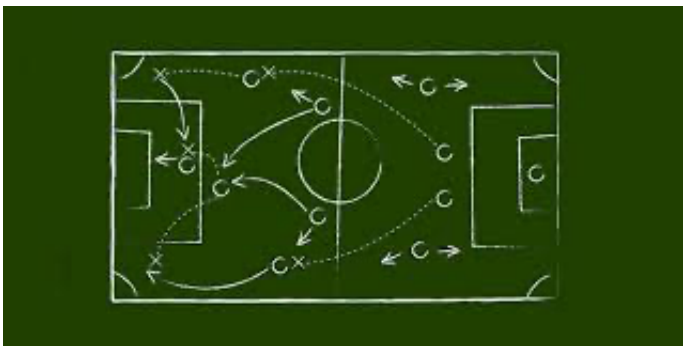
Importing the libraries

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

```
/Applications/anaconda3/lib/python3.11/site-packages/pandas/core/a
rrays/masked.py:60: UserWarning: Pandas requires version '1.3.6' o
r newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

Reading the Data



In [2]:
```
fifa=pd.read_csv('fifa21_male2.csv')
fifa.head()
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/26
07369135.py:1: DtypeWarning: Columns (78) have mixed types. Specif
y dtype option on import or set low_memory=False.
  fifa=pd.read_csv('fifa21_male2.csv')
```

Out[2]:

	ID	Name	Age	OVA	Nationality	Club	BOV	BP	Position	
0	2	G. Pasquale	33	69	Italy	Udinese	71	LWB	LM	https://cdn.sofifa.c
1	16	Luis García	37	71	Spain	KAS Eupen	70	CM	CM CAM CDM	https://cdn.sofifa.c
2	27	J. Cole	33	71	England	Coventry City	71	CAM	CAM RM RW LM	https://cdn.sofifa.c
3	36	D. Yorke	36	68	Trinidad & Tobago	Sunderland	70	ST	NaN	https://cdn.sofifa.c
4	41	Iniesta	36	81	Spain	Vissel Kobe	82	CAM	CM CAM	https://cdn.sofifa.c

5 rows × 107 columns

Checking out the Data Types

In [3]:
```
fifa.dtypes
```

Out[3]:
```
ID              int64
Name           object
Age             int64
OVA             int64
Nationality    object
                ...
CB             object
RCB            object
RB             object
GK             object
Gender         object
Length: 107, dtype: object
```

Identifying the null Values

In [4]:
```
fifa.isnull().sum()
```

Out[4]:
```
ID             0
Name           0
Age            0
OVA            0
Nationality    0
              ..
CB             0
RCB            0
RB             0
GK             0
Gender         0
Length: 107, dtype: int64
```

Cleaning the null values in the rows

In [5]:
```
fifa['Volleys'].replace({np.NaN:fifa['Volleys'].mean()},inplace=Tru
fifa['Curve'].replace({np.NaN:fifa['Curve'].mean()},inplace=True)
fifa['Agility'].replace({np.NaN:fifa['Agility'].mean()},inplace=Tru
fifa['Balance'].replace({np.NaN:fifa['Balance'].mean()},inplace=Tru
fifa['Jumping'].replace({np.NaN:fifa['Jumping'].mean()},inplace=Tru
fifa['Interceptions'].replace({np.NaN:fifa['Interceptions'].mean()}
fifa['Positioning'].replace({np.NaN:fifa['Positioning'].mean()},inp
fifa['Composure'].replace({np.NaN:fifa['Composure'].mean()},inplace
fifa['Sliding Tackle'].replace({np.NaN:fifa['Sliding Tackle'].mean(
```

...

Finding the total count of players available

In [6]:
```
fifa.Name.nunique()
```

Out[6]:
```
16176
```

What is the nationwise participation in the Game?

In [7]:
```
nat_cnt=fifa.groupby('Nationality').apply(lambda x:x['Name'].count(
nat_cnt.sort_values(by='Counts',ascending=False,inplace=True)
top_20_nat_cnt=nat_cnt[:20]
fig=px.bar(top_20_nat_cnt,x='Nationality',y='Counts',color='Counts'
fig.show()
```

...

Observations England has the highest number of players in FIFA 21 game. One of the major reasons in this regards is due to the EA franchise, which has predominatingly most user base in UK. Also in FIFA, English League has the most number of teams- generating the most number of players

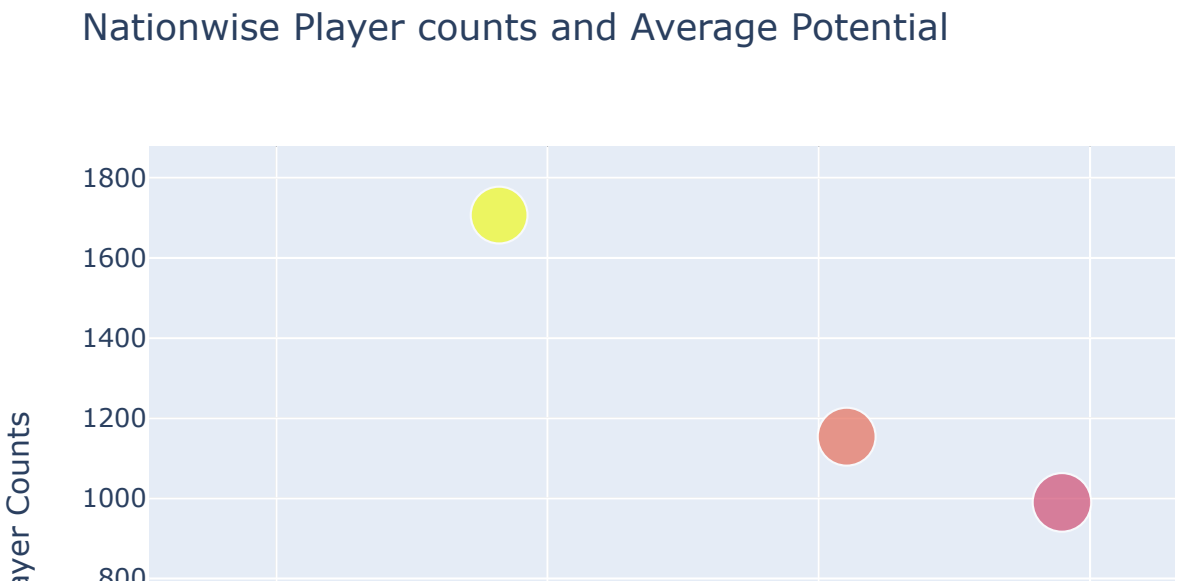
Which nation has the most number of over performing players?

Note We have only selected countries that have atleast 150 players in record. This is done in order to avoid any bias

In [8]:
```
cnt_best_avg=fifa.groupby('Nationality').apply(lambda x:np.average(
cnt_best_cnt=fifa.groupby('Nationality').apply(lambda x:x['OVA'].co
snt_best_avg_cnt=pd.merge(cnt_best_avg,cnt_best_cnt,how='inner',lef
sel_best_avg_cnt=snt_best_avg_cnt[snt_best_avg_cnt['Player Counts']
sel_best_avg_cnt.sort_values(by=['Overall Ratings','Player Counts']
px.scatter(sel_best_avg_cnt,x='Overall Ratings',y='Player Counts',c
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/20
36825340.py:1: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/20
```

```
36825340.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

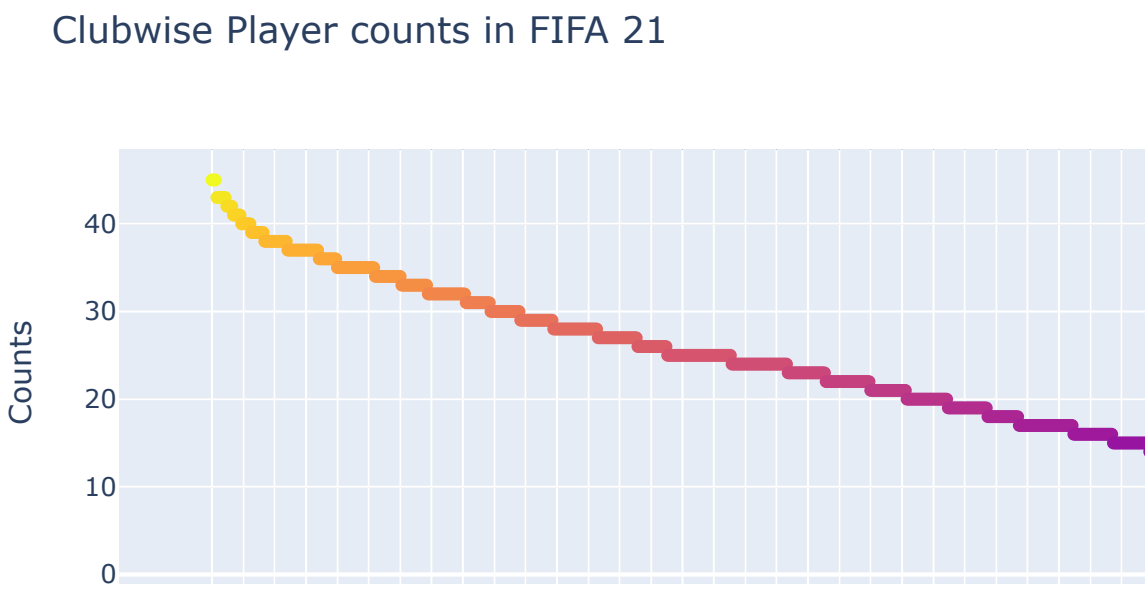


Observations England and Brazil are the teams that deserve a mention in this aspect. England since it has produced 1707 players, and still is having an average of 63.64, while Brazil has the highest average Ratings among the players

What has been the team wise player counts?

In [9]:
```
clb_cnt=fifa.groupby('Club').apply(lambda x:x['Name'].count()).rese
clb_cnt.sort_values(by='Counts',ascending=False,inplace=True)
fig=px.scatter(clb_cnt,x='Club',y='Counts',color='Counts',title='Cl
fig.show()
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/29
85839874.py:1: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```



Observation Chealsea and Manchester United have the most number of Players in record as a Club in FIFA 21- 45 players information for both the clubs. Just a general observation, the average count of players for the English Premier League is more tahn any other leage. This shows the prioritization of English football by FIFA

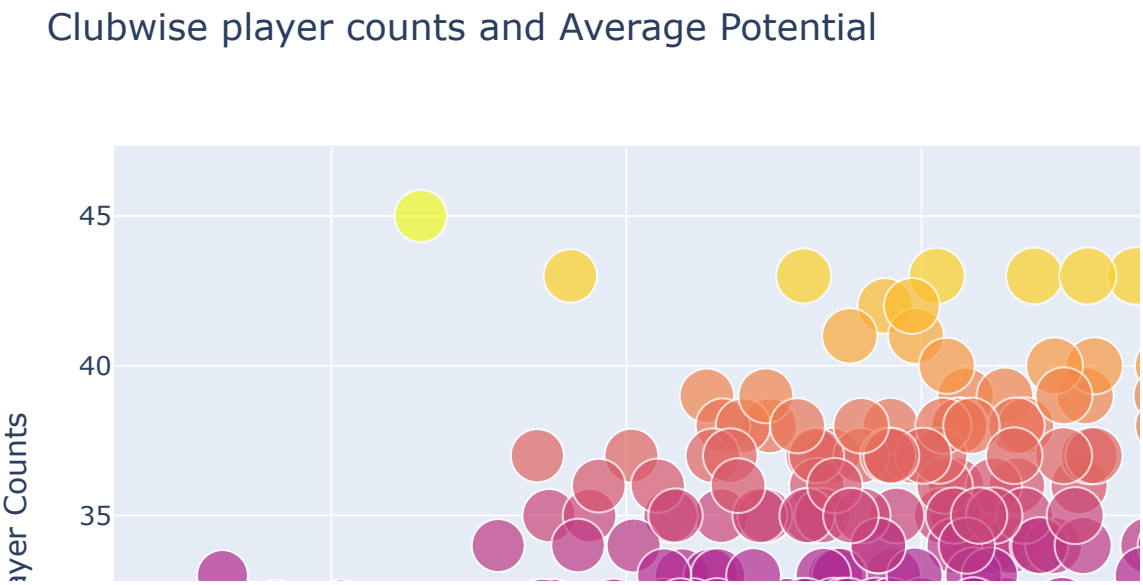
Which Team has the most number of Over Performing Players?

Note We have only selected clubs that have atleast 25 players in record. This is done in order to avoid any bias

In [10]:
```
cnt_best_avg=fifa.groupby('Club').apply(lambda x:np.average(x['OVA'
cnt_best_cnt=fifa.groupby('Club').apply(lambda x:x['OVA'].count()).
snt_best_avg_cnt=pd.merge(cnt_best_avg,cnt_best_cnt,how='inner',lef
sel_best_avg_cnt=snt_best_avg_cnt[snt_best_avg_cnt['Player Counts']
sel_best_avg_cnt.sort_values(by=['Overall Ratings','Player Counts']
px.scatter(sel_best_avg_cnt,x='Overall Ratings',y='Player Counts',c
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/28
13964405.py:1: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/28
13964405.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
```

```
ouping columns after groupby to silence this warning.
```



Observation As per the above chart, two teams deserve a special mention in this regard. Firstly it is Bayern Munich- The team which has the highest average rating among all the teams (81.26) from a set of 26 players. Another team is Manchester United- which has the highest average amon the teams with 45 layers. They have an average of 75.866 on the 45 players

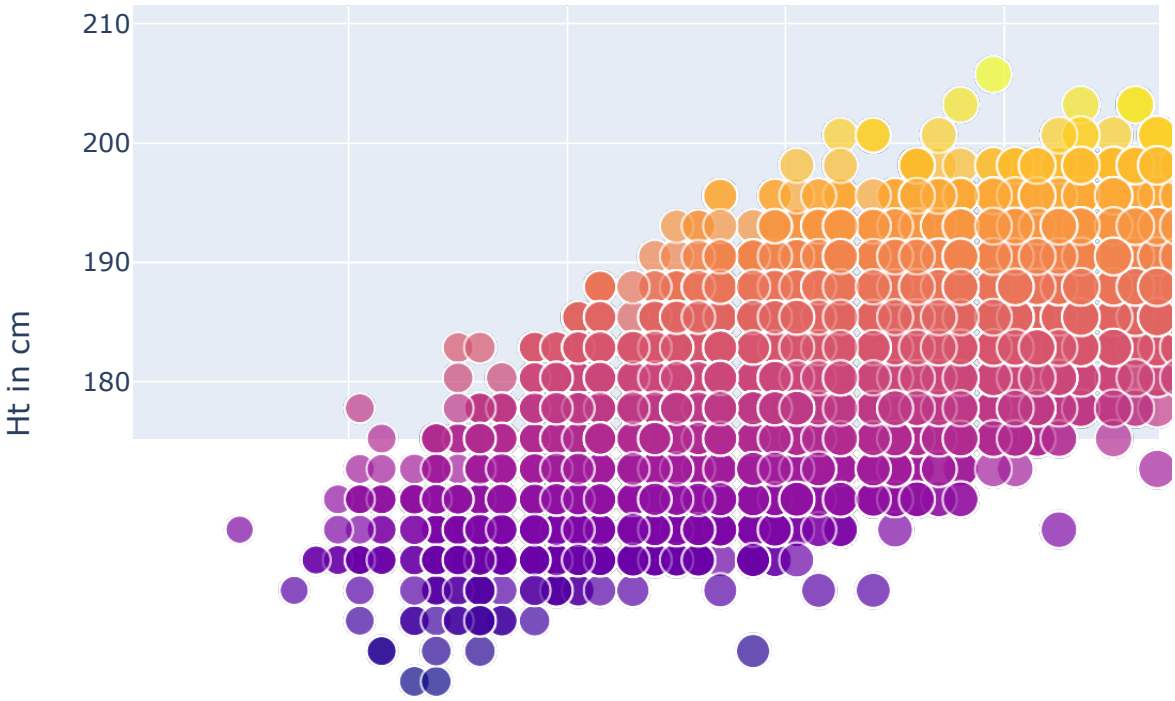
What is the Height vs Weight variation of the Footballers?

In [11]:
```
props=fifa[['Name','Nationality','Club','Height','Weight']]
props['Ht in ft']=pd.to_numeric(props['Height'].str[0])
props['Ht in in']=pd.to_numeric(props['Height'].str.split("\'").str
props['Ht in cm']=(props['Ht in ft']*12+props['Ht in in'])*2.54
props['Weight in lb']=pd.to_numeric(props['Weight'].str.strip('lbs'
fig=px.scatter(props,x='Weight in lb',y='Ht in cm',color='Ht in cm'
```

```
fig.show()
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/31
88248403.py:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/31
88248403.py:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/31
88248403.py:4: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/31
88248403.py:5: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

Height vs Weight Variation of the players in FIFA 21

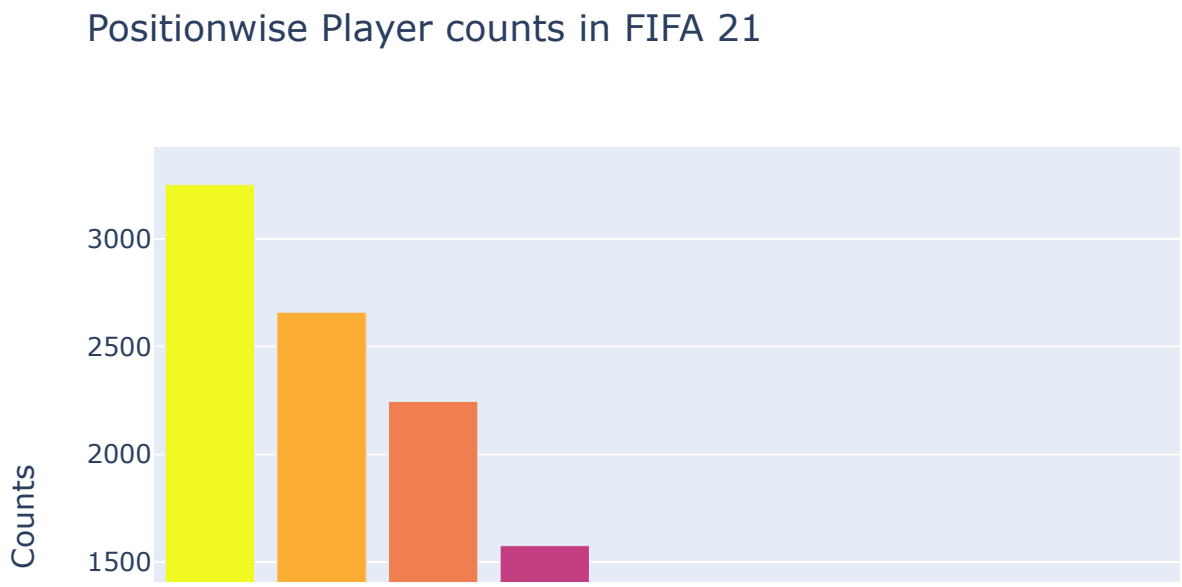


Observation Generally for a healthy football player, the height and weight are in a proportion. Else he/she willto be too weak/heavy- and not have the peak fitness form. This is seen from the above scatter plot.

Count of players position wise

In [12]:
```
pos_cnt=fifa.groupby('BP').apply(lambda x:x['Name'].count()).reset_
pos_cnt.sort_values(by='Counts',ascending=False,inplace=True)
top_20_pos_cnt=pos_cnt[:20]
fig=px.bar(top_20_pos_cnt,x='BP',y='Counts',color='Counts',title='P
fig.show()
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/37
60697610.py:1: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

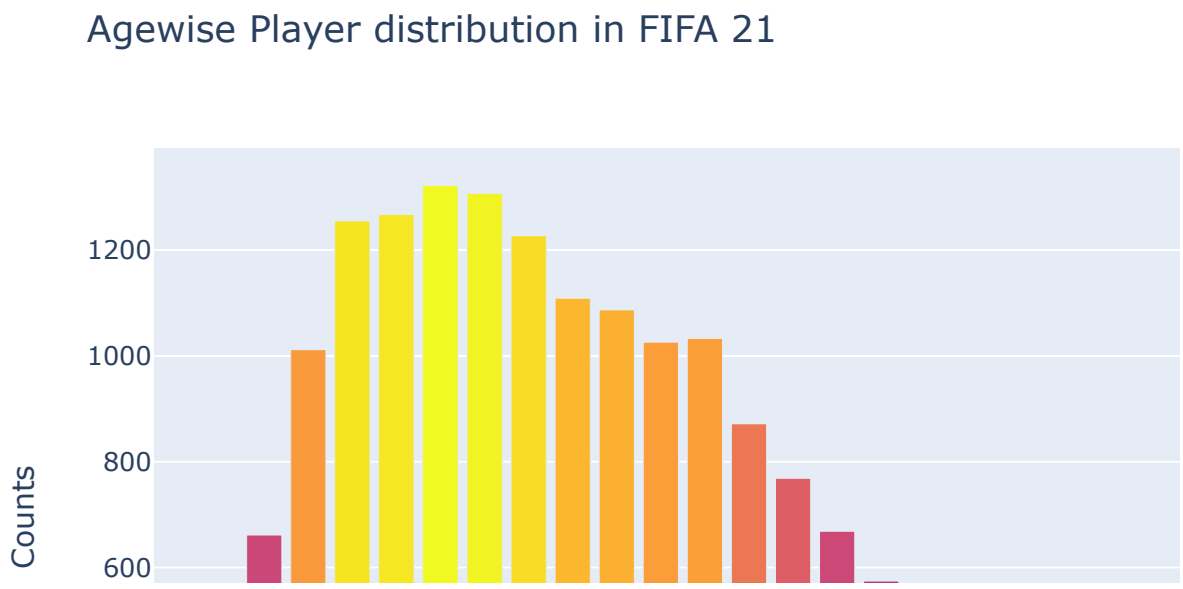


Observation In FIFA 21, the most number of player population is for the Center Back Position, which is followed by Striker and The Central attacking midfielder positions.

What is the age Distribution of the players?

In [13]:
```
age_cnt=fifa.groupby('Age').apply(lambda x:x['Name'].count()).reset
fig=px.bar(age_cnt,x='Age',y='Counts',color='Counts',title='Agewise
fig.show()
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/38
0229794.py:1: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```



Observation The chart looks like a normal curve which is left skewed. On an average 20-24 is the average age for most of the footballers in FIFA21

What is the distibution of Market Value and Wage for the players?

In [14]:
```
cost_prop=fifa[['Name','Club','Nationality','Wage','Value','BP']]
cost_prop['MultW']=np.where(cost_prop.Wage.str[-1]=='K',1000,1)
cost_prop['Wage']=cost_prop.Wage.str.strip('K')
cost_prop['Wage']=cost_prop.Wage.str.strip('€')
cost_prop['Wage']=pd.to_numeric(cost_prop['Wage'])
cost_prop['Wage in €']=cost_prop['Wage']*cost_prop['MultW']
cost_prop['MultV']=np.where(cost_prop.Value.str[-1]=='K',1000,np.wh
cost_prop['Value']=cost_prop.Value.str.strip('€')
cost_prop['Value']=cost_prop.Value.str.strip('K')
cost_prop['Value']=cost_prop.Value.str.strip('M')
cost_prop['Value']=pd.to_numeric(cost_prop['Value'])
cost_prop['Value in €']=cost_prop['Value']*cost_prop['MultV']
fig=px.scatter(cost_prop,x='Value in €',y='Wage in €',color='Value
fig.show()
```

...

Observation Lionel Messi tops the chart as the player with the highest wage. While Kylian Mbappe tops in terms of Market value. A general observation, we see the forward players usually having a higher Value and Wages. Is there a bias? Lets deep dive later!

How does the game attributes vary with position?

In [17]:
```
%matplotlib inline
import pandas as pd
import seaborn as sns


pos_head=fifa.groupby('BP').apply(lambda x:np.average(x['Heading Ac
pos_sp=fifa.groupby('BP').apply(lambda x:np.average(x['Short Passin
pos_d=fifa.groupby('BP').apply(lambda x:np.average(x['Dribbling']))
pos_c=fifa.groupby('BP').apply(lambda x:np.average(x['Curve'])).res
pos_fk=fifa.groupby('BP').apply(lambda x:np.average(x['FK Accuracy'
pos_lp=fifa.groupby('BP').apply(lambda x:np.average(x['Long Passing
pos_bc=fifa.groupby('BP').apply(lambda x:np.average(x['Ball Control
pos_ss=fifa.groupby('BP').apply(lambda x:np.average(x['Sprint Speed
pos_spo=fifa.groupby('BP').apply(lambda x:np.average(x['Shot Power'
pos_jm=fifa.groupby('BP').apply(lambda x:np.average(x['Jumping'])).

pos_overall1=pd.merge(pos_head,pos_sp,how='inner',left_on='BP',righ
```

```
pos_overall2=pd.merge(pos_d,pos_c,how='inner',left_on='BP',right_on
pos_overall3=pd.merge(pos_fk,pos_lp,how='inner',left_on='BP',right_
pos_overall4=pd.merge(pos_bc,pos_ss,how='inner',left_on='BP',right_
pos_overall5=pd.merge(pos_spo,pos_jm,how='inner',left_on='BP',right
pos_overall11=pd.merge(pos_overall1,pos_overall2,how='inner',left_o
pos_overall22=pd.merge(pos_overall3,pos_overall4,how='inner',left_o
pos_overall12=pd.merge(pos_overall11,pos_overall22,how='inner',left
pos_overall=pd.merge(pos_overall12,pos_overall5,how='inner',left_on
print('Overall Attributes of the Players in FIFA 21')
fig=plt.figure(figsize=(30,30))
for i in range(0,15):
    labels=np.array(['Heading Accuracy',
                     'Short Passing',
                     'Dribbling',
                     'Curve',
                     'FK Accuracy',
                     'Long Passing',
                     'Ball Control',
                     'Sprint Speed',
                     'Shot Power',
                     'Jumping'])
    stats=pos_overall.loc[i,labels].values

    angles=np.linspace(0, 2*np.pi, len(labels), endpoint=False)
    # close the plot
    stats=np.concatenate((stats,[stats[0]]))
    angles=np.concatenate((angles,[angles[0]]))


    ax = fig.add_subplot(5,3,i+1, polar=True)
    ax.plot(angles, stats, 'o-', linewidth=1)
    ax.fill(angles, stats,color='red',alpha=0.25)
    ax.set_thetagrids(angles[:-1] * 180/np.pi, labels)
    ax.set_title([pos_overall.loc[i,"BP"]])
    ax.grid(True)
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/30
05666391.py:6: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/30
05666391.py:7: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

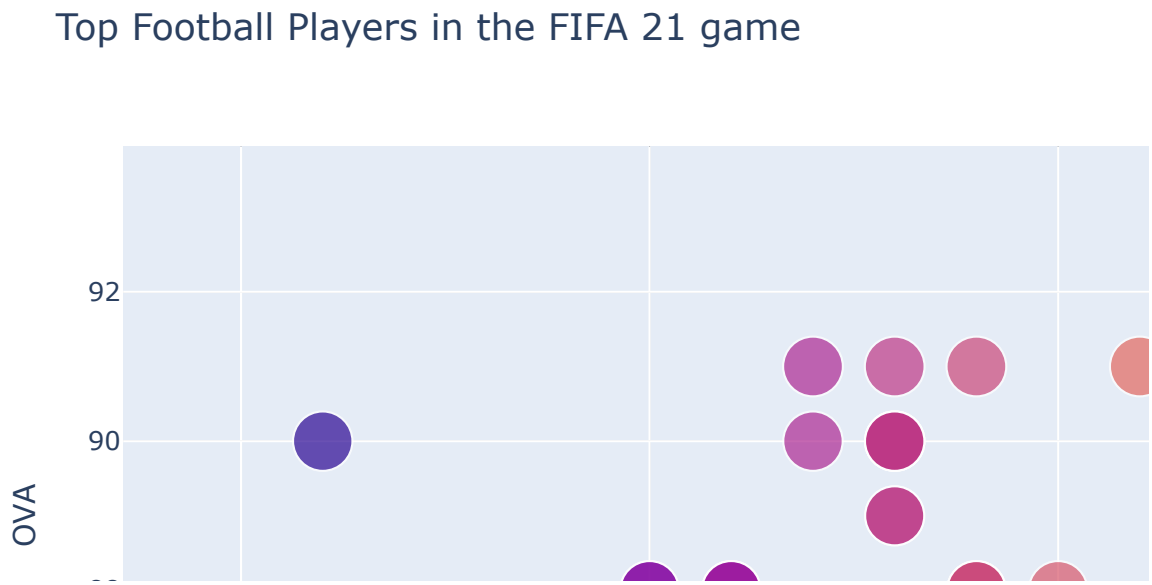
```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/30
```

Observation All overall game attributes have been considered. The paramters considered here have a slight bias against the Goalkeepers, But will be working upon to improve the same. Overall among the selected categories, we see the Center Half players and the Center Forwards have a better performance matrix as compared to the rest of the positions

Who are the best players in the game?

In [18]:
```
top_play=fifa[['Name','OVA',"Age",'Club','BP']]
top_play.sort_values(by='OVA',ascending=False,inplace=True)
top_30_play=top_play[:100]
fig=px.scatter(top_30_play,x='Age',y='OVA',color='Age',size='OVA',h
fig.show()
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/11
23887366.py:2: SettingWithCopyWarning:


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

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



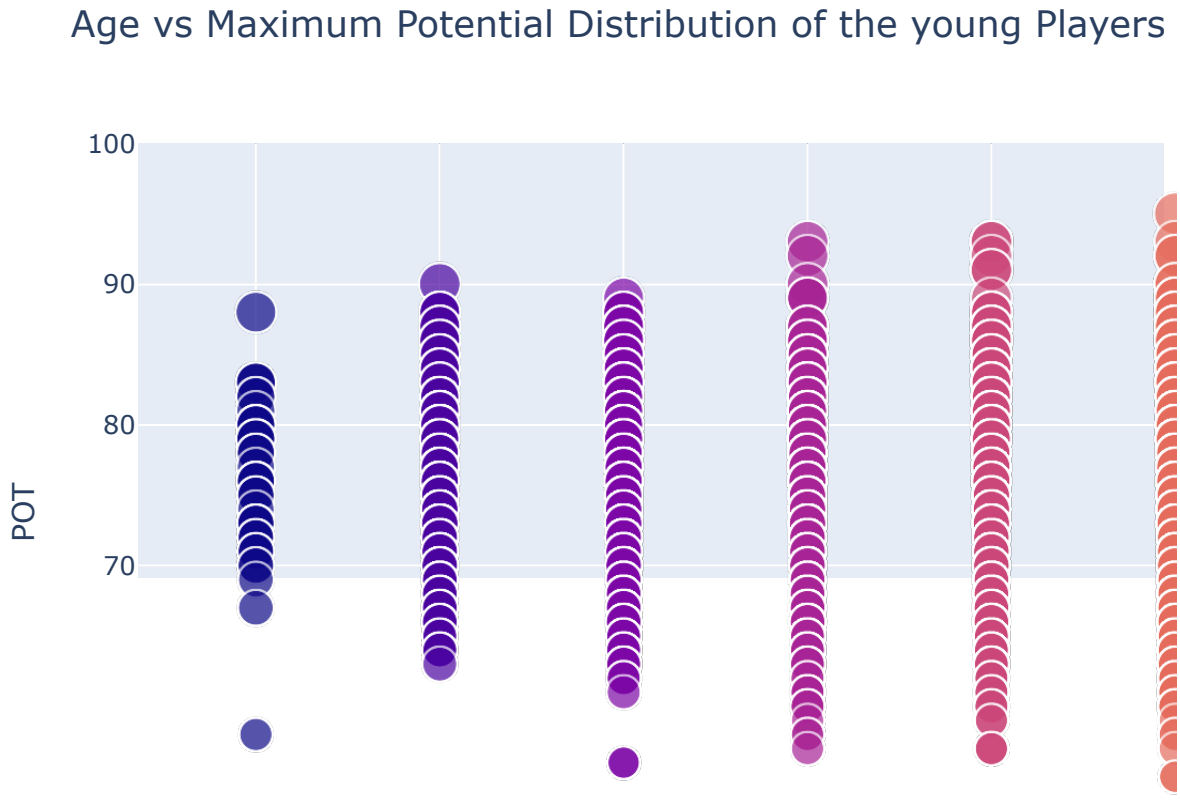
Observation I guess any fan of this beautiful game would accept that we have the 2 best players of this era currently playing- Lionel Messi and Cristiano Ronaldo. But they are now at the age of 33 and 35 respectively- and have a last few years in their hand. Among the youngest players, Kylian Mbappe, Jadon Snacho and Trent Alexander Arnold deserve a special mention.

Who are the Players with the highest Potential??

PS. We will remove the players whose meximum potential is their current form. This was done primarily in order to select the younger players who shall be the top players in the recent future. Hence we have also kept a filter which will help us to identify the best players under the age of 27

In [19]:
```
cond_1=fifa['OVA']!=fifa['POT']
cond_2=fifa['Age']<25
fifa_fil=fifa[cond_1 & cond_2]
pot_play=fifa_fil[['Name','Age','Nationality','Club','POT','BP','OV
pot_play.sort_values(by='POT',ascending=False,inplace=True)
top_pot_play=pot_play[:50]
fig=px.scatter(pot_play,x='Age',y='POT',size='POT',color='Age',hove
fig.show()
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/23
60760120.py:5: SettingWithCopyWarning:


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

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



Observation Kylian Mabappe tops the chart of players who have the maximum potential, and is justified as the cover picture of the FIFA game. Closely following are Sancho, Vinicius Jr., Felix and K. Havertz. Undoubtedly they are the most sought after players currently as seen by their market value and the release clause.

Lets Create some Fantasy Football Teams!!!

Most Preffered Formation:- 3-4-3

Why?#

Personnaly I am able to choose the best player from each position- and hence create the most powerful team. Secondly, as I am a fan of passing football, this formation helps to distribute my game to the maximum. Hence, I shall be making the teams here with this formation.

Best overall Team in Fifa 21

In [21]:
```
final_team=fifa[['Name','Age','OVA','BP','Club']]
final_team.sort_values(by='Age',inplace=True)
pos_play=final_team.groupby('BP').apply(lambda x:np.max(x['OVA'])).
player_pos=pd.merge(final_team,pos_play,how='inner',left_on=['BP','
pos_best=player_pos[['Name','Club','Age','BP','Overall Score']]
cm = sns.light_palette("yellow", as_cmap=True)
pos_best.style.background_gradient(cmap=cm).format(precision=2)
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/27
46767661.py:2: SettingWithCopyWarning:


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

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/27
46767661.py:3: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
```

```
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

Out[21]:

	Name	Club	Age	BP	Overall Score
0	T. Alexander-Arnold	Liverpool	21	RB	87
1	M. Rashford	Manchester United	22	RM	85
2	S. Gnabry	FC Bayern München	24	RM	85
3	A. Robertson	Liverpool	26	LB	87
4	H. Son	Tottenham Hotspur	27	LM	87
5	J. Oblak	Atlético Madrid	27	GK	91
6	Casemiro	Real Madrid	28	CDM	89
7	V. van Dijk	Liverpool	28	CB	90
8	Neymar Jr	Paris Saint-Germain	28	LW	91
9	K. De Bruyne	Manchester City	29	CAM	91
10	T. Kroos	Real Madrid	30	CM	88
11	P. Lahm	FC Bayern München	32	RWB	88
12	Filipe Luís	Atlético Madrid	32	LWB	85
13	K. Benzema	Real Madrid	32	CF	89
14	L. Messi	FC Barcelona	33	RW	93
15	Cristiano Ronaldo	Juventus	35	ST	92

Let's check out the Best playing 11 in our required formation

Observation Wow, I would not even dare to play against this team... too much power packed it is!

Lets make the Team with the players with the highest potential

Note The Potential cannot be equal to the Overall Score, and the age of the players must be smaller than 25

In [23]:
```
final_team=fifa_fil[['Name','Age','POT','BP','Club']]
final_team.sort_values(by='Age',inplace=True)
pos_play=final_team.groupby('BP').apply(lambda x:np.max(x['POT'])).
```

```
player_pos=pd.merge(final_team,pos_play,how='inner',left_on=['BP','
pos_best=player_pos[['Name','Club','Age','BP','Potential']]
cm = sns.light_palette("black", as_cmap=True)
pos_best.style.background_gradient(cmap=cm).format(precision=2)
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/26
41047666.py:2: SettingWithCopyWarning:


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

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/26
41047666.py:3: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

Out[23]:

	Name	Club	Age	BP	Potential
0	Ansu Fati	FC Barcelona	17	LW	90
1	J. Doku	RSC Anderlecht	18	RW	88
2	Vinícius Jr.	Real Madrid	19	RM	93
3	A. Davies	FC Bayern München	19	LB	89
4	S. Tonali	Milan	20	CM	91
5	João Félix	Atlético Madrid	20	CAM	93
6	J. Sancho	Borussia Dortmund	20	CAM	93
7	M. de Ligt	Juventus	20	CB	92
8	Dodô	Shakhtar Donetsk	21	RWB	86
9	T. Alexander-Arnold	Liverpool	21	RB	92
10	K. Mbappé	Paris Saint-Germain	21	ST	95
11	K. Havertz	Chelsea	21	CAM	93
12	G. Donnarumma	Milan	21	GK	92
13	E. Mammana	Zenit St. Petersburg	21	CDM	88
14	B. Ronhaar	Netherlands	23	CF	90
15	Reguilón	Tottenham Hotspur	23	LB	89

16	Grimaldo	SL Benfica	24	LWB	88
17	Rodri	Manchester City	24	CDM	88
18	L. Sané	FC Bayern München	24	LM	90

Lets select the playing XI which has the maximum potential to be the ultimate team in the recent future

Note: In case of 2 playes having the same potential, we have selected the younger player to represent the team.

Observation Only 1 player- Trent Alexander-Arnold is common between both the teams, indicating his terrific performance with respect to his age. But seriously, this team can be a real threat to the top team in FIFA 21! Challenge is now accepted!

Lets create the best fantasy Team for English Premier League!

In [25]:
```
epl_team=fifa[fifa.Club.isin(['Manchester United','Manchester City'
                              'Nottingham Forest','Portsmouth','S
                              'Huddersfield Town','Wolverhampton
                              'Bolton Wanderers','Liverpool','Der
                              'Birmingham City','Swansea City','C
                              'Watford','Swindon Town','Coventry
                              'Cardiff City','Burnley','Wigan Ath
                              'Bristol City','Sheffield United','
epl_final_team=epl_team[['Name','Age','OVA','BP','Club']]
epl_final_team.sort_values(by='Age',inplace=True)
pos_play=epl_final_team.groupby('BP').apply(lambda x:np.max(x['OVA'
player_pos=pd.merge(epl_final_team,pos_play,how='inner',left_on=['B
pos_best=player_pos[['Name','Club','Age','BP','Overall Score']]
cm = sns.light_palette("red", as_cmap=True)
pos_best.style.background_gradient(cmap=cm).format(precision=2)
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/34
78517245.py:10: SettingWithCopyWarning:


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

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/34
78517245.py:11: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

Out[25]:

	Name	Club	Age	BP	Overall Score
0	T. Alexander-Arnold	Liverpool	21	RB	87
1	M. Rashford	Manchester United	22	RM	85

2	B. Mendy	Manchester City	25	LWB	81
3	João Cancelo	Manchester City	26	RWB	83
4	Jonny	Wolverhampton Wanderers	26	LWB	81
5	A. Robertson	Liverpool	26	LB	87
6	Nélson Semedo	Wolverhampton Wanderers	26	RWB	83
7	Alisson	Liverpool	27	GK	90
8	H. Son	Tottenham Hotspur	27	LM	87
9	P. Pogba	Manchester United	27	CM	86
10	S. Mané	Liverpool	28	LW	90
11	M. Salah	Liverpool	28	RW	90
12	V. van Dijk	Liverpool	28	CB	90
13	Roberto Firmino	Liverpool	28	CF	87
14	K. De Bruyne	Manchester City	29	CAM	91
15	N. Kanté	Chelsea	29	CDM	88
16	S. Agüero	Manchester City	32	ST	89

Let's check out one of the best teams of English Premier League in Fifa 21

Observation Well if this team would play in the Premier League, even the Classic XI would fall. Truly a great combination of players

Now, Let's check for the Team I support... FC Barcelona

Who are the best agewise players in BARCA???

In [26]:
```
barca_team_all_players=fifa[fifa['Club']=='FC Barcelona']
```

```
barca_team_all_players
```

Out[26]:

	ID	Name	Age	OVA	Nationality	Club	BOV	BP	Position	
35	1615	L. Thuram	35	83	France	FC Barcelona	83	CB	CB	https://
104	10535	Xavi	34	86	Spain	FC Barcelona	85	CAM	CM	https://
117	13038	Puyol	35	83	Spain	FC Barcelona	82	CB	CB	https://
314	110082	Pinto	37	72	Spain	FC Barcelona	72	GK	NaN	https://
643	152729	Piqué	33	86	Spain	FC Barcelona	86	CB	CB	https://
752	158023	L. Messi	33	93	Argentina	FC Barcelona	93	RW	RW ST CF	https://
1632	180206	M. Pjanić	30	85	Bosnia Herzegovina	FC Barcelona	85	CM	CM CDM	https://
2385	189242	Coutinho	28	83	Brazil	FC Barcelona	83	CAM	CAM LW CM	https://
2398	189332	Jordi Alba	31	86	Spain	FC Barcelona	86	LB	LB	https://
2435	189511	Sergio Busquets	31	87	Spain	FC Barcelona	87	CDM	CDM	https://
2790	192448	M. ter Stegen	28	90	Germany	FC Barcelona	90	GK	GK	https://
2934	193290	M. Braithwaite	29	76	Denmark	FC Barcelona	76	ST	ST LM	https://
3087	194404	Neto	30	82	Brazil	FC Barcelona	82	GK	GK	https://
3102	194765	A. Griezmann	29	87	France	FC Barcelona	87	ST	ST CF LW	https://
3524	199564	Sergi Roberto	28	83	Spain	FC Barcelona	83	RB	RB CM	https://
3848	201400	Rafinha	27	79	Brazil	FC Barcelona	81	CAM	CM CAM RM	https://
4586	205600	S. Umtiti	26	83	France	FC Barcelona	85	CB	CB	https://
6844	220440	C. Lenglet	25	85	France	FC Barcelona	86	CB	CB	https://
7281	222737	Malcom	21	82	Brazil	FC Barcelona	82	CAM	RW	https://
8489	228702	F. de Jong	23	85	Netherlands	FC Barcelona	87	CM	CM	https://
8645	229337	Matheus Pereira	22	69	Brazil	FC Barcelona	71	CAM	CAM RM	https://

9089	230800	R. Manaj	23	69	Albania	FC Barcelona	71	ST	ST	https://
9258	231443	O. Dembélé	23	83	France	FC Barcelona	85	LM	LW RW	https://
9656	233113	Aleñá	22	74	Spain	FC Barcelona	76	CAM	CM RM CAM	https://
10936	237735	Matheus Fernandes	22	68	Brazil	FC Barcelona	70	CM	CM	https://
10957	237823	H. Abe	21	69	Japan	FC Barcelona	70	RM	LW LM	https://
11331	239249	Oriol Busquets	21	67	Spain	FC Barcelona	68	CDM	CDM CB	https://
11768	240683	L. Reis	20	70	Netherlands	FC Barcelona	72	CM	CM CDM	https://
11852	240915	Miranda	20	70	Spain	FC Barcelona	70	LB	LB LWB	https://
11953	241184	Junior Firpo	23	79	Spain	FC Barcelona	79	LB	LB LM	https://
12479	242816	Riqui Puig	20	75	Spain	FC Barcelona	77	CAM	CM CAM	https://
12529	242999	Álex Collado	20	65	Spain	FC Barcelona	68	CAM	CM LW	https://
13168	244778	Trincão	20	78	Portugal	FC Barcelona	79	CAM	RW LW CAM	https://
13371	245388	J. Todibo	20	75	France	FC Barcelona	77	CB	CB	https://
13572	246139	Iñaki Peña	21	65	Spain	FC Barcelona	65	GK	GK	https://
14795	251854	Pedri	17	72	Spain	FC Barcelona	75	CAM	LM CAM	https://
14886	252072	S. Ramos Mingo	17	62	Argentina	FC Barcelona	63	CB	CB LB	https://
15221	253004	Ansu Fati	17	76	Spain	FC Barcelona	76	LW	LW RW	https://
15279	253163	R. Araujo	21	67	Uruguay	FC Barcelona	69	CB	CB	https://
16407	257025	K. de la Fuente	18	63	United States	FC Barcelona	64	RM	LW RW	https://
16524	257315	Gustavo Maia	19	69	Brazil	FC Barcelona	70	CAM	LW RW CF	https://

41 rows × 107 columns

Observation We see some of the players are present in the list- who have retired from the game. This might be an issue in the data. Hence let us remove those players from the team level analysis. We only keep the list of those players who have their contract ending after 2020.

Keeping only the eligible players

In [27]:
```
barca_team_all_players['Contract Start']=pd.to_numeric(barca_team_a
barca_team_all_players['Contract End']=pd.to_numeric(barca_team_all
barca_team_all=barca_team_all_players[barca_team_all_players['Contr
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/36
50230660.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/36
50230660.py:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

Agewise best players in Barcelona Team...

In [29]:
```
barca_team=barca_team_all[['Name','Age','OVA','BP']]
barca_team.sort_values(by='Age',inplace=True)
age_play=barca_team.groupby('Age').apply(lambda x:np.max(x['OVA']))
player_age=pd.merge(barca_team,age_play,how='inner',left_on=['Age',
age_best=player_age[['Name','Age','BP','Overall Score']]
cm = sns.light_palette("green", as_cmap=True)
age_best.style.background_gradient(cmap=cm).format(precision=2)
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/25
```

```
54811345.py:2: SettingWithCopyWarning:


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

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/2554811345.py:3: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.
```

Out[29]:

	Name	Age	BP	Overall Score
0	Ansu Fati	17	LW	76
1	K. de la Fuente	18	RM	63
2	Gustavo Maia	19	CAM	69
3	Trincão	20	CAM	78
4	Malcom	21	CAM	82
5	Aleñá	22	CAM	74
6	F. de Jong	23	CM	85
7	C. Lenglet	25	CB	85
8	S. Umtiti	26	CB	83
9	Rafinha	27	CAM	79
10	M. ter Stegen	28	GK	90
11	A. Griezmann	29	ST	87
12	M. Pjanić	30	CM	85
13	Sergio Busquets	31	CDM	87
14	L. Messi	33	RW	93

Who are the best position players in Barca???

In [31]:
```
barca_team_2=barca_team_all[['Name','Age','OVA','BP']]
barca_team_2.sort_values(by='BP',inplace=True)
age_play=barca_team_2.groupby('BP').apply(lambda x:np.max(x['OVA']))
player_pos=pd.merge(barca_team_2,age_play,how='inner',left_on=['BP'
pos_best=player_pos[['Name','Age','BP','Overall Score']]
cm = sns.light_palette("blue", as_cmap=True)
pos_best.style.background_gradient(cmap=cm).format(precision=2)
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/17
96703359.py:2: SettingWithCopyWarning:


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

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/17
96703359.py:3: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

Out[31]:

	Name	Age	BP	Overall Score
0	Coutinho	28	CAM	83
1	Piqué	33	CB	86
2	Sergio Busquets	31	CDM	87
3	F. de Jong	23	CM	85
4	M. Pjanić	30	CM	85
5	M. ter Stegen	28	GK	90
6	Jordi Alba	31	LB	86
7	O. Dembélé	23	LM	83
8	Ansu Fati	17	LW	76
9	Sergi Roberto	28	RB	83
10	H. Abe	21	RM	69
11	L. Messi	33	RW	93
12	A. Griezmann	29	ST	87

Let's make the Dream formation for the Barca Team using the analysis above!

Observation Well this would be the Team I would want in Barca. Have still not been able to come out of the 8-2 nightmare by Bayern Munich. But Keeping the fingers crossed!

Finally, I would like to make a team for my very country... India

In [33]:
```
india_team_all_players=fifa[fifa['Nationality']=='India']
india_team_2=india_team_all_players[['Name','Age','OVA','BP']]
india_team_2.sort_values(by='BP',inplace=True)
age_play=india_team_2.groupby('BP').apply(lambda x:np.max(x['OVA']))
player_pos=pd.merge(india_team_2,age_play,how='inner',left_on=['BP'
pos_best=player_pos[['Name','Age','BP','Overall Score']]
cm = sns.light_palette("cyan", as_cmap=True)
pos_best.style.background_gradient(cmap=cm).format(precision=2)
```

```
/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/21
00336677.py:3: SettingWithCopyWarning:


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

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

/var/folders/dt/lnc313ld61bfwvn1rgjkmj_40000gn/T/ipykernel_3826/21
00336677.py:4: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_
groups=False` to exclude the groupings or explicitly select the gr
ouping columns after groupby to silence this warning.
```

Out[33]:

	Name	Age	BP	Overall Score
0	A. Swaminathan	28	CAM	61
1	A. Chakraborty	34	CB	61
2	B. Sidhu	32	CM	60
3	G. Chatterjee	34	GK	64
4	A. Ginti	26	LB	61
5	T. Agarwal	34	LM	58
6	L. Ralte	25	LW	61
7	B. Raj	32	RB	64
8	H. Bhandari	31	RM	62
9	U. Singh	22	RW	60
10	S. Chhetri	33	ST	67

Lets make the Team for India as done above!

In [ ]: