Data Mining Course - Project #2

Professors:
Dr. Farahani, Dr. Kheradpishe

Ali Nikkhah - 99422197

March 2021

Analyzing International football results from 1872 to 2021

Content

This dataset includes 42,082 results of international football matches starting from the very first official match in 1972 up to 2019. The matches range from FIFA World Cup to FIFI Wild Cup to regular friendly matches. The matches are strictly men's full internationals and the data does not include Olympic Games or matches where at least one of the teams was the nation's B-team, U-23 or a league select team.

results.csv includes the following columns:

- date - date of the match
- home_team - the name of the home team
- away_team - the name of the away team
- home_score - full-time home team score including extra time, not including penalty-shootouts
- away_score - full-time away team score including extra time, not including penalty-shootouts
- tournament - the name of the tournament
- city - the name of the city/town/administrative unit where the match was played
- country - the name of the country where the match was played
- neutral - TRUE/FALSE column indicating whether the match was played at a neutral venue

Note on team and country names: For home and away teams the current name of the team has been used. For example, when in 1882 a team who called themselves Ireland played against England, in this dataset, it is called Northern Ireland because the current team of Northern Ireland is the successor of the 1882 Ireland team. This is done so it is easier to track the history and statistics of teams.

For country names, the name of the country at the time of the match is used. So when Ghana played in Accra, Gold Coast in the 1950s, even though the names of the home team and the country don't match, it was a home match for Ghana. This is indicated by the neutral column, which says FALSE for those matches, meaning it was not at a neutral venue.

Acknowledgements

The data is gathered from several sources including but not limited to Wikipedia, fifa.com, rsssf.com and individual football associations' websites.

Inspiration

Some directions to take when exploring the data:

- Who is the best team of all time
- Which teams dominated different eras of football
- What trends have there been in international football throughout the ages - home advantage, total goals scored, distribution of teams' strength etc
- Can we say anything about geopolitics from football fixtures - how has the number of countries changed, which teams like to play each other
- Which countries host the most matches where they themselves are not participating in
- How much, if at all, does hosting a major tournament help a country's chances in the tournament
- Which teams are the most active in playing friendlies and friendly tournaments - does it help or hurt them

The world's your oyster, my friend.

Source: https://www.kaggle.com/martj42/international-football-results-from-1872-to-2017

Working with data

در ابتدا داده را بارگذاری کرده و ستون ها و ویژگی های آن را برسی می‌کنیم.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
matches = pd.read_csv("results.csv")

display(matches.shape)
matches.head(10)
```

```
(42082, 9)
```

Out[2]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False
5	1876-03-25	Scotland	Wales	4	0	Friendly	Glasgow	Scotland	False
6	1877-03-03	England	Scotland	1	3	Friendly	London	England	False
7	1877-03-05	Wales	Scotland	0	2	Friendly	Wrexham	Wales	False
8	1878-03-02	Scotland	England	7	2	Friendly	Glasgow	Scotland	False
9	1878-03-23	Scotland	Wales	9	0	Friendly	Glasgow	Scotland	False

In [3]:
```
matches["tournament"].value_counts().head(30)
```

Out[3]:
```
Friendly                             17189
FIFA World Cup qualification          7363
UEFA Euro qualification               2582
African Cup of Nations qualification  1719
FIFA World Cup                         900
Copa América                           813
AFC Asian Cup qualification            724
African Cup of Nations                 690
CECAFA Cup                             620
CFU Caribbean Cup qualification        606
British Championship                   505
Merdeka Tournament                     503
Gulf Cup                               380
AFC Asian Cup                          370
Island Games                           350
Gold Cup                               327
UEFA Nations League                    304
AFF Championship                       293
COSAFA Cup                             292
UEFA Euro                              286
Nordic Championship                    283
African Nations Championship           264
CFU Caribbean Cup                      251
Amílcar Cabral Cup                     235
King's Cup                             234
South Pacific Games                    205
UNCAF Cup                              164
Korea Cup                              159
Confederations Cup                     140
SAFF Cup                               136
Name: tournament, dtype: int64
```

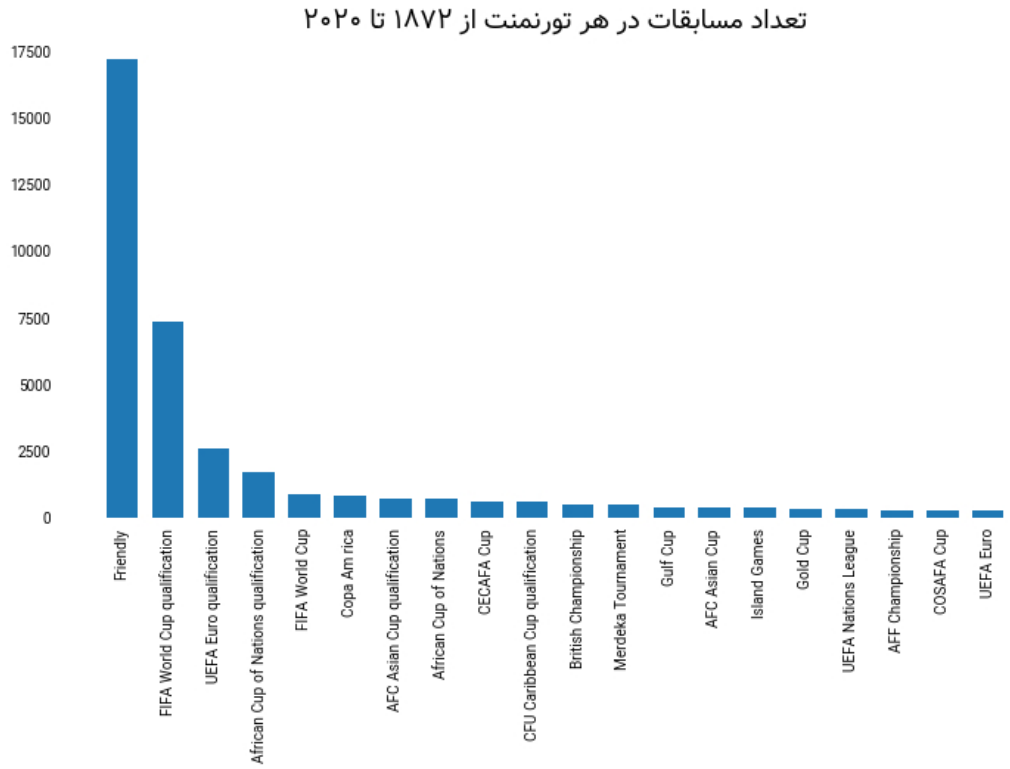
In [4]:
```
value_distribution_top_20 = matches["tournament"].value_counts().head(20)
df_top = value_distribution_top_20.to_frame()
df_top.reset_index(level=0, inplace=True)
df_top.columns = ['Tournament', 'Count']
df_top.head(35)
```

Out[4]:

	Tournament	Count
0	Friendly	17189
1	FIFA World Cup qualification	7363
2	UEFA Euro qualification	2582
3	African Cup of Nations qualification	1719
4	FIFA World Cup	900
5	Copa América	813
6	AFC Asian Cup qualification	724
7	African Cup of Nations	690
8	CECAFA Cup	620

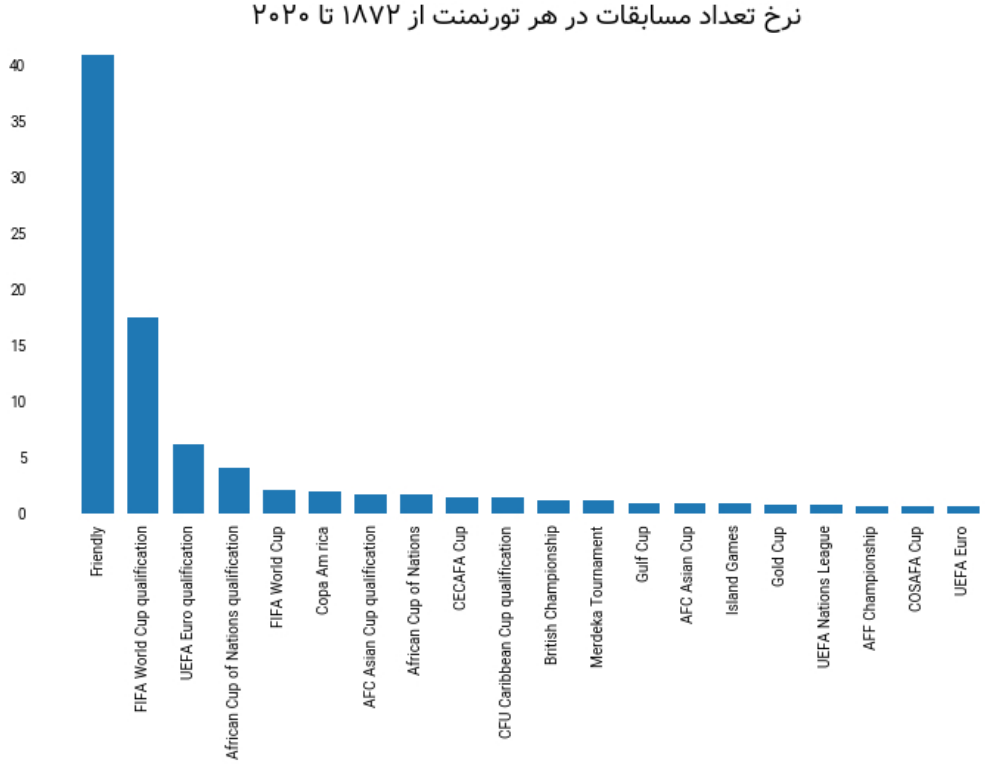
9	CFU Caribbean Cup qualification	606
10	British Championship	505
11	Merdeka Tournament	503
12	Gulf Cup	380
13	AFC Asian Cup	370
14	Island Games	350
15	Gold Cup	327
16	UEFA Nations League	304
17	AFF Championship	293
18	COSAFA Cup	292
19	UEFA Euro	286

In [5]:
```
from matplotlib import font_manager
from bidi.algorithm import get_display
from arabic_reshaper import reshape

font_dirs = ['./fonts/']
font_files = font_manager.findSystemFonts(fontpaths=font_dirs)

font_list = font_manager.createFontList(font_files)
font_manager.fontManager.ttflist.extend(font_list)

# plt.rcParams['font.sans-serif'] = "Comic Sans MS"
plt.rcParams['font.family'] = "Vazir"

fig = plt.figure(figsize=(12,6))
ax = fig.add_subplot(1,1,1)
ax.bar(df_top["Tournament"], df_top["Count"], color='tab:blue', width=0.7)

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.set_xticklabels(df_top["Tournament"], rotation=90)

ax.set_title(get_display(reshape("تعداد مسابقات در هر تورنمنت از ۱۸۷۲ تا ۲۰۲۰")), fontsize=18)
ax.tick_params(bottom = False, left = False)
```



In [6]:
```
value_distribution_top_20 = matches["tournament"].value_counts(normalize=True).head(20)
df_top_percentages = value_distribution_top_20.to_frame()
df_top_percentages.reset_index(level=0, inplace=True)
df_top_percentages.columns = ['Tournament', 'Percentage']
df_top_percentages.head(35)

fig = plt.figure(figsize=(12,6))
ax = fig.add_subplot(1,1,1)
ax.bar(df_top_percentages["Tournament"], df_top_percentages["Percentage"]*100, color='tab:blue', width=0.7)

ax.spines['right'].set_visible(False)
```

```
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.set_xticklabels(df_top_percentages["Tournament"], rotation=90)

ax.set_title(get_display(reshape(" ۲۰۲۰ تا ۱۸۷۲ نرخ تعداد مسابقات در هر تورنمنت از")), fontsize=18)
ax.tick_params(bottom = False, left = False)
```



از ۴۲۰۸۲ مسابقه ۱۷۱۸۹ آن دوستانه بوده است یعنی تقریبا ۴۰٪

از آنجایی که مسابقات دوستانه ملاک خوبی برای ارزیابی نیستند از آنها صرف نظر می‌کنیم و مسابقات رسمی را معیار قرار می‌دهیم.

اما بعد انجام تحلیل می‌توانیم با استفاده از داده های بازی های دوستانه مقایسه هایی را انجام دهیم.

In [7]:
```
friendly_matches = matches[matches["tournament"] == "Friendly"].copy()
friendly_matches.shape
```

Out[7]:
```
(17189, 9)
```

قبل از تحلیل بازی های رسمی میتوان تعداد گل های زده شده در هر بازی دوستانه و رسمی را برسی کرد.

In [8]:
```
friendly_matches["total_goals"] = friendly_matches["home_score"] + friendly_matches["away_score"]

goal_per_game_on_average = friendly_matches["total_goals"].sum()/friendly_matches.shape[0]
#or goal_per_game_on_average = friendly_matches["total_goals].mean()

print("Average number of goals in frienldy matches is {}".format(goal_per_game_on_average))
```

```
Average number of goals in frienldy matches is 2.896503577869568
```

In [9]:
```
#and now let's compare it to the average goals per game in official matches

official_matches = matches[matches["tournament"] != "Friendly"].copy()
display(official_matches.shape)

official_matches["total_goals"] = official_matches["home_score"] + official_matches["away_score"]

print("Average number of goals in official matches is {}".format(official_matches["total_goals"].mean()))
```

```
(24893, 9)
Average number of goals in official matches is 2.953521070180372
```

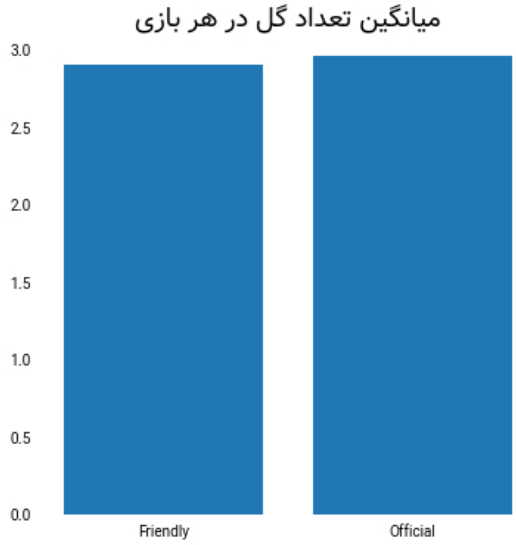
همانطور که مشاهده می‌کنید میانگین تعداد گل در مسابقات دوستانه از رسمی کمتر است.

In [10]:
```
data = ({'Match_types': ['Friendly', 'Official'], 'Average_number_of_goals_per_game': [2.898702213870456, 2.957
df_new = pd.DataFrame(data)
df_new.head()
```

Out[10]:

	Match_types	Average_number_of_goals_per_game
0	Friendly	2.898702
1	Official	2.957364

In [11]:
```
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(1,1,1)

ax.bar(df_new["Match_types"], df_new['Average_number_of_goals_per_game'], color='tab:blue')

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.set_title(get_display(reshape("میانگین تعداد گل در هر بازی")), fontsize=18)
ax.tick_params(bottom = False, left = False)
```



همانطور که مشاهده می‌کنید در کمال تعجب میانگین تعداد گل در مسابقات دوستانه از رسمی کمتر است. این بر خلاف انتظار است چرا که بازی های دوستانه از رقابت کمتر و سرگرمی بیشتری برخوردار هستند.

Main Analytics

حال به بررسی هر ستون یا ویژگی می‌پردازیم.

In [12]:
```
display(official_matches.shape)
official_matches.reset_index(level=0, inplace=True)
official_matches = official_matches.drop(['index'], axis=1)

display(official_matches.head(10))
official_matches.tail(10)
```

```
(24893, 10)
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	total_goals
0	1884-01-26	Northern Ireland	Scotland	0	5	British Championship	Belfast	Republic of Ireland	False	5
1	1884-02-09	Wales	Northern Ireland	6	0	British Championship	Wrexham	Wales	False	6
2	1884-02-23	Northern Ireland	England	1	8	British Championship	Belfast	Republic of Ireland	False	9
3	1884-03-15	Scotland	England	1	0	British Championship	Glasgow	Scotland	False	1
4	1884-03-17	Wales	England	0	4	British Championship	Wrexham	Wales	False	4
5	1884-03-29	Scotland	Wales	4	1	British Championship	Glasgow	Scotland	False	5
6	1885-02-28	England	Northern Ireland	4	0	British Championship	Manchester	England	False	4
7	1885-03-14	England	Wales	1	1	British Championship	Blackburn	England	False	2
8	1885-03-14	Scotland	Northern Ireland	8	2	British Championship	Glasgow	Scotland	False	10
9	1885-03-21	England	Scotland	1	1	British Championship	London	England	False	2

Out[12]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	total_goals
24883	2021-03-31	Ukraine	Kazakhstan	1	1	FIFA World Cup qualification	Kiev	Ukraine	False	2
24884	2021-03-31	Scotland	Faroe Islands	4	0	FIFA World Cup qualification	Glasgow	Scotland	False	4
24885	2021-03-31	Austria	Denmark	0	4	FIFA World Cup qualification	Vienna	Austria	False	4
24886	2021-03-	Moldova	Israel	1	4	FIFA World Cup	Chisinău	Moldova	False	5

	31					qualification				
24887	2021-03-31	England	Poland	2	1	FIFA World Cup qualification	London	England	False	3
24888	2021-03-31	Andorra	Hungary	1	4	FIFA World Cup qualification	Andorra la Vella	Andorra	False	5
24889	2021-03-31	San Marino	Albania	0	2	FIFA World Cup qualification	Serravalle	San Marino	False	2
24890	2021-03-31	Armenia	Romania	3	2	FIFA World Cup qualification	Yerevan	Armenia	False	5
24891	2021-03-31	Germany	North Macedonia	1	2	FIFA World Cup qualification	Duisburg	Germany	False	3
24892	2021-03-31	Liechtenstein	Iceland	1	4	FIFA World Cup qualification	Vaduz	Liechtenstein	False	5

In [13]:
```
official_matches.describe(include='all')
```

Out[13]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	total_goals
count	24893	24893	24893	24893.000000	24893.000000	24893	24893	24893	24893	24893.000000
unique	9025	293	291	NaN	NaN	111	1274	242	2	NaN
top	2008-10-11	Argentina	Uruguay	NaN	NaN	FIFA World Cup qualification	Kuala Lumpur	Malaysia	False	NaN
freq	52	384	332	NaN	NaN	7363	529	565	16845	NaN
mean	NaN	NaN	NaN	1.777809	1.175712	NaN	NaN	NaN	NaN	2.953521
std	NaN	NaN	NaN	1.845382	1.460573	NaN	NaN	NaN	NaN	2.134246
min	NaN	NaN	NaN	0.000000	0.000000	NaN	NaN	NaN	NaN	0.000000
25%	NaN	NaN	NaN	1.000000	0.000000	NaN	NaN	NaN	NaN	1.000000
50%	NaN	NaN	NaN	1.000000	1.000000	NaN	NaN	NaN	NaN	3.000000
75%	NaN	NaN	NaN	2.000000	2.000000	NaN	NaN	NaN	NaN	4.000000
max	NaN	NaN	NaN	31.000000	21.000000	NaN	NaN	NaN	NaN	31.000000

In [14]:
```
official_matches.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24893 entries, 0 to 24892
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   date         24893 non-null  object
 1   home_team    24893 non-null  object
 2   away_team    24893 non-null  object
 3   home_score   24893 non-null  int64 
 4   away_score   24893 non-null  int64 
 5   tournament   24893 non-null  object
 6   city         24893 non-null  object
 7   country      24893 non-null  object
 8   neutral      24893 non-null  bool  
 9   total_goals  24893 non-null  int64 
dtypes: bool(1), int64(3), object(6)
memory usage: 1.7+ MB
```

date

از آنجایی که این دیتاست شامل بازی های بین المللی در بازه ۱۴۸ ساله است، می‌توانیم داده را بر اساس بازه زمانی تقسیم کنیم و مقایسه‌هایی بین دهه‌های مختلف انجام دهیم.

home_team and away_team

می‌توانیم بفمیم کدام تیم بیشترین بازی‌ها را انجام داده است و آیا ارتباطی بین موفقیت و قهرمانی وجود دارد؟

home_score and away_score

می‌توانیم میانگین، بیشترین و کمترین را برای هر یک برسی کنیم.

tournament

تا اینجا در مورد بازی‌های دوستانه اطلاعاتی بدست آوردیم(تقریبا ۴۰٪ بازی‌ها دوستانه بودند)، در مورد بقیه بازی ها که رسمی هستند هم ۱۱۱ تورنمت مختلف وجود دارد که جام جهانی فیفا بیشترین تعداد مسابقات را دارد.

city and country

می‌توانیم بفمیم کدام کشورها بیشترین میزبانی را داشته اند که کشور مالزی و شهر کوالالامپور دارای بیشترین تعداد است اما میتوان اطلاعات بیشتری هم بدست آورد.

neutral

حدود ۶۰۰۰ بازی وجود دارد که در مکانی بیطرف برگزار شده‌اند! بیشتر این بازی‌ها مربوط به جام جهانی فیفا، اروپا و کوپا امریکا است اما نه همه آن.

1. Who is the best team of all time?

In [15]:
```
def detect_winner(home_score, away_score, home_team, away_team):
    if home_score>away_score:
        return home_team
    elif home_score<away_score:
        return away_team
    else:
        pass

official_matches['winner'] = official_matches.apply(
    lambda x: detect_winner(x['home_score'], x['away_score'], x['home_team'], x['away_team']), axis=1)



def detect_loser(home_score, away_score, home_team, away_team):
    if home_score>away_score:
        return away_team
    elif home_score<away_score:
        return home_team
    else:
        pass

official_matches['loser'] = official_matches.apply(
    lambda x: detect_loser(x['home_score'], x['away_score'], x['home_team'], x['away_team']), axis=1)

display(official_matches.shape)
official_matches.head(10)
```

```
(24893, 12)
```

Out[15]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	total_goals	winner	loser
0	1884-01-26	Northern Ireland	Scotland	0	5	British Championship	Belfast	Republic of Ireland	False	5	Scotland	Northern Ireland
1	1884-02-09	Wales	Northern Ireland	6	0	British Championship	Wrexham	Wales	False	6	Wales	Northern Ireland
2	1884-02-23	Northern Ireland	England	1	8	British Championship	Belfast	Republic of Ireland	False	9	England	Northern Ireland
3	1884-03-15	Scotland	England	1	0	British Championship	Glasgow	Scotland	False	1	Scotland	England
4	1884-03-17	Wales	England	0	4	British Championship	Wrexham	Wales	False	4	England	Wales
5	1884-03-29	Scotland	Wales	4	1	British Championship	Glasgow	Scotland	False	5	Scotland	Wales
6	1885-02-28	England	Northern Ireland	4	0	British Championship	Manchester	England	False	4	England	Northern Ireland
7	1885-03-14	England	Wales	1	1	British Championship	Blackburn	England	False	2	None	None
8	1885-03-14	Scotland	Northern Ireland	8	2	British Championship	Glasgow	Scotland	False	10	Scotland	Northern Ireland
9	1885-03-21	England	Scotland	1	1	British Championship	London	England	False	2	None	None

In [16]:
```
display(official_matches['winner'].isnull().sum())
official_matches['winner'].value_counts().head(20)
```

```
5388
```

Out[16]:
```
England          353
Argentina        341
Brazil           335
South Korea      321
Scotland         273
Uruguay          262
Sweden           262
Germany          257
Italy            241
Mexico           234
Spain            233
Denmark          216
Netherlands      211
France           210
Russia           205
Japan            200
Zambia           197
Portugal         193
Saudi Arabia     193
Thailand         190
Name: winner, dtype: int64
```

همانطور که می‌بینید ۵۳۸۸ بازی از ۲۴۸۹۳ بازی نتیجه مساوی داشته‌اند. همچنین انگلستان بیشترین تعداد برد را داشته است، اما بهتر است تعداد کل بازی ها را هم برسی کنیم که نرخ برد را هم بدست آوریم برای این کار فریم داده زیر را ایجاد می‌کنیم.

- "Team"
- "Games"
- "Wins"
- "Draws"
- "Loses"
- "Games to win ratio"
- "Goals Scored"
- "Goals Conceded"

Detecting Total Number of Games Played

In [17]:
```
df_home_teams = official_matches['home_team'].value_counts().to_frame()
df_home_teams.reset_index(level=0, inplace=True)
df_home_teams.columns = ['team', 'count']
display(df_home_teams.shape)
df_home_teams.head()
```

```
(293, 2)
```

Out[17]:

	team	count
0	Argentina	384
1	Brazil	342
2	South Korea	314
3	England	308
4	Scotland	278

In [18]:
```
df_away_teams = official_matches['away_team'].value_counts().to_frame()
df_away_teams.reset_index(level=0, inplace=True)
df_away_teams.columns = ['team', 'count']
display(df_away_teams.shape)
df_away_teams.head()
```

```
(291, 2)
```

Out[18]:

	team	count
0	Uruguay	332
1	England	297
2	Scotland	281
3	Northern Ireland	280
4	Paraguay	270

In [19]:
```
df_all_games = pd.merge(left=df_home_teams, right=df_away_teams, how='outer', on='team')
display(df_all_games.shape)
df_all_games.head()
```

```
(296, 3)
```

Out[19]:

	team	count_x	count_y
0	Argentina	384.0	229.0
1	Brazil	342.0	211.0
2	South Korea	314.0	259.0
3	England	308.0	297.0
4	Scotland	278.0	281.0

In [20]:
```
df_all_games = df_all_games.fillna(0)
df_all_games['games'] = df_all_games['count_x'] + df_all_games['count_y']
display(df_all_games.head(10))
display(df_all_games.tail(10))

#while doing an outer join as a result we had some NaNs in the dataset. (Example Monaco had only 3 away games, i
#so we replaced NaN-s with zeros instead.
```

	team	count_x	count_y	games
0	Argentina	384.0	229.0	613.0
1	Brazil	342.0	211.0	553.0
2	South Korea	314.0	259.0	573.0
3	England	308.0	297.0	605.0
4	Scotland	278.0	281.0	559.0
5	Thailand	256.0	268.0	524.0
6	Northern Ireland	255.0	280.0	535.0
7	Wales	254.0	257.0	511.0
8	Chile	247.0	205.0	452.0

9	Uruguay	242.0	332.0	574.0

	team	count_x	count_y	games
286	Yemen DPR	1.0	6.0	7.0
287	North Vietnam	1.0	4.0	5.0
288	Romani people	1.0	2.0	3.0
289	Yorkshire	1.0	0.0	1.0
290	Kabylia	1.0	5.0	6.0
291	Cascadia	1.0	5.0	6.0
292	Micronesia	1.0	3.0	4.0
293	Western Australia	0.0	29.0	29.0
294	Two Sicilies	0.0	4.0	4.0
295	Monaco	0.0	3.0	3.0

In [21]:
```
df_all_games = df_all_games.drop(["count_x", "count_y"], axis=1)

display(df_all_games.shape)
df_all_games.head(10)
```

```
(296, 2)
```

Out[21]:

	team	games
0	Argentina	613.0
1	Brazil	553.0
2	South Korea	573.0
3	England	605.0
4	Scotland	559.0
5	Thailand	524.0
6	Northern Ireland	535.0
7	Wales	511.0
8	Chile	452.0
9	Uruguay	574.0

Detecting Wins and Loses

In [22]:
```
df_winners = official_matches['winner'].value_counts().to_frame()
df_winners.reset_index(level=0, inplace=True)
df_winners.columns = ['team', 'wins']
display(df_winners.shape)
df_winners.head()
```

```
(285, 2)
```

Out[22]:

	team	wins
0	England	353
1	Argentina	341
2	Brazil	335
3	South Korea	321
4	Scotland	273

In [23]:
```
df_losers = official_matches['loser'].value_counts().to_frame()
df_losers.reset_index(level=0, inplace=True)
df_losers.columns = ['team', 'loses']
display(df_losers.shape)
df_losers.head()
```

```
(294, 2)
```

Out[23]:

	team	loses
0	Northern Ireland	283
1	Finland	239
2	Wales	236
3	Luxembourg	222
4	Thailand	211

In [24]:
```
df_winners_and_losers = pd.merge(left=df_winners, right=df_losers, how='outer', on='team')
df_winners_and_losers = df_winners_and_losers.fillna(0)

df_winners_and_losers.head()
```

Out[24]:

	team	wins	loses
0	England	353.0	107.0
1	Argentina	341.0	132.0
2	Brazil	335.0	102.0
3	South Korea	321.0	103.0
4	Scotland	273.0	162.0

In [25]:
```
df_five = pd.merge(left=df_all_games, right=df_winners_and_losers, how='outer', on='team')
df_five['ratio'] = (df_five['wins']/df_five['games'])

display(df_five.shape)
df_five.head()
```

```
(296, 5)
```

Out[25]:

	team	games	wins	loses	ratio
0	Argentina	613.0	341.0	132.0	0.556281
1	Brazil	553.0	335.0	102.0	0.605787
2	South Korea	573.0	321.0	103.0	0.560209
3	England	605.0	353.0	107.0	0.583471
4	Scotland	559.0	273.0	162.0	0.488372

In [26]:
```
#detecting draws based on the date we already have in the df above/adding draw column
df_five['draws'] = df_five['games'] - df_five['wins'] - df_five['loses']
```

مشکلی که وجود دارد این است که تیم هایی با فقط ۲ بازی رسمی هم وجود دارند که این باعث خطا در نرخ بردن می‌شود، برای جلوگیری از این می‌توانیم فقط تیم هایی که حداقل ۱۰۰ بازی رسمی داشته اند را فیلتر کنیم و بر اساس نرخ برد مرتب کنیم.

In [27]:
```
df_five = df_five[df_five['games'] > 150]

df_five.shape
```

Out[27]:
```
(137, 6)
```

In [28]:
```
df_five = df_five.sort_values('ratio', ascending=False)

#this next 2 lines are for first reseting the index so it starts from zero again, and delete the new added 'ind
df_five.reset_index(level=0, inplace=True)
df_five = df_five.drop(['index'], axis=1)

df_five.head(10)
```

Out[28]:

	team	games	wins	loses	ratio	draws
0	Germany	389.0	257.0	54.0	0.660668	78.0
1	Spain	368.0	233.0	62.0	0.633152	73.0
2	Netherlands	344.0	211.0	69.0	0.613372	64.0
3	Brazil	553.0	335.0	102.0	0.605787	116.0
4	Iran	296.0	175.0	46.0	0.591216	75.0
5	England	605.0	353.0	107.0	0.583471	145.0
6	France	363.0	210.0	73.0	0.578512	80.0
7	Russia	357.0	205.0	78.0	0.574230	74.0
8	Czech Republic	197.0	113.0	50.0	0.573604	34.0
9	Italy	424.0	241.0	67.0	0.568396	116.0

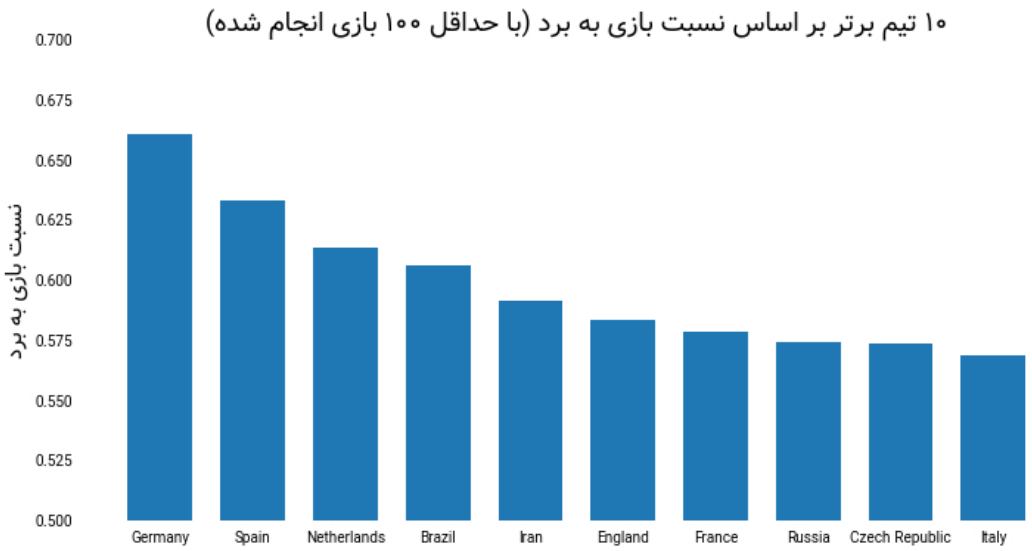
In [29]:
```
#rearranging the order of columns

df_five = df_five[['team', 'games', 'wins', 'draws', 'loses', 'ratio']].copy()
df_five.rename({"ratio": "games to win ratio"}, axis=1, inplace=True)
df_five.head(10)
```

Out[29]:

	team	games	wins	draws	loses	games to win ratio
0	Germany	389.0	257.0	78.0	54.0	0.660668
1	Spain	368.0	233.0	73.0	62.0	0.633152
2	Netherlands	344.0	211.0	64.0	69.0	0.613372
3	Brazil	553.0	335.0	116.0	102.0	0.605787
4	Iran	296.0	175.0	75.0	46.0	0.591216
5	England	605.0	353.0	145.0	107.0	0.583471
6	France	363.0	210.0	80.0	73.0	0.578512
7	Russia	357.0	205.0	74.0	78.0	0.574230

8	Czech Republic	197.0	113.0	34.0	50.0	0.573604
9	Italy	424.0	241.0	116.0	67.0	0.568396

In [30]:
```
fig = plt.figure(figsize=(12,6))
ax = fig.add_subplot(1,1,1)
ax.bar(df_five["team"][:10], df_five["games to win ratio"][:10], color='tab:blue', width=0.7)

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_visible(False)

ax.set_ylim(0.5, 0.7)

ax.set_title(get_display(reshape("۱۰ (با حداقل ۱۰۰ بازی انجام شده) تیم برتر بر اساس نسبت بازی به برد")), fontsi

ax.set_ylabel(get_display(reshape("نسبت بازی به برد")),fontsize=15)
ax.tick_params(bottom = False, left = False)
```



با نگاه به نمودار میله‌ای بالا میتوان نتیجه گیری هایی انجام داد.
اولا در فوتبال بهترین تیم ها تیم هایی هستند که نه تنها در طول سال ها از نظر میانگین نرخ برد و نتیجه بهتر بوده‌اند، بلکه بدون توجه به سایر آمارها جام های بزرگ را تصاحب کرده‌اند.
حال لیست برندگان جام های مختلف را تجزیه و تحلیل کرده و بر اساس این دو عامل نتیجه گیری می‌کنیم.

EURO

- Germany, Spain - 3
- France - 2
- Russia, Italy, Czech, Portugal, Netherlands, Denmark, Greece - 1

WORLD CUP

- Brazil - 5
- Germany, Italy - 4
- Argentina, France, Uruguay - 2
- England, Spain - 1

COPA AMERICA

- Uruguay - 15
- Argentina - 14
- Brazil - 9

بنابراین با در نظر گرفتن این داده ها ترکیب زیر را خواهیم داشت.

- Germany = 4 World Cups + 3 Euros
- Spain = 1 World Cups + 3 Euros
- Netherlands = 1 Euros
- Brazil = 5 World Cups + 9 Copa Americas
- England = 1 World Cup
- Russia = 1 Euros
- Czech = 1 Euros
- France = 2 World Cups + 2 Euros
- Italy = 4 World Cups + 1 Euros

با توجه به داده های بالا باید تیم های هلند، روسیه، انگلستان و چک را کنار بگذاریم زیراکه برد یک جام صرفا برای بهترین تیم بودن کافی نیست. پس بین ۵ تیم باقیمانده باید انتخاب کنیم.

- Germany = 4 World Cups + 3 Euros
- Spain = 1 World Cups + 3 Euros
- Brazil = 5 World Cups + 9 Copa Americas
- France = 2 World Cups + 2 Euros
- Italy = 4 World Cups + 1 Euros

به راحتی می‌توان گفت که با توجه به تعداد جام و نرخ بازی به برد بهترین تیم آلمان یا برزیل است.
انتخاب بین این دو کمی دشوار است اما از آنجایی که جام کوپا آمریکا کلاس پایین تری نسبت به جام یورو دارد، تیم آلمان امتیاز بیشتری می‌گیرد.
همچنین با مقایسه آمار بین این دو تیم داریم.

- World Cups won: **Brazil 5** - Germany 4
- Games to win ratio: Brazil 0.61 - **0.66 Germany**

آمار وضعیت تساوی را نشان می‌دهد
نکته جالب، اینکه تیمی که بیشترین تعداد برد را داشته است انگلستان است با ۴۳۰ برد!
حال باز هم می‌توانیم جلوتر برویم و ستون های بیشتری را معیار قرار دهیم البته که همه اینها بر اساس نرخ بازی به برد خواهد بود.

Finalizing the table with Goals and Points

In [31]:
```
official_matches.head(10)
```

Out[31]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	total_goals	winner	loser
0	1884-01-26	Northern Ireland	Scotland	0	5	British Championship	Belfast	Republic of Ireland	False	5	Scotland	Northern Ireland
1	1884-02-09	Wales	Northern Ireland	6	0	British Championship	Wrexham	Wales	False	6	Wales	Northern Ireland
2	1884-02-23	Northern Ireland	England	1	8	British Championship	Belfast	Republic of Ireland	False	9	England	Northern Ireland
3	1884-03-15	Scotland	England	1	0	British Championship	Glasgow	Scotland	False	1	Scotland	England
4	1884-03-17	Wales	England	0	4	British Championship	Wrexham	Wales	False	4	England	Wales
5	1884-03-29	Scotland	Wales	4	1	British Championship	Glasgow	Scotland	False	5	Scotland	Wales
6	1885-02-28	England	Northern Ireland	4	0	British Championship	Manchester	England	False	4	England	Northern Ireland
7	1885-03-14	England	Wales	1	1	British Championship	Blackburn	England	False	2	None	None
8	1885-03-14	Scotland	Northern Ireland	8	2	British Championship	Glasgow	Scotland	False	10	Scotland	Northern Ireland
9	1885-03-21	England	Scotland	1	1	British Championship	London	England	False	2	None	None

In [32]:
```
df_five.head()
```

Out[32]:

	team	games	wins	draws	loses	games to win ratio
0	Germany	389.0	257.0	78.0	54.0	0.660668
1	Spain	368.0	233.0	73.0	62.0	0.633152
2	Netherlands	344.0	211.0	64.0	69.0	0.613372
3	Brazil	553.0	335.0	116.0	102.0	0.605787
4	Iran	296.0	175.0	75.0	46.0	0.591216

حال ۴ دیتا فریم جدید از طریق groupby ایجاد می‌کنیم

- GOALS SCORED AT HOME
- GOALS SCORED AWAY
- GOALS CONCEDED AT HOME
- GOALS CONCEDED AWAY

سپس ۴ جدول را ادغام می‌کنیم(outer join) و بعد از آن به راحتی ۴ ستون خواهیم داشت که تعداد گل ها و تفاضل آنهارا نشان می‌دهند.

In [33]:
```
#GOALS SCORED AT HOME

goals_scored_at_home = official_matches.groupby("home_team").agg({'home_score':'sum'})
goals_scored_at_home.reset_index(level=0, inplace=True)
goals_scored_at_home.columns = ['team', 'scored']
```

```
goals_scored_at_home.columns = ['team', 'scored']

display(goals_scored_at_home.shape)
goals_scored_at_home.sort_values('scored', ascending=False).head()
```

```
(293, 2)
```

Out[33]:

	team	scored
11	Argentina	881
33	Brazil	857
77	England	742
239	South Korea	702
225	Scotland	554

In [34]:
```
#GOAL SCORED AWAY

goals_scored_away = official_matches.groupby("away_team").agg({'away_score':'sum'})
goals_scored_away.reset_index(level=0, inplace=True)
goals_scored_away.columns = ['team', 'scored']

display(goals_scored_away.shape)
goals_scored_away.sort_values('scored', ascending=False).head()
```

```
(291, 2)
```

Out[34]:

	team	scored
76	England	579
271	Uruguay	504
243	Sweden	449
222	Scotland	435
236	South Korea	405

In [35]:
```
#GOALS CONCEDED AT HOME

goals_conceded_at_home = official_matches.groupby("home_team").agg({'away_score':'sum'})
goals_conceded_at_home.reset_index(level=0, inplace=True)
goals_conceded_at_home.columns = ['team', 'conceded']

display(goals_conceded_at_home.shape)
goals_conceded_at_home.sort_values('conceded', ascending=False).head()
```

```
(293, 2)
```

Out[35]:

	team	conceded
183	Northern Ireland	446
87	Finland	368
279	Wales	360
11	Argentina	348
147	Luxembourg	340

In [36]:
```
#GOALS CONCEDED AWAY

goals_conceded_away = official_matches.groupby("away_team").agg({'home_score':'sum'})
goals_conceded_away.reset_index(level=0, inplace=True)
goals_conceded_away.columns = ['team', 'conceded']

display(goals_conceded_away.shape)
goals_conceded_away.sort_values('conceded', ascending=False).head()
```

```
(291, 2)
```

Out[36]:

	team	conceded
181	Northern Ireland	596
86	Finland	540
253	Thailand	524
193	Paraguay	498
277	Wales	487

In [37]:
```
#Now we'll add this 4 tables together.

total_goals_scored = pd.merge(left=goals_scored_at_home, right=goals_scored_away, how='outer', on='team')
total_goals_scored = total_goals_scored.fillna(0)

total_goals_conceded = pd.merge(left=goals_conceded_at_home, right=goals_conceded_away, how='outer', on='team')
total_goals_conceded = total_goals_conceded.fillna(0)

all_goals = pd.merge(left=total_goals_scored, right=total_goals_conceded, how='outer', on='team')
all_goals = all_goals.fillna(0)
```

```
all_goals['total scored'] = all_goals['scored_x'] + all_goals['scored_y']
all_goals['total conceded'] = all_goals['conceded_y'] + all_goals['conceded_y']
all_goals['goal difference'] = all_goals['total scored'] - all_goals['total conceded']

all_goals = all_goals.sort_values('goal difference', ascending=False)
all_goals.reset_index(level=0, inplace=True)
all_goals = all_goals.drop(['index'], axis=1)

all_goals_updated = all_goals[['team','total scored','total conceded','goal difference']].copy()

display(all_goals_updated.shape)
all_goals_updated.head(15)
```

```
(296, 4)
```

Out[37]:

	team	total scored	total conceded	goal difference
0	England	1321.0	544.0	777.0
1	Brazil	1219.0	466.0	753.0
2	South Korea	1107.0	496.0	611.0
3	Argentina	1205.0	612.0	593.0
4	Germany	915.0	362.0	553.0
5	Spain	801.0	378.0	423.0
6	Japan	725.0	322.0	403.0
7	Netherlands	749.0	350.0	399.0
8	Iran	614.0	224.0	390.0
9	Australia	632.0	246.0	386.0
10	France	696.0	330.0	366.0
11	Mexico	851.0	524.0	327.0
12	Italy	733.0	406.0	327.0
13	China PR	580.0	260.0	320.0
14	Sweden	979.0	672.0	307.0

In [38]:

```
#Now let's merge this with the rest

df_final = pd.merge(left=df_five, right=all_goals_updated, how='outer', on='team')
df_final = df_final.fillna(0)

display(df_final.shape)
df_final.head(10)
```

```
(296, 9)
```

Out[38]:

	team	games	wins	draws	loses	games to win ratio	total scored	total conceded	goal difference
0	Germany	389.0	257.0	78.0	54.0	0.660668	915.0	362.0	553.0
1	Spain	368.0	233.0	73.0	62.0	0.633152	801.0	378.0	423.0
2	Netherlands	344.0	211.0	64.0	69.0	0.613372	749.0	350.0	399.0
3	Brazil	553.0	335.0	116.0	102.0	0.605787	1219.0	466.0	753.0
4	Iran	296.0	175.0	75.0	46.0	0.591216	614.0	224.0	390.0
5	England	605.0	353.0	145.0	107.0	0.583471	1321.0	544.0	777.0
6	France	363.0	210.0	80.0	73.0	0.578512	696.0	330.0	366.0
7	Russia	357.0	205.0	74.0	78.0	0.574230	664.0	386.0	278.0
8	Czech Republic	197.0	113.0	34.0	50.0	0.573604	356.0	200.0	156.0
9	Italy	424.0	241.0	116.0	67.0	0.568396	733.0	406.0	327.0

پس انگلستان نه تنها بیشترین تعداد گل را دارد، بلکه تیمی است که بیشترین تعداد امتیاز و تفاضل گل را داشته است.
حال بیایید فرمولی را تعریف کنیم که امتیاز هر تیم را محاسبه کند و سپس ستونهای "امتیاز" و "امتیازات در هر بازی" را اضافه کند. سپس جدول را بر اساس ستون "امتیازات در هر بازی" مرتب می‌کنیم.

In [39]:

```
df_final['points'] = df_final['wins']*3 + df_final['draws']*1
df_final['points per game'] = df_final['points']/df_final['games']
df_final = df_final.sort_values('points per game', ascending=False)

#this next 2 lines are for first reseting the index so it starts from zero again, and then dropping the old one
df_final.reset_index(level=0, inplace=True)
df_final = df_final.drop(['index'], axis=1)

display(df_final.shape)
df_final.head(10)
```

```
(296, 11)
```

Out[39]:

	team	games	wins	draws	loses	games to win ratio	total scored	total conceded	goal difference	points	points per game
0	Germany	389.0	257.0	78.0	54.0	0.660668	915.0	362.0	553.0	849.0	2.182519

1	Spain	368.0	233.0	73.0	62.0	0.633152	801.0	378.0	423.0	772.0	2.097826
2	Brazil	553.0	335.0	116.0	102.0	0.605787	1219.0	466.0	753.0	1121.0	2.027125
3	Iran	296.0	175.0	75.0	46.0	0.591216	614.0	224.0	390.0	600.0	2.027027
4	Netherlands	344.0	211.0	64.0	69.0	0.613372	749.0	350.0	399.0	697.0	2.026163
5	England	605.0	353.0	145.0	107.0	0.583471	1321.0	544.0	777.0	1204.0	1.990083
6	Italy	424.0	241.0	116.0	67.0	0.568396	733.0	406.0	327.0	839.0	1.978774
7	France	363.0	210.0	80.0	73.0	0.578512	696.0	330.0	366.0	710.0	1.955923
8	South Korea	573.0	321.0	149.0	103.0	0.560209	1107.0	496.0	611.0	1112.0	1.940663
9	Russia	357.0	205.0	74.0	78.0	0.574230	664.0	386.0	278.0	689.0	1.929972

با نگاهی به این جدول جدید که بر اساس امتیاز در هر بازی مرتب شده است و همچنین جام های مهم را در نظر می گیریم، یک بار دیگر می توان 3 مکان زیر را به وضوح تشخیص داد.

1. Germany, Brazil
2. Spain, Italy, France
3. Netherlands, England

بله، هر 3 مکان به اشتراک گذاشته شده است، زیرا انتخاب بین آنها واقعاً اختیاری است، اما تفاوت بین هر گروه به اندازه کافی مشخص است تا آنها را در گروه ها قرار دهد.

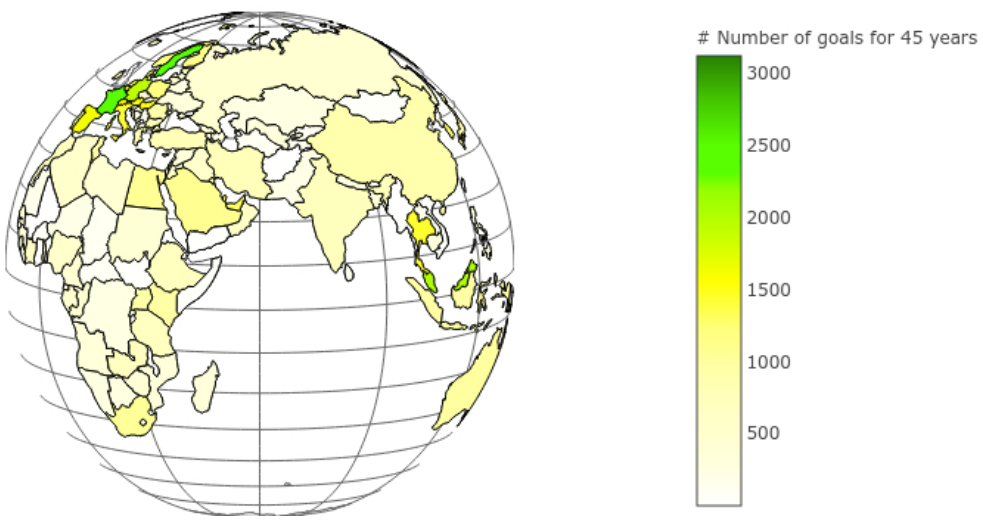
حقایق جالب

- انگلستان بیشترین بازی رسمی را انجام داده است و بعد از آن برزیل
- انگلستان بیشترین گل را در مسابقات رسمی به ثمر رسانده است و بعد از آن برزیل.
- انگلستان بهترین اختلاف گل را در تمام دوران دارد و بعد از آن برزیل.
- هلند را می‌توان بد شانس ترین تیم دانست، از آنجاییکه همیشه از نظر نسبت بازی به برد، امتیاز در هر بازی و تفاضل گل در بالای جدول بوده اما موفق به کسب جام نشده است.
- حتی اگر فوتبال 148 سال پیش ظهور کرده باشد، ۷۱ درصد از کل مسابقات در ۳۴ سال گذشته برگزار شده است و تنها ۶ درصد از تمام مسابقات در ۸۵ سال اول (۱۹۵۷-۱۸۷۲) برگزار شده است.

Data Visualizations

In [40]:
```
import plotly.offline as ply

years = matches['date'].map(lambda y: int(str(y)[:4]))
friendly_match = matches['tournament'].map(lambda y: 1 if y=='Friendly' else 0 )
YEARS = list(np.unique(years))
COUNTRY = list(np.unique(matches['country']))


friendly_match = matches['tournament'].map(lambda y: 1 if y=='Friendly' else 0 )
goalAll = pd.DataFrame(matches['home_score']+matches['away_score'])
goalAll.insert(1, 'Date', years)
goalAll.insert(2, 'Country', matches['country'])
goalAll.insert(3, 'Tournament', matches['tournament'])
goalAll.insert(4, 'Friendly', friendly_match)
goalAll.columns = ['Goals','Year','Country', 'Tournament', 'Friendly']


sum_goals = []
for country in COUNTRY:
    sum_goals.append(goalAll[goalAll['Country']==country]['Goals'].sum())

goalPerCountry = [dict(
        colorscale=[
        [
          0,
          "rgb(255,255,255)"
        ],
        [
          0.1,
          "rgb(255,255,220)"
        ],
        [
          0.2,
          "rgb(255,255,200)"
        ],
        [
          0.3,
          "rgb(255,255,170)"
        ],
        [
          0.4,
          "rgb(255,255,120)"
        ],
        [
```

```
        [
          0.5,
          "rgb(255,255,0)"
        ],
        [
          0.6,
          "rgb(200,255,0)"
        ],
        [
          0.7,
          "rgb(150,255,0)"
        ],
        [
          0.8,
          "rgb(0,255,0)"
        ],
        [
          0.9,
          "rgb(0,200,0)"
        ],
        [
          1,
          "rgb(0,129,0)"
        ]
      ],
        type = 'choropleth',
        locations = COUNTRY,
        z = sum_goals,
        locationmode = 'country names',
        text = COUNTRY,
        marker = dict(
            line = dict(color = 'rgb(0,0,0)', width = 1)),
            colorbar = dict(autotick = True, tickprefix = '',
            title = '# Number of goals for 45 years \n')
            )
       ]

layout = dict(
    title = 'تعداد گل زده شده توسط هر کشور',
    geo = dict(
        showframe = False,
        showocean = False,

        projection = dict(
        type = 'orthographic',
            rotation = dict(
                    lon = 60,
                    lat = 10),
        ),
        lonaxis =  dict(
                showgrid = True,
                gridcolor = 'rgb(102, 102, 102)'
            ),
        lataxis = dict(
                showgrid = True,
                gridcolor = 'rgb(102, 102, 102)'
                )
            ),
        )

figure = dict(data=goalPerCountry, layout=layout)
ply.iplot(figure, validate=False, filename='worldmap', image='jpeg')
```



In [ ]: