In [1]:

```python
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/shot-data-fifa-u-17-world-cup-2023/fifa_wc_u_17_xg.csv
```

In [2]:

```python
import plotly
import plotly.express as px
import plotly.figure_factory as ff
import plotly.graph_objects as go
```

In [3]:

```
df=pd.read_csv('/kaggle/input/shot-data-fifa-u-17-world-cup-2023/fifa_wc_u_17_xg.csv')
df
```

Out[3]:

	Unnamed: 0	md	team_name	player_name	minute	player_pos	x	y	distance_to_r	distance_to_l	...	inside_pen_box	insi
0	0	1	ecu	Michael Bermúdez	2	mf	26.7	39.00	26.868011	27.164131	...	0	0
1	1	1	idn	Jehan Pahlevi	9	fw	9.9	61.00	26.888845	19.672570	...	1	0
2	2	1	idn	Muhammad Kafiatur Rizky	9	mf	18.4	50.00	23.120554	19.353553	...	0	0
3	3	1	ecu	Elkin Ruiz	13	df	17.9	39.40	18.220044	18.481612	...	1	0
4	4	1	idn	Muhammad Kafiatur Rizky	17	mf	31.0	23.20	33.538634	37.331488	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1610	1610	0	ger	Paris Brunner	90	fw	2.4	28.08	8.275651	16.099888	...	1	0
1611	1611	0	ger	Max Moerstedt	90	fw	9.6	32.96	10.069836	14.630161	...	1	0
1612	1612	0	ger	David Odogu	91	df	22.1	43.36	23.293338	22.109265	...	0	0
1613	1613	0	fra	Nhoa Sangui	92	df	18.3	36.16	18.300699	19.908682	...	0	0
1614	1614	0	fra	Tidiane Diallo	98	fw	8.1	25.28	13.436086	20.397265	...	1	0

1615 rows × 26 columns

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1615 entries, 0 to 1614
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         1615 non-null   int64  
 1   md                 1615 non-null   int64  
 2   team_name          1615 non-null   object 
 3   player_name        1615 non-null   object 
 4   minute             1615 non-null   int64  
 5   player_pos         1615 non-null   object 
 6   x                  1615 non-null   float64
 7   y                  1615 non-null   float64
 8   distance_to_r      1615 non-null   float64
 9   distance_to_l      1615 non-null   float64
 10  distance           1615 non-null   int64  
 11  angle              1615 non-null   float64
 12  shot_technique     1615 non-null   object 
 13  player_in_the_way  1615 non-null   int64  
 14  pass_type          1615 non-null   object 
 15  situation          1615 non-null   object 
 16  inside_pen_box     1615 non-null   int64  
 17  inside_6_box       1615 non-null   int64  
 18  header             1615 non-null   int64  
 19  weaker_foot        1615 non-null   int64  
 20  strong_foot        1615 non-null   int64  
 21  other_part         1615 non-null   int64  
 22  big_chances        1615 non-null   int64  
 23  on_target          1615 non-null   int64  
 24  is_goal            1615 non-null   int64  
 25  xg                 1615 non-null   float64
dtypes: float64(6), int64(14), object(6)
memory usage: 328.2+ KB
```

In [5]:

```
df = df.drop('Unnamed: 0', axis=1) #Drop built-in index row
df
```

Out[5]:

	md	team_name	player_name	minute	player_pos	x	y	distance_to_r	distance_to_l	distance	...	inside_pen_box	inside
0	1	ecu	Michael Bermúdez	2	mf	26.7	39.00	26.868011	27.164131	27	...	0	0
1	1	idn	Jehan Pahlevi	9	fw	9.9	61.00	26.888845	19.672570	20	...	1	0
2	1	idn	Muhammad Kafiatur Rizky	9	mf	18.4	50.00	23.120554	19.353553	19	...	0	0
3	1	ecu	Elkin Ruiz	13	df	17.9	39.40	18.220044	18.481612	18	...	1	0
4	1	idn	Muhammad Kafiatur Rizky	17	mf	31.0	23.20	33.538634	37.331488	34	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1610	0	ger	Paris Brunner	90	fw	2.4	28.08	8.275651	16.099888	8	...	1	0
1611	0	ger	Max Moerstedt	90	fw	9.6	32.96	10.069836	14.630161	10	...	1	0
1612	0	ger	David Odogu	91	df	22.1	43.36	23.293338	22.109265	22	...	0	0
1613	0	fra	Nhoa Sangui	92	df	18.3	36.16	18.300699	19.908682	18	...	0	0
1614	0	fra	Tidiane Diallo	98	fw	8.1	25.28	13.436086	20.397265	13	...	1	0

1615 rows × 25 columns

In [6]:

```
df.nunique()
```

Out[6]:

```
md                     8
team_name             24
player_name          339
minute               104
player_pos             3
x                    331
y                    608
distance_to_r       1567
distance_to_l       1566
distance              50
angle               1577
shot_technique         5
player_in_the_way      9
pass_type             12
situation              7
inside_pen_box         2
inside_6_box           2
header                 2
weaker_foot            2
strong_foot            2
other_part             2
big_chances            2
on_target              2
is_goal                2
xg                  1450
dtype: int64
```

Data Columns

md - match day

0 = final; 1 = group stages day 1; 2 = group stages day 2; 3 = group stages day 3; 4 = semi final; 8 = quarter final; 16 = round of 16; 33 = third place play off

team_name

arg = argentina; bra = brazil; bur = burkina faso; can = canada; ecu = ecuador; eng = england; fra = france; ger = germany; idn = indonesia; irn = iran; jpn = japan; kor = south korea; mal = mali; mex = mexico; mor = morocco; ncd = new caledonia; nz = new zealand; pan = panama; pol = poland; sen = senegal; spa = spain; usa = usa; uzb = uzbekistan; vnz = venezuela

minute

the minute the goal was scored

player_pos

mf = midfielder; df = defender; fw = forward

x

x coordinate of player's position in (100; 100) field

y

y coordinate of player's position in (100;100) field

distance_to_l

player's distance to left goal post in yards

distance_to_r

player's distance to right goal post in yards

distance

player's distance to the centre of a goal

angle

angle of a shot to go into the goal

shot_technique

technique used for the shot

player_in_the_way

players in front of the goal during the shot

pass_type

pass; dribble = made drrible before shot; loose-ball = the ball came from opposite team; clearance = after clearence of opposite team; cross = high pass from flanck; lay-off = short pass, often using the first touch; free-kick = goal from free-kick; through-pass = low pass from back through opposite defence; low-cross = low pass from flanck; 0 = own goals; long-pass; penalty = goal from penalty

situation

fast-break = leaving behind opposite team's defence; open-play = normal game situation; corner-kick; free-kick; 0 = own goals; penalty; throw-in

inside_pen_box

1/0 whether the shot was from inside the penalty box

inside_6_box

1/0 whether the shot was from inside the 6 yard box

header

1/0 whether the shot was by head

weaker_foot

1/0 whether the shot was by weaker foot

strong_foot

1/0 whether the shot was by strong foot

other_part

1/0 whether the shot was by other part of the body

big_chances

1/0 whether the shot was big chance

on_target

1/0 whether the shot was on target

is_goal

1/0 whether the shot was a goal

xg

expected goal score

In [7]:

```
df[['md', 'team_name', 'player_name', 'minute', 'angle', 'shot_technique', 'player_in_the_way', 'pass_type', 'situation', 'big_chances', 'is_goal', 'xg']].loc[df['minute']>100]
```

Out[7]:

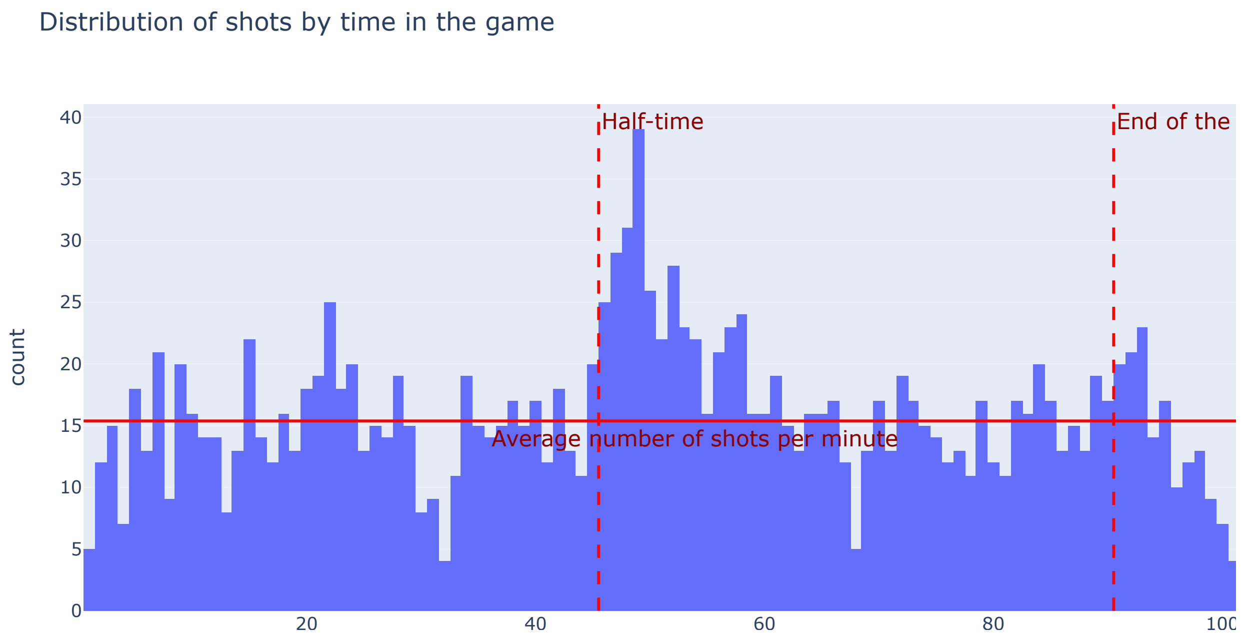
	md	team_name	player_name	minute	angle	shot_technique	player_in_the_way	pass_type	situation	big_chances	is_goal
20	1	ecu	Geremy De Jesus	103	22.878145	shot	4	lay-off	open-play	0	0
379	1	vnz	Alejandro Cichero	101	17.411218	shot	2	pass	open-play	0	0
421	2	ecu	Keny Arroyo	101	16.928909	shot	6	free-kick	free-kick	0	0
938	3	bra	Lorran Lucas	105	17.600849	shot	2	loose-ball	open-play	0	0
1215	4	mal	Sekou Kone	102	29.437848	shot	4	cross	corner-kick	0	0
1309	8	fra	Tidiam Gomis	101	15.446621	shot	3	pass	open-play	0	0
1470	16	eng	Josh Acheampong	102	18.230266	shot	4	loose-ball	open-play	0	0
1526	16	mor	Fouad Zahouani	101	30.863514	volley	5	cross	corner-kick	0	0

This table was used to check the metricks meanings

Data visualization

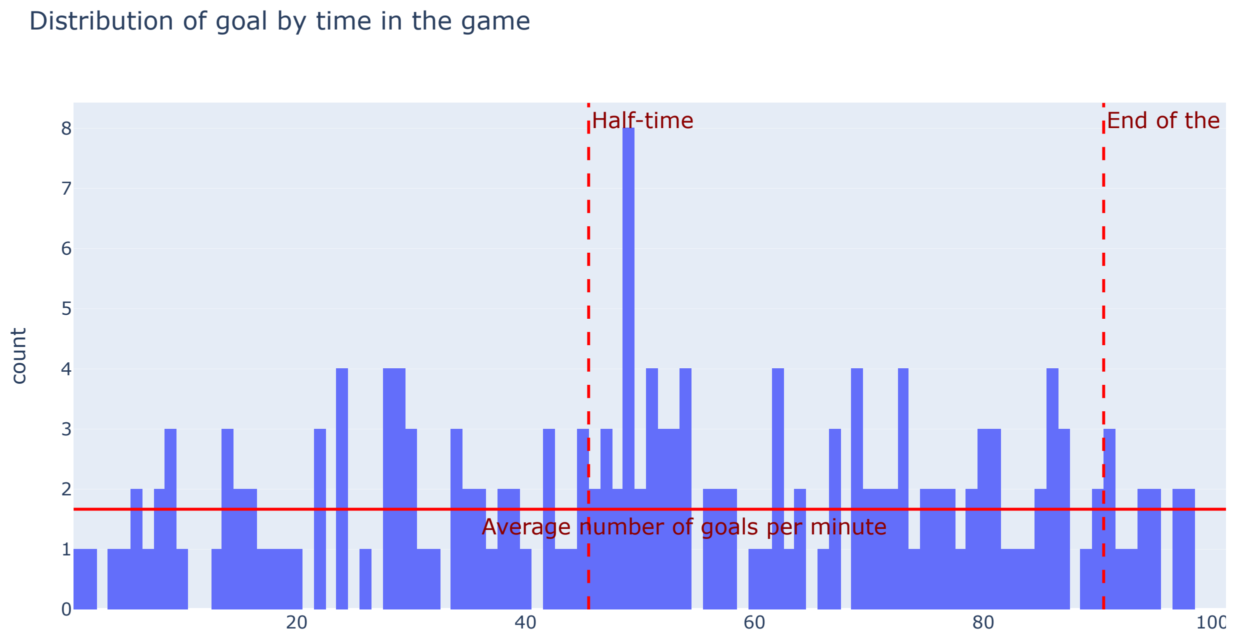
In [8]:

```python
fig = px.histogram(df, x="minute", title = "Distribution of shots by time in the game", nbins=105)
fig.add_hline(y = df['minute'].count()/105, annotation_text = "Average number of shots per minute",
              annotation_position="bottom", line_color="red", annotation=dict(font_size=15, font_color = 'darkred'))
fig.add_vline(x = 90.5, annotation_text = "End of the game", line_color="red", line_dash="dash", annotation=dict(font_size=15, font_color = 'darkred'))
fig.add_vline(x = 45.5, annotation_text = "Half-time", line_color="red", line_dash="dash", annotation=dict(font_size=15, font_color = 'darkred'))
fig.show()
```



In [9]:

```
fig = px.histogram(df.loc[df['is_goal'] == 1], x="minute", title = "Distribution of goal by time in the game", nbins=105)
fig.add_hline(y = df['minute'].loc[df['is_goal'] == 1].count()/105, annotation_text = "Average number of goals per minute",
              annotation_position="bottom", line_color="red", annotation=dict(font_size=15, font_color = 'darkred'))
fig.add_vline(x = 90.5, annotation_text = "End of the game", line_color="red", line_dash="dash", annotation=dict(font_size=15, font_color = 'darkred'))
fig.add_vline(x = 45.5, annotation_text = "Half-time", line_color="red", line_dash="dash", annotation=dict(font_size=15, font_color = 'darkred'))
fig.show()
```

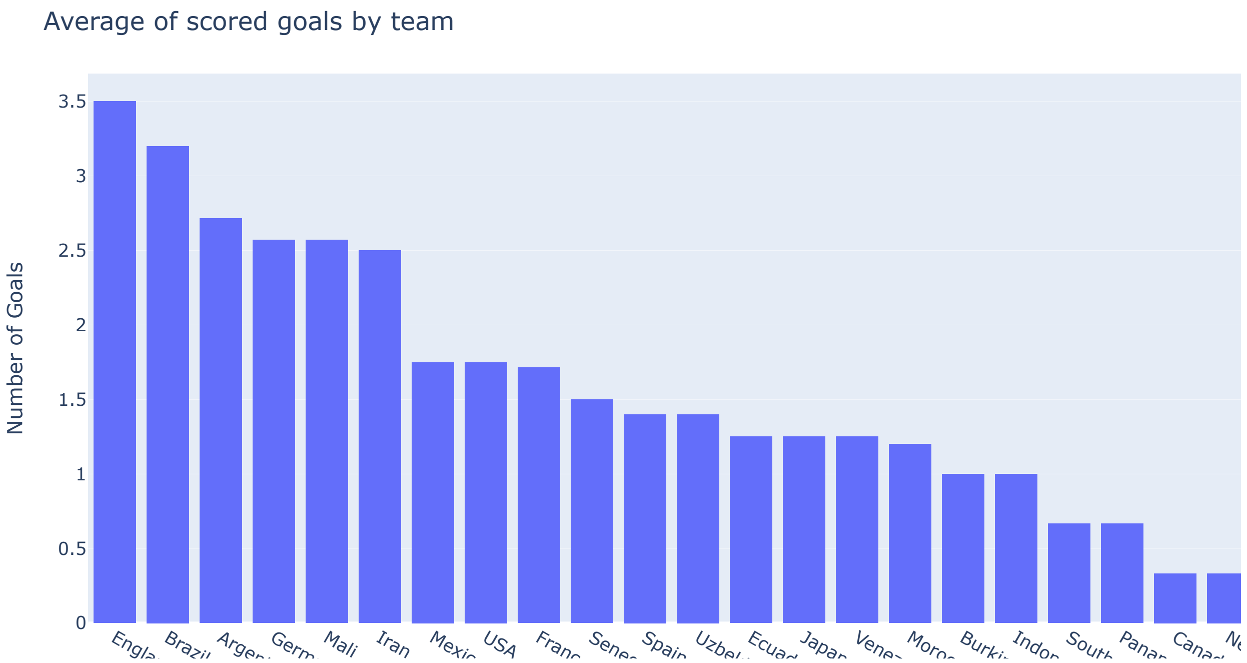


Here we can see the *Distribution of shots by time in the game* and the *Distribution of goals by time in the game*

The average shots per minute in this World Cup was near 15 shots/min. And the average goals per minute was near 1.7 goal/min.\ On the both graph we can see growth of shots and goal on 49 minute. In general there is growth of of shots per minute during period of 45-50 minutes. It can be explained by the fact that this minutes and only them are usually played twice in the match: during added minutes in the end of 1 time, and during the beggining of 2 time.\ At the end of the match growth of shots may be noticed due to team's attempt to change the score by the end of the match. Although there is no visible change in number of goals in last additional minutes, which means that it's highly unlikely to change score in last additional minutes.\ Drop of number of shots in last 10 minutes (95-105) can be a result that added time is usually not more than 5 minutes.

In [10]:

```
average_goals_in_match_by_country  = df.groupby('team_name').agg({'md': 'nunique','is_goal': 'sum'}).reset_index()
average_goals_in_match_by_country['average_n_of_goals'] = average_goals_in_match_by_country['is_goal']/average_goals_in_match_by_country['md']
fig = px.histogram(average_goals_in_match_by_country, x="team_name",y = 'average_n_of_goals').update_xaxes(categoryorder='total descending')
fig.update_layout(
    title_text='Average of scored goals by team',
    xaxis_title_text='Team name',
    yaxis_title_text='Number of Goals',
    legend_visible = False
                )
fig.update_xaxes(labelalias = {'arg': 'Argentina', 'bra': 'Brazil', 'bur': 'Burkina Faso', 'can': 'Canada', 'ecu': 'Ecuador', 'eng': 'England',
                            'fra': 'France', 'ger': 'Germany', 'idn': 'Indonesia', 'irn': 'Iran', 'jpn': 'Japan', 'kor': 'South Korea', 'mal': 'Mali',
                            'mex': 'Mexico', 'mor': 'Morocco', 'ncd': 'New Caledonia', 'nz': 'New Zealand', 'pan': 'Panama', 'pol': 'Poland', 'sen': 'Senegal',
                            'spa': 'Spain', 'usa': 'USA', 'uzb': 'Uzbekistan', 'vnz': 'Venezuela'}
               )
fig.show()
```



In [11]:

```
fig = go.Figure(data=[go.Table(header=dict(values=[['Team'], ['Number of matches']]),
                 cells=dict(values=[average_goals_in_match_by_country['team_name'], average_goals_in_match_by_country['md']]))
                     ])
fig.update_layout(title = 'Table of number of matches per team', width = 500, height = 700)
fig.show()
```

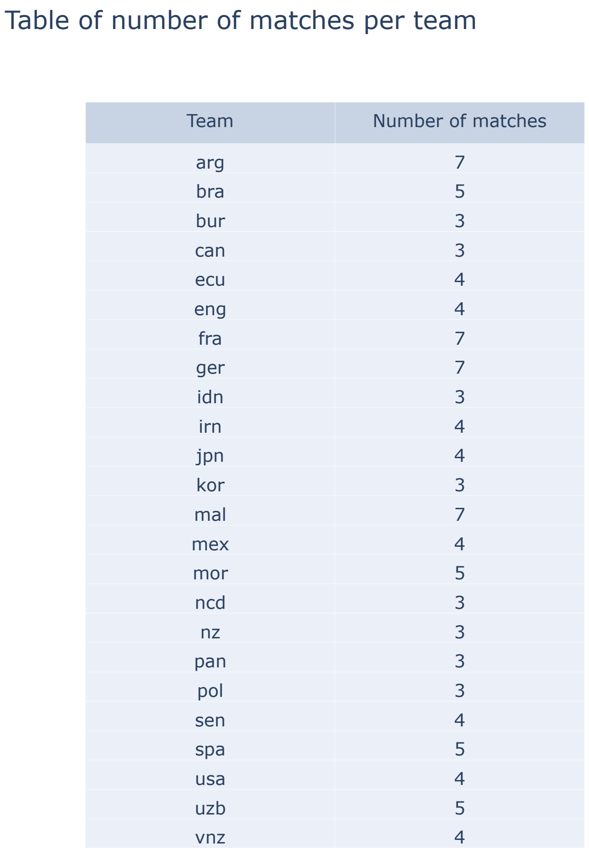
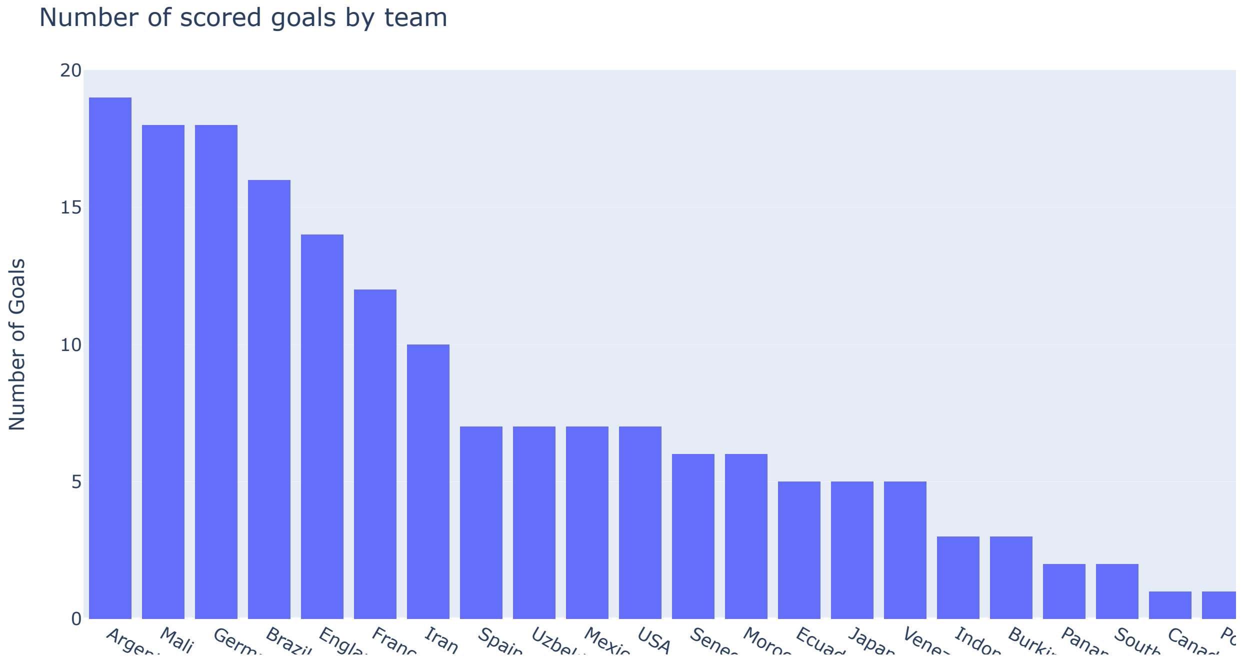


Table of number of matches per team

Team	Number of matches
arg	7
bra	5
bur	3
can	3
ecu	4
eng	4
fra	7
ger	7
idn	3
irn	4
jpn	4
kor	3
mal	7
mex	4
mor	5
ncd	3
nz	3
pan	3
pol	3
sen	4
spa	5
usa	4
uzb	5
vnz	4

In [12]:

```python
fig = px.histogram(df, x="team_name",y = 'is_goal').update_xaxes(categoryorder='total descending')
fig.update_layout(
    title_text='Number of scored goals by team',
    xaxis_title_text='Team name',
    yaxis_title_text='Number of Goals',
    legend_visible = False
                 )
fig.update_xaxes(labelalias = {'arg': 'Argentina', 'bra': 'Brazil', 'bur': 'Burkina Faso', 'can': 'Canada', 'ecu': 'Ecuador', 'eng': 'England',
                            'fra': 'France', 'ger': 'Germany', 'idn': 'Indonesia', 'irn': 'Iran', 'jpn': 'Japan', 'kor': 'South Korea', 'mal': 'Mali',
                            'mex': 'Mexico', 'mor': 'Morocco', 'ncd': 'New Caledonia', 'nz': 'New Zealand', 'pan': 'Panama', 'pol': 'Poland', 'sen': 'Senegal',
                            'spa': 'Spain', 'usa': 'USA', 'uzb': 'Uzbekistan', 'vnz': 'Venezuela'}
                )
fig.show()
```



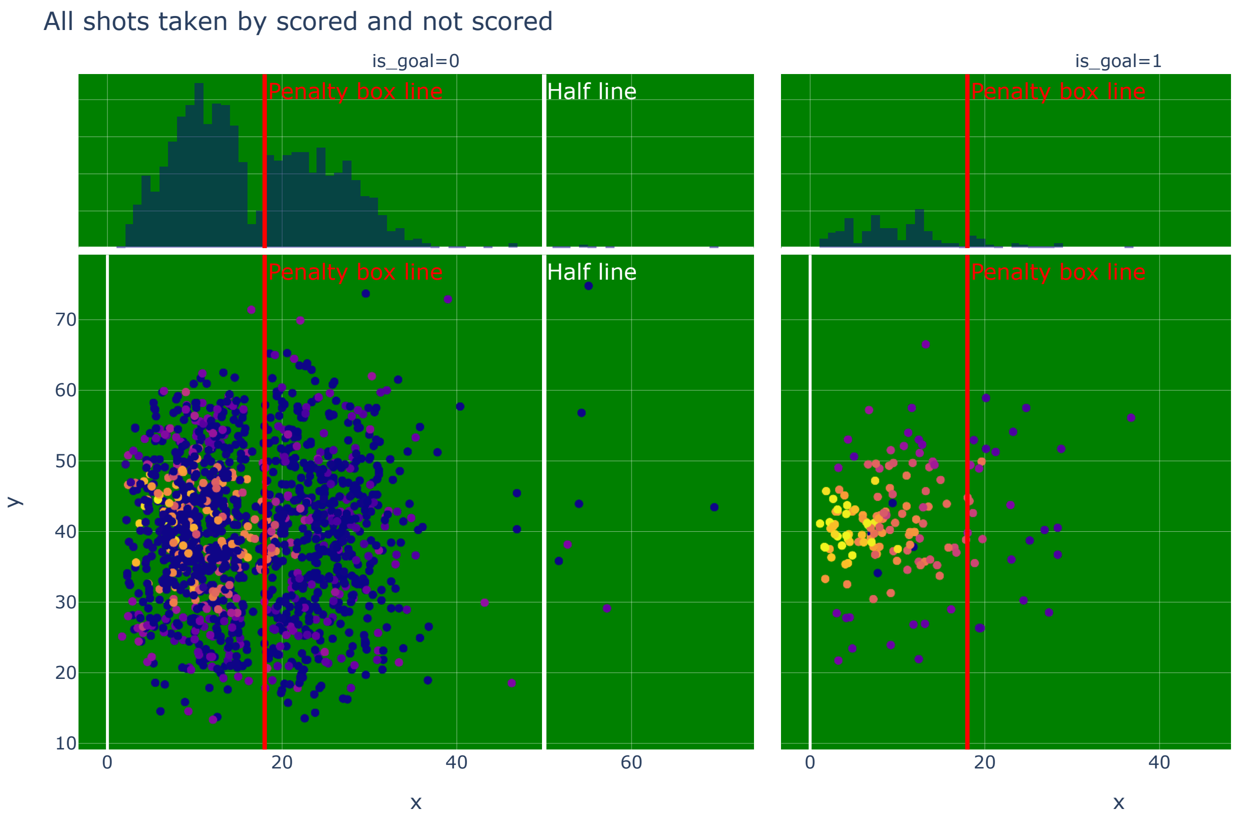
The *Average of scored goals by team* and *Number of Scored Goals by team* show certain insights

The Argentinian team has finished with the most goals during the U-17 World Cup, but its average of goals per match is lower than Brazil's and England's, even tho that England had only 4 games, Brazil had 5 games and Argentina had 7. This could indicate that in the group stage Brazil and England had easier opponents than Argentina.

In [13]:

```python
fig = px.scatter(df, x="x", y="y", color = 'xg', facet_col= 'is_goal', marginal_x = 'histogram', width=1120, height=600)
fig.update_layout(plot_bgcolor='green')
fig.update_layout(title_text='All shots taken by scored and not scored')
fig.add_vline(x = 50, annotation_text = "Half line", line_color="white", line_width =3, annotation=dict(font_size=15, font_color = 'white'))
fig.add_vline(x = 18, annotation_text = "Penalty box line", line_color="red", line_width =3, annotation=dict(font_size=15, font_color = 'red'))
fig.show()
```

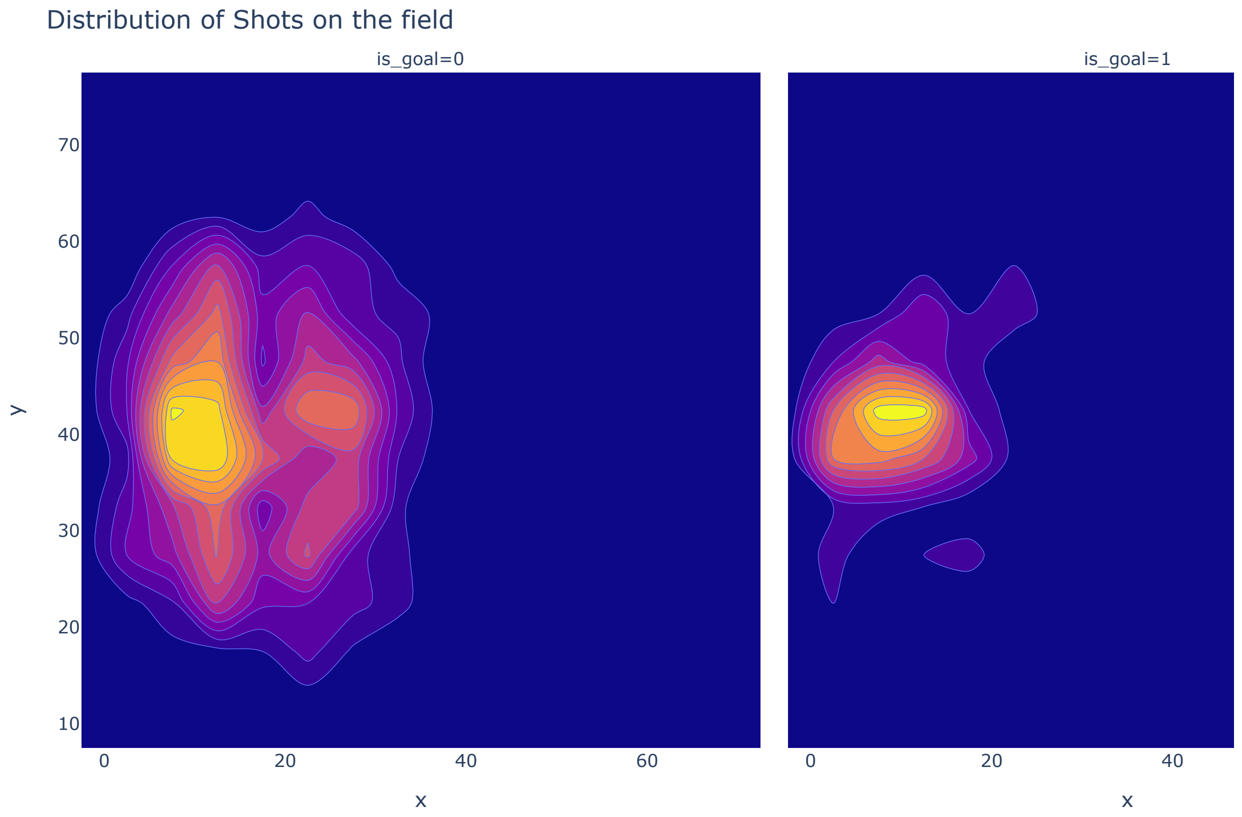
```
/opt/conda/lib/python3.10/site-packages/plotly/express/_core.py:2065: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
```



In [14]:

```
fig = px.density_contour(df, x="x", y="y", facet_col = "is_goal", width=1120, height=600)
fig.update_traces(contours_coloring = 'fill')
fig.update_layout(
    title_text='Distribution of Shots on the field'
)
fig.show()
```

```
/opt/conda/lib/python3.10/site-packages/plotly/express/_core.py:2065: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a
future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
```

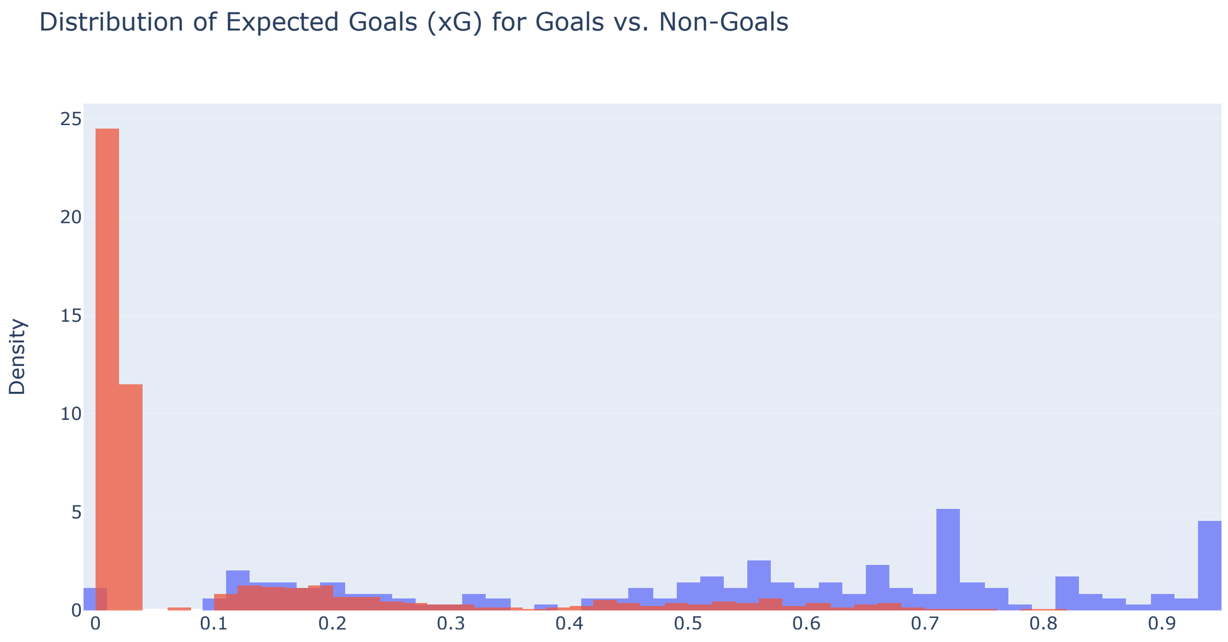


From *All shots taken by scored and not scored* we can see:

Most of succesful shots (goal) are taken behind of Penalty box line and none of them are in front of half line. For all the goals excpected goal (xg) is growing by closer it is to the goal.\ Although with shots we can see such a tendency as well, there are many shots close to goal with null xg. This could be explained by too mant defenders in front of the shot, hard pass for the shot, wrong movement of a ball.\ In distribution of shots we can see that right behind the penalty box line, the shots aren't taken for some reason. We can also see there that shots taken from there are right in front of the goal. This could be explaiend by small angles towards the goal in the beggining of the penalty box. And shots could be taken easier after few step. In the middle tho the angle is bigger and it is easier to shoot.

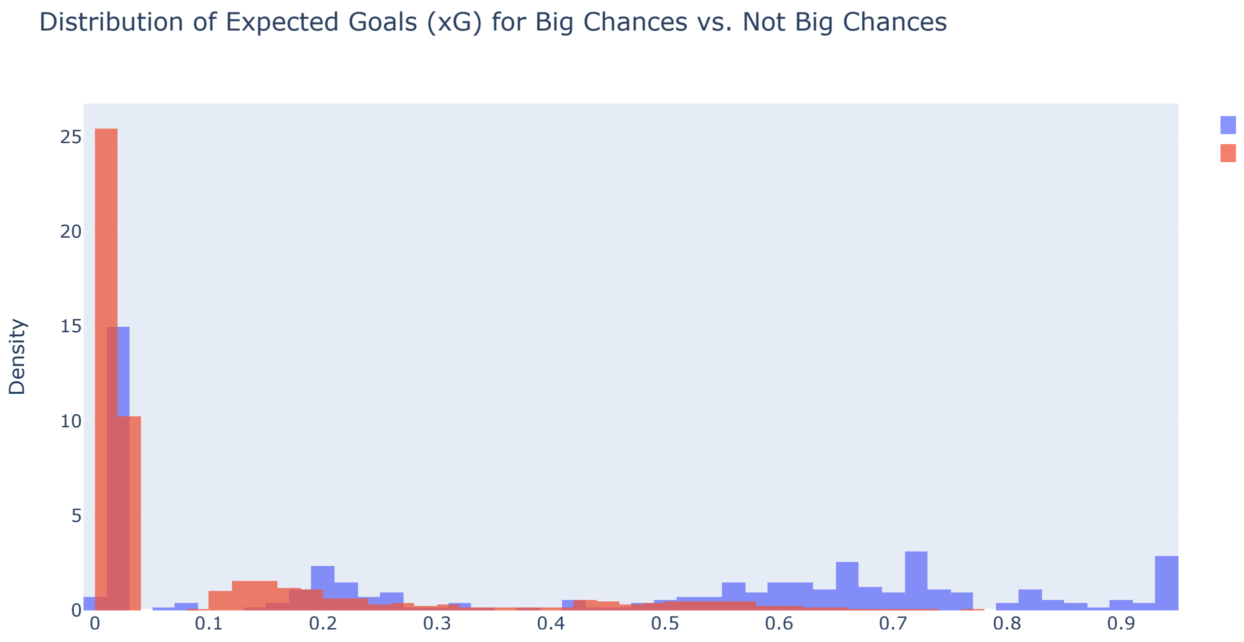
In [15]:

```
goals_xg = df[df['is_goal'] == 1]['xg']
non_goals_xg = df[df['is_goal'] == 0]['xg']

fig = go.Figure()
fig.add_trace(go.Histogram(x=goals_xg, nbinsx=50, name='Goals', histnorm='probability density', opacity=0.75))
fig.add_trace(go.Histogram(x=non_goals_xg, nbinsx=50, name='Non-Goals', histnorm='probability density', opacity=0.75))

fig.update_layout(
    title_text='Distribution of Expected Goals (xG) for Goals vs. Non-Goals',
    xaxis_title_text='Expected Goals (xG)',
    yaxis_title_text='Density',
    barmode='overlay'
)

fig.show()
```



In [16]:

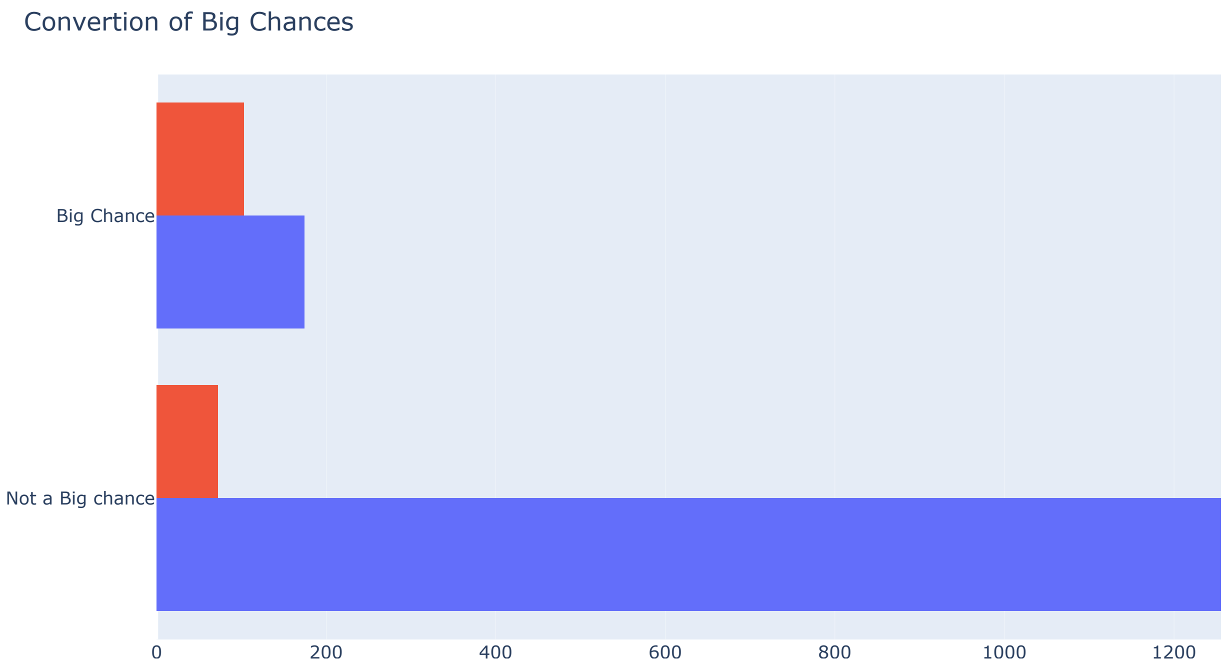
```
big_chances_xg = df[df['big_chances'] == 1]['xg']
not_big_chances_xg = df[df['big_chances'] == 0]['xg']

fig = go.Figure()
fig.add_trace(go.Histogram(x=big_chances_xg, nbinsx=50, name='Big Chance', histnorm='probability density', opacity=0.75))
fig.add_trace(go.Histogram(x=not_big_chances_xg, nbinsx=50, name='Not Big Chance', histnorm='probability density', opacity=0.75))

fig.update_layout(
    title_text='Distribution of Expected Goals (xG) for Big Chances vs. Not Big Chances',
    xaxis_title_text='Expected Goals (xG)',
    yaxis_title_text='Density',
    barmode='overlay'
)

fig.show()
```



In [17]:

```
fig = px.histogram(df, y="big_chances", color="is_goal", barmode = 'group')
fig.update_yaxes(labelalias = {0: 'Not a Big chance', 1: 'Big Chance'})
fig.update_layout(
    title_text='Convertion of Big Chances',
    xaxis_title_text='Number of shots',
    yaxis_title_text=''
)
fig.show()
```

```
/opt/conda/lib/python3.10/site-packages/plotly/express/_core.py:2065: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a
future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
```



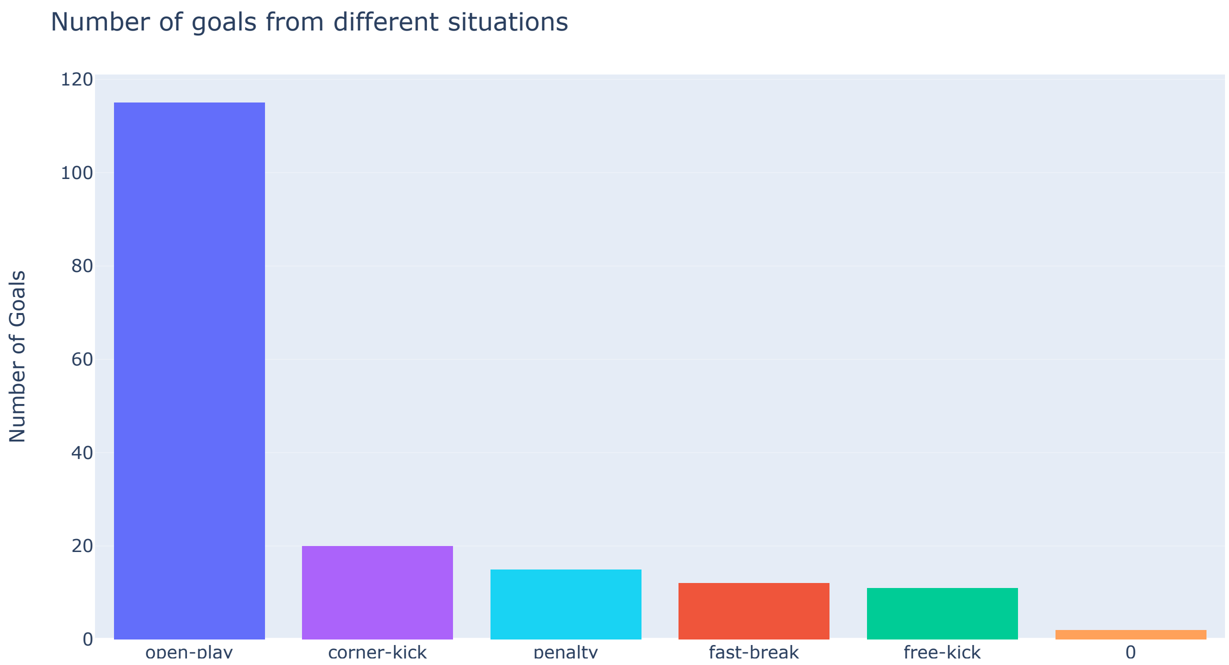
On *Distribution of Expected Goals (xG) for Goals vs. Non-Goals*,*Distribution of Expected Goals (xG) for Big Chances vs. Not Big Chances* and *Convertion of Big Chances* it is seen:

The XG for unsuccessful shots is unevenly distributed with right skew, which means that most of the unsuccessful shots are with low excpected goal metric. And successful goals have left skef opposite to unsuccessfull.\ The XG distribution for not a big chances is very similar to the distribution of unsuccessful shots with the right skew. But big chances distributions does not have the same left-skewed distribution as for successful goals. The explanation for that is that shot can be considered a big chance due to position of a player or the situation in the game but the shot itself is fault or out of the target, so the XG for the shot drops significantly.\ On the *Convertion of Big Chances* we can see that it is more likely to score the goal from big chance then form not a big chance, although mostly after big chance is still not scored.

```
In [18]:
```

```python
fig = px.histogram(df[df['is_goal']==1], x="situation", color = 'situation').update_xaxes(categoryorder='total descending')
fig.update_layout(
    title_text="Number of goals from different situations",
    xaxis_title_text='Situation',
    yaxis_title_text='Number of Goals'
)
fig.show()
```

```
/opt/conda/lib/python3.10/site-packages/plotly/express/_core.py:2065: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a
future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
```



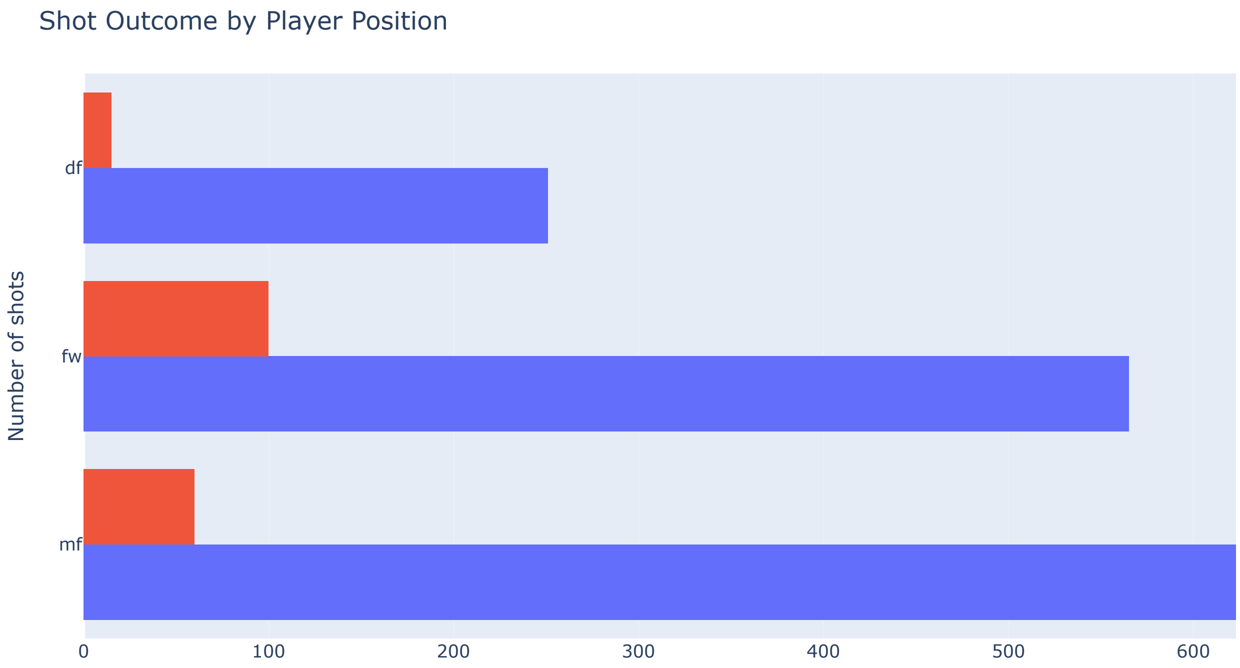
Number of goals from different situations

We can see from here that open-play situation happen approximately twice as much as all others.

In [19]:

```python
fig = px.histogram(df, y="player_pos", color = 'is_goal', barmode ="group")
fig.update_layout(
    title_text="Shot Outcome by Player Position",
    xaxis_title_text='Player Positions',
    yaxis_title_text='Number of shots'
)
fig.show()
```

```
/opt/conda/lib/python3.10/site-packages/plotly/express/_core.py:2065: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a
future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
```



Shot Outcome by Player Position shows

Midfielders shoot more, but have lesser successful outcomes of shots. And forwards hve better shot convertion into goal than midfielders with lesser shots. Can be explained by lesser amopunt of forwards on the field than midfielders and specific of their role. Defenders have laest number of shots and goals due to their position.

In [20]:

```
df.loc[(df['player_name'] == 'Agustín Ruberto') | (df['player_name'] == 'AgustÃn Ruberto')]
```

Out[20]:

	md	team_name	player_name	minute	player_pos	x	y	distance_to_r	distance_to_l	distance	...	inside_pen_box	inside_
65	1	arg	Agustín Ruberto	20	fw	9.5	45.8	13.648809	9.669023	10	...	1	0
78	1	arg	Agustín Ruberto	56	fw	15.2	34.7	15.255491	17.819371	15	...	1	0
86	1	arg	Agustín Ruberto	88	fw	12.7	27.4	15.337862	20.900957	15	...	1	0
88	1	arg	Agustín Ruberto	92	fw	27.4	28.6	28.381684	31.431195	28	...	0	0
623	2	arg	Agustín Ruberto	98	fw	23.3	54.2	29.565690	25.434819	25	...	0	0
804	3	arg	Agustín Ruberto	12	fw	3.6	44.2	8.955445	3.605551	4	...	1	1
811	3	arg	Agustín Ruberto	33	fw	22.5	49.8	26.394886	23.235533	23	...	0	0
814	3	arg	Agustín Ruberto	34	fw	2.5	38.6	3.606938	5.950630	4	...	1	1
816	3	arg	Agustín Ruberto	36	fw	23.8	35.8	23.800840	25.173001	24	...	0	0
821	3	arg	Agustín Ruberto	44	fw	13.6	42.5	15.073487	13.682471	14	...	1	0
825	3	arg	Agustín Ruberto	46	fw	15.9	37.8	16.001562	17.066048	16	...	1	0
1405	16	arg	Agustín Ruberto	70	fw	12.0	40.0	12.649111	12.649111	13	...	1	0
1407	16	arg	Agustín Ruberto	79	fw	13.8	36.1	13.800362	15.901258	14	...	1	0

13 rows × 25 columns

In [21]:

```
df.loc[(df['team_name'] == 'arg') & (df['is_goal'] == 1)]
```

Out[21]:

	md	team_name	player_name	minute	player_pos	x	y	distance_to_r	distance_to_l	distance	...	inside_pen_box	inside
88	1	arg	Agustín Ruberto	92	fw	27.4	28.60	28.381684	31.431195	28	...	0	0
601	2	arg	Claudio Echeverri	5	mf	26.9	40.30	27.241512	27.153269	27	...	0	0
602	2	arg	Valentino Acuña	8	mf	7.8	45.00	11.909660	7.863841	8	...	1	0
623	2	arg	Agustín Ruberto	98	fw	23.3	54.20	29.565690	25.434819	25	...	0	0
812	3	arg	Tobías Palacio	34	mf	2.3	37.60	2.801785	6.800735	3	...	1	1
825	3	arg	Agustín Ruberto	46	fw	15.9	37.80	16.001562	17.066048	16	...	1	0
827	3	arg	Ian Subiabre	52	fw	13.0	41.20	14.001428	13.298120	13	...	1	0
842	3	arg	Santiago Lopez	86	fw	17.9	38.90	18.133395	18.612361	18	...	1	0
1164	4	arg	AgustÃn Ruberto	36	fw	5.2	43.20	8.881441	5.261179	5	...	1	1
1169	4	arg	AgustÃn Ruberto	49	fw	11.2	34.64	11.282269	14.596219	11	...	1	0
1184	4	arg	AgustÃn Ruberto	97	fw	9.3	31.36	10.393248	15.692661	10	...	1	0
1255	8	arg	Claudio Echeverri	28	mf	23.0	43.84	24.299498	23.000557	23	...	0	0
1261	8	arg	Claudio Echeverri	58	mf	4.4	53.12	17.676380	10.125927	10	...	1	0
1267	8	arg	Claudio Echeverri	71	mf	10.1	37.60	10.225947	11.957006	10	...	1	0
1393	16	arg	Santiago Lopez	15	df	2.9	39.70	4.701064	5.186521	5	...	1	1
1396	16	arg	Santiago Lopez	22	fw	12.6	51.20	19.743353	14.512064	15	...	1	0
1399	16	arg	Claudio Echeverri	32	mf	13.2	35.80	13.201515	15.539627	13	...	1	0
1405	16	arg	Agustín Ruberto	70	fw	12.0	40.00	12.649111	12.649111	13	...	1	0
1407	16	arg	Agustín Ruberto	79	fw	13.8	36.10	13.800362	15.901258	14	...	1	0

19 rows × 25 columns

In [22]:

```
df['player_name'][1164]
```

Out[22]:

```
'AgustÃ\xadn Ruberto'
```

In [23]:

```
df.loc[(df['player_name'] == 'Agustín Ruberto') | (df['player_name'] == 'AgustÃ\xadn Ruberto')]
```

Out[23]:

	md	team_name	player_name	minute	player_pos	x	y	distance_to_r	distance_to_l	distance	...	inside_pen_box	inside
65	1	arg	Agustín Ruberto	20	fw	9.5	45.80	13.648809	9.669023	10	...	1	0
78	1	arg	Agustín Ruberto	56	fw	15.2	34.70	15.255491	17.819371	15	...	1	0
86	1	arg	Agustín Ruberto	88	fw	12.7	27.40	15.337862	20.900957	15	...	1	0
88	1	arg	Agustín Ruberto	92	fw	27.4	28.60	28.381684	31.431195	28	...	0	0
623	2	arg	Agustín Ruberto	98	fw	23.3	54.20	29.565690	25.434819	25	...	0	0
804	3	arg	Agustín Ruberto	12	fw	3.6	44.20	8.955445	3.605551	4	...	1	1
811	3	arg	Agustín Ruberto	33	fw	22.5	49.80	26.394886	23.235533	23	...	0	0
814	3	arg	Agustín Ruberto	34	fw	2.5	38.60	3.606938	5.950630	4	...	1	1
816	3	arg	Agustín Ruberto	36	fw	23.8	35.80	23.800840	25.173001	24	...	0	0
821	3	arg	Agustín Ruberto	44	fw	13.6	42.50	15.073487	13.682471	14	...	1	0
825	3	arg	Agustín Ruberto	46	fw	15.9	37.80	16.001562	17.066048	16	...	1	0
1158	4	arg	AgustÃn Ruberto	10	fw	23.2	47.92	26.083067	23.528842	24	...	0	0
1164	4	arg	AgustÃn Ruberto	36	fw	5.2	43.20	8.881441	5.261179	5	...	1	1
1169	4	arg	AgustÃn Ruberto	49	fw	11.2	34.64	11.282269	14.596219	11	...	1	0
1184	4	arg	AgustÃn Ruberto	97	fw	9.3	31.36	10.393248	15.692661	10	...	1	0
1405	16	arg	Agustín Ruberto	70	fw	12.0	40.00	12.649111	12.649111	13	...	1	0
1407	16	arg	Agustín Ruberto	79	fw	13.8	36.10	13.800362	15.901258	14	...	1	0
1575	33	arg	AgustÃn Ruberto	66	fw	10.5	36.64	10.519487	12.822621	11	...	1	0
1588	33	arg	AgustÃn Ruberto	92	fw	15.0	19.52	22.284308	28.710110	22	...	1	0

19 rows × 25 columns

In [24]:

```
df = df.replace(['AgustÃ\xadn Ruberto'],'Agustín Ruberto')
df.loc[(df['player_name'] == 'Agustín Ruberto') | (df['player_name'] == 'AgustÃ\xadn Ruberto')]
```

Out[24]:

	md	team_name	player_name	minute	player_pos	x	y	distance_to_r	distance_to_l	distance	...	inside_pen_box	inside
65	1	arg	Agustín Ruberto	20	fw	9.5	45.80	13.648809	9.669023	10	...	1	0
78	1	arg	Agustín Ruberto	56	fw	15.2	34.70	15.255491	17.819371	15	...	1	0
86	1	arg	Agustín Ruberto	88	fw	12.7	27.40	15.337862	20.900957	15	...	1	0
88	1	arg	Agustín Ruberto	92	fw	27.4	28.60	28.381684	31.431195	28	...	0	0
623	2	arg	Agustín Ruberto	98	fw	23.3	54.20	29.565690	25.434819	25	...	0	0
804	3	arg	Agustín Ruberto	12	fw	3.6	44.20	8.955445	3.605551	4	...	1	1
811	3	arg	Agustín Ruberto	33	fw	22.5	49.80	26.394886	23.235533	23	...	0	0
814	3	arg	Agustín Ruberto	34	fw	2.5	38.60	3.606938	5.950630	4	...	1	1
816	3	arg	Agustín Ruberto	36	fw	23.8	35.80	23.800840	25.173001	24	...	0	0
821	3	arg	Agustín Ruberto	44	fw	13.6	42.50	15.073487	13.682471	14	...	1	0
825	3	arg	Agustín Ruberto	46	fw	15.9	37.80	16.001562	17.066048	16	...	1	0
1158	4	arg	Agustín Ruberto	10	fw	23.2	47.92	26.083067	23.528842	24	...	0	0
1164	4	arg	Agustín Ruberto	36	fw	5.2	43.20	8.881441	5.261179	5	...	1	1
1169	4	arg	Agustín Ruberto	49	fw	11.2	34.64	11.282269	14.596219	11	...	1	0
1184	4	arg	Agustín Ruberto	97	fw	9.3	31.36	10.393248	15.692661	10	...	1	0
1405	16	arg	Agustín Ruberto	70	fw	12.0	40.00	12.649111	12.649111	13	...	1	0
1407	16	arg	Agustín Ruberto	79	fw	13.8	36.10	13.800362	15.901258	14	...	1	0
1575	33	arg	Agustín Ruberto	66	fw	10.5	36.64	10.519487	12.822621	11	...	1	0
1588	33	arg	Agustín Ruberto	92	fw	15.0	19.52	22.284308	28.710110	22	...	1	0

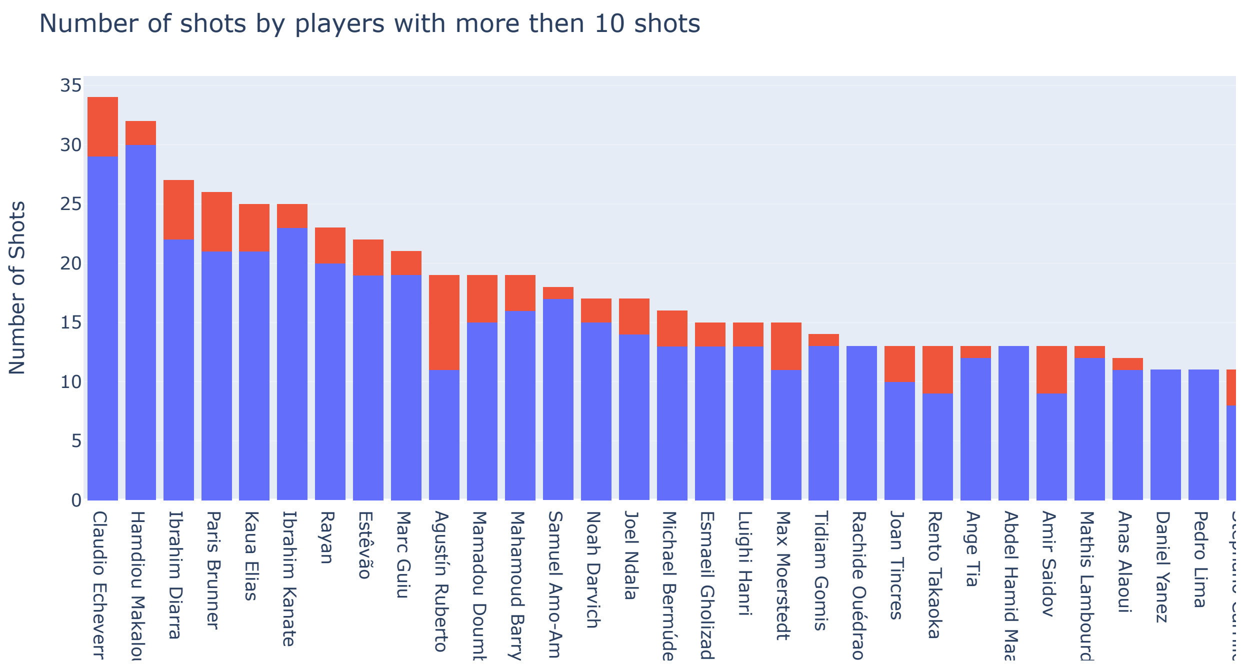
19 rows × 25 columns

From tables above we are fixing the problem of name difference. It was noticed after considering total goals of each players and official top scorer of the tournament with 8 goals ended up with 5 goals.

In [25]:

```
players_shots  = df.groupby('player_name').agg({'md': 'count','player_pos': 'first', 'is_goal': 'sum'}).reset_index()
fig = px.histogram(df.loc[df['player_name'].isin(players_shots['player_name'].loc[players_shots['md']>10])],
                   x="player_name", color = "is_goal").update_xaxes(categoryorder='total descending')
fig.update_layout(
    title_text="Number of shots by players with more then 10 shots",
    xaxis_title_text='Player',
    yaxis_title_text='Number of Shots'
)
fig.show()
```

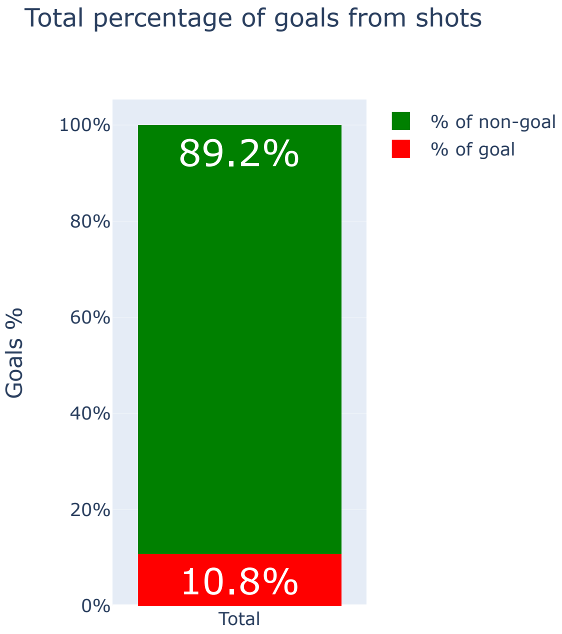
```
/opt/conda/lib/python3.10/site-packages/plotly/express/_core.py:2065: FutureWarning:

When grouping with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future version of pandas. Pass `(name,)` instead of `name` to silence this warning.
```



Number of shots by players with more then 10 shots

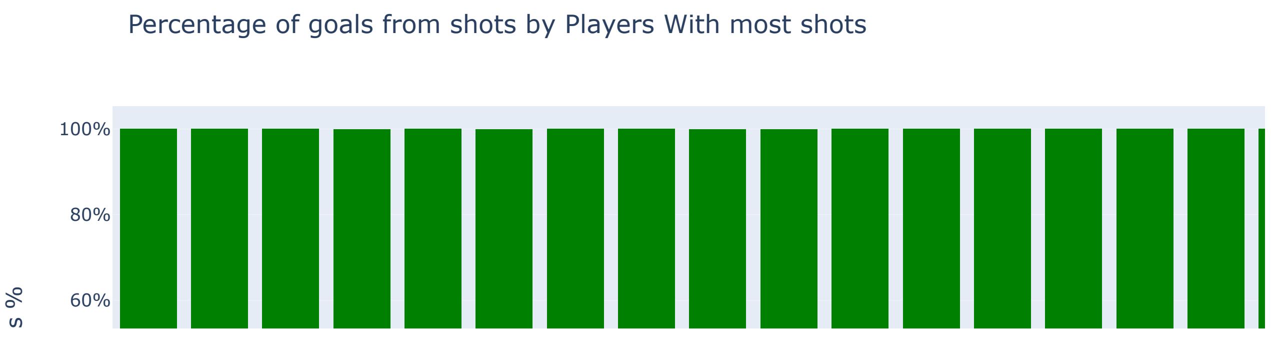
In [26]:

```
players_shots['not_goal'] = players_shots['md']-players_shots['is_goal']
percent_goal_total = (sum(players_shots['is_goal'])/sum(players_shots['md']))*100
percent_nongoal_total = (sum(players_shots['not_goal'])/sum(players_shots['md']))*100
fig = go.Figure()
fig.add_trace(go.Bar(
    y = [percent_goal_total],
    x = ['Total'],
    name="% of goal",
    text = str(round(percent_goal_total, 1)) + "%",
    textposition = 'inside',
    textfont = dict(
        size = 25
    ),
    marker=dict(
        color='red',
        line=dict(color='red', width=0.05)
    ),
    #orientation='h'
))
fig.add_trace(go.Bar(
    y = [percent_nongoal_total],
    x = ['Total'],
    name="% of non-goal",
    text = str(round(percent_nongoal_total, 1)) + "%",
    textposition = 'inside',
    textfont = dict(
        size = 25
    ),
    marker=dict(
        color='green',
        line=dict(color='green', width=0.05)
    ),
    #orientation='h'
))
fig.update_layout(
        yaxis=dict(
        title_text="Goals %",
        ticktext=["0%", "20%", "40%", "60%","80%","100%"],
        tickvals=[0, 20, 40, 60, 80, 100],
        tickmode="array",
        titlefont=dict(size=15),
    ),
    width = 400,
    #height = 300,
    title_text = 'Total percentage of goals from shots',
    barmode='stack')
fig.show()
```


Total percentage of goals from shots
Goals %
100%
80%
60%
40%
20%
0%
89.2%
10.8%
Total
🟩 % of non-goal
🟥 % of goal

In [27]:

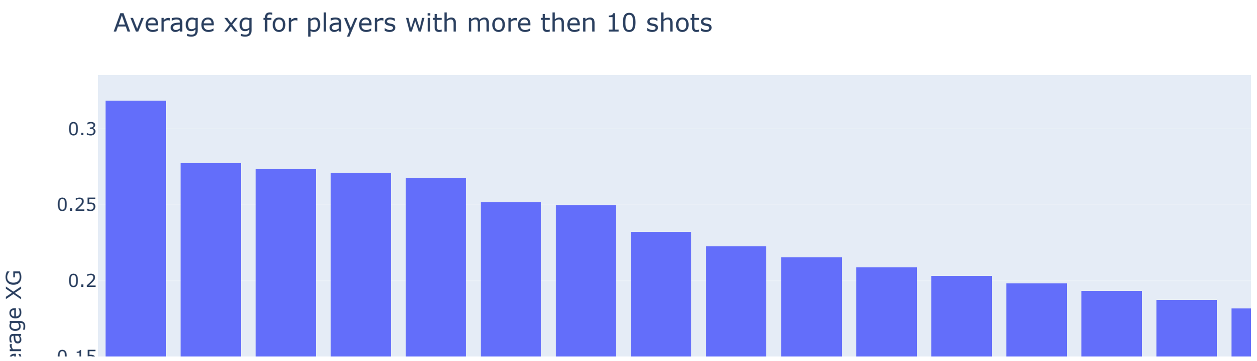
```
players_shots['% of goal'] = players_shots['is_goal']/players_shots['md']*100
players_shots['% of non-goal'] = players_shots['not_goal']/players_shots['md']*100
fig = go.Figure()
fig.add_trace(go.Bar(
    y=players_shots.loc[players_shots['md']>10]["% of goal"],
    x=players_shots.loc[players_shots['md']>10]['player_name'],
    name="% of goal",
    marker=dict(
        color='red',
        line=dict(color='red', width=0.05)
    )
))
fig.add_trace(go.Bar(
    y=players_shots.loc[players_shots['md']>10]["% of non-goal"],
    x=players_shots.loc[players_shots['md']>10]['player_name'],
    name="% of non-goal",
    marker=dict(
        color='green',
        line=dict(color='green', width=0.05)
    )
))
fig.update_layout(
        yaxis=dict(
        title_text="Goals %",
        ticktext=["0%", "20%", "40%", "60%","80%","100%"],
        tickvals=[0, 20, 40, 60, 80, 100],
        tickmode="array",
        titlefont=dict(size=15),
    ),
    xaxis_title_text='Players',
    title_text = 'Percentage of goals from shots by Players With most shots',
    barmode='stack')
fig.update_xaxes(categoryorder='max ascending')
fig.show()
```



Percentage of goals from shots by Players With most shots

In [28]:

```
fig = px.histogram(df.loc[df['player_name'].isin(players_shots['player_name'].loc[players_shots['md']>10])],
                   x="player_name", y = 'xg', histfunc = 'avg').update_xaxes(categoryorder='total descending')
fig.update_layout(
    title_text="Average xg for players with more then 10 shots",
    xaxis_title_text='Player',
    yaxis_title_text='Average XG'
)
fig.show()
```



Individual achievements

On the graphs above we can see that there three different leaders in all of the graphs:\ Claudio Echeverri is an argentinian player with 34 shots in 7 matches. He has made the most shots throught the whole tournament. Although his average xg for the shot is 0.13 and the expected number of goals per tournament 4.42. He has scored 5, overdoing expectancy. His convertion of a shot into a goal is 14.7%.\ Augustin Robert is an argentinian player with 19 shots in 7 matches. He has made the most goals in the tournament (8) and he has the biggest convertion of a shot into a goal (42.1%). His average xg for the shot is 0.26 and the expected number of goals per tournament is 4.94. He has overdone his expectancy 1.6 times.\ Rento Takaoka is japanese player with 13 shots in 4 matches. He had the biggest average excpected goal metric per shot(0.31), which gives him expectancy of 4.03 goals per tournament with his scored 4 goals, he fully done as excpected. His convertion of a shot into a goal is top-2 of a tournament - 30.8%.