

In [1]:

```
# Group stats
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [4]:

```
g_stats = pd.read_csv("group_stats.csv")
g_stats.head()
```

Out[4]:

	Unnamed: 0	group	rank	team	matches_played	wins	draws	losses	goals_scored	g
0	0	1	1	Netherlands	3	2	1	0	5	
1	1	1	2	Senegal	3	2	0	1	5	
2	2	1	3	Ecuador	3	1	1	1	4	
3	3	1	4	Qatar	3	0	0	3	1	
4	4	2	1	England	3	2	1	0	9	

In [5]:

```
g_stats.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 16 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Unnamed: 0                            32 non-null    int64
1   group                                32 non-null    int64
2   rank                                 32 non-null    int64
3   team                                 32 non-null    object
4   matches_played                       32 non-null    int64
5   wins                                 32 non-null    int64
6   draws                               32 non-null    int64
7   losses                               32 non-null    int64
8   goals_scored                         32 non-null    int64
9   goals_against                       32 non-null    int64
10  goal_difference                      32 non-null    int64
11  points                              32 non-null    int64
12  expected_goal_scored                32 non-null    float64
13  exp_goal_conceded                   32 non-null    float64
14  exp_goal_difference                  32 non-null    float64
15  exp_goal_difference_per_90          32 non-null    float64
dtypes: float64(4), int64(11), object(1)
memory usage: 4.1+ KB
```

In [6]:

```
g_stats.describe()
```

Out[6]:

	Unnamed: 0	group	rank	matches_played	wins	draws	losses	goals_scored
count	32.000000	32.000000	32.000000	32.0	32.000000	32.000000	32.000000	32.000000
mean	15.500000	4.500000	2.500000	3.0	1.187500	0.625000	1.187500	1.187500
std	9.380832	2.327951	1.135924	0.0	0.692704	0.609071	0.780302	0.780302
min	0.000000	1.000000	1.000000	3.0	0.000000	0.000000	0.000000	0.000000
25%	7.750000	2.750000	1.750000	3.0	1.000000	0.000000	1.000000	1.000000
50%	15.500000	4.500000	2.500000	3.0	1.000000	1.000000	1.000000	1.000000
75%	23.250000	6.250000	3.250000	3.0	2.000000	1.000000	2.000000	2.000000
max	31.000000	8.000000	4.000000	3.0	2.000000	2.000000	3.000000	3.000000

In [8]:

```
g_stats.isna().sum()
```

Out[8]:

```
Unnamed: 0      0
group           0
rank            0
team            0
matches_played  0
wins            0
draws           0
losses          0
goals_scored    0
goals_against   0
goal_difference  0
points          0
expected_goal_scored  0
exp_goal_conceded  0
exp_goal_difference  0
exp_goal_difference_per_90  0
dtype: int64
```

In [13]:

```
g_stats['Goalspermatch']=(g_stats['goals_scored']/g_stats['matches_played']).astype(int)
g_stats['pointspermatch']=(g_stats['points']/g_stats['matches_played']).astype(float)
print(g_stats.head())
```

	Unnamed: 0	group	rank	team	matches_played	wins	draws	losses
\								
0	0	1	1	Netherlands	3	2	1	0
1	1	1	2	Senegal	3	2	0	1
2	2	1	3	Ecuador	3	1	1	1
3	3	1	4	Qatar	3	0	0	3
4	4	2	1	England	3	2	1	0

	goals_scored	goals_against	goal_difference	points	expected_goal_score
d \					
0	5	1	4	7	2.
4					
1	5	4	1	6	3.
8					
2	4	3	1	4	3.
7					
3	1	7	-6	0	1.
4					
4	9	2	7	7	5.
2					

	exp_goal_conceded	exp_goal_difference	exp_goal_difference_per_90	\
0	2.7	-0.3	-0.11	
1	2.5	1.3	0.43	
2	2.6	1.2	0.39	
3	3.5	-2.1	-0.71	
4	2.3	2.9	0.97	

	Goalspermatch	pointspermatch
0	1	2.333333
1	1	2.000000
2	1	1.333333
3	0	0.000000
4	3	2.333333

In [15]:

```
#total goals
total_goals= g_stats['goals_scored'].sum()
print(total_goals)
```

120

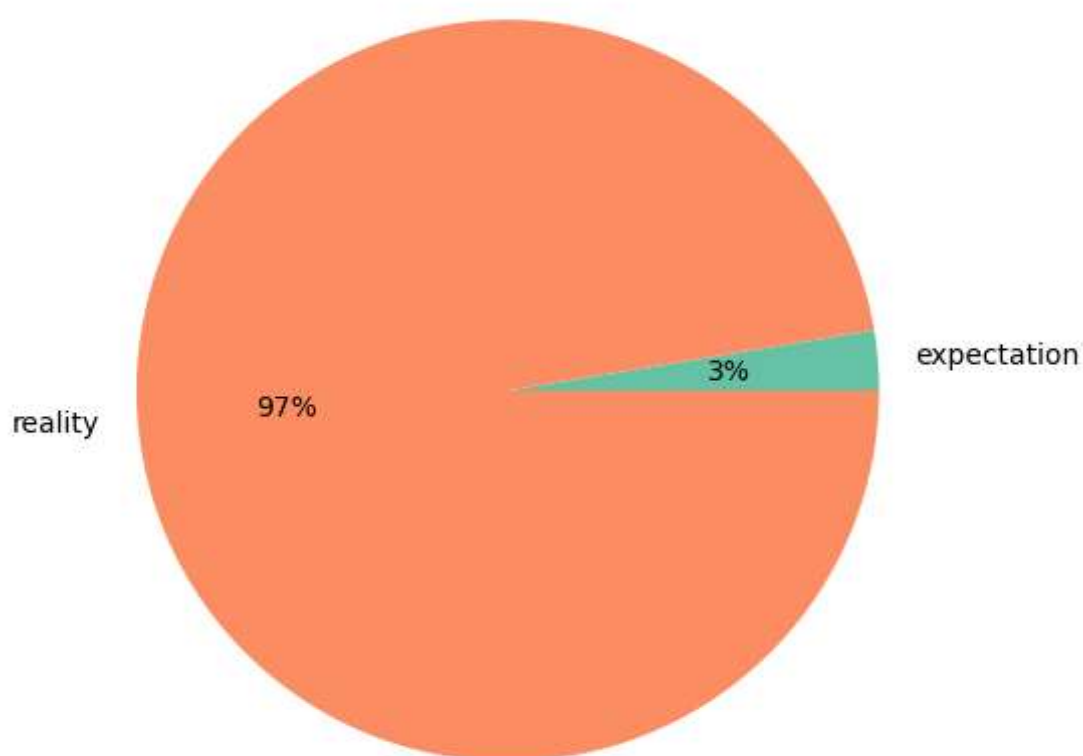
In [24]:

```
#Total draws
led_to_more_darama =g_stats['draws'].sum()
print(led_to_more_darama)
```

20

In [23]:

```
#pie chart on expected vs the actual result
plt.figure(figsize=(13,6))
in_game = g_stats['expected_goal_scored'].sum() - total_goals
data =[in_game,total_goals]
labels= ['expectation','reality']
color = sns.color_palette('Set2')
plt.pie(data, labels =labels , colors= color , autopct='%.0f%%')
plt.show()
```



In [25]:

```
#Groups available
g_stats['group'].unique()
```

Out[25]:

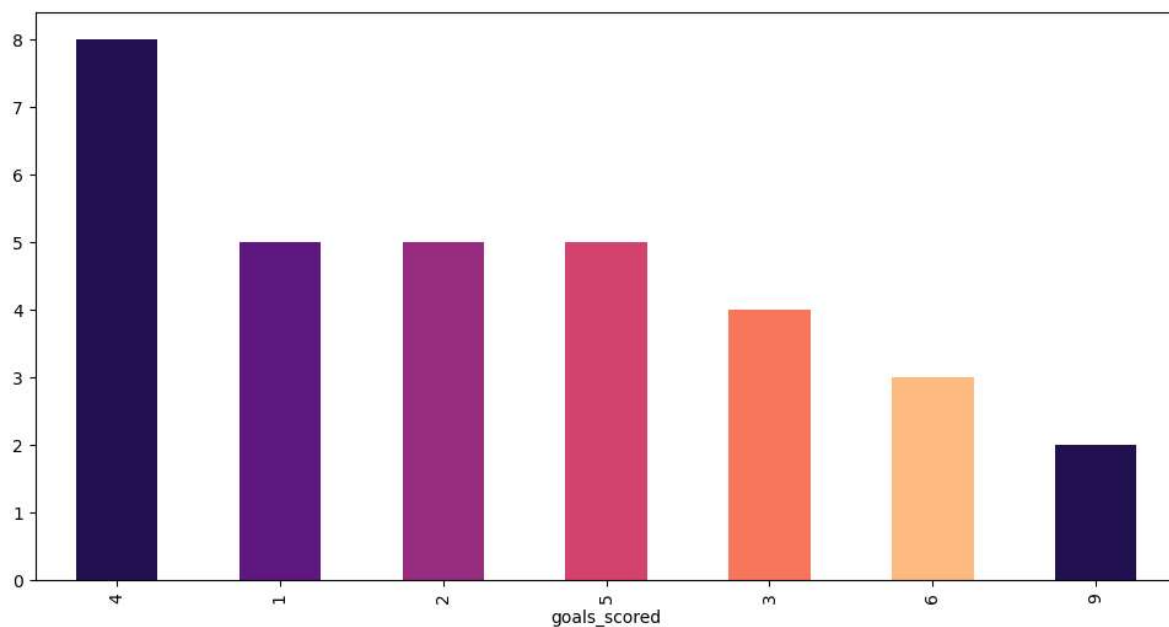
```
array([1, 2, 3, 4, 5, 6, 7, 8], dtype=int64)
```

In [48]:

```
#most goals scored by a team
top_g = g_stats.groupby('goals_scored').size().sort_values(ascending=False)
top_g.head(10).plot(kind='bar',figsize=(12,6),color=sns.color_palette("magma"))
```

Out[48]:

<AxesSubplot: xlabel='goals\_scored'>

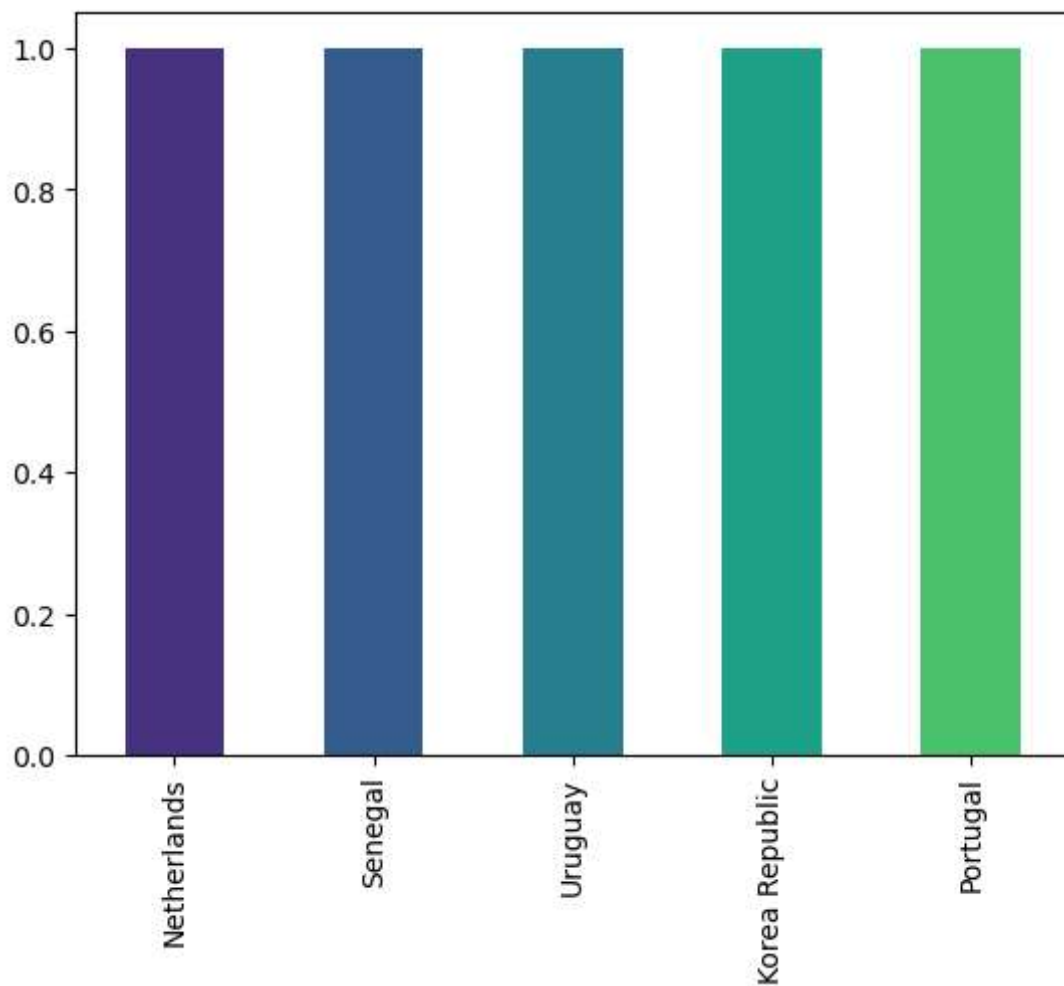


In [50]:

```
g_stats['team'].value_counts().nlargest(5).plot(kind='bar',color=sns.color_palette("viridi
```

Out[50]:

<AxesSubplot: >



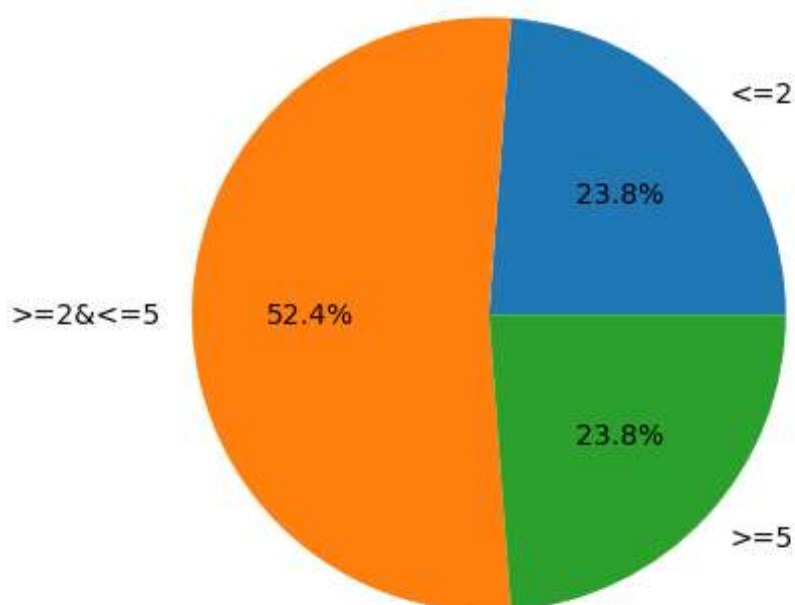
In [70]:

```
goal_under= g_stats[g_stats['goals_scored']<=2]
goal_normal=g_stats[(g_stats['goals_scored']>=2)& (g_stats['goals_scored']<=5)]
goal_above=g_stats[g_stats['goals_scored']>=5]
```

In [78]:

```
x= np.array([goal_under['team'].count(),goal_normal['team'].count(),goal_above['team'].count()])
mylables=["<=2",">=2<=5",">=5"]
plt.title('average goals scored around',fontsize=20)
plt.pie(x,labels=mylables,autopct="%.1f%")
plt.show()
```

average goals scored around

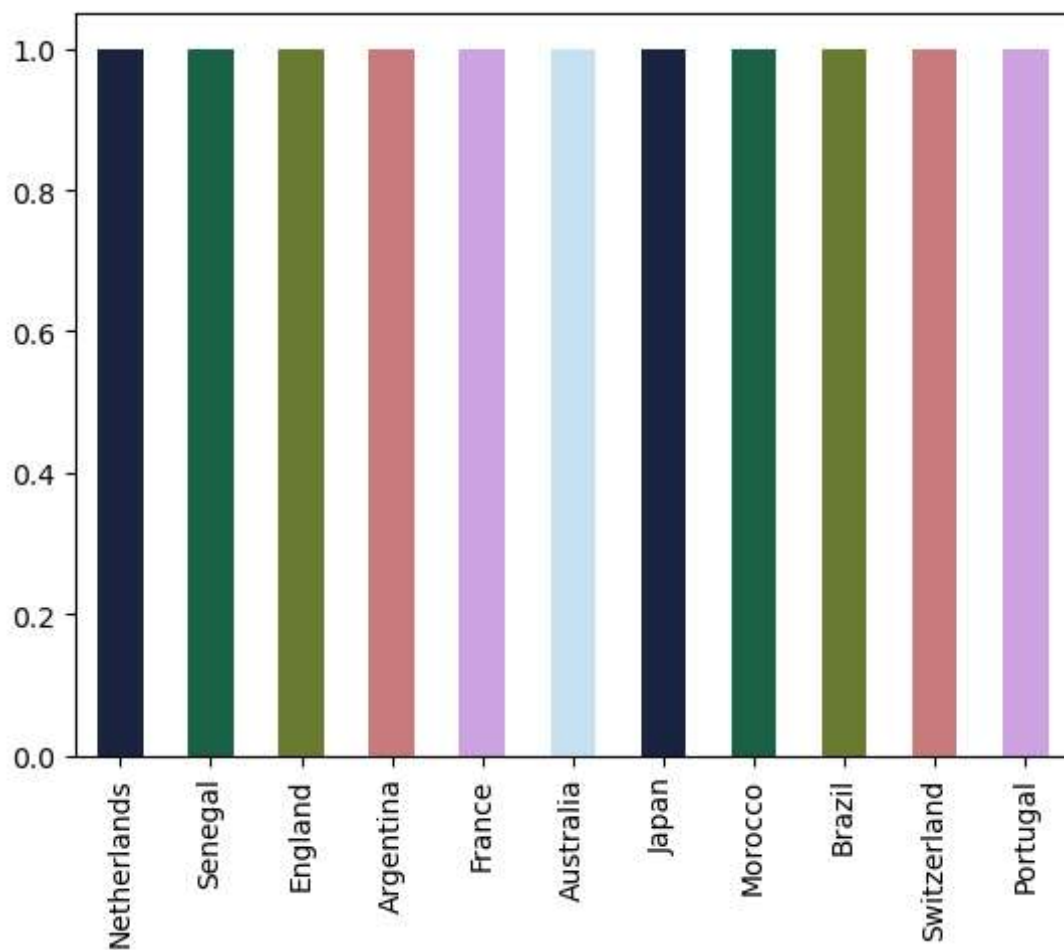


In [91]:

```
#team who won the spot for the next round with flying colors  
team_win = g_stats[g_stats['wins']==2]  
team_win['team'].value_counts().plot(kind='bar',color= sns.color_palette("cubehelix"))
```

Out[91]:

<AxesSubplot: >





In [94]:

```
# Teams who failed to qualify with flying colors
goal_under[goal_under["points"]<=2]
```

Out[94]:

Unnamed: 0	group	rank	team	matches_played	wins	draws	losses	goals_scored	goals_conceded
3	3	1	4	Qatar	3	0	0	3	1
7	7	2	4	Wales	3	0	1	2	1
15	15	4	4	Denmark	3	0	1	2	1
23	23	6	4	Canada	3	0	0	3	2

In [100]:

```
#
plt.figure(figsize=(12,6))
sns.boxplot(x='team', y='goals_scored', data= g_stats)
plt.xticks(rotation=90)
```

Out[100]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]),
 [Text(0, 0, 'Netherlands'),
  Text(1, 0, 'Senegal'),
  Text(2, 0, 'Ecuador'),
  Text(3, 0, 'Qatar'),
  Text(4, 0, 'England'),
  Text(5, 0, 'United States'),
  Text(6, 0, 'IR Iran'),
  Text(7, 0, 'Wales'),
  Text(8, 0, 'Argentina'),
  Text(9, 0, 'Poland'),
  Text(10, 0, 'Mexico'),
  Text(11, 0, 'Saudi Arabia'),
  Text(12, 0, 'France'),
  Text(13, 0, 'Australia'),
  Text(14, 0, 'Tunisia')])
```

In [99]:

```
num_team =g_stats.groupby('team').size()
data =(g_stats.groupby('team')['goal_difference'].sum())/num_team
data.sort_values(ascending =False)
```

Out[99]:

team	
England	7.0
Spain	6.0
Netherlands	4.0
Croatia	3.0
France	3.0
Morocco	3.0
Argentina	3.0
Brazil	2.0
Portugal	2.0
Switzerland	1.0
United States	1.0
Germany	1.0
Senegal	1.0
Japan	1.0
Ecuador	1.0
Tunisia	0.0
Uruguay	0.0
Poland	0.0
Korea Republic	0.0
Cameroon	0.0
Mexico	-1.0
Australia	-1.0
Belgium	-1.0
Saudi Arabia	-2.0
Ghana	-2.0
Denmark	-2.0
IR Iran	-3.0
Serbia	-3.0
Canada	-5.0
Wales	-5.0
Qatar	-6.0
Costa Rica	-8.0

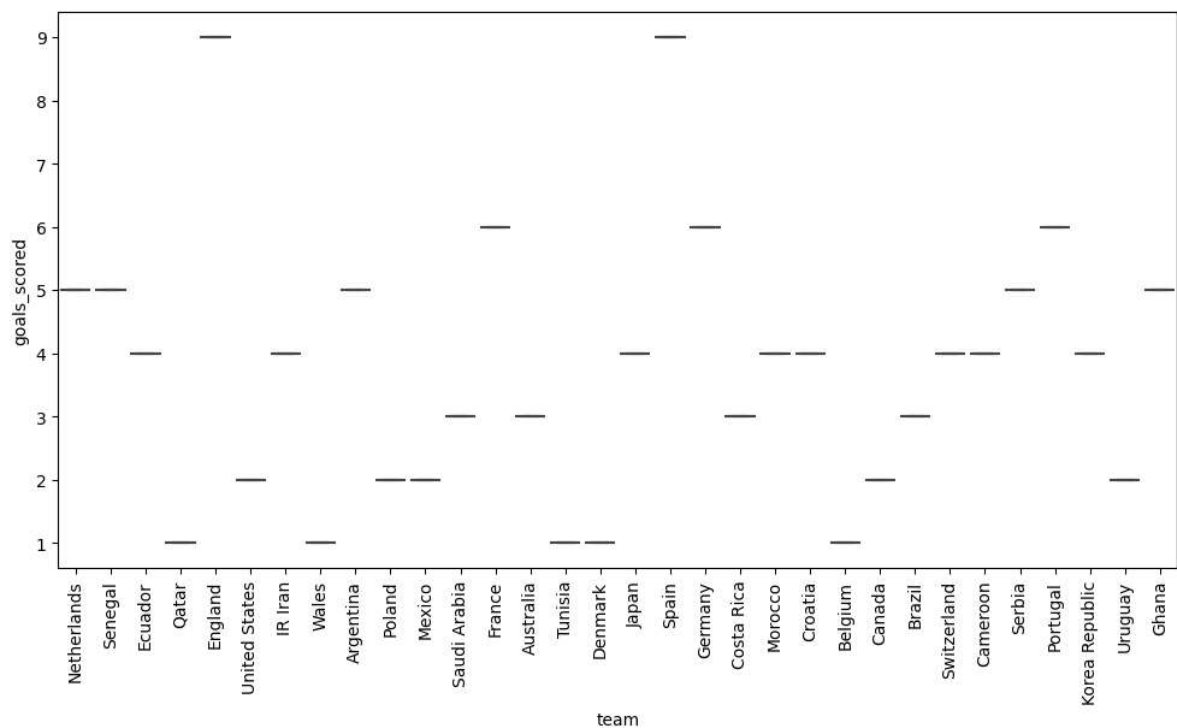
dtype: float64

In [101]:

```
#  
plt.figure(figsize=(12,6))  
sns.boxplot(x='team', y='goals_scored', data= g_stats)  
plt.xticks(rotation=90)
```

Out[101]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,  
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]),  
[Text(0, 0, 'Netherlands'),  
 Text(1, 0, 'Senegal'),  
 Text(2, 0, 'Ecuador'),  
 Text(3, 0, 'Qatar'),  
 Text(4, 0, 'England'),  
 Text(5, 0, 'United States'),  
 Text(6, 0, 'IR Iran'),  
 Text(7, 0, 'Wales'),  
 Text(8, 0, 'Argentina'),  
 Text(9, 0, 'Poland'),  
 Text(10, 0, 'Mexico'),  
 Text(11, 0, 'Saudi Arabia'),  
 Text(12, 0, 'France'),  
 Text(13, 0, 'Australia'),  
 Text(14, 0, 'Tunisia'),  
 Text(15, 0, 'Denmark'),  
 Text(16, 0, 'Japan'),  
 Text(17, 0, 'Spain'),  
 Text(18, 0, 'Germany'),  
 Text(19, 0, 'Costa Rica'),  
 Text(20, 0, 'Morocco'),  
 Text(21, 0, 'Croatia'),  
 Text(22, 0, 'Belgium'),  
 Text(23, 0, 'Canada'),  
 Text(24, 0, 'Brazil'),  
 Text(25, 0, 'Switzerland'),  
 Text(26, 0, 'Cameroon'),  
 Text(27, 0, 'Serbia'),  
 Text(28, 0, 'Portugal'),  
 Text(29, 0, 'Korea Republic'),  
 Text(30, 0, 'Uruguay'),  
 Text(31, 0, 'Ghana')])
```



In [ ]: