Part_I_exploration_template

May 26, 2022

1 Part I - (Exploration of FIFA 2019 Dataset)

1.1 by Paul Anugwom

1.2 Introduction

This dataset consists of all the details of the players who participated in the FIFA World Cup 2019 and their final value at which they were to be sold to other the clubs.Data can be found in https://www.kaggle.com/datasets/blurredmachine/fifa-2019-world-cup-dataset..

1.3 Preliminary Wrangling

```
In [1]: # import all packages and set plots to be embedded inline
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sb

        %matplotlib inline
```

Load in your dataset and describe its properties through the questions below. Try and motivate your exploration goals through this section.

```
In [2]: #load data set and view the structure
        df=pd.read_csv('FIFA_data.csv')
```

```
In [3]: #view dataframe
        df.head(5)
```

```
Out[3]:    Unnamed: 0      ID               Name  Age  \
        0           0  158023           L. Messi   31
        1           1   20801  Cristiano Ronaldo   33
        2           2  190871          Neymar Jr   26
        3           3  193080             De Gea   27
        4           4  192985       K. De Bruyne   27

                                                Photo Nationality  \
```

```
0  https://cdn.sofifa.org/players/4/19/158023.png   Argentina
1   https://cdn.sofifa.org/players/4/19/20801.png    Portugal
2  https://cdn.sofifa.org/players/4/19/190871.png      Brazil
3  https://cdn.sofifa.org/players/4/19/193080.png       Spain
4  https://cdn.sofifa.org/players/4/19/192985.png     Belgium

                                Flag  Overall  Potential  \
0  https://cdn.sofifa.org/flags/52.png       94         94
1  https://cdn.sofifa.org/flags/38.png       94         94
2  https://cdn.sofifa.org/flags/54.png       92         93
3  https://cdn.sofifa.org/flags/45.png       91         93
4   https://cdn.sofifa.org/flags/7.png       91         92

                  Club      ...       Composure Marking StandingTackle  \
0         FC Barcelona      ...            96.0    33.0           28.0
1             Juventus      ...            95.0    28.0           31.0
2  Paris Saint-Germain      ...            94.0    27.0           24.0
3    Manchester United      ...            68.0    15.0           21.0
4      Manchester City      ...            88.0    68.0           58.0

   SlidingTackle GKDiving  GKHandling  GKKicking  GKPositioning GKReflexes  \
0           26.0      6.0        11.0       15.0           14.0        8.0
1           23.0      7.0        11.0       15.0           14.0       11.0
2           33.0      9.0         9.0       15.0           15.0       11.0
3           13.0     90.0        85.0       87.0           88.0       94.0
4           51.0     15.0        13.0        5.0           10.0       13.0

  Release Clause
0        226.5M
1        127.1M
2        228.1M
3        138.6M
4        196.4M

[5 rows x 89 columns]
```

In [4]:
```
#data shape
df.shape
```

Out[4]:
```
(18206, 89)
```

In [5]:
```
#data information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18206 entries, 0 to 18205
Data columns (total 89 columns):
Unnamed: 0                18206 non-null int64
ID                        18206 non-null int64
```

```
Name                      18206 non-null object
Age                       18206 non-null int64
Photo                     18206 non-null object
Nationality               18206 non-null object
Flag                      18206 non-null object
Overall                   18206 non-null int64
Potential                 18206 non-null int64
Club                      17965 non-null object
Club Logo                 18206 non-null object
Value                     18206 non-null object
Wage                      18206 non-null object
Special                   18206 non-null int64
Preferred Foot            18158 non-null object
International Reputation  18158 non-null float64
Weak Foot                 18158 non-null float64
Skill Moves               18158 non-null float64
Work Rate                 18158 non-null object
Body Type                 18158 non-null object
Real Face                 18158 non-null object
Position                  18146 non-null object
Jersey Number             18146 non-null float64
Joined                    16653 non-null object
Loaned From               1264 non-null object
Contract Valid Until      17917 non-null object
Height                    18158 non-null object
Weight                    18158 non-null object
LS                        16121 non-null object
ST                        16121 non-null object
RS                        16121 non-null object
LW                        16121 non-null object
LF                        16121 non-null object
CF                        16121 non-null object
RF                        16121 non-null object
RW                        16121 non-null object
LAM                       16121 non-null object
CAM                       16121 non-null object
RAM                       16121 non-null object
LM                        16121 non-null object
LCM                       16121 non-null object
CM                        16121 non-null object
RCM                       16121 non-null object
RM                        16121 non-null object
LWB                       16121 non-null object
LDM                       16121 non-null object
CDM                       16121 non-null object
RDM                       16121 non-null object
RWB                       16121 non-null object
LB                        16121 non-null object
```

```
LCB                       16121 non-null object
CB                        16121 non-null object
RCB                       16121 non-null object
RB                        16121 non-null object
Crossing                  18158 non-null float64
Finishing                 18158 non-null float64
HeadingAccuracy           18158 non-null float64
ShortPassing              18158 non-null float64
Volleys                   18158 non-null float64
Dribbling                 18158 non-null float64
Curve                     18158 non-null float64
FKAccuracy                18158 non-null float64
LongPassing               18158 non-null float64
BallControl               18158 non-null float64
Acceleration              18158 non-null float64
SprintSpeed               18158 non-null float64
Agility                   18158 non-null float64
Reactions                 18158 non-null float64
Balance                   18158 non-null float64
ShotPower                 18158 non-null float64
Jumping                   18158 non-null float64
Stamina                   18158 non-null float64
Strength                  18158 non-null float64
LongShots                 18158 non-null float64
Aggression                18158 non-null float64
Interceptions             18158 non-null float64
Positioning               18158 non-null float64
Vision                    18158 non-null float64
Penalties                 18158 non-null float64
Composure                 18158 non-null float64
Marking                   18158 non-null float64
StandingTackle            18158 non-null float64
SlidingTackle             18158 non-null float64
GKDiving                  18158 non-null float64
GKHandling                18158 non-null float64
GKKicking                 18158 non-null float64
GKPositioning             18158 non-null float64
GKReflexes                18158 non-null float64
Release Clause            16642 non-null object
dtypes: float64(38), int64(6), object(45)
memory usage: 12.4+ MB
```

1.3.1 Wrangling

I will narrow down this datasets to the variables of interest for this project.

In [6]:
```
#target columns
t_columns= ['ID','Name','Value','Age','Nationality','Overall','Potential','Club','Wage',
soccer_df=df[t_columns]
```

In [7]:
```
#View the target df
soccer_df.head()
```

Out[7]:

	ID	Name	Value	Age	Nationality	Overall	Potential	\
0	158023	L. Messi	110.5M	31	Argentina	94	94	
1	20801	Cristiano Ronaldo	77M	33	Portugal	94	94	
2	190871	Neymar Jr	118.5M	26	Brazil	92	93	
3	193080	De Gea	72M	27	Spain	91	93	
4	192985	K. De Bruyne	102M	27	Belgium	91	92	

	Club	Wage	Preferred Foot	International Reputation	\
0	FC Barcelona	565K	Left	5.0	
1	Juventus	405K	Right	5.0	
2	Paris Saint-Germain	290K	Right	5.0	
3	Manchester United	260K	Right	4.0	
4	Manchester City	355K	Right	4.0	

	Work Rate	Body Type	Position	Dribbling	BallControl	Height	Weight
0	Medium/ Medium	Messi	RF	97.0	96.0	5'7	159lbs
1	High/ Low	C. Ronaldo	ST	88.0	94.0	6'2	183lbs
2	High/ Medium	Neymar	LW	96.0	95.0	5'9	150lbs
3	Medium/ Medium	Lean	GK	18.0	42.0	6'4	168lbs
4	High/ High	Normal	RCM	86.0	91.0	5'11	154lbs

In [8]:
```
# target df information
soccer_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18206 entries, 0 to 18205
Data columns (total 18 columns):
ID                          18206 non-null int64
Name                        18206 non-null object
Value                       18206 non-null object
Age                         18206 non-null int64
Nationality                 18206 non-null object
Overall                     18206 non-null int64
Potential                   18206 non-null int64
Club                        17965 non-null object
Wage                        18206 non-null object
Preferred Foot              18158 non-null object
International Reputation    18158 non-null float64
Work Rate                   18158 non-null object
Body Type                   18158 non-null object
Position                    18146 non-null object
Dribbling                   18158 non-null float64
```

```
BallControl                18158 non-null float64
Height                     18158 non-null object
Weight                     18158 non-null object
dtypes: float64(3), int64(4), object(11)
memory usage: 2.5+ MB
```

```
In [9]: #Replace columns spaces with underscore and lowercase for the headings
        soccer_df= soccer_df.rename(columns=str.lower)
        soccer_df.columns = soccer_df.columns.str.replace(' ','_')
```

```
In [10]: #validate action above
         soccer_df.head()
```

Out[10]:

	id	name	value	age	nationality	overall	potential	club	wage	preferred_foot	international_reputation	work_rate	body_type	position	dribbling	ballcontrol	height	weight
0	158023	L. Messi	110.5M	31	Argentina	94	94	FC Barcelona	565K	Left	5.0	Medium/ Medium	Messi	RF	97.0	96.0	5'7	159lbs
1	20801	Cristiano Ronaldo	77M	33	Portugal	94	94	Juventus	405K	Right	5.0	High/ Low	C. Ronaldo	ST	88.0	94.0	6'2	183lbs
2	190871	Neymar Jr	118.5M	26	Brazil	92	93	Paris Saint-Germain	290K	Right	5.0	High/ Medium	Neymar	LW	96.0	95.0	5'9	150lbs
3	193080	De Gea	72M	27	Spain	91	93	Manchester United	260K	Right	4.0	Medium/ Medium	Lean	GK	18.0	42.0	6'4	168lbs
4	192985	K. De Bruyne	102M	27	Belgium	91	92	Manchester City	355K	Right	4.0	High/ High	Normal	RCM	86.0	91.0	5'11	154lbs

```
In [11]: # Check for Null values in Club column
         pd.isnull(soccer_df["club"]).sum()
```

```
Out[11]: 241
```

```
In [12]: #Drop Null values above
         soccer_df.dropna(inplace=True)
```

```
In [13]: # Validate action above
         soccer_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17917 entries, 0 to 18205
Data columns (total 18 columns):
```

```
id                          17917 non-null int64
name                        17917 non-null object
value                       17917 non-null object
age                         17917 non-null int64
nationality                 17917 non-null object
overall                     17917 non-null int64
potential                   17917 non-null int64
club                        17917 non-null object
wage                        17917 non-null object
preferred_foot              17917 non-null object
international_reputation    17917 non-null float64
work_rate                   17917 non-null object
body_type                   17917 non-null object
position                    17917 non-null object
dribbling                   17917 non-null float64
ballcontrol                 17917 non-null float64
height                      17917 non-null object
weight                      17917 non-null object
dtypes: float64(3), int64(4), object(11)
memory usage: 2.6+ MB
```

In [14]:
```
# Remove currency, M and K symbol from value and wage
def parse_money(s):
    if s.startswith(''):
        s = s[1:]
    multiplier = None
    if s.endswith('M'):
        s = s[:-1]
        multiplier = 1e6
    elif s.endswith('K'):
        s = s[:-1]
        multiplier = 1e3
    f = float(s)
    if multiplier:
        f = f * multiplier
    return f


soccer_df['value'] = soccer_df['value'].apply(parse_money)
soccer_df['wage'] = soccer_df['wage'].apply(parse_money)
soccer_df.head()
```

Out[14]:

	id	name	value	age	nationality	overall	\
0	158023	L. Messi	110500000.0	31	Argentina	94	
1	20801	Cristiano Ronaldo	77000000.0	33	Portugal	94	
2	190871	Neymar Jr	118500000.0	26	Brazil	92	
3	193080	De Gea	72000000.0	27	Spain	91	

```
4  192985       K. De Bruyne  102000000.0   27     Belgium       91

   potential                 club      wage preferred_foot  \
0         94         FC Barcelona  565000.0           Left
1         94             Juventus  405000.0          Right
2         93  Paris Saint-Germain  290000.0          Right
3         93    Manchester United  260000.0          Right
4         92      Manchester City  355000.0          Right

   international_reputation       work_rate   body_type position  dribbling  \
0                       5.0  Medium/ Medium       Messi       RF       97.0
1                       5.0       High/ Low  C. Ronaldo       ST       88.0
2                       5.0    High/ Medium      Neymar       LW       96.0
3                       4.0  Medium/ Medium        Lean       GK       18.0
4                       4.0      High/ High      Normal      RCM       86.0

   ballcontrol height  weight
0         96.0    5'7  159lbs
1         94.0    6'2  183lbs
2         95.0    5'9  150lbs
3         42.0    6'4  168lbs
4         91.0   5'11  154lbs
```

```
In [15]: #Remove zero rows in value column
         soccer_df= soccer_df[soccer_df.value != 0]
```

```
In [16]: #check unique values in body type
         soccer_df['body_type'].unique()
```

```
Out[16]: array(['Messi', 'C. Ronaldo', 'Neymar', 'Lean', 'Normal', 'Courtois',
                'Stocky', 'PLAYER_BODY_TYPE_25', 'Shaqiri', 'Akinfenwa'], dtype=object)
```

```
In [17]: # remove player names in body type column
         soccer_df = soccer_df[soccer_df['body_type'] != 'Messi']
         soccer_df = soccer_df[soccer_df['body_type'] != 'C. Ronaldo']
         soccer_df = soccer_df[soccer_df['body_type'] != 'Neymar']
         soccer_df = soccer_df[soccer_df['body_type'] != 'Courtois']
         soccer_df = soccer_df[soccer_df['body_type'] != 'PLAYER_BODY_TYPE_25']
         soccer_df = soccer_df[soccer_df['body_type'] != 'Shaqiri']
         soccer_df = soccer_df[soccer_df['body_type'] != 'Akinfenwa']
```

```
In [18]: #Validate action above
         soccer_df['body_type'].unique()
```

```
Out[18]: array(['Lean', 'Normal', 'Stocky'], dtype=object)
```

Lets clean up weight and height. Also I will change all the numerical columns from float to integer, change weight and height from objects to integer and float respectively.I will convert position,'body type', and 'work rate' to category datatype

```
In [19]: #lets remove lbs from weight
         soccer_df['weight'] = soccer_df['weight'].str.replace('lbs', '')

In [20]: # remove ' replace with . in height column
         soccer_df['height'] = soccer_df['height'].str.replace("'", '.')

In [21]: #Change datatypes
         soccer_df['wage'] = soccer_df['wage'].astype(int)
         soccer_df['value'] = soccer_df['value'].astype(int)
         soccer_df['dribbling'] = soccer_df['dribbling'].astype(int)
         soccer_df['ballcontrol'] = soccer_df['ballcontrol'].astype(int)
         soccer_df['height'] = soccer_df['height'].astype(float)
         soccer_df['weight'] = soccer_df['weight'].astype(int)

         # convert body type,position, and work rate to category datatype
         soccer_df.body_type = soccer_df.body_type.astype("category")
         soccer_df.work_rate = soccer_df.work_rate.astype("category")

In [22]: #validate actions above
         soccer_df.head(5)

Out[22]:       id          name      value  age nationality  overall  potential  \
         3  193080        De Gea   72000000   27       Spain       91         93
         4  192985  K. De Bruyne  102000000   27     Belgium       91         92
         5  183277     E. Hazard   93000000   27     Belgium       91         91
         6  177003     L. Modri    67000000   32     Croatia       91         91
         7  176580     L. Suárez   80000000   31     Uruguay       91         91

                        club    wage preferred_foot  international_reputation  \
         3  Manchester United  260000          Right                       4.0
         4    Manchester City  355000          Right                       4.0
         5            Chelsea  340000          Right                       4.0
         6        Real Madrid  420000          Right                       4.0
         7       FC Barcelona  455000          Right                       5.0

                 work_rate body_type position  dribbling  ballcontrol  height  weight
         3  Medium/ Medium      Lean       GK         18           42    6.40     168
         4      High/ High    Normal      RCM         86           91    5.11     154
         5    High/ Medium    Normal       LF         95           94    5.80     163
         6      High/ High      Lean      RCM         90           93    5.80     146
         7    High/ Medium    Normal       RS         87           90    6.00     190

In [23]: soccer_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 17899 entries, 3 to 18205
Data columns (total 18 columns):
id                        17899 non-null int64
name                      17899 non-null object
```

```
value                       17899 non-null int64
age                         17899 non-null int64
nationality                 17899 non-null object
overall                     17899 non-null int64
potential                   17899 non-null int64
club                        17899 non-null object
wage                        17899 non-null int64
preferred_foot              17899 non-null object
international_reputation    17899 non-null float64
work_rate                   17899 non-null category
body_type                   17899 non-null category
position                    17899 non-null object
dribbling                   17899 non-null int64
ballcontrol                 17899 non-null int64
height                      17899 non-null float64
weight                      17899 non-null int64
dtypes: category(2), float64(2), int64(9), object(5)
memory usage: 2.4+ MB
```

1.3.2 What is the structure of your dataset?

The data contains 89 attributes(columns) and 18206 rows.After cleaning the data and removing columns not important for my use in this project, I now have a dataset with 17899 rows and 18 columns.

1.3.3 What is/are the main feature(s) of interest in your dataset?

For this project, the target variable is the the value of the players.My interest is to find out how the variables affects the value of the players at the end of the tournament.

1.3.4 What features in the dataset do you think will help support your investigation into your feature(s) of interest?

For the purpose of this project, I will be looking at how these variables affects the value of the players:I will pay attention to these variables:Age,Nationality,Overall,Potential,Club,Value,Wage,Preferred Foot,International Reputation,Work Rate,Body Type,Positiondribbling,ballcontrol,weight,and height.

```
In [24]: # Lets look at basic statistics of the Numerical variables
         soccer_df.describe()
```

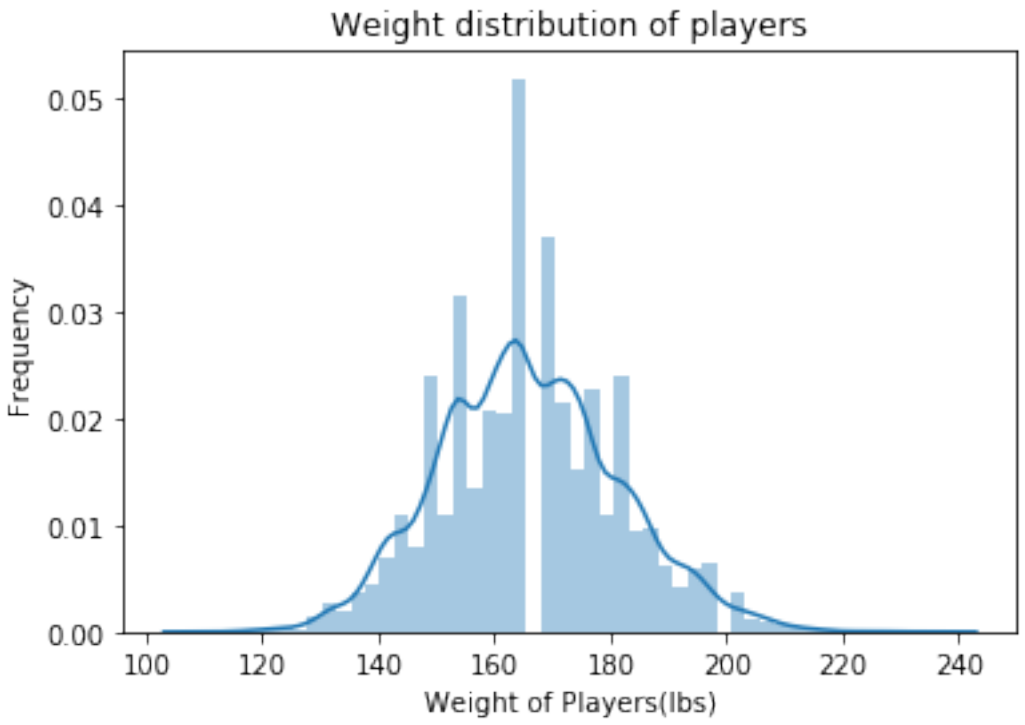
```
Out[24]:                   id         value           age       overall     potential  \
         count   17899.000000  1.789900e+04  17899.000000  17899.000000  17899.000000
         mean   214306.337226  2.426182e+06     25.094530     66.232862     71.329516
         std     29848.929035  5.442303e+06      4.659618      6.912926      6.128109
         min        16.000000  1.000000e+04     16.000000     47.000000     48.000000
         25%    200276.500000  3.250000e+05     21.000000     62.000000     67.000000
         50%    221714.000000  7.000000e+05     25.000000     66.000000     71.000000
```

```
75%    236516.500000  2.100000e+06     28.000000     71.000000     75.000000
max    246620.000000  1.020000e+08     45.000000     91.000000     95.000000

                wage  international_reputation     dribbling   ballcontrol  \
count   17899.000000              17899.000000  17899.000000  17899.000000
mean     9786.691994                  1.112688     55.413543     58.408459
std     21291.299963                  0.390960     18.890807     16.653685
min      1000.000000                  1.000000      4.000000      5.000000
25%      1000.000000                  1.000000     49.000000     54.000000
50%      3000.000000                  1.000000     61.000000     63.000000
75%      9000.000000                  1.000000     68.000000     69.000000
max    455000.000000                  5.000000     95.000000     95.000000

             height        weight
count  17899.000000  17899.000000
mean       5.797012    165.958098
std        0.448376     15.590111
min        5.100000    110.000000
25%        5.110000    154.000000
50%        5.900000    165.000000
75%        6.100000    176.000000
max        6.900000    236.000000
```

Observation from Statistics above 1. Oldest player in the dataset is 45years and the youngest player is 16 years,and 75% of the players are below 29 years 1. The most expensive player rated at a value of €102M, and the least expensive is rated at €10,000 1. The average wage of the players is €3,000 , while the highest earn player receives about €455,000.

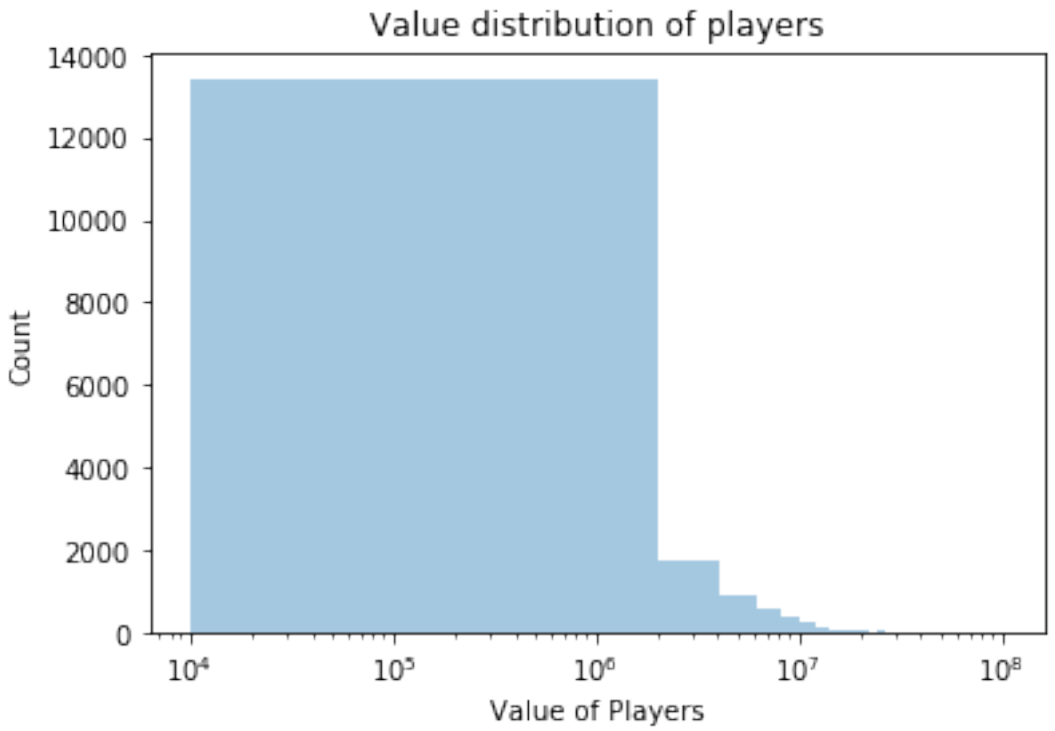
1.4 Univariate Exploration.

```
In [25]: #histogram plot of Weight
         sb.distplot(soccer_df['weight'])
         # Add labels
         plt.title('Weight distribution of players')
         plt.xlabel('Weight of Players(lbs)')
         plt.ylabel('Frequency');
```



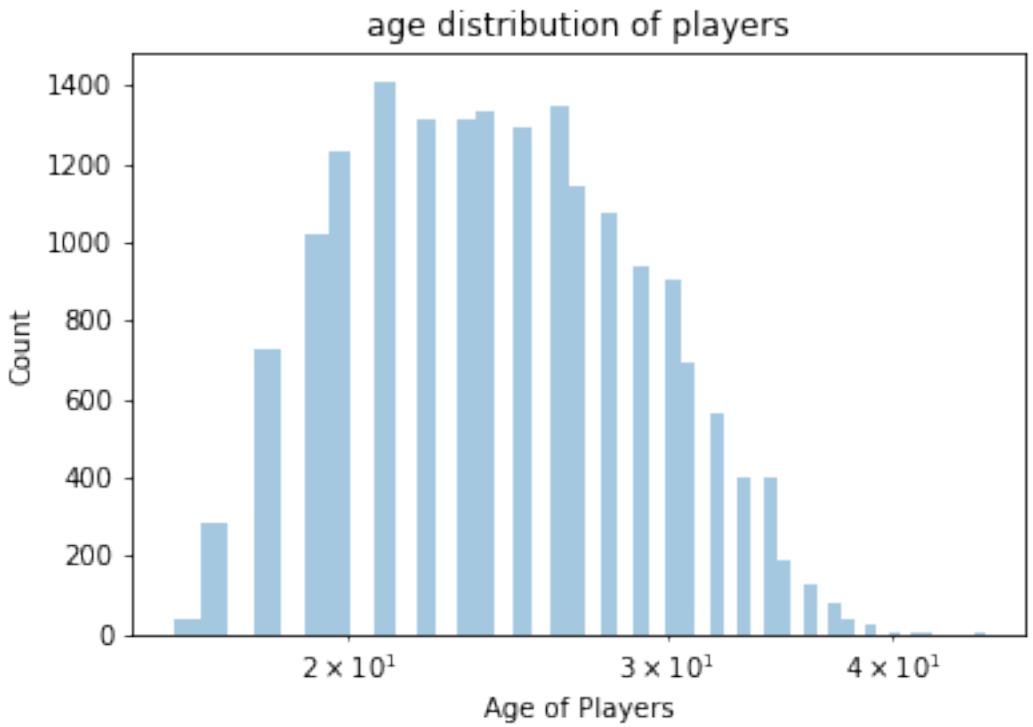
The weight distribution above is slightly normal.Most of the players are betweeen 140lbs to 200lbs.

```
In [26]: #histogram plot of Value
         sb.distplot(soccer_df['value'],kde = False)
         plt.xscale('log')
         # Add labels
         plt.title('Value distribution of players')
         plt.xlabel('Value of Players')
         plt.ylabel('Count');
```



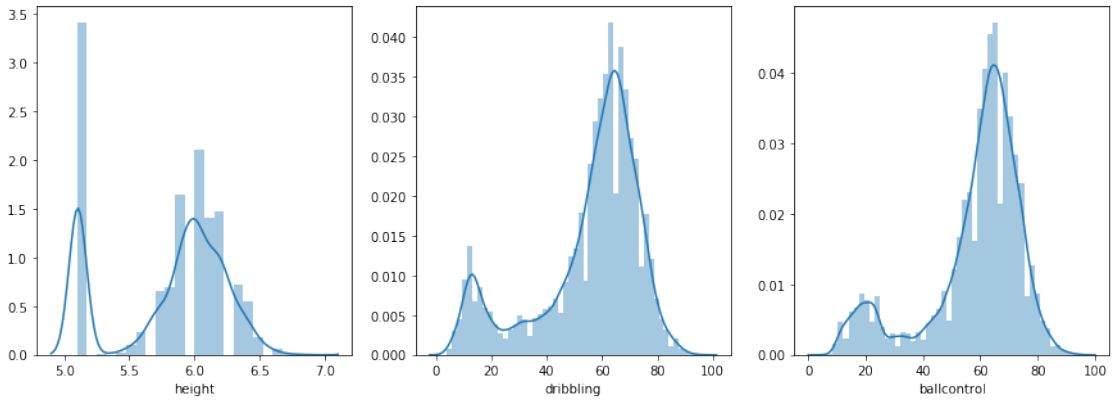
The value distribution is bimodal, skewed to the right, with most player valued between €10,000 and €1,000,000

```
In [27]: sb.distplot(soccer_df['age'],kde = False)
         plt.xscale('log')
         # Add labels
         plt.title('age distribution of players')
         plt.xlabel('Age of Players')
         plt.ylabel('Count');
```



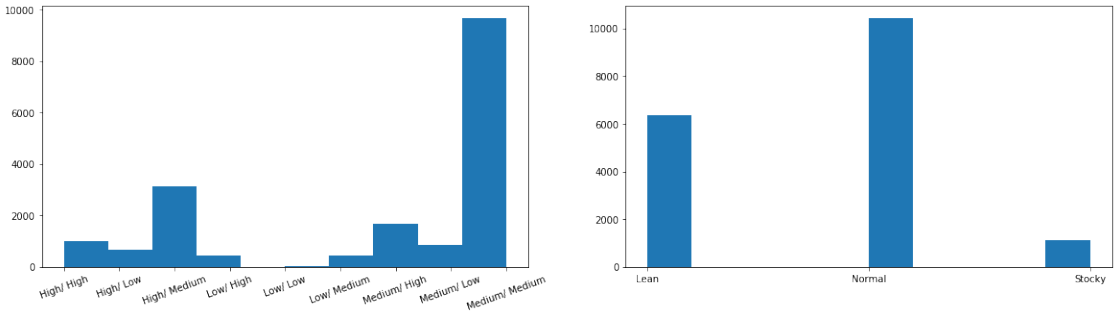
From the age distribution above, the age of most players are in the 20s which is as observed from the statistical information earlier on.

```
In [28]: #univariate plots of height, dribbling, and ballcontrol grades
         plt.figure(figsize=[15,5])

         #subplot 1
         plt.subplot(1,3,1)
         sb.distplot(soccer_df['height'])

         #subplot2
         plt.subplot(1,3,2)
         sb.distplot(soccer_df['dribbling'])

         #subplot 3
         plt.subplot(1,3,3)
         sb.distplot(soccer_df['ballcontrol']);
```



The height, dribbling and ballcontrol distributions are bimodal.The dribbling and ballcontrol follows a similar trend in their distribution,I think there might be an interesting relationship between these two.

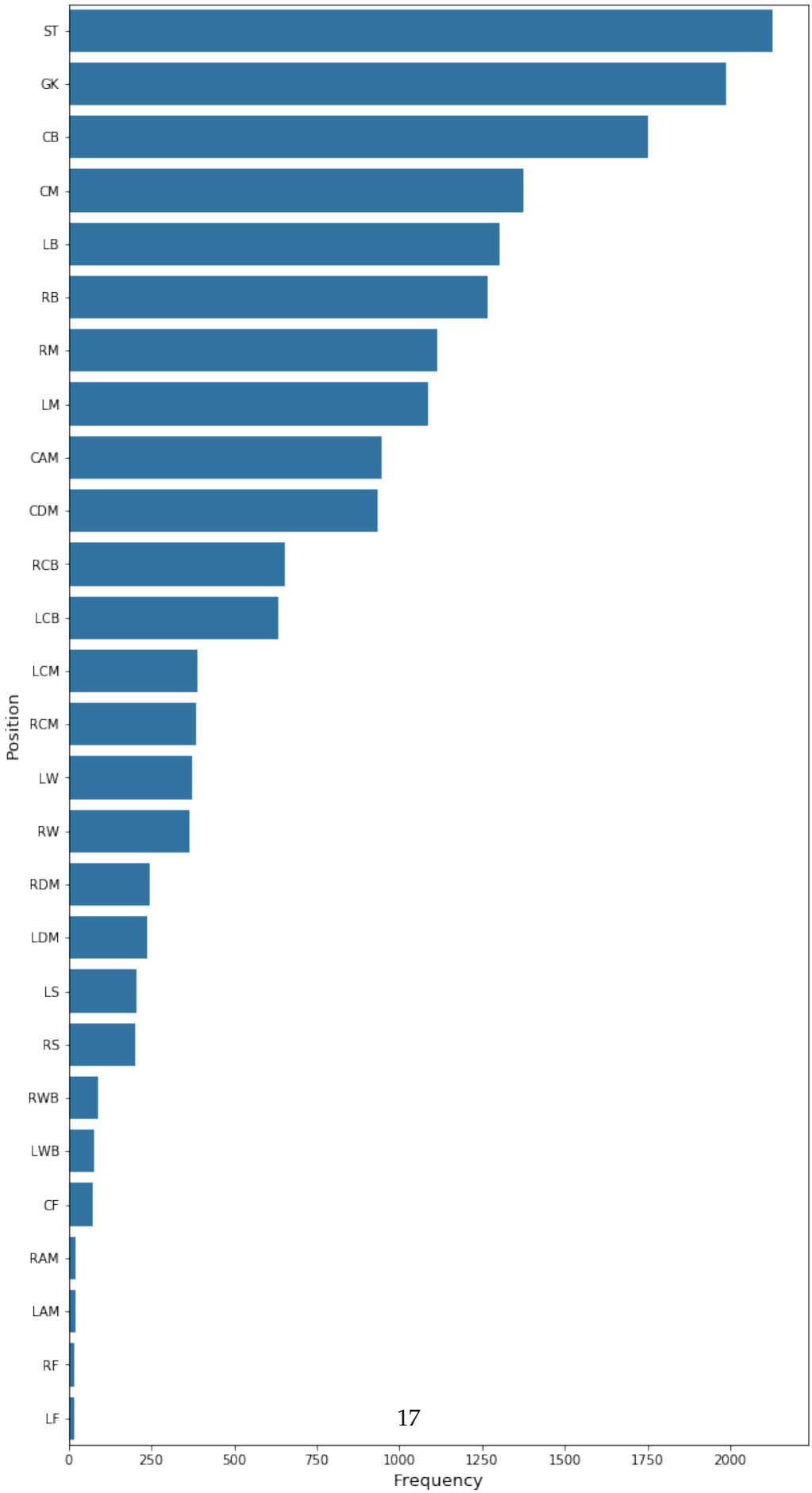
```
In [29]: #univariate plots of work rate, body type, and position
         plt.figure(figsize=[20,5])

         #subplot 1
         plt.subplot(1,2,1)
         plt.hist(data=soccer_df, x='work_rate');
         plt.xticks(rotation=20)

         #subplot2
         plt.subplot(1,2,2)
         plt.hist(data=soccer_df, x='body_type');
```



More of the players are of normal body type.Most of players work rate are medium/medium, followed by high/medium and the few only have a work rate of low/low.

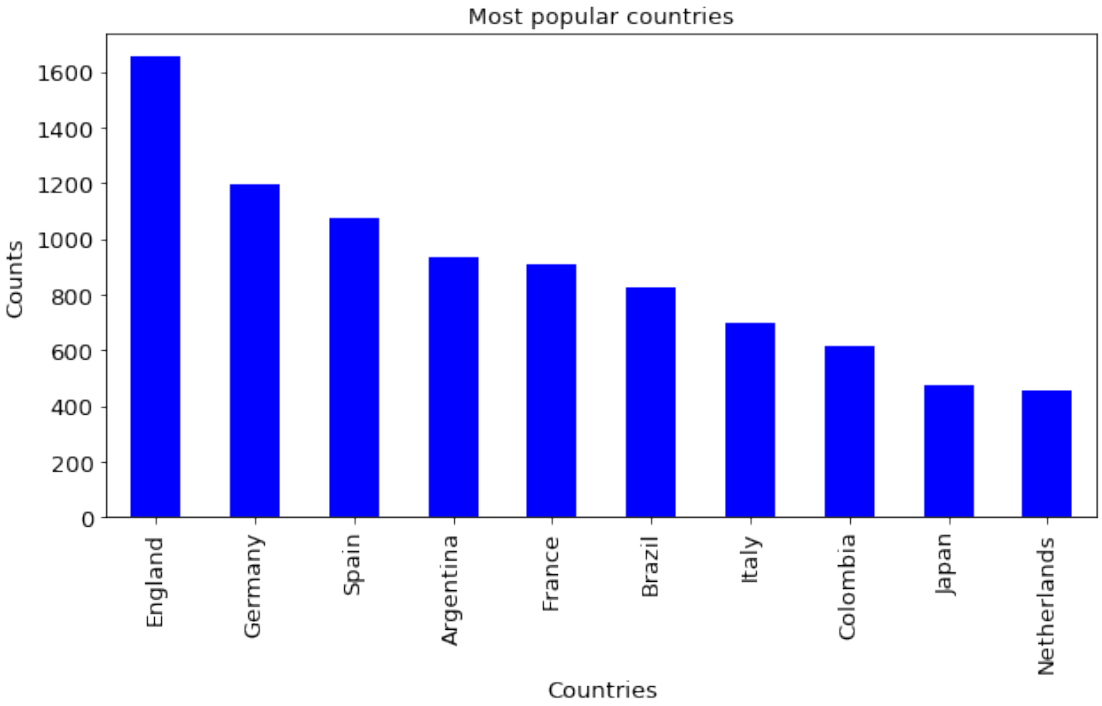
```
In [30]: position_count = soccer_df['position'].value_counts()
         position_order = position_count.index
```

In [31]:
```
#Bar chart of Positions of players
plt.figure(figsize=[10,20])
base_color = sb.color_palette()[0]
sb.countplot(data = soccer_df, y = 'position', color = base_color, order = position_cou
plt.ylabel('Position', color = 'black', fontsize = '13')
plt.xlabel('Frequency', color = 'black', fontsize = '13');
```


ST
GK
CB
CM
LB
RB
RM
LM
CAM
CDM
RCB
LCB
LCM
RCM
LW
RW
RDM
LDM
LS
RS
RWB
LWB
CF
RAM
LAM
RF
LF
Position
0
250
500
750
1000
1250
1500
1750
2000
Frequency

Most of the players in this dataset are Strikers(ST), followed by Goal Keepers(GK),Centre backs(CB), Central Mildfileders(CM) and Left Backs(LB).

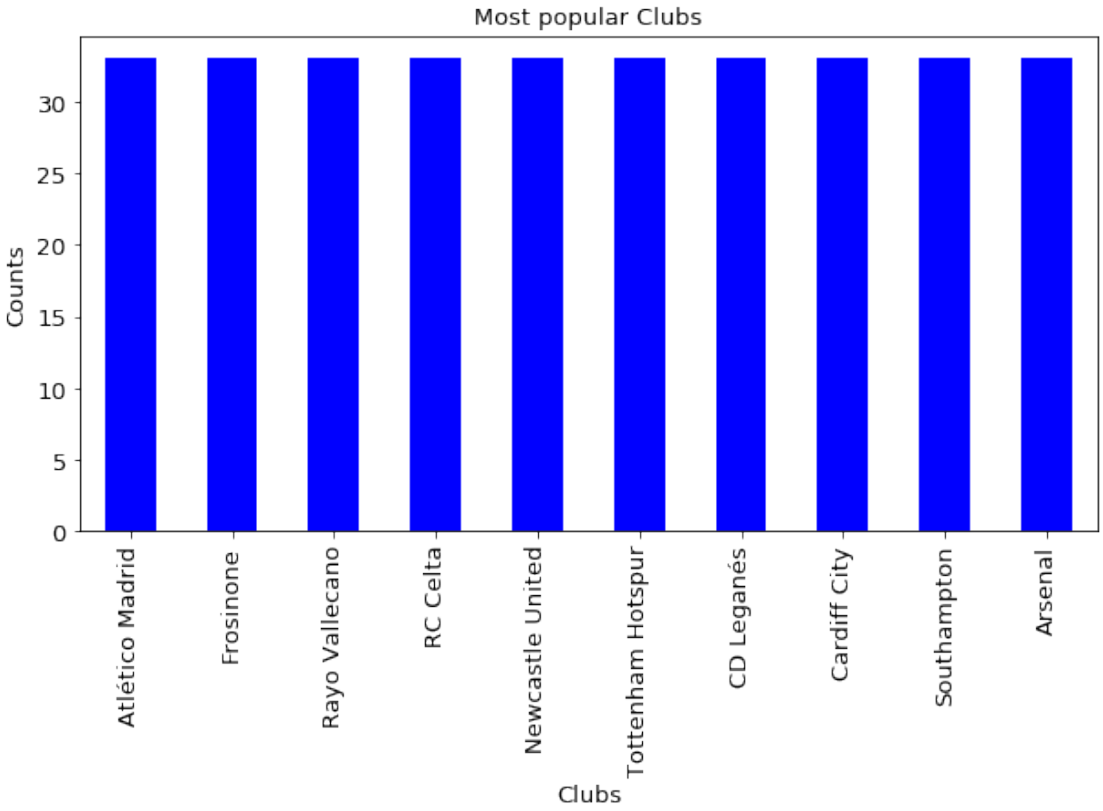
```
In [32]: #Top 10 nations of the players
         top10_nations = soccer_df['nationality'].value_counts().head(10)

In [33]: #Plot the bar chart of most popular nations in the tournament
         plt.figure(figsize=[10,5])
         top10_nations.plot.bar(color = 'blue', fontsize = 13)
         #Add labels
         plt.title('Most popular countries', color = 'black', fontsize = '13')
         plt.xlabel('Countries', color = 'black', fontsize = '13')
         plt.ylabel('Counts', color = 'black', fontsize = '13');
```



Interesting! England is the top footballing nation in the data.Most of the players in the data are from Europe, followed by South America, then Asia.

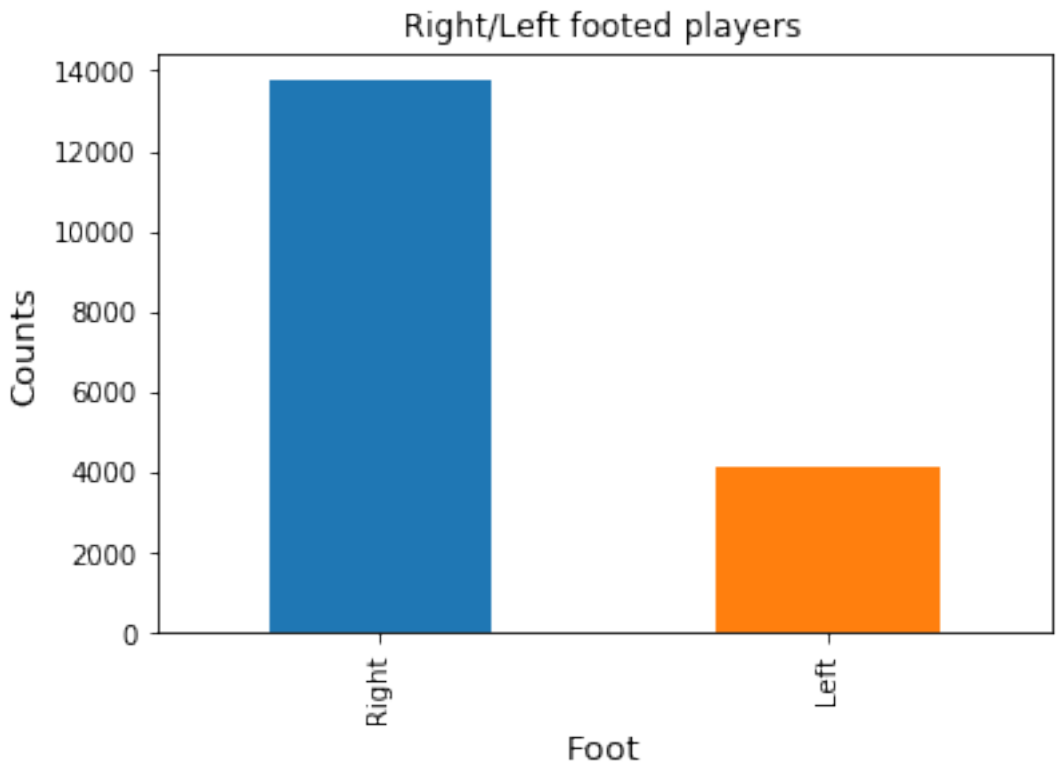
```
In [34]: #Top 10 clubs of the players
         top10_clubs = soccer_df['club'].value_counts().head(10)
         #Plot the bar chart of most popularClubs in the tournament
         plt.figure(figsize=[10,5])
         top10_clubs.plot.bar(color = 'blue', fontsize = 13)
         #Add labels
         plt.title('Most popular Clubs', color = 'black', fontsize = '13')
```

```
plt.xlabel('Clubs', color = 'black', fontsize = '13')
plt.ylabel('Counts', color = 'black', fontsize = '13');
```



Most of the popular clubs of players in the tournament is from Europe, and most are from England.

```
In [35]: #Preferred Foot of players
         soccer_df['preferred_foot'].value_counts().plot.bar(title="Right/Left footed players")
         plt.ylabel('Counts', color = 'black', fontsize = '13')
         plt.xlabel('Foot', color = 'black', fontsize = '13');
```



Most of the players in the data are Right footed.

1.4.1 Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

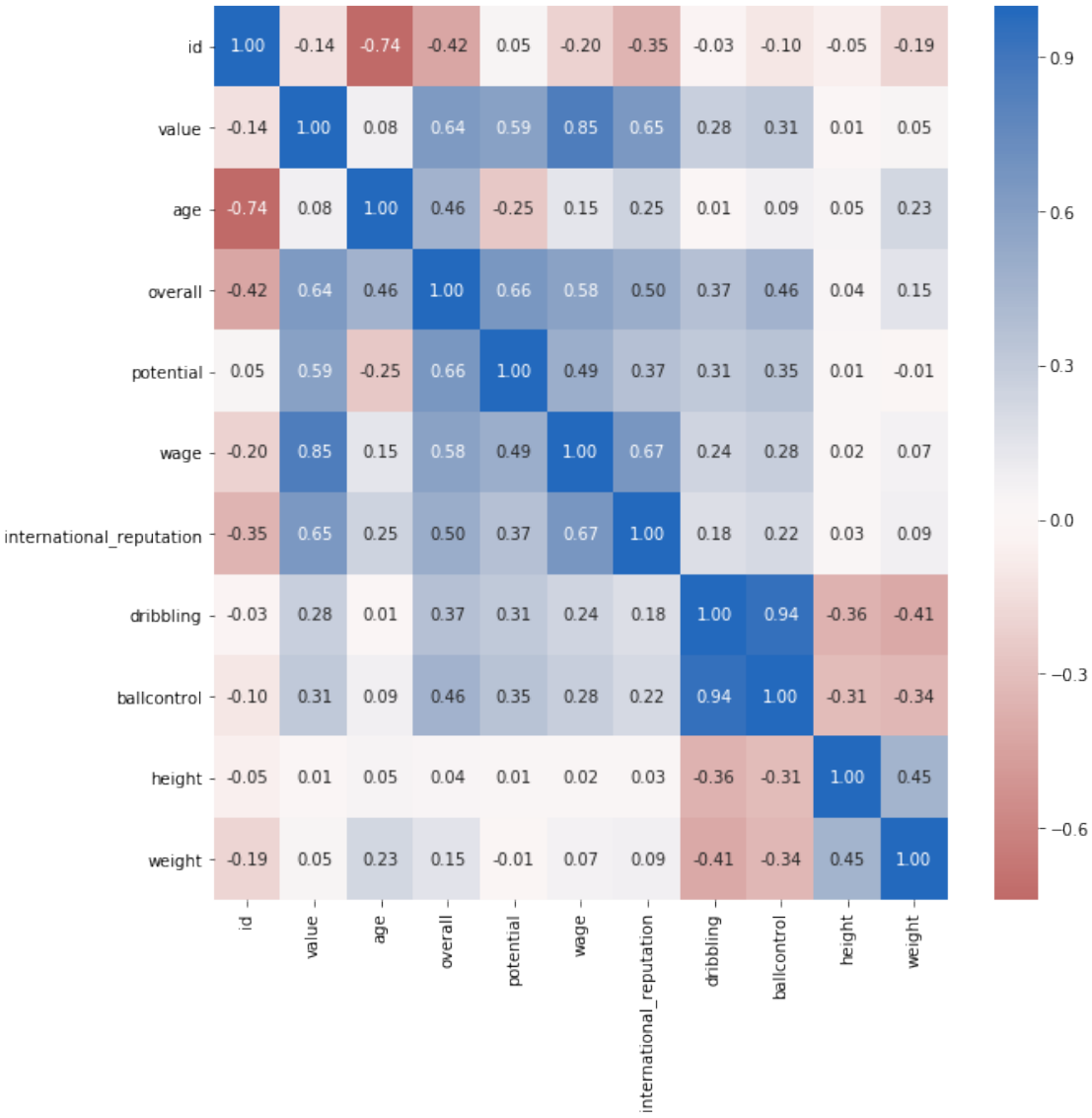
I started by looking at the histograms of the numerical variables, then I looked at the distributions of the qualitative variables.I noticed that the height, dribbling and ballcontrol distributions are bimodal.I also noticed a similarity between the distributions of dribbling and ballcontrol.I also observed that most players in the dataset are in 20s of age, most of the players are also strikers, and most players are right footed.

1.4.2 Of the features you investigated, were there any Unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

On plotting the chart for the positions, I noticed that the distribution is better oriented in the y axis because of the number of bars, and I have to increase the figure size on the y axis to make my plots more clear.

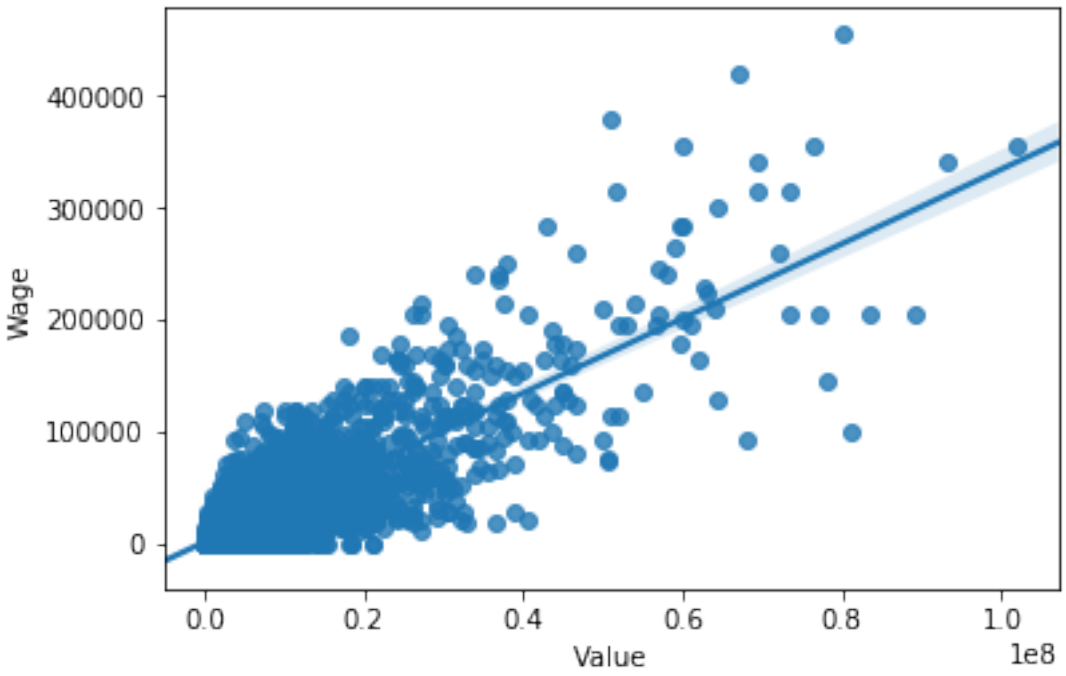
1.5 Bivariate Exploration

```
In [52]: #Correlation matrix for numerical variables
         plt.figure(figsize=[15,15])
         sb.heatmap(soccer_df.corr(),annot=True,fmt='.2f',cmap='vlag_r',center=0);
```



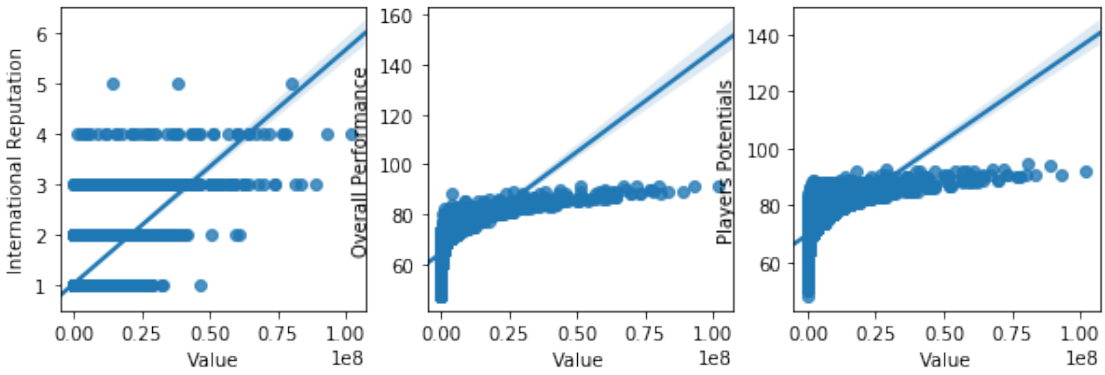
From the correlation heatmap above, there is strong positive reputation between value of players and their wages, there is moderate positive correlation between value and international reputation,value and overall perfomance of players, value and potentials of players.Also, there is a very strong positive correlation between ballcontrol and dribbling.

```
In [37]: #Correlation between value and wage
         sb.regplot(data = soccer_df, x='value', y='wage');
         plt.xlabel('Value')
         plt.ylabel('Wage');
```



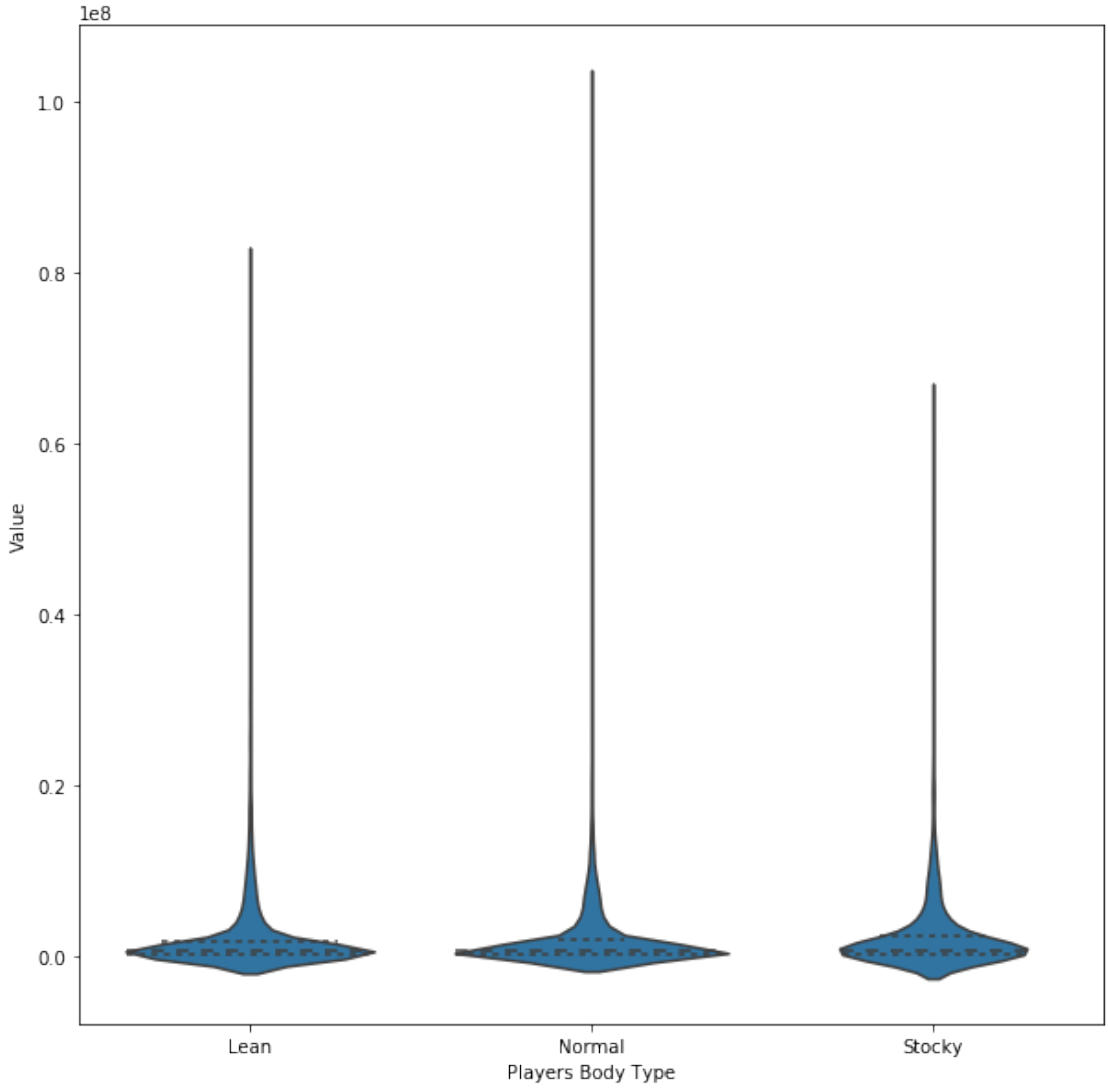
This show a strong positive correlation as seen from heatmap correlation

```
In [38]: #Correlation between value and wage
         plt.figure(figsize=[10,3])
         #subplot 1
         plt.subplot(1,3,1)
         sb.regplot(data = soccer_df, x='value', y='international_reputation');
         plt.xlabel('Value')
         plt.ylabel('International Reputation');

         #subplot 2
         plt.subplot(1,3,2)
         sb.regplot(data = soccer_df, x='value', y='overall');
         plt.xlabel('Value')
         plt.ylabel('Overall Performance');

         #subplot 3
         plt.subplot(1,3,3)
         sb.regplot(data = soccer_df, x='value', y='potential');
         plt.xlabel('Value')
         plt.ylabel('Players Potentials');
```



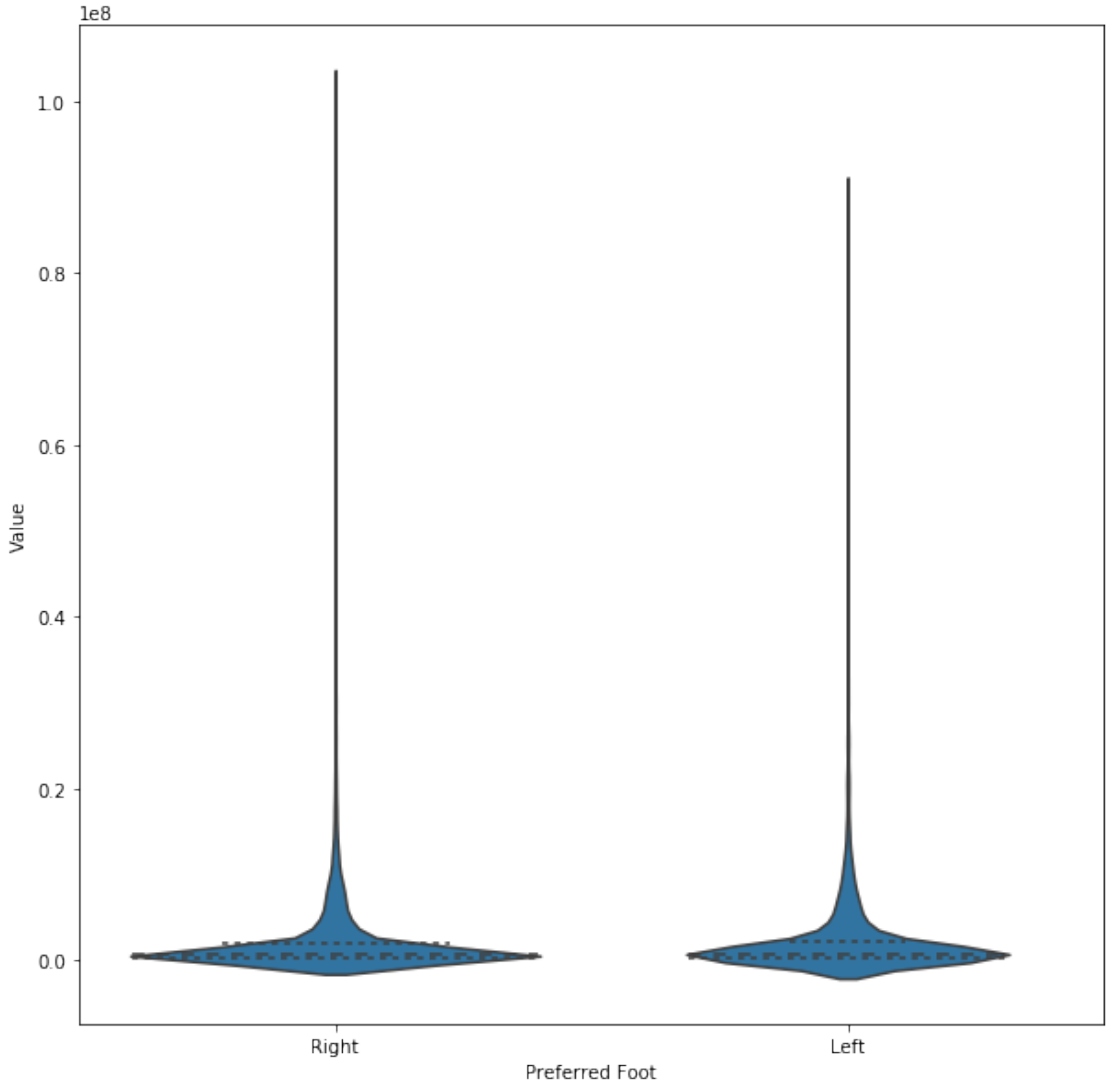
The above plots shows a moderate positive correlation between value of players and their international reputations,overall perfomances, and players potentials.

```
In [39]: # Violin plot of value vs body type
         plt.figure(figsize=[10,10])
         base_color = sb.color_palette()[0]
         sb.violinplot(data=soccer_df,x = 'body_type',y= 'value',color = base_color,
                       inner = 'quartile')
         plt.xlabel('Players Body Type')
         plt.ylabel('Value');
```



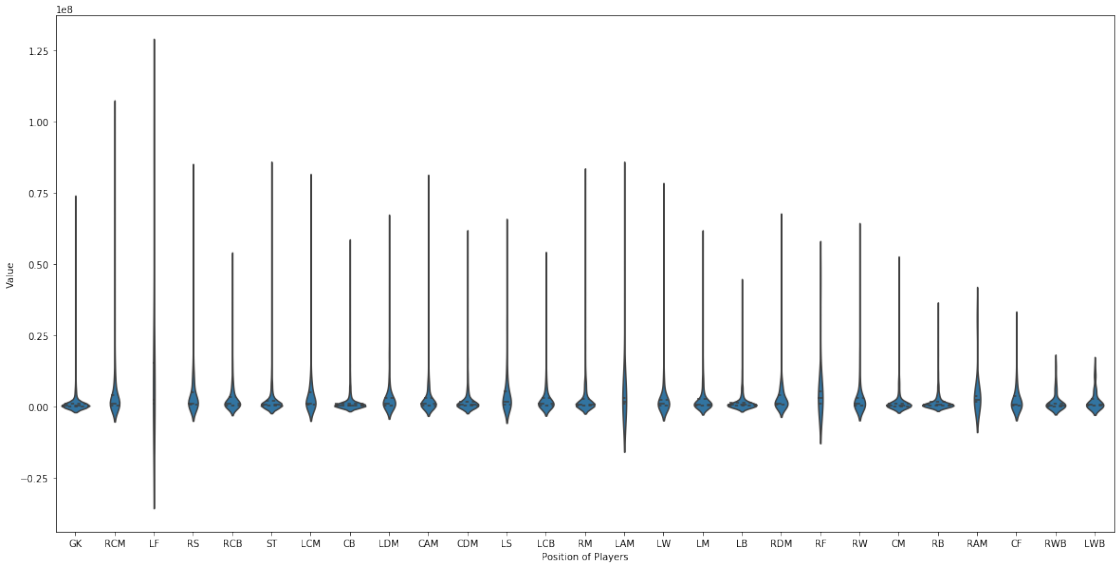
From above, we can see that more of players with normal body type are valued most.

```
In [40]: #Violin plot of value vs Preferred foot
         plt.figure(figsize=[10,10])
         base_color = sb.color_palette()[0]
         sb.violinplot(data=soccer_df,x = 'preferred_foot',y= 'value',color = base_color,
                       inner = 'quartile')
         plt.xlabel('Preferred Foot')
         plt.ylabel('Value');
```



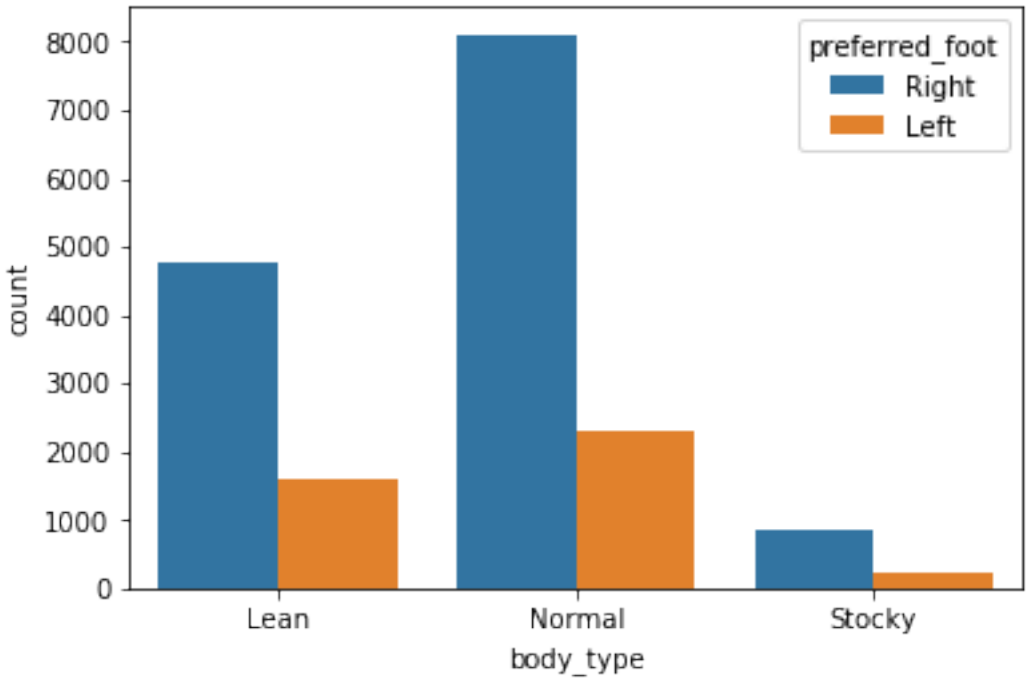
More of the Right foot players in the dataset are valued more.

```
In [41]: #Violin plot of value vs Position
         plt.figure(figsize=[20,10])
         base_color = sb.color_palette()[0]
         sb.violinplot(data=soccer_df,x = 'position',y= 'value',color = base_color,
                       inner = 'quartile')
         plt.xlabel('Position of Players')
         plt.ylabel('Value');
```



Left foot(LF) players are valued most, while the Left Wing Backs(LWB) and Right Wing Backs(RWB)are valued least from the dataset

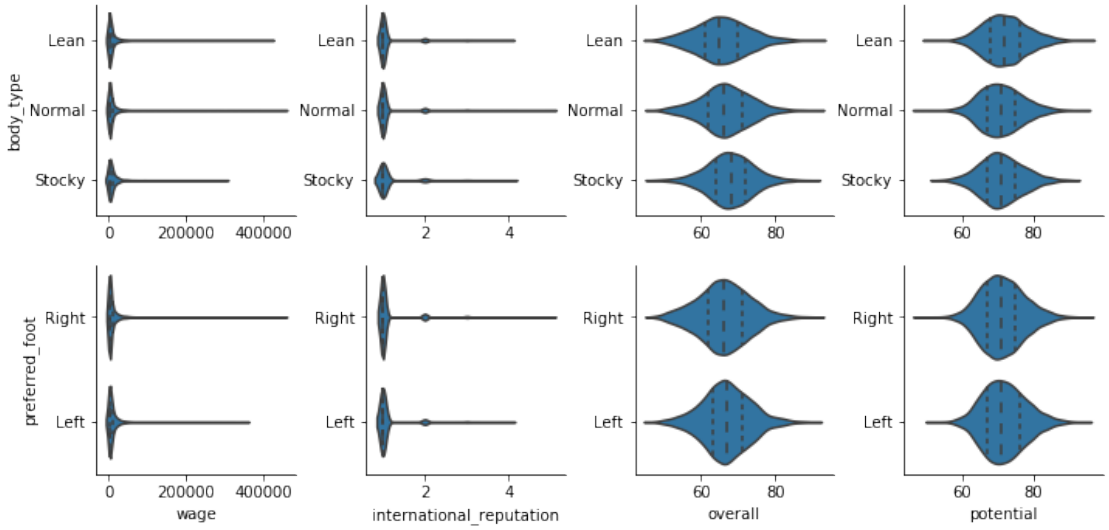
```
In [42]: #Clustered bar chart of body type and preferred foot
         sb.countplot(data=soccer_df,x='body_type',hue='preferred_foot');
```



From above,players with normal body type who are either left or right footed are more in the dataset.Most of the players with normal body type are right footed.

```
In [43]: #wage,international reputation versus body type and preferred foot
         plt.figure(figsize=[15,15]);
         g=sb.PairGrid(data=soccer_df,x_vars=['wage','international_reputation','overall','poten
         g.map(sb.violinplot,inner='quartile');
```

```
<matplotlib.figure.Figure at 0x7f433cc0ca58>
```



From above, players with normal body types received more wages, and has more international reputation.Those players that are right footed received more wages and also have more international reputation.There seems to be no different in overall rating of the players based on body type and preferred foot.Lean and normal body players have high potential than stocky body-type players.However,there is no recognisable difference between right-footed and left-footed players potentials.

1.5.1 Some of the relationships observed

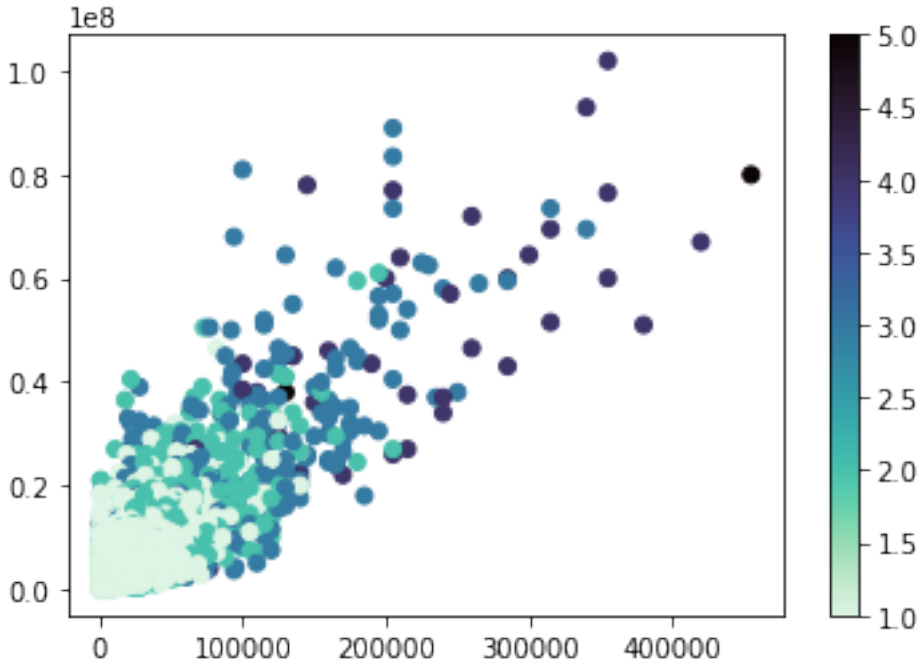
I observed that there is a positive relationship between value of players and Overall perfomance,potential of players,players wages,and international reputation. I also noticed a strong positive relationship between dribbling and ballcontrol, though both have low correlation with wage and value of players.Players with normal body type and are right foooted are valued most, and also these players receives more wages.

1.5.2 Observed interesting relationships between the other features (not the main feature(s) of interest)

I was surprise to observe that left foot positions(LF) are valued more than other positions.Also the Right Wing Back(RWB) and and Left Wing Back(LWB) positions are least valued.

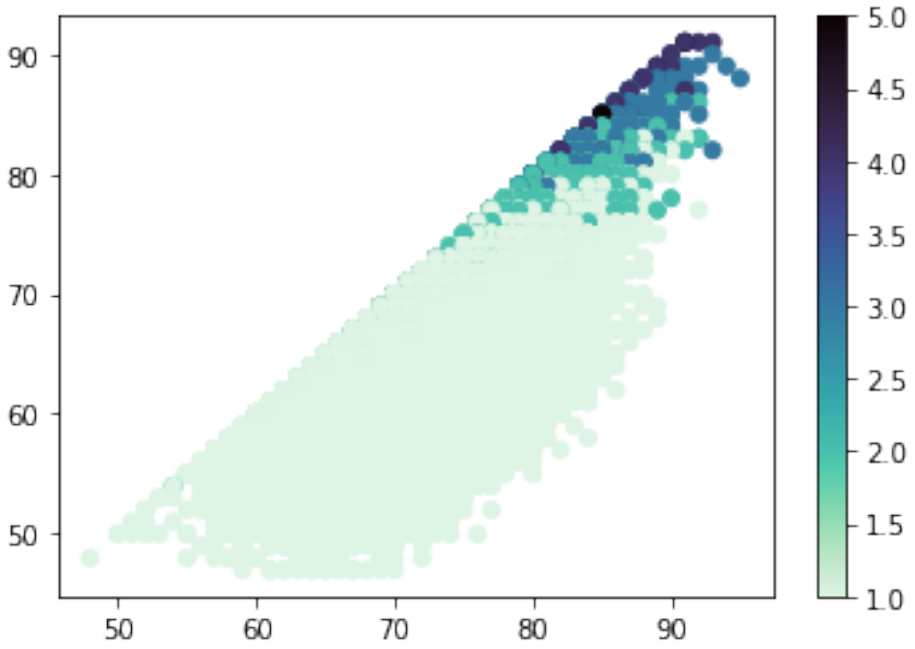
1.6 Multivariate Exploration

```
In [44]: #scatter plot of wage,value and international-reputation
         plt.scatter(data=soccer_df,x='wage',y='value',c='international_reputation',cmap='mako_r
         plt.colorbar();
         g.add_legend();
```



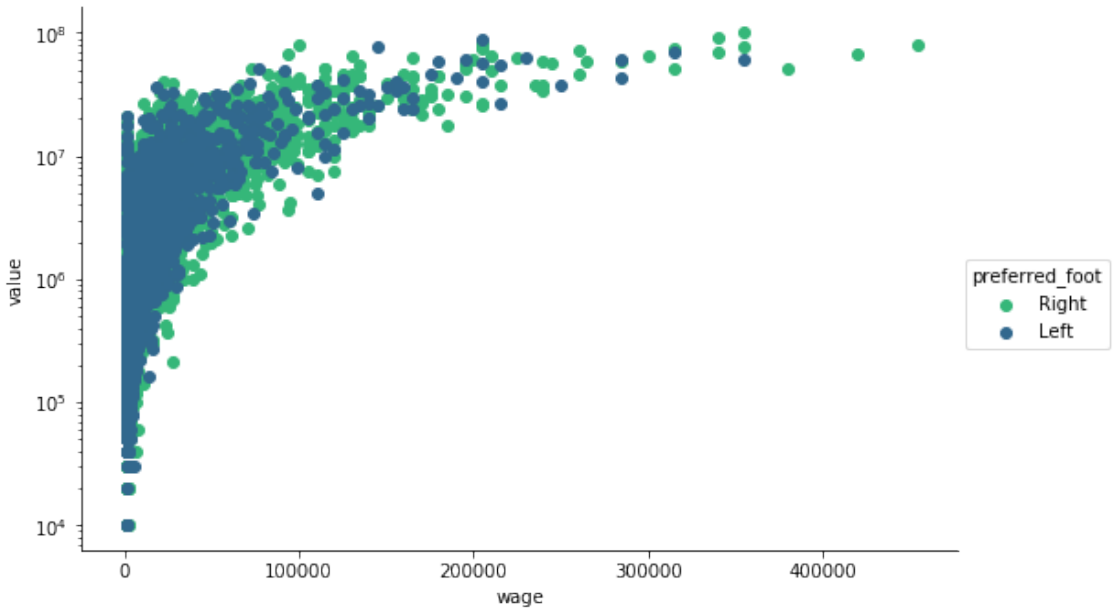
Plots above shows that players with high international reputation are more valued and also earns more.Moreover, more of the players in the tournament have low international reputation.

```
In [45]: plt.scatter(data=soccer_df,x='potential',y='overall',c='international_reputation',cmap=
         plt.colorbar();
         g.add_legend();
```



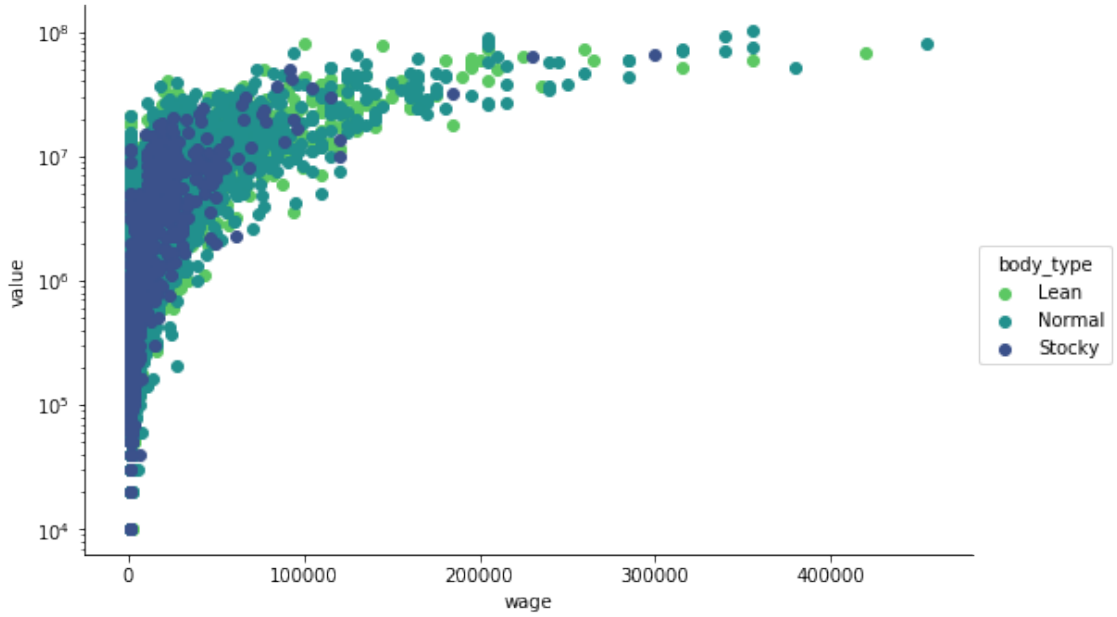
Most of the players with lower internation reputation have lower overall rating and potentials.Players with high international reputations are highly rated(overall) and have higher potential rating.

```
In [46]: #Scatter plot of Value, Wage and preferred Foot of players
         g = sb.FacetGrid(data = soccer_df, hue = 'preferred_foot', size = 5, aspect = 1.5, pale
         g.map(plt.scatter, 'wage','value',alpha=1);
         g.set(yscale = 'log') # need to set scaling before customizing ticks)
         g.add_legend();
```



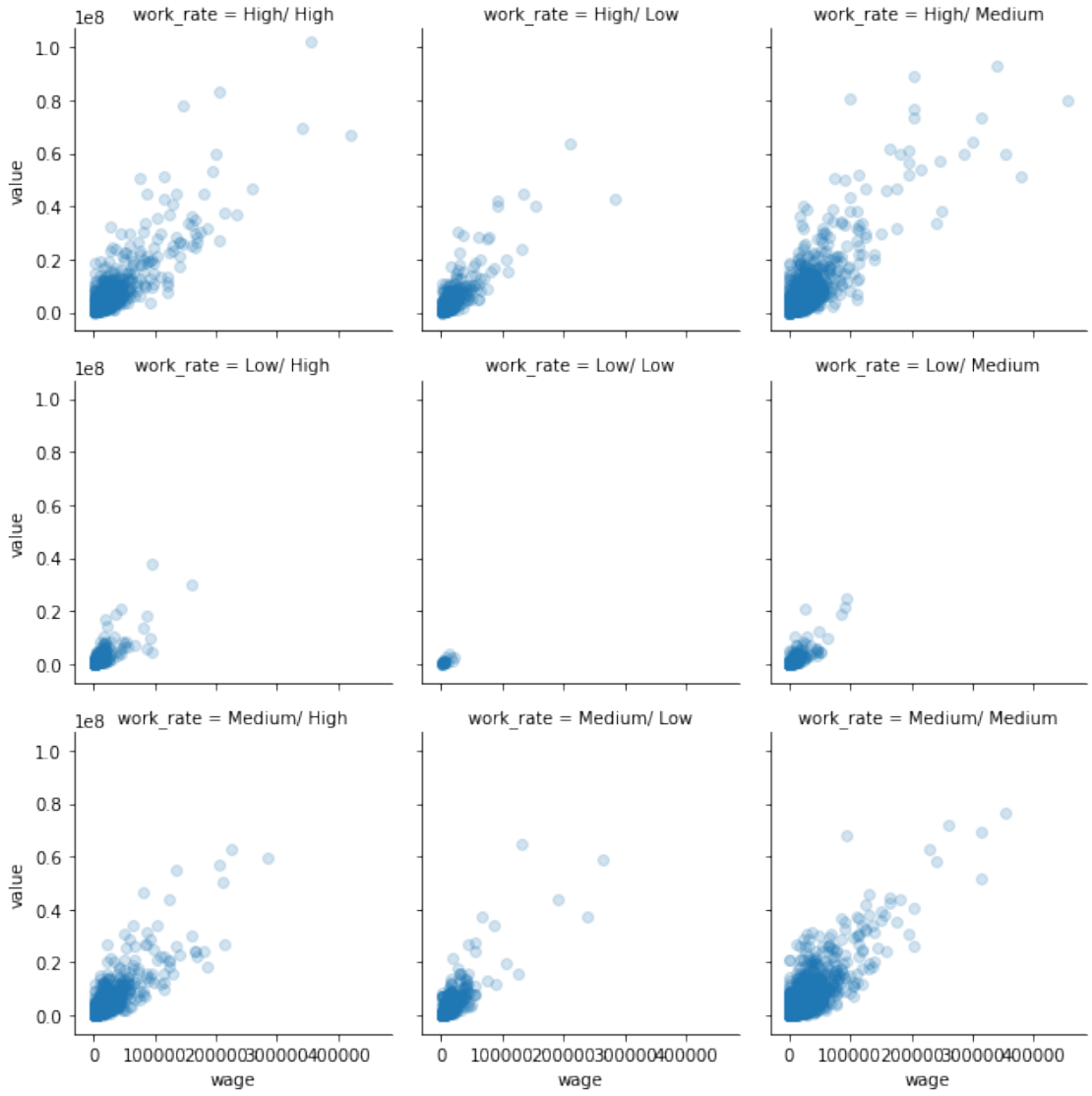
More of left footed points are closer the x and y axis,while more of the rigt footed points are farther away showing that more of the right footed players have high values and also earns more wages.

```
In [47]: #Scatter plot of Value, Wage and body-type of players
         g = sb.FacetGrid(data = soccer_df, hue = 'body_type',size = 5, aspect = 1.5, palette =
         g.map(plt.scatter, 'wage','value',alpha=1);
         g.set(yscale = 'log') # need to set scaling before customizing ticks)
         g.add_legend();
```



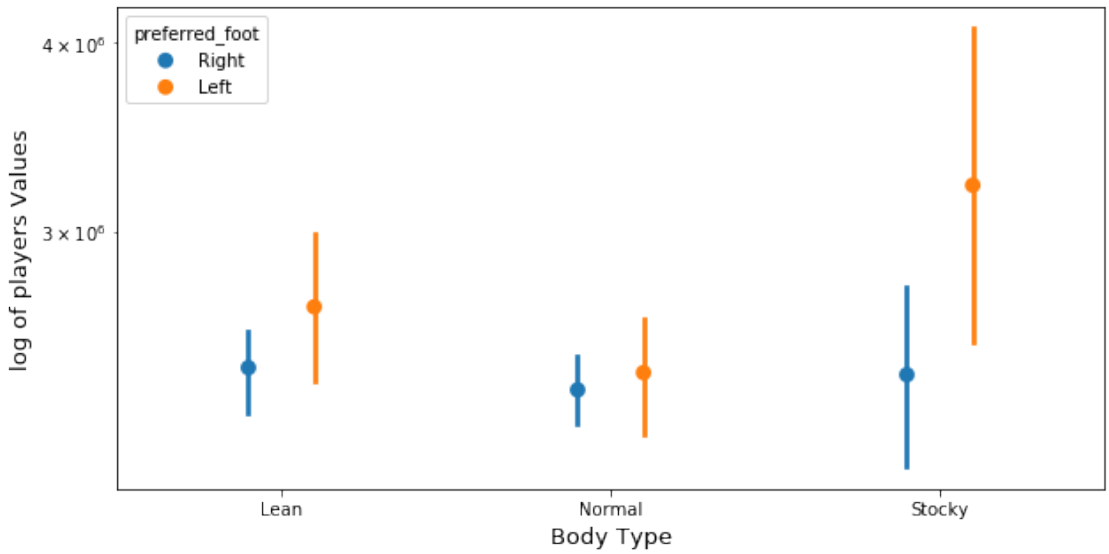
More of stocky body-type points are closer to the x and y axis,while more of the lean and normal body-types points are farther away showing that more of the lean and normal body-type players are more valued and also earns more wages.More of the normal body-types are farther away, which shows that more normal body-type players have high values and earns more wages.

```
In [48]: #Facet plot of  value, wage and work-rate
         g=sb.FacetGrid(data = soccer_df, col = 'work_rate', size = 3, col_wrap = 3)
         g.map(plt.scatter,'wage','value',alpha = 1/5);
```



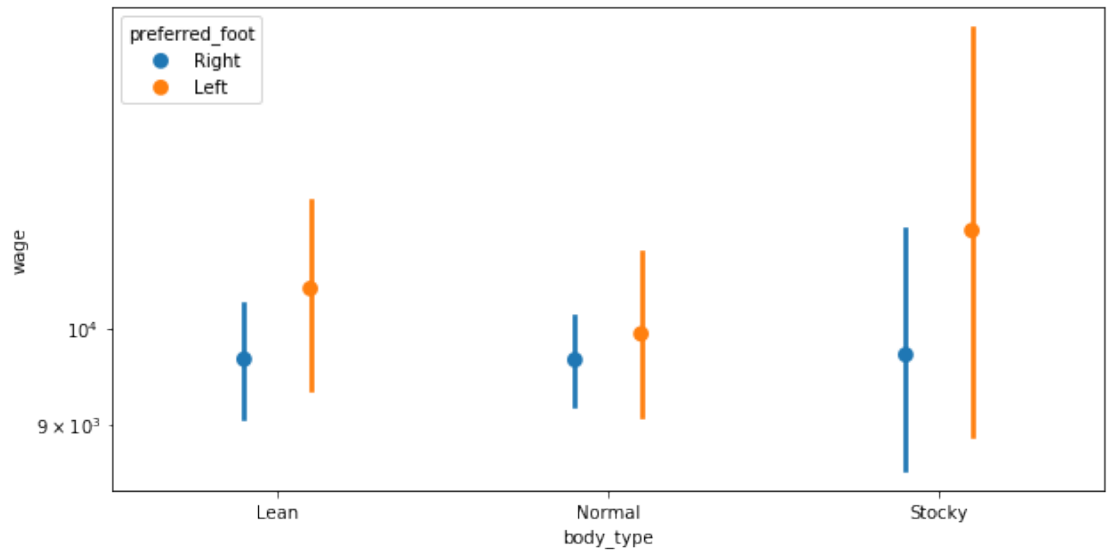
They all follow similar pattern showing a correlation between the numerical variables with the work-rate.The high/high, high/medium and the medium/medium are thicker, with the medium/medium being the thickest. It implies that more of the high valued and paid players have a work-rate of medium/medium.

```
In [49]: #Plot of Body-type,Value and Preferred Foot
         plt.figure(figsize=[10,5])
         ax = sb.pointplot(data = soccer_df, x = 'body_type', y = 'value', hue = 'preferred_foot
                           dodge = 0.2, linestyles = "")
         plt.yscale('log');
         plt.ylabel('log of players Values', color = 'black', fontsize = '13')
         plt.xlabel('Body Type', color = 'black', fontsize = '13');
```



From the above, I observed that there is general increase in values along the body-type axis, and increase in value from right foot to left foot among in body-type cluster.

```
In [50]: #Plot of Body-type,wage and Preferred Foot
         plt.figure(figsize=[10,5])
         ax = sb.pointplot(data = soccer_df, x = 'body_type', y = 'wage', hue = 'preferred_foot'
                           dodge = 0.2, linestyles = "")
         plt.yscale('log');
```



From the above chart,there is an increase in wage from right foot to left foot in each cluster. the stocky players who are left footed earns more than the lean and normal body-type players

who are left footed.Also,stocky players who are right footed earns more than the lean and normal body-type players who are right foooted.

1.6.1 Some of the Relationships Observed

I looked further into the relationships among international reputation,overall rating and players potentials, I observed a positive relationship among these three variants.Also, looking at the plots of value,wage and players work-rate, I observed a positive correlation, and most valued players who earn high wages have a medium/medium work-rate.Also,looking at the interaction between body-type,preferred foot versus value/wage, I noticed a similar trend in increase on wages and values within each cluster.

1.6.2 Interesting or surprising interactions between features

Surprisingly, I observed that the stocky body-type players that are either left footed or right footed are more valued and also earns more than their counterparts who are lean or of normal body-type. Meanwhile, earlier on, I observed that most of the players are of normal body-type and right footed.

1.7 Conclusions

1. Most of the players in this tournament are within the 20s age
2. Most of the players in the dataset are from England, and most of the players play in English league
3. left Foot(LF) position players are more valued than the rest position.Also players in Left wing back(LWB) and right wing back(RWB) positions are the least valued.
4. Strikers(ST) are the most popular players position in the dataset
5. Players with normal body-type are more in the dataset, also most of the players in the datasets are right footed.
6. Most of the players work-rate are medium/medium
7. More of the players with medium/medium work-rate are more valued and earns high wages
8. Players with high international-reputation have high overall rating and potential
9. There is a high positive correlation between players values and their wages
10. Players values and wages have a positive correlation with international-reputation, potential and overall rating of players
11. Majority of the right footed players are highly valuable and earns high wages
12. More number of the Lean and Normal body-type players are highly valuable and earns high wages.
13. Stocky players are the least popular in this dataset.But few Stocky players who are left footed are the most valuable and earns more than the normal and lean players.

```
In [51]: #Saving the cleaned master dataset for use in part 2
         soccer_df.to_csv('soccer_df1.csv',index=False)
```