Practico Mentoria - Analisis Exploratorio y Curación de Datos

Autor: Melania Omonte

Importaciones

In [1]:

```
%matplotlib inline

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp

from sklearn import preprocessing

import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
# Seteamos una semilla para Reproducibilidad
np.random.seed(0)
```

Carga de los Datasets

In [3]:

```
player_df = pd.read_csv('./Datasets/football_player.csv')
team_df = pd.read_csv('./Datasets/football_team.csv')
match_df = pd.read_csv('./Datasets/football_match.csv')
```

Exploremos un poco los Datasets

Players Dataset

In [4]:

```
print("Shape 'player_df' = {}".format(player_df.shape))
player_df.sample(5)
```

```
Shape 'player_df' = (11060, 40)
```

Out[4]:

	player name	birthday	height_m	weight_kg	overall_rating	potential	preferred foot	crossing	finishing	heading accuracy	...	vision	penalt
1534	Carlos Acuna	1988-06-23	1.78	71.21	67.33	71.81	right	51.52	67.43	68.86	...	44.00	66
7238	Max Christiansen	1996-09-25	1.88	83.91	64.09	74.45	right	47.73	37.73	60.82	...	56.45	47
10999	Zakaria M'Sila	1992-04-06	1.78	74.84	59.00	65.10	left	57.30	50.90	50.20	...	54.30	54
2669	Dimitrija Lazarevski	1982-09-23	1.78	74.84	59.00	61.00	left	51.00	38.00	54.00	...	NaN	61

	player name	birthday	height_m	weight_kg	overall_rating	potential	preferred foot	crossing	finishing	heading accuracy	...	vision	penal
1403	Bruma	1994-10-24	[illegible]	[illegible]	[illegible]	[illegible]	right	[illegible]	[illegible]	51.09	...	[illegible]	[illegible]

5 rows × 40 columns

Team Dataset

In [5]:

```
print("Shape 'team_df'    = {}".format(team_df.shape))
team_df.sample(5)
```

```
Shape 'team_df'    = (288, 22)
```

Out[5]:

	team long name	team short name	buildUpPlaySpeed	buildUpPlaySpeedClass	buildUpPlayDribblingClass	buildUpPlayPassing	buildUpPlayPassin
185	Lechia Gdańsk	LGD	50.83	Balanced	Little	48.33	
222	FC Penafiel	PEN	54.00	Balanced	Normal	39.00	
198	Pogoń Szczecin	POG	55.67	Balanced	Little	42.00	
197	Podbeskidzie Bielsko-Biała	POD	62.00	Balanced	Little	58.50	
173	Excelsior	EXC	57.67	Balanced	Little	60.00	

5 rows × 22 columns

Match Dataset

In [6]:

```
print("Shape 'match_df'   = {}".format(match_df.shape))
match_df.sample(5)
```

```
Shape 'match_df'   = (25979, 12)
```

Out[6]:

	country name	league name	season	stage	date	home team long name	home short long name	away team long name	away short long name	home team goal	away team goal	total goal
12042	Italy	Italy Serie A	2012/2013	35	2013-05-05 00:00:00	Catania	CAT	Siena	SIE	3	0	3
8575	Germany	Germany 1. Bundesliga	2010/2011	25	2011-03-05 00:00:00	VfB Stuttgart	STU	FC Schalke 04	S04	1	0	1
9067	Germany	Germany 1. Bundesliga	2012/2013	12	2012-11-17 00:00:00	Eintracht Frankfurt	EFR	FC Augsburg	AUG	4	2	6
13165	Italy	Italy Serie A	2015/2016	34	2016-04-21 00:00:00	Milan	ACM	Carpi	CAP	0	0	0
14904	Netherlands	Netherlands Eredivisie	2013/2014	2	2013-08-10 00:00:00	Heracles Almelo	HER	PEC Zwolle	ZWO	1	3	4

Exploremos un poco los Datasets y sus correspondientes Tipos

Players Dtypes

Players Dtypes

In [7]:

```
player_df.dtypes
```

Out[7]:

```
player name              object
birthday                 object
height_m                float64
weight_kg               float64
overall_rating          float64
potential               float64
preferred foot           object
crossing                float64
finishing               float64
heading accuracy        float64
short passing           float64
volleys                 float64
dribbling               float64
curve                   float64
free kick accuracy      float64
long passing            float64
ball control            float64
acceleration            float64
sprint speed            float64
agility                 float64
reactions               float64
balance                 float64
shot power              float64
jumping                 float64
stamina                 float64
strength                float64
long shots              float64
aggression              float64
interceptions           float64
positioning             float64
vision                  float64
penalties               float64
marking                 float64
standing tackle         float64
sliding tackle          float64
gk_diving               float64
gk_handling             float64
gk_kicking              float64
gk_positioning          float64
gk_reflexes             float64
dtype: object
```

Match Dtypes

In [8]:

```
match_df.dtypes
```

Out[8]:

```
country name              object
league name               object
season                    object
stage                      int64
date                      object
home team long name       object
home short long name      object
away team long name       object
away short long name      object
home team goal             int64
away team goal             int64
total goal                 int64
dtype: object
```

Team Dtypes

In [9]:

```
team_df.dtypes
```

Out[9]:

```
team long name                    object
team short name                   object
buildUpPlaySpeed                 float64
buildUpPlaySpeedClass             object
buildUpPlayDribblingClass         object
buildUpPlayPassing               float64
buildUpPlayPassingClass           object
buildUpPlayPositioningClass       object
chanceCreationPassing            float64
chanceCreationPassingClass        object
chanceCreationCrossing           float64
chanceCreationCrossingClass       object
chanceCreationShooting           float64
chanceCreationShootingClass       object
chanceCreationPositioningClass    object
defencePressure                  float64
defencePressureClass              object
defenceAggression                float64
defenceAggressionClass            object
defenceTeamWidth                 float64
defenceTeamWidthClass             object
defenceDefenderLineClass          object
dtype: object
```

---

1. Importacion de los datos

Calculemos el rango de fechas de los partidos

Antes de calcular el rango de fechas de los partidos, debemos validar que tipo de objeto es la fecha

In [10]:

```
match_df.dtypes['date']
```

Out[10]:

```
dtype('O')
```

Como la fecha es un campo del tipo object, no podremos calcular el rango solicitado, por lo tanto tendremos que cambiar el tipo, asi podemos generar el valor solicitado.

Modificamos el tipo "date", para poder calcular el rango

In [11]:

```
match_df2 = pd.read_csv("./Datasets/football_match.csv", parse_dates=["date"])
```

Visualizamos que haya cambiado el tipo "date"

In [12]:

```
match_df2.dtypes['date']
```

Out[12]:

Out[12]:

```
dtype('<M8[ns]')
```

Validamos que se cambio el tipo a datetime64[ns]

Realizamos la diferencia, para poder calcular el rango solicitado

In [13]:

```
match_df2['date'].max() - match_df2['date'].min()
```

Out[13]:

```
Timedelta('2868 days 00:00:00')
```

Rta: El rango de fechas entre partidos es 2868 dias.

2. Etiquetas de variables/columnas: no usar caracteres especiales

Chequar que no haya caracteres fuera de `a-Z`, `0-9` y `_` en los nombres de columnas de los Dataframes:

- `player_df`
- `team_df`
- `match_df`

Exploramos los Datasets y validamos que no hayan caracteres fuera de lo solicitado

Match DataSet

In [14]:

```
match_df.columns[~match_df.columns.str.match(r'^(\w+)$')]
```

Out[14]:

```
Index(['country name', 'league name', 'home team long name',
       'home short long name', 'away team long name', 'away short long name',
       'home team goal', 'away team goal', 'total goal'],
      dtype='object')
```

Chequeamos que existen varias columnas que tienen caracteres fuera de "a-Z, 0-9 y _" en el dataset match.

Player DataSet

In [15]:

```
player_df.columns[~player_df.columns.str.match(r'^(\w+)$')]
```

Out[15]:

```
Index(['player name', 'preferred foot', 'heading accuracy', 'short passing',
       'free kick accuracy', 'long passing', 'ball control', 'sprint speed',
       'shot power', 'long shots', 'standing tackle', 'sliding tackle'],
      dtype='object')
```

Chequeamos que existen varias columnas que tienen caracteres fuera de "a-Z, 0-9 y _" en el dataset player.

Team DataSet

In [16]:

```
team_df.columns[~team_df.columns.str.match(r'^(\w+)$')]
```

Out[16]:

```
Index(['team long name', 'team short name'], dtype='object')
```

Chequeamos que existen 2 columnas que tienen caracteres fuera de "a-Z, 0-9 y _" en el dataset team.

Reemplazamos los valores fuera de "a-Z, 0-9 y _" en el dataset team

In [17]:

```
team_df.columns = team_df.columns.str.replace(' ', '_')
team_df.head()
```

Out[17]:

	team_long_name	team_short_name	buildUpPlaySpeed	buildUpPlaySpeedClass	buildUpPlayDribblingClass	buildUpPlayPassing	buil
0	KRC Genk	GEN	56.33	Balanced	Little	44.33	
1	Beerschot AC	BAC	46.00	Balanced	Little	41.50	
2	SV Zulte-Waregem	ZUL	55.50	Balanced	Little	52.67	
3	Sporting Lokeren	LOK	64.00	Balanced	Little	53.50	
4	KSV Cercle Brugge	CEB	53.67	Balanced	Little	44.17	

5 rows × 22 columns

Validamos que se hayan reemplazado bien los campos en el dataset Team

In [18]:

```
team_df.columns[~team_df.columns.str.match(r'^(\w+)$')]
```

Out[18]:

```
Index([], dtype='object')
```

Validamos que se reemplazaron exitosamente los campos, ya que la consulta anterior no nos devuelve ningun campo.

Reemplazamos los valores fuera de "a-Z, 0-9 y _" en el dataset Player

In [19]:

```
player_df.columns = player_df.columns.str.replace(' ', '_')
player_df.head()
```

Out[19]:

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...	vis
0	Aaron Appindangoye	1992-02-29	1.83	84.82	63.60	67.60	right	48.60	43.60	70.60	...	5:
1	Aaron Cresswell	1989-12-15	1.70	66.22	66.97	74.48	left	70.79	49.45	52.94	...	5
2	Aaron Doran	1991-05-13	1.70	73.94	67.00	74.19	right	68.12	57.92	58.69	...	6

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...	vis
3	Aaron Galindo	1982-05-08	[illegible]	[illegible]	[illegible]	[illegible]	right	[illegible]	[illegible]	[illegible]	...	[illegible]
4	Aaron Hughes	1979-11-08	1.83	69.85	73.24	74.68	right	45.08	38.84	73.04	...	46

5 rows × 40 columns

Validamos que se hayan reemplazado bien los campos en el dataset Player

In [20]:

```
player_df.columns[~player_df.columns.str.match(r'^(\w+)$')]
```

Out[20]:

```
Index([], dtype='object')
```

Validamos que se reemplazaron exitosamente los campos, ya que la consulta anterior no nos devuelve ningun campo.

Reemplazamos los valores fuera de "a-Z, 0-9 y _" en el dataset Match

In [21]:

```
match_df.columns = match_df.columns.str.replace(' ', '_')
match_df.head()
```

Out[21]:

	country_name	league_name	season	stage	date	home_team_long_name	home_short_long_name	away_team_long_name	a
0	Belgium	Belgium Jupiler League	2008/2009	1	2008-08-17 00:00:00	KRC Genk	GEN	Beerschot AC	
1	Belgium	Belgium Jupiler League	2008/2009	1	2008-08-16 00:00:00	SV Zulte-Waregem	ZUL	Sporting Lokeren	
2	Belgium	Belgium Jupiler League	2008/2009	1	2008-08-16 00:00:00	KSV Cercle Brugge	CEB	RSC Anderlecht	
3	Belgium	Belgium Jupiler League	2008/2009	1	2008-08-17 00:00:00	KAA Gent	GEN	RAEC Mons	
4	Belgium	Belgium Jupiler League	2008/2009	1	2008-08-16 00:00:00	FCV Dender EH	DEN	Standard de Liège	

Validamos que se hayan reemplazado bien los campos en el dataset Match

In [22]:

```
match_df.columns[~match_df.columns.str.match(r'^(\w+)$')]
```

Out[22]:

```
Index([], dtype='object')
```

Validamos que se reemplazaron exitosamente los campos, ya que la consulta anterior no nos devuelve ningun campo.

3. Agregar nuevas caracteristicas

Agregar al Dataframe `player_df` una nueva columna que sea `imc` correspondiente al **Indice de Masa Corporal**

In [23]:

In [23]:

```
from sklearn import preprocessing
player_df_mod = pd.read_csv('./Datasets/football_player.csv')
player_df_mod.head()
```

Out[23]:

	player name	birthday	height_m	weight_kg	overall_rating	potential	preferred foot	crossing	finishing	heading accuracy	...	vision	penalties
0	Aaron Appindangoye	1992-02-29	1.83	84.82	63.60	67.60	right	48.60	43.60	70.60	...	53.60	47.60
1	Aaron Cresswell	1989-12-15	1.70	66.22	66.97	74.48	left	70.79	49.45	52.94	...	57.45	53.12
2	Aaron Doran	1991-05-13	1.70	73.94	67.00	74.19	right	68.12	57.92	58.69	...	69.38	60.54
3	Aaron Galindo	1982-05-08	1.83	89.81	69.09	70.78	right	57.22	26.26	69.26	...	53.78	41.74
4	Aaron Hughes	1979-11-08	1.83	69.85	73.24	74.68	right	45.08	38.84	73.04	...	46.48	52.96

5 rows × 40 columns

Saco el Cuadrado de la altura, para poder sacar el IMC. Una vez definido el cuadrado, calculo el IMC

In [24]:

```
#altura_cuadrado=player_df_mod['height_m']**2

def imc(peso,altura):
  return peso / (altura*altura)
```

Agrego la columna IMC en el dataFrame Player

In [25]:

```
player_df_mod['IMC'] = player_df_mod.apply(lambda x: imc(x.weight_kg, x.height_m), axis=1)
player_df_mod.describe()
```

Out[25]:

	height_m	weight_kg	overall_rating	potential	crossing	finishing	heading accuracy	short passing	volley
count	11060.000000	11060.000000	11060.000000	11060.000000	11060.000000	11060.000000	11060.000000	11060.000000	10582.00000
mean	1.817847	76.375393	66.821222	72.090216	52.853855	47.862155	56.100191	60.367143	47.11097
std	0.063278	6.799564	6.237719	5.800313	16.169989	18.109552	15.655413	13.508685	17.34028
min	1.570000	53.070000	43.000000	51.000000	6.000000	5.000000	8.000000	10.570000	3.75000
25%	1.780000	72.120000	62.820000	68.000000	43.440000	32.440000	49.097500	55.620000	33.25000
50%	1.830000	76.200000	66.720000	72.000000	56.300000	49.855000	58.805000	63.000000	49.30000
75%	1.850000	81.190000	70.952500	76.000000	64.710000	63.060000	66.750000	69.007500	60.73750
max	2.080000	110.220000	92.190000	95.230000	89.360000	92.230000	93.110000	95.180000	90.79000

8 rows × 38 columns

Grafico de distribucion de IMC

In [26]:

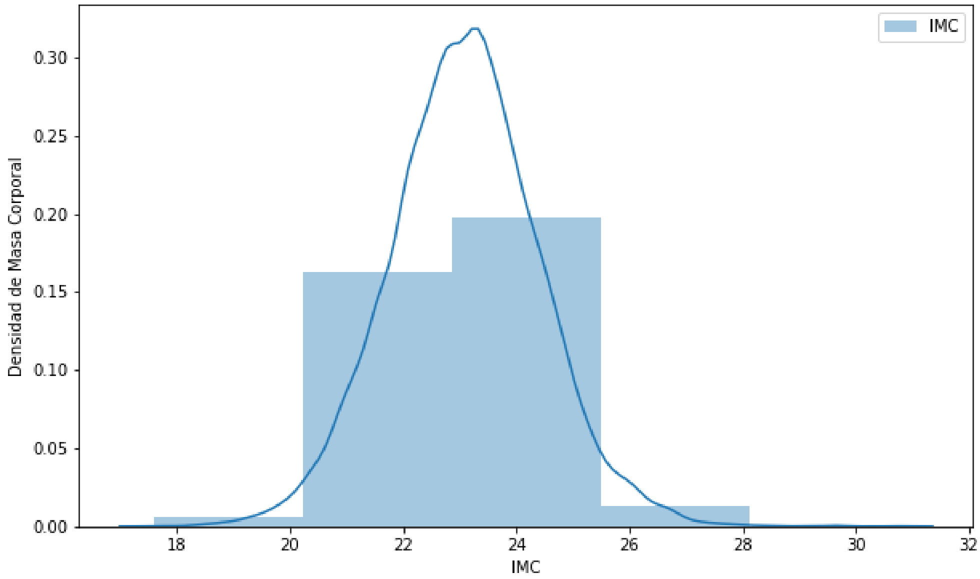
```
plt.figure(figsize=(10,6))
```

```
## Grafico la distribucion de Masa Corporal"
sns.distplot(player_df_mod.IMC, kde=True, bins=5, label='IMC')

plt.ylabel('Densidad de Masa Corporal')
plt.legend()
```

Out[26]:

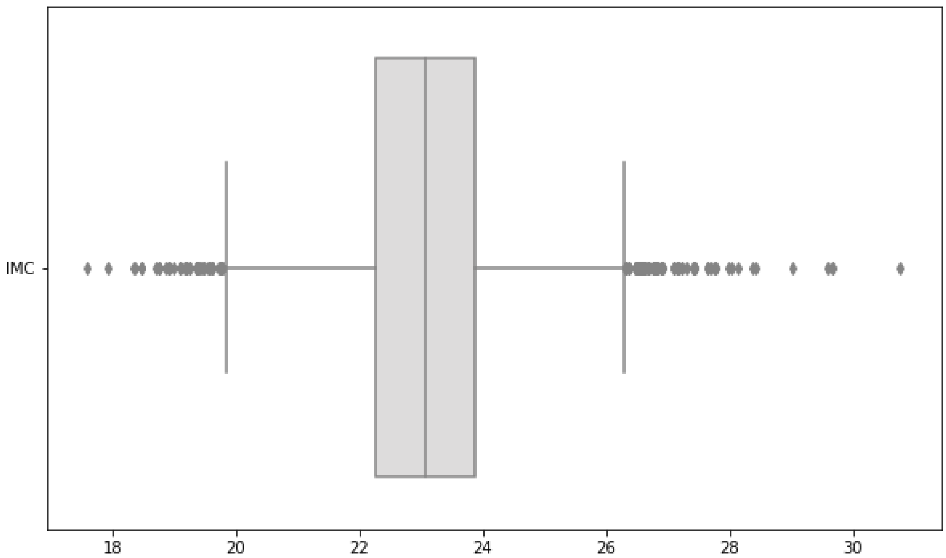
```
<matplotlib.legend.Legend at 0x2c0bb4d93c8>
```



Visualizamos los valores atipicos para el calculo realizado de IMC

In [27]:

```
#Para observar valores atípicos visualizamos el gráfico de caja...
plt.figure(figsize=(10,6))
data = player_df_mod[['IMC']]
sns.boxplot(data = data, orient="h", palette="coolwarm")
#sns.stripplot(data=data, color='black')
plt.show()
```



4. Tratar valores faltantes

Veamos cuantos valores nulos tenemos

In [28]:

```
player_missing_values_count = player_df.isnull().sum()

player_missing_values_count[player_missing_values_count > 0]
```

Out[28]:

```
volleys          478
curve            478
agility          478
balance          478
jumping          478
vision           478
sliding_tackle   478
dtype: int64
```

Tenemos 478 valores nulos en 7 columnas del DataFrame Player

In [29]:

```
len( player_df.dropna())/len(player_df)
```

Out[29]:

```
0.9567811934900543
```

In [30]:

```
len(player_df.dropna(subset=['volleys']))/len(player_df)
```

Out[30]:

```
0.9567811934900543
```

In [31]:

```
player_df[player_df.volleys.isnull()]
```

Out[31]:

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
25	Abdelmajid Oulmers	1978-09-12	1.73	64.86	68.80	69.40	right	60.00	50.00	62.00	...
30	Abdeslam Ouaddou	1978-11-01	1.90	82.10	76.60	78.60	right	67.20	34.00	78.00	...
31	Abdessalam Benjelloun	1985-01-28	1.88	81.19	63.33	71.33	right	42.00	66.17	41.50	...
83	Aco Stojkov	1983-04-29	1.78	74.84	59.67	62.67	right	59.67	57.67	62.67	...
85	Adailton	1977-01-24	1.75	73.03	71.83	74.00	left	54.67	74.50	62.17	...
175	Adrian Paluchowski	1987-08-19	1.80	74.84	55.00	60.50	right	43.00	57.00	49.00	...
190	Adriano	1982-01-21	1.75	76.20	68.75	74.00	right	52.00	71.00	67.00	...
203	Afonso Alves,24	1981-01-30	1.85	73.94	80.29	85.29	right	60.43	83.57	73.57	...
253	Alan Haydock	1976-01-13	1.75	72.12	63.33	65.67	right	60.00	43.00	62.67	...
275	Albert Baning	1985-03-19	1.93	81.19	65.00	78.00	right	35.00	30.00	56.60	...
289	Alberto Fontana	1967-01-23	1.85	73.03	76.00	77.25	right	22.00	28.00	37.00	...
343	Aleksandar Mitreski	1980-08-05	1.85	74.84	68.20	73.20	right	54.60	25.60	68.60	...

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
379	Alessandro Grandoni	1977-07-22	1.80	76.20	76.00	78.75	right	62.75	52.50	82.00	...
405	Alex Bruno	1982-05-09	1.88	86.18	69.80	75.40	right	30.60	35.40	71.80	...
456	Alexander Laas	1984-05-05	1.73	69.85	70.33	77.00	left	75.00	58.00	61.00	...
476	Alexandre Di Gregorio	1980-02-12	1.75	79.83	58.33	61.00	right	55.00	54.00	49.00	...
485	Alexandre Quennoz	1978-09-21	1.80	79.83	61.67	72.00	right	45.33	23.33	61.33	...
534	Allan Russell	1980-12-13	1.85	77.11	63.00	67.00	right	48.00	67.00	63.00	...
577	Amadou Alassane	1983-04-07	1.88	76.20	60.50	72.00	right	38.25	61.50	70.00	...
584	Amdy Faye	1977-03-12	1.83	78.02	73.20	75.80	right	55.40	51.00	74.40	...
655	Andre Leone	1979-02-12	1.83	81.19	71.80	81.00	left	23.00	27.00	78.00	...
675	Andrea Ardito	1977-01-08	1.73	60.78	71.00	77.00	left	69.00	58.00	69.00	...
713	Andrea Zanchetta	1975-02-02	1.80	73.94	76.00	80.00	right	70.00	64.50	77.50	...
756	Andrew McNeil	1987-01-19	1.80	83.01	61.50	67.50	right	21.00	21.00	21.00	...
774	Andriy Shevchenko	1976-09-29	1.83	72.12	81.20	89.80	right	70.60	85.20	80.00	...
798	Angel Manuel Vivar Dorado	1974-02-12	1.83	78.02	71.80	78.60	right	66.00	61.40	64.00	...
804	Angelo Martha	1982-04-29	1.85	82.10	60.67	64.00	right	33.00	31.00	62.00	...
827	Anthony Bentem	1990-03-19	1.80	74.84	56.33	66.67	right	34.00	24.00	45.00	...
885	Antonio Carlos Dos Santos	1979-10-03	1.88	68.04	70.60	79.00	left	68.80	45.80	62.80	...
887	Antonio Chimenti	1970-06-30	1.83	83.01	71.50	78.00	right	19.83	22.00	22.83	...
...	...	...	...	...	...	...	...	...	...	...	...
10449	Tony Heurtebis	1975-01-15	1.80	73.03	66.60	75.00	right	28.40	22.00	22.00	...
10475	Tugay Kerimoglou	1970-08-24	1.75	73.03	73.00	81.50	right	64.00	43.00	61.50	...
10499	Ulrich Le-Pen	1974-01-23	1.75	68.04	71.60	78.60	left	74.20	69.80	62.60	...
10504	Umit Ozat	1976-10-30	1.85	73.94	71.33	76.00	left	82.00	32.00	68.00	...
10528	Vahid Hashemian	1976-07-21	1.83	78.02	72.40	78.20	right	54.80	77.00	83.60	...
10545	Valerien Ismael	1975-09-28	1.90	83.01	79.60	85.20	right	50.00	47.00	83.60	...
10575	Veldin Muharemovic	1984-12-06	1.83	77.11	54.33	60.00	right	49.00	39.67	40.00	...
10627	Vincent Hognon	1974-08-16	1.83	74.84	75.00	75.67	right	33.00	25.00	79.33	...
10640	Vincent Provoost	1984-02-07	1.78	76.20	58.75	63.50	right	46.00	36.00	56.50	...
10670	Vitor Lima	1981-08-18	1.78	63.96	62.00	64.00	right	46.00	46.00	53.00	...
10671	Vitor Moreno	1980-11-29	1.80	86.18	58.50	62.00	right	52.00	48.00	55.00	...
10674	Vittorio Villano	1988-02-02	1.65	62.14	51.67	74.00	right	48.67	39.00	37.00	...
10684	Vladimir Stojkovic	1983-07-28	1.93	93.89	75.75	83.50	right	19.00	19.50	19.00	...
	Vojtech	1983									

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
10692	Vojtech	1983-	1.83	78.93	59.40	66.60	left	43.80	63.20	61.20	...
10696	Vukasin Devic	1984-03-15	1.85	74.84	65.50	70.00	right	32.00	40.00	66.00	...
10711	Walid Regragui	1975-09-23	1.78	66.22	67.00	73.00	right	73.00	56.00	66.00	...
10738	Wellington	1985-08-17	1.75	71.21	63.86	74.57	left	67.57	32.86	51.57	...
10742	Wender	1975-04-17	1.78	73.94	72.33	75.33	left	74.00	67.00	58.00	...
10757	Wesllem	1985-04-21	1.75	72.12	63.33	68.00	right	54.00	59.33	60.00	...
10802	Willy Grondin	1974-10-12	1.78	78.02	57.40	77.00	right	20.00	18.60	18.20	...
10894	Yasin Karaca	1983-12-16	1.70	67.13	61.00	63.00	right	62.00	56.00	33.00	...
10909	Yazid Mansouri,30	1978-02-25	1.75	68.95	69.80	72.00	right	63.00	55.40	70.80	...
10913	Yildiray Basturk	1978-12-24	1.70	68.95	78.50	82.67	right	78.67	60.33	37.83	...
10924	Yoav Ziv	1981-03-16	1.75	74.84	64.25	68.00	right	66.00	64.75	65.50	...
10934	Yohan Lachor,29	1976-08-03	1.83	79.83	64.33	65.33	right	38.33	39.33	74.33	...
10949	Yoshito Okubo	1982-06-09	1.68	60.78	71.00	77.00	right	72.00	69.00	55.00	...
10978	Yuri Cornelisse	1975-05-08	1.83	78.02	64.00	71.25	left	63.00	62.00	68.00	...
11020	Ze Manuel	1975-02-22	1.68	64.86	72.67	74.83	right	72.50	68.17	51.50	...
11024	Ze Vitor	1982-02-11	1.75	73.94	61.60	64.80	right	39.60	60.60	60.60	...
11027	Zeljko Kalac	1972-12-16	2.03	94.80	72.50	80.50	right	30.00	29.00	38.00	...

478 rows × 40 columns

Eliminamos los valores nulos

In [32]:

```
player_df_mod = player_df_mod.dropna(subset=['volleys'])
```

In [33]:

```
player_df_mod.describe()
```

Out[33]:

	height_m	weight_kg	overall_rating	potential	crossing	finishing	heading accuracy	short passing	volley
count	10582.000000	10582.000000	10582.000000	10582.000000	10582.000000	10582.000000	10582.000000	10582.000000	10582.00000
mean	1.818073	76.385504	66.884030	72.125307	52.925430	47.874585	56.039142	60.447359	47.11097
std	0.063466	6.814978	6.173155	5.732213	16.209403	18.158696	15.631791	13.481913	17.34028
min	1.570000	53.070000	43.750000	51.000000	6.000000	5.000000	8.000000	10.570000	3.75000
25%	1.780000	72.120000	62.902500	68.050000	43.505000	32.430000	49.150000	55.860000	33.25000
50%	1.830000	76.200000	66.790000	72.060000	56.520000	50.000000	58.710000	63.000000	49.30000
75%	1.850000	81.190000	70.950000	76.000000	64.800000	63.137500	66.710000	69.040000	60.73750
max	2.080000	110.220000	92.190000	95.230000	89.360000	92.230000	93.110000	95.180000	90.79000

8 rows × 38 columns

Validamos que se hayan validado esos valores nulos

In [34]:

```
player_missing_values_count = player_df_mod.isnull().sum()

player_missing_values_count[player_missing_values_count > 0]
```

Out[34]:

```
Series([], dtype: int64)
```

Validamos que se eliminaron exitosamente los campos nulos, ya que no devuelve ningun valor la consulta realizada.

In [35]:

```
player_df[player_df.volleys.isnull()]
```

Out[35]:

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
25	Abdelmajid Oulmers	1978-09-12	1.73	64.86	68.80	69.40	right	60.00	50.00	62.00	...
30	Abdeslam Ouaddou	1978-11-01	1.90	82.10	76.60	78.60	right	67.20	34.00	78.00	...
31	Abdessalam Benjelloun	1985-01-28	1.88	81.19	63.33	71.33	right	42.00	66.17	41.50	...
83	Aco Stojkov	1983-04-29	1.78	74.84	59.67	62.67	right	59.67	57.67	62.67	...
85	Adailton	1977-01-24	1.75	73.03	71.83	74.00	left	54.67	74.50	62.17	...
175	Adrian Paluchowski	1987-08-19	1.80	74.84	55.00	60.50	right	43.00	57.00	49.00	...
190	Adriano	1982-01-21	1.75	76.20	68.75	74.00	right	52.00	71.00	67.00	...
203	Afonso Alves,24	1981-01-30	1.85	73.94	80.29	85.29	right	60.43	83.57	73.57	...
253	Alan Haydock	1976-01-13	1.75	72.12	63.33	65.67	right	60.00	43.00	62.67	...
275	Albert Baning	1985-03-19	1.93	81.19	65.00	78.00	right	35.00	30.00	56.60	...
289	Alberto Fontana	1967-01-23	1.85	73.03	76.00	77.25	right	22.00	28.00	37.00	...
343	Aleksandar Mitreski	1980-08-05	1.85	74.84	68.20	73.20	right	54.60	25.60	68.60	...
379	Alessandro Grandoni	1977-07-22	1.80	76.20	76.00	78.75	right	62.75	52.50	82.00	...
405	Alex Bruno	1982-05-09	1.88	86.18	69.80	75.40	right	30.60	35.40	71.80	...
456	Alexander Laas	1984-05-05	1.73	69.85	70.33	77.00	left	75.00	58.00	61.00	...
476	Alexandre Di Gregorio	1980-02-12	1.75	79.83	58.33	61.00	right	55.00	54.00	49.00	...
485	Alexandre Quennoz	1978-09-21	1.80	79.83	61.67	72.00	right	45.33	23.33	61.33	...
534	Allan Russell	1980-12-13	1.85	77.11	63.00	67.00	right	48.00	67.00	63.00	...
577	Amadou Alassane	1983-04-07	1.88	76.20	60.50	72.00	right	38.25	61.50	70.00	...
584	Amdy Faye	1977-03-12	1.83	78.02	73.20	75.80	right	55.40	51.00	74.40	...
655	Andre Leone	1979-02-12	1.83	81.19	71.80	81.00	left	23.00	27.00	78.00	...
675	Andrea	1977-	1.73	60.78	71.00	77.00	left	69.00	58.00	69.00	...

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
[illegible]	Ardito	01-08	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...
713	Andrea Zanchetta	1975-02-02	1.80	73.94	76.00	80.00	right	70.00	64.50	77.50	...
756	Andrew McNeil	1987-01-19	1.80	83.01	61.50	67.50	right	21.00	21.00	21.00	...
774	Andriy Shevchenko	1976-09-29	1.83	72.12	81.20	89.80	right	70.60	85.20	80.00	...
798	Angel Manuel Vivar Dorado	1974-02-12	1.83	78.02	71.80	78.60	right	66.00	61.40	64.00	...
804	Angelo Martha	1982-04-29	1.85	82.10	60.67	64.00	right	33.00	31.00	62.00	...
827	Anthony Bentem	1990-03-19	1.80	74.84	56.33	66.67	right	34.00	24.00	45.00	...
885	Antonio Carlos Dos Santos	1979-10-03	1.88	68.04	70.60	79.00	left	68.80	45.80	62.80	...
887	Antonio Chimenti	1970-06-30	1.83	83.01	71.50	78.00	right	19.83	22.00	22.83	...
...	...	...	...	...	...	...	...	...	...	...	...
10449	Tony Heurtebis	1975-01-15	1.80	73.03	66.60	75.00	right	28.40	22.00	22.00	...
10475	Tugay Kerimoglou	1970-08-24	1.75	73.03	73.00	81.50	right	64.00	43.00	61.50	...
10499	Ulrich Le-Pen	1974-01-23	1.75	68.04	71.60	78.60	left	74.20	69.80	62.60	...
10504	Umit Ozat	1976-10-30	1.85	73.94	71.33	76.00	left	82.00	32.00	68.00	...
10528	Vahid Hashemian	1976-07-21	1.83	78.02	72.40	78.20	right	54.80	77.00	83.60	...
10545	Valerien Ismael	1975-09-28	1.90	83.01	79.60	85.20	right	50.00	47.00	83.60	...
10575	Veldin Muharemovic	1984-12-06	1.83	77.11	54.33	60.00	right	49.00	39.67	40.00	...
10627	Vincent Hognon	1974-08-16	1.83	74.84	75.00	75.67	right	33.00	25.00	79.33	...
10640	Vincent Provoost	1984-02-07	1.78	76.20	58.75	63.50	right	46.00	36.00	56.50	...
10670	Vitor Lima	1981-08-18	1.78	63.96	62.00	64.00	right	46.00	46.00	53.00	...
10671	Vitor Moreno	1980-11-29	1.80	86.18	58.50	62.00	right	52.00	48.00	55.00	...
10674	Vittorio Villano	1988-02-02	1.65	62.14	51.67	74.00	right	48.67	39.00	37.00	...
10684	Vladimir Stojkovic	1983-07-28	1.93	93.89	75.75	83.50	right	19.00	19.50	19.00	...
10692	Vojtech Schulmeister	1983-09-09	1.83	78.93	59.40	66.60	left	43.80	63.20	61.20	...
10696	Vukasin Devic	1984-03-15	1.85	74.84	65.50	70.00	right	32.00	40.00	66.00	...
10711	Walid Regragui	1975-09-23	1.78	66.22	67.00	73.00	right	73.00	56.00	66.00	...
10738	Wellington	1985-08-17	1.75	71.21	63.86	74.57	left	67.57	32.86	51.57	...
10742	Wender	1975-04-17	1.78	73.94	72.33	75.33	left	74.00	67.00	58.00	...
10757	Wesllem	1985-04-21	1.75	72.12	63.33	68.00	right	54.00	59.33	60.00	...
10802	Willy Grondin	1974-10-12	1.78	78.02	57.40	77.00	right	20.00	18.60	18.20	...
10894	Yasin Karaca	1983-12-16	1.70	67.13	61.00	63.00	right	62.00	56.00	33.00	...
10909	Yazid Mansouri,30	1978-02-25	1.75	68.95	69.80	72.00	right	63.00	55.40	70.80	...
10913	Yildiray Basturk	1978-12-24	1.70	68.95	78.50	82.67	right	78.67	60.33	37.83	...

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
10924	Yoav Ziv	1981-03-16	1.75	74.84	64.25	68.00	right	66.00	64.75	65.50	...
10934	Yohan Lachor,29	1976-08-03	1.83	79.83	64.33	65.33	right	38.33	39.33	74.33	...
10949	Yoshito Okubo	1982-06-09	1.68	60.78	71.00	77.00	right	72.00	69.00	55.00	...
10978	Yuri Cornelisse	1975-05-08	1.83	78.02	64.00	71.25	left	63.00	62.00	68.00	...
11020	Ze Manuel	1975-02-22	1.68	64.86	72.67	74.83	right	72.50	68.17	51.50	...
11024	Ze Vitor	1982-02-11	1.75	73.94	61.60	64.80	right	39.60	60.60	60.60	...
11027	Zeljko Kalac	1972-12-16	2.03	94.80	72.50	80.50	right	30.00	29.00	38.00	...

478 rows × 40 columns

Algunas tecnicas para tratar los *missing values*:

- **Eliminar** muestras o variables que tienen datos faltantes.
- **Imputar** los valores perdidos, es decir, sustituirlos por estimaciones por ejemplo la `media`, la `moda` ó usando `KNN`.

A) Analizar si es conveniente **Eliminar** las muestras o variables con datos faltantes del Dataframe `player_df`.

B) Aplicar la **Imputacion** usando la `media` o `moda` sobre las columnas con *missing values* del Dataframe `player_df`.

¿Eliminar los *missing values*? Justificar

Elimino los valores en un dataset modificado

Para eliminar los valores missing, debemos realizar un analisis sobre esos campos.Las columnas con valores missing son: volleys 478 curve 478 agility 478 balance 478 jumping 478 vision 478 sliding_tackle 478

Una de las soluciones para resolver este problema podria ser llenar estos los valores con ceros. Pero esta solucion no seria la mas optima, porque por ejemplo para un jugador, tendriamos que su agilidad es cero, y no seria representativo.

Otra de las opciones para resolver este problema es decidir eliminar estos valores, suponiendo que los valores missing pueden ser errores, de esta manera subsanamos dicho error.

Otra de las opciones es la imputacion de la Media o Moda, segun corresponda, analisis detallado mas abajo.

Imputacion usando Media y Moda

Reemplazo de Valores Faltantes usando la moda

In [36]:

```
# Rellenamos usando la Moda
# player_df.fillna(player_df.mode(), inplace=True)

player_df_reemplazo_nan_moda = player_df

for column in ['volleys','agility','curve','balance','jumping','vision','sliding_tackle']:
    player_df_reemplazo_nan_moda[column].fillna(player_df_reemplazo_nan_moda[column].mode()[0], inp
lace=True)

player_df_reemplazo_nan_moda
```

Out[36]:

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
0	Aaron Appindangoye	1992-02-29	1.83	84.82	63.60	67.60	right	48.60	43.60	70.60	...

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
1	Aaron Cresswell	1989-12-15	1.70	66.22	66.97	74.48	left	70.79	49.45	52.04	...
2	Aaron Doran	1991-05-13	1.70	73.94	67.00	74.19	right	68.12	57.92	58.69	...
3	Aaron Galindo	1982-05-08	1.83	89.81	69.09	70.78	right	57.22	26.26	69.26	...
4	Aaron Hughes	1979-11-08	1.83	69.85	73.24	74.68	right	45.08	38.84	73.04	...
5	Aaron Hunt	1986-09-04	1.83	73.03	77.26	80.15	left	73.89	72.81	65.52	...
6	Aaron Kuhl	1996-01-30	1.73	66.22	60.57	76.00	right	47.57	31.57	46.57	...
7	Aaron Lennon	1987-04-16	1.65	63.05	79.77	82.00	right	78.04	65.96	30.46	...
8	Aaron Lennox	1993-02-19	1.90	82.10	48.00	56.86	right	12.00	15.00	16.00	...
9	Aaron Meijers	1987-10-28	1.75	77.11	67.05	69.42	left	63.89	46.05	56.84	...
10	Aaron Mokoena	1980-11-25	1.83	82.10	71.62	76.00	right	26.00	56.12	79.75	...
11	Aaron Mooy	1990-09-15	1.75	68.04	66.29	73.14	right	66.57	61.21	43.46	...
12	Aaron Muirhead	1990-08-30	1.88	76.20	62.25	70.00	right	48.00	23.00	65.00	...
13	Aaron Niguez	1989-04-26	1.70	64.86	66.93	75.30	left	59.56	66.22	53.89	...
14	Aaron Ramsey	1990-12-26	1.78	69.85	78.50	84.68	right	72.88	70.92	56.75	...
15	Aaron Splaine	1996-10-13	1.73	73.94	54.62	62.62	left	54.38	51.38	42.38	...
16	Aaron Taylor-Sinclair	1991-04-08	1.83	79.83	62.61	69.50	left	59.61	23.83	58.39	...
17	Aaron Wilbraham	1979-10-21	1.90	72.12	61.77	64.14	right	47.36	64.45	72.86	...
18	Aatif Chahechouhe	1986-07-02	1.75	68.04	69.38	74.50	right	69.50	78.00	57.19	...
19	Abasse Ba	1976-07-12	1.88	83.91	65.60	70.60	right	41.00	33.00	73.20	...
20	Abdelaziz Barrada	1989-06-19	1.78	73.03	71.86	78.86	right	70.21	58.36	50.36	...
21	Abdelfettah Boukhriss	1986-10-22	1.85	73.03	64.00	69.00	left	47.00	29.00	65.00	...
22	Abdelhamid El Kaoutari	1990-03-17	1.80	73.03	68.29	73.57	left	61.14	20.05	62.33	...
23	Abdelkader Ghezzal	1984-12-05	1.83	78.02	68.69	71.31	right	63.50	63.85	65.73	...
24	Abdellah Zoubir	1991-12-05	1.80	73.03	59.00	69.00	right	49.00	56.00	47.00	...
25	Abdelmajid Oulmers	1978-09-12	1.73	64.86	68.80	69.40	right	60.00	50.00	62.00	...
26	Abdelmalek Cherrad	1981-01-14	1.85	74.84	61.60	73.00	right	49.00	64.00	63.40	...
27	Abdelmalek El Hasnaoui	1994-02-09	1.80	72.12	63.00	72.00	left	42.00	52.00	39.00	...
28	Abdelouahed Chakhsi	1986-10-01	1.83	77.11	54.33	59.50	right	53.00	34.33	48.00	...
29	Abderrazak Jadid	1983-06-01	1.78	71.21	63.31	65.54	right	64.69	57.23	47.23	...
...	...	...	...	...	...	...	...	...	...	...	...
11030	Zezinho	1992-09-23	1.75	73.03	69.00	82.00	right	53.00	63.00	63.00	...
11031	Zhi Zheng	1980-08-20	1.80	74.84	70.43	70.14	right	70.00	67.86	66.00	...
11032	Zhi-Gin Lam	1991-06-04	1.75	66.22	66.28	74.28	right	66.61	44.83	38.89	...

	player_name	birthday	height_m	weight_kg	overall_rating	potential	preferred_foot	crossing	finishing	heading_accuracy	...
11033	Zie Diabate	1989-03-02	1.78	64.86	62.61	67.50	left	54.83	26.28	54.28	...
11034	Ziggy Gordon	1993-04-23	1.80	77.11	62.30	71.10	right	57.70	30.30	52.30	...
11035	Ziguy Badibanga	1991-11-26	1.73	69.85	63.75	72.25	right	60.00	58.00	34.00	...
11036	Zinedine Machach	1996-01-05	1.85	73.94	63.44	76.44	right	64.11	53.89	54.33	...
11037	Zinho Gano	1993-10-13	1.98	92.99	62.73	70.47	left	48.93	63.80	66.40	...
11038	Ziri Hammar	1992-07-25	1.80	73.94	64.18	72.64	right	67.00	60.00	59.00	...
11039	Zizo	1996-01-10	1.75	67.13	60.40	72.40	right	41.00	44.00	36.40	...
11040	Zlatan Bajramovic	1979-08-12	1.83	78.93	75.57	82.86	right	73.86	67.57	75.00	...
11041	Zlatan Ibrahimovic	1981-10-03	1.96	94.80	88.29	90.05	right	72.38	90.00	79.71	...
11042	Zlatan Ljubijankic	1983-12-15	1.85	79.83	70.35	73.10	right	68.80	68.70	71.25	...
11043	Zlatko Dedic	1984-10-05	1.83	77.11	67.67	71.75	right	57.50	68.46	63.58	...
11044	Zlatko Janjic	1986-05-07	1.88	83.01	64.54	65.85	right	63.31	67.15	62.31	...
11045	Zlatko Junuzovic	1987-09-26	1.73	68.95	75.46	79.00	right	73.74	67.15	55.90	...
11046	Zola Matumona	1981-11-26	1.65	64.86	66.00	66.14	left	64.71	57.57	46.00	...
11047	Zoltan Gera	1979-04-22	1.83	74.84	74.42	76.65	right	81.92	73.46	70.04	...
11048	Zoltan Stieber	1988-10-16	1.75	67.13	69.44	76.26	left	64.30	67.15	53.93	...
11049	Zoltan Szelesi	1981-11-22	1.83	79.83	67.00	65.00	right	57.86	55.00	65.29	...
11050	Zoran Josipovic	1995-08-25	1.88	74.84	59.55	72.55	right	28.00	59.64	66.00	...
11051	Zoran Rendulic	1984-05-22	1.90	81.19	64.40	72.00	right	37.00	17.00	78.00	...
11052	Zoran Tosic	1987-04-28	1.70	71.21	77.04	80.70	left	69.00	75.09	51.35	...
11053	Zouhaier Dhaouadhi	1988-01-01	1.80	72.12	64.00	65.38	left	65.00	60.00	36.75	...
11054	Zouhair Feddal	1989-01-01	1.90	78.02	65.76	69.38	left	47.52	29.57	68.48	...
11055	Zoumana Camara	1979-04-03	1.83	76.20	74.38	75.46	right	42.00	27.00	75.15	...
11056	Zsolt Laczko	1986-12-18	1.83	79.83	65.69	71.62	left	67.25	46.75	60.31	...
11057	Zsolt Low	1979-04-29	1.80	69.85	67.57	72.86	left	63.14	44.57	59.86	...
11058	Zurab Khizanishvili	1981-10-06	1.85	78.02	70.75	78.12	right	46.75	43.00	79.00	...
11059	Zvjezdan Misimovic	1982-06-05	1.80	79.83	80.00	81.70	right	78.20	72.60	57.40	...

11060 rows × 40 columns

Validamos que se hayan reemplazado bien los valores y que no hayan valores missing:

In [37]:

```
player_missing_values_count = player_df.isnull().sum()

player_missing_values_count[player_missing_values_count > 0]
```

Out[37]:

```
Series([], dtype: int64)
```

Se comprueba exitosamente que no hay valores missing, una vez que se reemplazaron los datos por su moda.

Reemplazo de Valores Faltantes usando la Media

In [38]:

```
# Rellenamos usando la Moda
player_df_reemplazo_nan_media = player_df
player_df_reemplazo_nan_media.fillna(player_df_reemplazo_nan_media.mean(), inplace=True)
```

In [39]:

```
player_missing_values_count = player_df_reemplazo_nan_media.isnull().sum()

player_missing_values_count[player_missing_values_count > 0]
```

Out[39]:

```
Series([], dtype: int64)
```

Se comprueba exitosamente que no hay valores missing, una vez que se reemplazaron los datos por su media.

5. Normalizacion de columnas

Normalizar la columna `crossing` usando **Min-Max**.

Normalizar la columna `short_passing` usando **Z-score**.

Normalizando la columna crossing, usando Min-Max

Normalizamos la columna y mostramos un listado antes de la normalizacion y despues de la misma

In [40]:

```
# TODO
print(player_df.crossing.head(10))

scaler = preprocessing.MinMaxScaler()
player_df[["crossing"]] = scaler.fit_transform(player_df[["crossing"]])

print(player_df.crossing.head(10))
```

```
0    48.60
1    70.79
2    68.12
3    57.22
4    45.08
5    73.89
6    47.57
7    78.04
8    12.00
9    63.89
Name: crossing, dtype: float64
0    0.511036
1    0.777231
2    0.745202
3    0.614443
4    0.468810
5    0.814419
6    0.498680
7    0.864203
8    0.071977
```

```
9    0.694458
Name: crossing, dtype: float64
```

Normalizamos la columna short_passing usando Z-score

Normalizamos la columna y mostramos un listado antes de la normalizacion y despues de la misma.

In [41]:

```
print(player_df["short_passing"].head(10))

scaler = preprocessing.MinMaxScaler()
player_df[["short_passing"]] = sp.stats.zscore(player_df[["short_passing"]])

print(player_df["short_passing"].head(10))
```

```
0    60.60
1    62.27
2    65.12
3    64.70
4    64.76
5    78.26
6    63.57
7    76.27
8    23.00
9    68.95
Name: short_passing, dtype: float64
0    0.017238
1    0.140868
2    0.351853
3    0.320761
4    0.325202
5    1.324605
6    0.237107
7    1.177285
8   -2.766282
9    0.635387
Name: short_passing, dtype: float64
```

6. Codificar variables

> Las variables categóricas deben ser etiquetadas como variables numéricas, no como cadenas.

Codificar la variable `country_name` del Dataframe `match_df`

La columna "country_name" antes de ser etiquetada como variable numerica

In [42]:

```
print(set(match_df["country_name"]))
```

```
{'Portugal', 'Scotland', 'Switzerland', 'Italy', 'Poland', 'Spain', 'Belgium', 'Germany',
'Netherlands', 'France', 'England'}
```

Etiqueto como variables numericas a la columna "country_name"

In [43]:

```
le = preprocessing.LabelEncoder()
match_df[["country_name"]] = le.fit_transform(match_df[["country_name"]])
```

Visualizo la columna, con los datos ya transformados

In [44]:

```
print(set(match_df["country_name"]))
```

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

A la columna "country_name", la etiquedamos con variables numericas

In [ ]:

In [ ]: