

Importing all the necessary libraries

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

```
pd.set_option('display.max_columns',None)
```

```
# Reading the csv file in to the data frame
fifa_df = pd.read_csv('data.csv')
```

```
fifa_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18944 entries, 0 to 18943
Columns: 106 entries, sofifa_id to rb
dtypes: float64(18), int64(44), object(44)
memory usage: 15.3+ MB
```

```
fifa_df.head()
```

	sofifa_id	player_url	short_name	long_name	age	do
0	158023	https://sofifa.com/player/158023/lionel-messi/...	L. Messi	Lionel Andrés Messi Cuccittini	33	1987 06-2
1	20801	https://sofifa.com/player/20801/c-ronaldo-dos-...	Cristiano Ronaldo	Cristiano Ronaldo dos Santos Aveiro	35	1985 02-0
2	200389	https://sofifa.com/player/200389/jan-oblak/210002	J. Oblak	Jan Oblak	27	1993 01-0
3	188545	https://sofifa.com/player/188545/robert-lewand-...	R. Lewandowski	Robert Lewandowski	31	1988 08-2
4	190871	https://sofifa.com/player/190871/neymar-da-sil-...	Neymar Jr	Neymar da Silva Santos Júnior	28	1992 02-0

```
# Analysing the description of the overall data
fifa_df.shape
```

```
➡ (18944, 106)
```

```
fifa_df.describe()
```

➡		sofifa_id	age	height_cm	weight_kg	league_rank	overall
	count	18944.000000	18944.000000	18944.000000	18944.000000	18719.000000	18944.000000
	mean	226242.402872	25.225823	181.190773	75.016892	1.357070	65.677787
	std	27171.091056	4.697354	6.825672	7.057140	0.739327	7.002278
	min	41.000000	16.000000	155.000000	50.000000	1.000000	47.000000
	25%	210030.500000	21.000000	176.000000	70.000000	1.000000	61.000000
	50%	232314.500000	25.000000	181.000000	75.000000	1.000000	66.000000
	75%	246760.250000	29.000000	186.000000	80.000000	1.000000	70.000000
	max	258970.000000	53.000000	206.000000	110.000000	4.000000	93.000000

```
fifa_df.duplicated().sum() #--> No Duplicated Values in rows
```

```
➡ np.int64(0)
```

```
# Maintaining a Copy of my Data Cell
df_main= fifa_df.copy()
```

```
fifa_df.columns
```

```
➡ Index(['sofifa_id', 'player_url', 'short_name', 'long_name', 'age', 'dob',
        'height_cm', 'weight_kg', 'nationality', 'club_name',
        ...,
        'lwb', 'ldm', 'cdm', 'rdm', 'rwb', 'lb', 'lcb', 'cb', 'rcb', 'rb'],
        dtype='object', length=106)
```

Three Columns Related to the GoalKeeping

```
fifa_df.drop(columns=['sofifa_id','player_url','dob','body_type','team_jersey_number','con
```

```
fifa_df.shape
```

```
➡ (18944, 87)
```

```
fifa_df.head()
```



	short_name	age	height_cm	weight_kg	nationality	club_name	league_name	le
0	L. Messi	33	170	72	Argentina	FC Barcelona	Spain Primera Division	
1	Cristiano Ronaldo	35	187	83	Portugal	Juventus	Italian Serie A	
2	J. Oblak	27	188	87	Slovenia	Atlético Madrid	Spain Primera Division	
3	R. Lewandowski	31	184	80	Poland	FC Bayern München	German 1. Bundesliga	
4	Neymar Jr	28	175	68	Brazil	Paris Saint-Germain	French Ligue 1	

Combining the GoalKeeper Attributes

```
# Combining the GoalKeeper Attributes
fifa_df['gk_average']=(fifa_df['gk_diving']+fifa_df['gk_handling']+fifa_df['gk_kicking']+fifa_df['gk_reflexes']+fifa_df['gk_speed'])
# Deleting the Unecessary goal attributes
fifa_df.drop(columns=['gk_diving','gk_handling','gk_kicking','gk_reflexes','gk_speed','gk_average'],inplace=True)
```

Combining the Attacking Attributes

```
fifa_df['attacking_average']=(fifa_df['attacking_crossing']+fifa_df['attacking_finishing']+fifa_df['attacking_heading_accuracy']+fifa_df['attacking_dribbling']+fifa_df['attacking_volleys'])
# Dropping the same columns from the Data Set
fifa_df.drop(columns=['attacking_crossing','attacking_finishing','attacking_heading_accuracy','attacking_dribbling','attacking_volleys'],inplace=True)
```

```
fifa_df.shape
```



```
(18944, 78)
```

Height and the Weight Parameters can be Combined together using BMI kg/m^2

```
# Calculating the BMI Using the Proper Standards
fifa_df["BMI"]=fifa_df['weight_kg']/((fifa_df['height_cm']*0.01)**2)
# Dropping the Attributes
fifa_df.drop(columns=['weight_kg','height_cm'],inplace=True)
fifa_df.shape
```

↔ (18944, 77)

```
fifa_df.head()
```

↔

	short_name	age	nationality	club_name	league_name	league_rank	overall	po
0	L. Messi	33	Argentina	FC Barcelona	Spain Primera Division	1.0	93	
1	Cristiano Ronaldo	35	Portugal	Juventus	Italian Serie A	1.0	92	
2	J. Oblak	27	Slovenia	Atlético Madrid	Spain Primera Division	1.0	91	
3	R. Lewandowski	31	Poland	FC Bayern München	German 1. Bundesliga	1.0	91	
4	Neymar Jr	28	Brazil	Paris Saint-Germain	French Ligue 1	1.0	91	

Combine Skill Parameters

```
# Combining the Skill Movement and the Power Parameters
# Average can be Considered cause all the parameters lie along the same range

fifa_df['skill_avg']=(fifa_df['skill_ball_control']+fifa_df['skill_curve']+fifa_df['skill_
# Dropping the Columns
fifa_df.drop(columns=['skill_ball_control','skill_curve','skill_dribbling','skill_fk_accur
fifa_df.shape
```

↔ (18944, 73)

Combine Mentality as one as all attributes has the same scaling or range of values

```
fifa_df['Mentality_avg']=(fifa_df['mentality_aggression']+fifa_df['mentality_composure']+f
# Dropping all the mentality columns
fifa_df.drop(columns=['mentality_aggression','mentality_composure','mentality_interception
```

```
fifa_df.shape
```

↔ (18944, 68)

Combining Movements

```
fifa_df.head()
```



	short_name	age	nationality	club_name	league_name	league_rank	overall	po
0	L. Messi	33	Argentina	FC Barcelona	Spain Primera Division	1.0	93	
1	Cristiano Ronaldo	35	Portugal	Juventus	Italian Serie A	1.0	92	
2	J. Oblak	27	Slovenia	Atlético Madrid	Spain Primera Division	1.0	91	
3	R. Lewandowski	31	Poland	FC Bayern München	German 1. Bundesliga	1.0	91	
4	Neymar Jr	28	Brazil	Paris Saint-Germain	French Ligue 1	1.0	91	

```
fifa_df['Movements_avg']=(fifa_df['movement_acceleration']+fifa_df['movement_agility']+fifa_df['movement_balance'])
fifa_df.drop(columns=['movement_acceleration','movement_agility','movement_balance','movement_avg'],inplace=True)
```

Combining Powers

```
fifa_df['Power_avg']=(fifa_df['power_jumping']+fifa_df['power_long_shots']+fifa_df['power_shot_power'])
# Dropping the Columns From the Data Sets
fifa_df.drop(columns=['power_jumping','power_long_shots','power_shot_power','power_stamina'],inplace=True)
```

```
fifa_df.shape
```



```
(18944, 60)
```

```
#Checking if all the Parameters of the Defending are defined on the same scale
print(fifa_df['defending_marking'].max())
print(fifa_df['defending_sliding_tackle'].max())
print(fifa_df['defending_standing_tackle'].max())
# Hence it can be Proved that all the Parameters in the Defending Sections are Defined on the same scale
#and can be combined using the mean or the weighted mean
fifa_df['defending_avg']=(fifa_df['defending_marking']+fifa_df['defending_sliding_tackle']+fifa_df['defending_standing_tackle'])
# Dropping the Columns from the Data Set
fifa_df.drop(columns=['defending_marking','defending_sliding_tackle','defending_standing_tackle'],inplace=True)
```

nan
93
90

	short_name	age	nationality	club_name	league_name	league_rank	overall
0	L. Messi	33	Argentina	FC Barcelona	Spain Primera Division	1.0	93
1	Cristiano Ronaldo	35	Portugal	Juventus	Italian Serie A	1.0	92
2	J. Oblak	27	Slovenia	Atlético Madrid	Spain Primera Division	1.0	91
3	R. Lewandowski	31	Poland	FC Bayern München	German 1. Bundesliga	1.0	91
4	Neymar Jr	28	Brazil	Paris Saint-Germain	French Ligue 1	1.0	91
...
18939	K. Angulo	24	Colombia	América de Cali	Colombian Liga Postobón	1.0	47
18940	Zhang Mengxuan	21	China PR	Chongqing Dangdai Lifan FC SWM Team	Chinese Super League	1.0	47
18941	Wang Zhenghao	20	China PR	Tianjin TEDA FC	Chinese Super League	1.0	47
18942	Chen Zitong	23	China PR	Shijiazhuang Ever Bright F.C.	Chinese Super League	1.0	47
18943	Song Yue	28	China PR	Tianjin TEDA FC	Chinese Super League	1.0	47

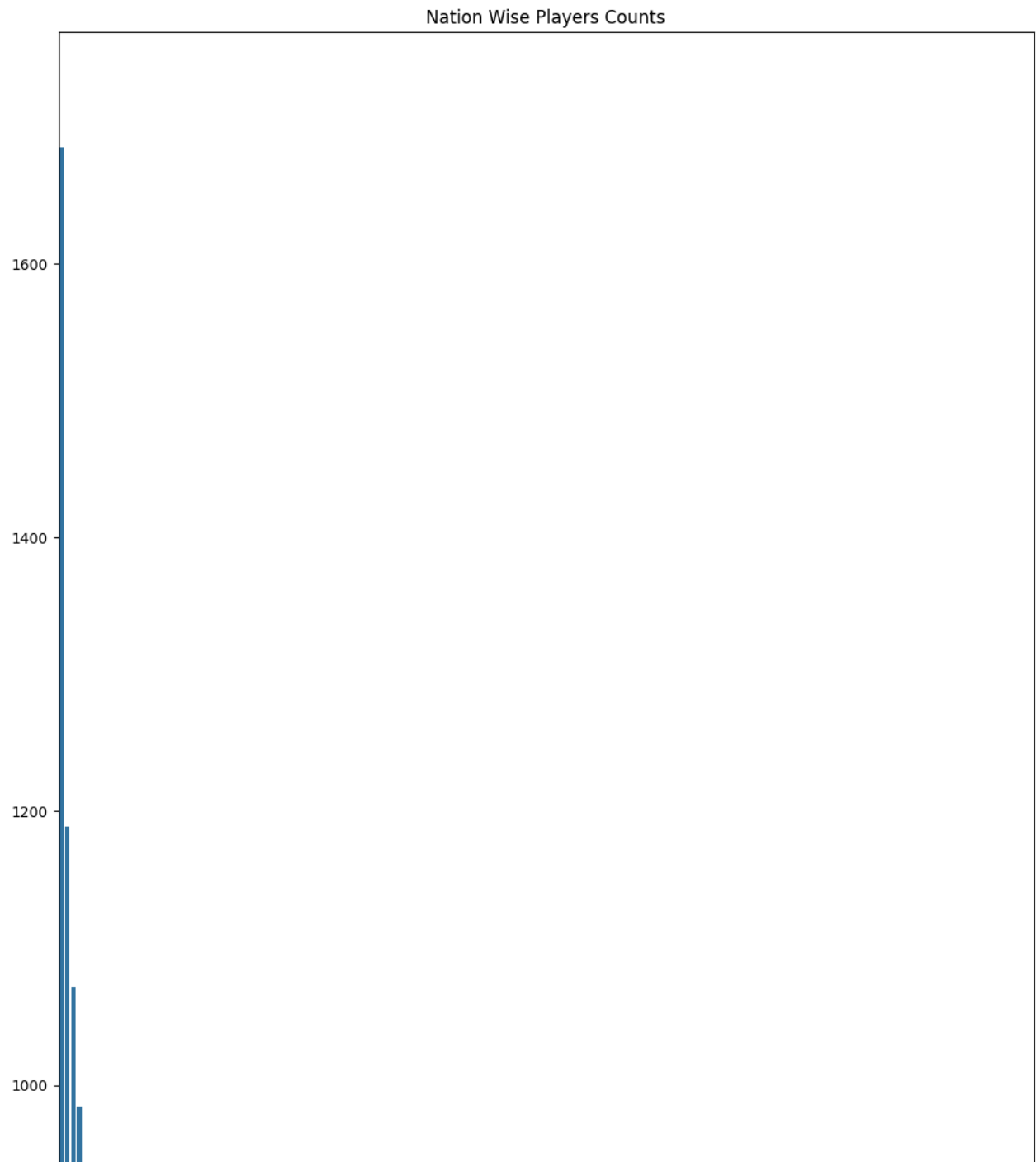
18944 rows x 58 columns

value_counts() It is used to calculate the occurrence of each and every element in the dataframe or series`

```
fifa_df['nationality'].value_counts()  
#Total Unique Contries are 163
```

```
# Creating New Series with Countries more that 850 Value Counts and Plotting it  
countries_series=fifa_df['nationality'].value_counts()
```

```
# Now Drawing a Barplot --> Using Seaborn  
mp.figure(figsize = (12, 30))  
sns.barplot(countries_series)  
mp.title("Nation Wise Players Counts");  
mp.show()
```



✓ Analysing the Goalkeeper Data

```
(fifa_df.team_position == 'GK').sum()
```

```
⇒ np.int64(681)
```

```
fifa_df.shape
```

```
⇒ (18944, 61)
```

```
# The Below Method will fetch all the indices for the goalkeepers
```

```
goal_keepers=[]
```

```
for i in range(0,18944):
```

```
    if(fifa_df.at[i,'team_position']=='GK'):
```

```
        goal_keepers.append(i)
```

```
    else:
```

```
        continue
```

```
len(goal_keepers) # Here we have successfully seperated the data of all the Goal Keepers
```

```
⇒ 681
```

```
# 681 players are Goalkeeper & Creating their seperate data frame to analyse them
```

```
goalkeepers_df =fifa_df.iloc[goal_keepers]
```

```
type(goalkeepers_df)
```

```
⇒ pandas.core.frame.DataFrame
```

Successfully Seperated the Goalkeeper Data

```
goalkeepers_df.head()
```




	short_name	age	nationality	club_name	league_name	league_rank	overall	p
2	J. Oblak	27	Slovenia	Atlético Madrid	Spain Primera Division	1.0	91	
7	M. ter Stegen	28	Germany	FC Barcelona	Spain Primera Division	1.0	90	
9	Alisson	27	Brazil	Liverpool	English Premier League	1.0	90	
12	T. Courtois	28	Belgium	Real Madrid	Spain Primera Division	1.0	89	
16	M. Neuer	34	Germany	FC Bayern München	German 1. Bundesliga	1.0	89	

```
# dropping all the unnecessary columns from goalkeepers_df
goalkeepers_df.drop(columns=['pace','shooting','passing','dribbling','defending','physic'])
```



C:\Users\Sian\AppData\Local\Temp\ipykernel_6732\3940782774.py:2: SettingWithCopyWarning: 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/stable/10min/5min.html#copy-on-write>
goalkeepers_df.drop(columns=['pace','shooting','passing','dribbling','defending'])

```
goalkeepers_df.columns
```



```
Index(['short_name', 'age', 'nationality', 'club_name', 'league_name',
       'league_rank', 'overall', 'potential', 'value_eur', 'wage_eur',
       'player_positions', 'international_reputation', 'skill_moves',
       'work_rate', 'real_face', 'team_position', 'nation_position',
       'player_traits', 'defending_marking', 'defending_standing_tackle',
       'defending_sliding_tackle', 'ls', 'st', 'rs', 'lw', 'lf', 'cf', 'rf',
       'rw', 'lam', 'cam', 'ram', 'lm', 'lcm', 'cm', 'rcm', 'rm', 'lwb', 'ldm',
       'cdm', 'rdm', 'rwb', 'lb', 'lcb', 'cb', 'rcb', 'rb', 'gk_average',
       'attacking_average', 'BMI', 'skill_avg', 'Mentality_avg',
       'Movements_avg', 'Power_avg', 'defending_avg'],
      dtype='object')
```

```
goalkeepers_df['goalk_df_avg']=(goalkeepers_df['defending_standing_tackle']+goalkeepers_df['defending_sliding_tackle'])
goalkeepers_df.drop(columns=['defending_standing_tackle','defending_sliding_tackle'])
```

➡ C:\Users\Sian\AppData\Local\Temp\ipykernel_6732\2462242937.py:1: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html#copy-on-write>
goalkeepers_df['goalk_df_avg']=(goalkeepers_df['defending_standing_tackle']+goalkeepers_df['standing_tackle'])

	short_name	age	nationality	club_name	league_name	league_rank	overall
2	J. Oblak	27	Slovenia	Atlético Madrid	Spain Primera Division	1.0	91
7	M. ter Stegen	28	Germany	FC Barcelona	Spain Primera Division	1.0	90
9	Alisson	27	Brazil	Liverpool	English Premier League	1.0	90
12	T. Courtois	28	Belgium	Real Madrid	Spain Primera Division	1.0	89
16	M. Neuer	34	Germany	FC Bayern München	German 1. Bundesliga	1.0	89
...
17293	T. Savin	19	England	Accrington Stanley	English League One	3.0	56
17604	Han Jiaqi	20	China PR	Guangzhou R&F FC	Chinese Super League	1.0	55
17714	R. Strączek	21	Poland	Stal Mielec	Polish T-Mobile Ekstraklasa	1.0	55
17749	J. Brady	23	Republic of Ireland	Shelbourne FC	Rep. Ireland Airtricity League	1.0	55
18510	W. Jiménez	30	Venezuela	Llaneros de Guanare	Venezuelan Primera División	1.0	52

681 rows x 53 columns

Analysing all the properties Important to Goalkeepers

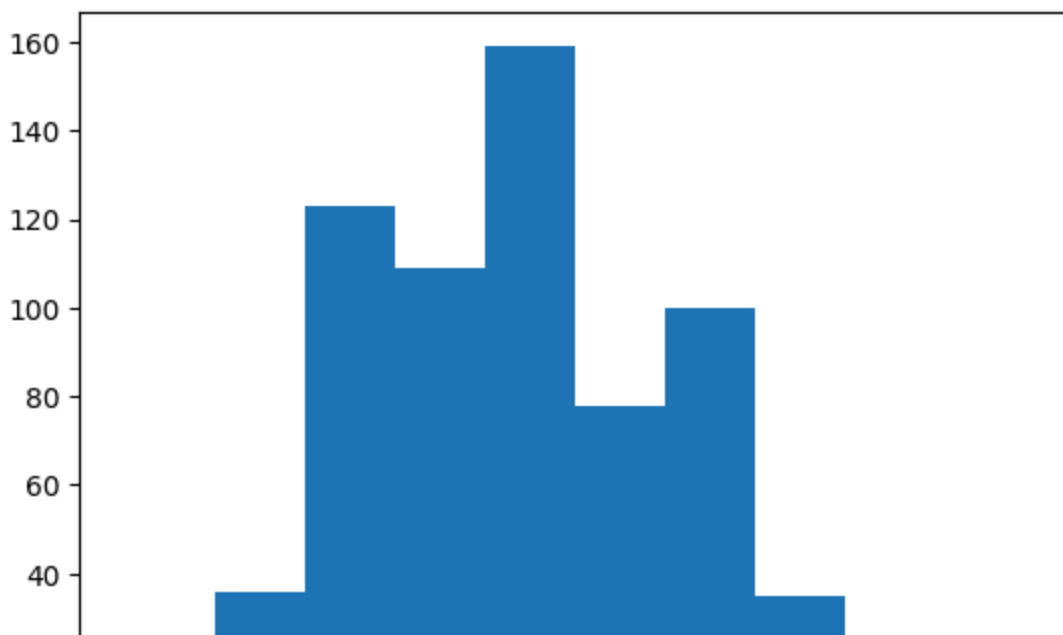
```
goalkeepers_df['age'].describe()
# The Mean age for all the GoalKeepers is 28
# Counting the Number of Distribution in Between
# 18-28 age group goal keepers
print((goalkeepers_df.age>=18).sum()) #--> 680
print((goalkeepers_df.age<=28).sum())#--> 347
print((goalkeepers_df.age>28).sum())#--> 334
print(goalkeepers_df.shape)
```

```
⇒ 681
   347
   334
   (681, 55)
```

Age Analysis of Goalkeepers Data

```
mp.hist(goalkeepers_df['age'])
```

```
⇒ (array([ 21.,  36., 123., 109., 159.,  78., 100.,  35.,  16.,   4.]),
   array([18. , 20.5, 23. , 25.5, 28. , 30.5, 33. , 35.5, 38. , 40.5, 43. ]),
   <BarContainer object of 10 artists>)
```



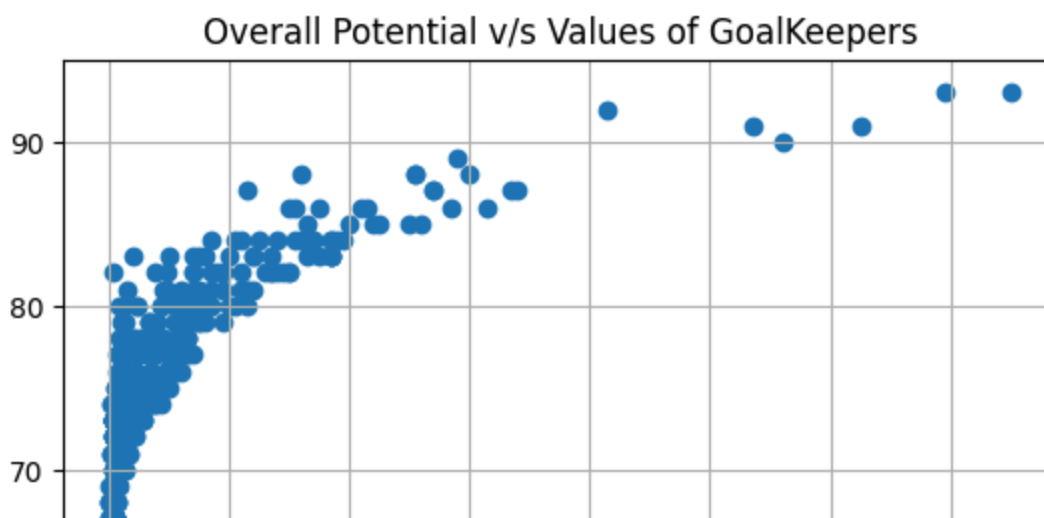
```
goalkeepers_df.head()
```



	short_name	age	nationality	club_name	league_name	league_rank	overall	p
2	J. Oblak	27	Slovenia	Atlético Madrid	Spain Primera Division	1.0	91	
7	M. ter Stegen	28	Germany	FC Barcelona	Spain Primera Division	1.0	90	
9	Alisson	27	Brazil	Liverpool	English Premier League	1.0	90	
12	T. Courtois	28	Belgium	Real Madrid	Spain Primera Division	1.0	89	
16	M. Neuer	34	Germany	FC Bayern München	German 1. Bundesliga	1.0	89	

Overall Potential v/s Values Chart

```
x_=list(goalkeepers_df['value_eur'])
y_=list(goalkeepers_df['potential'])
mp.title('Overall Potential v/s Values of GoalKeepers')
mp.scatter(x_,y_)
mp.grid(True)
```



```
print("The Final Size of the Fifa Data Set after Cleaning,Aggregation, etc is ",fifa_df.sh
```



```
The Final Size of the Fifa Data Set after Cleaning,Aggregation, etc is (18944,
```

Exporting The Filtered Data Set

```
fifa_df.to_csv('processed_fifa_data',index=False)
```

Plotting With Respect to gk_avg

```
x_axis=list(goalkeepers_df['gk_average'])  
y_axis=list(goalkeepers_df['skill_avg'])  
mp.title('Overall Skill v/s Goalkeepers Specific Skills')  
mp.scatter(x_axis,y_axis)  
mp.grid(True)
```



Overall Skill v/s Goalkeepers Specific Skills