Exploratory Data Analysis - English Premier League 2020/2021

The dataset includes lots of different statistics about games.

- xG, xA: Expected goals and expected assists of each individual player.
- Scored and Assists: Goal scored and Assists.
- Passes: Passes attempted and percentage of passes completed of eachindividual player.
- Penalty: Penalty scored and Penalty attempts.

There are also basic stats such as yellow cards, red cards, age, club representing, nationality, position, starts and minutes.

Import Libraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
%matplotlib inline
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/) t
hat 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
```

Dataset

Here we are displaying the first 5 rows of the datasets which includes the statistics of each player from the English Premier League.

In [2]:
```
epl = pd.read_csv('/Users/andres_th14/Downloads/EPL_20_21.csv')
epl.head()
```

Out[2]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_A
0	Mason Mount	Chelsea	ENG	MF,FW	21	36	32	2890	6	5	
1	Edouard Mendy	Chelsea	SEN	GK	28	31	31	2745	0	0	
2	Timo Werner	Chelsea	GER	FW	24	35	29	2602	6	8	
3	Ben Chilwell	Chelsea	ENG	DF	23	27	27	2286	3	5	
4	Reece James	Chelsea	ENG	DF	20	32	25	2373	1	2	

Data Exploration

In [3]:
```
# Number of rows and columns
rows, cols = epl.shape
print('Number of players: {}'.format(rows))
print('Number of stats per player: {}'.format(cols))
```

```
Number of players: 532
Number of stats per player: 18
```

In [4]:
```
list(epl.columns)
```

Out[4]:
```
['Name',
 'Club',
 'Nationality',
 'Position',
 'Age',
 'Matches',
 'Starts',
 'Mins',
 'Goals',
 'Assists',
 'Passes_Attempted',
 'Perc_Passes_Completed',
 'Penalty_Goals',
 'Penalty_Attempted',
 'xG',
 'xA',
 'Yellow_Cards',
 'Red_Cards']
```

In [5]:
```
epl.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 532 entries, 0 to 531
Data columns (total 18 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Name                   532 non-null    object 
 1   Club                   532 non-null    object 
 2   Nationality            532 non-null    object 
 3   Position               532 non-null    object 
 4   Age                    532 non-null    int64  
 5   Matches                532 non-null    int64  
 6   Starts                 532 non-null    int64  
 7   Mins                   532 non-null    int64  
 8   Goals                  532 non-null    int64  
 9   Assists                532 non-null    int64  
 10  Passes_Attempted       532 non-null    int64  
 11  Perc_Passes_Completed  532 non-null    float64
 12  Penalty_Goals          532 non-null    int64  
 13  Penalty_Attempted      532 non-null    int64  
 14  xG                     532 non-null    float64
 15  xA                     532 non-null    float64
 16  Yellow_Cards           532 non-null    int64  
 17  Red_Cards              532 non-null    int64  
dtypes: float64(3), int64(11), object(4)
memory usage: 74.9+ KB
```

In [6]:
```
epl_club_position=epl.groupby('Club').Position.value_counts()
epl_club_position
```

Out[6]:
```
Club                      Position
Arsenal                   DF          11
                          FW           6
                          MF           5
                          GK           3
                          FW,DF        1
                                      ..
Wolverhampton Wanderers   FW,MF        3
                          MF           3
                          GK           2
                          MF,FW        2
                          MF,DF        1
Name: Position, Length: 145, dtype: int64
```

In [7]:
```
epl.describe()
```

Out[7]:

	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempt
count	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.0000
mean	25.500000	19.535714	15.714286	1411.443609	1.853383	1.287594	717.7500
std	4.319404	11.840459	11.921161	1043.171856	3.338009	2.095191	631.3725
min	16.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.0000
25%	22.000000	9.000000	4.000000	426.000000	0.000000	0.000000	171.5000
50%	26.000000	21.000000	15.000000	1345.000000	1.000000	0.000000	573.5000
75%	29.000000	30.000000	27.000000	2303.500000	2.000000	2.000000	1129.5000
max	38.000000	38.000000	38.000000	3420.000000	23.000000	14.000000	3214.0000

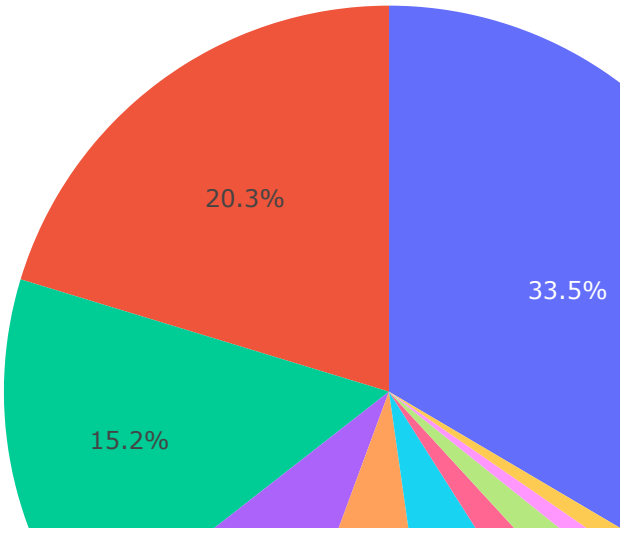
Exploratory Data Analysis and Data Visualization

Number of Players In Each Position

In [8]:
```
# Number of players in the EPL by each position
epl_position = epl['Position'].value_counts()
print(epl_position)
```

```
DF       178
MF       108
FW        81
FW,MF     47
GK        42
MF,FW     36
DF,MF     15
MF,DF     13
DF,FW      6
FW,DF      6
Name: Position, dtype: int64
```

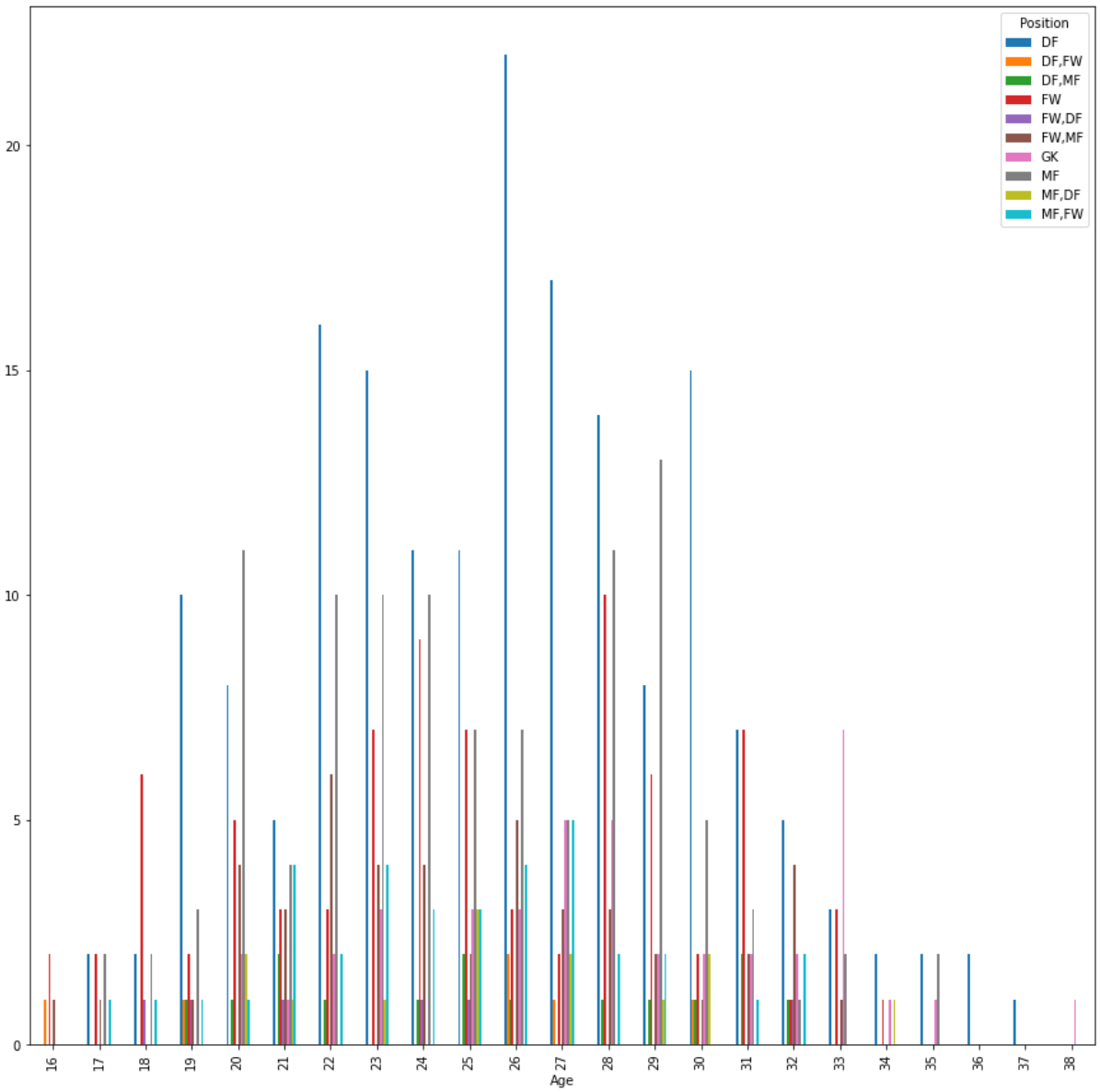
In [9]:
```
fig = px.pie(epl_position, values=epl_position.values, names=epl_positio
n.index)
fig.show()
```



Players By Age In Each Position

In [10]:
```
pd.crosstab(epl['Age'], epl['Position']).plot(kind='bar',figsize=(15,15
))
```

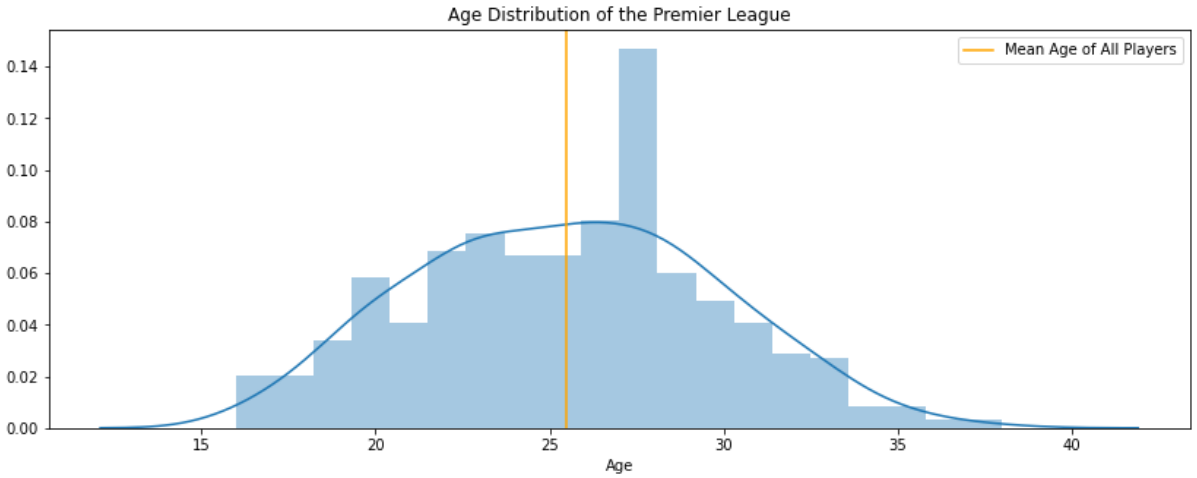
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x125cff370>



Age Distribution of the Premier League

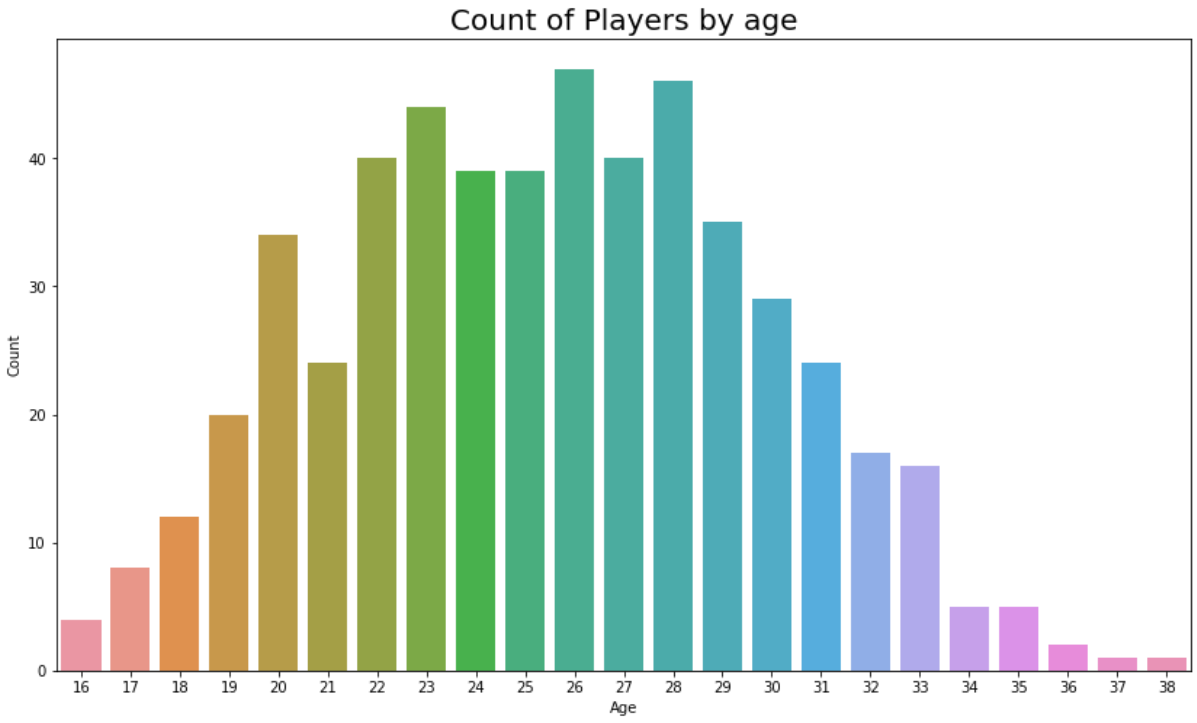
In [11]:
```
plt.figure(figsize=(14,5))
plt.title('Age Distribution of the Premier League')
sns.distplot(a=epl['Age'], kde=True, bins=20)
plt.axvline(x=np.mean(epl['Age']),c='orange',label='Mean Age of All Players')
plt.legend()
```

Out[11]: <matplotlib.legend.Legend at 0x125e213a0>



Players by Age of the Premier League

In [12]:
```
plt.figure(figsize= (14,8))

ax = sns.countplot(x='Age', data=epl)
ax.set_title(label='Count of Players by age', fontsize=20)

ax.set_xlabel(xlabel='Age')
ax.set_ylabel(ylabel='Count')

plt.show()
```



In [13]:
```
print("Oldest Player/s: ")
epl.loc[epl['Age'] == epl['Age'].max()]
```

```
Oldest Player/s:
```

Out[13]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes
22	Willy Caballero	Chelsea	ARG	GK	38	1	1	90	0	0	

In [14]:
```
print("Youngest Player/s: ")
epl.loc[epl['Age'] == epl['Age'].min()]
```

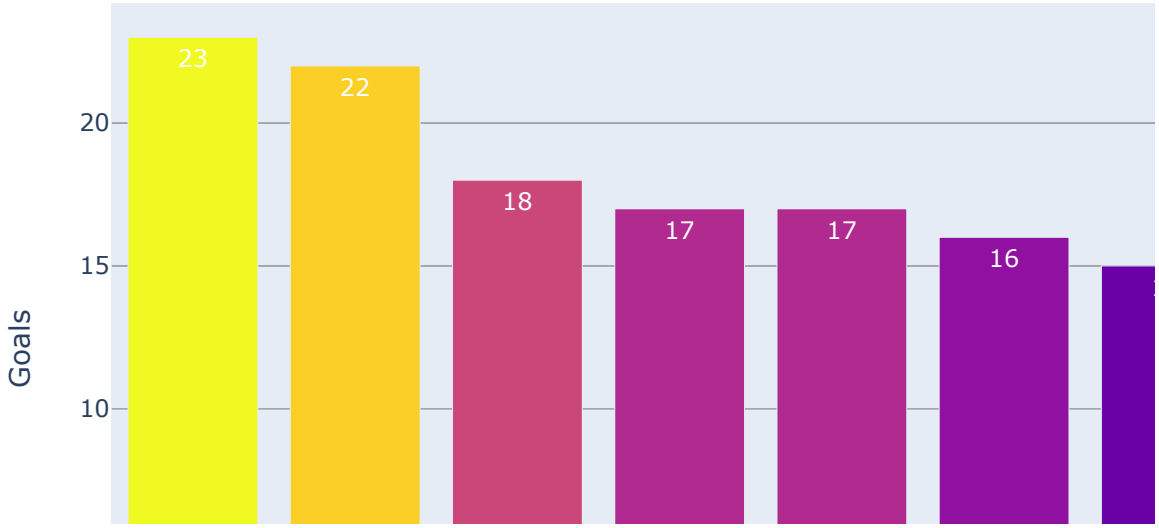
```
Youngest Player/s:
```

Out[14]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists
76	Shola Shoretire	Manchester United	ENG	FW	16	2	0	11	0	0
182	Dane Scarlett	Tottenham Hotspur	ENG	FW	16	1	0	1	0	0
284	Carney Chukwuemeka	Aston Villa	ENG	FW,MF	16	2	0	20	0	0
530	Antwoine Hackford	Sheffield United	ENG	DF,FW	16	1	0	11	0	0

Premier League Top Goalscorers

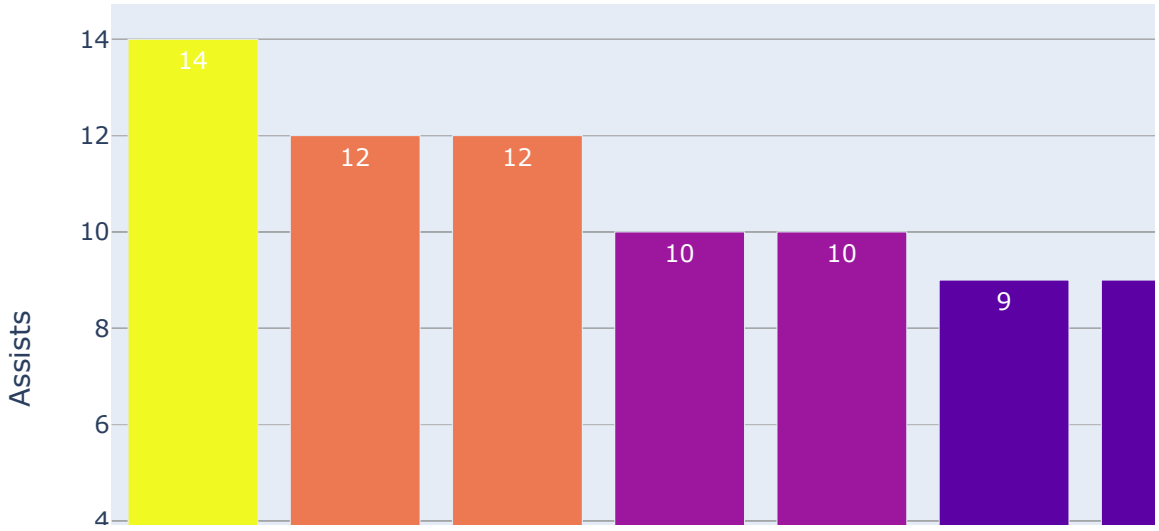
In [15]:
```
epl_top_goals = epl.sort_values(by=['Goals'], ascending=False)[:10]
fig = px.bar(epl_top_goals, x='Name', y='Goals',color='Goals',hover_data
=['Club','Age'],text='Goals')
fig.show()
```



Harry Kane from Tottenham Hotspur was the Premier League Top Scorer with 23 goals this season.

Premier League Top Assists

In [16]:
```
epl_top_assists = epl.sort_values(by=['Assists'], ascending=False)[:10]

fig = px.bar(epl_top_assists, x='Name', y='Assists',color='Assists',hove
r_data=['Club','Age'],text='Assists')
fig.show()
```



Harry Kane also had the most assists in the Premier League with 14 assists this season.

Goals per 90 minutes

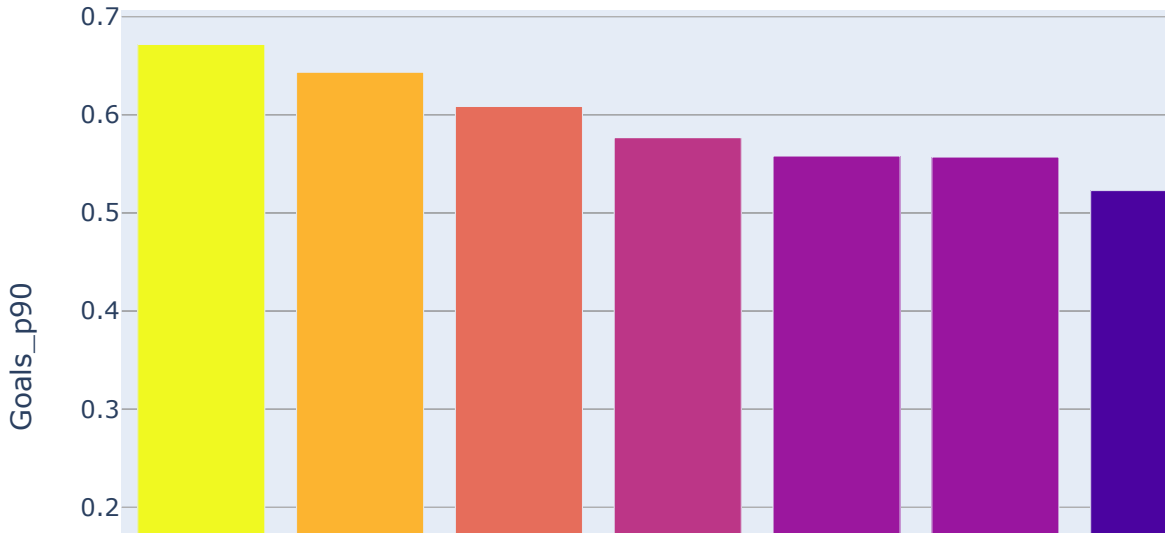
- Goals Per90 A player or team goal tally divided into 90 minute chunks. We do this to normalize for actual time played, as it produces far more accurate rates of goal scoring than using appearances, starts, etc.

In [17]:
```
epl['Goals_p90'] = epl['Goals']/epl['Mins']*90
epl.head()
```

Out[17]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_A
0	Mason Mount	Chelsea	ENG	MF,FW	21	36	32	2890	6	5	
1	Edouard Mendy	Chelsea	SEN	GK	28	31	31	2745	0	0	
2	Timo Werner	Chelsea	GER	FW	24	35	29	2602	6	8	
3	Ben Chilwell	Chelsea	ENG	DF	23	27	27	2286	3	5	
4	Reece James	Chelsea	ENG	DF	20	32	25	2373	1	2	

In [18]:
```
epl_goals_p90 = epl[epl['Mins']>1500].sort_values(by=['Goals_p90'], ascending=False)[:10]

fig = px.bar(epl_goals_p90, x='Name', y='Goals_p90',color='Goals_p90',hover_data=['Club','Age'])
fig.show()
```



Harry Kane had the highest goals per 90 with 0.67 from sorting all players that has played beyond 1500 minutes in the season.

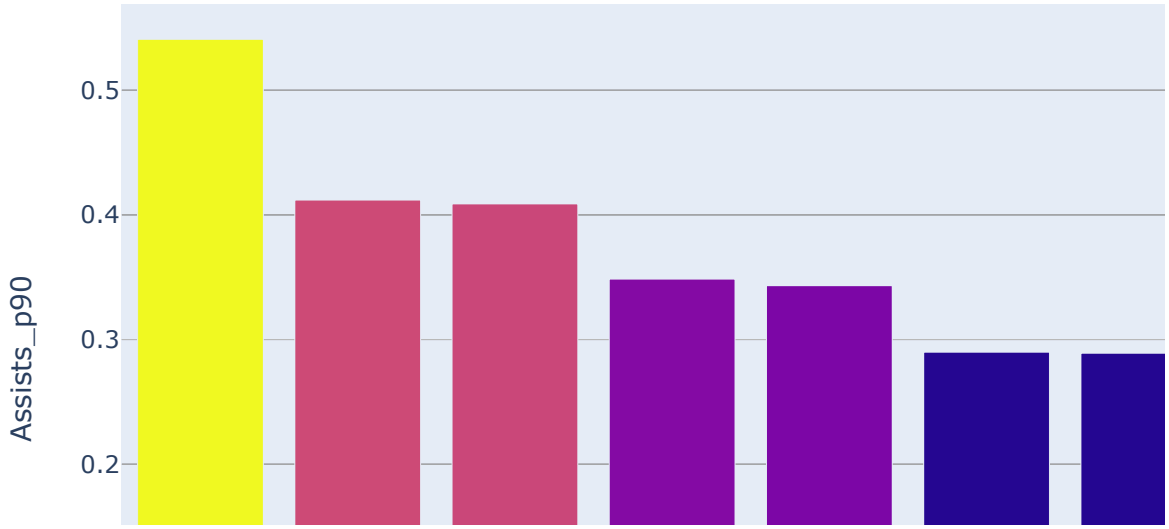
Assists Per 90 Minutes

In [19]:
```
epl['Assists_p90'] = epl['Assists']/epl['Mins']*90
epl.head()
```

Out[19]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_A
0	Mason Mount	Chelsea	ENG	MF,FW	21	36	32	2890	6	5	
1	Edouard Mendy	Chelsea	SEN	GK	28	31	31	2745	0	0	
2	Timo Werner	Chelsea	GER	FW	24	35	29	2602	6	8	
3	Ben Chilwell	Chelsea	ENG	DF	23	27	27	2286	3	5	
4	Reece James	Chelsea	ENG	DF	20	32	25	2373	1	2	

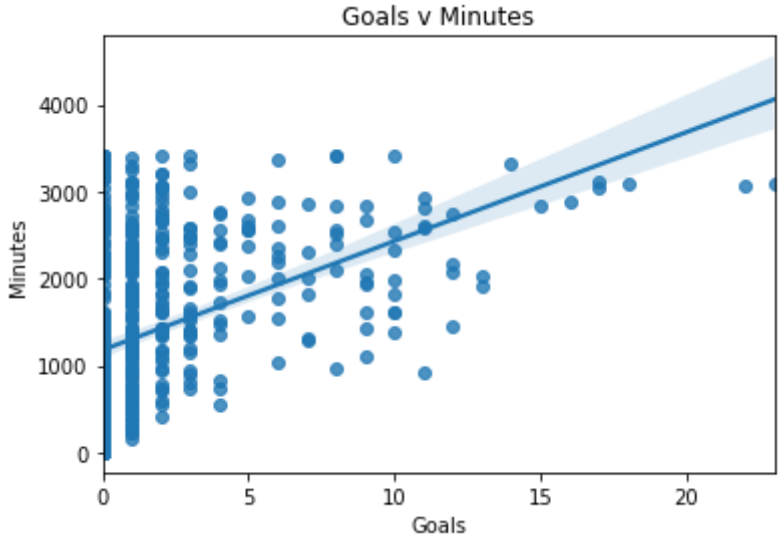
In [20]:
```
epl_Assists_p90 = epl[epl['Mins']>1500].sort_values(by=['Assists_p90'],
ascending=False)[:10]

fig = px.bar(epl_Assists_p90, x='Name', y='Assists_p90',color='Assists_p
90',hover_data=['Club','Age'])
fig.show()
```



Kevin De Bruyne from Manchester City had the highest Assists per 90 with 0.54 from sorting all players that has played beyond 1500 minutes in the season.

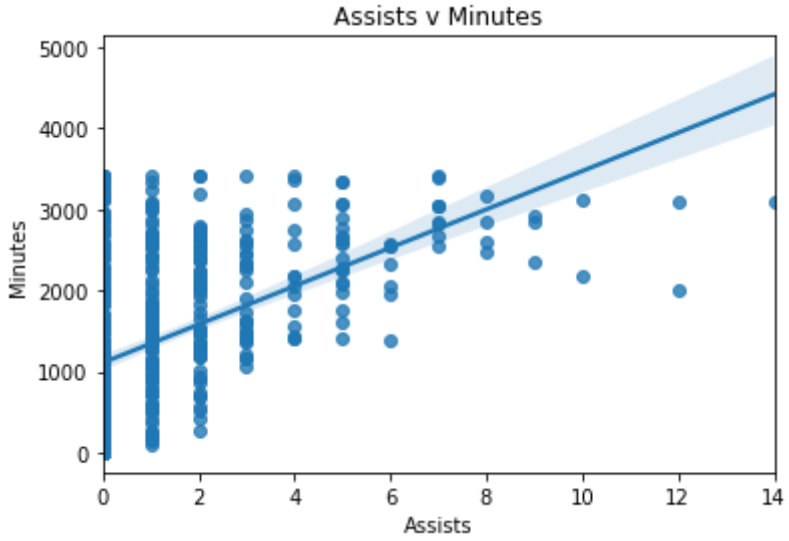
Goals and Minutes

In [21]:
```
plt.figure()
x=epl['Goals']
y=epl['Mins']

sns.regplot(x,y)
plt.title('Goals v Minutes')
plt.xlabel('Goals')
plt.ylabel('Minutes')
plt.show()
```



Assists and Minutes

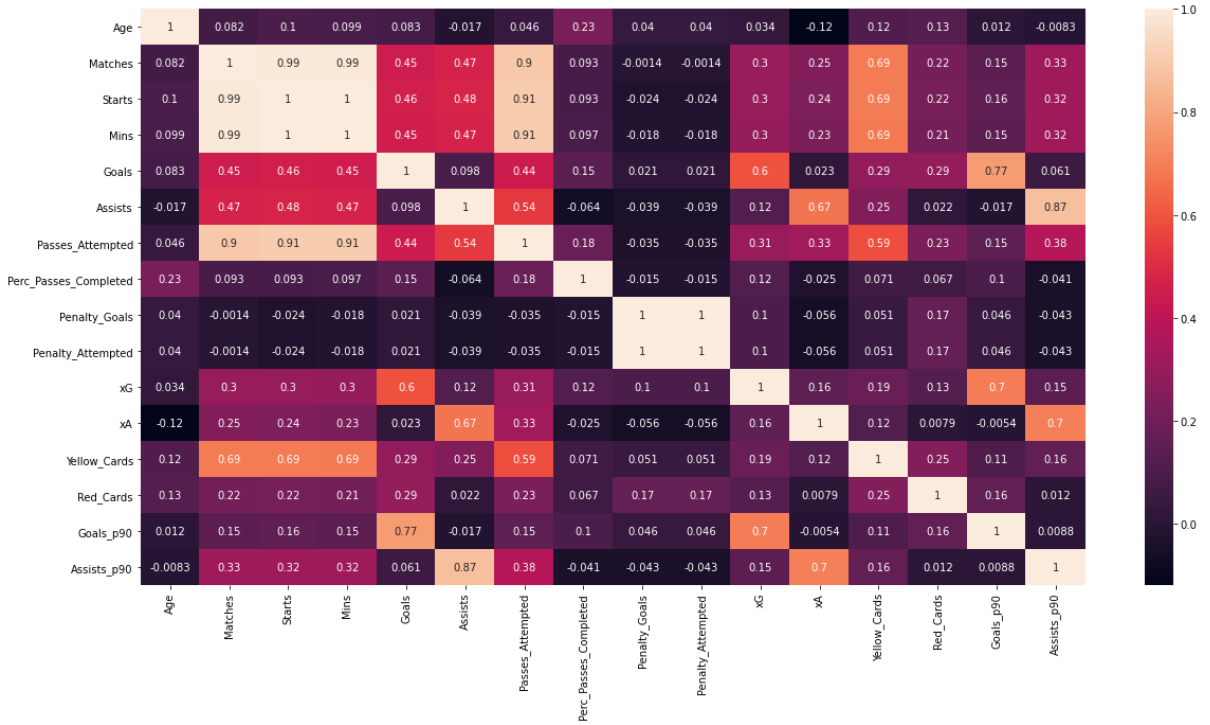
In [22]:
```
plt.figure()
x=epl['Assists']
y=epl['Mins']

sns.regplot(x,y)
plt.title('Assists v Minutes')
plt.xlabel('Assists')
plt.ylabel('Minutes')
plt.show()
```



Defenders of the Premier League

In [23]:
```
# Taking subsets of defenders data for analysis
epl_defender = epl[epl['Position'] == 'DF']
```

In [24]:
```
plt.figure(figsize=(20,10))
sns.heatmap(epl_defender.corr(), annot=True)
plt.show()
```



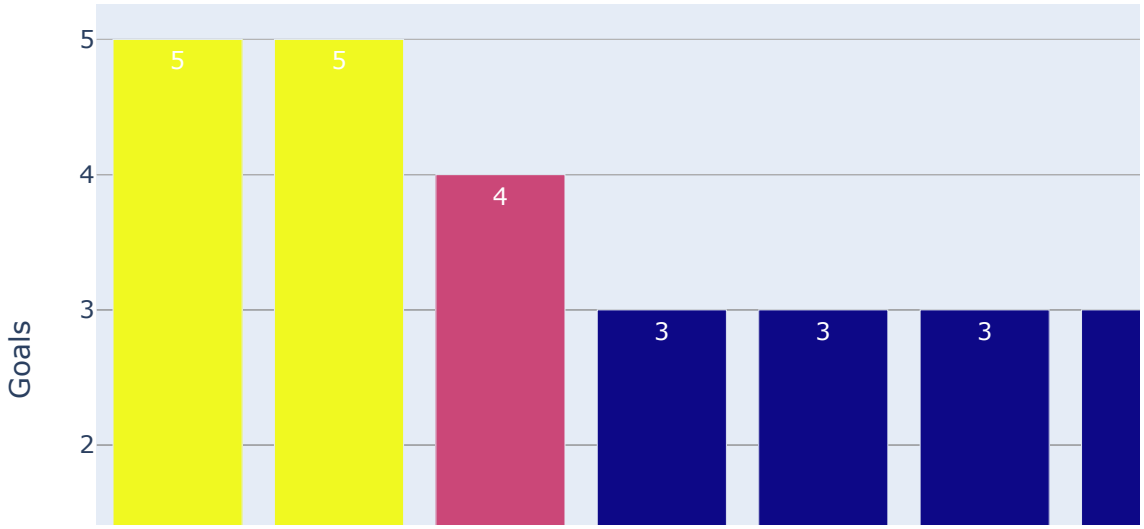
In [25]:
```
epl_defender.describe()
```

Out[25]:

	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempt
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.0000
mean	25.780899	18.207865	16.685393	1490.617978	0.724719	0.764045	909.3370
std	4.204151	11.411489	11.708412	1035.320792	0.995786	1.465401	705.9164
min	17.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.0000
25%	23.000000	9.000000	7.000000	580.250000	0.000000	0.000000	309.5000
50%	26.000000	18.000000	15.000000	1400.000000	0.000000	0.000000	781.0000
75%	28.750000	28.000000	27.000000	2372.750000	1.000000	1.000000	1475.7500
max	37.000000	38.000000	38.000000	3404.000000	5.000000	8.000000	3214.0000

Top 10 Defenders With Most Goals Scored

In [26]:
```
epl_top_goals_defender = epl_defender.sort_values(by=['Goals'], ascending=False)[:10]

fig = px.bar(epl_top_goals_defender, x='Name', y='Goals',color='Goals',hover_data=['Club','Age'],text='Goals')
fig.show()
```



Lewis Dunk from Brighton and Kurt Zouma from Chelsea FC were the Defender top scorer in the Premier League with 5 goals each.

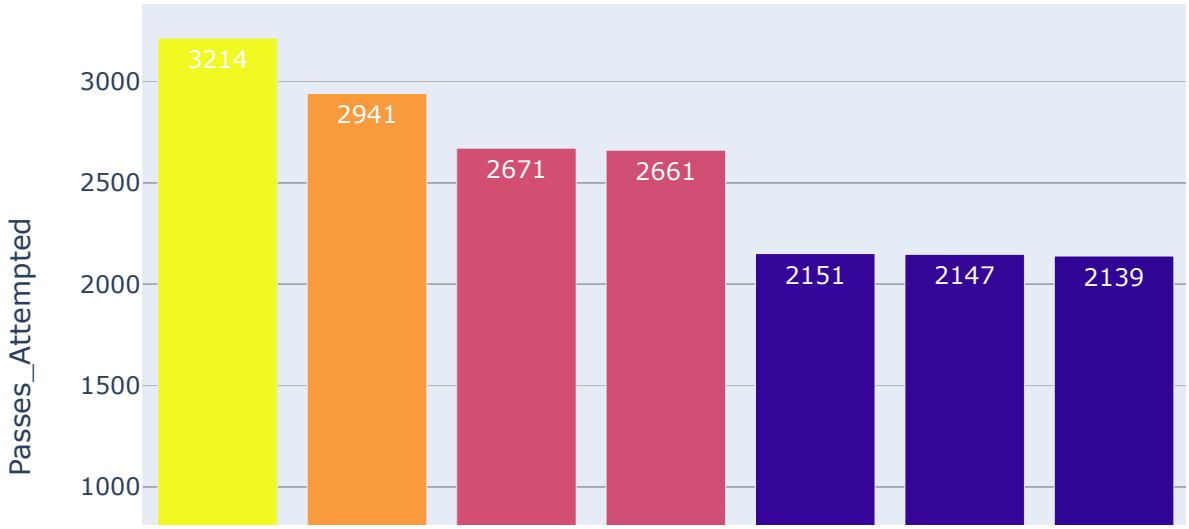
Top 10 Defenders With Most Assists

In [27]:
```
epl_top_assists_defender = epl_defender.sort_values(by=['Assists'], ascending=False)[:10]

fig = px.bar(epl_top_assists_defender, x='Name', y='Assists',color='Assists',hover_data=['Club','Age'],text='Assists')
fig.show()
```

Aaron Cresswell from West Ham was the Defender top assister in the Premier League with 8 assists.

Top 10 Defenders With Most Passes Attempted

In [28]:
```
epl_attempt_pass_defender = epl_defender.sort_values(by=['Passes_Attempted'], ascending=False)[:10]
fig = px.bar(epl_attempt_pass_defender, x='Name', y='Passes_Attempted',color='Passes_Attempted',hover_data=['Club','Age'],text='Passes_Attempted')
fig.show()
```



Andrew Robertson from Liverpool had the most passes attempted in the league with 3214 passes.

Premier League Clubs Defenders With Most Passes Attempted

In [29]:
```
epl_defender['Passes_Attempted'].groupby(epl_defender['Club']).sum().sort_values(ascending=False)
```

Out[29]:
```
Club
Manchester City            12861
Chelsea                    12648
Liverpool FC                9827
Leicester City              9616
Manchester United           9615
Arsenal                     9047
Wolverhampton Wanderers     8683
Tottenham Hotspur           8593
Fulham                      8513
Southampton                 8309
Brighton                    8153
Everton                     7315
Aston Villa                 7095
West Ham United             6677
Leeds United                6609
Crystal Palace              6319
West Bromwich Albion        5715
Burnley                     5573
Sheffield United            5432
Newcastle United            5262
Name: Passes_Attempted, dtype: int64
```

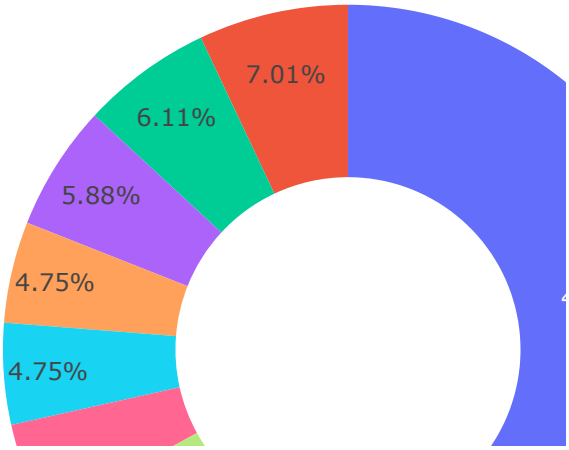
Premier League Clubs Defenders Average Percentage of Passes Completed beyond 15 Games Played

In [30]:
```
epl_perc_defender = epl_defender[epl_defender['Matches'] >= 15].sort_val
ues('Perc_Passes_Completed',ascending=False)
epl_club_pass = epl_perc_defender['Perc_Passes_Completed'].groupby(epl_p
erc_defender['Club']).mean().sort_values(ascending=False)
print(epl_club_pass)
```

```
Club
Manchester City            90.866667
Chelsea                    88.571429
Manchester United          86.925000
Wolverhampton Wanderers    84.433333
Leicester City             84.333333
Arsenal                    83.600000
Everton                    83.140000
Leeds United               83.125000
Southampton                81.840000
Tottenham Hotspur          81.642857
Liverpool FC               80.800000
Sheffield United           79.725000
West Ham United            79.400000
Brighton                   79.300000
Aston Villa                79.250000
Fulham                     79.100000
Newcastle United           78.242857
Crystal Palace             78.216667
Burnley                    71.450000
West Bromwich Albion       70.540000
Name: Perc_Passes_Completed, dtype: float64
```

Players Nationality of the Premier League

In [31]:
```
epl_Nationality = epl['Nationality'].value_counts().head(20)
fig = go.Figure(data=[go.Pie(labels=epl_Nationality.index, values=epl_Na
tionality.values, hole=.5)])
fig.show()
```



Number of Players in Each Club

In [32]:
```
grouped_by_club = epl.groupby('Club').size()
print(grouped_by_club)
```

```
Club
Arsenal                    29
Aston Villa                24
Brighton                   27
Burnley                    25
Chelsea                    27
Crystal Palace             24
Everton                    29
Fulham                     28
Leeds United               23
Leicester City             27
Liverpool FC               28
Manchester City            24
Manchester United          29
Newcastle United           27
Sheffield United           27
Southampton                29
Tottenham Hotspur          24
West Bromwich Albion       30
West Ham United            24
Wolverhampton Wanderers    27
dtype: int64
```

Premier League Goals of each Club

In [33]:
```
epl_club_goals = epl['Goals'].groupby(epl['Club']).sum().sort_values(asc
ending=False)
print(epl_club_goals)
```

```
Club
Manchester City            82
Manchester United          70
Tottenham Hotspur          66
Liverpool FC               65
Leicester City             64
Leeds United               60
West Ham United            60
Chelsea                    56
Arsenal                    53
Aston Villa                52
Southampton                47
Everton                    45
Newcastle United           44
Crystal Palace             39
Brighton                   39
Wolverhampton Wanderers    34
West Bromwich Albion       33
Burnley                    32
Fulham                     26
Sheffield United           19
Name: Goals, dtype: int64
```