In [1]:

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

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [2]:

```
fifa=pd.read_csv(r"D:\Full Stack Data Science\18 Aug\18th_resume project\S
fifa
```

Out[2]:

	Unnamed: 0	ID	Name	Age	Photo	Nationali
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argenti
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portug
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Bra
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spa
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgiu
...	...	...	...	...	...	
18202	18202	238813	J. Lundstram	19	https://cdn.sofifa.org/players/4/19/238813.png	Engla
18203	18203	243165	N. Christoffersson	19	https://cdn.sofifa.org/players/4/19/243165.png	Swed
18204	18204	241638	B. Worman	16	https://cdn.sofifa.org/players/4/19/241638.png	Engla
18205	18205	246268	D. Walker-Rice	17	https://cdn.sofifa.org/players/4/19/246268.png	Engla
18206	18206	246269	G. Nugent	16	https://cdn.sofifa.org/players/4/19/246269.png	Engla

18207 rows × 89 columns

In [3]:

```
fifa.shape
```

Out[3]:

```
(18207, 89)
```

In [4]:
```
1 fifa.columns
```

Out[4]:
```
Index(['Unnamed: 0', 'ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
       'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Specia
l',
       'Preferred Foot', 'International Reputation', 'Weak Foot',
       'Skill Moves', 'Work Rate', 'Body Type', 'Real Face', 'Position',
       'Jersey Number', 'Joined', 'Loaned From', 'Contract Valid Until',
       'Height', 'Weight', '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', 'Crossing',
       'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribblin
g',
       'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
       'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
       'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
       'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
       'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandlin
g',
       'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause'],
      dtype='object')
```

In [5]:
```
1 fifa.describe()
```

Out[5]:

	Unnamed: 0	ID	Age	Overall	Potential	Special	I
count	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18
mean	9103.000000	214298.338606	25.122206	66.238699	71.307299	1597.809908	
std	5256.052511	29965.244204	4.669943	6.908930	6.136496	272.586016	
min	0.000000	16.000000	16.000000	46.000000	48.000000	731.000000	
25%	4551.500000	200315.500000	21.000000	62.000000	67.000000	1457.000000	
50%	9103.000000	221759.000000	25.000000	66.000000	71.000000	1635.000000	
75%	13654.500000	236529.500000	28.000000	71.000000	75.000000	1787.000000	
max	18206.000000	246620.000000	45.000000	94.000000	95.000000	2346.000000	

8 rows × 44 columns

In [6]:
```
1 fifa.isnull().sum()
```

Out[6]:
```
Unnamed: 0           0
ID                   0
Name                 0
Age                  0
Photo                0
                   ...
GKHandling          48
GKKicking           48
GKPositioning       48
GKReflexes          48
Release Clause    1564
Length: 89, dtype: int64
```

In [7]:
```
1 fifa.info()
```

```
61  FKAccuracy                18159 non-null  float64
62  LongPassing               18159 non-null  float64
63  BallControl               18159 non-null  float64
64  Acceleration              18159 non-null  float64
65  SprintSpeed               18159 non-null  float64
66  Agility                   18159 non-null  float64
67  Reactions                 18159 non-null  float64
68  Balance                   18159 non-null  float64
69  ShotPower                 18159 non-null  float64
70  Jumping                   18159 non-null  float64
71  Stamina                   18159 non-null  float64
72  Strength                  18159 non-null  float64
73  LongShots                 18159 non-null  float64
74  Aggression                18159 non-null  float64
75  Interceptions             18159 non-null  float64
76  Positioning               18159 non-null  float64
77  Vision                    18159 non-null  float64
78  Penalties                 18159 non-null  float64
79  Composure                 18159 non-null  float64
80  Marking                   18159 non-null  float64
```

In [8]:
```
1 fifa['Body Type'].value_counts()
```

Out[8]:
```
Body Type
Normal                10595
Lean                   6417
Stocky                 1140
Messi                     1
C. Ronaldo                1
Neymar                    1
Courtois                  1
PLAYER_BODY_TYPE_25       1
Shaqiri                   1
Akinfenwa                 1
Name: count, dtype: int64
```

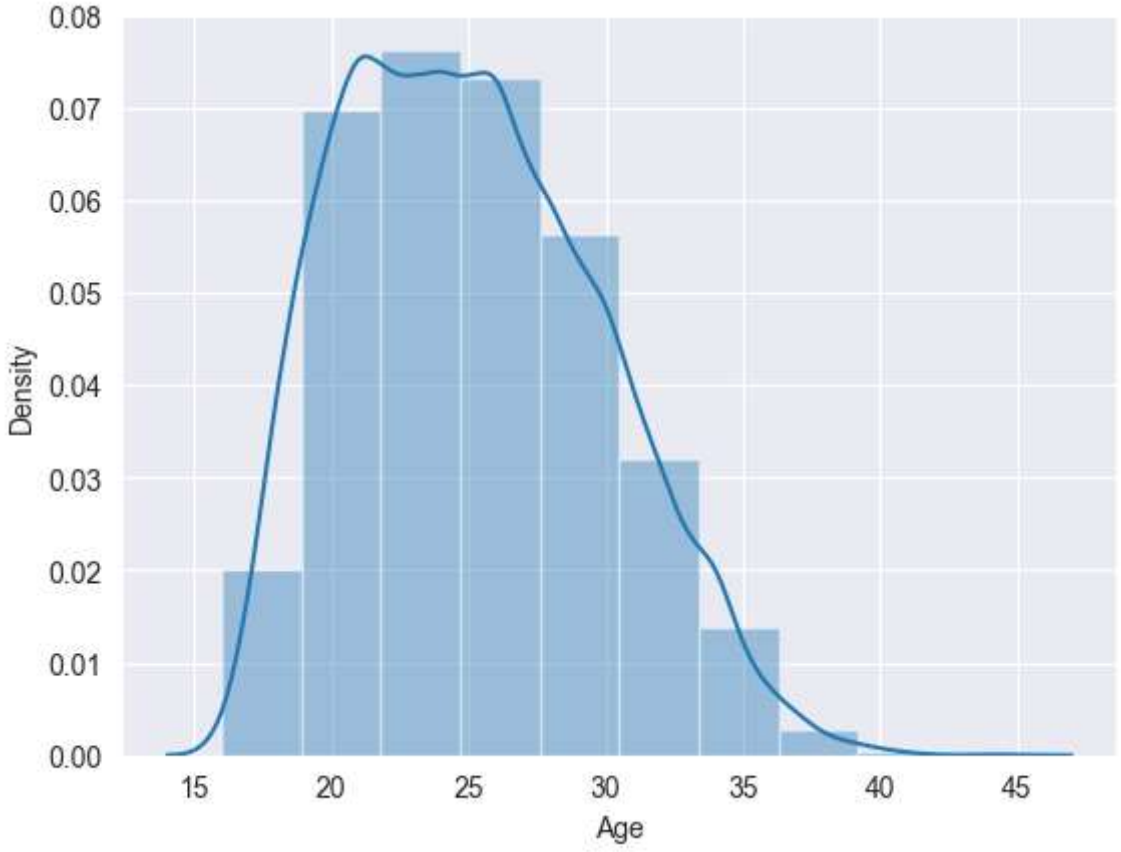
Interpretation

- This dataset contains 89 variables.
- Out of the 89 variables, 44 are numerical variables. 38 are of float64 data type and remaining 6 are of int64 data type.
- The remaining 45 variables are of character data type.

Explore Age variable

In [9]:
```
1 sns.set_style('darkgrid')
2 x=fifa.Age
3 sns.distplot(x,bins=10)
```

Out[9]: <Axes: xlabel='Age', ylabel='Density'>



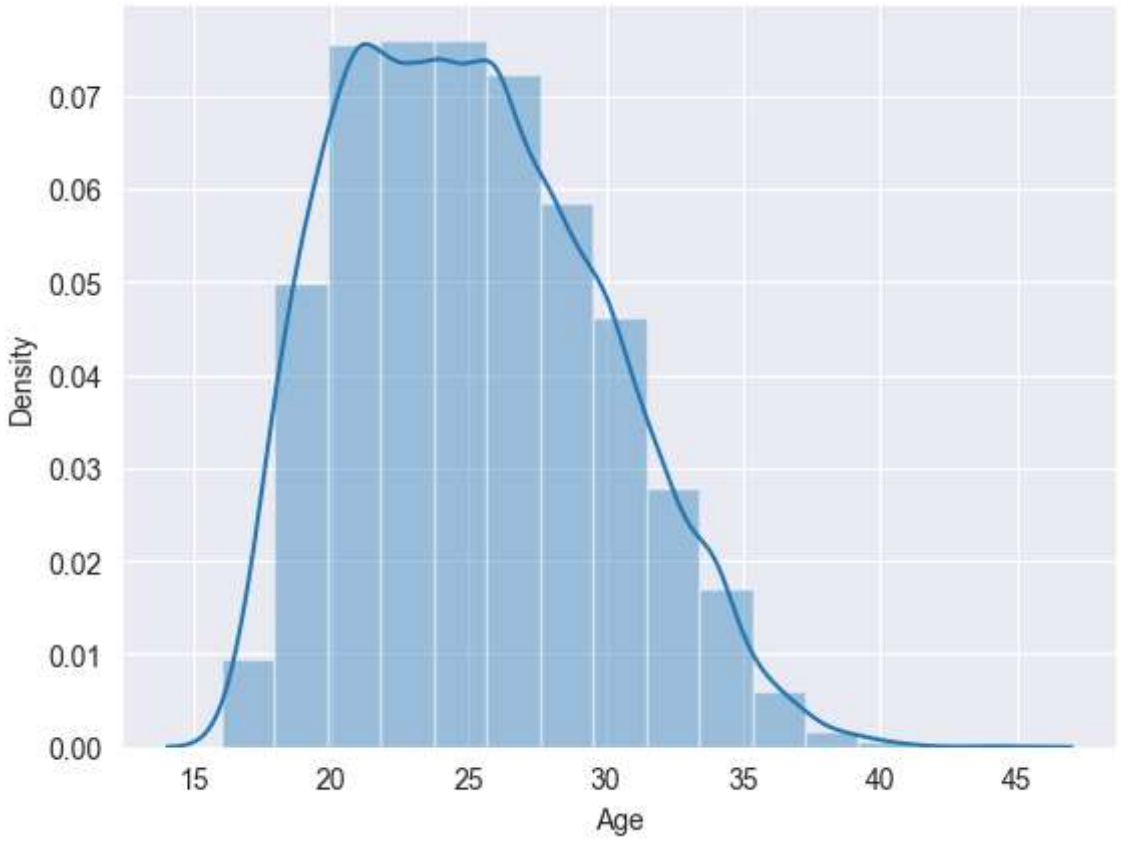
Interpretation

It can be seen that the age variable is slightly positively skewed.

we can use pandas series object to get an informative axis label as follows

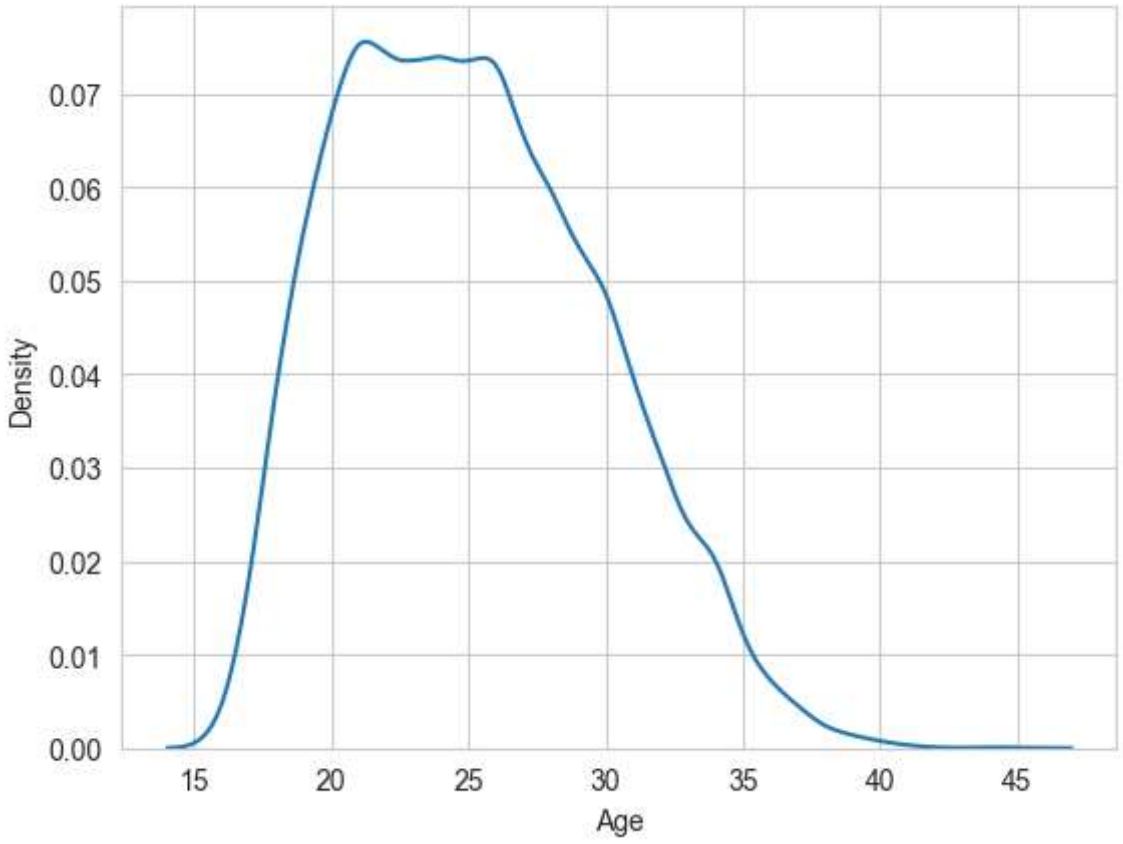
In [10]:

```
1 x=fifa.Age
2 pd.Series(x,name='Age Variable')
3 ax=sns.distplot(x,bins=15)
```



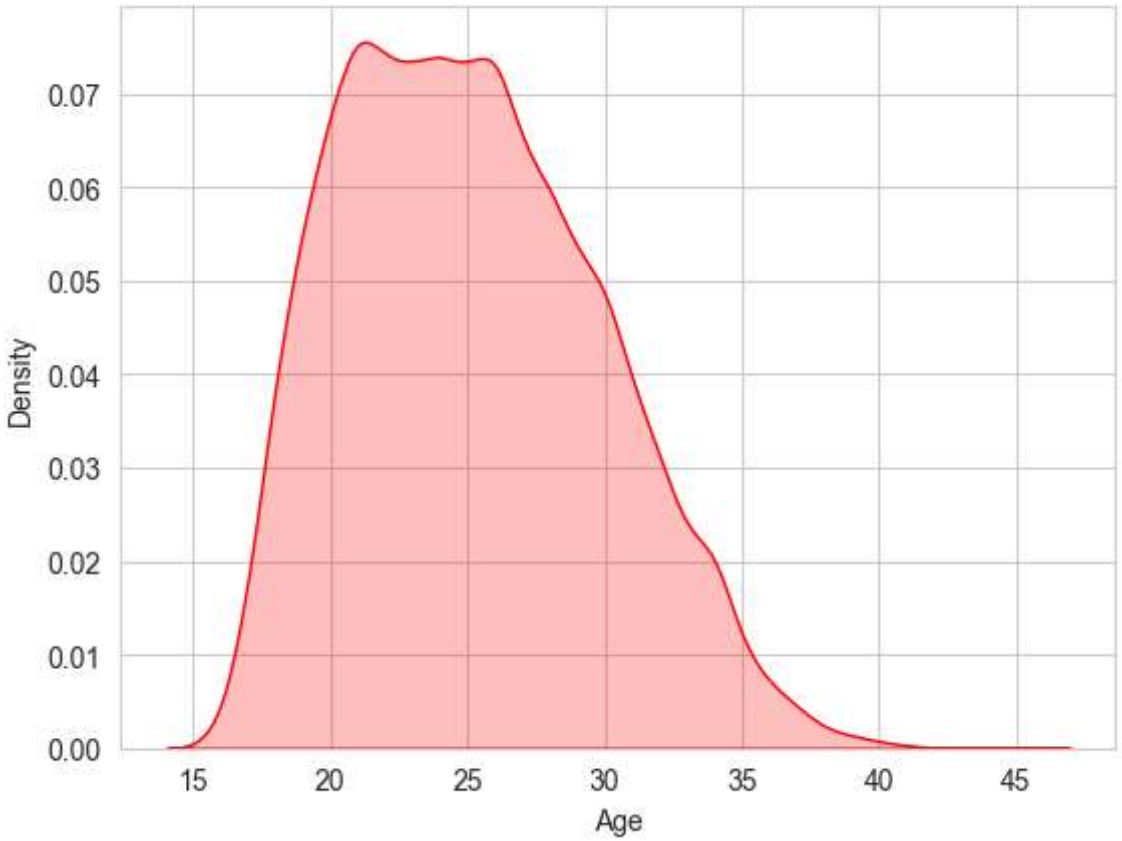
Seaborn Kernel Density Estimation(KDE) Plot

In [11]:
```
1 sns.set_style('whitegrid')
2 ax=sns.kdeplot(fifa.Age)
```



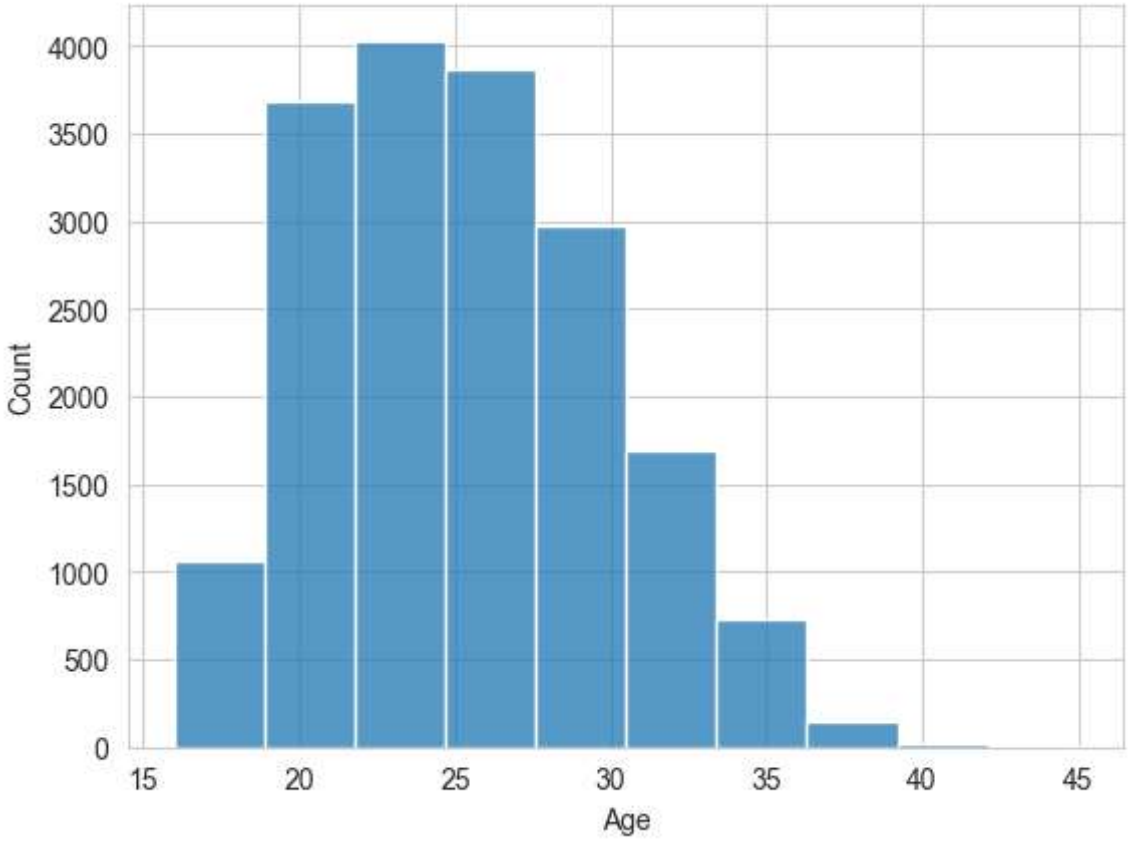
In [12]:

```
sns.set_style('whitegrid')
ax=sns.kdeplot(fifa.Age,color='r',shade=True,shade_lowest=True)
```



Histogram

In [13]:
```
1 ax=sns.histplot(x,bins=10)
```



Explore Prefered foot variable

In [14]:
```
1 fifa['Preferred Foot'].nunique()
```

Out[14]:
```
2
```

In [15]:
```
1 fifa['Preferred Foot'].unique()
```

Out[15]:
```
array(['Left', 'Right', nan], dtype=object)
```

There are two types of values in preferred foot

1. Left
2. Right

In [16]:
```
1 fifa['Preferred Foot'].value_counts()
```

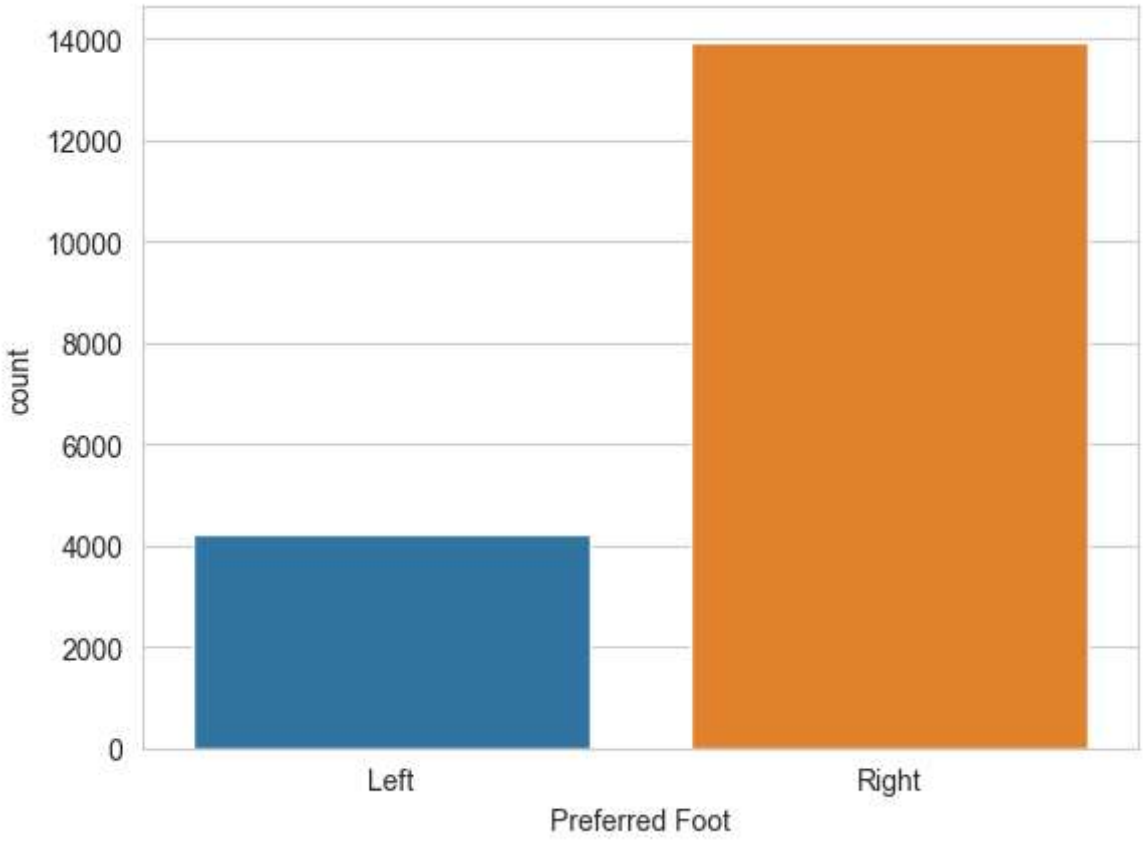
Out[16]:
```
Preferred Foot
Right    13948
Left      4211
Name: count, dtype: int64
```

- 1. There are 13948 times preferred Right foot
- 2. Ther are 4211 times preferred Left foot

Visualize distribution of values with countplot() .

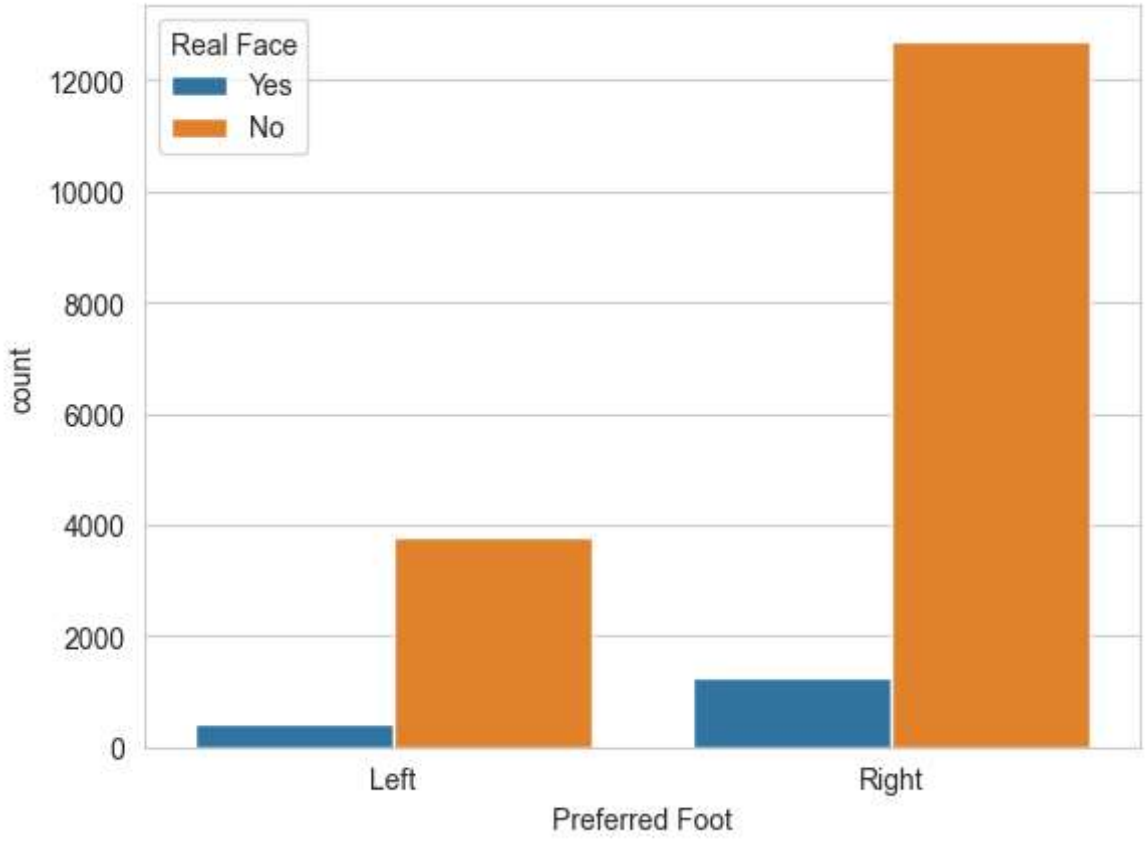
In [17]:
```
1 sns.set_style('whitegrid')
2 sns.countplot(data=fifa,x='Preferred Foot')
```

Out[17]:
```
<Axes: xlabel='Preferred Foot', ylabel='count'>
```



In [18]:
```
sns.set_style('whitegrid')
sns.countplot(data=fifa,x='Preferred Foot',hue='Real Face')
```

Out[18]: <Axes: xlabel='Preferred Foot', ylabel='count'>

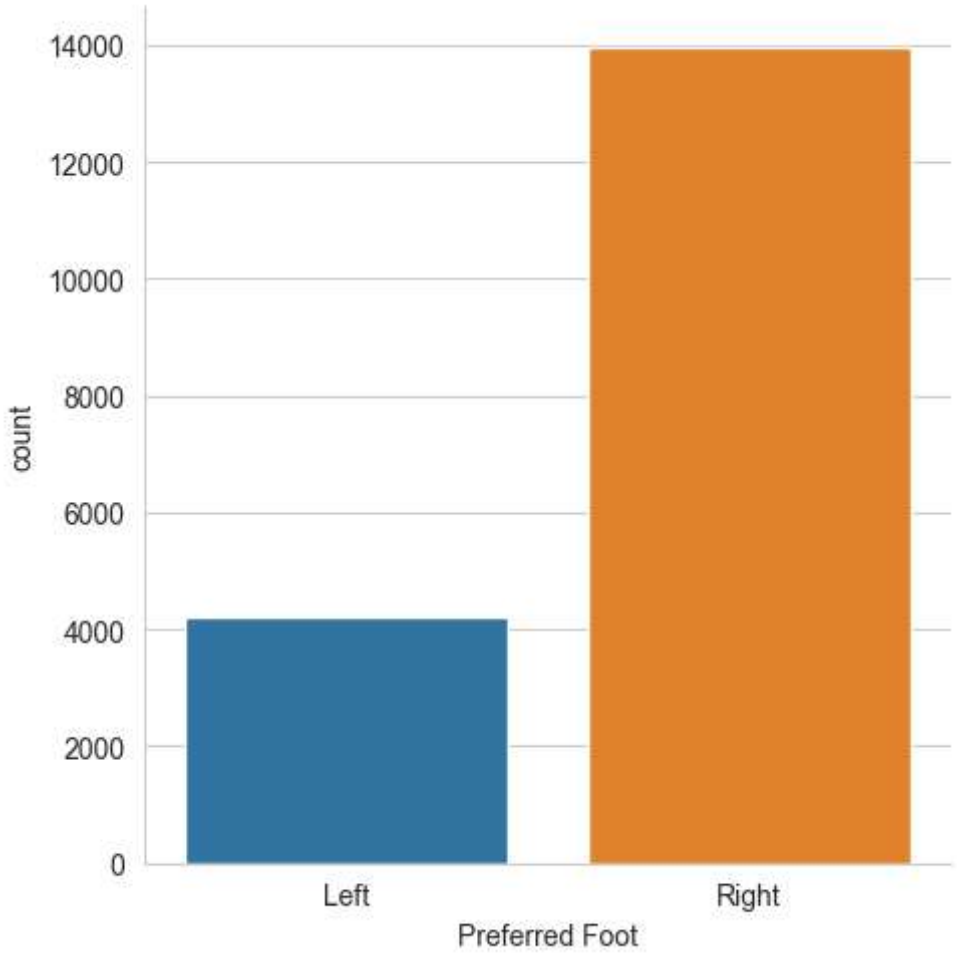


Catplot()

we use Catplot function to plot categorical scatterplots

In [19]:
```
1 sns.catplot(data=fifa ,x='Preferred Foot',kind='count')
```

Out[19]: <seaborn.axisgrid.FacetGrid at 0x1e7e1f136d0>



Explore Interntional Reputation variable

In [20]:
```
1 fifa['International Reputation'].nunique()
```

Out[20]: 5

In [21]:
```
1 fifa['International Reputation'].value_counts()
```

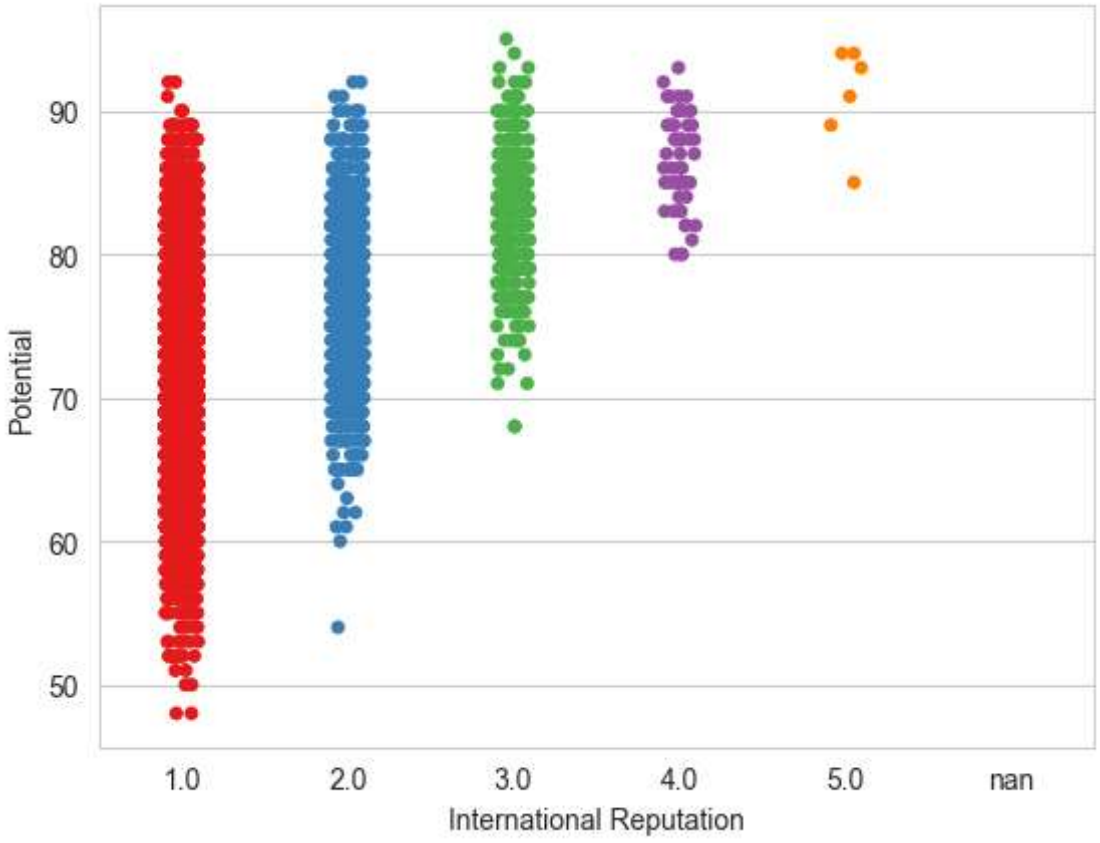
Out[21]:
```
International Reputation
1.0    16532
2.0     1261
3.0      309
4.0       51
5.0        6
Name: count, dtype: int64
```

Stripplot()

- This function draws a scatterplot where one variable is categorical.
- We will plot a stripplot with 'International Reputation' as Categorical variable and 'Potential' as other variable.

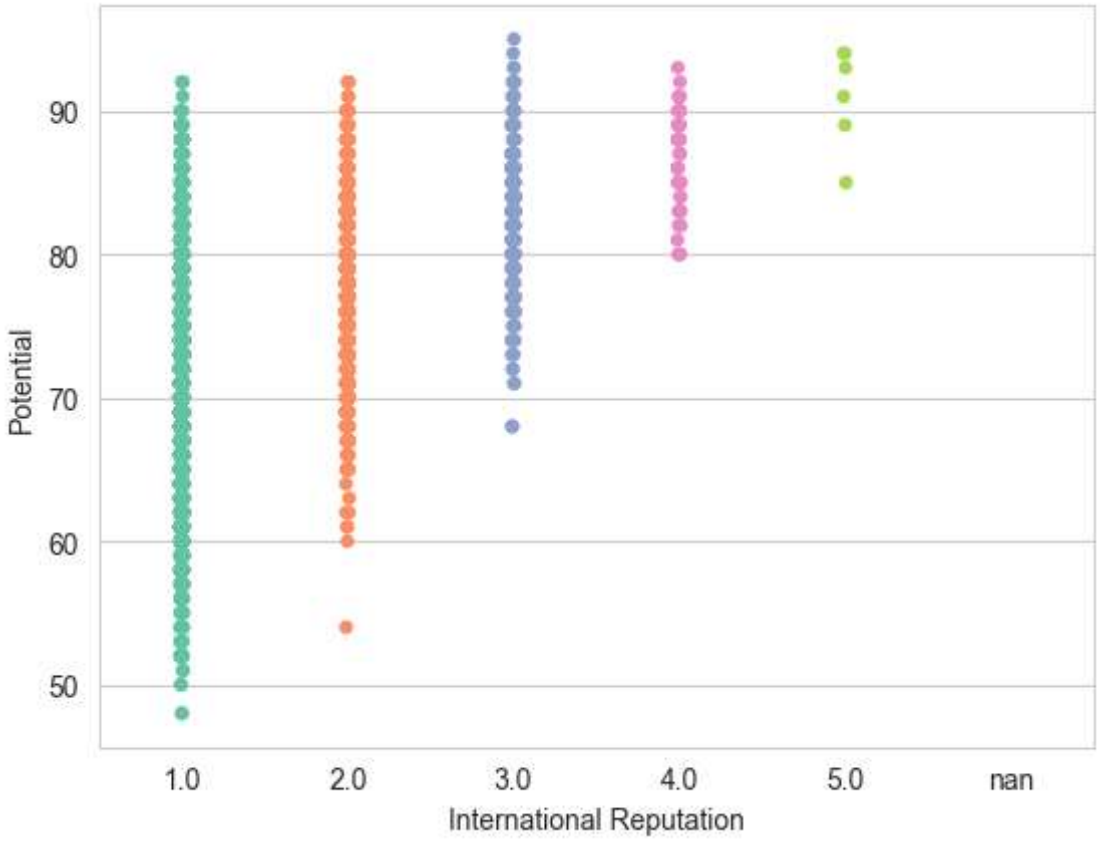
In [22]:
```
1 sns.stripplot(data=fifa,x='International Reputation',y='Potential',palett
```

Out[22]: <Axes: xlabel='International Reputation', ylabel='Potential'>



In [23]:
```
1 sns.stripplot(data=fifa,x='International Reputation',y='Potential',jitter
```

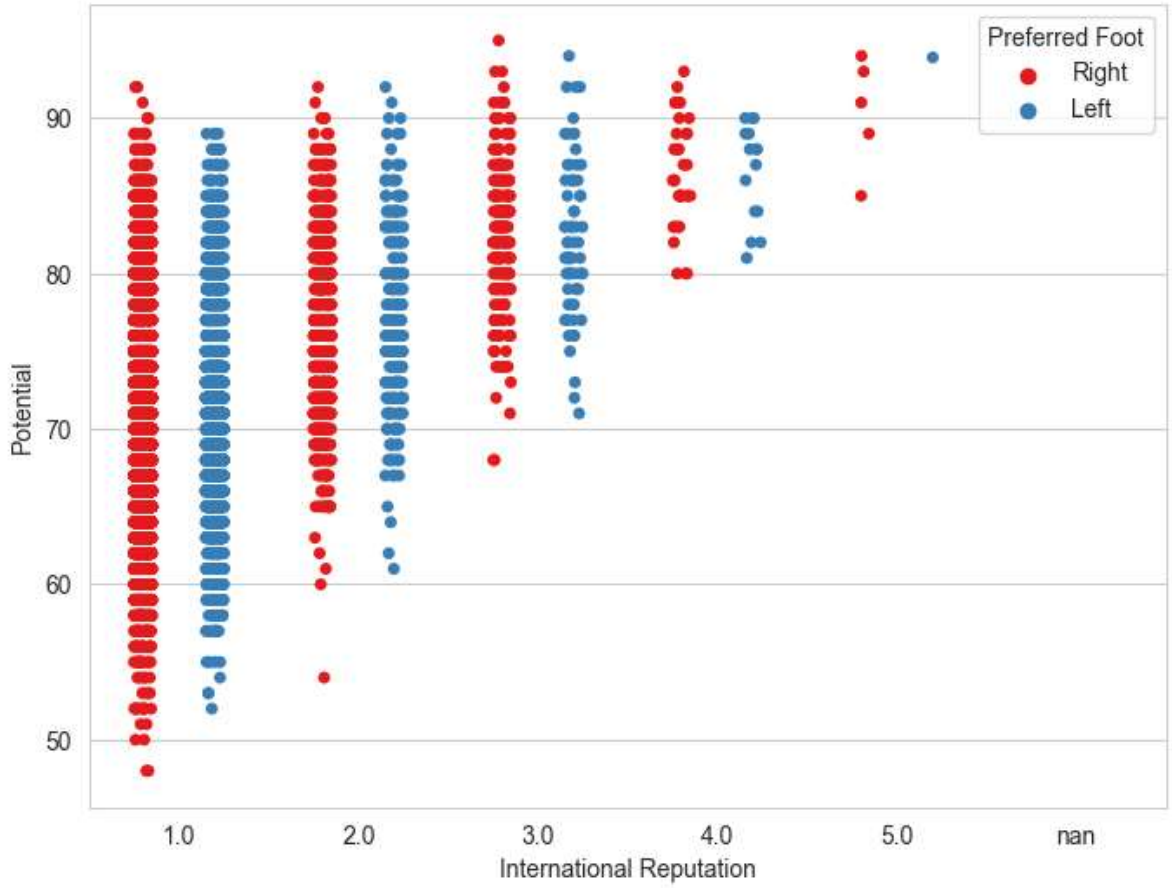
Out[23]: <Axes: xlabel='International Reputation', ylabel='Potential'>



we can nest the strips within a second categorical variable Preferred Foot as follows

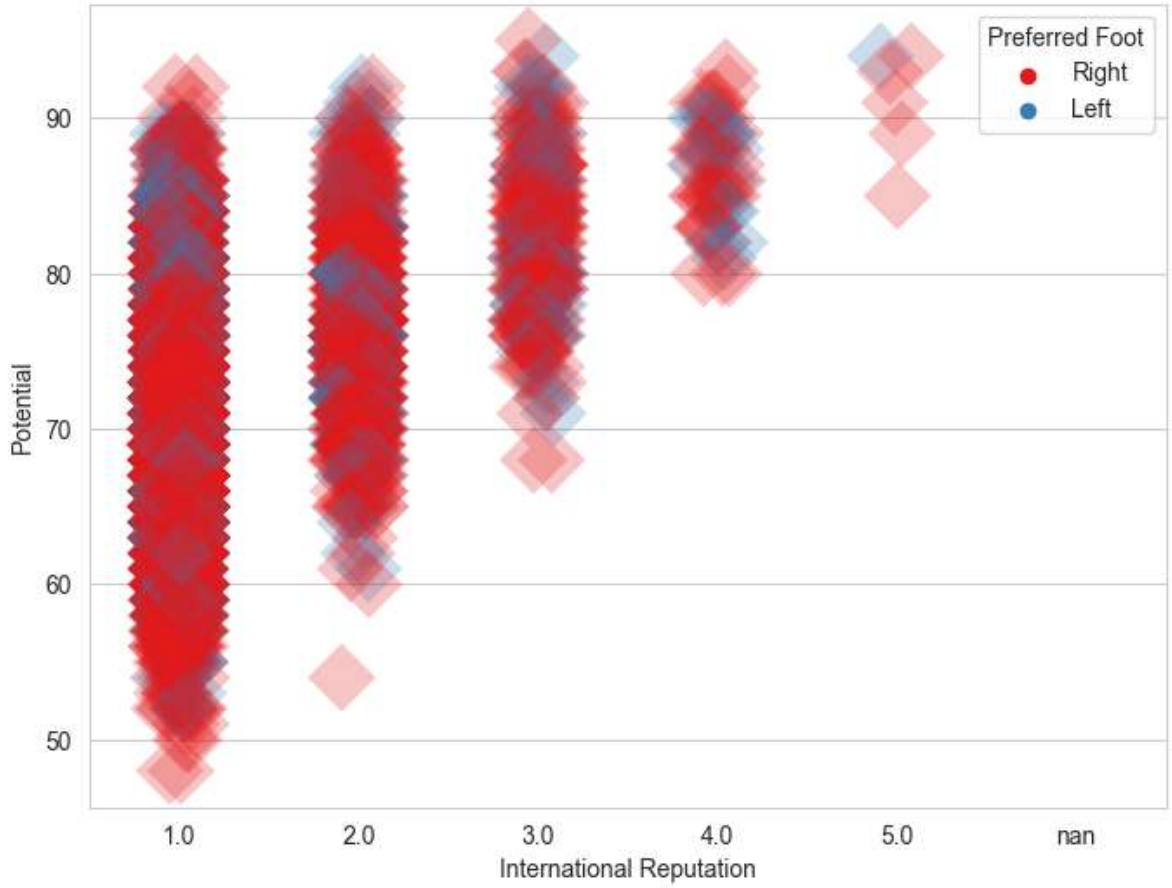
In [24]:
```
1 f, ax=plt.subplots(figsize=(8,6))
2 sns.stripplot(data=fifa,x='International Reputation',y='Potential',hue='Pr
```

Out[24]: <Axes: xlabel='International Reputation', ylabel='Potential'>



In [25]:
```
1 f, ax=plt.subplots(figsize=(8,6))
2 sns.stripplot(data=fifa,x='International Reputation',y='Potential',hue='Pr
```

Out[25]: <Axes: xlabel='International Reputation', ylabel='Potential'>

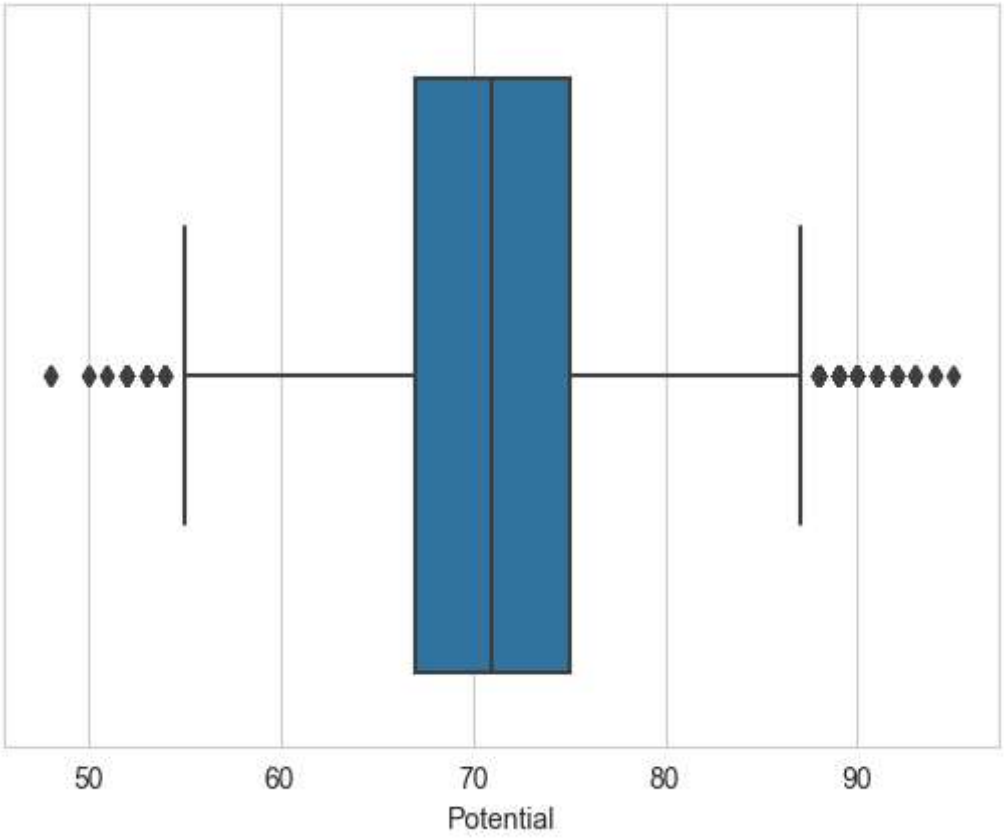


boxplot()

- This function draws a box plot to show distribution with respect to categories.

In [26]:
```
1 sns.boxplot(x=fifa.Potential)
```

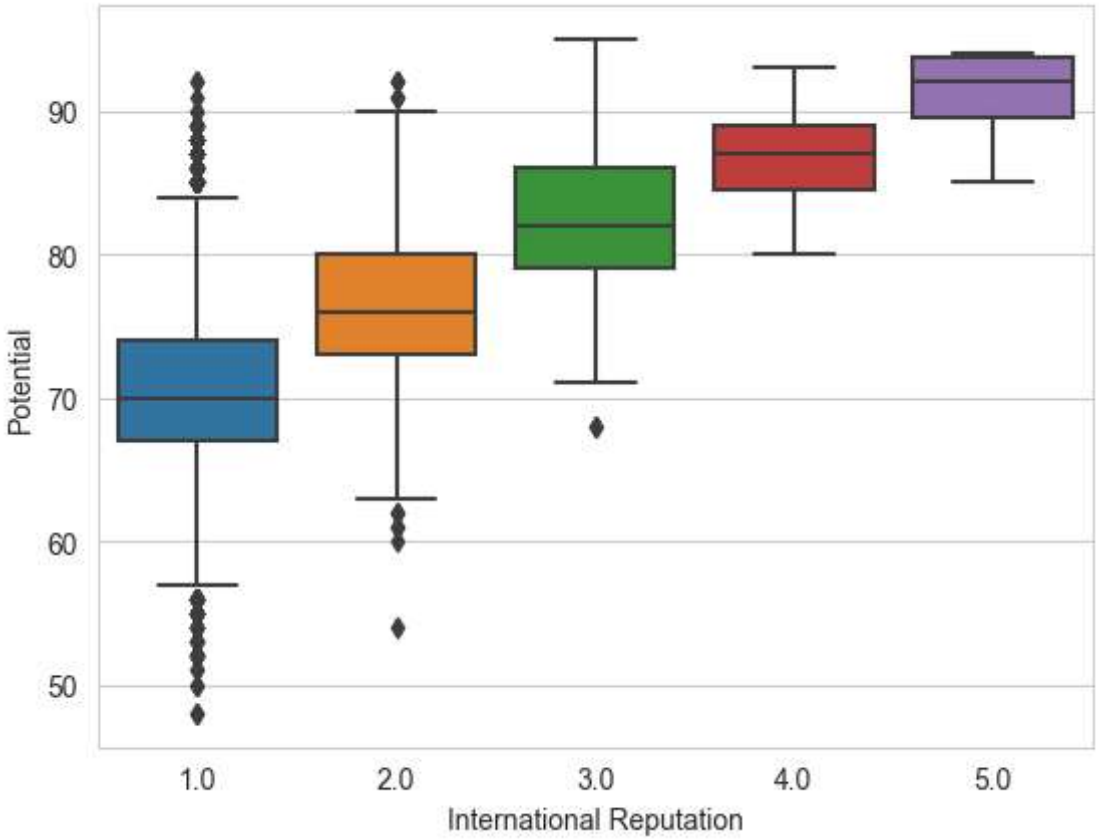
Out[26]: <Axes: xlabel='Potential'>



In [27]:
```
1 ### Boxplot International reputation vs Potential
```

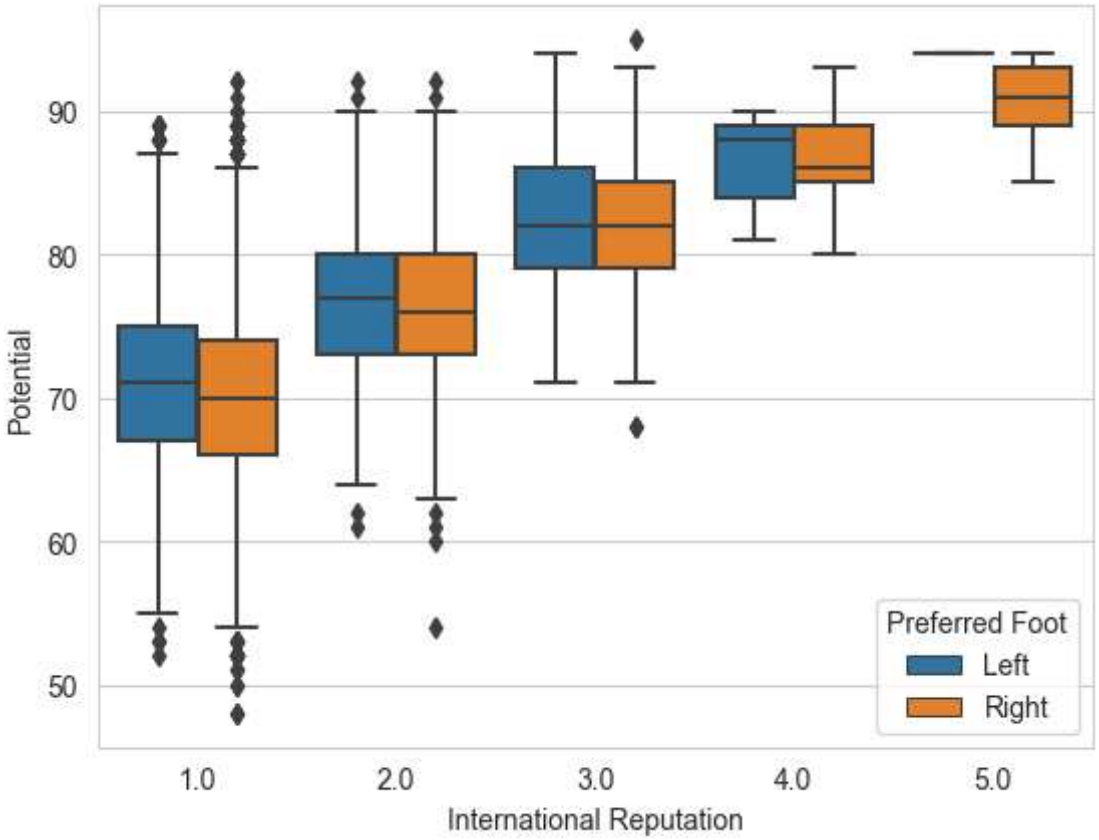
In [28]:
```
1 sns.boxplot(data=fifa,x='International Reputation',y='Potential')
```

Out[28]: <Axes: xlabel='International Reputation', ylabel='Potential'>



In [29]:
```
1 sns.boxplot(data=fifa,x='International Reputation',y='Potential',hue='Pre
```

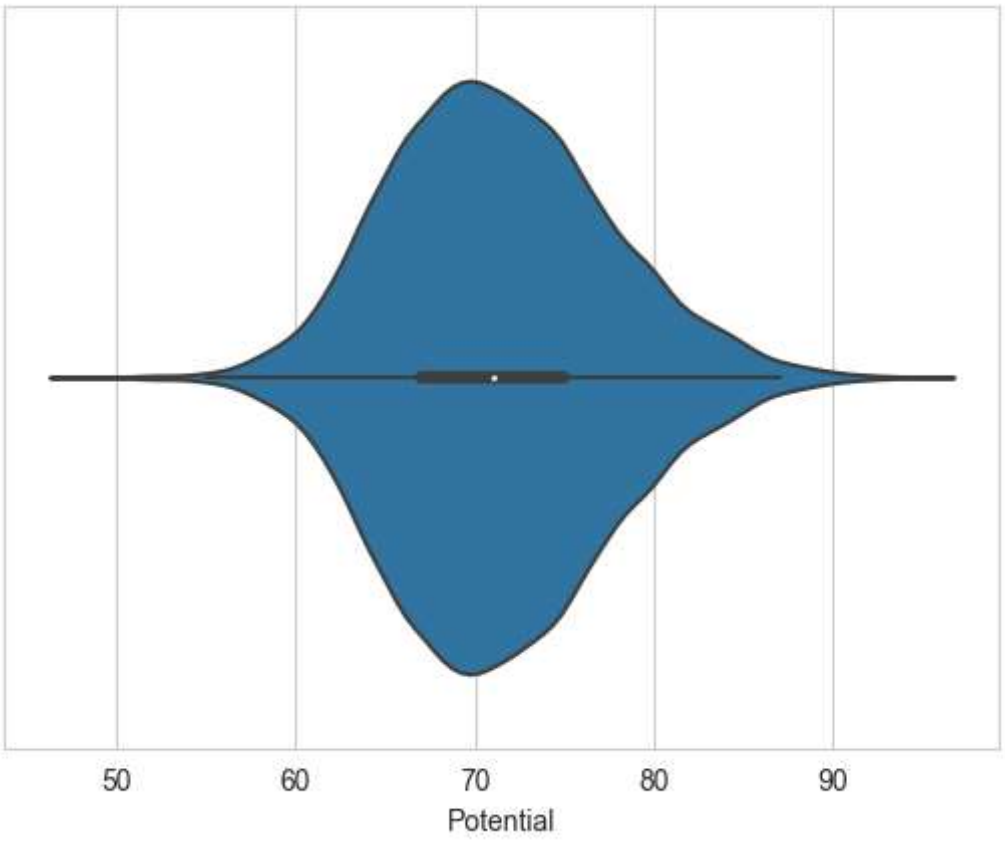
Out[29]: <Axes: xlabel='International Reputation', ylabel='Potential'>



Violinplot()

In [30]:
```
1 sns.violinplot(x=fifa.Potential)
```

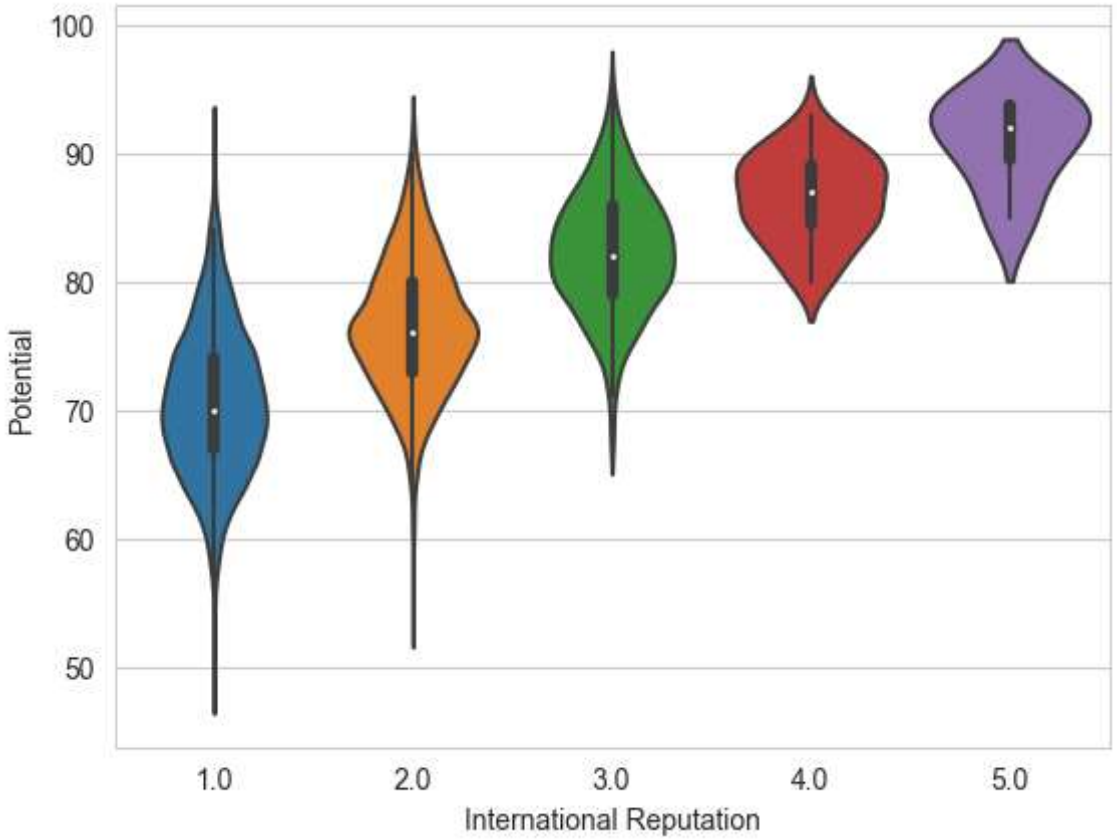
Out[30]: <Axes: xlabel='Potential'>



We can draw the vertical violinplot grouped by the categorical variable International Reputation as follows

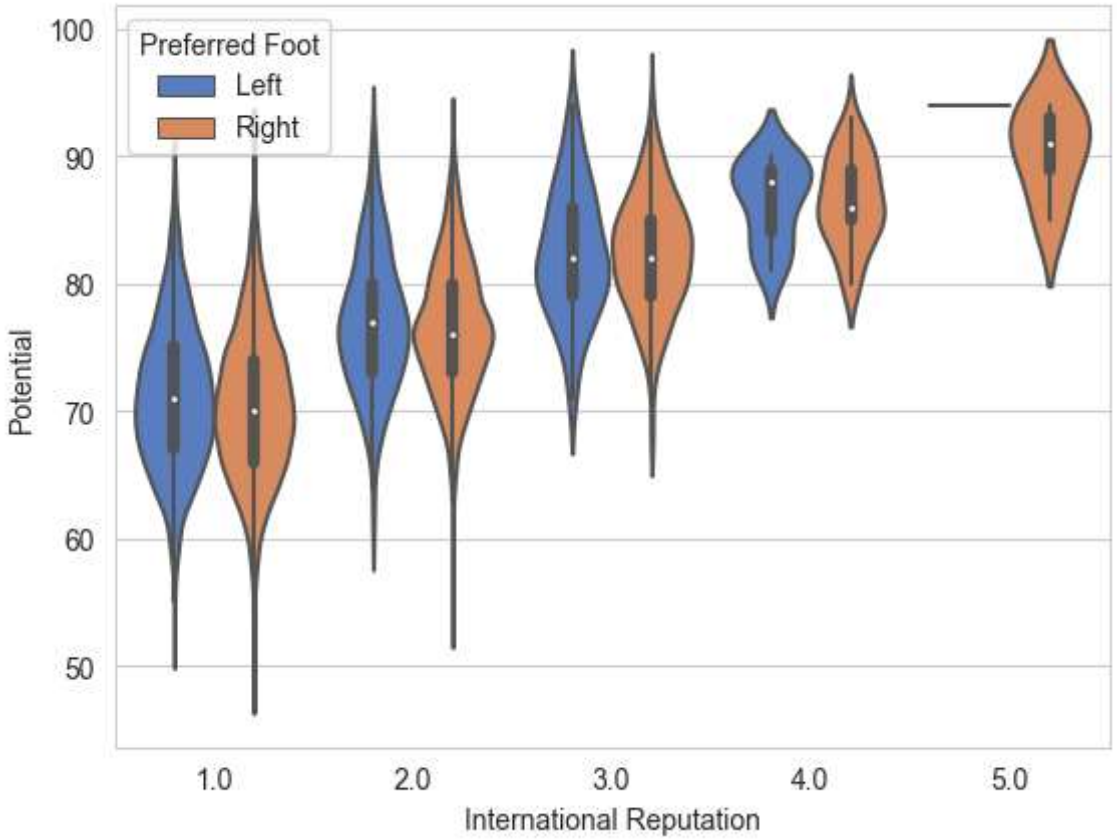
In [31]:
```
1 sns.violinplot(data=fifa,x='International Reputation',y='Potential')
```

Out[31]: <Axes: xlabel='International Reputation', ylabel='Potential'>



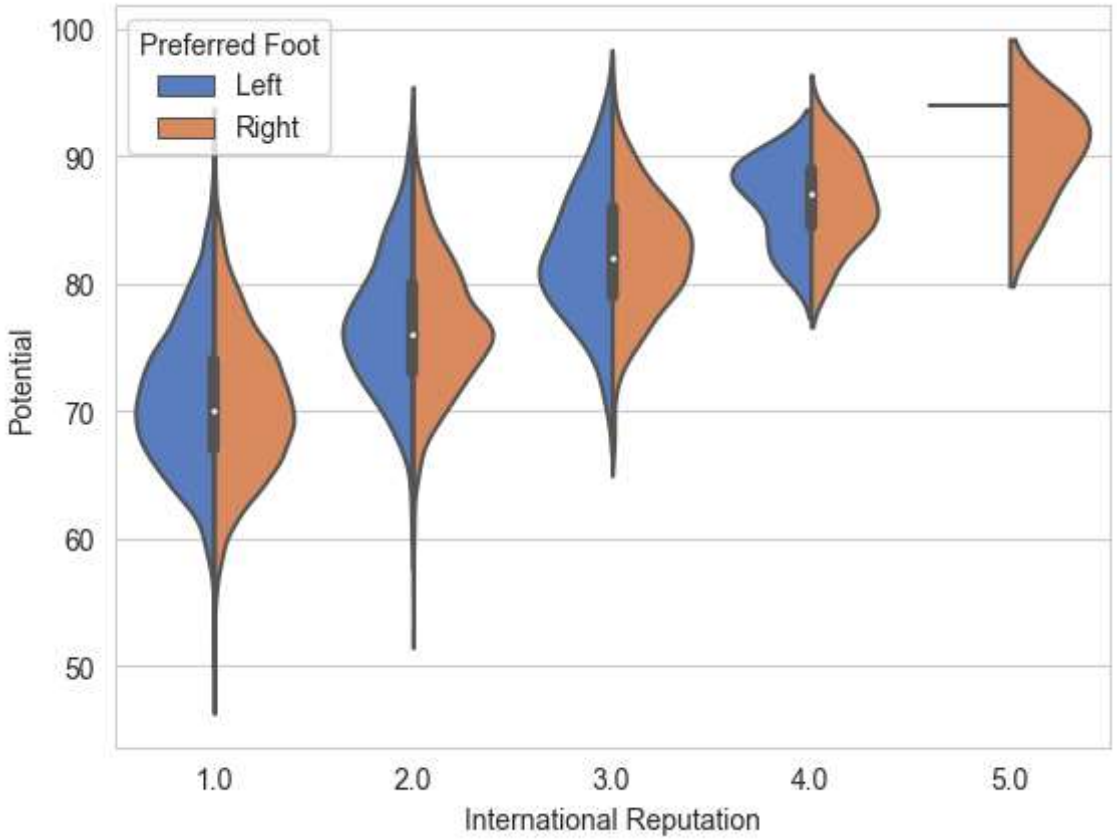
In [32]:
```
1 sns.violinplot(data=fifa,x='International Reputation',y='Potential',hue='
```

Out[32]: <Axes: xlabel='International Reputation', ylabel='Potential'>



In [33]:
```
1 sns.violinplot(data=fifa,x='International Reputation',y='Potential',hue='
```

Out[33]: <Axes: xlabel='International Reputation', ylabel='Potential'>

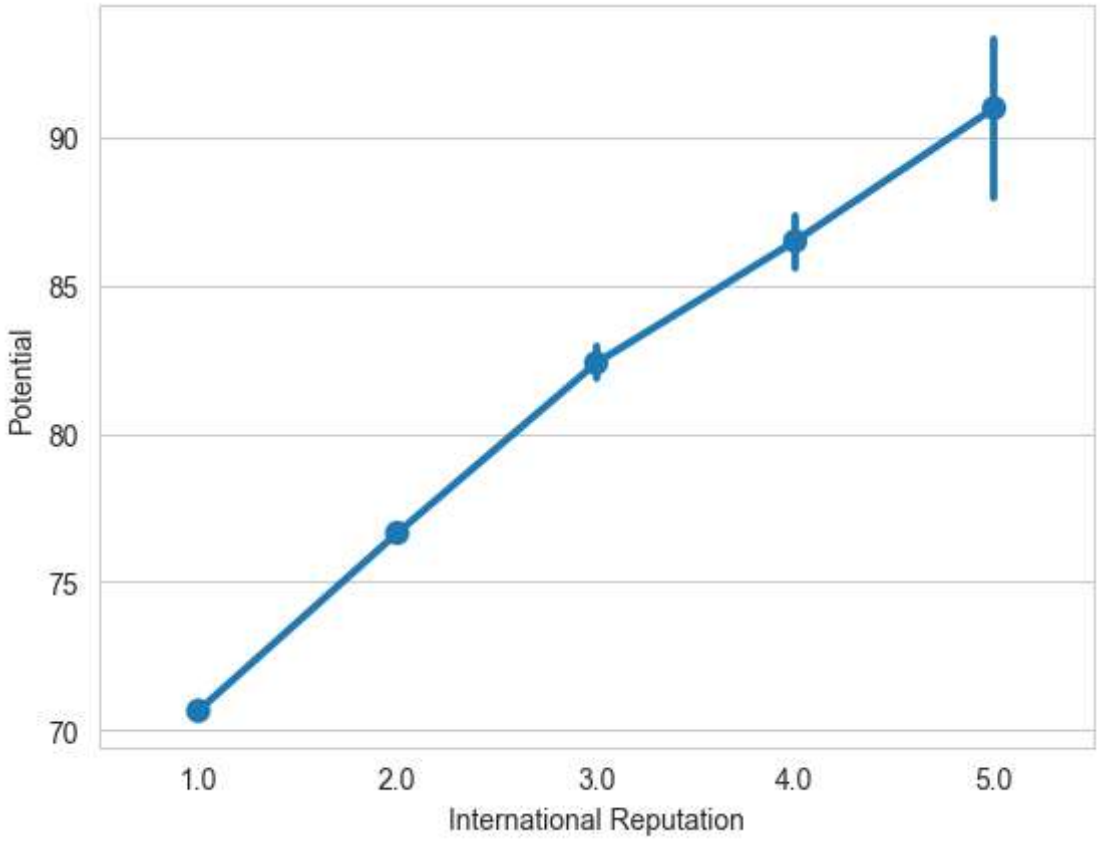


pointplot()

This function show point estimates and confidence intervals using scatter plot

In [34]:
```
1 sns.pointplot(data=fifa,x='International Reputation',y='Potential')
```

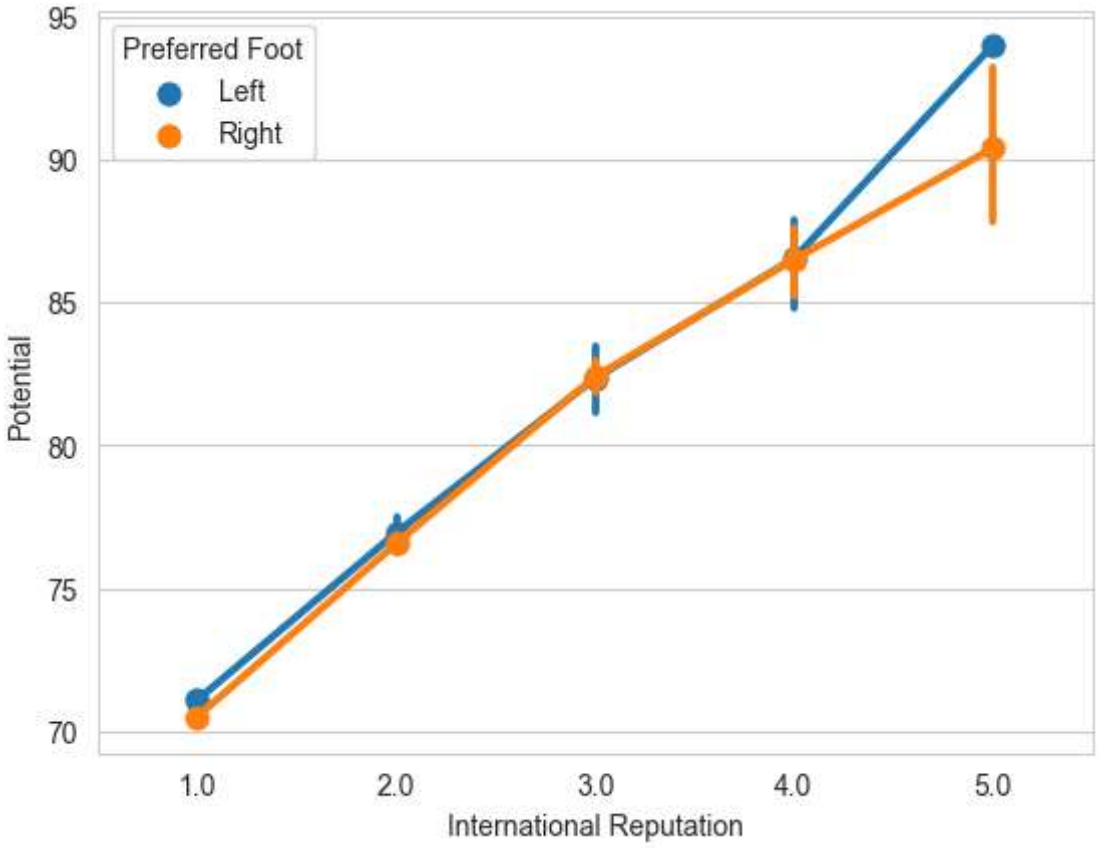
Out[34]: <Axes: xlabel='International Reputation', ylabel='Potential'>



We can draw a set of vertical points with nested grouping by a two variables as follows

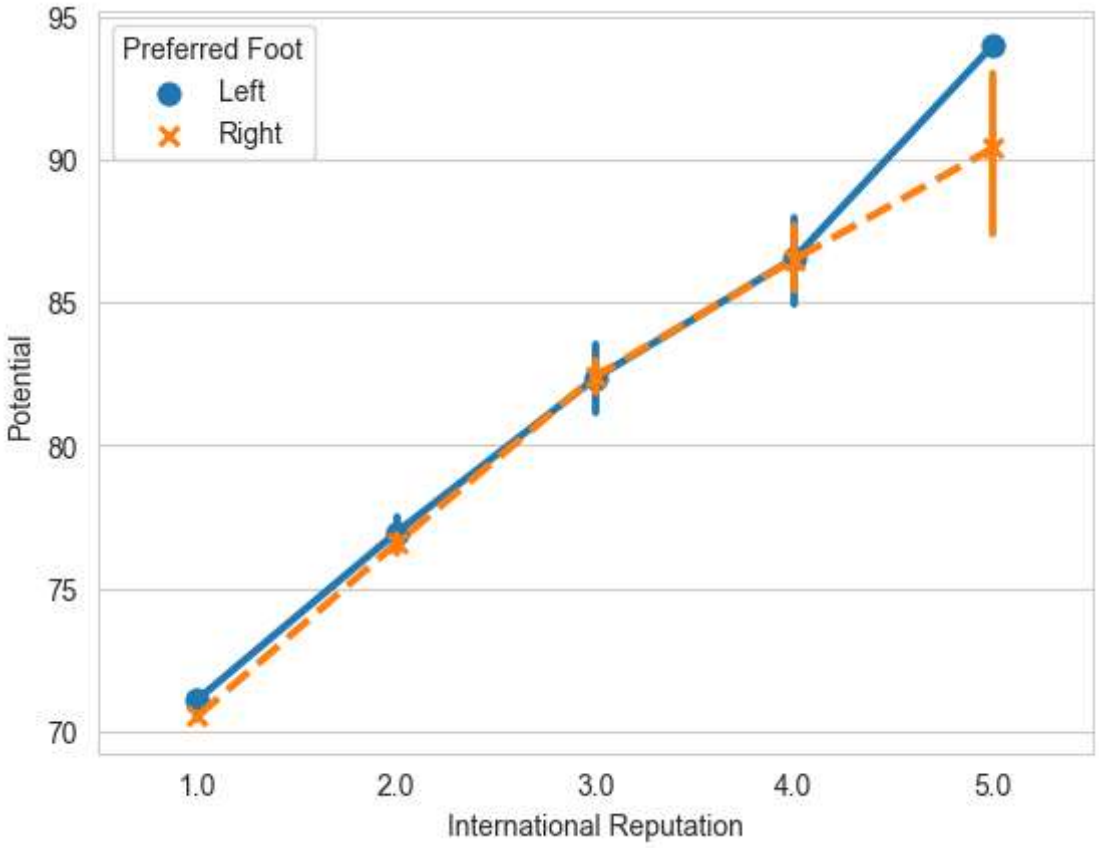
In [35]:
```
1 sns.pointplot(data=fifa,x='International Reputation',y='Potential',hue='P
```

Out[35]: <Axes: xlabel='International Reputation', ylabel='Potential'>



In [36]:
```
1 sns.pointplot(data=fifa,x='International Reputation',y='Potential',hue='P
```

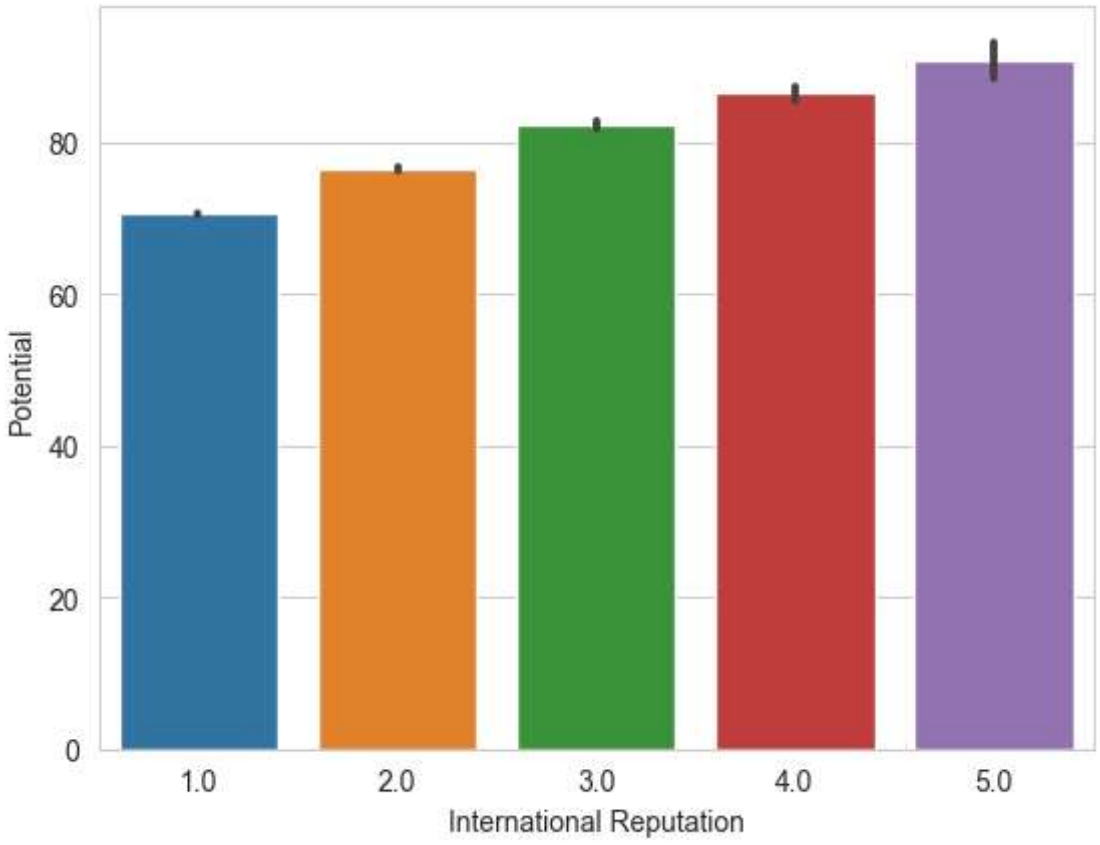
Out[36]: <Axes: xlabel='International Reputation', ylabel='Potential'>



barplot()

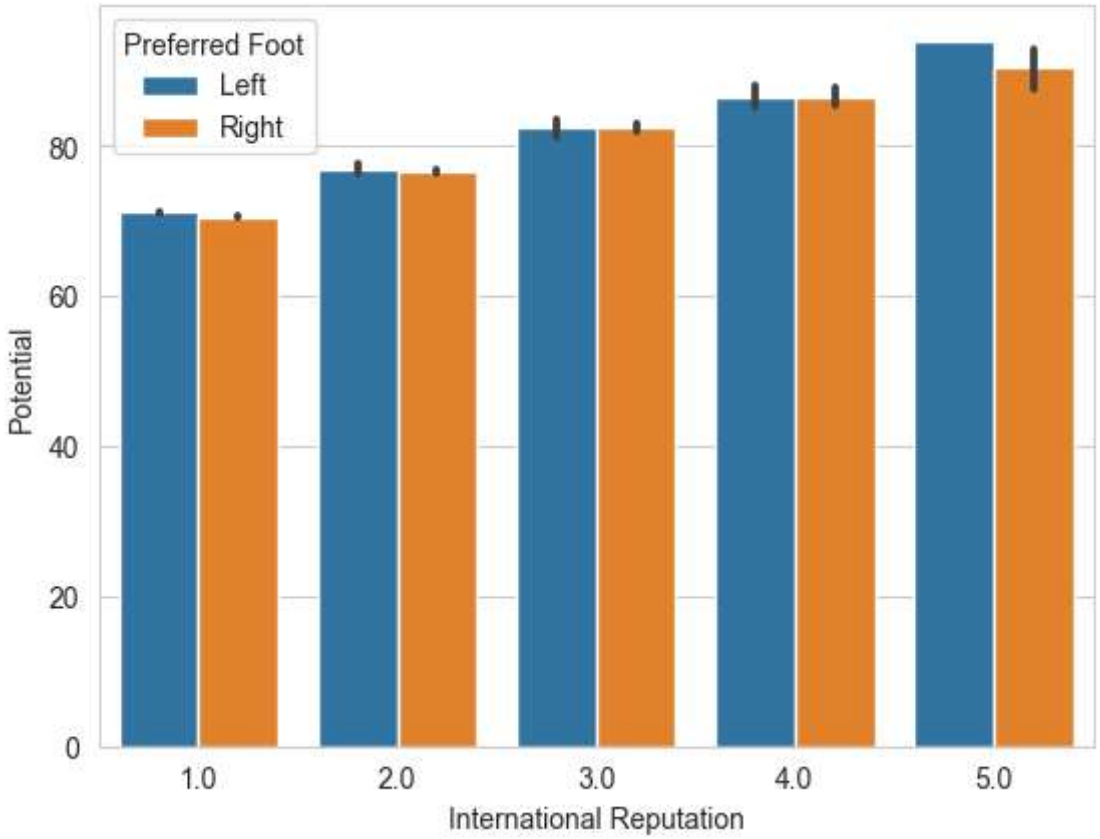
In [37]:
```
1 sns.barplot(data=fifa,x='International Reputation',y='Potential' )
```

Out[37]: <Axes: xlabel='International Reputation', ylabel='Potential'>



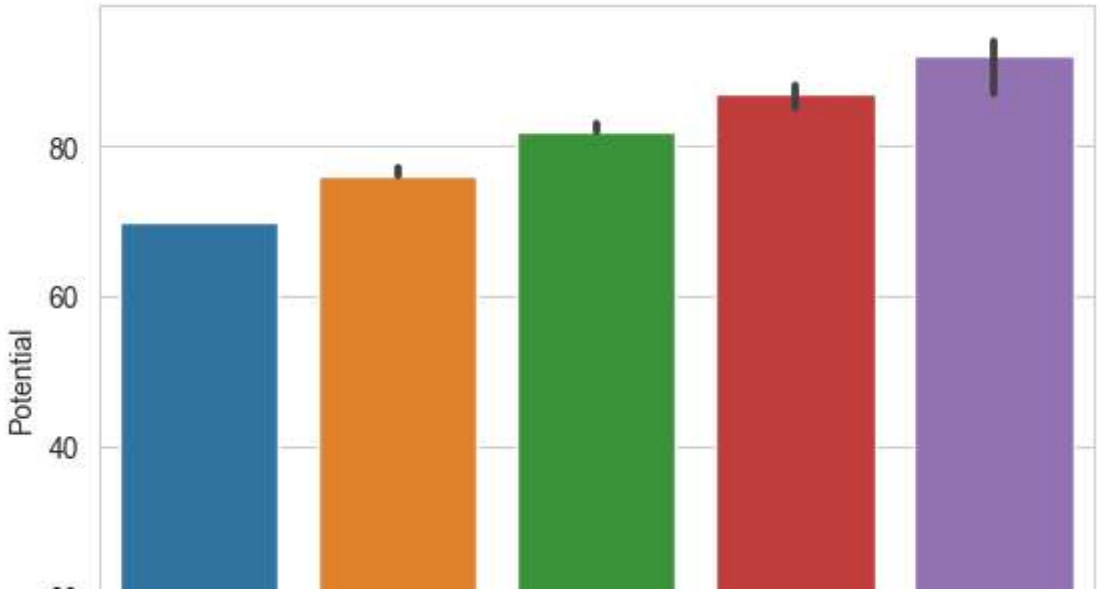
In [38]:
```
1 sns.barplot(data=fifa,x='International Reputation',y='Potential' ,hue='Pr
```

Out[38]: <Axes: xlabel='International Reputation', ylabel='Potential'>



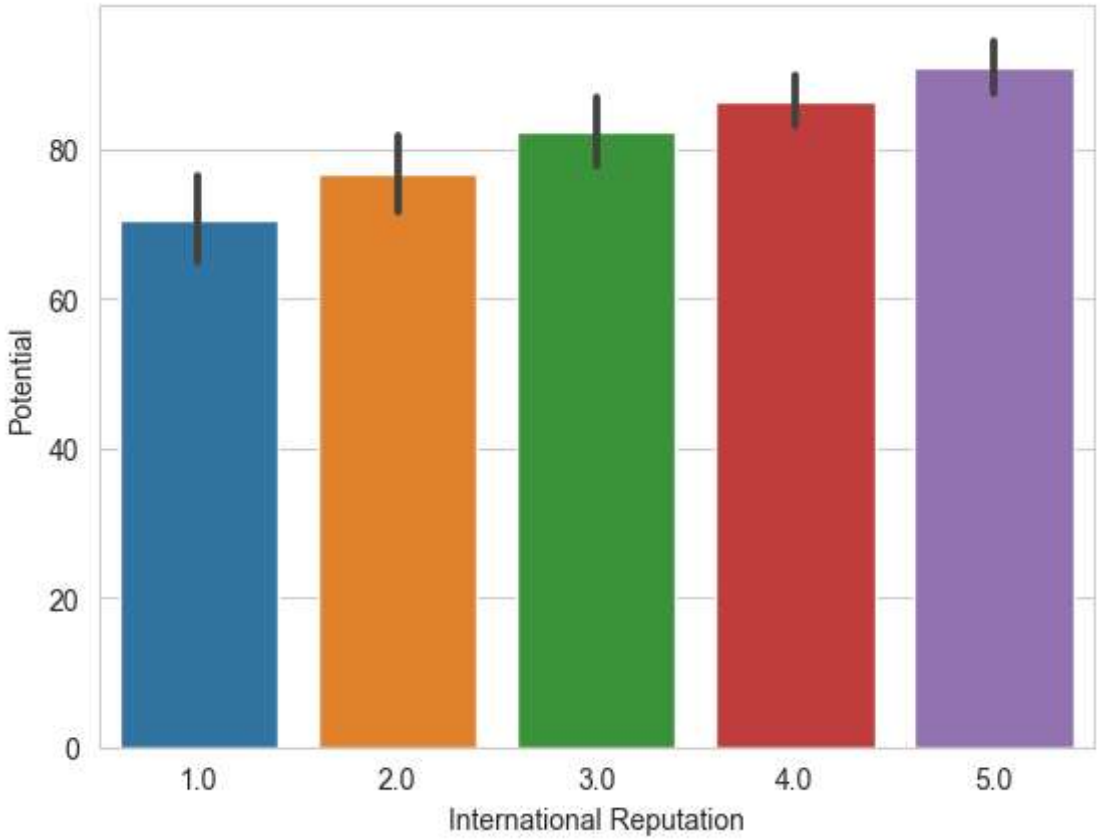
In [39]:
```
1 from numpy import median
2 sns.barplot(data=fifa,x='International Reputation',y='Potential',estimato
```

Out[39]: <Axes: xlabel='International Reputation', ylabel='Potential'>



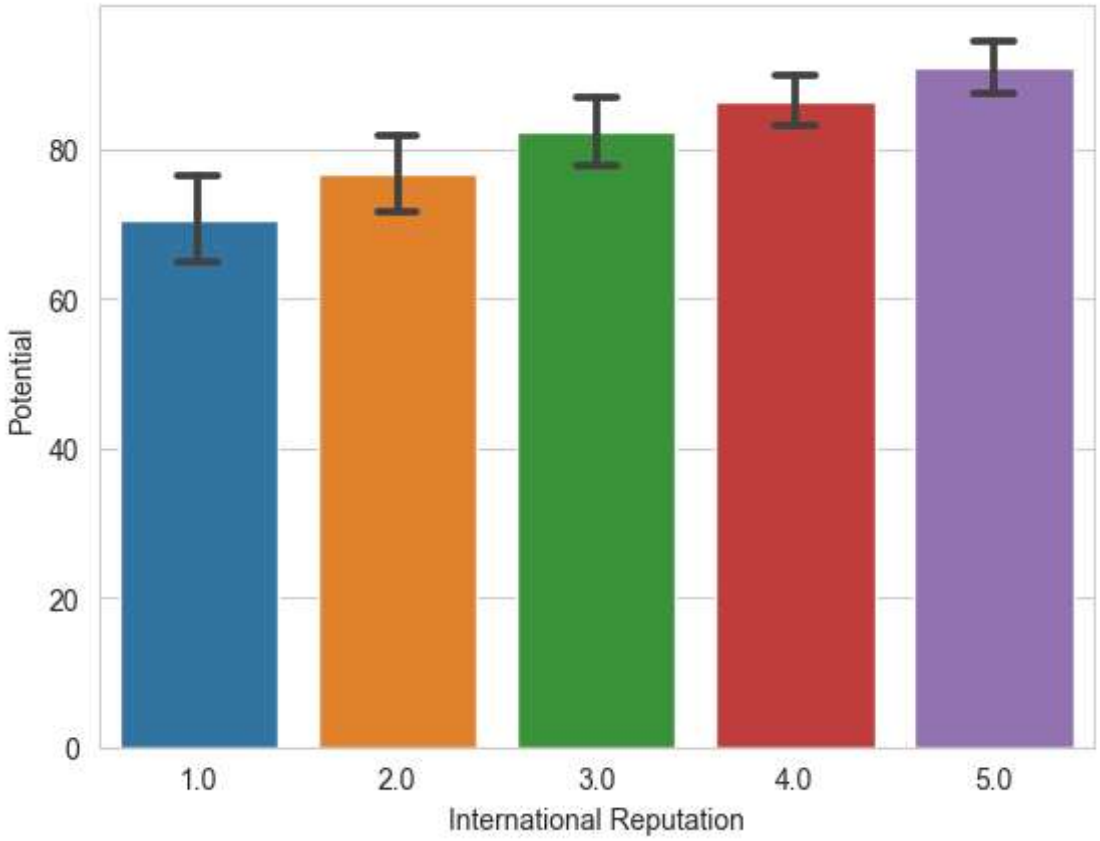
In [40]:
```
1 sns.barplot(data=fifa,x='International Reputation',y='Potential',ci='sd')
```

Out[40]: <Axes: xlabel='International Reputation', ylabel='Potential'>



In [41]:
```
1 sns.barplot(data=fifa,x='International Reputation',y='Potential',ci='sd',
```

Out[41]: <Axes: xlabel='International Reputation', ylabel='Potential'>

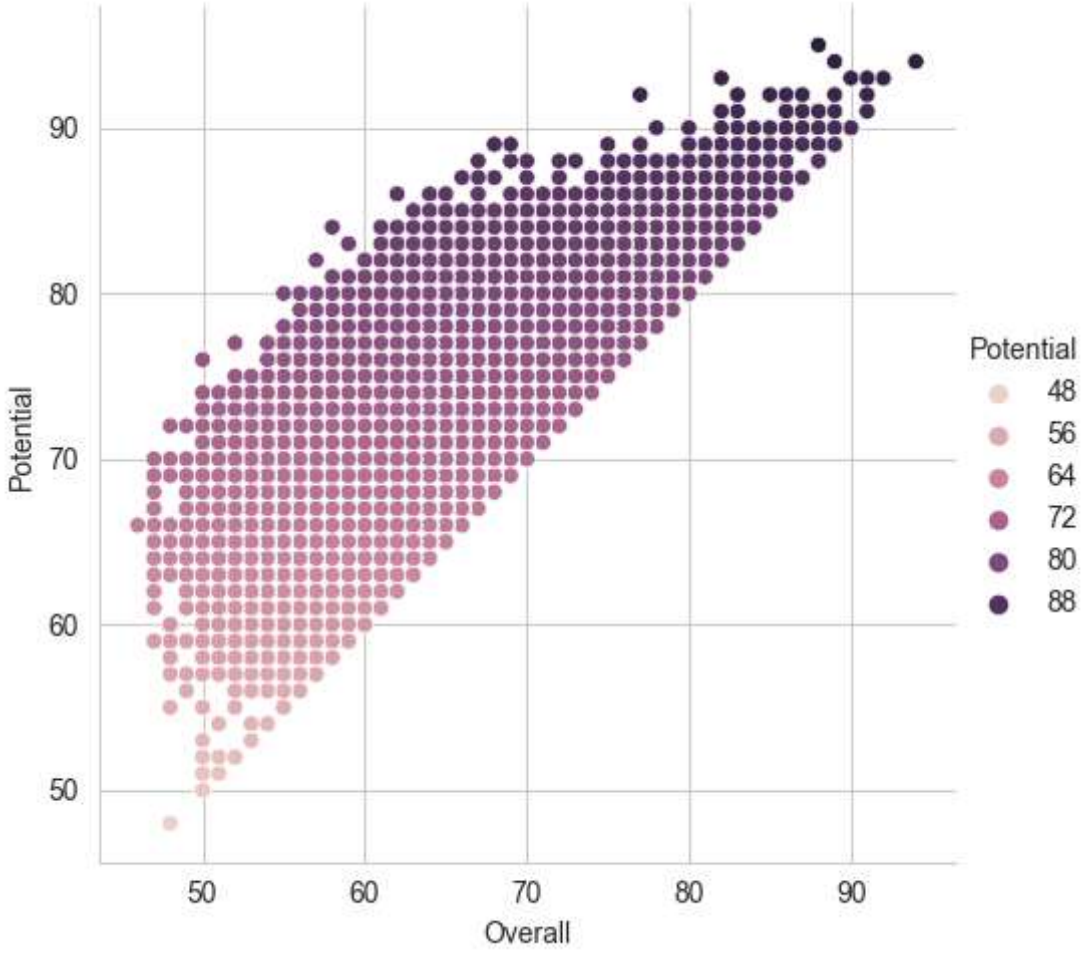


Visualizing statistical relationship with seaborn relplot() function

We can plot a scatterplot with variables Heigh and Weight with Seaborn relplot() function as follows-

In [42]:

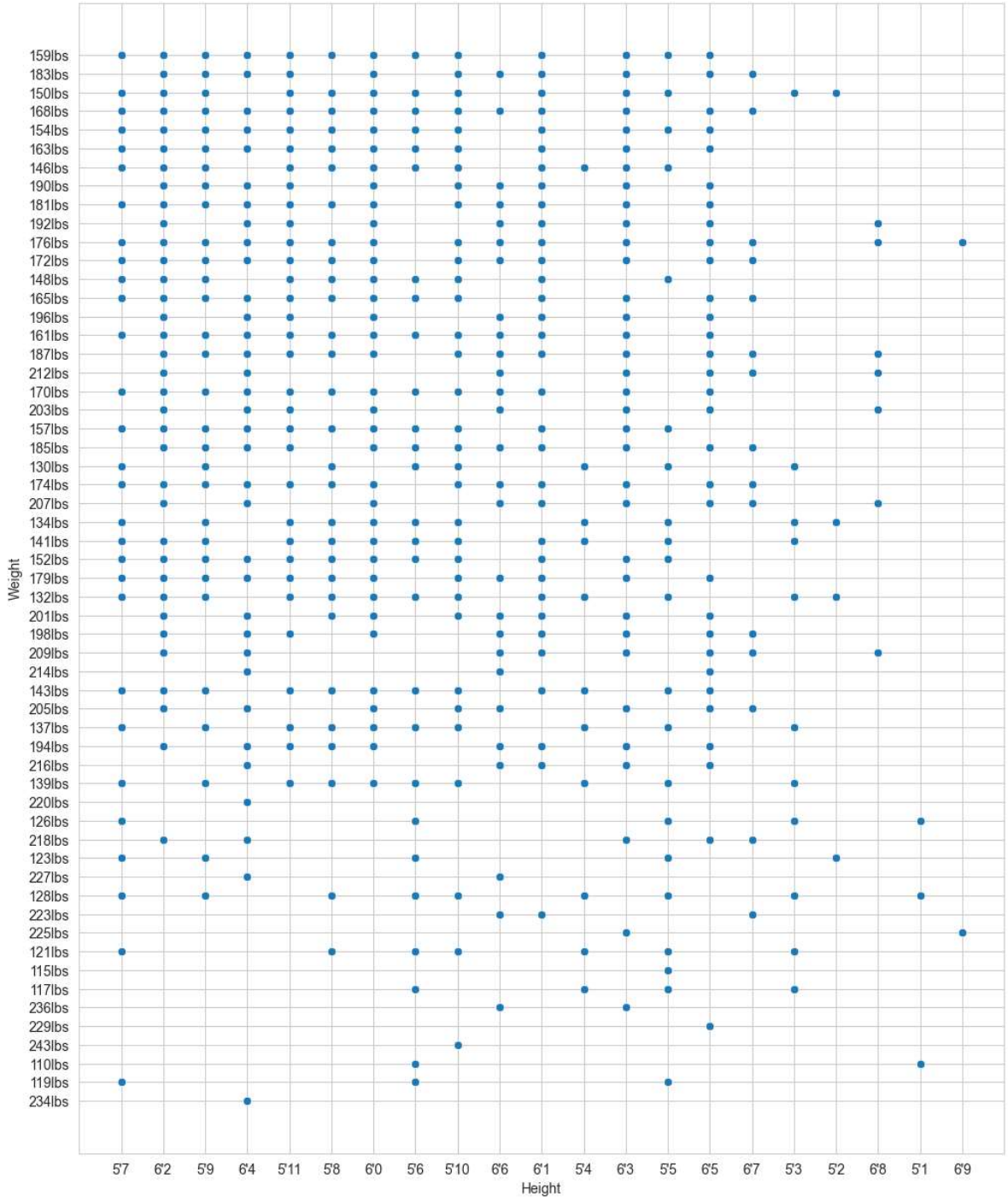
```
1 g=sns.relplot(data=fifa,x='Overall',y='Potential',hue='Potential')
```



Scatterplot()

In [43]:
```
1 f, ax=plt.subplots(figsize=(12,15))
2
3 sns.scatterplot(data=fifa,x='Height',y='Weight')
```

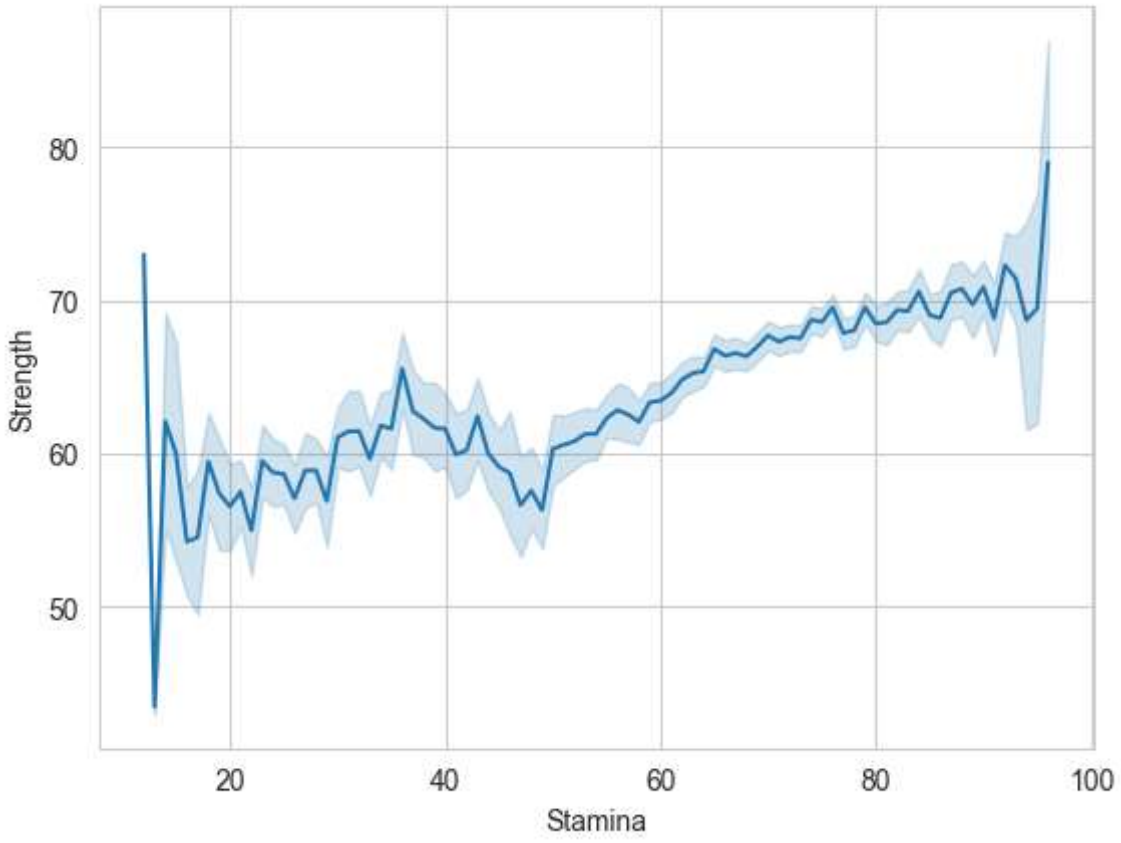
Out[43]: <Axes: xlabel='Height', ylabel='Weight'>



lineplot()

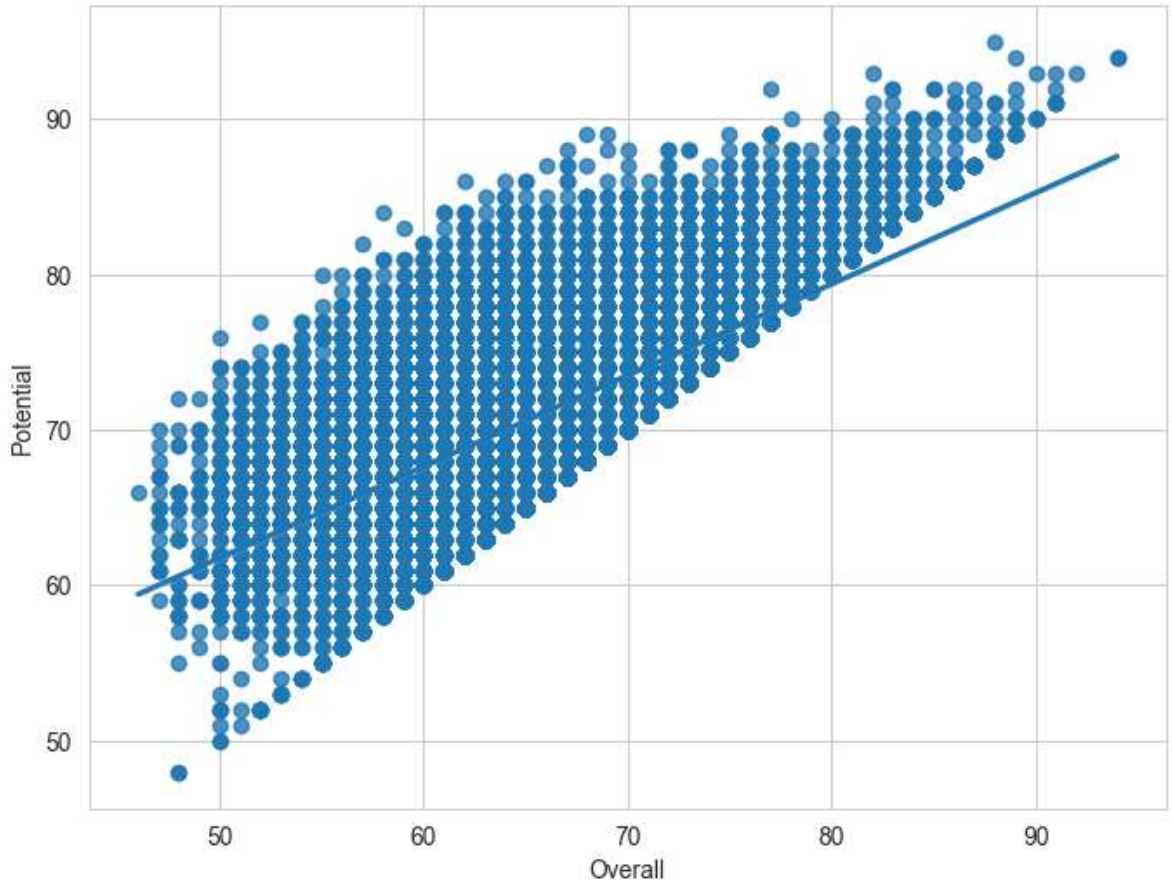
In [44]:
```
1 sns.lineplot(data=fifa,x='Stamina',y='Strength')
```

Out[44]: <Axes: xlabel='Stamina', ylabel='Strength'>



regplot()

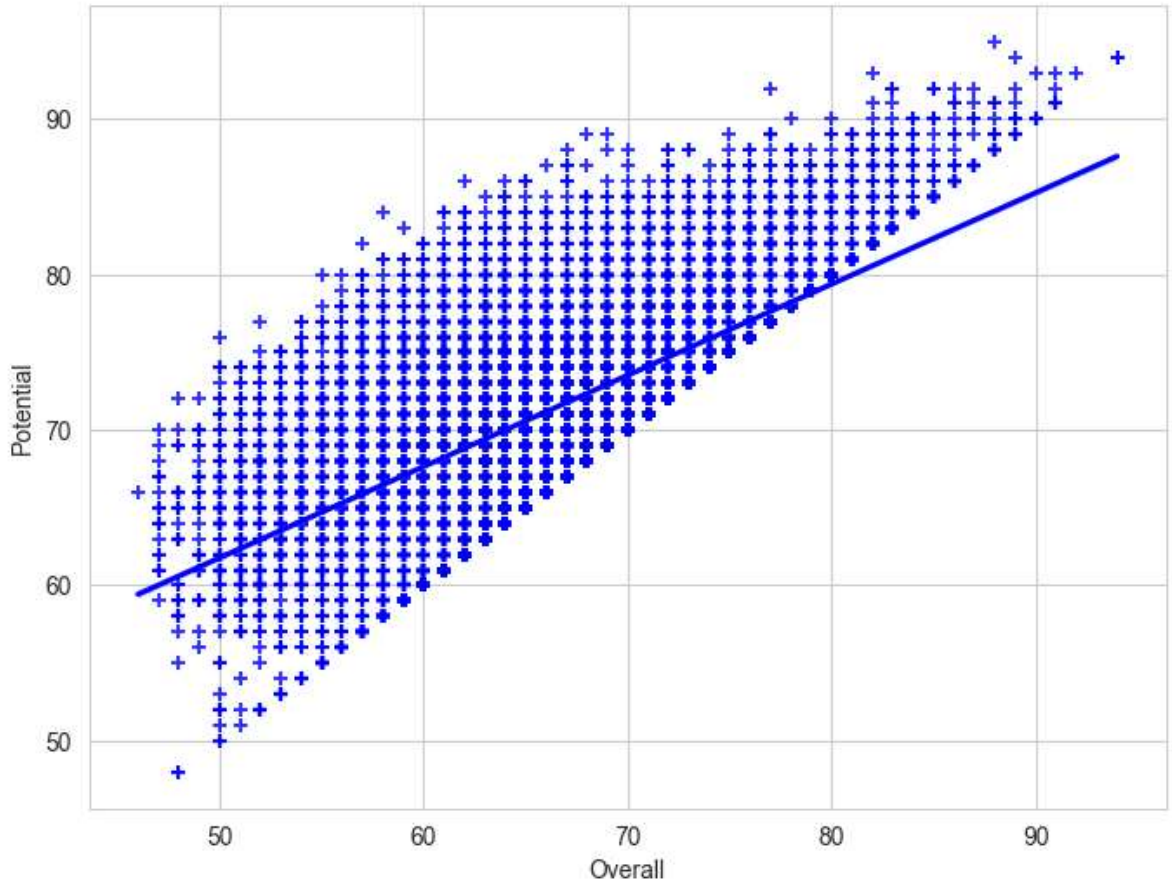
In [45]:
```
1 plt.subplots(figsize=(8,6))
2 g=sns.regplot(data=fifa,x='Overall',y='Potential')
```



We use a diffrent color and marker as follows

In [46]:

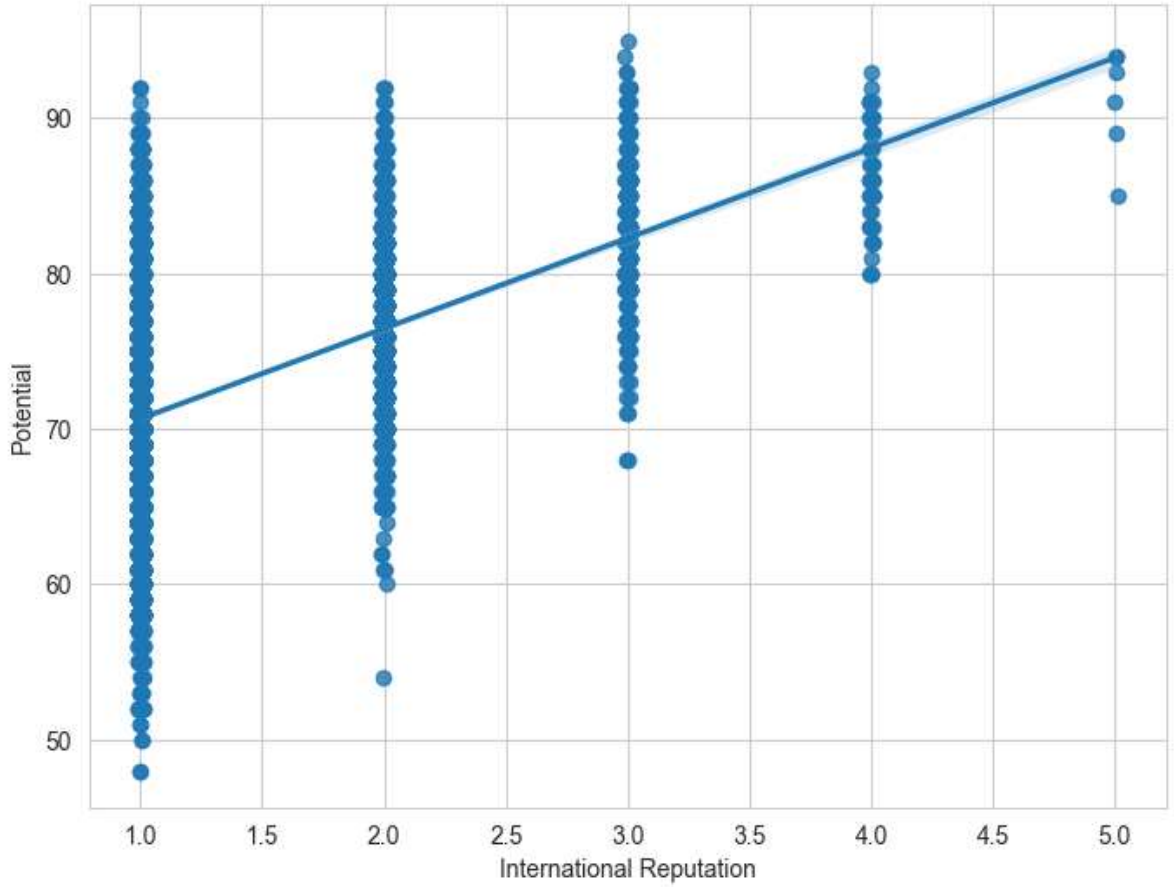
```
1 f, ax = plt.subplots(figsize=(8, 6))
2
3 ax=sns.regplot(data=fifa,x='Overall',y='Potential',color='b',marker='+')
```



We can plot a discrete variable and add some jitter as follows

In [47]:
```
1 f, ax=plt.subplots(figsize=(8,6))
2
3 sns.regplot(data=fifa,x='International Reputation',y='Potential',x_jitter=
```

Out[47]: <Axes: xlabel='International Reputation', ylabel='Potential'>

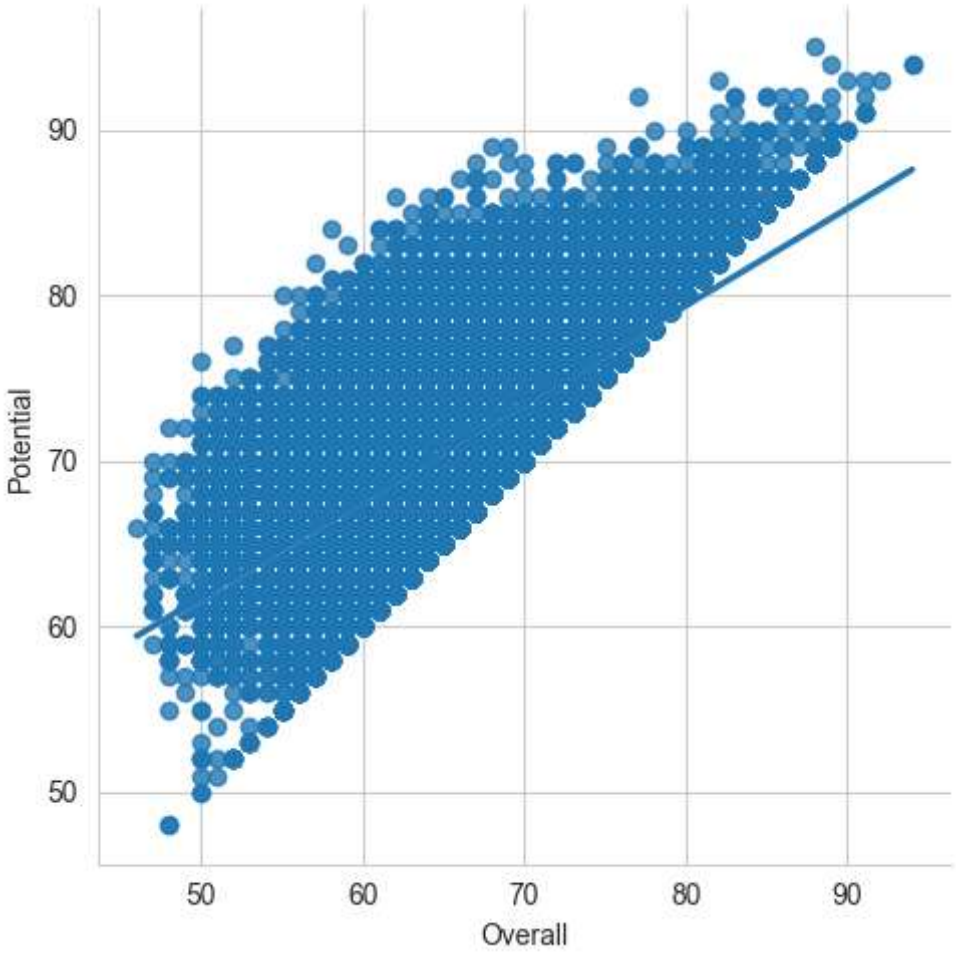


Implot()

- This function combine regplot() and FacetGrid

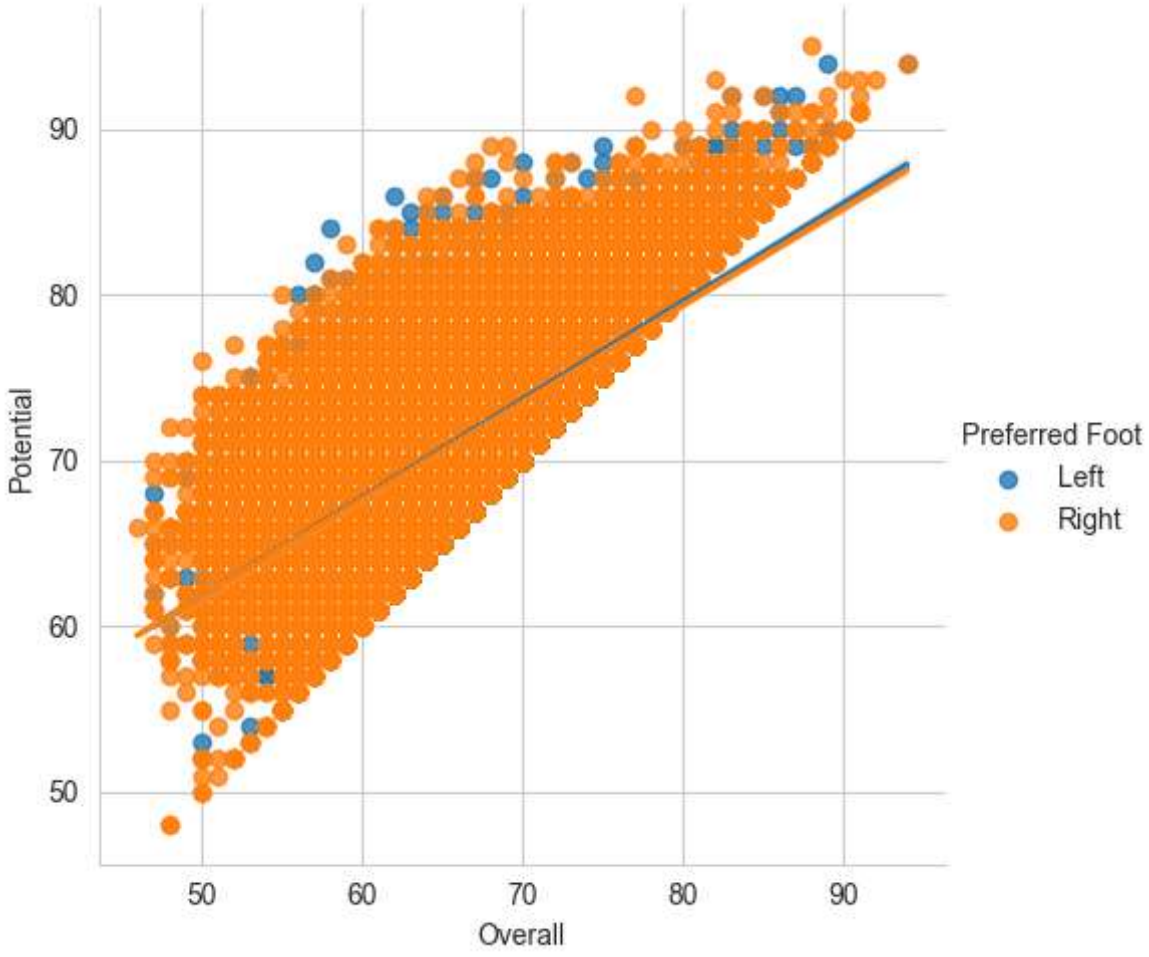
In [48]:
```
1 sns.lmplot(data=fifa,x='Overall',y='Potential')
```

Out[48]: <seaborn.axisgrid.FacetGrid at 0x1e7e4f22f90>



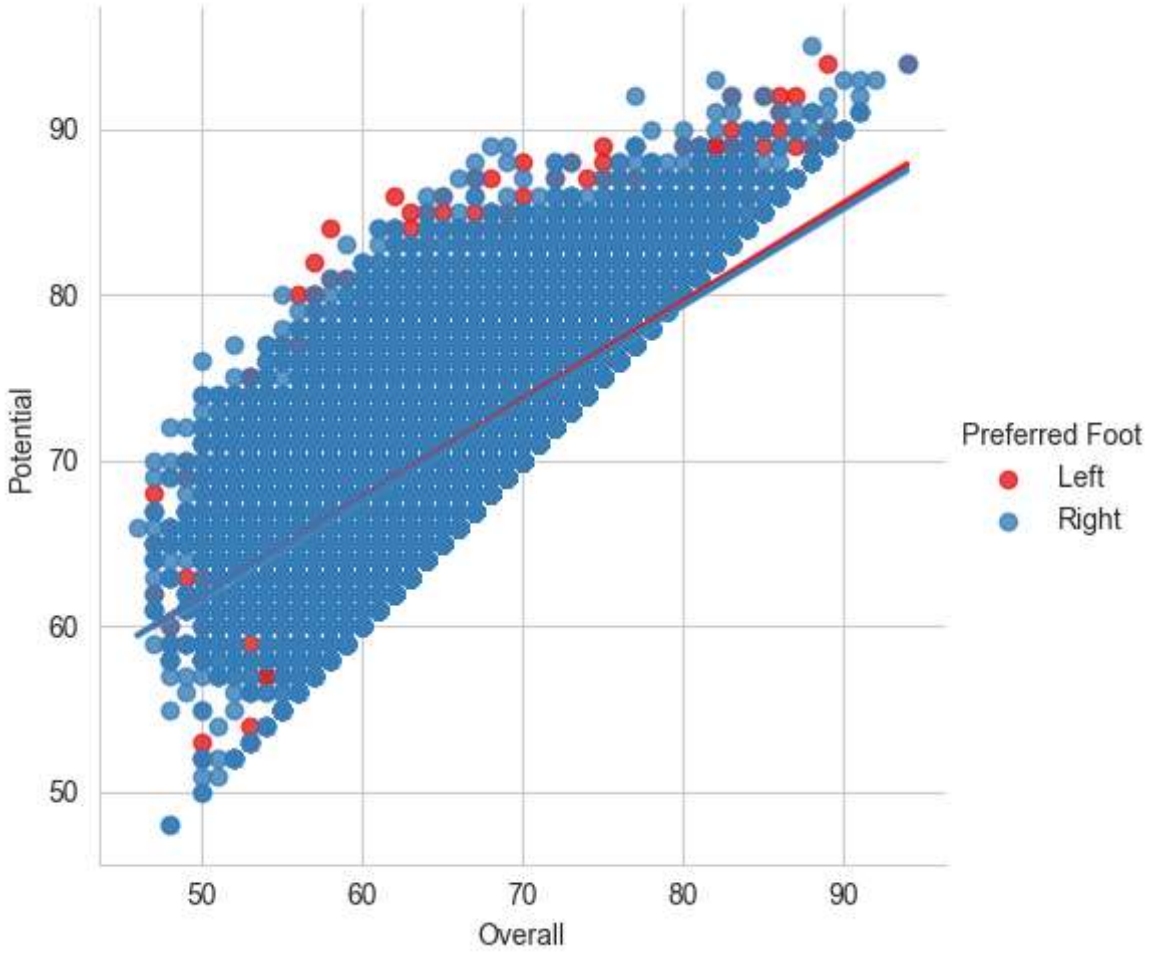
In [49]:
```
1 sns.lmplot(data=fifa,x='Overall',y='Potential',hue='Preferred Foot')
```

Out[49]: <seaborn.axisgrid.FacetGrid at 0x1e7e62263d0>



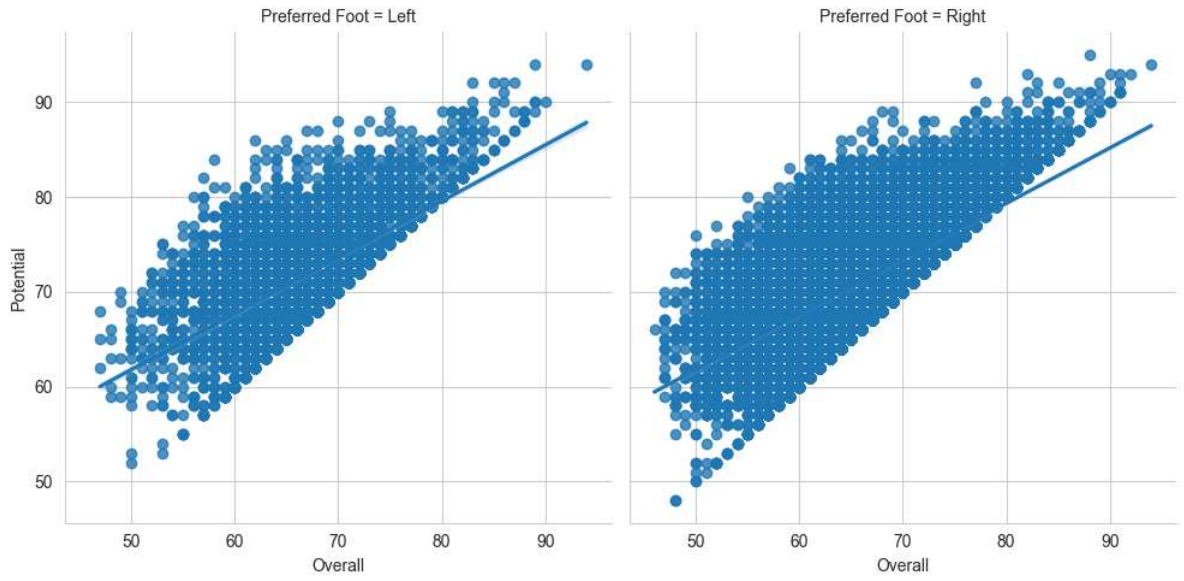
In [50]:
```
1 sns.lmplot(data=fifa,x='Overall',y='Potential',hue='Preferred Foot',palet
```

Out[50]: <seaborn.axisgrid.FacetGrid at 0x1e7e6243f10>



In [51]:
```
1 sns.lmplot(data=fifa,x='Overall',y='Potential',col='Preferred Foot')
```

Out[51]: <seaborn.axisgrid.FacetGrid at 0x1e7e5f263d0>

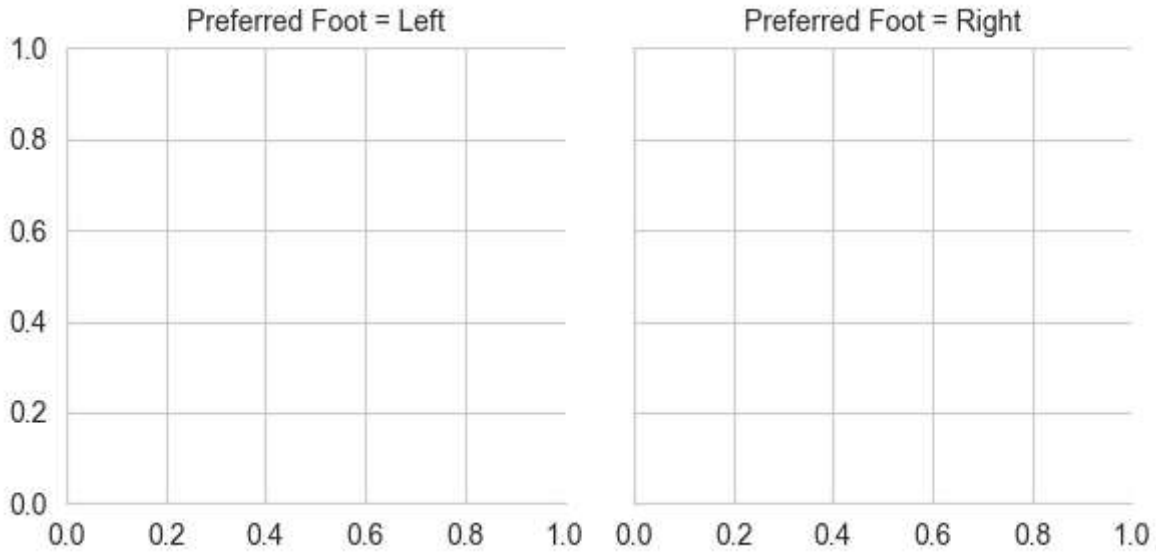


Multi-Plot Grids

FacetGrid()

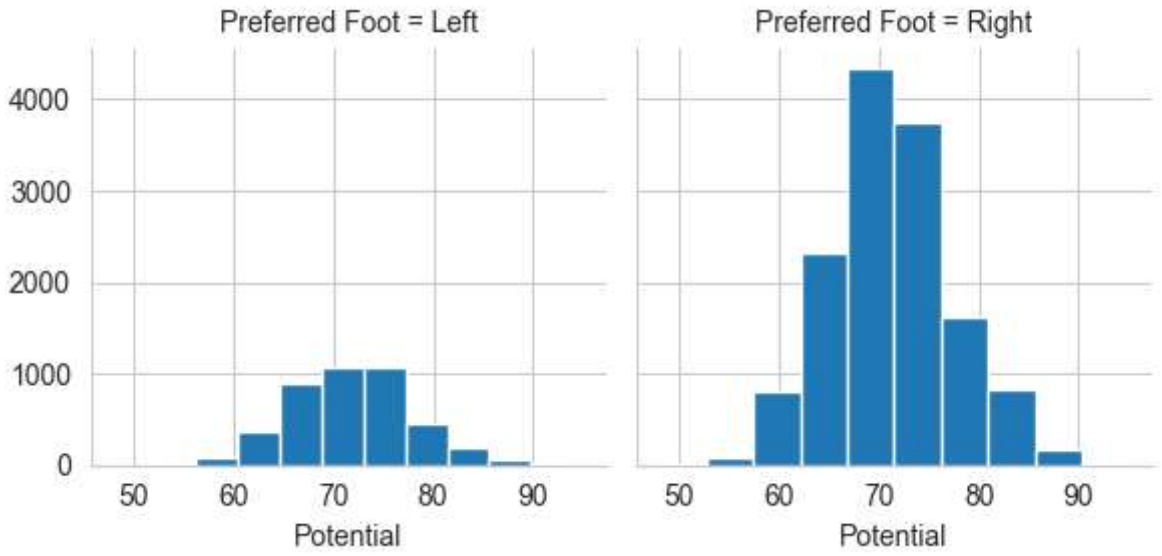
We can initialize a 1*2 grid of facet using the fifa dataset

In [52]:
```
1 g=sns.FacetGrid(fifa,col='Preferred Foot')
```

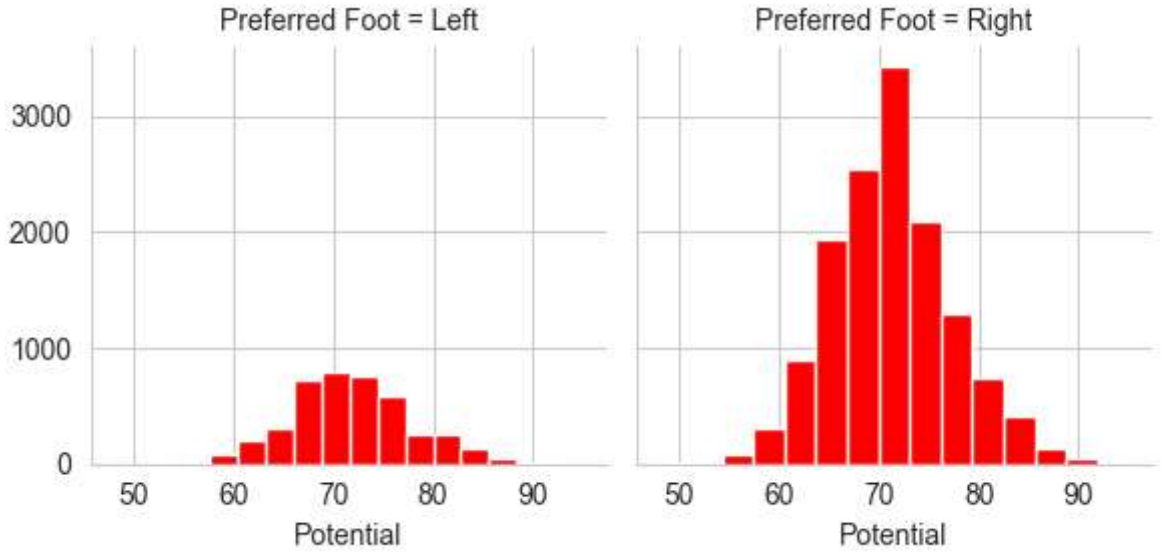


We can draw a univariate plot of potential variable on each Facet

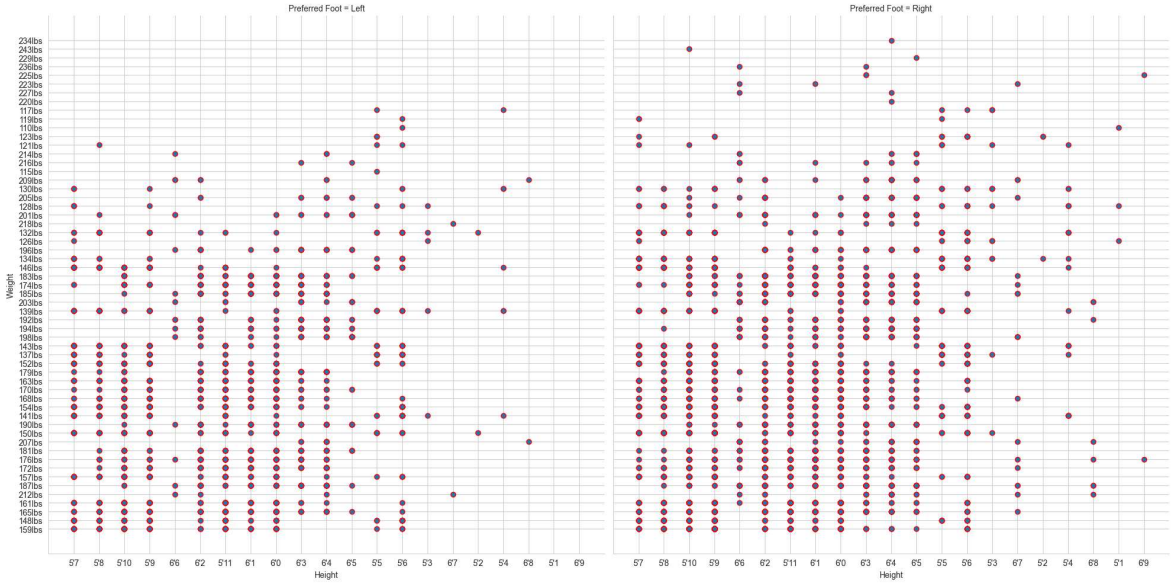
In [53]:
```
1 g=sns.FacetGrid(fifa,col='Preferred Foot')
2 g=g.map(plt.hist ,'Potential')
```



In [54]:
```
1 g=sns.FacetGrid(fifa,col='Preferred Foot')
2 g=g.map(plt.hist ,'Potential',bins=15,color='r')
```

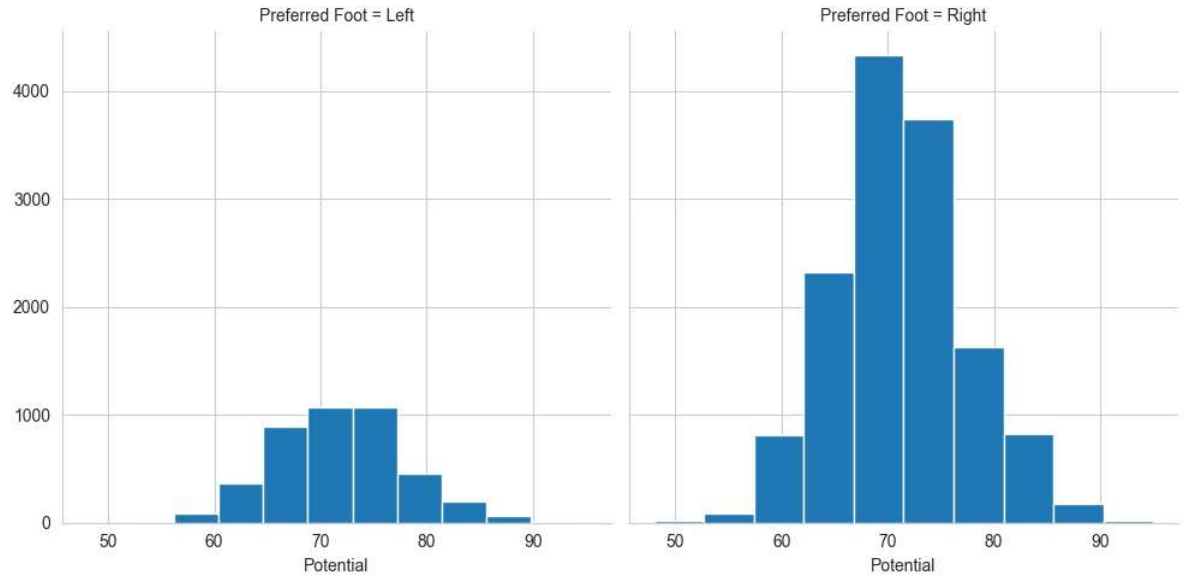


In [55]:
```
ax=sns.FacetGrid(fifa,col='Preferred Foot',height=10)
ax=ax.map(plt.scatter,'Height','Weight',edgecolor='r').add_legend()
```



In [56]:
```
g=sns.FacetGrid(fifa,col='Preferred Foot',height=5,aspect=1)
g.map(plt.hist,'Potential')
```

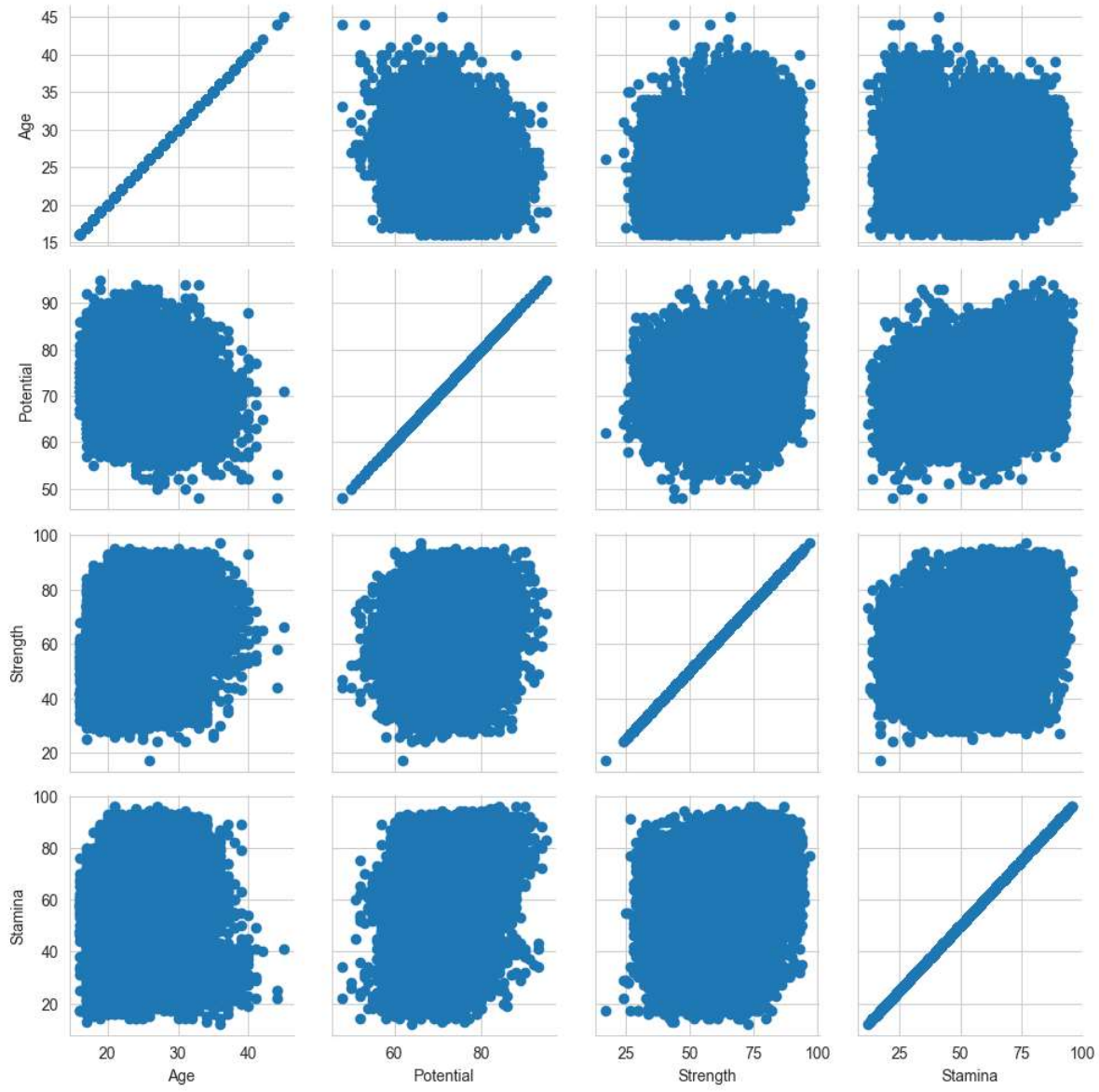
Out[56]: <seaborn.axisgrid.FacetGrid at 0x1e7e60eb3d0>



Pairgrid()

In [57]:
```
1 fifa_num=fifa[['Age','Potential','Strength','Stamina','Preferred Foot']]
2 g=sns.PairGrid(fifa_num,aspect=1)
3 g.map(plt.scatter)
```

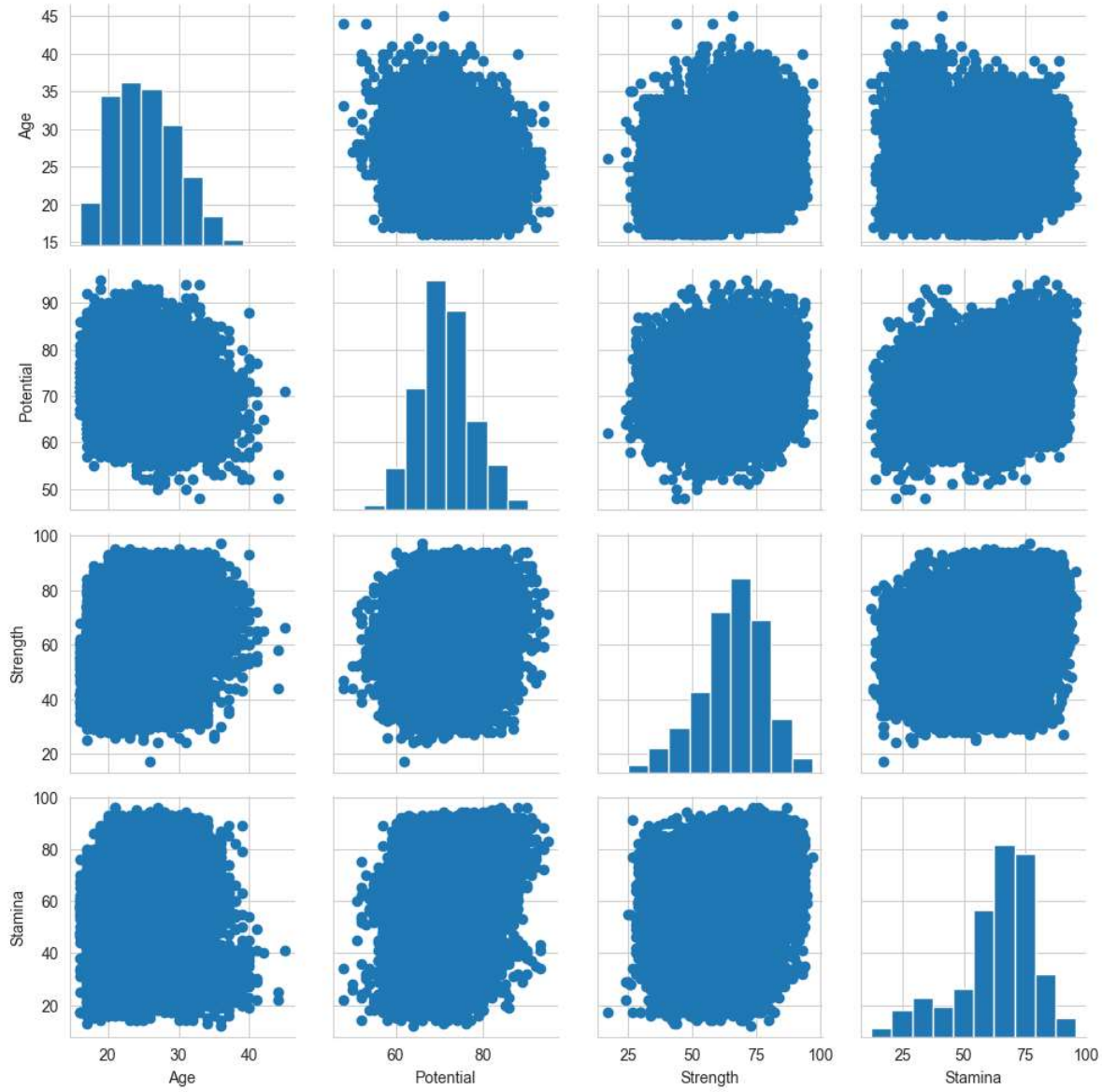
Out[57]: <seaborn.axisgrid.PairGrid at 0x1e7ee7cb310>



We can show a univariate distribution on the diagonal as follows

In [58]:

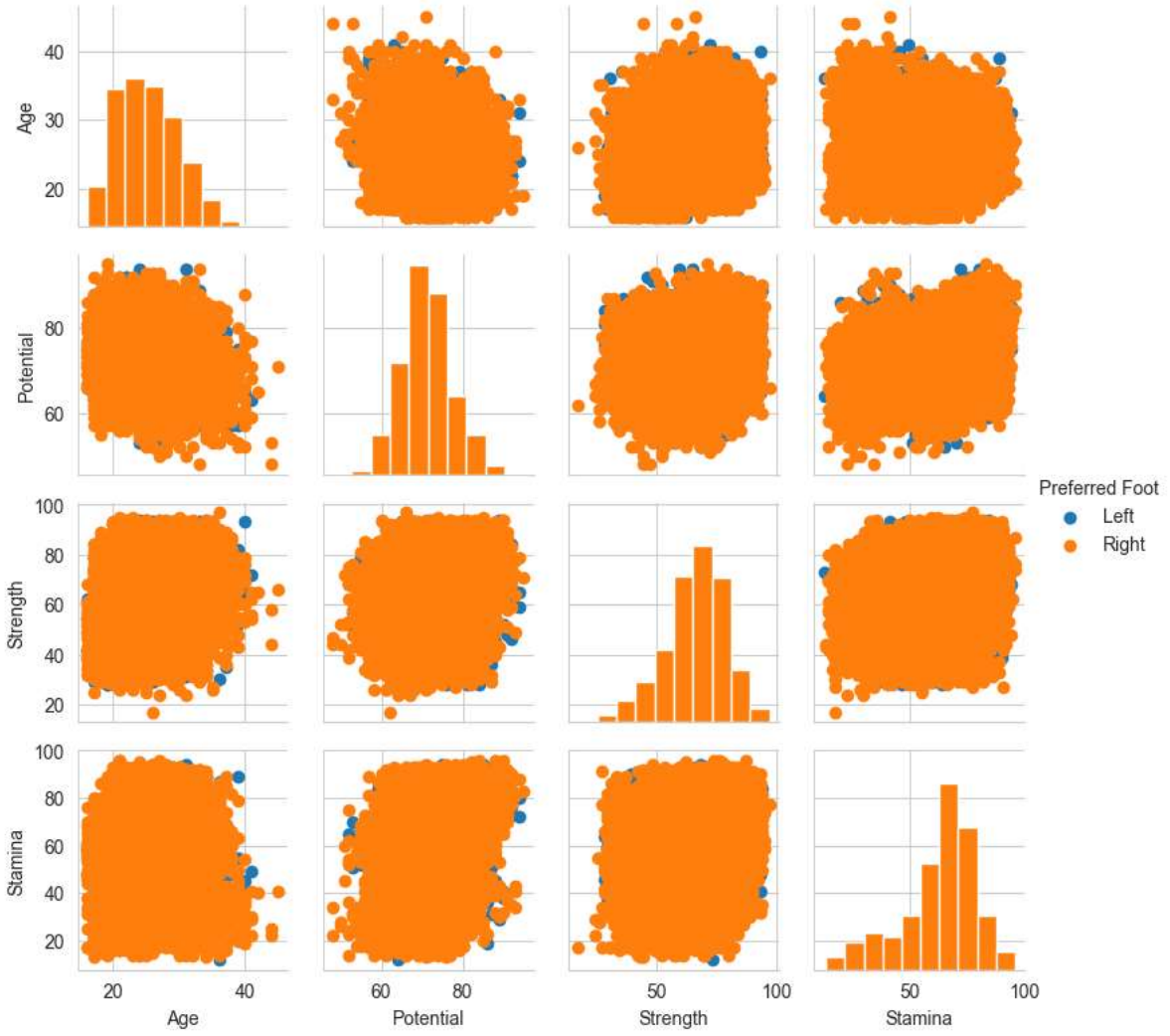
```
1 g=sns.PairGrid(fifa_num)
2 g=g.map_diag(plt.hist)
3 g=g.map_offdiag(plt.scatter)
```



We can color the points using the categorical variable Preferred Foot as follows

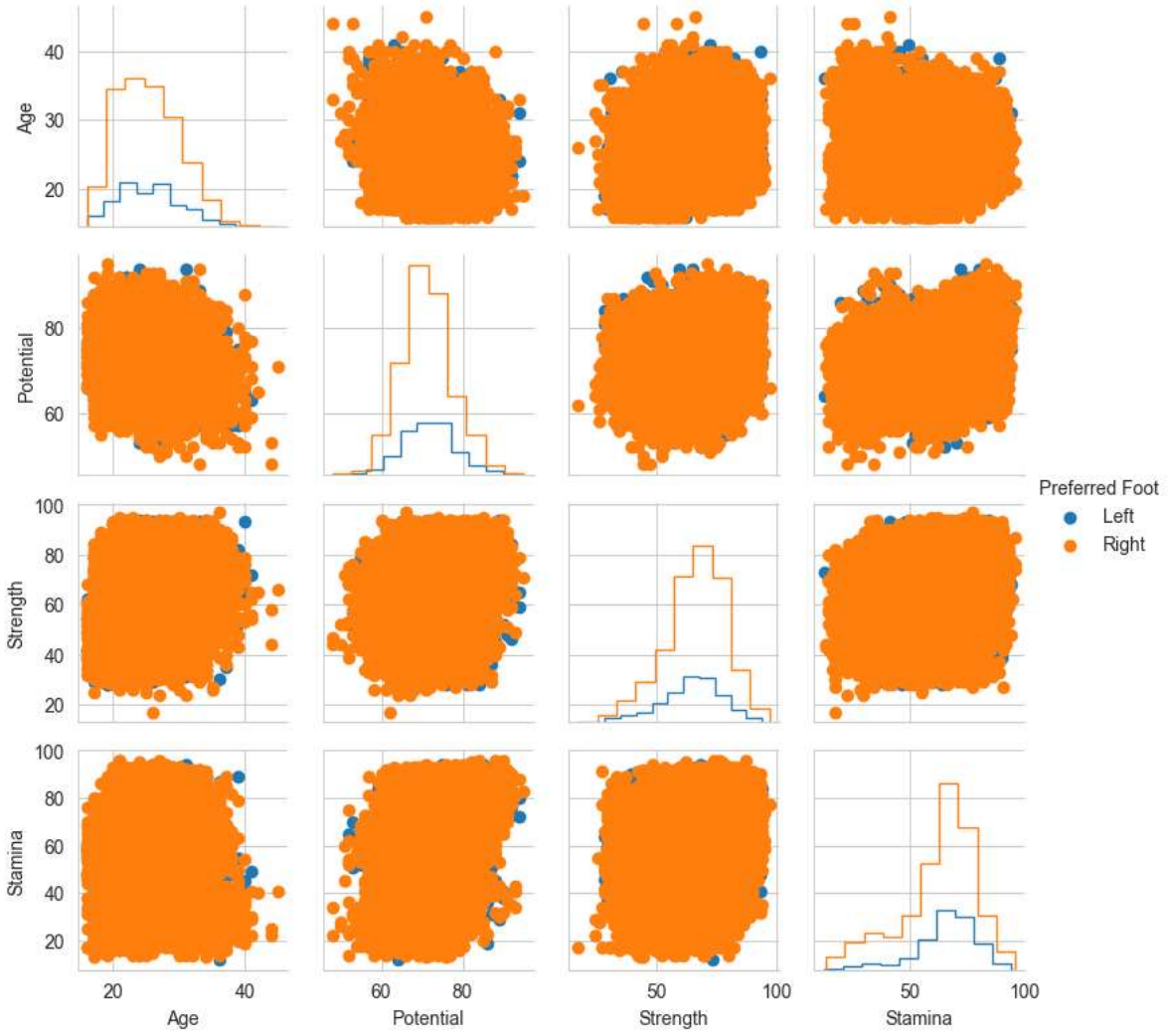
In [59]:
```
g=sns.PairGrid(fifa_num,hue='Preferred Foot',height=2)
g=g.map_diag(plt.hist)
g=g.map_offdiag(plt.scatter)
g.add_legend()
```

Out[59]: <seaborn.axisgrid.PairGrid at 0x1e7e4de5b10>



In [60]:
```
g=sns.PairGrid(fifa_num,hue='Preferred Foot',height=2)
g=g.map_diag(plt.hist,histtype='step')
g=g.map_offdiag(plt.scatter)
g.add_legend()
```

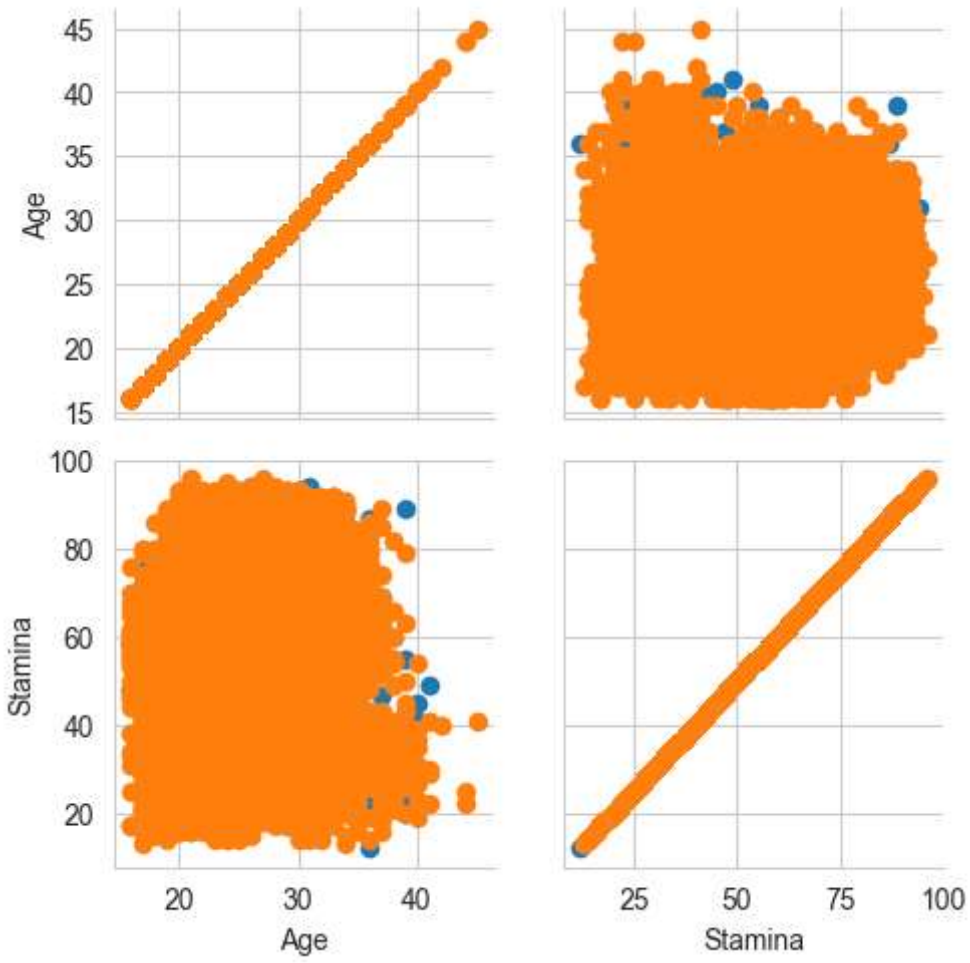
Out[60]: <seaborn.axisgrid.PairGrid at 0x1e7ef7c4410>



We can plot a subset of variables as follows

In [61]:
```
g=sns.PairGrid(fifa_num,vars=('Age','Stamina'),hue='Preferred Foot')
g.map(plt.scatter)
```

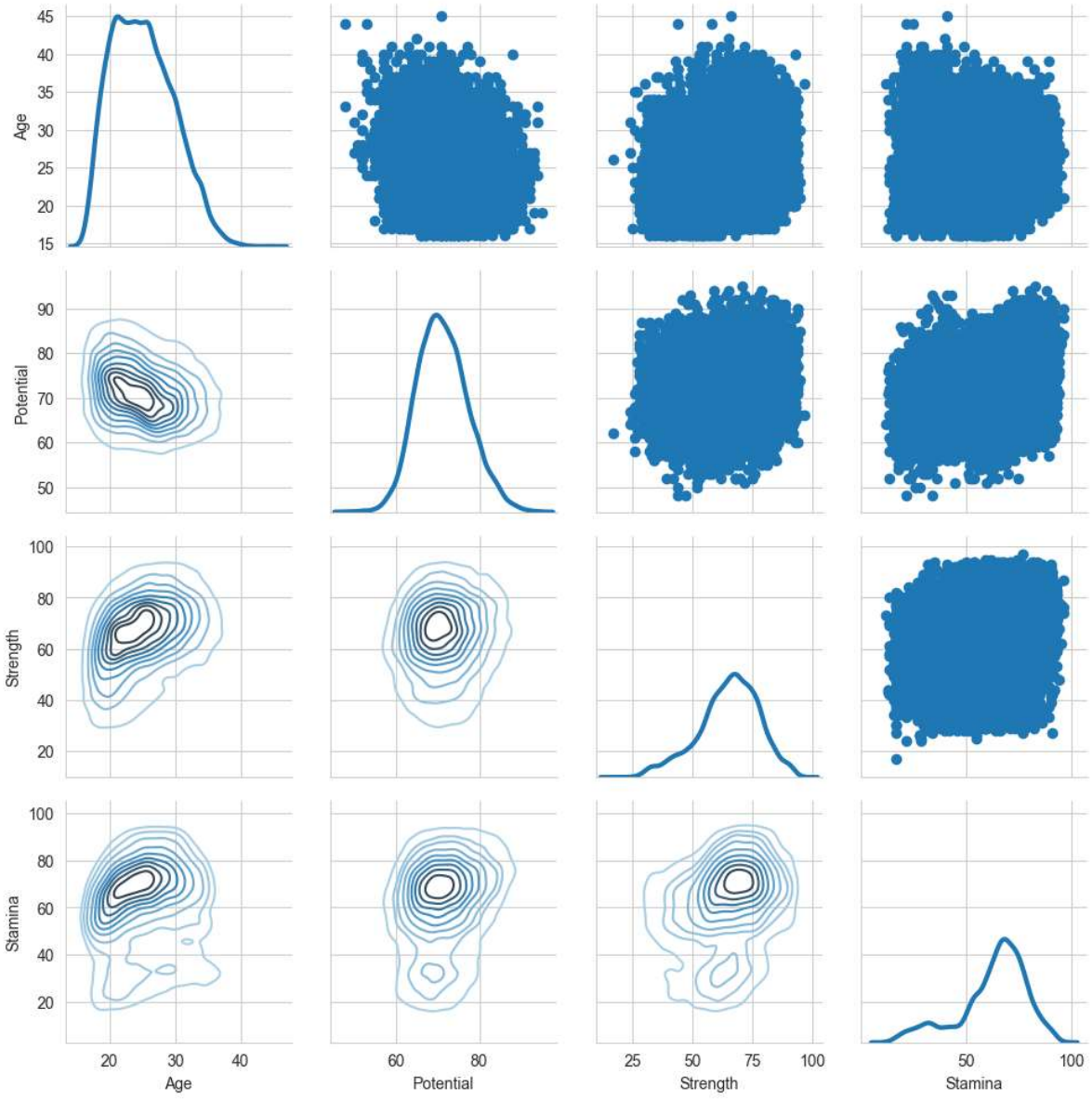
Out[61]: <seaborn.axisgrid.PairGrid at 0x1e7ec798e50>



We can use different functions on the upper and lower triangles as follows

In [62]:

```
g=sns.PairGrid(fifa_num)
g=g.map_upper(plt.scatter)
g=g.map_lower(sns.kdeplot,cmap='Blues_d')
g=g.map_diag(sns.kdeplot,lw=3)
```

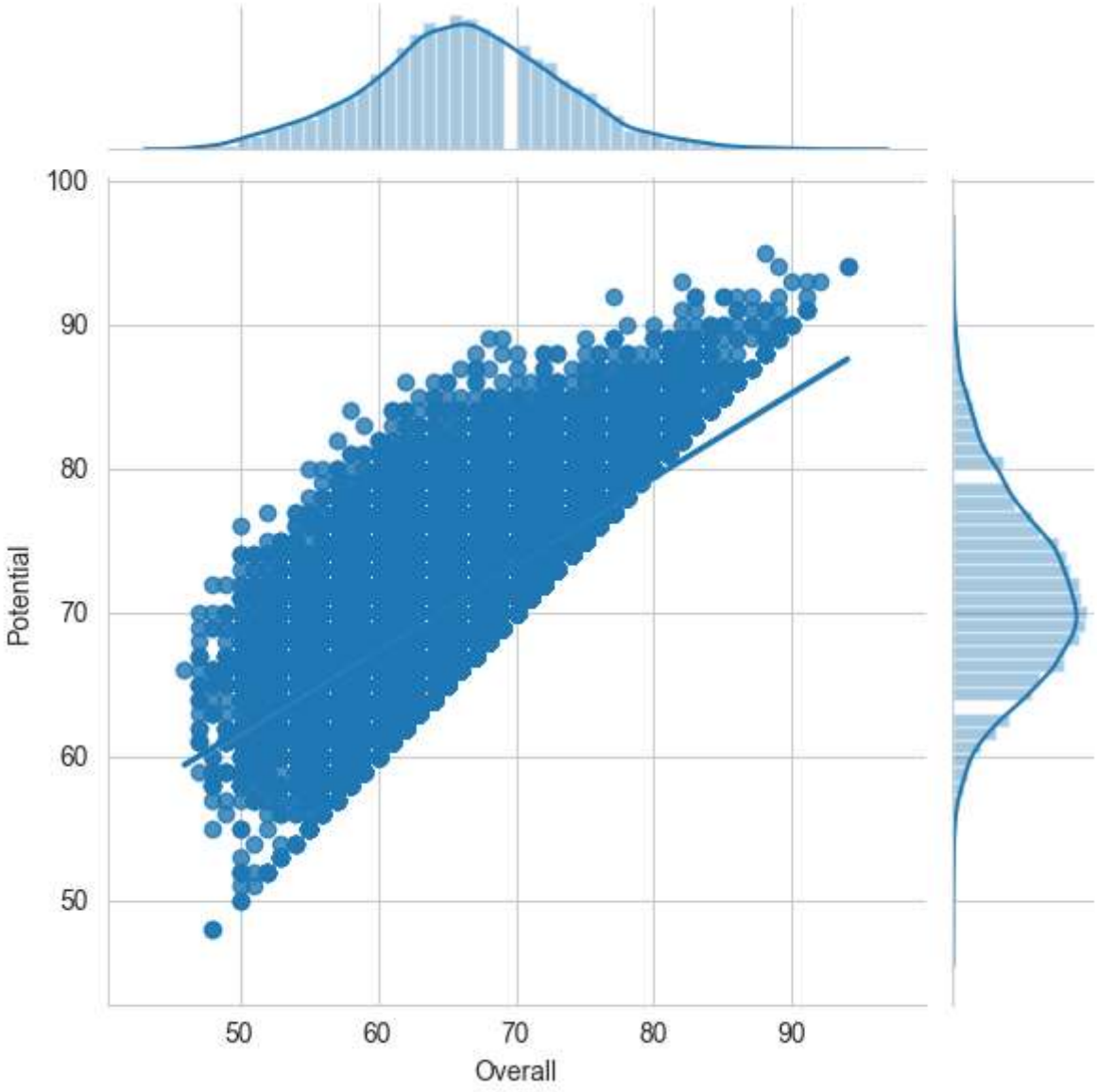


Jointgrid()

We can initialize the figure and plots using default parameters as follows

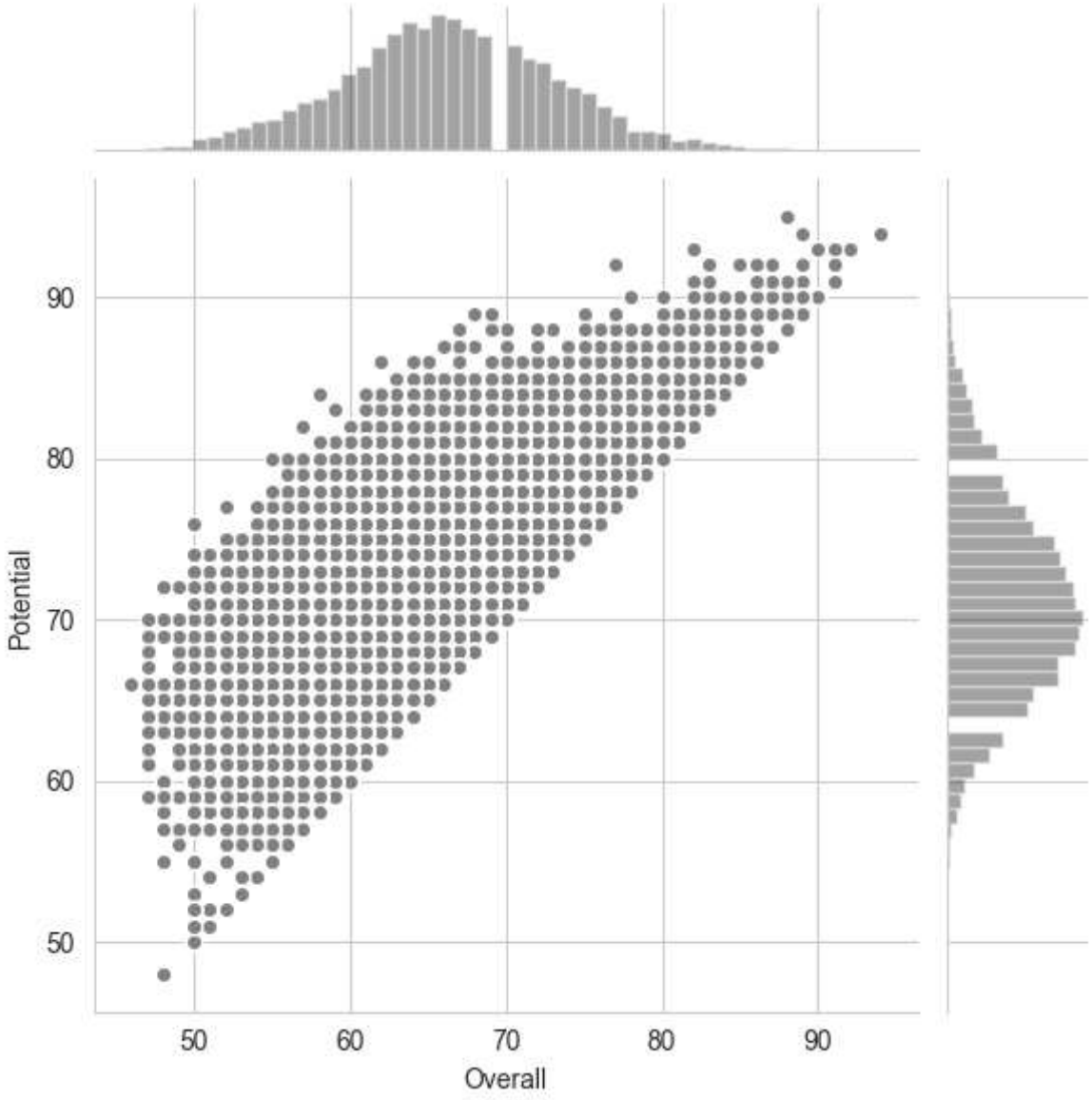
In [63]:
```
g=sns.JointGrid(data=fifa,x='Overall',y='Potential')
g.plot(sns.regplot,sns.distplot)
```

Out[63]: <seaborn.axisgrid.JointGrid at 0x1e7f239d2d0>



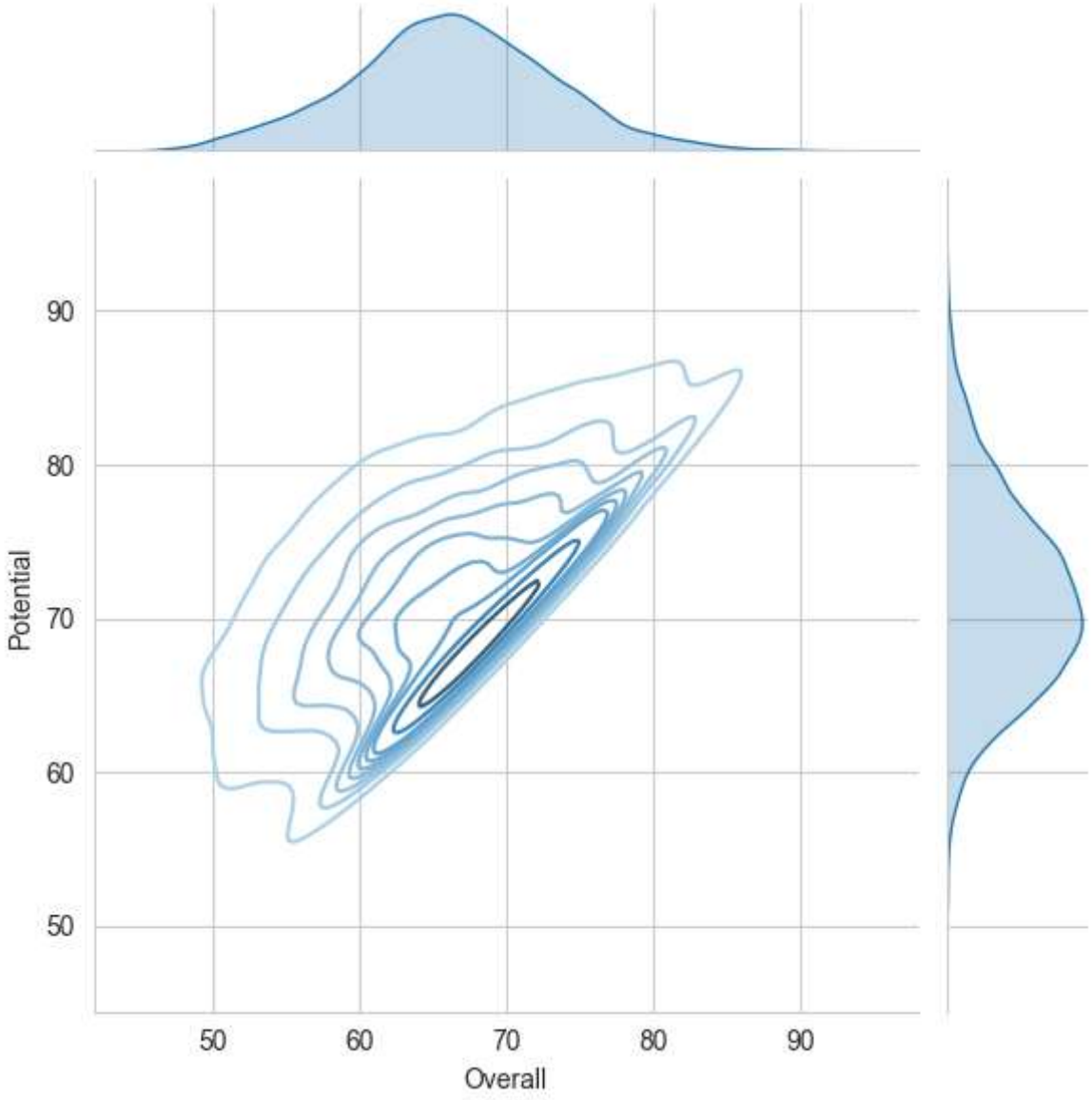
In [64]:
```
g=sns.JointGrid(data=fifa,x='Overall',y='Potential')
g.plot_joint(plt.scatter,color='0.5',edgecolor='white')
g.plot_marginals(sns.distplot,kde=False,color='0.1')
```

Out[64]: <seaborn.axisgrid.JointGrid at 0x1e7f3a37410>



In [65]:
```
g=sns.JointGrid(data=fifa,x='Overall',y='Potential')
g.plot_joint(sns.kdeplot,cmap='Blues_d')
g.plot_marginals(sns.kdeplot,shade=True)
```

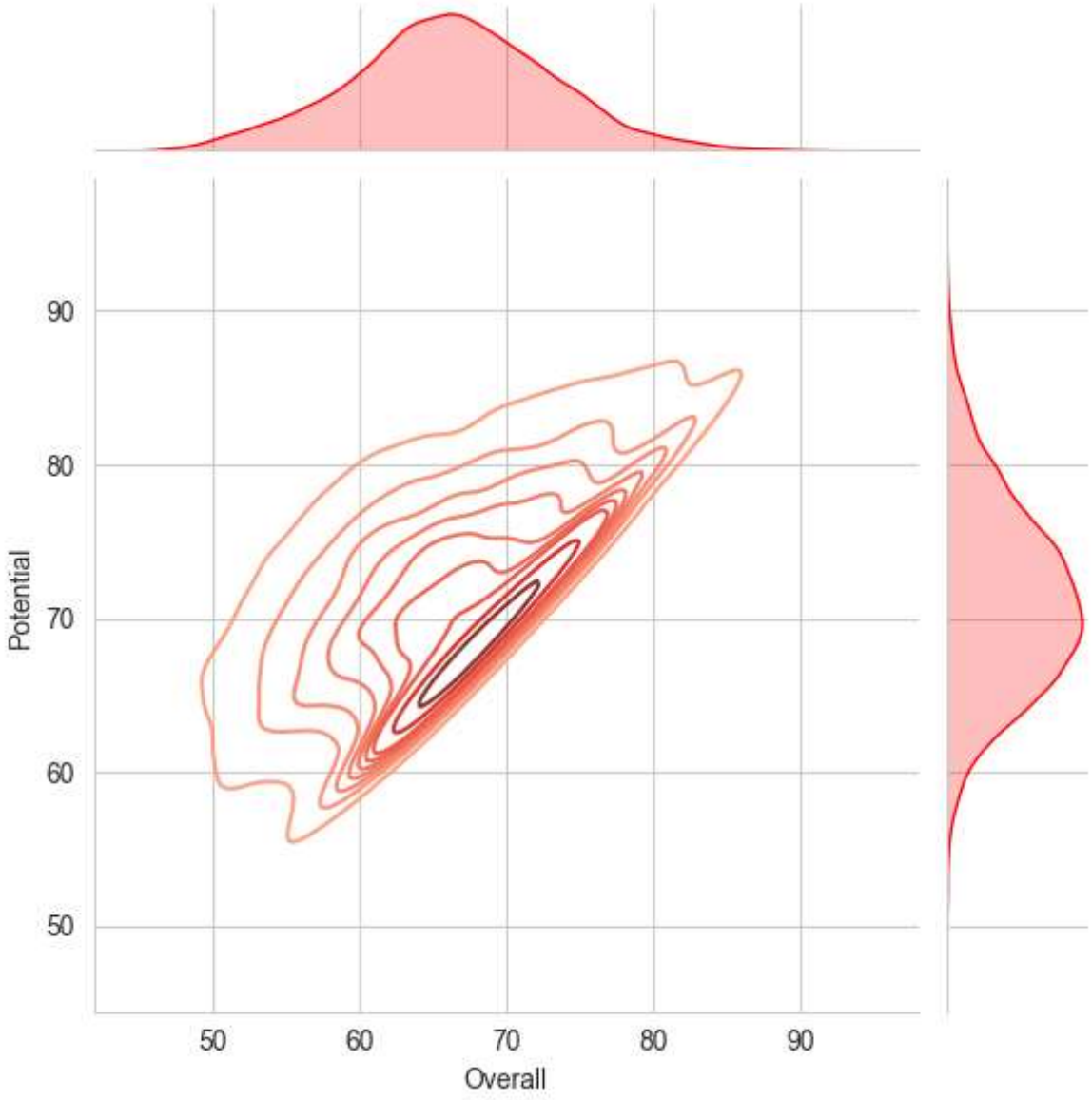
Out[65]: <seaborn.axisgrid.JointGrid at 0x1e7f3c4f410>



we can draw a smaller plot with relatively larger marginal axes as follows

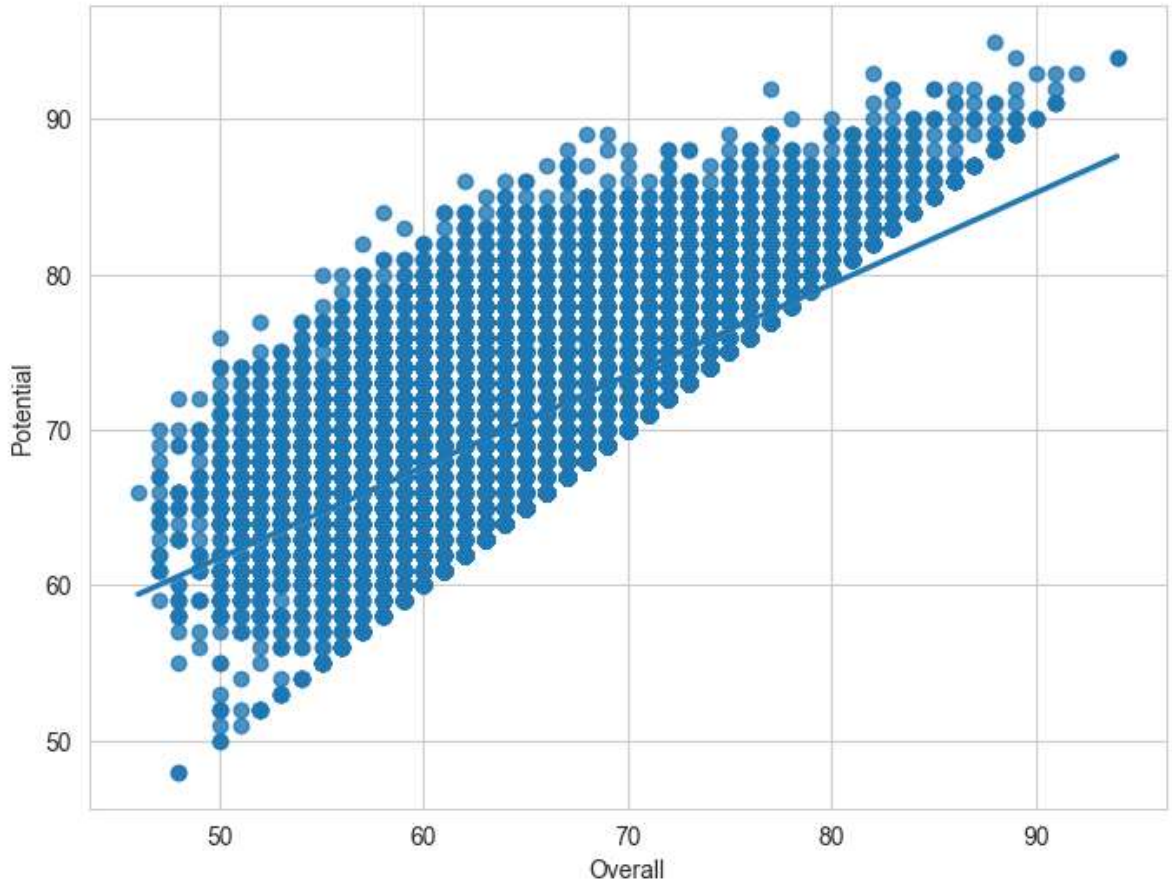
In [66]:
```
1 g=sns.JointGrid(data=fifa,x='Overall',y='Potential')
2 g.plot_joint(sns.kdeplot,cmap='Reds_d')
3 g.plot_marginals(sns.kdeplot,shade=True,color='r')
```

Out[66]: <seaborn.axisgrid.JointGrid at 0x1e7eb135010>



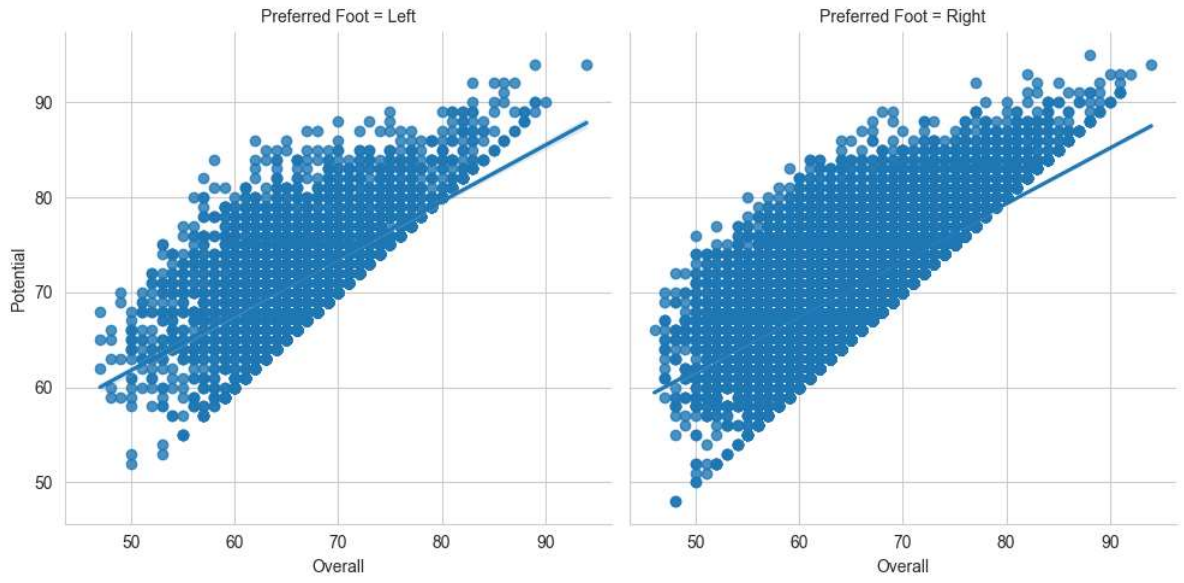
Controling the size and shape of the plot

In [67]:
```
f, ax=plt.subplots(figsize=(8,6))
ax=sns.regplot(data=fifa,x='Overall',y='Potential')
```



In [71]:
```
1 sns.lmplot(data=fifa,x='Overall',y='Potential',col='Preferred Foot')
```

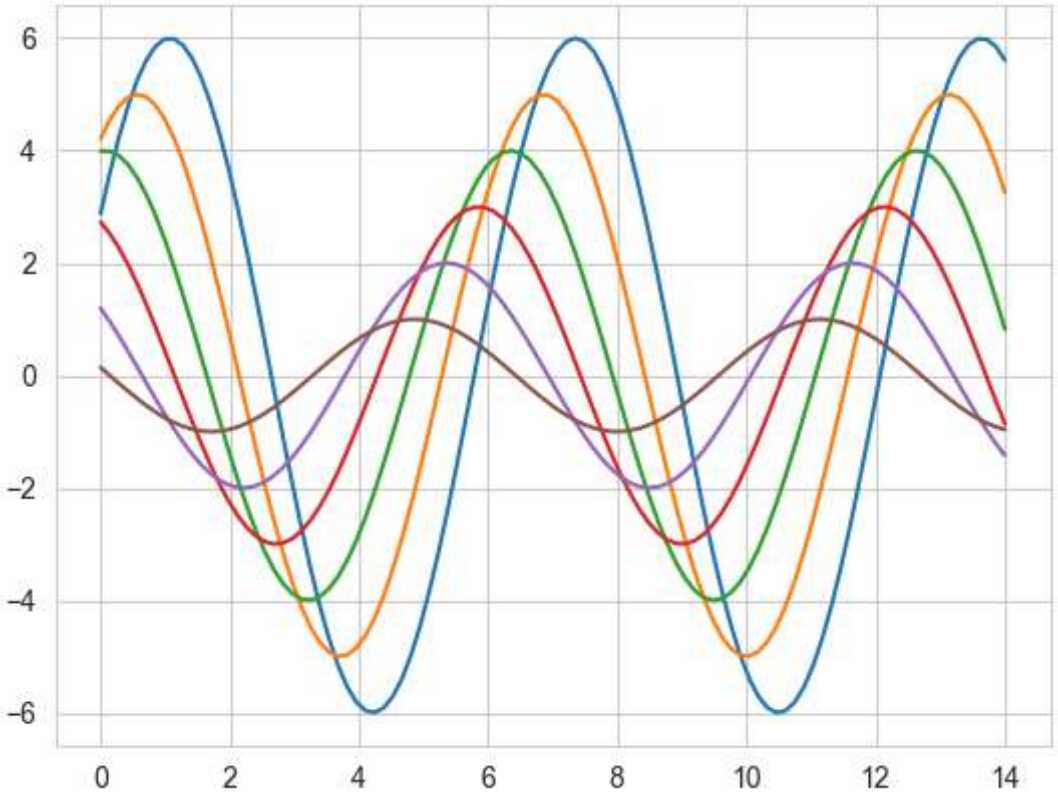
Out[71]: <seaborn.axisgrid.FacetGrid at 0x1e7eff33690>



Sine Waves

In [84]:
```
1 def sinplot(flip=1):
2     x=np.linspace(0,14,100)
3     for i in range(1,7):
4         plt.plot(x,np.sin(x+i*0.5)*(7-i)*flip)
```

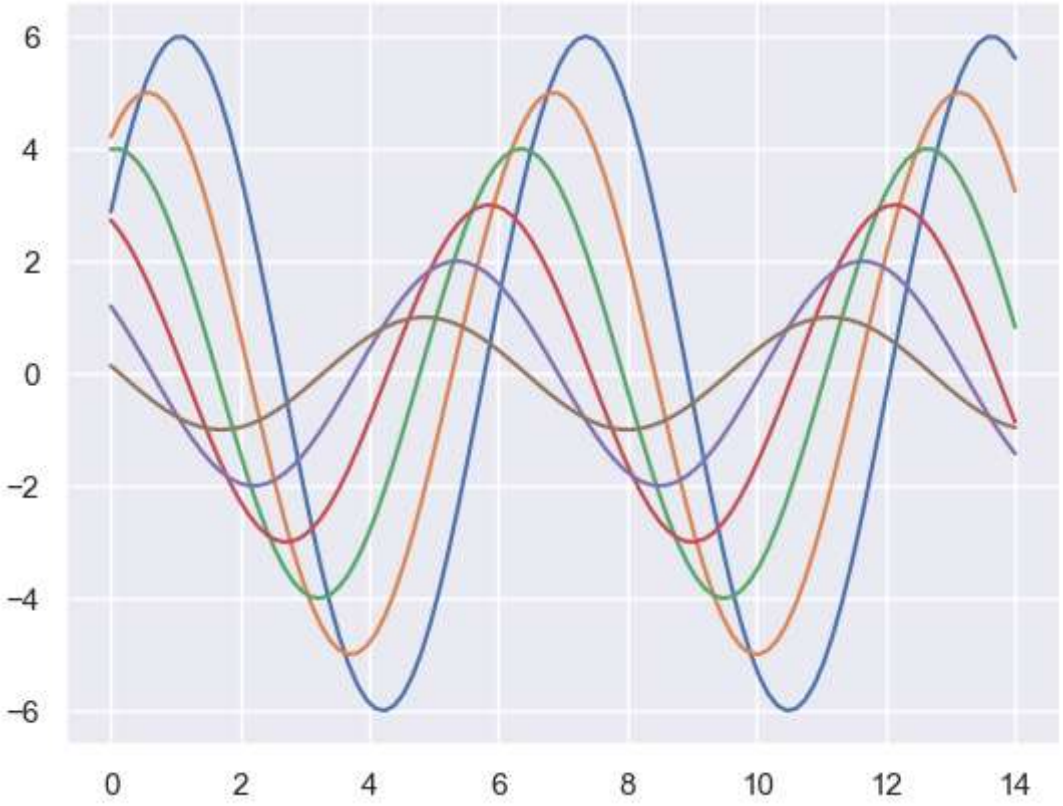
In [85]:
```
1 sinplot()
```



To switch to seaborn defaults, we need to call the set() function as follows -

In [86]:

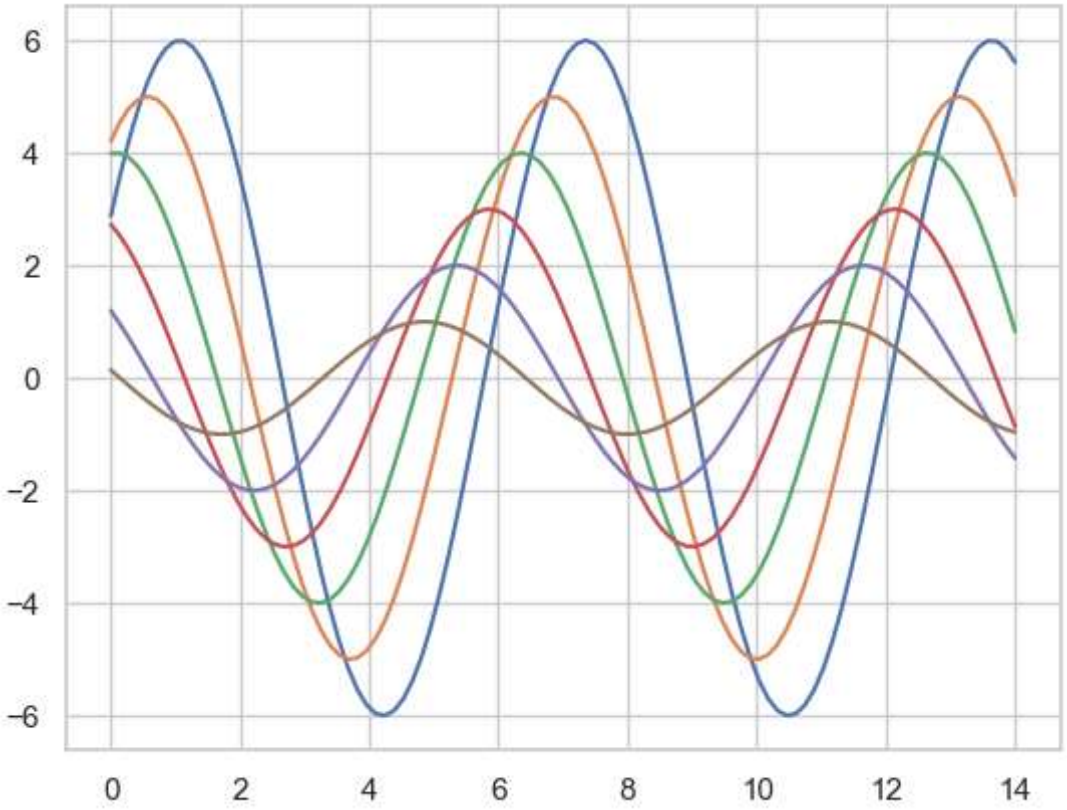
```
1 sns.set()
2 sinplot()
```



We can set different styles as follows

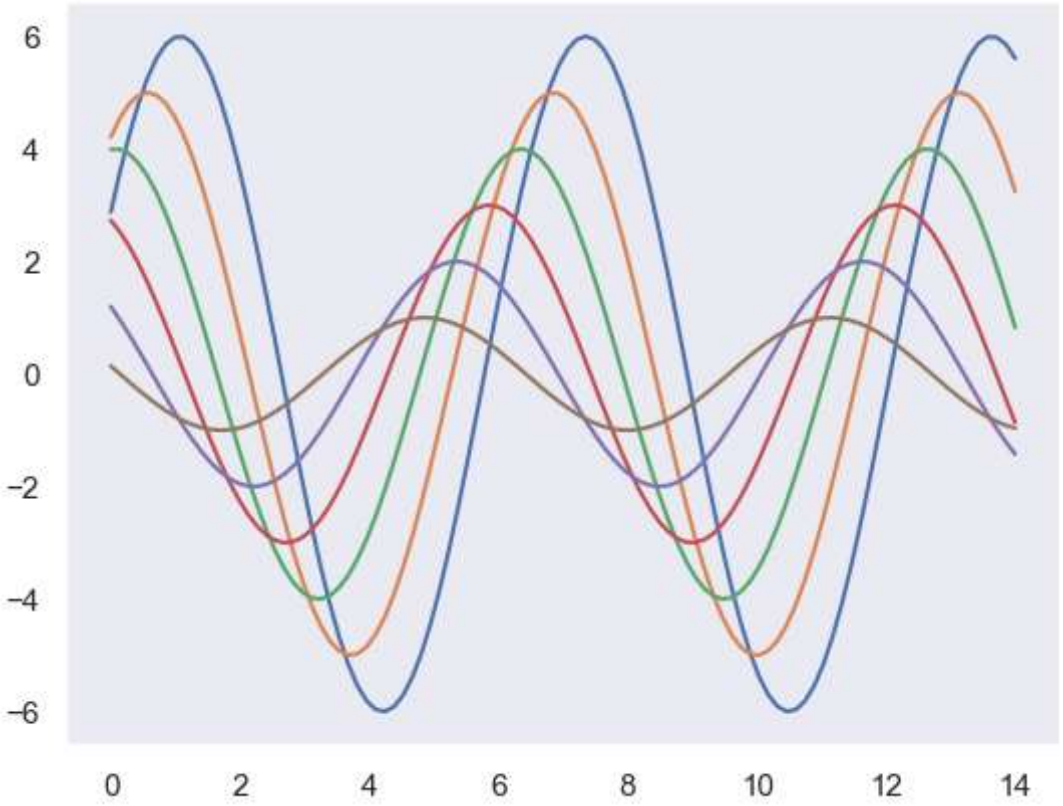
In [87]:

```
1 sns.set_style('whitegrid')
2 sinplot()
```



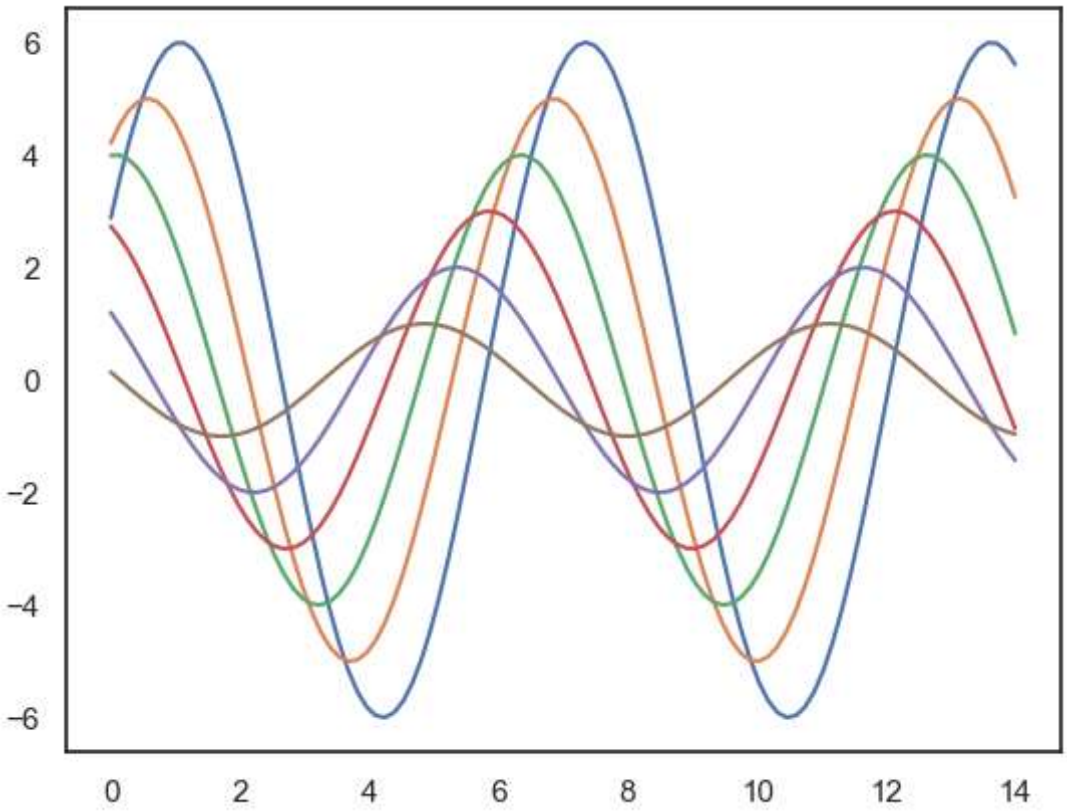
In [88]:

```
1 sns.set_style('dark')
2 sinplot()
```

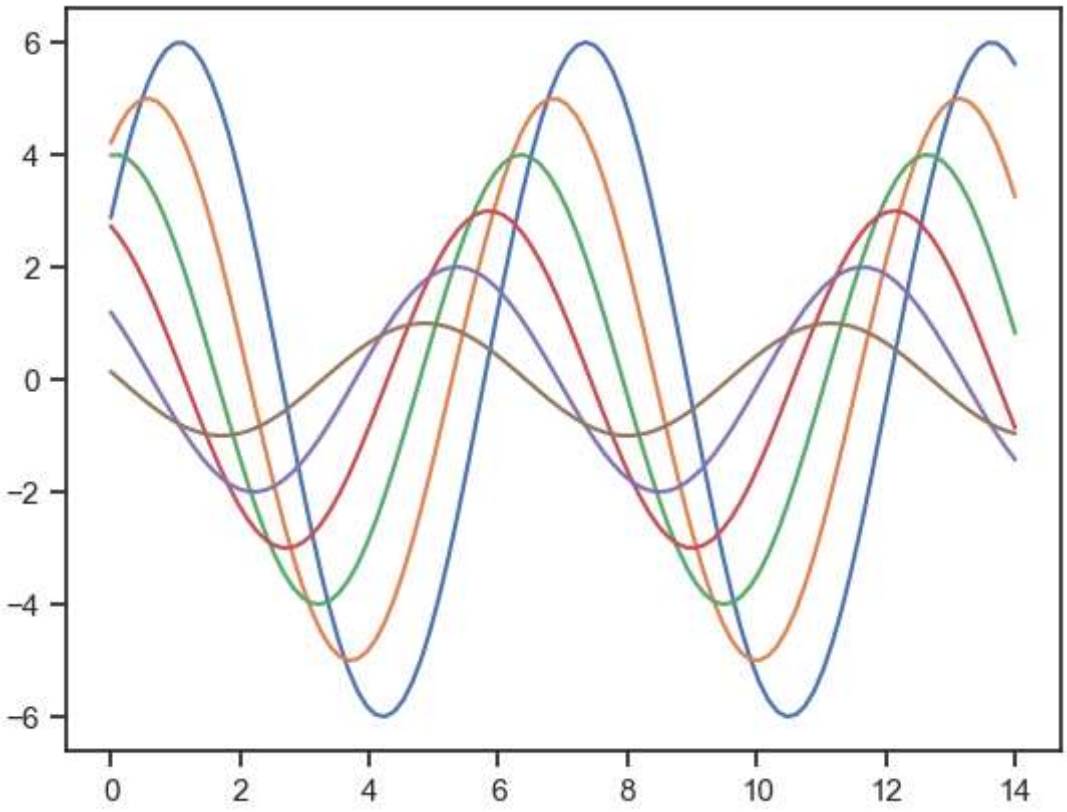


In [89]:

```
1 sns.set_style('white')
2 sinplot()
```



In [90]:
```
1 sns.set_style('ticks')
2 sinplot()
```



In [ ]:
```
1
```