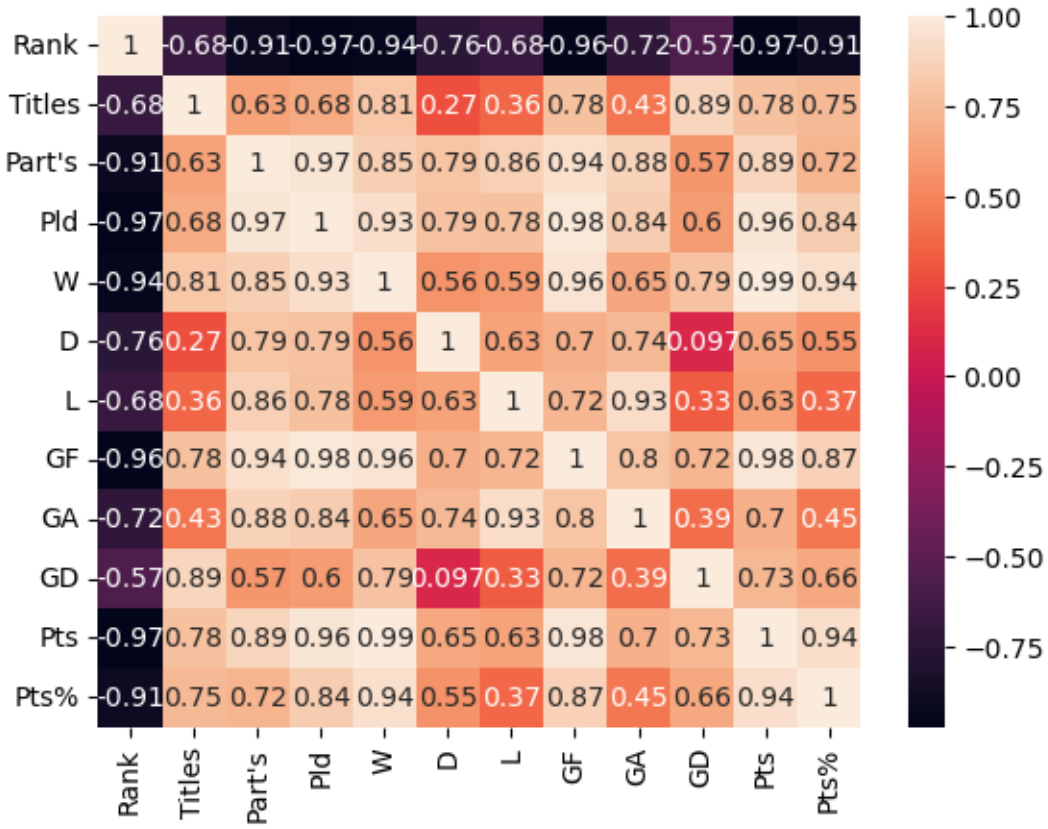
[28]: 0.25428508858245424

Trying to see if there is a correlation between winning many titles and the goal scored in the competition

[29]:
```
heatmap_df=teams_df.drop(["Team"], axis=1).corr()
```

[30]:
```
sns.heatmap(heatmap_df,annot=True)
```

[30]: <AxesSubplot:>



[31]:
```
participation_df=dic_dfs[3]
participation_df.head()
```

[31]:

	Year	Host	Champion (titles)	Winning coach	Top scorer (goals)	Most valuable player
0	1957	Sudan	Egypt (1)	Mourad Fahmy	Ad-Diba (5)	El Gohary
1	1959	Egypt	Egypt (2)	Pál Titkos	Mahmoud El Gohary (3)	El Deeba
2	1962	Ethiopia	Ethiopia (1)	Ydnekatchew Tessema	Mengistu Worku (3)	Mengistu Worku
3	1963	Ghana	Ghana (1)	Charles Gyamfi	Badawi Abdel Fattah (3)	Hassan El Shazly
4	1965	Tunisia	Ghana (2)	Charles Gyamfi	Hassan El Shazly (6)	Hassan El Shazly
					Ben Acheampong (3) Osei Kofi (3) Eustache …	Osei Kofi

[32]:
```
participation_df.isnull().sum()
```

[32]:
```
Year                    0
Host                    0
Champion (titles)       0
Winning coach           0
Top scorer (goals)      0
Most valuable player    0
dtype: int64
```

[33]:
```
participation_df.dtypes
```

[33]:
```
Year                     int64
Host                    object
Champion (titles)       object
Winning coach           object
Top scorer (goals)      object
Most valuable player    object
dtype: object
```

[34]:
```
import scipy.cluster.hierarchy as sch
```

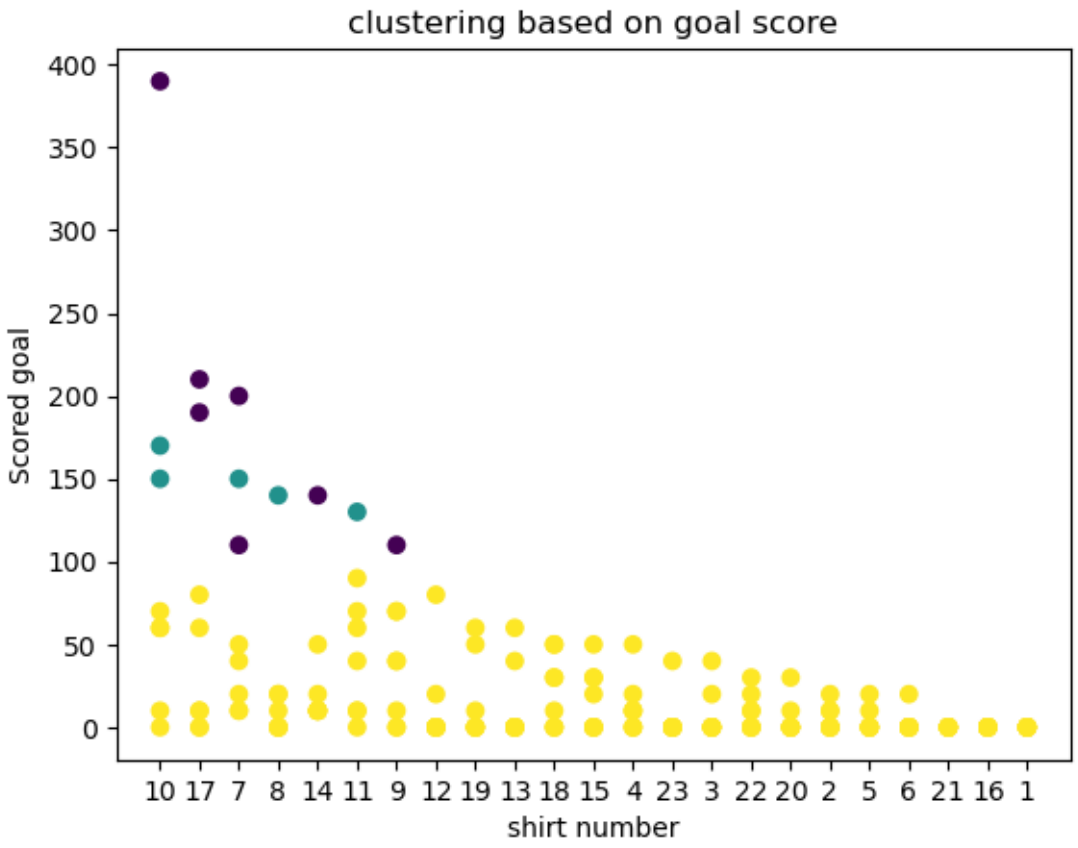
2 Clustering

[35]:
```
df=scored_df[scored_df[' Year '] == 2019]
df=df[[' Shirt Number ','Goals']]
```

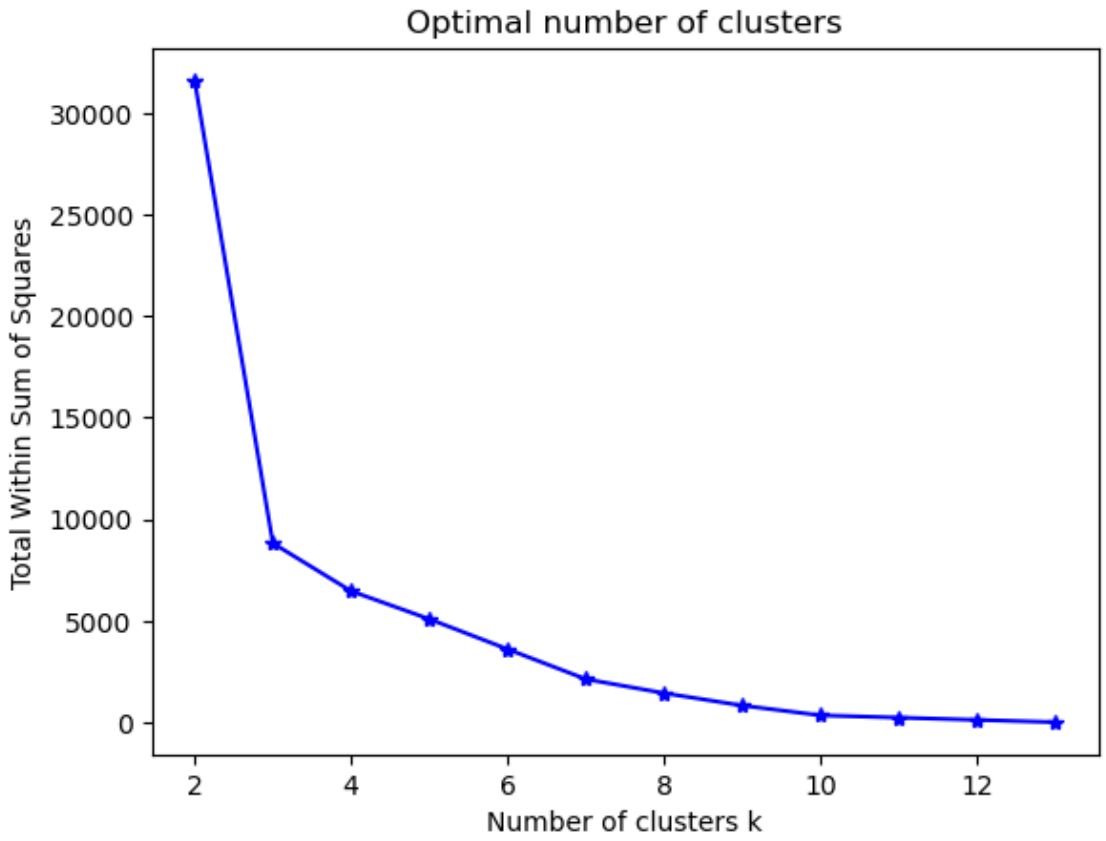
In this context and for now, data gathered might not serve on prediction of the next winning team. Though we can group them based on similar characteristics after having applied different skillsets of EDA in the first section

[36]:
```
from sklearn.cluster import KMeans, DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn.decomposition import PCA

X = StandardScaler().fit_transform(df)

y_pred = DBSCAN(eps =0.5,min_samples=5,metric='euclidean').fit_predict(X)
plt.scatter(df[' Shirt Number '],df['Goals'],c = y_pred)
plt.xlabel("shirt number")
plt.ylabel("Scored goal")
plt.title("clustering based on goal score")
```

[36]:
```
Text(0.5, 1.0, 'clustering based on goal score')
```



[37]:
```
data=teams_df.drop(["Team"], axis=1)
wss = []

K = range(2,14)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=123)
    kmeans = kmeans.fit(data)
    wss.append(kmeans.inertia_)
plt.plot(K, wss, "b*-")
plt.xlabel("Number of clusters k")
plt.ylabel("Total Within Sum of Squares")
plt.title("Optimal number of clusters")
plt.show()
```



[38]:
```
def find_optimal_cluster_number_kmeans(data, lower_bound, upper_bound,␣
↪random_state):

    "Find optimal number of cluster according to silhouette score."

    silhouette_average = []
    K = range(lower_bound, upper_bound)

    for k in K:
        kmeans = KMeans(n_clusters=k, random_state=random_state)
        cluster_labels=kmeans.fit_predict(data)
        silhouette_score = metrics.silhouette_score(data, cluster_labels)
        silhouette_average.append([k, silhouette_score])

    silhouette_average = np.array(silhouette_average)
    print("n_clusters =", int(silhouette_average[np.argmax(silhouette_average[:
↪,1:2])][0]),
          "The average silhouette_score is : %.4f" % silhouette_average[np.
↪argmax(silhouette_average[:,1:2])][1])
```

[39]:
```
find_optimal_cluster_number_kmeans(data, 2, 14, random_state=123)
```

```
n_clusters = 3 The average silhouette_score is : 0.6311
```

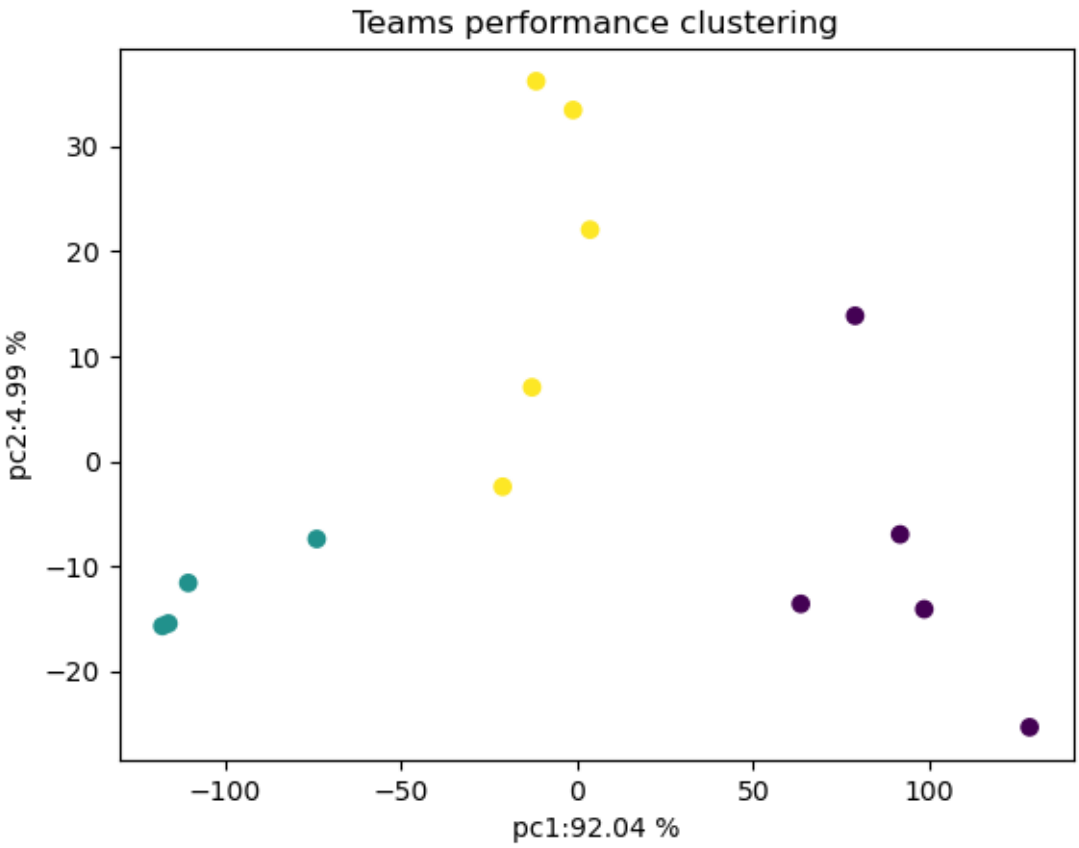
[40]:
```
pca = PCA()
pca_data = pca.fit_transform(data)
pca_data = pd.DataFrame(pca_data, columns=["pc"+str(i+1) for i in␣
 ↪range(len(data.columns))])
```

[41]:
```
print("pca.explained variance ratio:\n ", " ".join(map("{:.3f}".format, pca.
 ↪explained_variance_ratio_)))
```

```
pca.explained variance ratio:
  0.920 0.050 0.023 0.004 0.002 0.001 0.000 0.000 0.000 0.000 0.000 0.000
```

[42]:
```
pca_data1 = pca_data[["pc1","pc2"]].copy()
data1 = data.copy()
kmeans = KMeans(n_clusters=3, random_state=2464063)
data1["clusters"] = kmeans.fit_predict(data1)
```

[43]:
```
plt.scatter(pca_data1["pc1"], pca_data1["pc2"], c=data1.clusters)
plt.title("Teams performance clustering")
plt.xlabel("pc1:" + "{:.2f}".format(pca.explained_variance_ratio_[0] * 100) + "␣
 ↪%")
plt.ylabel("pc2:" + "{:.2f}".format(pca.explained_variance_ratio_[1] * 100) + "␣
 ↪%")
plt.show()
```



Finally, we have finished this work by testing some required data skills demanded in day to day job. **EDA python, DataFrame manipulation ML**

2.1 What's next ?

1) Thinking about data communications
2) Based on this, predict scenarios and outcomes from first match to the final. Beating 1xBet
3) Scraping realtime data by implementing a data mining process facilitating (2)
4) Create a compelling data story telling

[ ]: