Ejercicio – Masajeo de Datos II

In [ ]:
```
import pandas as pd
import numpy as np
import os
```

In [ ]:
```
cwd = os.getcwd()
cwd
```

Out[ ]:
```
'e:\\WORK IN PROGRESS\\Data Analytics course\\parte 2 python\\week 17'
```

In [ ]:
```
os.chdir('e:\\WORK IN PROGRESS\\Data Analytics course\\parte 2 python\\week 17')
```

In [ ]:
```
cwd = os.getcwd()
cwd
```

Out[ ]:
```
'e:\\WORK IN PROGRESS\\Data Analytics course\\parte 2 python\\week 17'
```

In [ ]:
```
# utilice el código del ejercicio anterior para no tener que repetir refinamientos
# de datos.
# Por ese motivo tiene un nombre diferente.

df = pd.read_csv('fifa_eda_2.csv')
```

In [ ]:
```
df.shape
```

Out[ ]:
```
(17918, 21)
```

In [ ]:
```
df.sample(20)
```

Out[ ]:

	Unnamed: 0	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	...	International Reputation	Skill Moves	Position	Joined	C
10526	10694	238014	R. Janssen	28	Netherlands	65	65	VVV-Venlo	425.0	2.0	...	1.0	2.0	LB	2016	2(
4909	4979	206654	Pablo Marí	24	Spain	70	76	Deportivo de La Coruña	2100.0	39.0	...	1.0	2.0	LCB	2016	2(
6974	7092	228980	M. Tirpan	24	Belgium	68	72	Sporting Lokeren	975.0	5.0	...	1.0	2.0	RB	2018	2(
10610	10779	122066	R. König	35	Germany	65	65	FSV Zwickau	250.0	1.0	...	1.0	2.0	RS	2016	2(
4191	4251	231002	J. Šimunović	23	Bosnia Herzegovina	71	76	Celtic	2700.0	24.0	...	1.0	2.0	CB	2015	2(
16853	17137	238447	H. Ishikawa	19	Japan	55	64	Sagan Tosu	130.0	1.0	...	1.0	2.0	CM	2017	2(
219	219	162895	Cesc Fàbregas	31	Spain	83	83	Chelsea	22000.0	140.0	...	4.0	3.0	CM	2014	2(
1893	1909	202151	K. Stafylidis	24	Greece	75	79	FC Augsburg	7000.0	18.0	...	1.0	3.0	LB	2015	2(
13784	14030	240156	L. Marseiler	21	Germany	61	71	SpVgg Unterhaching	400.0	1.0	...	1.0	3.0	RM	2015	2(
1158	1167	199393	N. Pallois	30	France	77	77	FC Nantes	6500.0	25.0	...	2.0	2.0	LCB	2017	2(
5695	5784	220968	A. El Messaoudi	22	Morocco	69	74	Fortuna Sittard	1200.0	4.0	...	1.0	2.0	LCM	2017	2(
16614	16893	243372	J. Nieva	19	Colombia	56	70	América de Cali	180.0	1.0	...	1.0	2.0	CDM	2018	2(
5619	5707	234002	Caio Tafarell	26	Brazil	69	69	Vitória	925.0	6.0	...	1.0	2.0	CDM	2018	2(
7388	7512	210401	C. Donis	23	Greece	68	75	Panathinaikos FC	1200.0	1.0	...	1.0	3.0	CM	2012	2(
13353	13597	243868	V.	22	Congo	62	70	Vejle	375.0	2.0	...	1.0	2.0	CB	2018	2(

	Unnamed: 0	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	...	International Reputation	Skill Moves	Position	Joined	C
			Emmersón					Boldklub								
552	554	49031	S. Sorrentino	39	Italy	80	80	Chievo Verona	1000.0	10.0	...	2.0	1.0	GK	2016	2(
14005	14256	227185	K. Danielak	26	Poland	61	61	Arka Gdynia	270.0	1.0	...	1.0	2.0	RM	2018	2(
15522	15790	210418	A. Banasiak	28	Poland	59	59	Zagłębie Sosnowiec	130.0	1.0	...	1.0	2.0	LDM	2017	2(
16443	16720	231457	D. Baur	19	Scotland	56	72	Heart of Midlothian	180.0	1.0	...	1.0	2.0	CB	2015	2(
9985	10147	183347	J. Leutwiler	29	Canada	65	65	Blackburn Rovers	375.0	8.0	...	1.0	1.0	GK	2017	2(

In [ ]:
```
# Se elimina la columna 'Unnamed: 0' con .drop

df.drop('Unnamed: 0',axis='columns', inplace= True)
```

- Visualización de una muestra de 20 líneas.

In [ ]:
```
df.sample(20)
```

Out[ ]:

	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Preferred Foot	International Reputation	Skill Moves	Position	Joined	Contra Val Un
4564	227293	Jonas	26	Brazil	71	72	Al Ittihad	2300.0	21.0	Right	1.0	2.0	LDM	2018	2021-0 (
6675	207660	D. Wydra	24	Austria	68	71	FC Erzgebirge Aue	950.0	3.0	Right	1.0	2.0	CB	2017	2019-0 (
601	208920	N. Aké	23	Netherlands	79	84	Bournemouth	14500.0	55.0	Left	1.0	2.0	CB	2017	2022-0 (
8734	243991	Valverde	33	Spain	66	66	Extremadura UD	400.0	2.0	Right	1.0	3.0	RM	2017	2019-0 (
4685	184320	R. Bianco	30	Italy	70	70	Perugia	1600.0	3.0	Right	1.0	3.0	CM	2017	2019-0 (
6592	231706	K. Londoño	24	Colombia	68	70	Once Caldas	1100.0	2.0	Right	1.0	3.0	RM	2018	2021-0 (
12087	195124	D. Amoo	27	England	63	63	Cambridge United	375.0	3.0	Right	1.0	3.0	RM	2017	2019-0 (
9483	166847	C. McCann	30	Republic of Ireland	66	66	Atlanta United	475.0	2.0	Left	1.0	3.0	LB	2016	2018-0 (
15152	219701	J. Wrąbel	22	Poland	59	69	Śląsk Wrocław	190.0	1.0	Right	1.0	1.0	GK	2013	2019-0 (
12463	222870	G. Cooper	21	England	63	72	Peterborough United	575.0	2.0	Left	1.0	3.0	LM	2018	2021-0 (
1528	205295	H. Soudani	30	Algeria	76	76	Nottingham Forest	7500.0	51.0	Left	1.0	4.0	ST	2018	2021-0 (
12093	229942	A. Disasi	20	France	63	74	Stade de Reims	550.0	2.0	Right	1.0	2.0	CB	2017	2021-0 (
232	230168	Raphaelito Anjos	30	Brazil	82	82	Grêmio	14000.0	32.0	Right	1.0	1.0	GK	2018	2021-0 (
9560	239572	B. Jayiya	28	South Africa	66	66	Kaizer Chiefs	650.0	1.0	Right	1.0	3.0	RW	2017	2020-0 (

	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Preferred Foot	International Reputation	Skill Moves	Position	Joined	Contra Val Un
10946	199197	P. Salomon	30	Austria	64	64	SV Mattersburg	425.0	5.0	Right	1.0	2.0	LCM	2018	2019-0
13312	199311	Lee Hyun Woong	30	Korea Republic	62	62	Gyeongnam FC	280.0	1.0	Right	1.0	2.0	CM	2017	2022-0
14831	227760	A. Nesbitt	21	Scotland	60	74	Milton Keynes Dons	450.0	2.0	Right	1.0	2.0	LM	2017	2019-0
16287	231356	M. Millar	21	Australia	57	66	Central Coast Mariners	140.0	1.0	Right	1.0	3.0	RWB	2018	2019-0
10125	239703	M. Biankadi	23	Congo	65	72	FC Hansa Rostock	750.0	1.0	Right	1.0	3.0	LM	2018	2020-0
5109	230520	Sevinho	30	Brazil	70	70	Cruzeiro	1700.0	17.0	Right	1.0	3.0	ST	2018	2021-0

In [ ]:
```
df.describe()
```

Out[ ]:

	ID	Age	Overall	Potential	Value	Wage	International Reputation	Skill Moves	Joined	Height
count	17918.000000	17918.000000	17918.000000	17918.000000	17907.000000	17918.000000	17918.000000	17918.000000	17918.000000	17918.000000
mean	214225.713472	25.105257	66.236801	71.329334	2450.132909	9.883748	1.113908	2.362875	2016.427391	5.946753
std	30042.543245	4.675372	6.929243	6.144098	5633.206685	22.142769	0.395495	0.756309	2.033693	0.220812
min	16.000000	16.000000	46.000000	48.000000	10.000000	1.000000	1.000000	1.000000	1991.000000	5.083333
25%	200235.250000	21.000000	62.000000	67.000000	325.000000	1.000000	1.000000	2.000000	2016.000000	5.750000
50%	221701.500000	25.000000	66.000000	71.000000	700.000000	3.000000	1.000000	2.000000	2017.000000	5.916667
75%	236508.750000	28.000000	71.000000	75.000000	2100.000000	9.000000	1.000000	3.000000	2018.000000	6.083333
max	246620.000000	45.000000	94.000000	95.000000	118500.000000	565.000000	5.000000	5.000000	2018.000000	6.750000

- Generar la media, mínimo y máximo de los jugadores en total, y por país (groupby)

In [ ]:
```
# Solo para este ejercicio voy a crear un dataframe nuevo con aquellas columnas que
#  me interesa conocer la media, el minimo y el maximo.
df_new= df[['Age','Overall','Potential','Value','Wage','International Reputation',
'Skill Moves','Joined','Height','Weight','Release Clause']]
```

In [ ]:
```
# Aqui generamos la media, minimo y maximo de los jugadores en total con la funcion
#  aggregate.
df_new.aggregate(['mean', 'min', 'max'])
```

Out[ ]:

	Age	Overall	Potential	Value	Wage	International Reputation	Skill Moves	Joined	Height	Weight	Release Clause
mean	25.105257	66.236801	71.329334	2450.132909	9.883748	1.113908	2.362875	2016.427391	5.946753	165.968858	4585.060974
min	16.000000	46.000000	48.000000	10.000000	1.000000	1.000000	1.000000	1991.000000	5.083333	110.000000	13.000000
max	45.000000	94.000000	95.000000	118500.000000	565.000000	5.000000	5.000000	2018.000000	6.750000	243.000000	228100.000000

In [ ]:
```
# Para agrupar los datos por paises, al nuevo dataframe que cree le voy a agragar
# la columna de nacionalidad del dataframe original.
# Despues utilizo las funciones groupby y aggregate para agruparlos.

df_new = df[['Age','Overall','Potential','Value','Wage','International Reputation',
'Skill Moves','Joined','Height','Weight','Release Clause','Nationality']]

df_new.groupby('Nationality').aggregate(['mean', 'min', 'max'])
```

Out[ ]:

	Age			Overall			Potential			Value	...	Joined	Height				
	mean	min	max	mean	min	max	mean	min	max	mean	...	max	mean	min	max	mean	min
Nationality																	
Afghanistan	22.500000	20	26	61.000000	59	63	67.750000	64	71	342.500000	...	2018	5.729167	5.666667	5.833333	149.250000	143.0
Albania	24.076923	18	34	66.025641	52	81	71.743590	62	86	1959.871795	...	2018	5.950855	5.583333	6.416667	164.384615	137.0
Algeria	27.050000	19	35	70.633333	58	85	72.983333	62	85	4670.916667	...	2018	5.947222	5.500000	6.416667	167.250000	132.0
Andorra	28.000000	28	28	62.000000	62	62	64.000000	64	64	290.000000	...	2018	6.000000	6.000000	6.000000	174.000000	174.0
Angola	25.866667	19	34	67.600000	60	78	71.533333	64	82	2018.333333	...	2018	5.877778	5.583333	6.166667	166.533333	143.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
Uzbekistan	29.500000	29	30	67.500000	60	75	67.500000	60	75	3100.000000	...	2017	5.958333	5.916667	6.000000	164.000000	163.0
Venezuela	24.305085	18	34	67.474576	53	81	73.355932	55	84	2534.322034	...	2018	5.861582	5.250000	6.416667	163.203390	121.0
Wales	24.789062	17	37	64.203125	47	88	69.929688	56	88	1769.257812	...	2018	5.961589	5.500000	6.500000	166.531250	132.0
Zambia	22.222222	18	29	65.222222	56	73	73.777778	66	83	1129.444444	...	2018	5.907407	5.583333	6.250000	162.000000	141.0
Zimbabwe	26.846154	22	34	69.769231	64	78	72.461538	66	83	2780.769231	...	2018	5.852564	5.500000	6.166667	164.000000	132.0

163 rows × 33 columns

Generar 2 jugadores ficticios con todos los campos del dataset y añadirlos al mismo.

Visualizarlos con .tail(). Correr nuevamente las estadísticas y compararlas.

In [ ]:
```
# Creo 2 nuevos jugadores con los campos del dataset creado en el anterior apartado
df_new_2=df_new
df_new_2 = df_new_2.append({'Age':27,'Overall':50,'Potential':52,'Value':1000,
'Wage':10,'International Reputation':1.4,'Skill Moves':2.0,'Joined':2014,
'Height':5.71,'Weight': 120,'Release Clause':100,'Nationality':'Colombia'},
ignore_index=True)
df_new_2 = df_new_2.append({'Age':26,'Overall':55,'Potential':57,'Value':1200,
'Wage':20,'International Reputation':2.4,'Skill Moves':3.4,'Joined':2018,
```

```
'Height':6.0,'Weight': 160,'Release Clause':1000,'Nationality':'Italy'},
ignore_index=True)
```

```
C:\Users\oscah\AppData\Local\Temp\ipykernel_5004\2207523662.py:3: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df_new_2 = df_new_2.append({'Age':27,'Overall':50,'Potential':52,'Value':1000,'Wage':10,'International Reputation':1.4,'Skill Moves':2.0,'Joined':2014,'Height':5.71,'Weight': 120,'Release Clause':100,'Nationality':'Colombia'},ignore_index=True)
C:\Users\oscah\AppData\Local\Temp\ipykernel_5004\2207523662.py:4: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df_new_2 = df_new_2.append({'Age':26,'Overall':55,'Potential':57,'Value':1200,'Wage':20,'International Reputation':2.4,'Skill Moves':3.4,'Joined':2018,'Height':6.0,'Weight': 160,'Release Clause':1000,'Nationality':'Italy'},
```

In [ ]:
```
df_new_2.tail()
#Aqui los identifico en las dos ultimas filas de mi dataframe.
```

Out[ ]:

	Age	Overall	Potential	Value	Wage	International Reputation	Skill Moves	Joined	Height	Weight	Release Clause	Nationality
17915	16	47	67	60.0	1.0	1.0	2.0	2017	5.666667	148.0	165.0	England
17916	17	47	66	60.0	1.0	1.0	2.0	2018	5.833333	154.0	143.0	England
17917	16	46	66	60.0	1.0	1.0	2.0	2018	5.833333	176.0	165.0	England
17918	27	50	52	1000.0	10.0	1.4	2.0	2014	5.710000	120.0	100.0	Colombia
17919	26	55	57	1200.0	20.0	2.4	3.4	2018	6.000000	160.0	1000.0	Italy

In [ ]:
```
# Para hacer la comparacion de las estadisticas descriptivas de las dos tablas,
#  creo dos tablas con dichos valores y uso despues
# la funcion compare para tener todos los datos en la misma tabla
# Inicialmente pense en usar la funcion merge pero investigando me di cuenta que
#  podia utilizar la funcion compare!

df_new_statistics = df_new.aggregate(['mean', 'min', 'max'])
```

```
C:\Users\oscah\AppData\Local\Temp\ipykernel_5004\1155129321.py:5: FutureWarning: ['Nationality'] did not aggregate successfully. If any error is raised this will raise in a future version of pandas. Drop these columns/ops to avoid this warning.
  df_new_statistics = df_new.aggregate(['mean', 'min', 'max'])
```

In [ ]:
```
df_new_2_statistics = df_new_2.aggregate(['mean', 'min', 'max'])
```

```
C:\Users\oscah\AppData\Local\Temp\ipykernel_5004\2849332022.py:1: FutureWarning: ['Nationality'] did not aggregate successfully.
If any error is raised this will raise in a future version of pandas. Drop these columns/ops to avoid this warning.
  df_new_2_statistics = df_new_2.aggregate(['mean', 'min', 'max'])
```

In [ ]:
```
df_new_statistics.compare(df_new_2_statistics,align_axis=0, keep_shape=True)

# En esta tabla es mucho mas evidente que solo en la media de todos los campos
#  hubo cambios. Pero ninguno significativo
# Yaque solo 2 datos no tienen un gran incidencia en una tabla con mas de 15000
# datos.
```

Out[ ]:

		Age	Overall	Potential	Value	Wage	International Reputation	Skill Moves	Joined	Height	Weight	Release Clause	Nationalit
mean	**self**	25.105257	66.236801	71.329334	2450.132909	9.883748	1.113908	2.362875	2016.427391	5.946753	165.968858	4585.060974	NaN
	other	25.105413	66.235268	71.327455	2449.982132	9.884319	1.113996	2.362913	2016.427344	5.946743	165.965960	4584.610632	NaN
min	**self**	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	other	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	**self**	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	other	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Generar una lambda que añada la zona horaria y la capital del país de nacionalidad

In [ ]:
```
# Retomo con el dataset original  'df'

import pycountry_convert

# para añadir la zona horario necesito tener el codigo de cada paìs.
#  Para obtenerlo, importo la libreria "pycountry_convert"
# y uso la funciòn "country_name_to_country_alpha2". Tuve que cambiar
#  el modo en el que el nombre de algunos paìses estaba
# escrito en el dataset ya que la libreria no los reconocìa.
```

In [ ]:
```
df['Nationality']= df['Nationality'].replace({'England':'United Kingdom',
'Wales':'United Kingdom','Bosnia Herzegovina':'Bosnia and Herzegovina',
'Korea Republic':'South Korea', 'Scotland':'United Kingdom',
```

```
'Central African Republicblic':'Central African Republic',
'Central African Rep.':'Central African Republic',
'DR Congo':'Congo','Republic of Ireland':'Ireland',
'FYR Macedonia':'Republic of North Macedonia','China PR':'China',
'Guinea Bissau':'Guinea-Bissau','Kosovo':'Serbia',
'Northern Ireland':'United Kingdom','Curacao':'Curaçao',
'Trinidad & Tobago':'Trinidad and Tobago',
'São Tomé & Príncipe':'Sao Tome and Principe',"Korea DPR":"North Korea",
'St Kitts Nevis':'Saint Kitts and Nevis',
'Antigua & Barbuda':'Antigua and Barbuda','St Lucia':'Saint Lucia'},
 regex = True)
```

In [ ]:
```
# Se crea la columna "code country"
df['code country']=df['Nationality'].apply
(lambda x:pycountry_convert.country_name_to_country_alpha2
(x, cn_name_format="default"))
```

In [ ]:
```
import pytz
from countryinfo import CountryInfo
```

In [ ]:
```
# Y ahora se procde con la creaciòn de la columna ZonaHoraria.

df['ZonaHoraria']=df['code country'].apply(lambda x:pytz.country_timezones[x])
```

In [ ]:
```
# Para la creaciòn de la columna "capital" uso la libreria "CountryInfo" con la
# funciòn capital.
# Para poder usarla debo nuevamente cambiar el nombre de algunos paises y quitar
# el de otros.
# Ya que la libreria no los reconoce o no son considerados en tal librerìa.
```

In [ ]:
```
# Estos son los 5 paìses que no estan presente en la librerìa. Por lo cual antes
#  de eliminarlos,
# guardo una copia de cada uno para despues concatenarlos en un solo data set
# para despues agregarles la capital manualmente.

df_montenegro=df.loc[df['Nationality']=='Montenegro']
df_Republic_of_North_Macedonia=df.loc[df['Nationality']=='Republic of North Macedonia']
df_Curaçao=df.loc[df['Nationality']=='Curaçao']
df_Palestine=df.loc[df['Nationality']=='Palestine']
df_andorra=df.loc[df['Nationality']=='Andorra']

df_concat = pd.concat(
```

```
[df_montenegro, df_Republic_of_North_Macedonia,df_Curaçao,df_Palestine,df_andorra],
axis=0)
df_concat['capital']=''
```

In [ ]:
```
# Agrego la capital de cada uno manualmente.

df_concat.loc[df["Nationality"]=='Montenegro', "capital"] = 'Podgorica'
df_concat.loc[df["Nationality"]=='Republic of North Macedonia', "capital"] ='Skopje'
df_concat.loc[df["Nationality"]=='Curaçao', "capital"] ='Willemstad'
df_concat.loc[df["Nationality"]=='Palestine', "capital"] ='Ramallah'
df_concat.loc[df["Nationality"]=='Andorra', "capital"] ='Andorra la Vella'
```

In [ ]:
```
# Los cancelo del dataset original "df" para que no me presente un "keyerror"
df = df.drop(df[df['Nationality']=='Montenegro'].index)
df = df.drop(df[df['Nationality']=='Republic of North Macedonia'].index)
df = df.drop(df[df['Nationality']=='Curaçao'].index)
df = df.drop(df[df['Nationality']=='Palestine'].index)
df = df.drop(df[df['Nationality']=='Andorra'].index)
```

In [ ]:
```
# Para estos paìses solo tuve que cambiar el modo en el cual estaban escritos
# en el dataset original para que fueran
# reconocidos por la libreria

df['Nationality']= df['Nationality'].replace({'Congo':'Democratic Republic of the Congo',
'Gambia':'The Gambia','Sao Tome and Principe':'São Tomé and Príncipe'})
```

In [ ]:
```
# Finalmente vuelvo al data set original para hallar la columna "capital".
df['capital']=df['Nationality'].apply(lambda x:CountryInfo(x).capital())
```

In [ ]:
```
df.sample(5)
```

Out[ ]:

	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Preferred Foot	...	Joined	Contract Valid Until	Height	Weight	Release Clause
5828	229456	L. Dickmann	21	Italy	69	78	SPAL	1400.0	5.0	Right	...	2014	2020-01-01	5.833333	154.0	2800.0
12550	216746	M. Koj	24	Poland	63	67	Górnik Zabrze	400.0	1.0	Right	...	2017	2019-01-01	6.166667	179.0	570.0
17282	237889	J. Keaney	19	Ireland	53	66	Sligo Rovers	110.0	1.0	Right	...	2017	2019-01-01	6.083333	146.0	239.0
11780	239592	J. Macías	18	Mexico	64	75	Guadalajara	725.0	4.0	Right	...	2016	2020-01-01	5.833333	161.0	1400.0
2005	162280	A. Mierzejewski	31	Poland	75	75	Changchun Yatai FC	6000.0	12.0	Left	...	2018	2019-01-01	5.750000	161.0	12900.0

5 rows × 23 columns

In [ ]:
```
# Finalmente concateno los dos dataframes. El original con el que le tuve que
# escribir las capitales manualmente.

df_final = pd.concat([df, df_concat], axis=0)
```

In [ ]:
```
df_final.sample(20)
```

Out[ ]:

	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Preferred Foot	...	Joined	Contract Valid Until	Height	Weight	
4468	188094	J. McLaughlin	29	United Kingdom	71	72	Sunderland	1800.0	8.0	Right	...	2018	2020-01-01	6.166667	183.0	3
5994	226430	S. Hakšabanovič	19	Montenegro	69	79	Málaga CF	1600.0	15.0	Right	...	2016	2019-06-30	5.666667	163.0	4
11453	226716	H. Mascorro	21	Mexico	64	74	Club León	725.0	5.0	Right	...	2016	2019-05-31	5.583333	150.0	4
13988	138091	A. Egholm	35	Denmark	61	61	SønderjyskE	60.0	2.0	Right	...	2017	2018-01-01	6.166667	190.0	
4322	176772	A. Oukidja	29	Algeria	71	72	FC Metz	1800.0	4.0	Right	...	2018	2021-01-01	6.000000	174.0	3
1383	204156	S. Sanogo	29	Ivory Coast	76	76	BSC Young Boys	6000.0	22.0	Left	...	2014	2021-01-01	6.000000	179.0	9
4087	241720	André Pereira	23	Portugal	71	78	FC Porto	3400.0	8.0	Left	...	2017	2019-01-01	6.166667	168.0	7
4195	190555	Mossa	29	Spain	71	71	Real Oviedo	1800.0	7.0	Left	...	2017	2019-01-01	5.833333	159.0	2
9097	188522	Fábinho	33	Brazil	66	66	Philadelphia Union	250.0	2.0	Left	...	2016	2021-01-01	5.583333	150.0	
11828	240371	A. Jakubech	21	Slovakia	64	75	LOSC Lille	600.0	3.0	Right	...	2017	2022-01-01	6.166667	181.0	1
2393	187043	S. Johansen	27	Norway	74	75	Fulham	6500.0	45.0	Left	...	2016	2019-01-01	6.000000	172.0	12
15897	240584	B. Verbong	18	Netherlands	58	70	VVV-Venlo	160.0	1.0	Right	...	2016	2019-01-01	6.083333	176.0	
4098	191034	Cyriac	27	Ivory Coast	71	71	Sivasspor	2500.0	11.0	Right	...	2017	2020-01-01	5.666667	165.0	5

	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Preferred Foot	...	Joined	Contract Valid Until	Height	Weight	
5493	206571	M. Maddison	24	United Kingdom	70	73	Peterborough United	2300.0	6.0	Left	...	2014	2020-01-01	5.916667	157.0	4
1990	172768	A. Surman	31	United Kingdom	75	75	Bournemouth	5500.0	53.0	Left	...	2014	2019-01-01	5.833333	161.0	10
2824	212079	M. Pérez	25	Mexico	73	78	Guadalajara	4500.0	26.0	Right	...	2012	2020-01-01	5.583333	165.0	8
13636	238576	Jeong Tae Wook	21	South Korea	62	75	Jeju United FC	525.0	1.0	Right	...	2018	2021-01-01	6.333333	194.0	
9282	205460	C. Schoppenhauer	26	Germany	66	68	FC St. Pauli	650.0	5.0	Right	...	2017	2019-01-01	6.083333	170.0	1
7659	219680	C. Hountondji	24	Benin	67	72	New York City FC	875.0	2.0	Right	...	2018	2023-01-01	6.416667	179.0	1
11133	221001	U. Saltnes	25	Norway	64	67	FK Bodø/Glimt	550.0	1.0	Right	...	2012	2020-01-01	6.166667	172.0	