FIFA 21 messy, raw dataset for cleaning/ exploring

- Convert the height and weight columns to numerical forms
- Remove the unnecessary newline characters from all columns that have them.
- Based on the 'Joined' column, check which players have been playing at a club for more than 10 years!
- 'Value', 'Wage' and "Release Clause' are string columns. Convert them to numbers. For eg, "M" in value column is Million, so multiply the row values by 1,000,000, etc.
- Some columns have 'star' characters. Strip those columns of these stars and make the columns numerical
- Which players are highly valuable but still underpaid (on low wages)?

In [ ]:
```
import pandas as pd
fifa_df = pd.read_csv(r'F:\Aprendizado\Projetos\Fifa 21\fifa21_raw_data_v2.csv',
```

In [ ]:
```
fifa_df.head(5)
```

Out[ ]:

	ID	Name	LongName	photoUrl	
0	158023	L. Messi	Lionel Messi	https://cdn.sofifa.com/players/158/023/21_60.png	http://s
1	20801	Cristiano Ronaldo	C. Ronaldo dos Santos Aveiro	https://cdn.sofifa.com/players/020/801/21_60.png	h
2	200389	J. Oblak	Jan Oblak	https://cdn.sofifa.com/players/200/389/21_60.png	http:
3	192985	K. De Bruyne	Kevin De Bruyne	https://cdn.sofifa.com/players/192/985/21_60.png	http://s
4	190871	Neymar Jr	Neymar da Silva Santos Jr.	https://cdn.sofifa.com/players/190/871/21_60.png	http://so

5 rows × 77 columns

In [ ]:
```
fifa_df.columns.tolist()
```

Out[ ]:
```
['ID',
 'Name',
 'LongName',
 'photoUrl',
 'playerUrl',
 'Nationality',
 'Age',
 '↓OVA',
 'POT',
 'Club',
 'Contract',
 'Positions',
 'Height',
 'Weight',
 'Preferred Foot',
 'BOV',
 'Best Position',
 'Joined',
 'Loan Date End',
 'Value',
 'Wage',
 'Release Clause',
 'Attacking',
 'Crossing',
 'Finishing',
 'Heading Accuracy',
 'Short Passing',
 'Volleys',
 'Skill',
 'Dribbling',
 'Curve',
 'FK Accuracy',
 'Long Passing',
 'Ball Control',
 'Movement',
 'Acceleration',
 'Sprint Speed',
 'Agility',
 'Reactions',
 'Balance',
 'Power',
 'Shot Power',
 'Jumping',
 'Stamina',
 'Strength',
 'Long Shots',
 'Mentality',
 'Aggression',
 'Interceptions',
 'Positioning',
 'Vision',
 'Penalties',
 'Composure',
 'Defending',
 'Marking',
 'Standing Tackle',
 'Sliding Tackle',
 'Goalkeeping',
 'GK Diving',
 'GK Handling',
```

```
 'GK Kicking',
 'GK Positioning',
 'GK Reflexes',
 'Total Stats',
 'Base Stats',
 'W/F',
 'SM',
 'A/W',
 'D/W',
 'IR',
 'PAC',
 'SHO',
 'PAS',
 'DRI',
 'DEF',
 'PHY',
 'Hits']
```

In [ ]:
```
fifa_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18979 entries, 0 to 18978
Data columns (total 77 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   ID                18979 non-null  object
 1   Name              18979 non-null  object
 2   LongName          18979 non-null  object
 3   photoUrl          18979 non-null  object
 4   playerUrl         18979 non-null  object
 5   Nationality       18979 non-null  object
 6   Age               18979 non-null  object
 7   ↓OVA              18979 non-null  object
 8   POT               18979 non-null  object
 9   Club              18979 non-null  object
 10  Contract          18979 non-null  object
 11  Positions         18979 non-null  object
 12  Height            18979 non-null  object
 13  Weight            18979 non-null  object
 14  Preferred Foot    18979 non-null  object
 15  BOV               18979 non-null  object
 16  Best Position     18979 non-null  object
 17  Joined            18979 non-null  object
 18  Loan Date End     1013 non-null   object
 19  Value             18979 non-null  object
 20  Wage              18979 non-null  object
 21  Release Clause    18979 non-null  object
 22  Attacking         18979 non-null  object
 23  Crossing          18979 non-null  object
 24  Finishing         18979 non-null  object
 25  Heading Accuracy  18979 non-null  object
 26  Short Passing     18979 non-null  object
 27  Volleys           18979 non-null  object
 28  Skill             18979 non-null  object
 29  Dribbling         18979 non-null  object
 30  Curve             18979 non-null  object
 31  FK Accuracy       18979 non-null  object
 32  Long Passing      18979 non-null  object
 33  Ball Control      18979 non-null  object
 34  Movement          18979 non-null  object
 35  Acceleration      18979 non-null  object
 36  Sprint Speed      18979 non-null  object
 37  Agility           18979 non-null  object
 38  Reactions         18979 non-null  object
 39  Balance           18979 non-null  object
 40  Power             18979 non-null  object
 41  Shot Power        18979 non-null  object
 42  Jumping           18979 non-null  object
 43  Stamina           18979 non-null  object
 44  Strength          18979 non-null  object
 45  Long Shots        18979 non-null  object
 46  Mentality         18979 non-null  object
 47  Aggression        18979 non-null  object
 48  Interceptions     18979 non-null  object
 49  Positioning       18979 non-null  object
 50  Vision            18979 non-null  object
 51  Penalties         18979 non-null  object
 52  Composure         18979 non-null  object
 53  Defending         18979 non-null  object
 54  Marking           18979 non-null  object
```

```
 55  Standing Tackle   18979 non-null  object
 56  Sliding Tackle    18979 non-null  object
 57  Goalkeeping       18979 non-null  object
 58  GK Diving         18979 non-null  object
 59  GK Handling       18979 non-null  object
 60  GK Kicking        18979 non-null  object
 61  GK Positioning    18979 non-null  object
 62  GK Reflexes       18979 non-null  object
 63  Total Stats       18979 non-null  object
 64  Base Stats        18979 non-null  object
 65  W/F               18979 non-null  object
 66  SM                18979 non-null  object
 67  A/W               18979 non-null  object
 68  D/W               18979 non-null  object
 69  IR                18979 non-null  object
 70  PAC               18979 non-null  object
 71  SHO               18979 non-null  object
 72  PAS               18979 non-null  object
 73  DRI               18979 non-null  object
 74  DEF               18979 non-null  object
 75  PHY               18979 non-null  object
 76  Hits              16384 non-null  object
dtypes: object(77)
memory usage: 11.1+ MB
```

In [ ]:
```
fifa_df.Height.unique()
```

Out[ ]:
```
array(['170cm', '187cm', '188cm', '181cm', '175cm', '184cm', '191cm',
       '178cm', '193cm', '185cm', '199cm', '173cm', '168cm', '176cm',
       '177cm', '183cm', '180cm', '189cm', '179cm', '195cm', '172cm',
       '182cm', '186cm', '192cm', '165cm', '194cm', '167cm', '196cm',
       '163cm', '190cm', '174cm', '169cm', '171cm', '197cm', '200cm',
       '166cm', '6\'2"', '164cm', '198cm', '6\'3"', '6\'5"', '5\'11"',
       '6\'4"', '6\'1"', '6\'0"', '5\'10"', '5\'9"', '5\'6"', '5\'7"',
       '5\'4"', '201cm', '158cm', '162cm', '161cm', '160cm', '203cm',
       '157cm', '156cm', '202cm', '159cm', '206cm', '155cm'], dtype=object)
```

In [ ]:
```
# Convert the height and weight columns to numerical forms
def convert_to_cm(height):
    if 'cm' in height:
        return float(height.replace('cm', ''))
    elif '\'' in height:
        feet, inches = map(int, height.replace('"', '').split('\''))
        return (feet * 12 + inches) * 2.54
    else:
        raise ValueError("Wrong Format Number: {}".format(height))
```

In [ ]:
```
fifa_df['Height'] = fifa_df['Height'].apply(convert_to_cm)
```

In [ ]:
```
display(fifa_df['Height'])
```

```
0        170.0
1        187.0
2        188.0
3        181.0
4        175.0
         ...
18974    178.0
18975    175.0
18976    179.0
18977    175.0
18978    188.0
Name: Height, Length: 18979, dtype: float64
```

In [ ]:
```
# extracting \n in Clubs Name
fifa_df['Club'] = fifa_df['Club'].str.replace('\n', '')
```

In [ ]:
```
fifa_df.Weight.unique()
```

Out[ ]:
```
array(['72kg', '83kg', '87kg', '70kg', '68kg', '80kg', '71kg', '91kg',
       '73kg', '85kg', '92kg', '69kg', '84kg', '96kg', '81kg', '82kg',
       '75kg', '86kg', '89kg', '74kg', '76kg', '64kg', '78kg', '90kg',
       '66kg', '60kg', '94kg', '79kg', '67kg', '65kg', '59kg', '61kg',
       '93kg', '88kg', '97kg', '77kg', '62kg', '63kg', '95kg', '100kg',
       '58kg', '183lbs', '179lbs', '172lbs', '196lbs', '176lbs', '185lbs',
       '170lbs', '203lbs', '168lbs', '161lbs', '146lbs', '130lbs',
       '190lbs', '174lbs', '148lbs', '165lbs', '159lbs', '192lbs',
       '181lbs', '139lbs', '154lbs', '157lbs', '163lbs', '98kg', '103kg',
       '99kg', '102kg', '56kg', '101kg', '57kg', '55kg', '104kg', '107kg',
       '110kg', '53kg', '50kg', '54kg', '52kg'], dtype=object)
```

In [ ]:
```
# fixing Weight column
def convert_weight_to_kg(weight):
    if 'kg' in weight:
        return float(weight.replace('kg', ''))
    elif 'lbs' in weight:
        pounds = float(weight.replace('lbs', ''))
        return pounds * 0.45359237  # 1 libra é aproximadamente igual a 0,453592
    else:
        raise ValueError("Formato de peso inválido: {}".format(weight))
```

In [ ]:
```
fifa_df['Weight'] = fifa_df['Weight'].apply(convert_weight_to_kg)
```

In [ ]:
```
print(fifa_df['Weight'])
```

```
0        72.0
1        83.0
2        87.0
3        70.0
4        68.0
         ...
18974    66.0
18975    65.0
18976    74.0
18977    69.0
18978    75.0
Name: Weight, Length: 18979, dtype: float64
```

In [ ]:
```
# Fixing Money in three columns
def convert_value_to_euro(value):
    if 'M' in value:
```

```
        return float(value.replace('€', '').replace('M', '')) * 1000000
    elif 'K' in value:
        return float(value.replace('€', '').replace('K', '')) * 1000
    else:
        return float(value.replace('€', ''))
```

In [ ]:
```
fifa_df['Value'] = fifa_df['Value'].apply(convert_value_to_euro)
fifa_df['Release Clause'] = fifa_df['Release Clause'].apply(convert_value_to_eur
fifa_df['Wage'] = fifa_df['Wage'].apply(convert_value_to_euro)
```

In [ ]:
```
# separeting Date in Year, Month and Day of
fifa_df.Joined
```

Out[ ]:
```
0         Jul 1, 2004
1        Jul 10, 2018
2        Jul 16, 2014
3        Aug 30, 2015
4         Aug 3, 2017
             ...
18974    Jul 13, 2018
18975     Aug 1, 2020
18976     Mar 8, 2019
18977    Sep 22, 2020
18978    Jul 29, 2019
Name: Joined, Length: 18979, dtype: object
```

In [ ]:
```
fifa_df['Joined'] = pd.to_datetime(fifa_df['Joined'])

fifa_df['Joined Year'] = fifa_df['Joined'].dt.year
fifa_df['Joined Month'] = fifa_df['Joined'].dt.month
fifa_df['Joined Day'] = fifa_df['Joined'].dt.day
```

In [ ]:
```
new_columns = ['ID','Name','LongName','Nationality','Age','↓OVA','POT','Club','C
           'BOV','Best Position','Joined','Loan Date End','Value','Wage','Releas
           'Short Passing','Volleys','Skill','Dribbling','Curve','FK Accuracy','
           'Agility','Reactions','Balance','Power','Shot Power','Jumping','Stami
           'Positioning','Vision','Penalties','Composure','Defending','Marking',
           'GK Kicking','GK Positioning','GK Reflexes','Total Stats','Base Stats
```

In [ ]:
```
fifa_df2 = fifa_df[new_columns]
fifa_df2.rename(columns = {'↓OVA':'OVA'},inplace=True)
fifa_df2.head()

fifa_new_df = fifa_df2.drop_duplicates()
```

```
C:\Users\Fabio Premero\AppData\Local\Temp\ipykernel_8068\192569946.py:2: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  fifa_df2.rename(columns = {'↓OVA':'OVA'},inplace=True)
```

In [ ]:
```
# Which players are highly valuable but still underpaid (on low wages)? (hint: s
import matplotlib.pyplot as plt

df = fifa_new_df[['Wage','Value']]

plt.figure(figsize=(15, 6))
```

```
plt.scatter(df['Wage'], df['Value'], color='blue', alpha=0.7)
plt.title('Scatter Plot: Wage vs. Value')
plt.xlabel('Wage')
plt.ylabel('Value')
plt.grid(True)
plt.show()
```