worldcup

April 30, 2023

1 2022 Qatar World Cup Analysis

The World Cup is one of the, if not the largest, sporting event in the world. The World Cup is a tournament consisting of some of the best qualifying nations in international football. Countries are divided into groups, where the top two teams from each group move on to the 'knockout rounds', effectively dwindling down the competition until one team is left standing.The World Cup is a global event that transcends borders, cultures, and languages, captivating audiences with the sheer talent, passion, and dedication of the world's top footballers. The World Cup also serves as a platform for countries to showcase their national identity and pride, and to unite their citizens around a common goal. Whether you are a casual fan or a die-hard supporter, the World Cup is an unforgettable experience that captures the imagination and inspires a sense of unity that extends far beyond the pitch. And for the players, it is considered the highest honor in the sport; one that demands a lot of passion and excellence in order to persevere and bring home glory to their country.

The 2022 World Cup in Qatar has recently concluded with an Argentinian extra-time victory in the final over France; a dramatic victory that perfectly captures the essence of the tournament. The dataset that we will be utilizing in our project focuses on each individual match from the tournament, and contains a large amount of data about the match itself, specifically for each team; possession, shots attempted, shots on goal, total passes, etc. We are hoping to showcase some trends, and insights that can summarize and visualize the tournament effectively through data.

Our goal is to provide some background on the tournament; the teams, the players, the matches, and the results. We will be using the dataset to answer some questions that we have about the tournament, and to provide some insights that can help us understand the tournament better.

Outside our background analysis, some questions we hope to answer are:

- Who were the most standout players in the tournament?
  - Who was the most efficient player throughout the tournament? (we can calculate some statistics for this, such as G/A per 90, and goals per shot attempted, etc.)
  - Which goalkeeper was most responsible for their team's success? (we can calculate some statistics for this, such as saves per 90, and goals conceded per shot on target, etc.)
- As the rounds go on, do teams tend to play more conservatively, or do they play more aggressively? (this can be measured by shots attempted, possession, etc.)
- We already know that the final was between Argentina and France, but which teams were the most dominant throughout the tournament?

1.1 Importing packages

```
[ ]: %matplotlib inline
import warnings
warnings.simplefilter(action='ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotnine import *
```

1.2 Loading, cleaning, and exploring datasets

We have three main datasets which we will be working with in our project; our main dataset is the `matches` dataset which contains information about the matches played in the 2022 Qatar World Cup. The second dataset, `countries`, relates to the individual country statistics, which we will be aggregating from our information about each match in order to get a better understanding of the teams and their performance. Lastly, we are going to take a look at a `players` dataset, which contains much information about the actual participants in the tournament, and their performance in the matches.

1.2.1 Matches Dataset - loading and cleaning

```
[ ]: matches = pd.read_csv('cup.csv')
matches.head()
```

```
[ ]:           team1         team2 possession team1 possession team2  \
0          QATAR       ECUADOR              42%              50%
1        ENGLAND          IRAN              72%              19%
2        SENEGAL   NETHERLANDS              44%              45%
3  UNITED STATES         WALES              51%              39%
4      ARGENTINA  SAUDI ARABIA              64%              24%

  possession in contest  number of goals team1  number of goals team2  \
0                    8%                      0                      2
1                    9%                      6                      2
2                   11%                      0                      2
3                   10%                      1                      1
4                   12%                      1                      2

           date     hour category  …  penalties scored team1  \
0  20 NOV 2022  17 : 00  Group A  …                       0
1  21 NOV 2022  14 : 00  Group B  …                       0
2  21 NOV 2022  17 : 00  Group A  …                       0
3  21 NOV 2022  20 : 00  Group B  …                       0
```

```
4  22 NOV 2022  11 : 00  Group C  …                        1

   penalties scored team2  goal preventions team1  goal preventions team2  \
0                       1                       6                       5
1                       1                       8                      13
2                       0                       9                      15
3                       1                       7                       7
4                       0                       4                      14

   own goals team1  own goals team2  forced turnovers team1  \
0                0                0                      52
1                0                0                      63
2                0                0                      63
3                0                0                      81
4                0                0                      65

   forced turnovers team2  defensive pressures applied team1  \
0                      72                               256
1                      72                               139
2                      73                               263
3                      72                               242
4                      80                               163

   defensive pressures applied team2
0                               279
1                               416
2                               251
3                               292
4                               361

[5 rows x 88 columns]
```

[ ]:
```
matches = matches[[
    'team1',
    'team2',
    'possession team1',
    'possession team2',
    'possession in contest',
    'number of goals team1',
    'number of goals team2',
    'category',
    'total attempts team1',
    'total attempts team2',
    'conceded team1',
    'conceded team2',
    'goal inside the penalty area team1',
    'goal inside the penalty area team2',
```

```
    'goal outside the penalty area team1',
    'goal outside the penalty area team2',
    'assists team1',
    'assists team2',
    'yellow cards team1',
    'yellow cards team2',
    'red cards team1',
    'red cards team2',
    'fouls against team1',
    'fouls against team2',
    'offsides team1',
    'offsides team2',
    'passes team1',
    'passes team2',
    'passes completed team1',
    'passes completed team2',
    'crosses team1',
    'crosses team2',
    'crosses completed team1',
    'crosses completed team2',
    'corners team1',
    'corners team2',
    'free kicks team1',
    'free kicks team2',
    'penalties scored team1',
    'penalties scored team2',
    'goal preventions team1',
    'goal preventions team2',
    'own goals team1',
    'own goals team2',
    'forced turnovers team1',
    'forced turnovers team2'
]]

matches['team1'] = matches['team1'].str.title()
matches['team2'] = matches['team2'].str.title()

matches['GroupID'] = matches['category'].apply(lambda x: x[-1] if 'Group' in x␣
 ↪else np.nan)
matches['category'] = matches['category'].apply(lambda x: x[:-2] if 'Group' in␣
 ↪x else x)

matches['possession team1'] = matches['possession team1'].str[:-1].astype(int)
matches['possession team2'] = matches['possession team2'].str[:-1].astype(int)
matches['possession in contest'] = matches['possession in contest'].str[:-1].
 ↪astype(int)
```

```
matches.head()
```

[ ]:

	team1	team2	possession team1	possession team2	\
0	Qatar	Ecuador	42	50	
1	England	Iran	72	19	
2	Senegal	Netherlands	44	45	
3	United States	Wales	51	39	
4	Argentina	Saudi Arabia	64	24	

	possession in contest	number of goals team1	number of goals team2	\
0	8	0	2	
1	9	6	2	
2	11	0	2	
3	10	1	1	
4	12	1	2	

	category	total attempts team1	total attempts team2	…	free kicks team2	\
0	Group	5	6	…	17	
1	Group	13	8	…	10	
2	Group	14	9	…	14	
3	Group	6	7	…	15	
4	Group	14	3	…	16	

	penalties scored team1	penalties scored team2	goal preventions team1	\
0	0	1	6	
1	0	1	8	
2	0	0	9	
3	0	1	7	
4	1	0	4	

	goal preventions team2	own goals team1	own goals team2	\
0	5	0	0	
1	13	0	0	
2	15	0	0	
3	7	0	0	
4	14	0	0	

	forced turnovers team1	forced turnovers team2	GroupID
0	52	72	A
1	63	72	B
2	63	73	A
3	81	72	B
4	65	80	C

[5 rows x 47 columns]

[ ]:
```
matches['category'].unique()
```

[ ]:
```
array(['Group', 'Round of 16', 'Quarter-final', 'Semi-final',
       'Play-off for third place', 'Final'], dtype=object)
```

[ ]:
```
# dividing the data based on what stage of the tournament the match was in
groupMatches = matches.loc[matches['category'] == 'Group']
ro16 = matches.loc[matches['category'] == 'Round of 16']
quarterfinals = matches.loc[matches['category'] == 'Quarter-final']
semifinals = matches.loc[matches['category'] == 'Semi-final']
thirdPlaceMatch = matches.loc[matches['category'] == 'Play-off for third place']
final = matches.loc[matches['category'] == 'Final']
```

1.2.2 Countries Dataset - loading and cleaning

[ ]:
```
countries = pd.read_csv('precup_rank.csv', index_col='Nation')
countries.head()
```

[ ]:

Nation	Rank	Points	World Cup Wins
Brazil	1	1841.30	5
Belgium	2	1816.71	0
Argentina	3	1773.88	3
France	4	1759.78	2
England	5	1728.47	1

[ ]:
```
# we are creating a lot of columns, and it doesn't look pretty, but this is␣
 ↪what we want to do; is there a better way to do this?
# also, a problem that might arise is that it will become hard to distinguish␣
 ↪actual zeroes from missing values, which we might need to research a better␣
 ↪way for that as well
countries['Possession in Group'] = 0
countries['Possession in Round of 16'] = 0
countries['Possession in Quarter-final'] = 0
countries['Possession in Semi-final'] = 0
countries['Possession in Play-off for third place'] = 0
countries['Possession in Final'] = 0

countries['Total Passes in Group'] = 0
countries['Total Passes in Round of 16'] = 0
countries['Total Passes in Quarter-final'] = 0
countries['Total Passes in Semi-final'] = 0
countries['Total Passes in Play-off for third place'] = 0
countries['Total Passes in Final'] = 0

countries['Completed Passes in Group'] = 0
countries['Completed Passes in Round of 16'] = 0
countries['Completed Passes in Quarter-final'] = 0
countries['Completed Passes in Semi-final'] = 0
```

```
countries['Completed Passes in Play-off for third place'] = 0
countries['Completed Passes in Final'] = 0

countries['Shot Attempts in Group'] = 0
countries['Shot Attempts in Round of 16'] = 0
countries['Shot Attempts in Quarter-final'] = 0
countries['Shot Attempts in Semi-final'] = 0
countries['Shot Attempts in Play-off for third place'] = 0
countries['Shot Attempts in Final'] = 0

countries['Total Goals in Group'] = 0
countries['Total Goals in Round of 16'] = 0
countries['Total Goals in Quarter-final'] = 0
countries['Total Goals in Semi-final'] = 0
countries['Total Goals in Play-off for third place'] = 0
countries['Total Goals in Final'] = 0

countries['Close Range Goals in Group'] = 0
countries['Close Range Goals in Round of 16'] = 0
countries['Close Range Goals in Quarter-final'] = 0
countries['Close Range Goals in Semi-final'] = 0
countries['Close Range Goals in Play-off for third place'] = 0
countries['Close Range Goals in Final'] = 0

countries['Long Range Goals in Group'] = 0
countries['Long Range Goals in Round of 16'] = 0
countries['Long Range Goals in Quarter-final'] = 0
countries['Long Range Goals in Semi-final'] = 0
countries['Long Range Goals in Play-off for third place'] = 0
countries['Long Range Goals in Final'] = 0

countries['Conceded in Group'] = 0
countries['Conceded in Round of 16'] = 0
countries['Conceded in Quarter-final'] = 0
countries['Conceded in Semi-final'] = 0
countries['Conceded in Play-off for third place'] = 0
countries['Conceded in Final'] = 0

countries['Assists in Group'] = 0
countries['Assists in Round of 16'] = 0
countries['Assists in Quarter-final'] = 0
countries['Assists in Semi-final'] = 0
countries['Assists in Play-off for third place'] = 0
countries['Assists in Final'] = 0

countries['Fouls in Group'] = 0
countries['Fouls in Round of 16'] = 0
```

```
countries['Fouls in Quarter-final'] = 0
countries['Fouls in Semi-final'] = 0
countries['Fouls in Play-off for third place'] = 0
countries['Fouls in Final'] = 0

countries['Yellow Cards in Group'] = 0
countries['Yellow Cards in Round of 16'] = 0
countries['Yellow Cards in Quarter-final'] = 0
countries['Yellow Cards in Semi-final'] = 0
countries['Yellow Cards in Play-off for third place'] = 0
countries['Yellow Cards in Final'] = 0

countries['Red Cards in Group'] = 0
countries['Red Cards in Round of 16'] = 0
countries['Red Cards in Quarter-final'] = 0
countries['Red Cards in Semi-final'] = 0
countries['Red Cards in Play-off for third place'] = 0
countries['Red Cards in Final'] = 0

countries['Offsides in Group'] = 0
countries['Offsides in Round of 16'] = 0
countries['Offsides in Quarter-final'] = 0
countries['Offsides in Semi-final'] = 0
countries['Offsides in Play-off for third place'] = 0
countries['Offsides in Final'] = 0

countries['Saves in Group'] = 0
countries['Saves in Round of 16'] = 0
countries['Saves in Quarter-final'] = 0
countries['Saves in Semi-final'] = 0
countries['Saves in Play-off for third place'] = 0
countries['Saves in Final'] = 0

countries['Penalties in Group'] = 0
countries['Penalties in Round of 16'] = 0
countries['Penalties in Quarter-final'] = 0
countries['Penalties in Semi-final'] = 0
countries['Penalties in Play-off for third place'] = 0
countries['Penalties in Final'] = 0

countries['Free Kicks in Group'] = 0
countries['Free Kicks in Round of 16'] = 0
countries['Free Kicks in Quarter-final'] = 0
countries['Free Kicks in Semi-final'] = 0
countries['Free Kicks in Play-off for third place'] = 0
countries['Free Kicks in Final'] = 0
```

```
countries['Corners in Group'] = 0
countries['Corners in Round of 16'] = 0
countries['Corners in Quarter-final'] = 0
countries['Corners in Semi-final'] = 0
countries['Corners in Play-off for third place'] = 0
countries['Corners in Final'] = 0

countries['Own Goals in Group'] = 0
countries['Own Goals in Round of 16'] = 0
countries['Own Goals in Quarter-final'] = 0
countries['Own Goals in Semi-final'] = 0
countries['Own Goals in Play-off for third place'] = 0
countries['Own Goals in Final'] = 0

countries['Forced Turnovers in Group'] = 0
countries['Forced Turnovers in Round of 16'] = 0
countries['Forced Turnovers in Quarter-final'] = 0
countries['Forced Turnovers in Semi-final'] = 0
countries['Forced Turnovers in Play-off for third place'] = 0
countries['Forced Turnovers in Final'] = 0

countries['GroupID'] = ''
```

Now that we have our `countries` dataset prepared for data to enter it, we need to start to modify our `matches` dataset, so that any redundant information is discarded, and all the information we need is correctly represented.

We are now going to write some code which will allow us to clean up the way some of this data looks. Ideally, we want to observe these statistics based on country, while we have it here as `team1` or `team2`, which isn't really helpful if we want to get a context of a particular country. So, using the `countries` dataset that we have introduced earlier that just contains their FIFA rank at the time of the World Cup, we will be adding each countries individual statistics to the dataset.

[ ]:
```
def parseRound(round):
    teamID = ''
    for team in countries.index:
        roundMatches = round.loc[((round['team1'] == team) | (round['team2'] ==␣
 ↪team))]
        for _, match in roundMatches.iterrows():

            if match['team1'] == team:
                teamID = '1'
            else:
                teamID = '2'

            if match['category'] == 'Group':
                countries.loc[team, f'GroupID'] = match[f'GroupID']
```

```
            #print(team, match[f'number of goals team{teamID}'])
            countries.loc[team, f'Possession in {match["category"]}'] +=␣
 ↪match[f'possession team{teamID}']
            countries.loc[team, f'Total Passes in {match["category"]}'] +=␣
 ↪match[f'passes team{teamID}']
            countries.loc[team, f'Completed Passes in {match["category"]}'] +=␣
 ↪match[f'passes completed team{teamID}']
            countries.loc[team, f'Total Goals in {match["category"]}'] +=␣
 ↪match[f'number of goals team{teamID}']
            countries.loc[team, f'Close Range Goals in {match["category"]}'] +=␣
 ↪match[f'goal inside the penalty area team{teamID}']
            countries.loc[team, f'Long Range Goals in {match["category"]}'] +=␣
 ↪match[f'goal outside the penalty area team{teamID}']
            countries.loc[team, f'Conceded in {match["category"]}'] +=␣
 ↪match[f'conceded team{teamID}']
            countries.loc[team, f'Assists in {match["category"]}'] +=␣
 ↪match[f'assists team{teamID}']
            countries.loc[team, f'Own Goals in {match["category"]}'] +=␣
 ↪match[f'own goals team{teamID}']
            countries.loc[team, f'Forced Turnovers in {match["category"]}'] +=␣
 ↪match[f'forced turnovers team{teamID}']
            countries.loc[team, f'Saves in {match["category"]}'] +=␣
 ↪match[f'goal preventions team{teamID}']
            countries.loc[team, f'Penalties in {match["category"]}'] +=␣
 ↪match[f'penalties scored team{teamID}']
            countries.loc[team, f'Free Kicks in {match["category"]}'] +=␣
 ↪match[f'free kicks team{teamID}']
            countries.loc[team, f'Corners in {match["category"]}'] +=␣
 ↪match[f'corners team{teamID}']
            countries.loc[team, f'Fouls in {match["category"]}'] +=␣
 ↪match[f'fouls against team{teamID}']
            countries.loc[team, f'Offsides in {match["category"]}'] +=␣
 ↪match[f'offsides team{teamID}']
            countries.loc[team, f'Yellow Cards in {match["category"]}'] +=␣
 ↪match[f'yellow cards team{teamID}']
            countries.loc[team, f'Red Cards in {match["category"]}'] +=␣
 ↪match[f'red cards team{teamID}']
            countries.loc[team, f'Shot Attempts in {match["category"]}'] +=␣
 ↪match[f'total attempts team{teamID}']


parseRound(groupMatches)
parseRound(ro16)
parseRound(quarterfinals)
parseRound(semifinals)
parseRound(thirdPlaceMatch)
```

```
parseRound(final)
```

[ ]:
```
countries['Average Possession in Group'] = round(countries['Possession in␣
 ↪Group'] / 3, 2)
countries['Goals Per Game in Group'] = round(countries['Total Goals in Group'] /
 ↪ 3, 2)

countries['Total Goals'] = countries['Total Goals in Group'] + countries['Total␣
 ↪Goals in Round of 16'] + \
    countries['Total Goals in Quarter-final'] + countries['Total Goals in␣
 ↪Semi-final'] + countries['Total Goals in Play-off for third place'] \
        + countries['Total Goals in Final']

countries.head()
```

[ ]:

Nation	Rank	Points	World Cup Wins	Possession in Group	\
Brazil	1	1841.30	5	160	
Belgium	2	1816.71	0	149	
Argentina	3	1773.88	3	181	
France	4	1759.78	2	156	
England	5	1728.47	1	181	

Nation	Possession in Round of 16	Possession in Quarter-final	\
Brazil	47	45	
Belgium	0	0	
Argentina	53	44	
France	48	36	
England	54	54	

Nation	Possession in Semi-final	Possession in Play-off for third place	\
Brazil	0	0	
Belgium	0	0	
Argentina	34	0	
France	34	0	
England	0	0	

Nation	Possession in Final	Total Passes in Group	…	\
Brazil	0	1698	…	
Belgium	0	1779	…	
Argentina	46	2005	…	
France	40	1873	…	
England	0	1947	…	

```
         Forced Turnovers in Group  Forced Turnovers in Round of 16  \
Nation
Brazil                         211                              73
Belgium                        180                               0
Argentina                      176                              67
France                         223                              71
England                        171                              60

         Forced Turnovers in Quarter-final  Forced Turnovers in Semi-final  \
Nation
Brazil                                  77                               0
Belgium                                  0                               0
Argentina                               79                              85
France                                  54                              72
England                                 49                               0

         Forced Turnovers in Play-off for third place  \
Nation
Brazil                                              0
Belgium                                             0
Argentina                                           0
France                                              0
England                                             0

         Forced Turnovers in Final  GroupID  Average Possession in Group  \
Nation
Brazil                           0       G                         53.33
Belgium                          0       F                         49.67
Argentina                       87       C                         60.33
France                         104       D                         52.00
England                          0       B                         60.33

         Goals Per Game in Group  Total Goals
Nation
Brazil                      1.00            8
Belgium                     0.33            1
Argentina                   1.67           15
France                      2.00           16
England                     3.00           13

[5 rows x 121 columns]
```

[ ]:
```
groupA = countries.loc[countries['GroupID'] == 'A']
groupB = countries.loc[countries['GroupID'] == 'B']
groupC = countries.loc[countries['GroupID'] == 'C']
groupD = countries.loc[countries['GroupID'] == 'D']
groupE = countries.loc[countries['GroupID'] == 'E']
```

```
groupF = countries.loc[countries['GroupID'] == 'F']
groupG = countries.loc[countries['GroupID'] == 'G']
groupH = countries.loc[countries['GroupID'] == 'H']

print(f'Group A average team ranking: {groupA["Rank"].mean()}')
print(f'Group B average team ranking: {groupB["Rank"].mean()}')
print(f'Group C average team ranking: {groupC["Rank"].mean()}')
print(f'Group D average team ranking: {groupD["Rank"].mean()}')
print(f'Group E average team ranking: {groupE["Rank"].mean()}')
print(f'Group F average team ranking: {groupF["Rank"].mean()}')
print(f'Group G average team ranking: {groupG["Rank"].mean()}')
print(f'Group H average team ranking: {groupH["Rank"].mean()}')
```

```
Group A average team ranking: 30.0
Group B average team ranking: 15.0
Group C average team ranking: 23.25
Group D average team ranking: 20.5
Group E average team ranking: 18.25
Group F average team ranking: 19.25
Group G average team ranking: 20.0
Group H average team ranking: 28.0
```

It seems that based on the mean ranks of our groups, Group B has the highest average rank, which coins this group 'The Group of Death', a term designated to the toughest group in the tournament. On the other hand, Group A seems to have the lowest average rank. Let's take a look at the two groups, and see what that's about.

[ ]:
```
groupB[['Rank']]
```

[ ]:

Nation	Rank
England	5
United States	16
Wales	19
Iran	20

Looking at the group of death, we can see that all 4 teams are ranked in the top 20, with 3 of the teams being very close to each other in rank.

[ ]:
```
groupA[['Rank']]
```

[ ]:

Nation	Rank
Netherlands	8
Senegal	18
Ecuador	44
Qatar	50

This group is much different than Group B; while we can see that we have 2 top 20 nations here,

but two of the lower ranked countries in the tournaments as well, in Ecuador and Qatar (the host country)

1.2.3 Players Dataset - loading and cleaning

[ ]:
```
players = pd.read_csv('player_stats.csv')

duplicate_columns = ['player', 'club', 'position', 'age', 'team', 'birth_year',␣
 ↪'minutes_90s']
players = players[duplicate_columns]

keeper_columns =␣
 ↪['position','team','age','club','birth_year','games','games_starts','minutes','minutes_90s']

player_defense = pd.read_csv('player_defense.csv').drop(duplicate_columns[2:],␣
 ↪axis=1)
player_shooting = pd.read_csv('player_shooting.csv').drop(duplicate_columns[2:
 ↪], axis=1)
player_possession = pd.read_csv('player_possession.csv').
 ↪drop(duplicate_columns[2:], axis=1)
player_keepers = pd.read_csv('player_keepers.csv').drop(keeper_columns, axis=1)
player_passing = pd.read_csv('player_passing.csv').drop(duplicate_columns[2:],␣
 ↪axis=1)

player_keepers = pd.merge(players.loc[players['position'] == 'GK'],␣
 ↪player_keepers, on='player', how='left')


# merging all of the above dataframes into one
players = players.merge(player_defense, on='player')
players = players.merge(player_shooting, on='player')
players = players.merge(player_possession, on='player')
players = players.merge(player_passing, on='player')

#players = players.merge(player_keepers, on='player') -> for some reason␣
 ↪everything fucks up with keepers added


players['age'] = players['age'].astype(str).str[:2].astype(int)
players['G/A'] = players['goals'] + players['assists']

players
```

[ ]:
```
                  player            club position  age        team  \
0            Aaron Mooy          Celtic       MF   32   Australia
1          Aaron Ramsey            Nice       MF   31       Wales
2     Abdelhamid Sabiri        Sampdoria       MF   26     Morocco
```

```
3       Abdelkarim Hassan       Al Sadd SC       DF   29       Qatar
4    Abderrazak Hamdallah       Al-Ittihad       FW   32     Morocco
..                    …              …      …  …         …
675       Ángel Di María         Juventus       MF   34   Argentina
676      Ángelo Preciado             Genk       DF   24     Ecuador
677         Éder Militão      Real Madrid       DF   24      Brazil
678         Óscar Duarte         Al-Wehda       DF   33  Costa Rica
679       İlkay Gündoğan  Manchester City       MF   32     Germany

     birth_year  minutes_90s  tackles  tackles_won  tackles_def_3rd  …  \
0          1990          4.0      9.0            6              4.0  …
1          1990          3.0      2.0            0              0.0  …
2          1996          2.0      3.0            1              1.0  …
3          1993          3.0      7.0            3              5.0  …
4          1990          0.8      0.0            0              0.0  …
..          …           …       …          …               …  …
675        1988          3.2      3.0            1              2.0  …
676        1998          2.9      7.0            5              3.0  …
677        1998          3.9      7.0            6              4.0  …
678        1989          3.0      4.0            2              4.0  …
679        1990          2.1      3.0            1              1.0  …

     assists  xg_assist  pass_xa  xg_assist_net  assisted_shots  \
0          0        0.1      0.1           -0.1             1.0
1          0        0.0      0.1            0.0             1.0
2          1        0.9      0.1            0.1             3.0
3          0        0.0      0.1            0.0             1.0
4          0        0.0      0.0            0.0             0.0
..       …        …      …            …             …
675        1        0.6      0.7            0.4            10.0
676        1        0.4      0.2            0.6             4.0
677        0        0.0      0.1            0.0             1.0
678        0        0.3      0.0           -0.3             2.0
679        0        0.1      0.3           -0.1             1.0

     passes_into_final_third  passes_into_penalty_area  \
0                       22.0                       1.0
1                        7.0                       1.0
2                        3.0                       0.0
3                       13.0                       1.0
4                        1.0                       0.0
..                       …                       …
675                      3.0                      11.0
676                      4.0                       6.0
677                     20.0                       1.0
678                      1.0                       0.0
679                     18.0                       4.0
```

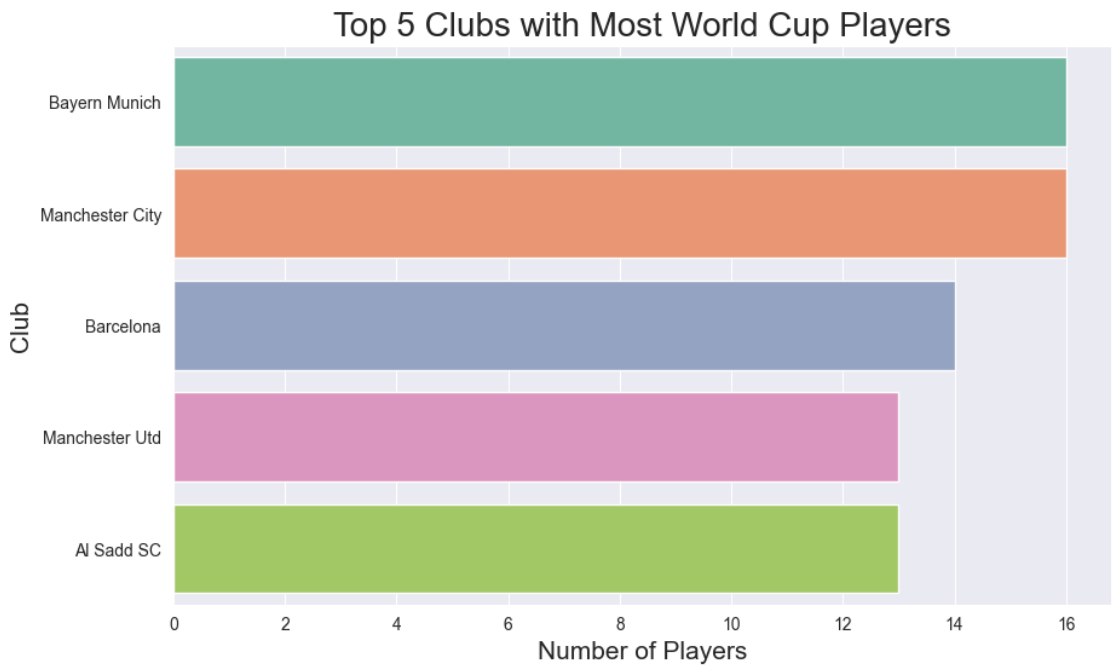
```
     crosses_into_penalty_area  progressive_passes  G/A
0                          0.0                14.0    0
1                          0.0                 5.0    0
2                          0.0                 0.0    1
3                          0.0                 8.0    0
4                          0.0                 0.0    0
..                         …                  …  …
675                        3.0                17.0    2
676                        5.0                 6.0    1
677                        1.0                12.0    0
678                        0.0                 0.0    0
679                        0.0                15.0    1

[680 rows x 78 columns]
```

1.3 Visualizing the players, and their performances

Some things we want to see with our players:

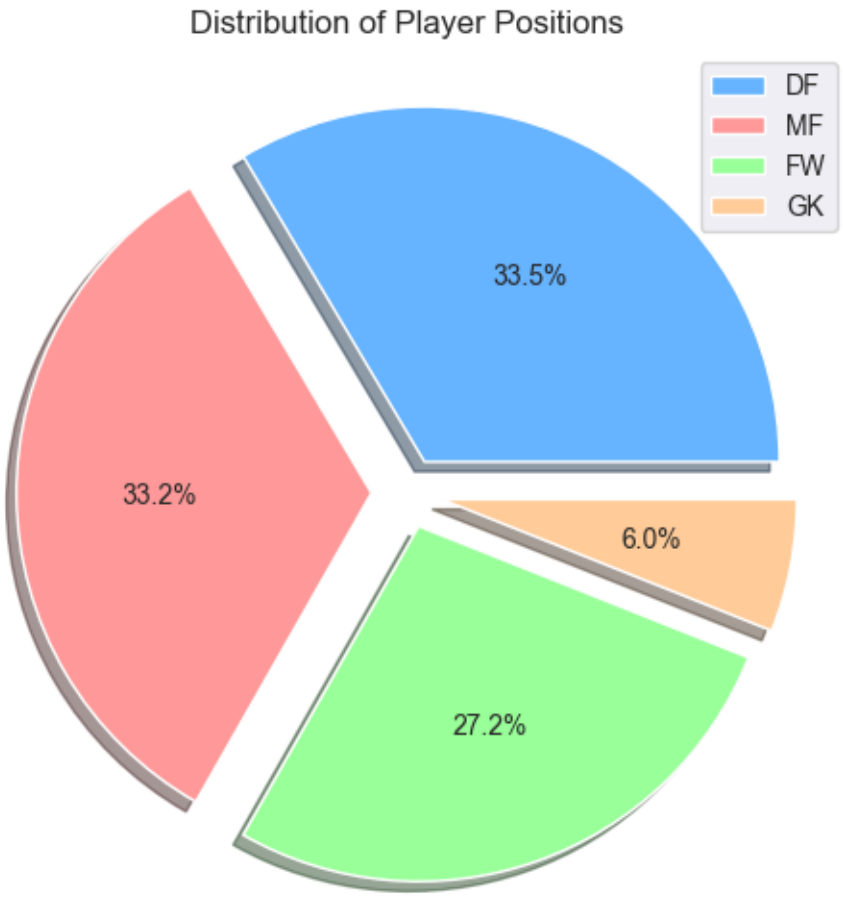
- Most represented clubs
- Distribution of positions
- Average age of players
- Top goal scorers
- Top assisters
- Top players under 21 goals + assists (g/a)

[ ]:
```
plt.figure(figsize=(10, 6))
sns.set_style('darkgrid')
sns.countplot(y='club', data=players, order=players['club'].value_counts().
 ↪iloc[:5].index, palette='Set2')
plt.title('Top 5 Clubs with Most World Cup Players', fontsize=20)
plt.xlabel('Number of Players', fontsize=15)
plt.ylabel('Club', fontsize=15)
plt.show()
```

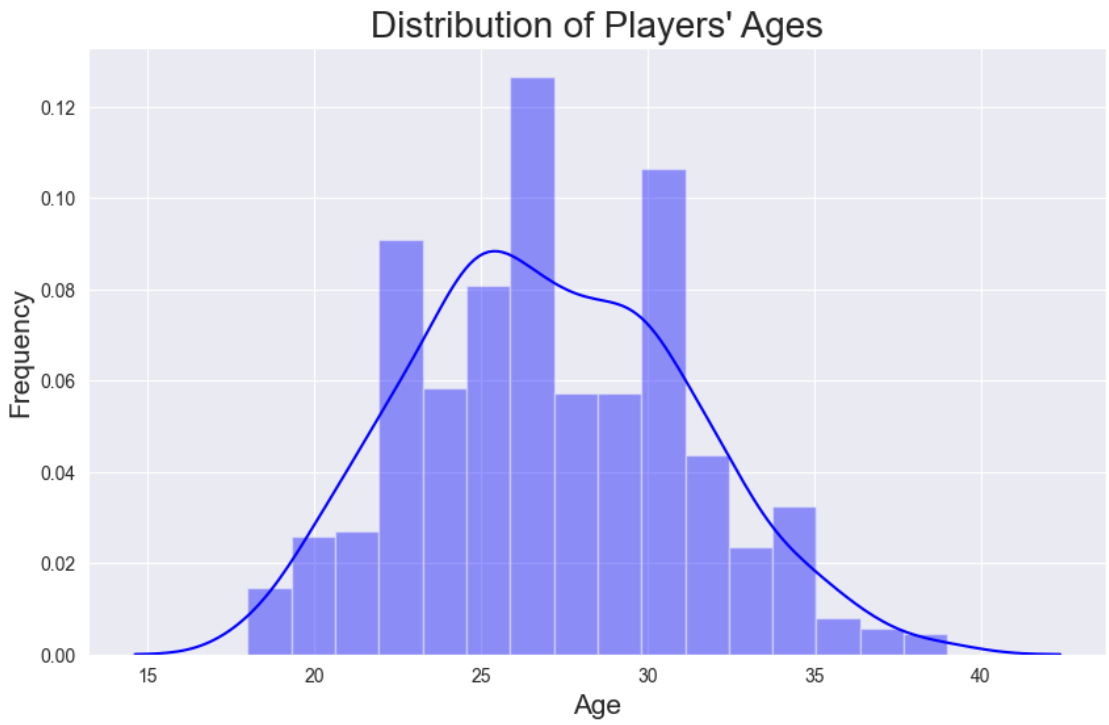


Al Sadd SC is an interesting outlier because all of these other clubs are very well known and highly regarded around the world, but Al Sadd SC does not fit the mold of the rest of these historic clubs. The reason behind the strong showing from Al Saad SC is because it is a club in Qatar (who is the host nation of this World Cup), and the majority of the players from the Qatar national team actually play for that club.

[ ]:
```
colors = ['#66b3ff','#ff9999','#99ff99','#ffcc99']

ax = plt.figure(figsize=(10, 6))

ax = players['position'].value_counts().plot(kind='pie',
                                             autopct='%1.1f%%', shadow=True,␣
 ↪explode=[0.1, 0.1, 0.1, 0.1], colors=colors,
                                             legend=True, title='Distribution of␣
 ↪Player Positions', ylabel='', labeldistance=None)
```



[ ]:
```
plt.figure(figsize=(10, 6))
sns.set_style('darkgrid')
sns.distplot(players['age'], color='blue')
plt.title('Distribution of Players\' Ages', fontsize=20)
plt.xlabel('Age', fontsize=15)
plt.ylabel('Frequency', fontsize=15)
plt.show()
```



[ ]:
```
players['age'].mean()
```

[ ]:
```
27.054411764705883
```

Based on the visual, and with our `.mean()` as further proof, we can see that there are a lot of players in the 27 yera old range, with other spikes at around the 30-32 range, and at the 21-23 range as well. This makes us interested, though, in who the oldest and youngest players were in the tournament.

[ ]:
```
players.sort_values('age', ascending=False).head()[['player', 'age', 'team',␣
↪'position', 'goals']].sort_values('goals', ascending=False)
```

[ ]:

	player	age	team	position	goals
519	Pepe	39	Portugal	DF	1
142	Dani Alves	39	Brazil	DF	0
75	Atiba Hutchinson	39	Canada	MF	0
616	Thiago Silva	38	Brazil	DF	0
108	Bryan Ruiz	37	Costa Rica	MF	0

[ ]:
```
players.sort_values('age').head()[['player', 'age', 'team', 'position',␣
↪'goals']].sort_values('goals', ascending=False)
```

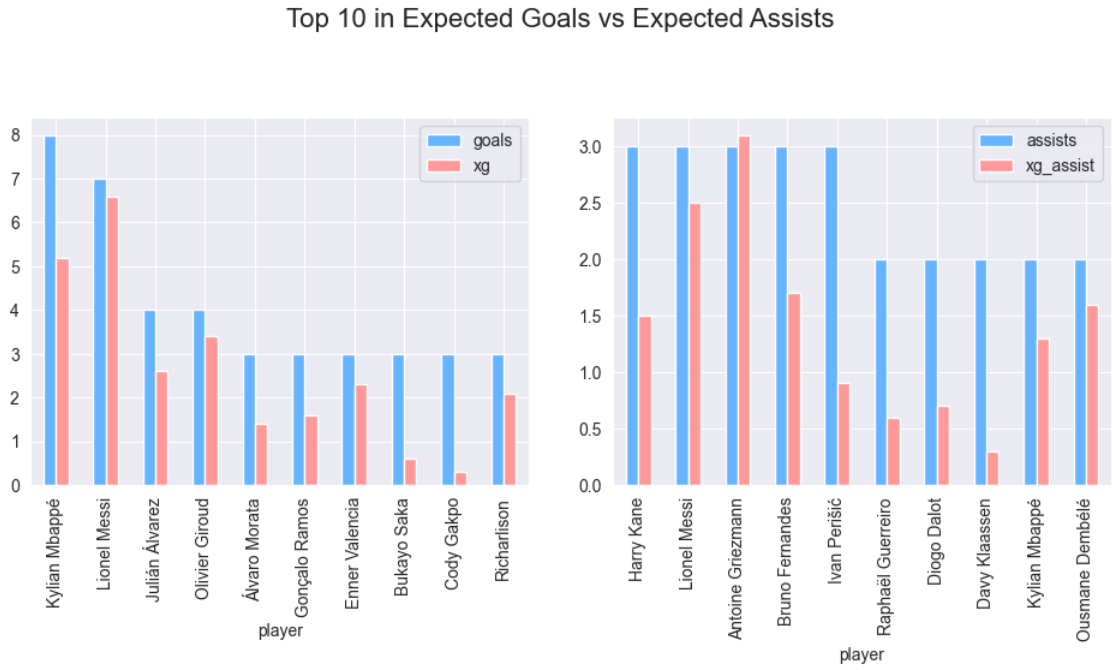
[ ]:

	player	age	team	position	goals
217	Gavi	18	Spain	MF	1

```
215            Garang Kuol   18  Australia       FW      0
666      Youssoufa Moukoko   18    Germany       FW      0
94       Bilal El Khannous   18    Morocco       MF      0
7    Abdul Fatawu Issahaku   18      Ghana       FW      0
```

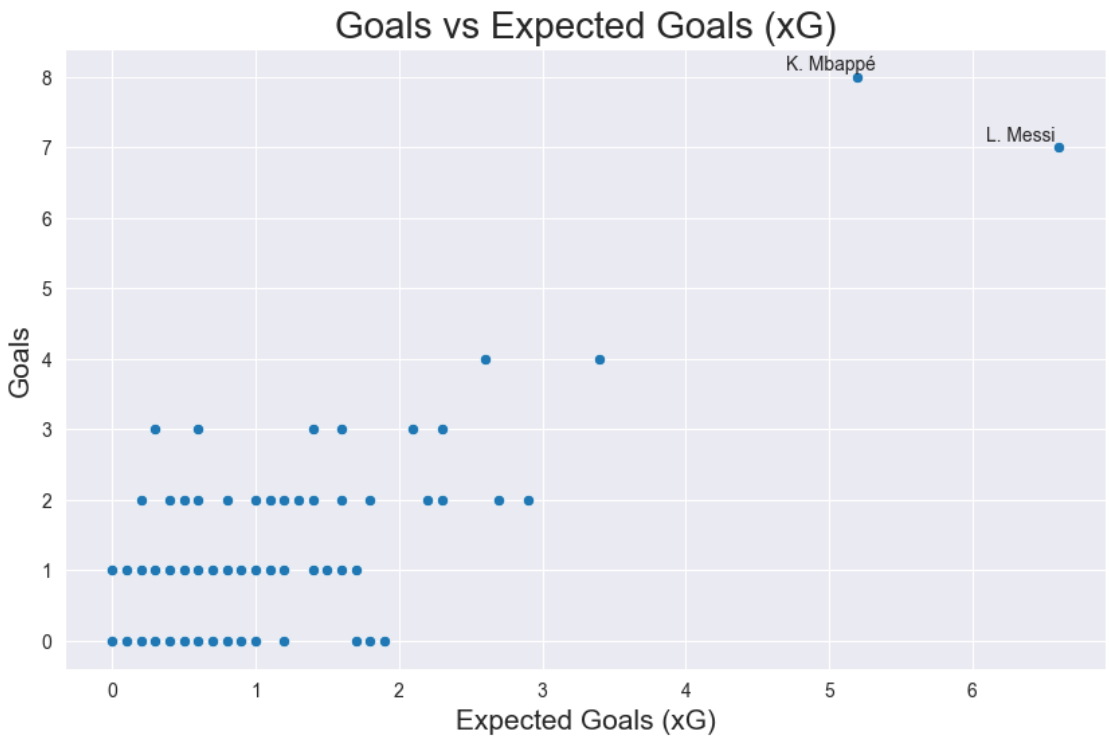
It is interesting that there are no players in the tournament over the age of 40, and that there were no players under the age of 18.

[ ]:
```
# fix this; kind of ugly
_top_xg = players.sort_values('goals', ascending=False).head(10)
_top_xa = players.sort_values('assists', ascending=False).head(10)
top_xg = _top_xg[['player', 'goals', 'xg']]
top_xassist = _top_xa[['player', 'assists', 'xg_assist']]
top_xg = top_xg.set_index('player')
top_xassist = top_xassist.set_index('player')


fig, axes = plt.subplots(1, 2, figsize=(15, 5))
sns.set_style('darkgrid')

top_xg.plot.bar(figsize=(10, 6), color=['#66b3ff','#ff9999'], ax=axes[0])
top_xassist.plot.bar(figsize=(10, 6), color=['#66b3ff','#ff9999'], ax=axes[1])

fig.suptitle('Top 10 in Expected Goals vs Expected Assists', fontsize=16)
fig.tight_layout(pad=3.0)
```



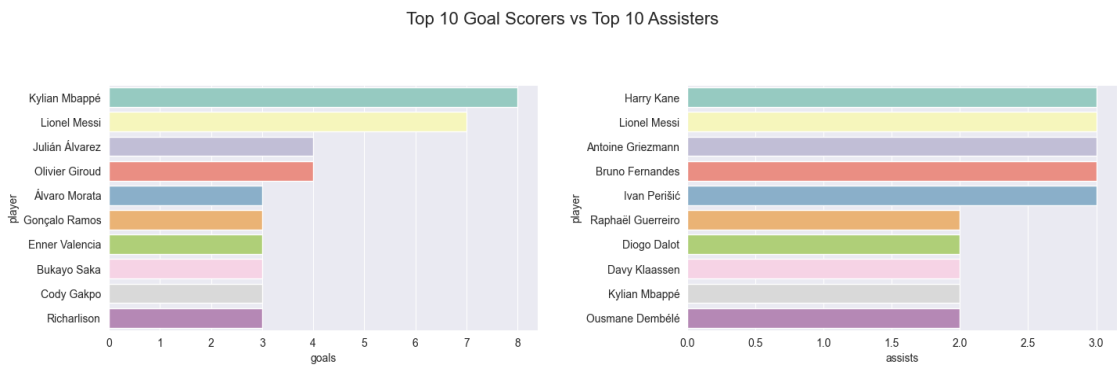
In football, xG (expected goals) is an interesting advanced statistic that is used to measure the quality of a shot. It is a metric that is used to measure the likelihood of a shot resulting in a goal, based on the quality of the shot (under certain determined metrics). This is interesting because we can observe that someone like Lionel Messi, scored around the same amount of goals as his xG, meaning that he was statistically scoring all the goals that he was expected to. Players like Bukayo Saka and Cody Gakpo both had relatively low xG (under 1 xG), but finished with 3 goals each, so they massively outperformed their xG.

[ ]:
```
plt.figure(figsize=(10, 6))
sns.set_style('darkgrid')

sns.scatterplot(x='xg', y='goals', data=players, palette='Set3')
plt.title('Goals vs Expected Goals (xG)', fontsize=20)
plt.xlabel('Expected Goals (xG)', fontsize=15)
plt.ylabel('Goals', fontsize=15)

# Find the outliers
outliers = players[players['goals'] > players['goals'].quantile(.97) + 1.5 *␣
 ↪(players['goals'].quantile(.97) - players['goals'].quantile(.03))]


# Add labels to the outliers
for i, row in outliers.iterrows():
    playerName = row['player']
    try:
        formattedPlayerName = f'{playerName[0]}. {playerName.split(" ")[1]}'
    except:
        # they only go by one name: ex. Richarlison or Gavi
        formattedPlayerName = playerName
    plt.annotate(formattedPlayerName, xy=(row['xg'], row['goals']),␣
 ↪xytext=(row['xg']-.5, row['goals']+.1))
```



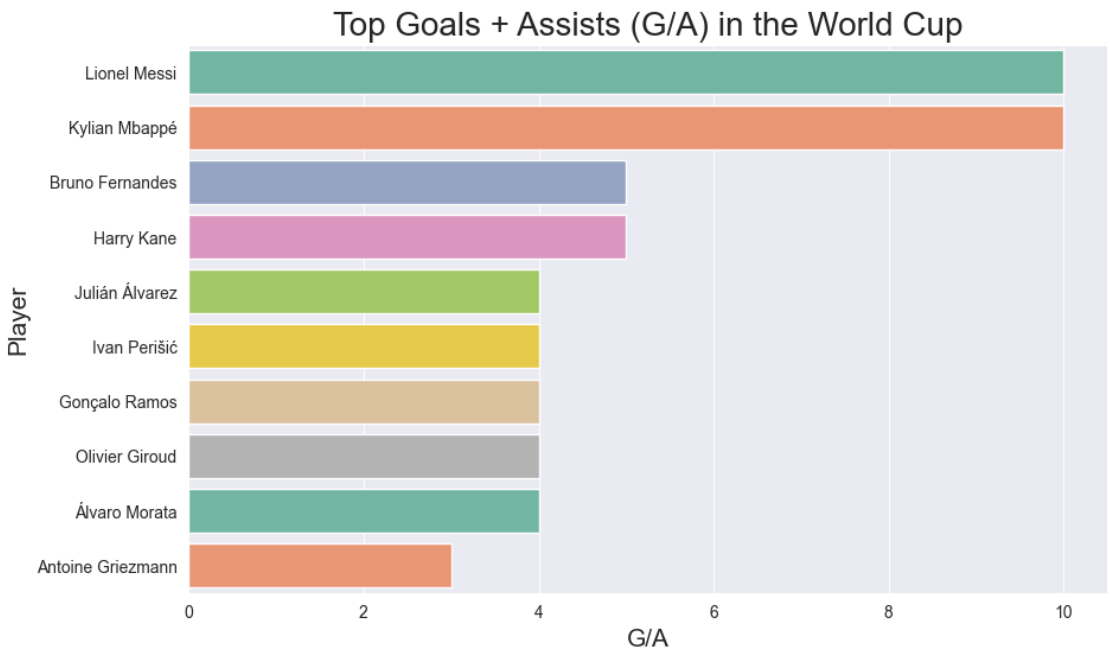
And if it wasn't already obvious by the previous graph, Lionel Messi and Kylian Mbappe were the most efficient scorers in the tournament with both the highest goals and xG.

[ ]:
```
fig, axes = plt.subplots(1, 2, figsize=(15, 5))
sns.set_style('darkgrid')

sns.barplot(x='goals', y='player', data=players.sort_values('goals',␣
 ↪ascending=False).head(10), palette='Set3', ax=axes[0])

sns.barplot(x='assists', y='player', data=players.sort_values('assists',␣
 ↪ascending=False).head(10), palette='Set3', ax=axes[1])

fig.suptitle('Top 10 Goal Scorers vs Top 10 Assisters', fontsize=16)
fig.tight_layout(pad=3.0)
```



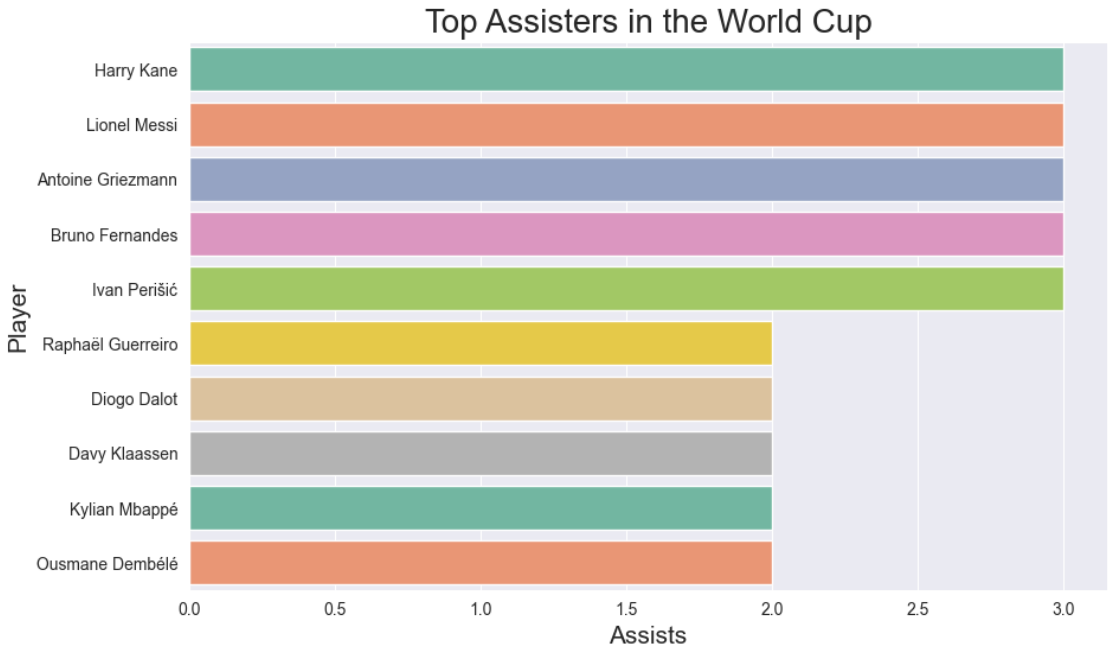
[ ]:
```
plt.figure(figsize=(10, 6))
sns.set_style('darkgrid')

sns.barplot(x='G/A', y='player', data=players.sort_values('G/A',␣
 ↪ascending=False).head(10), palette='Set2')
plt.title('Top Goals + Assists (G/A) in the World Cup', fontsize=20)
plt.xlabel('G/A', fontsize=15)
plt.ylabel('Player', fontsize=15)
```

[ ]:
```
Text(0, 0.5, 'Player')
```



[ ]:
```
plt.figure(figsize=(10, 6))
sns.set_style('darkgrid')

sns.barplot(x='assists', y='player', data=players.sort_values('assists',␣
 ↪ascending=False).head(10), palette='Set2')
plt.title('Top Assisters in the World Cup', fontsize=20)
plt.xlabel('Assists', fontsize=15)
plt.ylabel('Player', fontsize=15)
```

[ ]:
```
Text(0, 0.5, 'Player')
```



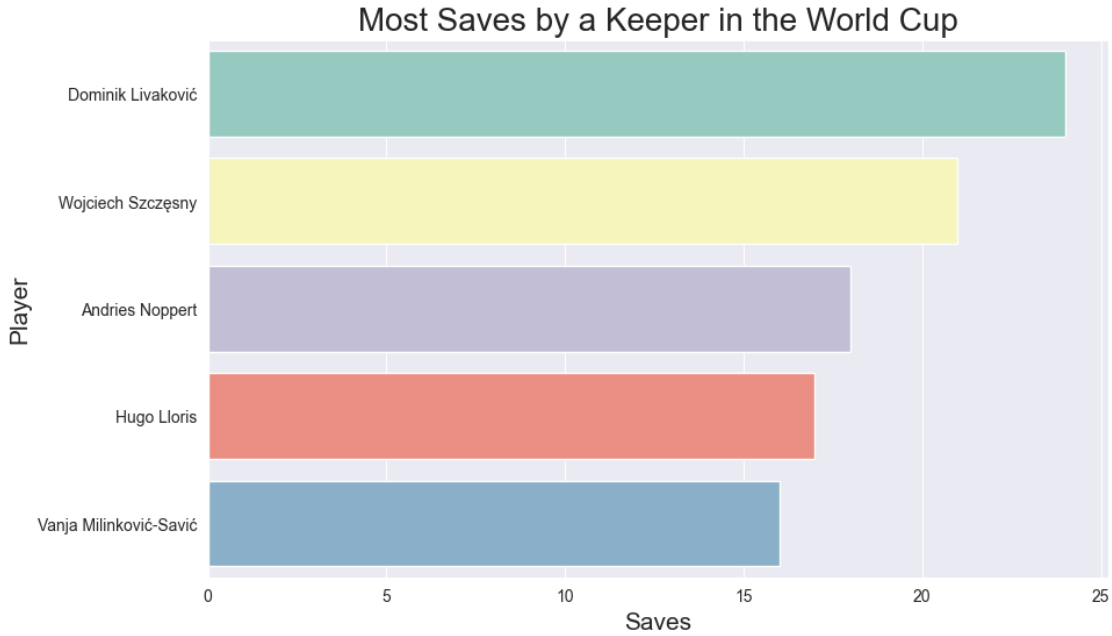
[ ]:
```
player_keepers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 41 entries, 0 to 40
Data columns (total 22 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   player                   41 non-null     object
 1   club                     41 non-null     object
 2   position                 41 non-null     object
 3   age                      41 non-null     object
 4   team                     41 non-null     object
 5   birth_year               41 non-null     int64
 6   minutes_90s              41 non-null     float64
 7   goals_against            41 non-null     int64
```

```
 8   goals_against_per90      41 non-null     float64
 9   shots_on_target_against  41 non-null     int64
 10  saves                    41 non-null     int64
 11  save_pct                 40 non-null     float64
 12  wins                     41 non-null     int64
 13  ties                     41 non-null     int64
 14  losses                   41 non-null     int64
 15  clean_sheets             41 non-null     int64
 16  clean_sheets_pct         40 non-null     float64
 17  pens_att                 41 non-null     int64
 18  pens_allowed             41 non-null     int64
 19  pens_saved               41 non-null     int64
 20  pens_missed              41 non-null     int64
 21  pens_save_pct            17 non-null     float64
dtypes: float64(5), int64(12), object(5)
memory usage: 7.4+ KB
```

[ ]:
```
plt.figure(figsize=(10, 6))
sns.set_style('darkgrid')

sns.barplot(x='saves', y='player', data=player_keepers.sort_values('saves',␣
 ↪ascending=False).head(5), palette='Set3')
plt.title('Most Saves by a Keeper in the World Cup', fontsize=20)
plt.xlabel('Saves', fontsize=15)
plt.ylabel('Player', fontsize=15)
```

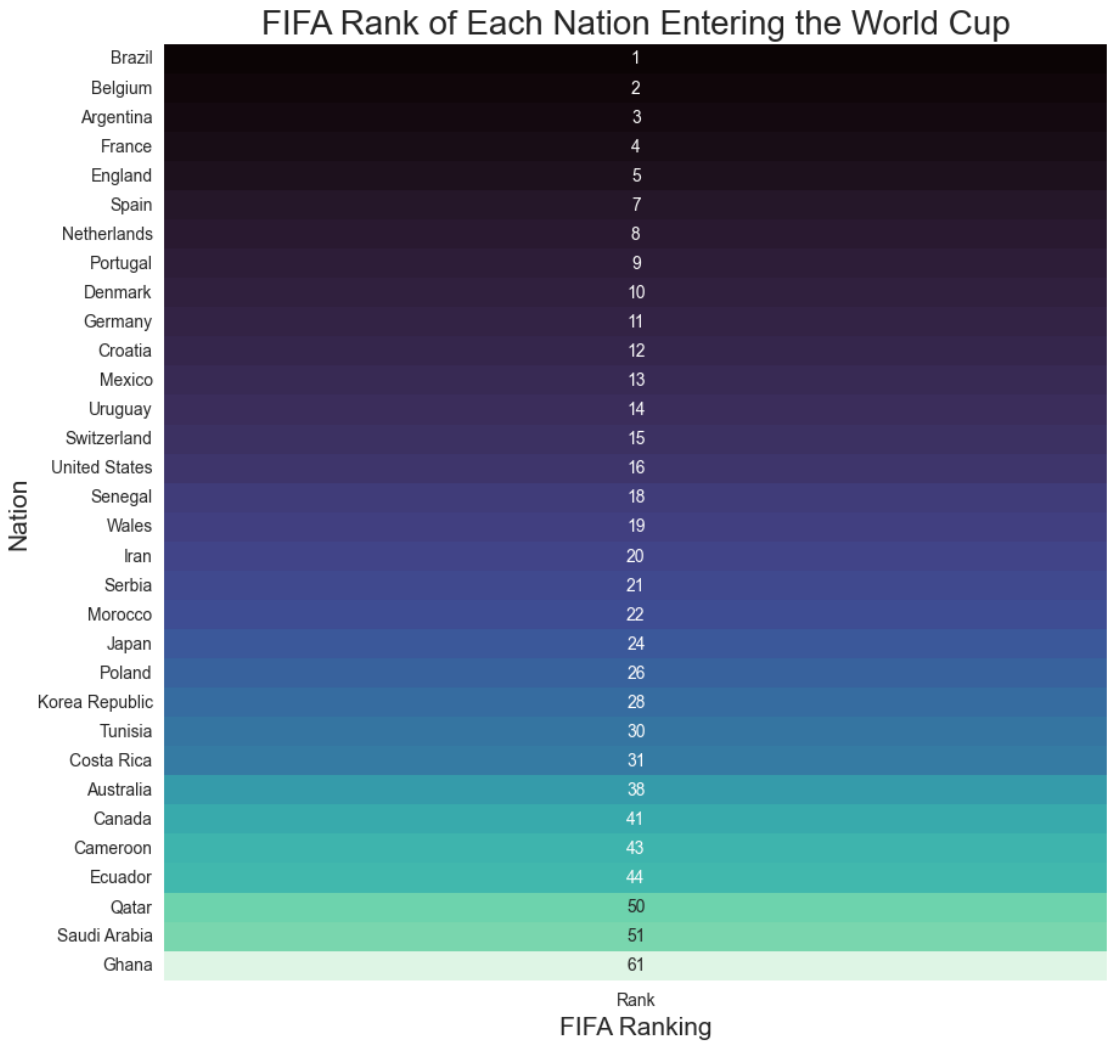
[ ]:
```
Text(0, 0.5, 'Player')
```



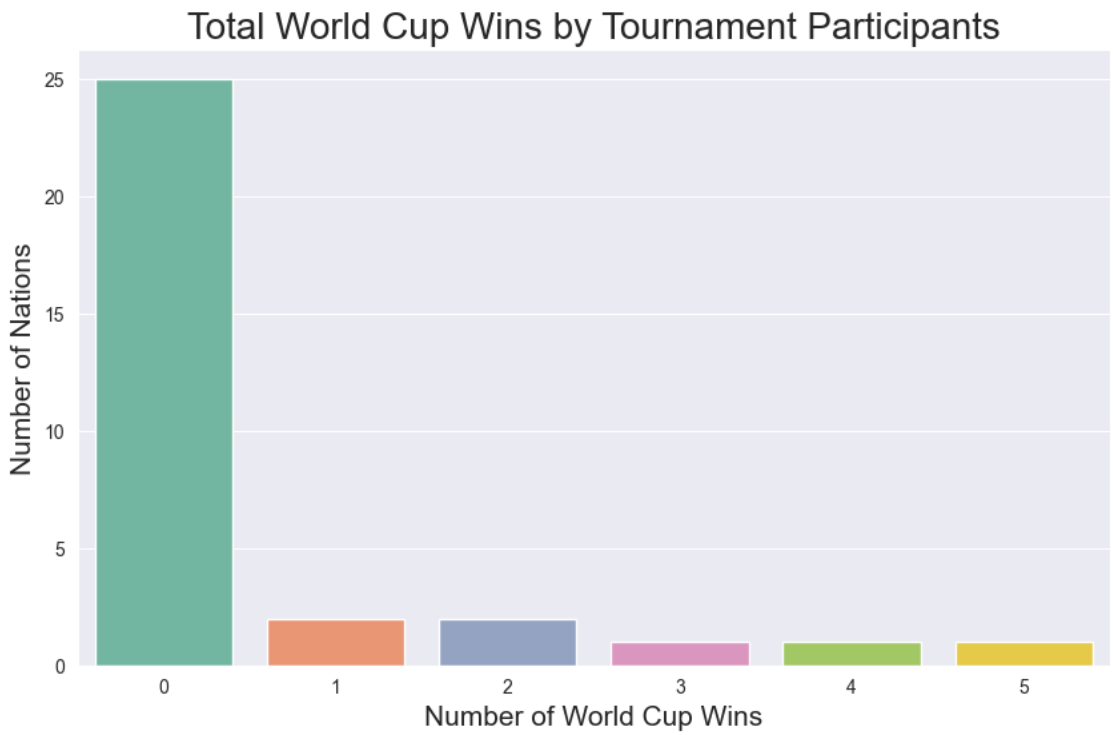
Another race to follow during the World Cup is that of the Best Young Player award, which is given to the best player in the tournament under the age of 21. Many factors contribute to the award, including how far the player's team goes in the tournament, and how well they perform. We are going to look at some of the stats of the top 5 players under 21 in the tournament, and see how they compare to each other, and ultimately who should have won the award based on statistics.

1.4 Background analysis: visualizing the Countries

[ ]:
```
plt.figure(figsize=(10, 10))
sns.set_style('darkgrid')

sns.heatmap(countries['Rank'].to_frame(), annot=True, fmt='g', cmap='mako',␣
 ↪cbar=False)
plt.title('FIFA Rank of Each Nation Entering the World Cup', fontsize=20)
plt.xlabel('FIFA Ranking', fontsize=15)
plt.ylabel('Nation', fontsize=15)
plt.show()
```



[ ]:
```python
# plot the value counts of the column 'world cup wins'
plt.figure(figsize=(10, 6))
sns.set_style('darkgrid')

sns.countplot(x='World Cup Wins', data=countries, palette='Set2')
plt.title('Total World Cup Wins by Tournament Participants', fontsize=20)
plt.xlabel('Number of World Cup Wins', fontsize=15)
plt.ylabel('Number of Nations', fontsize=15)
plt.show()
```



It is easy to see that most of the countries have never won a World Cup, and a few countries have won one, or two. There are three countries however that have won 3 or more, so let's see who they are.

[ ]:
```
countries[countries['World Cup Wins'] >= 3]['World Cup Wins'].to_frame().
↪sort_values('World Cup Wins', ascending=False)
```

[ ]:
```
           World Cup Wins
Nation
Brazil                  5
Germany                 4
Argentina               3
```

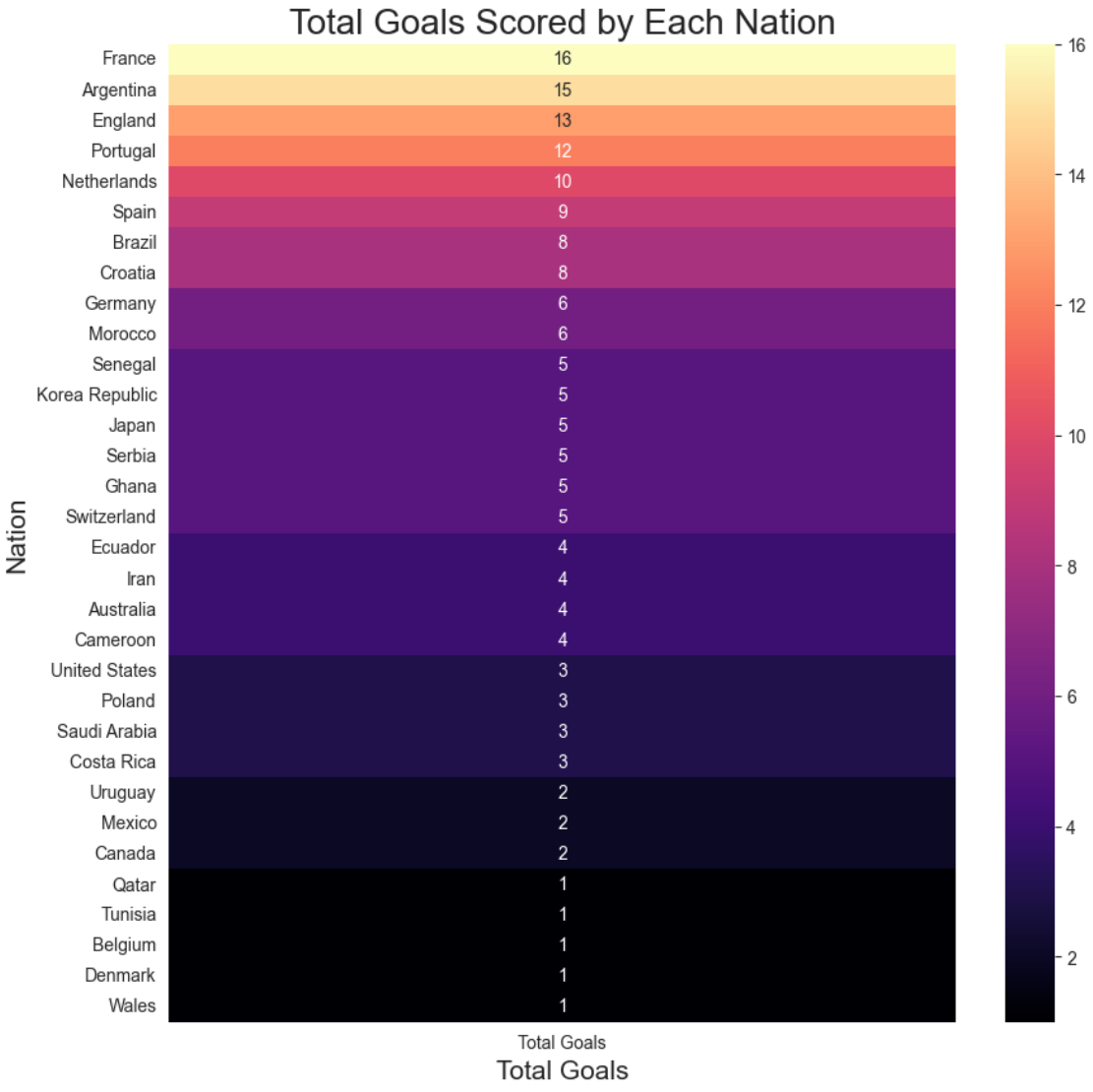
[ ]:
```
TOTAL_WORLD_CUPS = 22
TOTAL_WORLD_CUPS - countries['World Cup Wins'].sum()
```

[ ]:
```
4
```

This is notable because the total number of World Cups played is 22 (counting this one), and we have 18 of the winners represented here. That would mean that 4 World Cup winners did not qualify for this World Cup, which is a very interesting statistic. The other 4 missing World Cups were won by Italy, who failed to qualify to this World Cup due to a surprise 1-0 upset loss in a World Cup qualifying match to North Macedonia.

1.5 Visualizing the Group Stage

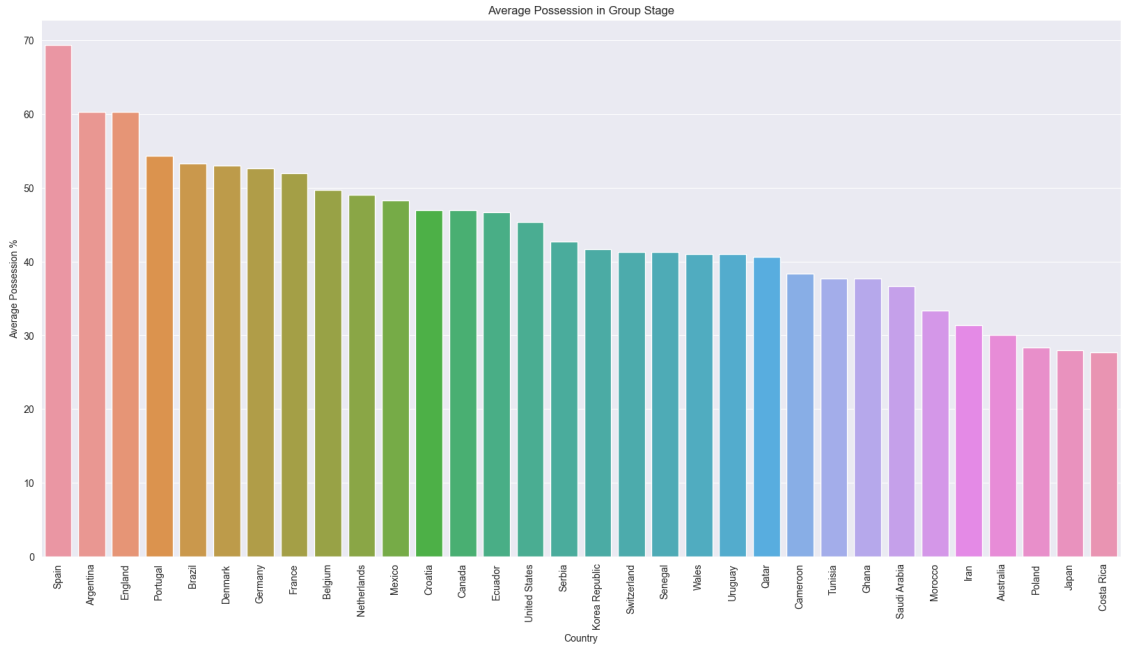
```
[ ]: # create a heatmap using the Total Goals column
plt.figure(figsize=(10, 10))
sns.set_style('darkgrid')

sns.heatmap(countries[['Total Goals']].sort_values(by='Total Goals',␣
 ↪ascending=False), annot=True, cmap='magma')
plt.title('Total Goals Scored by Each Nation', fontsize=20)
plt.xlabel('Total Goals', fontsize=15)
plt.ylabel('Nation', fontsize=15)
plt.show()
```



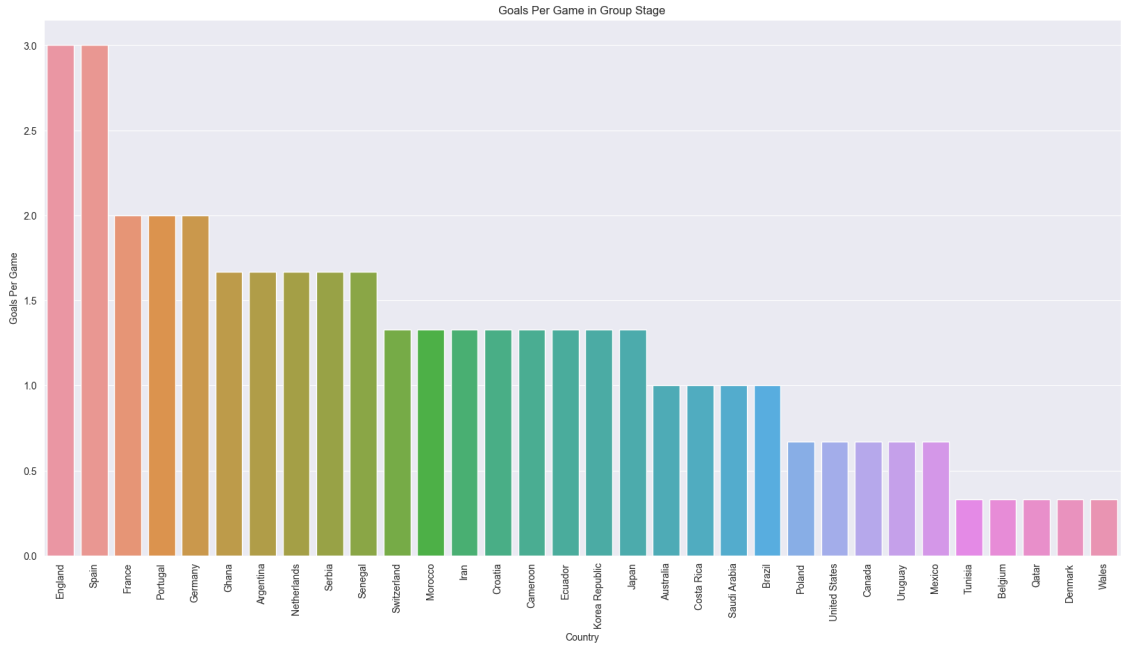
It makes sense that France and Argentina were the two top scorers tournament wide, because they

were the two finalists, and went all the way.

[ ]:
```
# i want this to be colored by groupid but for some reason it doesnt want to␣
 ↪work...any ideas?
plt.figure(figsize=(20, 10))
sns.set_style('darkgrid')

sns.barplot(x=countries['Average Possession in Group'].
 ↪sort_values(ascending=False).index, y=countries['Average Possession in␣
 ↪Group'].sort_values(ascending=False))
plt.xticks(rotation=90)
plt.xlabel('Country')
plt.ylabel('Average Possession %')
plt.title('Average Possession in Group Stage')
plt.show()
```

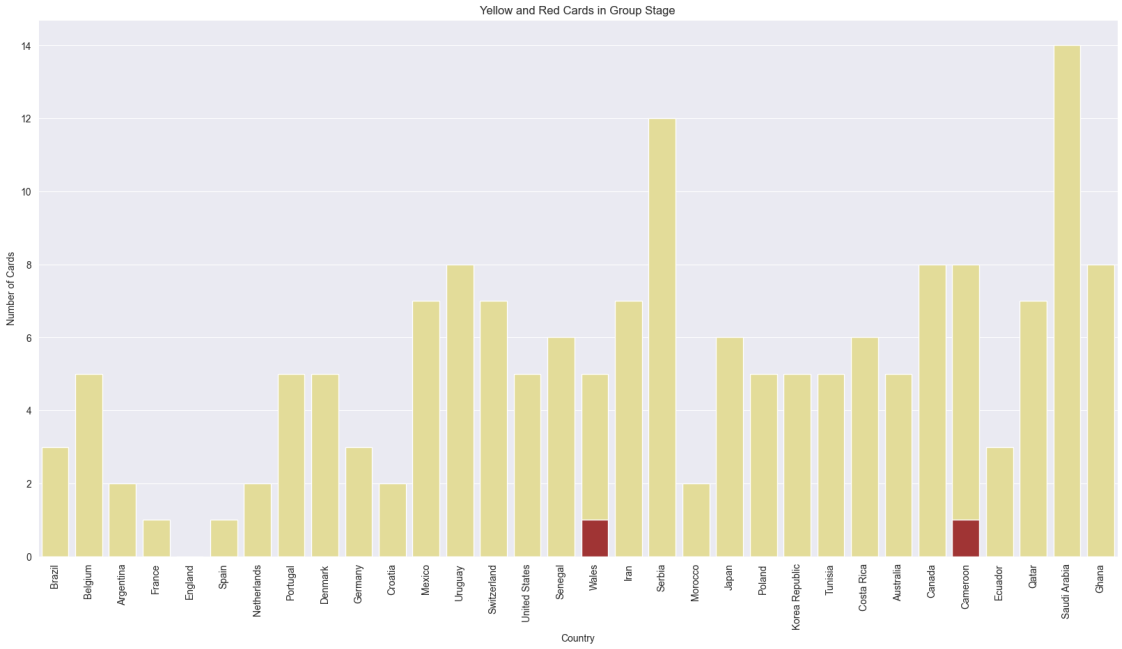


[ ]:
```
plt.figure(figsize=(20, 10))
sns.set_style('darkgrid')

sns.barplot(x=countries['Goals Per Game in Group'].sort_values(ascending=False).
 ↪index, y=countries['Goals Per Game in Group'].sort_values(ascending=False))
plt.xticks(rotation=90)
plt.xlabel('Country')
plt.ylabel('Goals Per Game')
plt.title('Goals Per Game in Group Stage')
plt.show()
```



It appears that this particular visualization gives us many different groups in tiers in which they found goals in the group stages. England and Spain were king among them with 3 goals averaged per match (who were also both in the top 3 of possession as well), while the likes of Tunisia, Belgium, Qatar, Denmark, and Wales were among the lowest with under 0.5 goals per match. Let's look further at these two kings of the Group Stage to see what else they have in common.

[ ]:
```
# create a multibar bar graph comparing yellow and red cards
plt.figure(figsize=(20, 10))
sns.set_style('darkgrid')

sns.barplot(x=countries.index, y=countries['Yellow Cards in Group'],␣
 ↪color='khaki')
sns.barplot(x=countries.index, y=countries['Red Cards in Group'],␣
 ↪color='firebrick')
plt.xticks(rotation=90)
plt.xlabel('Country')
plt.ylabel('Number of Cards')
plt.title('Yellow and Red Cards in Group Stage')
plt.show()
```



We can see that there were a lot of yellow cards distributed during the group stage, but not as many red cards. Wales and Cameroon were the only nations to have a player get sent off during the group stages, and England was the only nation not to pick up a single card during the round. Saudia Arabia, followed by Serbia were by far the most carded nations.

1.6 Visualizing the Round of 16

1.7 Visualizing the Quarter-Finals

1.8 Visualizing the Semi-Finals

1.9 Visualizing the Third-Place Match

1.10 Visualizing the Final