In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style='whitegrid')
import os
import warnings

warnings.filterwarnings('ignore')
```

extracting data set fifa

In [2]:

```
df=pd.read_csv(r'C:\Users\nlnar\Downloads\25th, 26th- Advanced EDA project\25th, 26
```

In [3]:

```
df
```

Out[3]:

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

18207 rows × 89 columns

In [4]:

```
df.columns
```

Out[4]:

```
Index(['Unnamed: 0', 'ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
       'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Special',
       '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', 'Dribbling',
       'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
       'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
       'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
       'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
       'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
       'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause'],
      dtype='object')
```

In [5]:
```
len(df.columns)
```

Out[5]:
```
89
```

In [6]:
```
df.info()                           # summary of the data set
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18207 entries, 0 to 18206
Data columns (total 89 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Unnamed: 0                18207 non-null  int64
 1   ID                        18207 non-null  int64
 2   Name                      18207 non-null  object
 3   Age                       18207 non-null  int64
 4   Photo                     18207 non-null  object
 5   Nationality               18207 non-null  object
 6   Flag                      18207 non-null  object
 7   Overall                   18207 non-null  int64
 8   Potential                 18207 non-null  int64
 9   Club                      17966 non-null  object
 10  Club Logo                 18207 non-null  object
 11  Value                     18207 non-null  object
 12  Wage                      18207 non-null  object
 13  Special                   18207 non-null  int64
 14  Preferred Foot            18159 non-null  object
 15  International Reputation  18159 non-null  float64
 16  Weak Foot                 18159 non-null  float64
 17  Skill Moves               18159 non-null  float64
 18  Work Rate                 18159 non-null  object
 19  Body Type                 18159 non-null  object
 20  Real Face                 18159 non-null  object
 21  Position                  18147 non-null  object
 22  Jersey Number             18147 non-null  float64
 23  Joined                    16654 non-null  object
 24  Loaned From               1264 non-null   object
 25  Contract Valid Until      17918 non-null  object
 26  Height                    18159 non-null  object
 27  Weight                    18159 non-null  object
 28  LS                        16122 non-null  object
 29  ST                        16122 non-null  object
 30  RS                        16122 non-null  object
 31  LW                        16122 non-null  object
 32  LF                        16122 non-null  object
 33  CF                        16122 non-null  object
 34  RF                        16122 non-null  object
 35  RW                        16122 non-null  object
 36  LAM                       16122 non-null  object
 37  CAM                       16122 non-null  object
 38  RAM                       16122 non-null  object
 39  LM                        16122 non-null  object
 40  LCM                       16122 non-null  object
 41  CM                        16122 non-null  object
 42  RCM                       16122 non-null  object
 43  RM                        16122 non-null  object
 44  LWB                       16122 non-null  object
 45  LDM                       16122 non-null  object
 46  CDM                       16122 non-null  object
 47  RDM                       16122 non-null  object
 48  RWB                       16122 non-null  object
 49  LB                        16122 non-null  object
 50  LCB                       16122 non-null  object
 51  CB                        16122 non-null  object
 52  RCB                       16122 non-null  object
 53  RB                        16122 non-null  object
 54  Crossing                  18159 non-null  float64
 55  Finishing                 18159 non-null  float64
 56  HeadingAccuracy           18159 non-null  float64
 57  ShortPassing              18159 non-null  float64
 58  Volleys                   18159 non-null  float64
 59  Dribbling                 18159 non-null  float64
 60  Curve                     18159 non-null  float64
 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
 81  StandingTackle            18159 non-null  float64
 82  SlidingTackle             18159 non-null  float64
 83  GKDiving                  18159 non-null  float64
 84  GKHandling                18159 non-null  float64
 85  GKKicking                 18159 non-null  float64
 86  GKPositioning             18159 non-null  float64
 87  GKReflexes                18159 non-null  float64
 88  Release Clause            16643 non-null  object
dtypes: float64(38), int64(6), object(45)
memory usage: 12.4+ MB
```

```
In [7]: df.describe()
```

Out[7]:

	Unnamed: 0	ID	Age	Overall	Potential	Speci
count	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.00000
mean	9103.000000	214298.338606	25.122206	66.238699	71.307299	1597.80990
std	5256.052511	29965.244204	4.669943	6.908930	6.136496	272.58601
min	0.000000	16.000000	16.000000	46.000000	48.000000	731.00000
25%	4551.500000	200315.500000	21.000000	62.000000	67.000000	1457.00000
50%	9103.000000	221759.000000	25.000000	66.000000	71.000000	1635.00000
75%	13654.500000	236529.500000	28.000000	71.000000	75.000000	1787.00000
max	18206.000000	246620.000000	45.000000	94.000000	95.000000	2346.00000

8 rows × 44 columns

In [8]:

```
df.describe().T
```

Out[8]:

	count	mean	std	min	25%	50%	75%
Unnamed: 0	18207.0	9103.000000	5256.052511	0.0	4551.5	9103.0	13654.5
ID	18207.0	214298.338606	29965.244204	16.0	200315.5	221759.0	236529.5
Age	18207.0	25.122206	4.669943	16.0	21.0	25.0	28.0
Overall	18207.0	66.238699	6.908930	46.0	62.0	66.0	71.0
Potential	18207.0	71.307299	6.136496	48.0	67.0	71.0	75.0
Special	18207.0	1597.809908	272.586016	731.0	1457.0	1635.0	1787.0
International Reputation	18159.0	1.113222	0.394031	1.0	1.0	1.0	1.0
Weak Foot	18159.0	2.947299	0.660456	1.0	3.0	3.0	3.0
Skill Moves	18159.0	2.361308	0.756164	1.0	2.0	2.0	3.0
Jersey Number	18147.0	19.546096	15.947765	1.0	8.0	17.0	26.0
Crossing	18159.0	49.734181	18.364524	5.0	38.0	54.0	64.0
Finishing	18159.0	45.550911	19.525820	2.0	30.0	49.0	62.0
HeadingAccuracy	18159.0	52.298144	17.379909	4.0	44.0	56.0	64.0
ShortPassing	18159.0	58.686712	14.699495	7.0	54.0	62.0	68.0
Volleys	18159.0	42.909026	17.694408	4.0	30.0	44.0	57.0
Dribbling	18159.0	55.371001	18.910371	4.0	49.0	61.0	68.0
Curve	18159.0	47.170824	18.395264	6.0	34.0	48.0	62.0
FKAccuracy	18159.0	42.863153	17.478763	3.0	31.0	41.0	57.0
LongPassing	18159.0	52.711933	15.327870	9.0	43.0	56.0	64.0
BallControl	18159.0	58.369459	16.686595	5.0	54.0	63.0	69.0
Acceleration	18159.0	64.614076	14.927780	12.0	57.0	67.0	75.0
SprintSpeed	18159.0	64.726967	14.649953	12.0	57.0	67.0	75.0
Agility	18159.0	63.503607	14.766049	14.0	55.0	66.0	74.0
Reactions	18159.0	61.836610	9.010464	21.0	56.0	62.0	68.0
Balance	18159.0	63.966573	14.136166	16.0	56.0	66.0	74.0
ShotPower	18159.0	55.460047	17.237958	2.0	45.0	59.0	68.0
Jumping	18159.0	65.089432	11.820044	15.0	58.0	66.0	73.0
Stamina	18159.0	63.219946	15.894741	12.0	56.0	66.0	74.0
Strength	18159.0	65.311967	12.557000	17.0	58.0	67.0	74.0
LongShots	18159.0	47.109973	19.260524	3.0	33.0	51.0	62.0

	count	mean	std	min	25%	50%	75%
Aggression	18159.0	55.868991	17.367967	11.0	44.0	59.0	69.0
Interceptions	18159.0	46.698276	20.696909	3.0	26.0	52.0	64.0
Positioning	18159.0	49.958478	19.529036	2.0	38.0	55.0	64.0
Vision	18159.0	53.400903	14.146881	10.0	44.0	55.0	64.0
Penalties	18159.0	48.548598	15.704053	5.0	39.0	49.0	60.0
Composure	18159.0	58.648274	11.436133	3.0	51.0	60.0	67.0
Marking	18159.0	47.281623	19.904397	3.0	30.0	53.0	64.0
StandingTackle	18159.0	47.697836	21.664004	2.0	27.0	55.0	66.0
SlidingTackle	18159.0	45.661435	21.289135	3.0	24.0	52.0	64.0
GKDiving	18159.0	16.616223	17.695349	1.0	8.0	11.0	14.0
GKHandling	18159.0	16.391596	16.906900	1.0	8.0	11.0	14.0
GKKicking	18159.0	16.232061	16.502864	1.0	8.0	11.0	14.0
GKPositioning	18159.0	16.388898	17.034669	1.0	8.0	11.0	14.0
GKReflexes	18159.0	16.710887	17.955119	1.0	8.0	11.0	14.0

In [9]:

```
df.isnull()
```

Out[9]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag	Overall	Potential	Club
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
18202	False	False	False	False	False	False	False	False	False	False
18203	False	False	False	False	False	False	False	False	False	False
18204	False	False	False	False	False	False	False	False	False	False
18205	False	False	False	False	False	False	False	False	False	False
18206	False	False	False	False	False	False	False	False	False	False

18207 rows × 89 columns

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
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 [11]:

```
df.shape
```

Out[11]:

```
(18207, 89)
```

In [12]:

```
df.head()
```

Out[12]:

	Unnamed: 0	ID	Name	Age	Photo	Nationali
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentin
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

5 rows × 89 columns

In [13]:

```
df.tail()
```

Out[13]:

	Unnamed: 0	ID	Name	Age	Photo
18202	18202	238813	J. Lundstram	19	https://cdn.sofifa.org/players/4/19/238813.png
18203	18203	243165	N. Christoffersson	19	https://cdn.sofifa.org/players/4/19/243165.png
18204	18204	241638	B. Worman	16	https://cdn.sofifa.org/players/4/19/241638.png
18205	18205	246268	D. Walker-Rice	17	https://cdn.sofifa.org/players/4/19/246268.png
18206	18206	246269	G. Nugent	16	https://cdn.sofifa.org/players/4/19/246269.png

5 rows × 89 columns

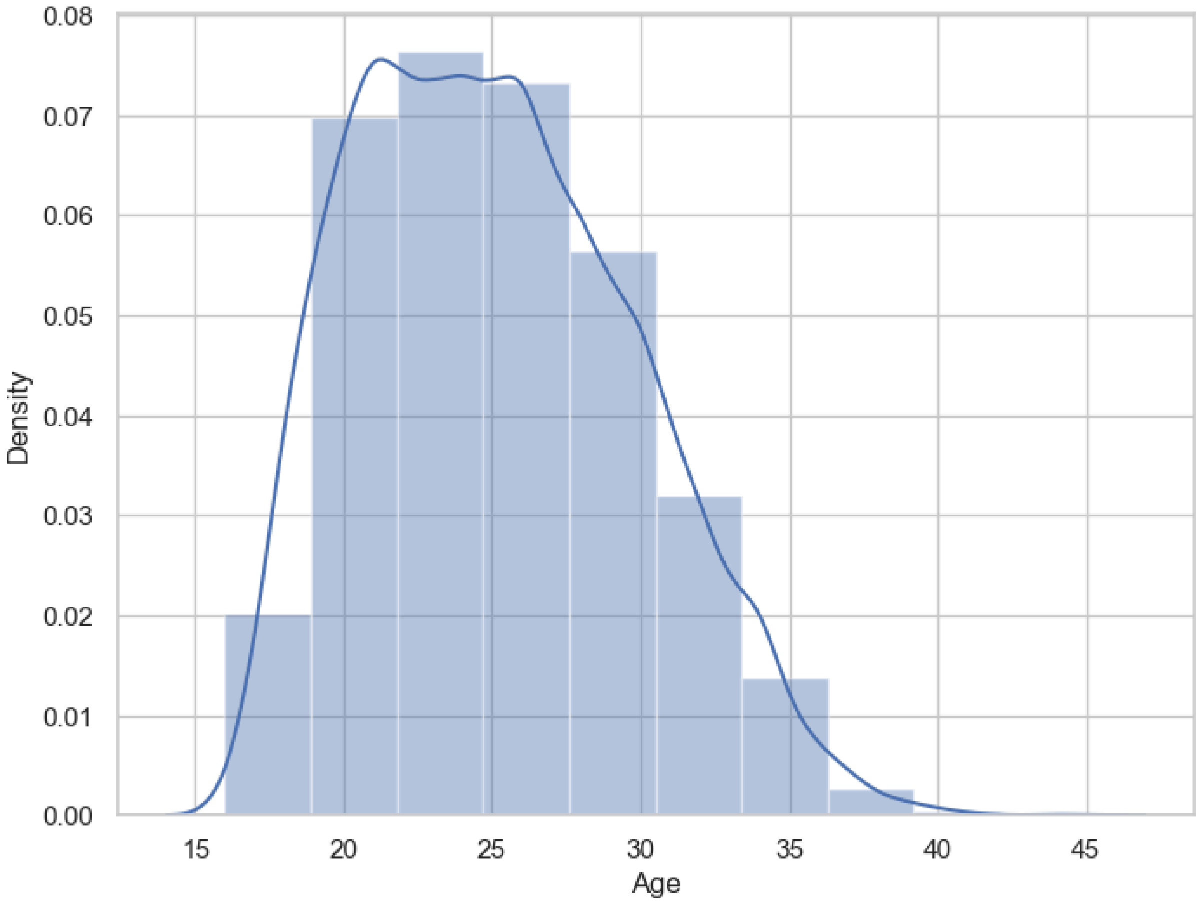
In [14]:

```
df['Body Type'].value_counts()
```

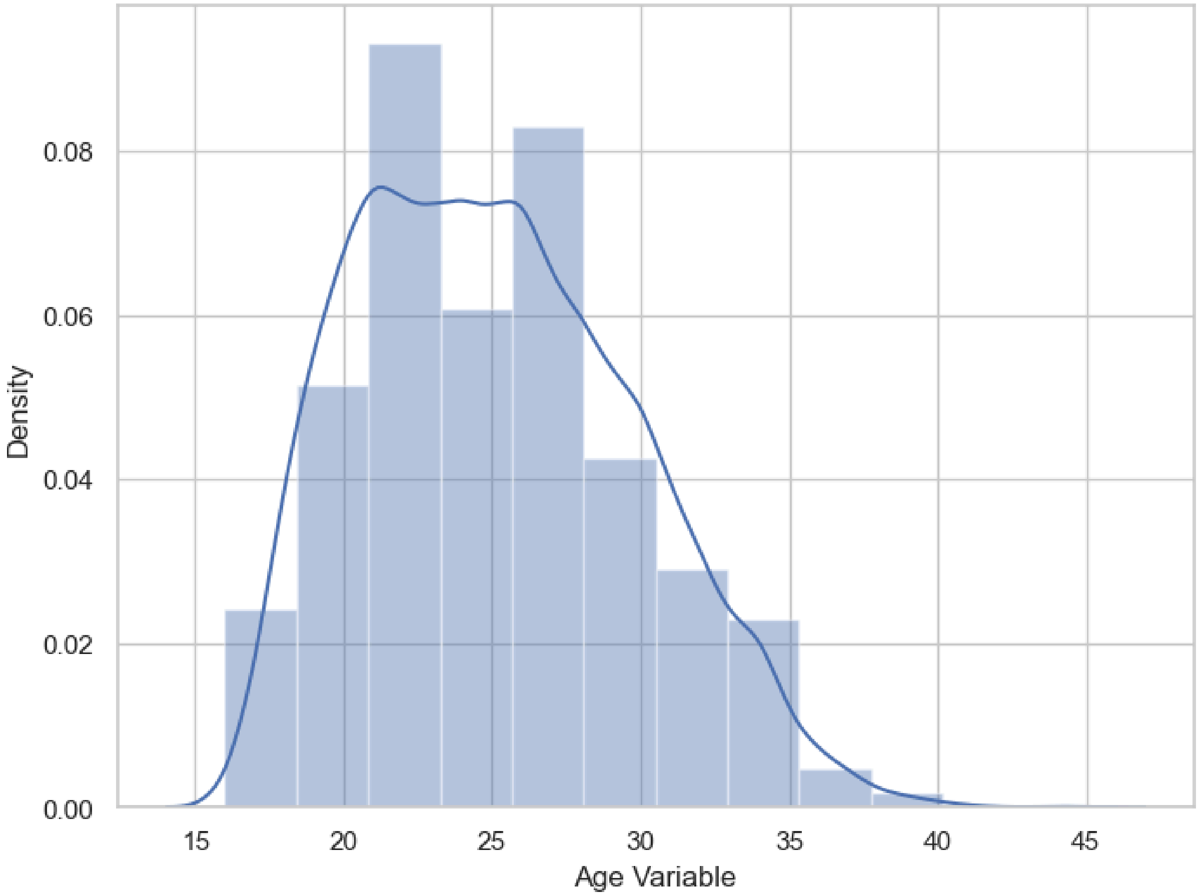
```
Out[14]: Body Type
         Normal                 10595
         Lean                    6417
         Stocky                  1140
         Messi                      1
         C. Ronaldo                 1
         Neymar                     1
         Courtois                   1
         PLAYER_BODY_TYPE_25        1
         Shaqiri                    1
         Akinfenwa                  1
         Name: count, dtype: int64
```

visualisation of age with seaborn distplot

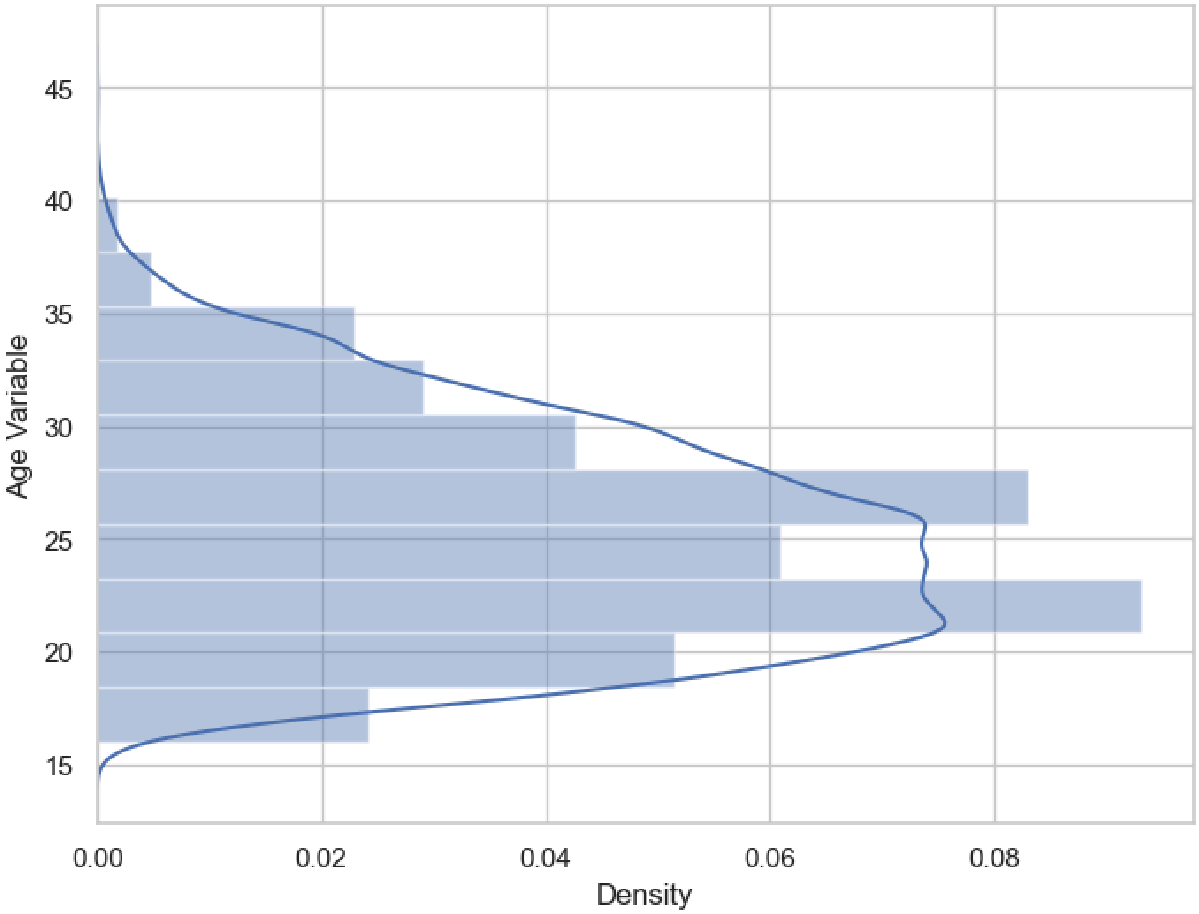
```
In [15]: f,ax=plt.subplots(figsize=(8,6))
         x=df['Age']
         ax=sns.distplot(x,bins=10)
         plt.show()
```



```
In [16]: f,ax=plt.subplots(figsize=(8,6))
         x=df['Age']
         x=pd.Series(x, name='Age Variable')
         ax=sns.distplot(x,bins=12)
         plt.show()
```

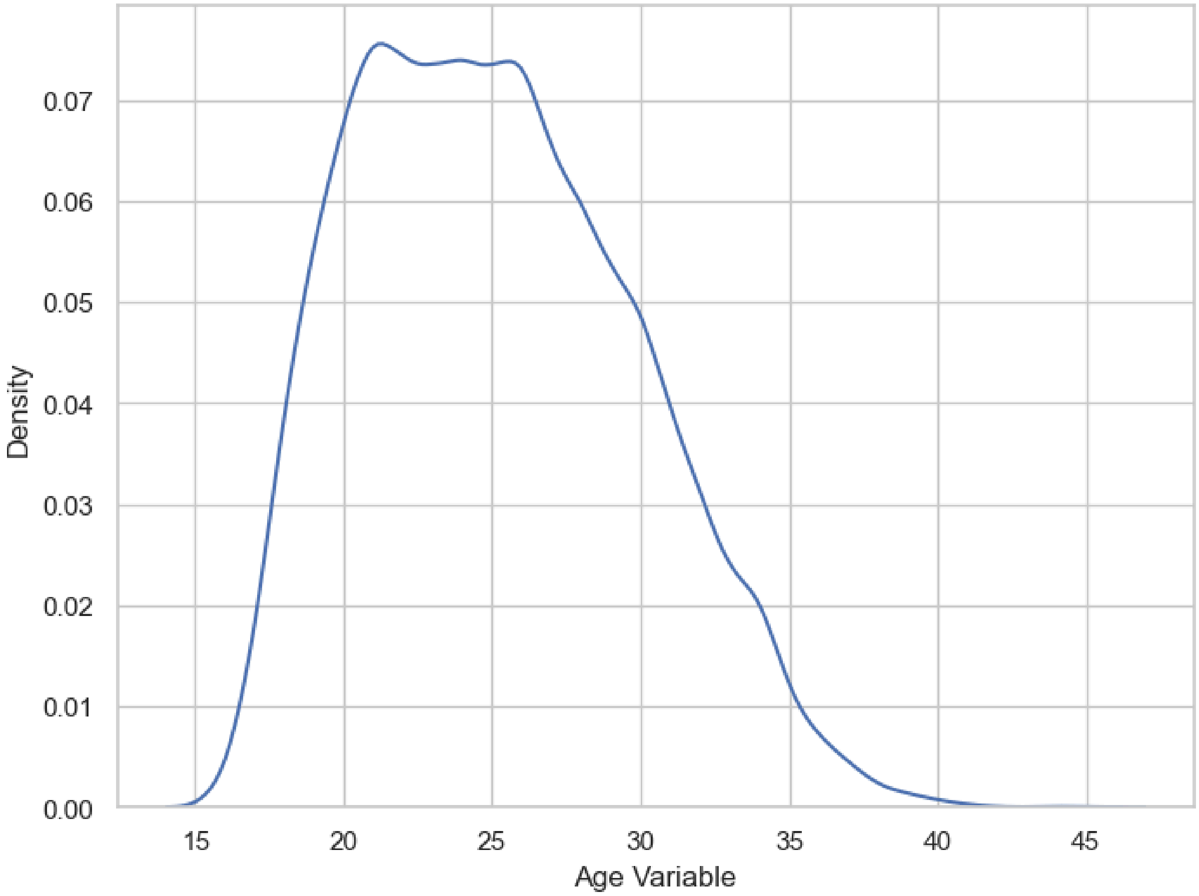


In [17]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['Age']
x=pd.Series(x, name='Age Variable')
ax=sns.distplot(x,bins=12,vertical=True)
plt.show()
```



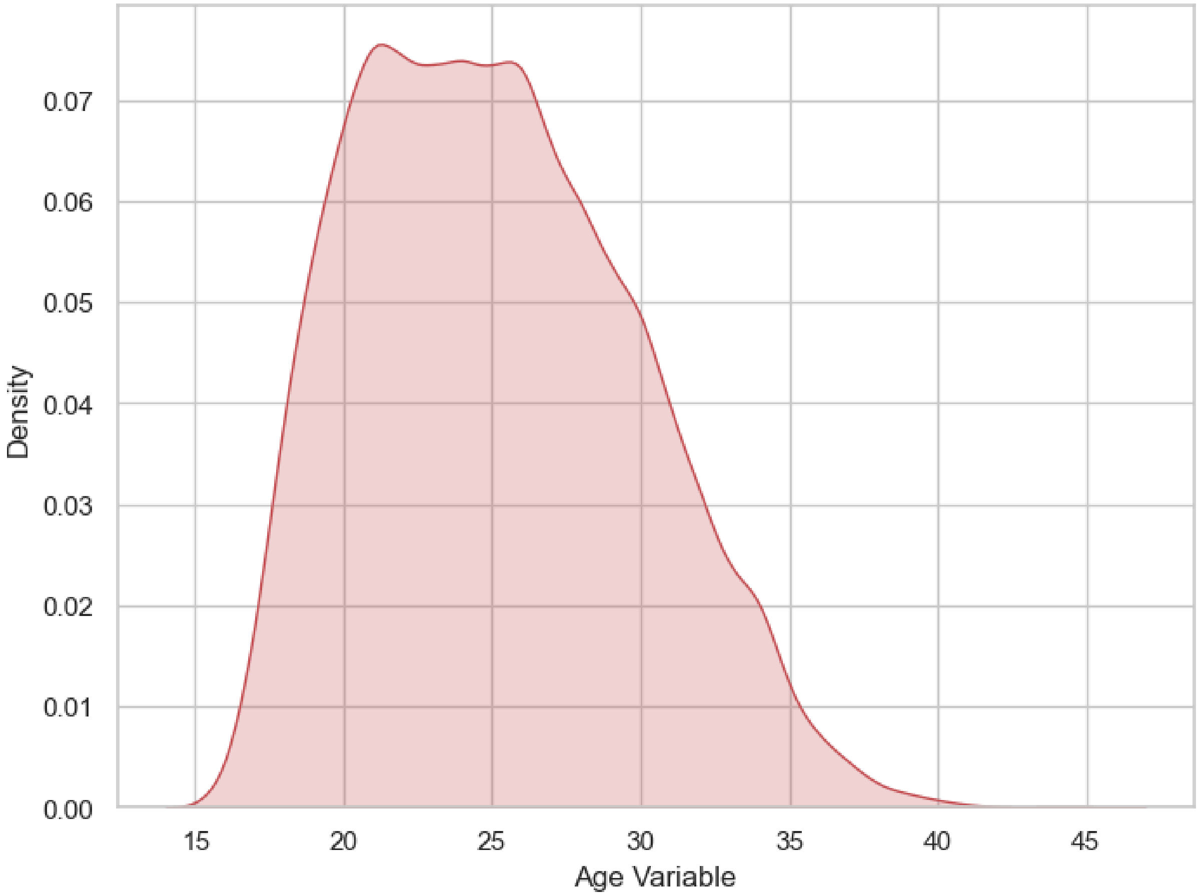
In [18]:

```
f,ax=plt.subplots(figsize=(8,6))
x=df['Age']
x=pd.Series(x, name='Age Variable')
ax=sns.kdeplot(x)                                # kde plot
plt.show()
```

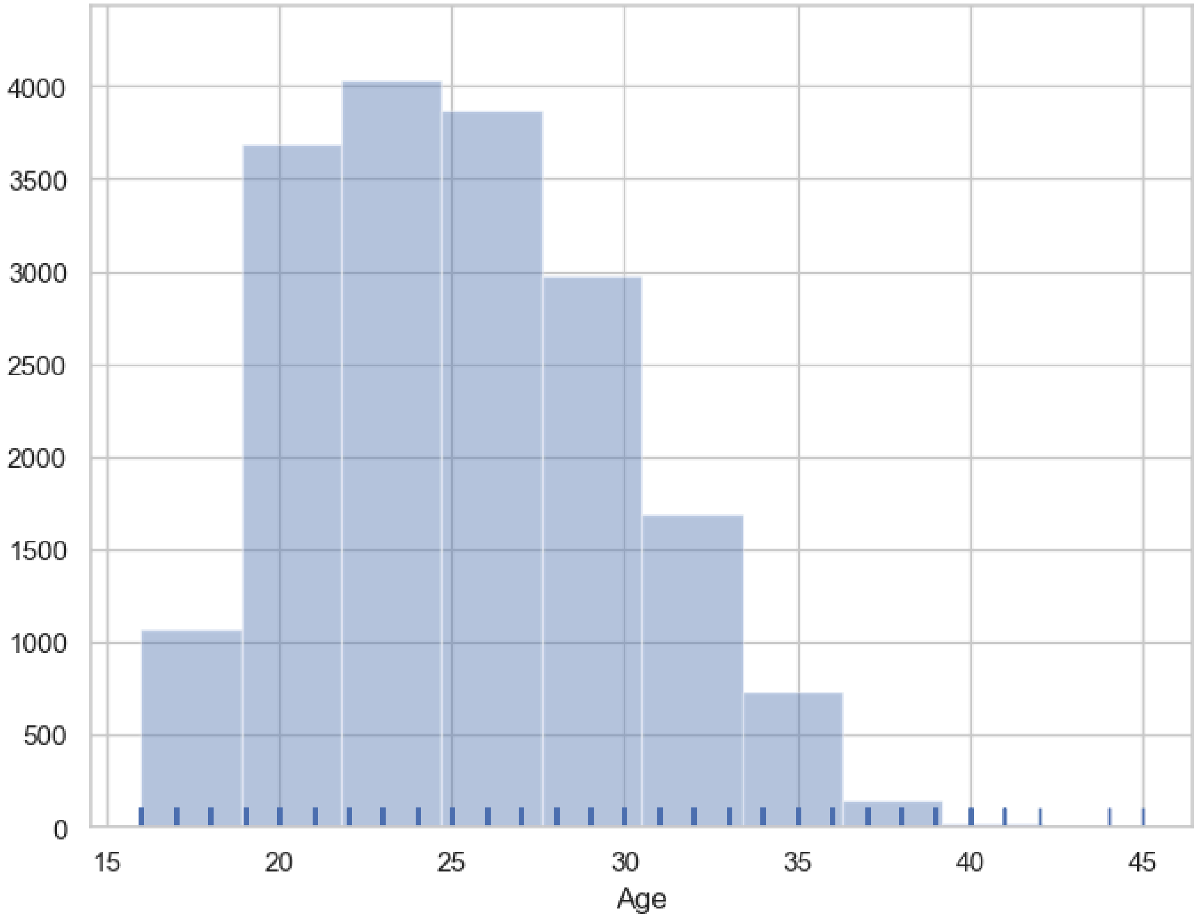


In [19]:

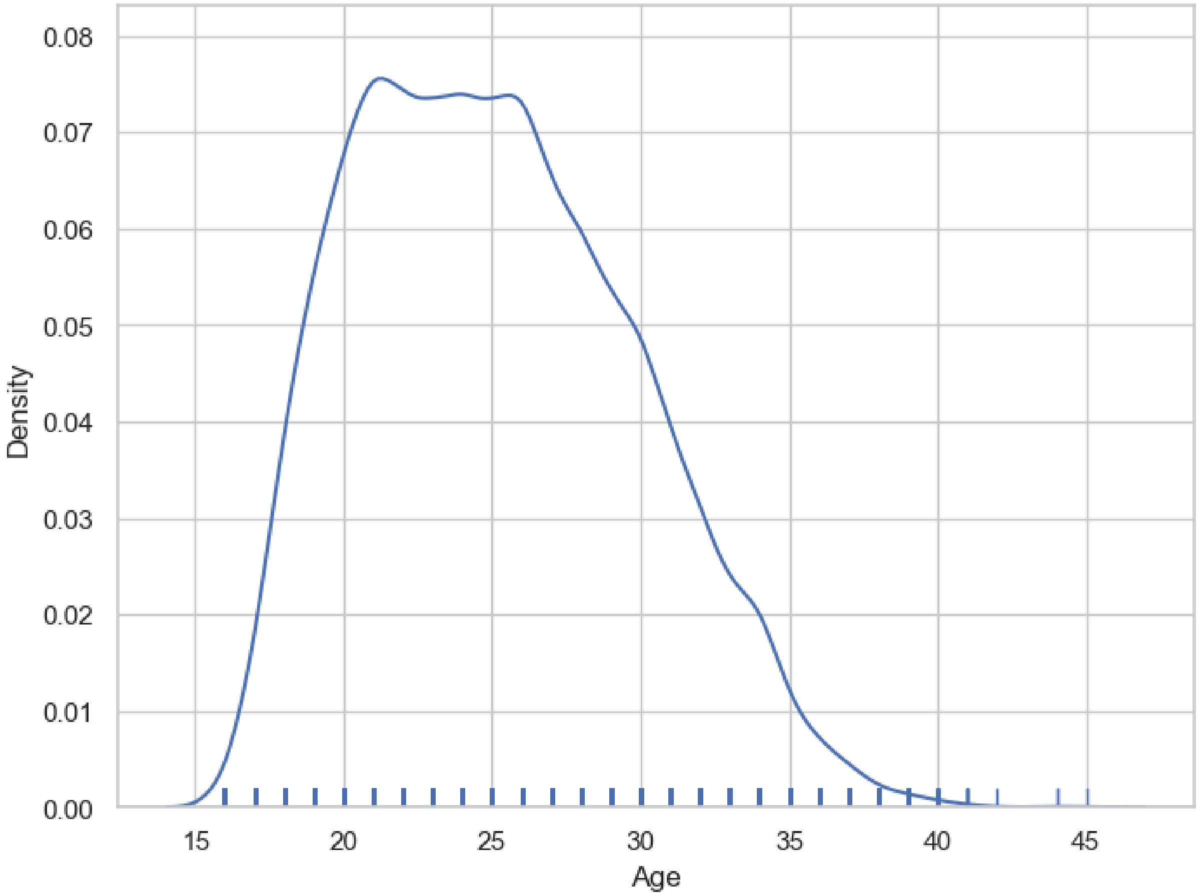
```
f,ax=plt.subplots(figsize=(8,6))
x=df['Age']
x=pd.Series(x, name='Age Variable')
ax=sns.kdeplot(x,shade=True,color='r')                          # kde plot od ag
plt.show()
```



```
In [20]: f,ax=plt.subplots(figsize=(8,6))
         x=df['Age']
         ax=sns.distplot(x,kde=False,rug=True,bins=10)                             # hist p
         plt.show()
```



```
In [21]: f,ax=plt.subplots(figsize=(8,6))
         x=df['Age']
         ax=sns.distplot(x,hist=False,rug=True,bins=10)                              # hist
         plt.show()
```



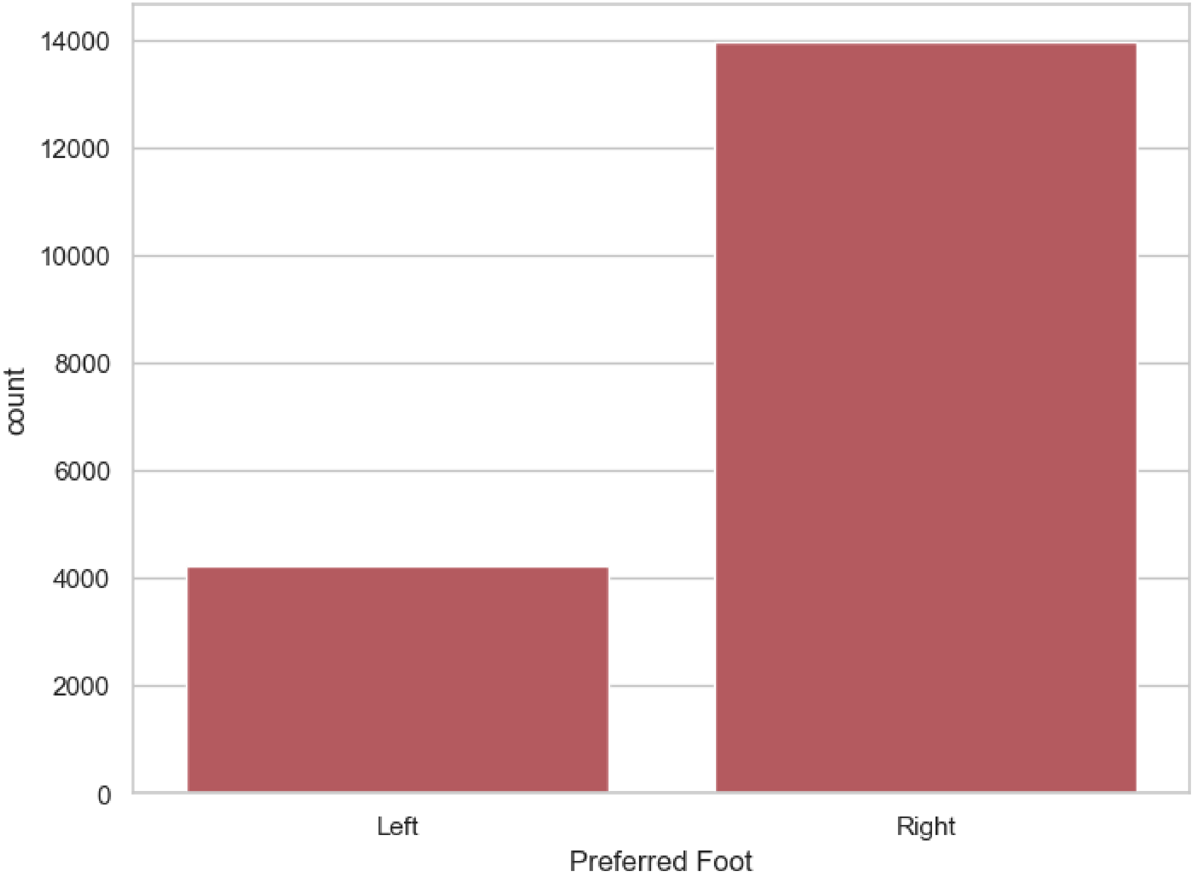
```
In [22]: df['Preferred Foot'].nunique()
```

```
Out[22]: 2
```

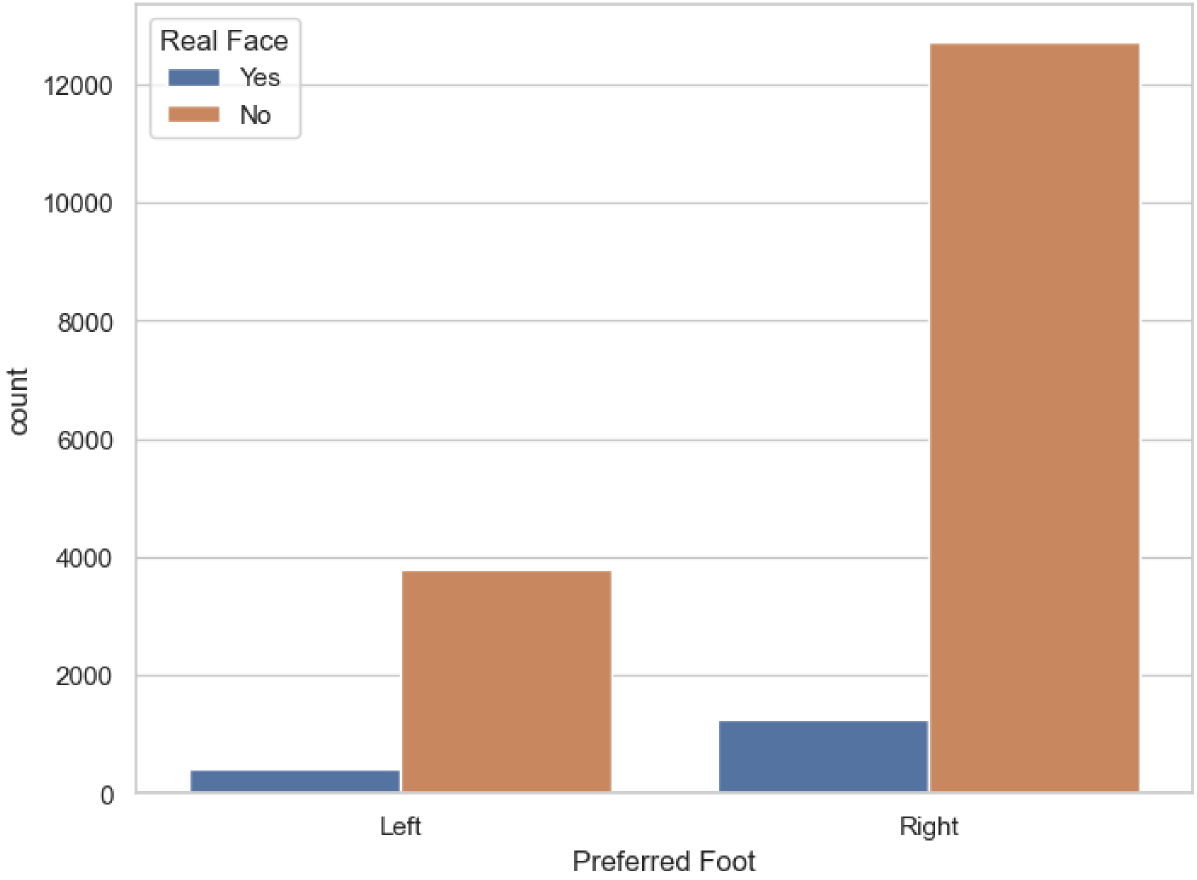
```
In [23]: df['Preferred Foot'].value_counts()
```

```
Out[23]: Preferred Foot
         Right    13948
         Left      4211
         Name: count, dtype: int64
```

```
In [26]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.countplot(x='Preferred Foot',data=df,color='r')
         plt.show()
```



```
In [27]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.countplot(x='Preferred Foot',data=df,hue='Real Face')
         plt.show()
```

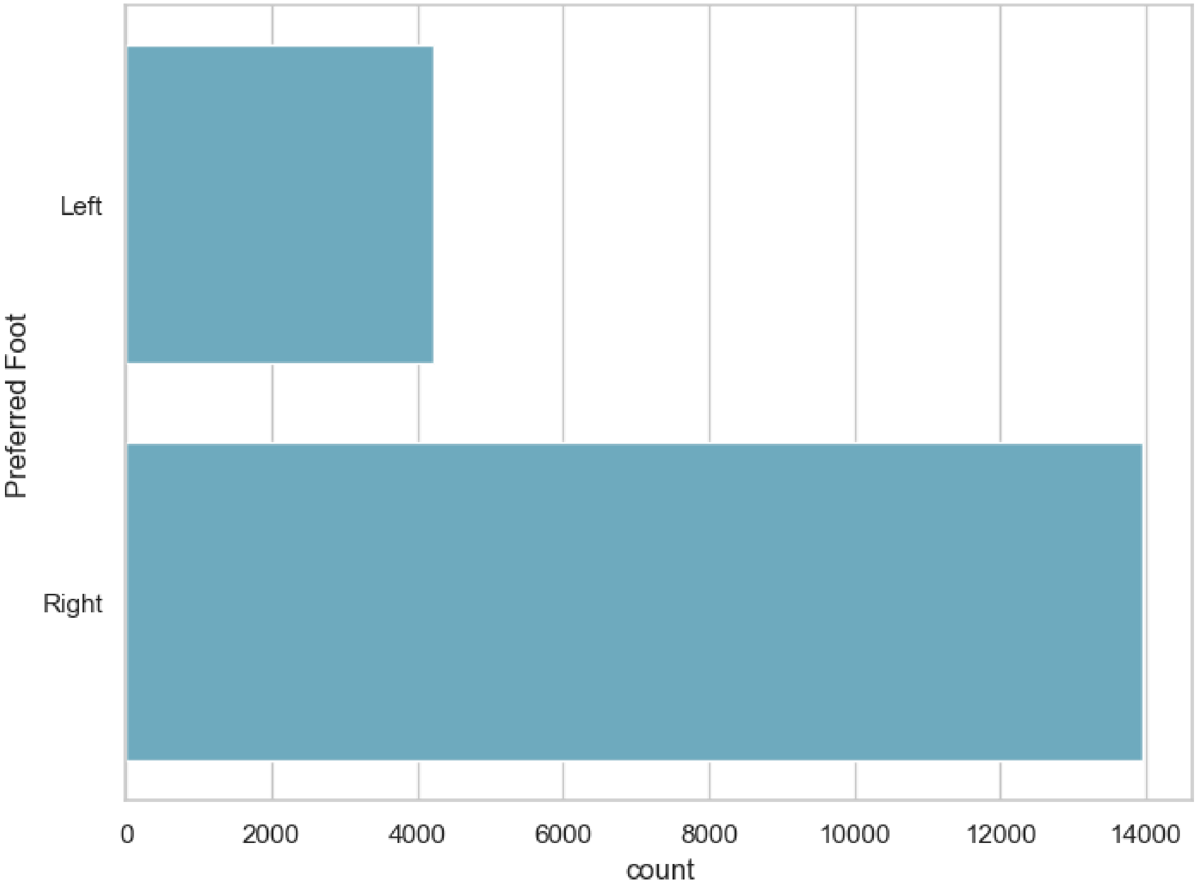


In [28]:
```
df.columns
```

Out[28]:
```
Index(['Unnamed: 0', 'ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
       'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Special',
       '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', 'Dribbling',
       'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
       'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
       'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
       'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
       'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
       'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause'],
      dtype='object')
```

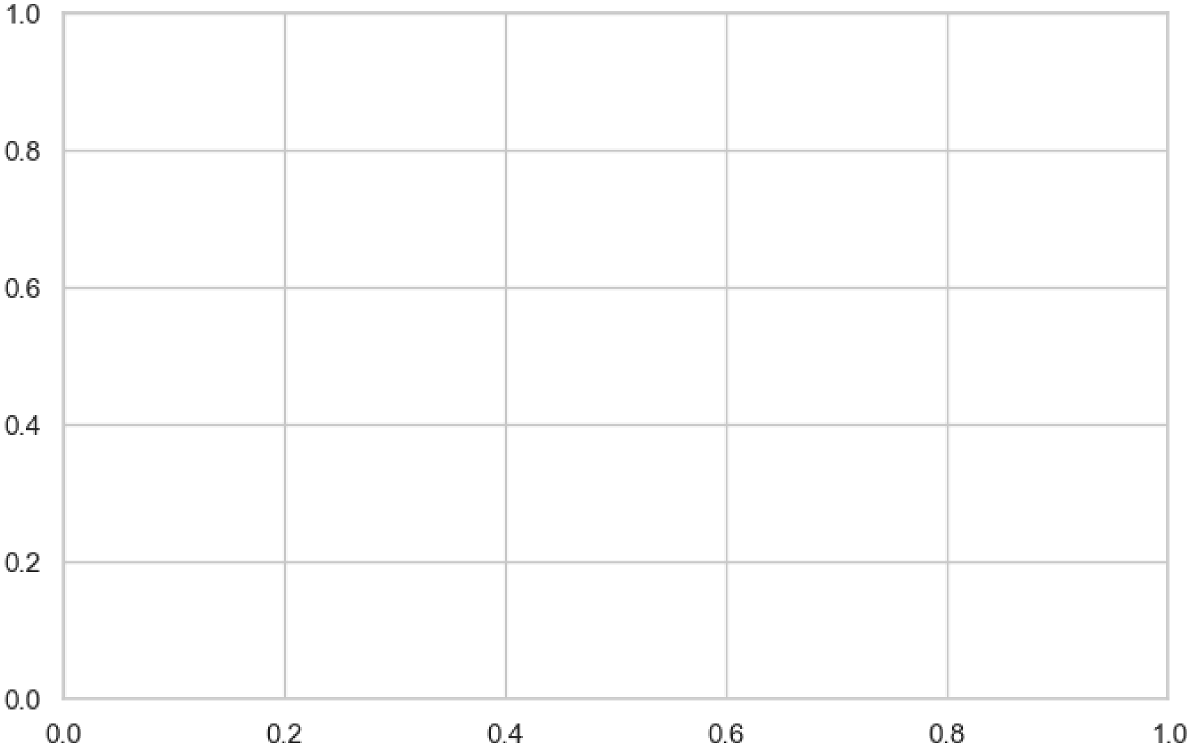
In [30]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(y='Preferred Foot',data=df,color='c')    # default it ts vertical
plt.show()
```

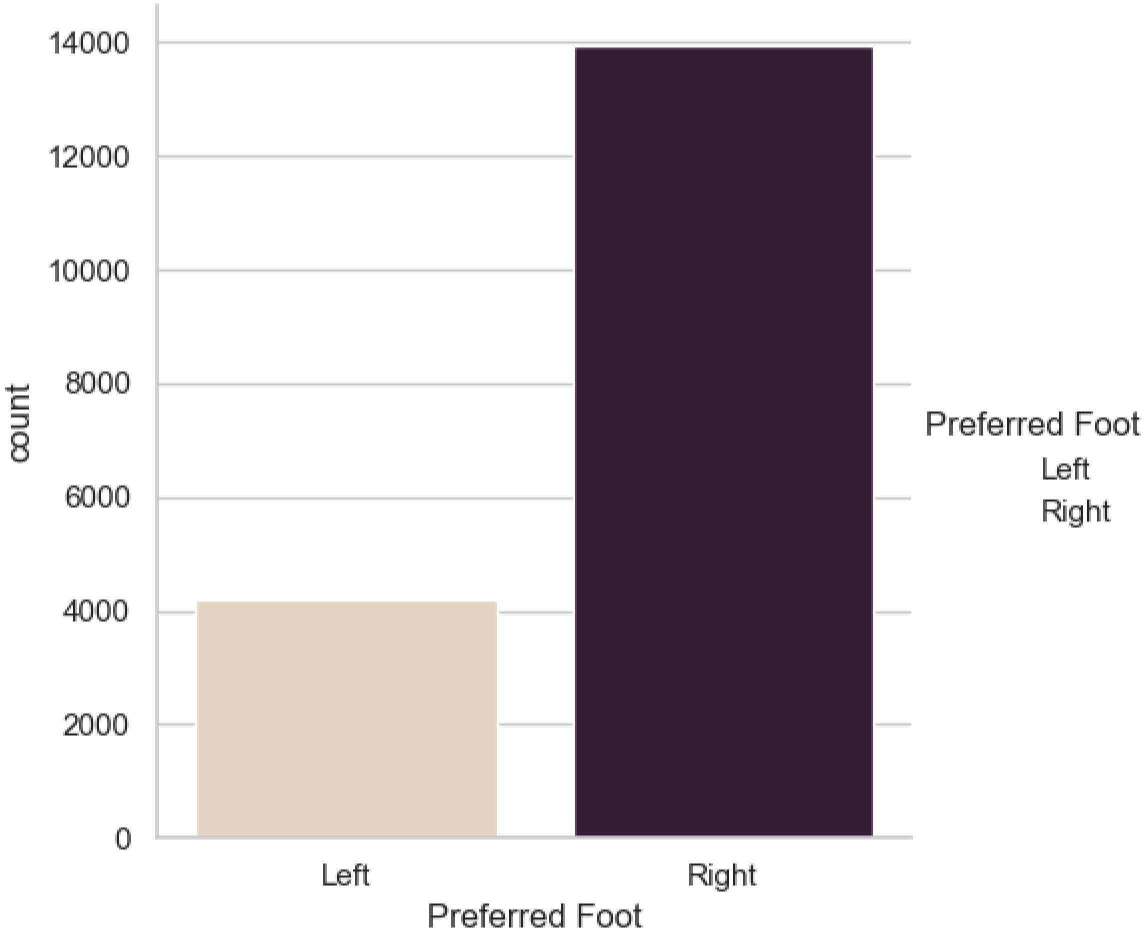
1.0
0.8
0.6
0.4
0.2
0.0
Left


Preferred Foot
Left
Right
0
2000
4000
6000
8000
10000
12000
14000
count

In [31]:

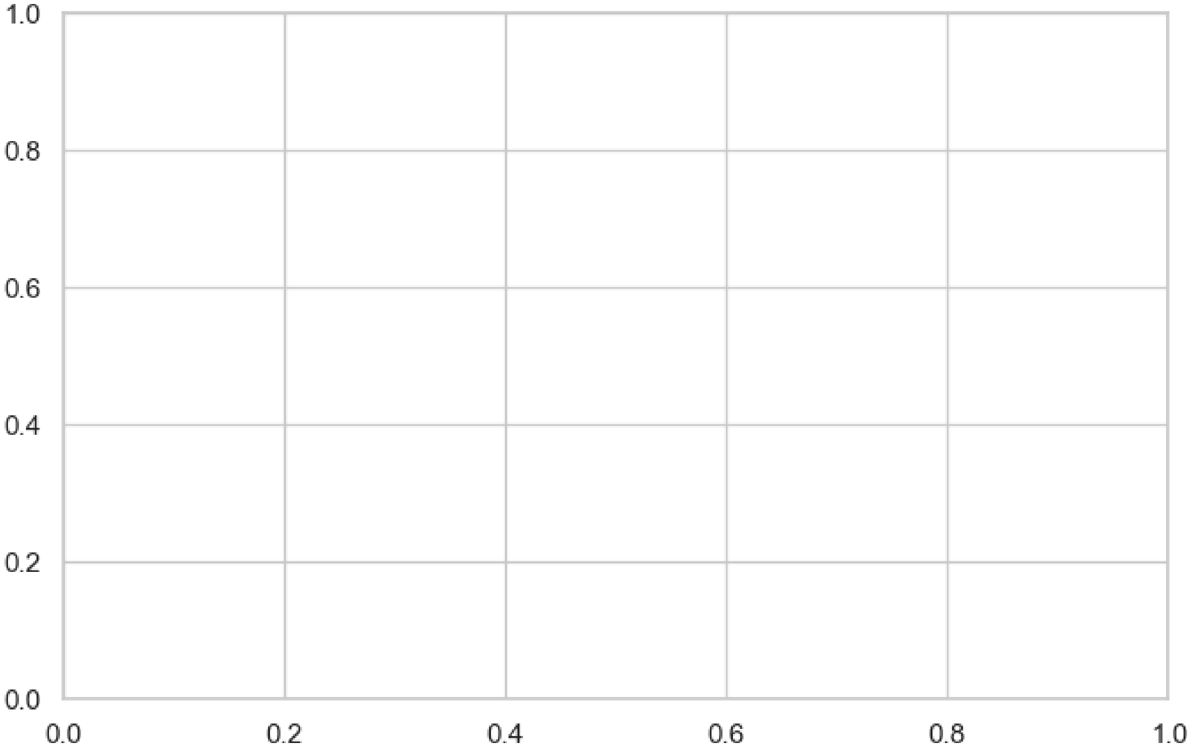
```
f.ax=plt.subplots(figsize=(8,5))
ax=sns.catplot(x='Preferred Foot',data=df,kind='count',palette='ch:.25')
plt.show()
```

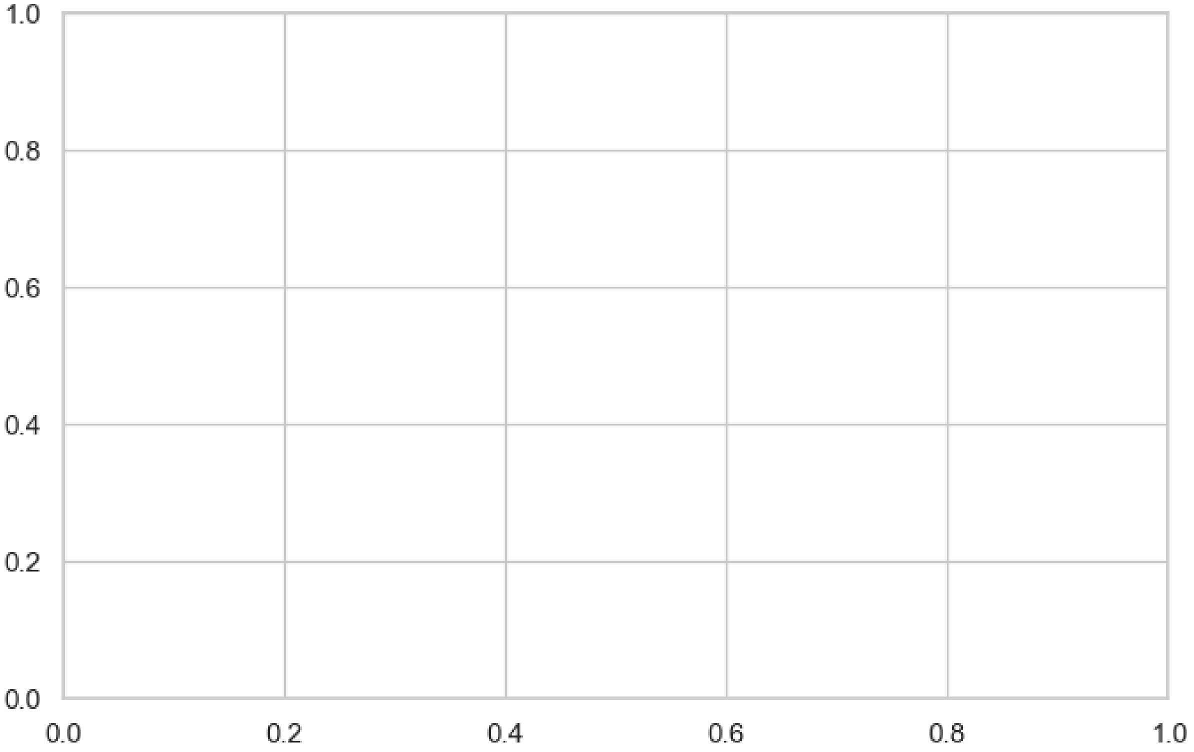


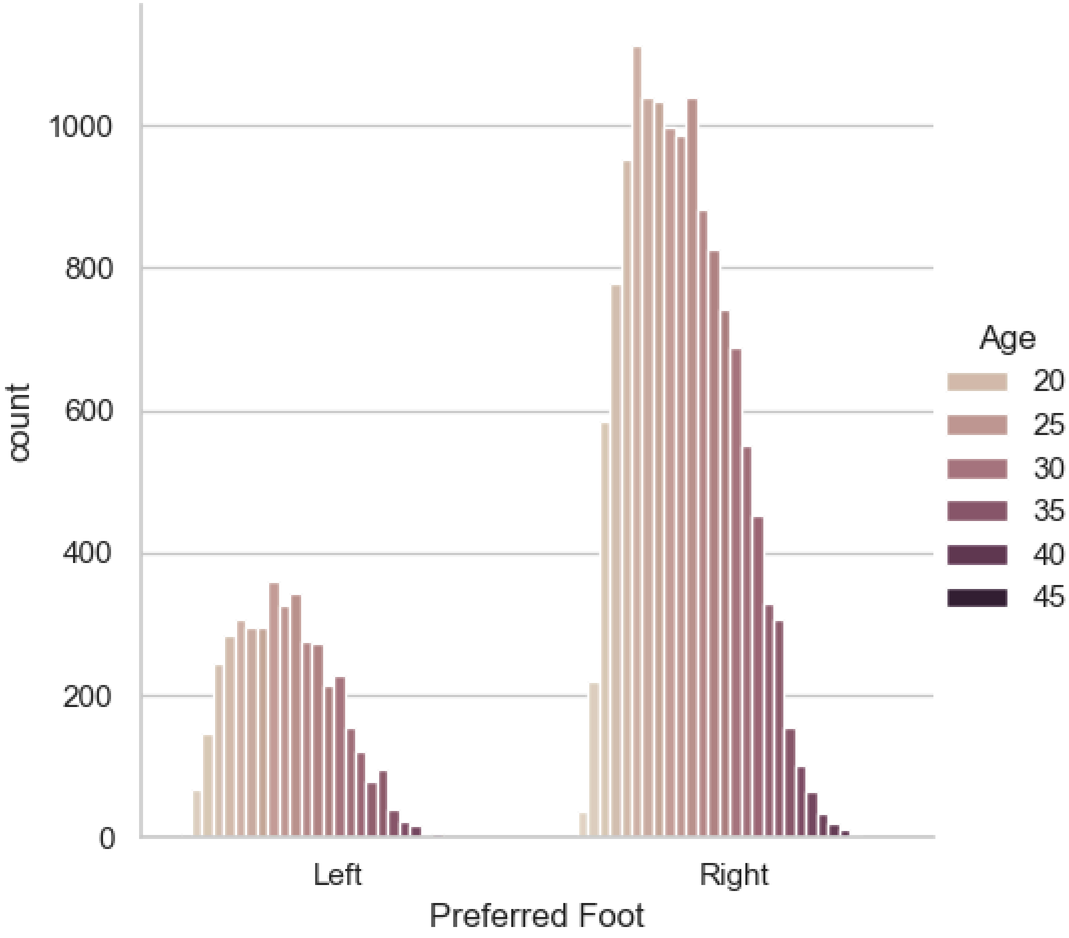


In [33]:

```
f.ax=plt.subplots(figsize=(8,5))
ax=sns.catplot(x='Preferred Foot',data=df,kind='count',palette='ch:.25',hue='Age')
plt.show()
```







INTERNATIONAL REPUTATION

```
In [34]: df['International Reputation'].nunique()
```

```
Out[34]: 5
```

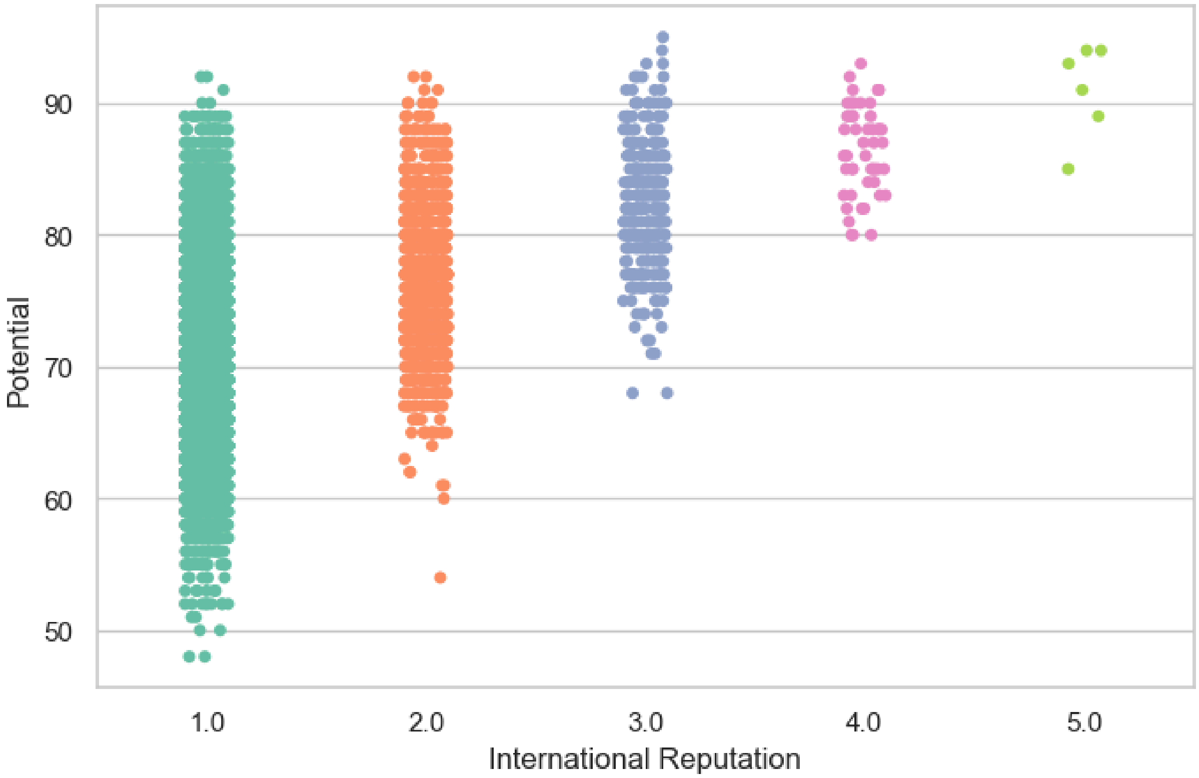
```
In [35]: df['International Reputation'].value_counts()
```

```
Out[35]: International Reputation
         1.0    16532
         2.0     1261
         3.0      309
         4.0       51
         5.0        6
         Name: count, dtype: int64
```

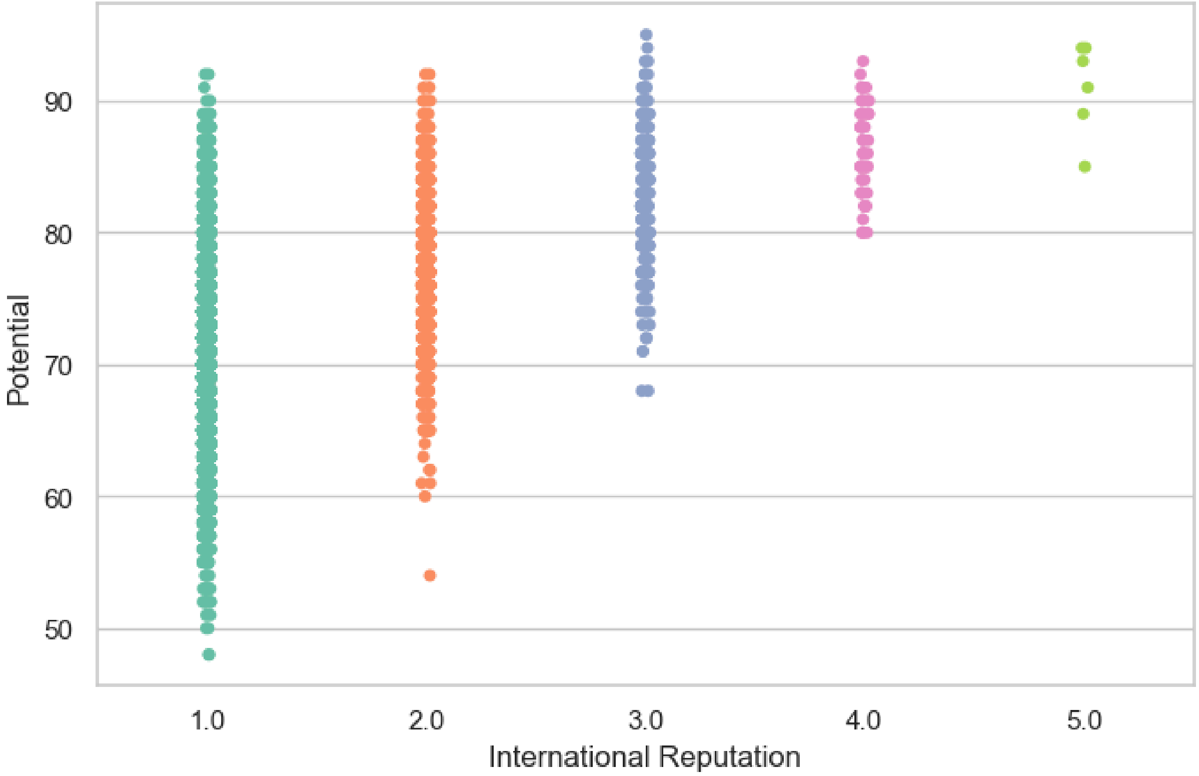
```
In [36]: df['International Reputation'].value_counts
```

```
Out[36]: <bound method IndexOpsMixin.value_counts of 0        5.0
         1        5.0
         2        5.0
         3        4.0
         4        4.0
                 ...
         18202    1.0
         18203    1.0
         18204    1.0
         18205    1.0
         18206    1.0
         Name: International Reputation, Length: 18207, dtype: float64>
```

```
In [43]: f,ax=plt.subplots(figsize=(8,5))
         ax=sns.stripplot(x='International Reputation',y='Potential',data=df,palette='Set2')
         plt.show()
```

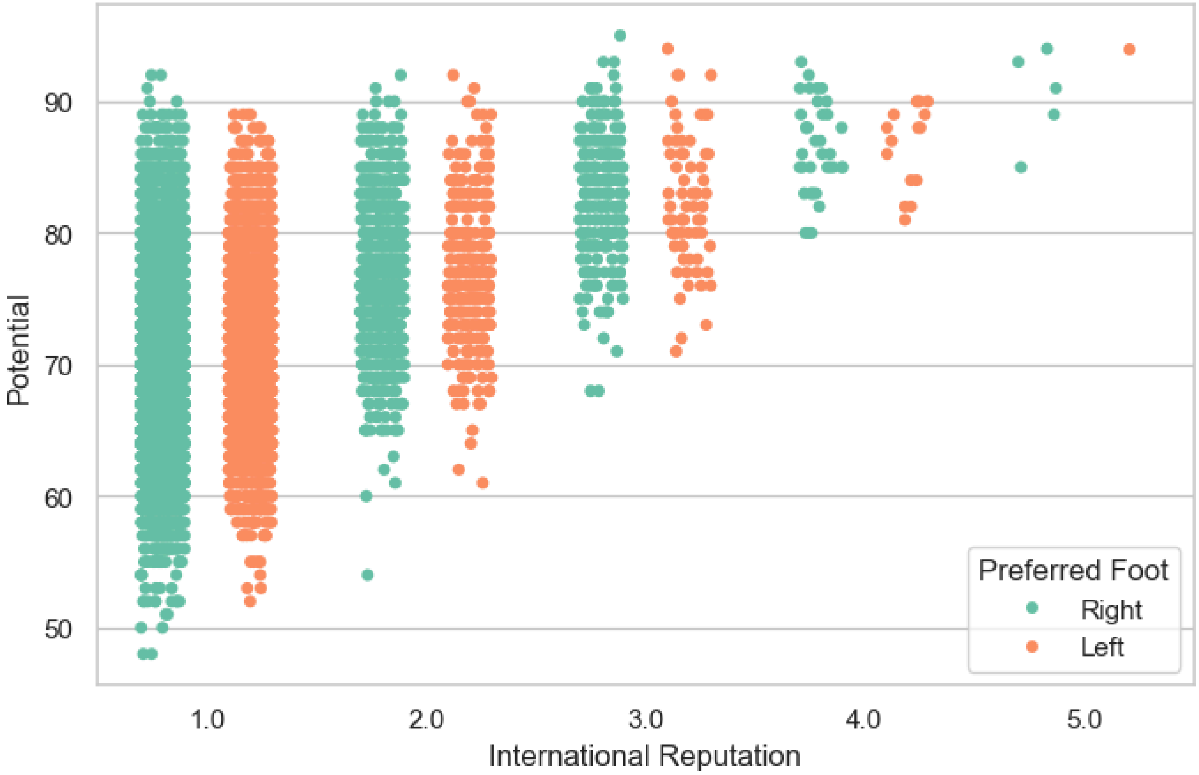


```
In [44]: f,ax=plt.subplots(figsize=(8,5))
         ax=sns.stripplot(x='International Reputation',y='Potential',data=df,palette='Set2',
         plt.show()
```



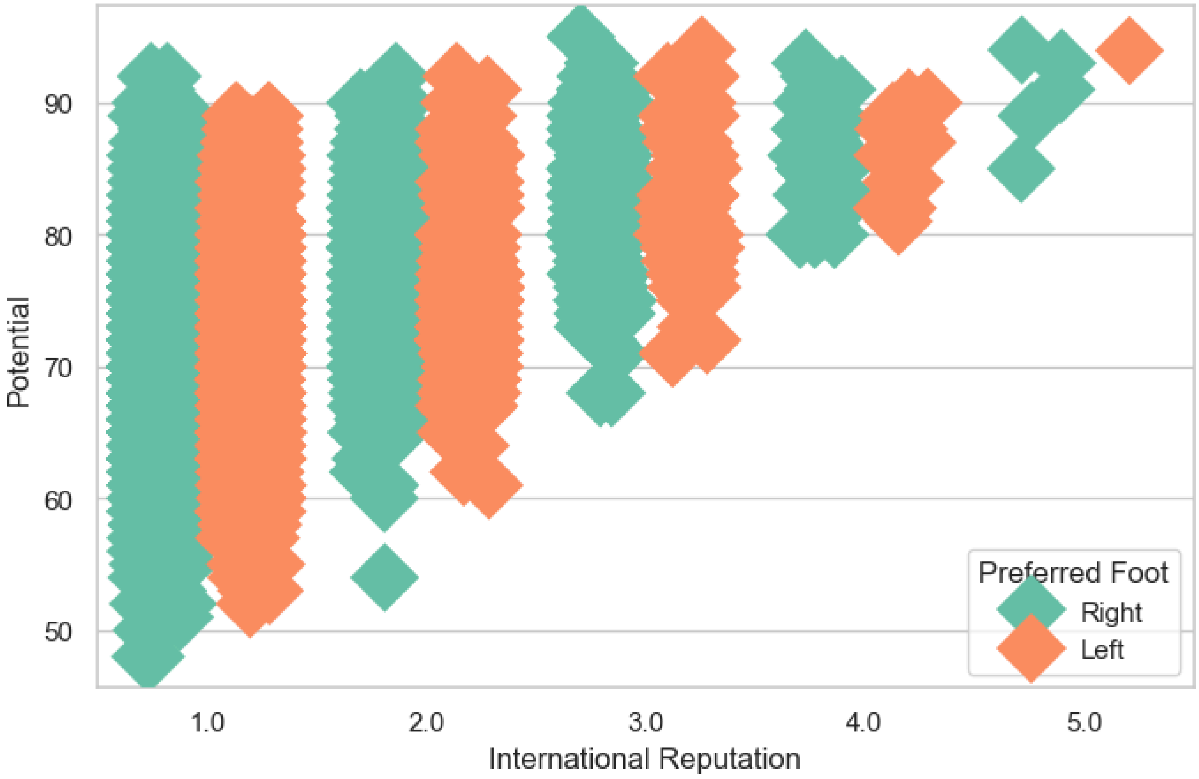
In [46]:

```
f,ax=plt.subplots(figsize=(8,5))
ax=sns.stripplot(x='International Reputation',y='Potential', hue='Preferred Foot',d
plt.show()
```



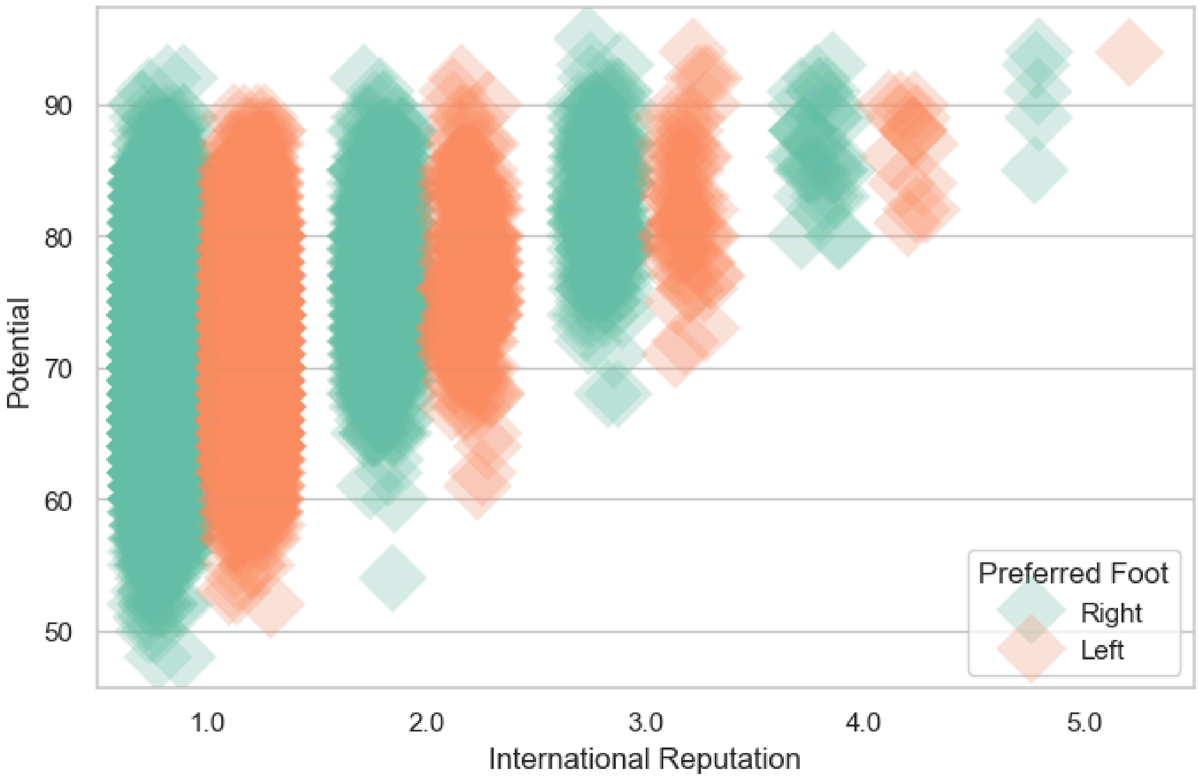
In [47]:

```
f,ax=plt.subplots(figsize=(8,5))
ax=sns.stripplot(x='International Reputation',y='Potential', hue='Preferred Foot',d
plt.show()
```



In [49]:

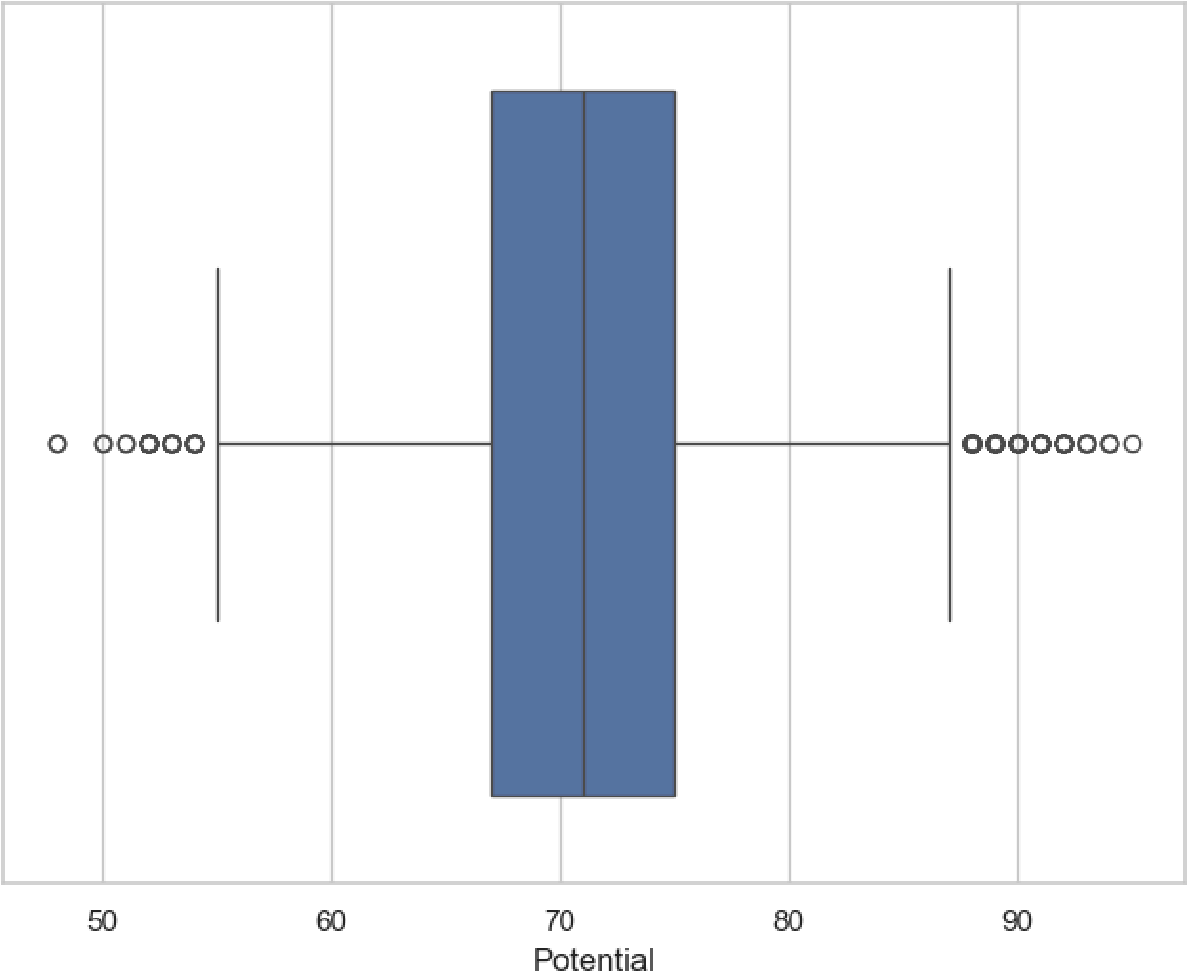
```
f,ax=plt.subplots(figsize=(8,5))
ax=sns.stripplot(x='International Reputation',y='Potential', hue='Preferred Foot',d
plt.show()
```



In [50]:

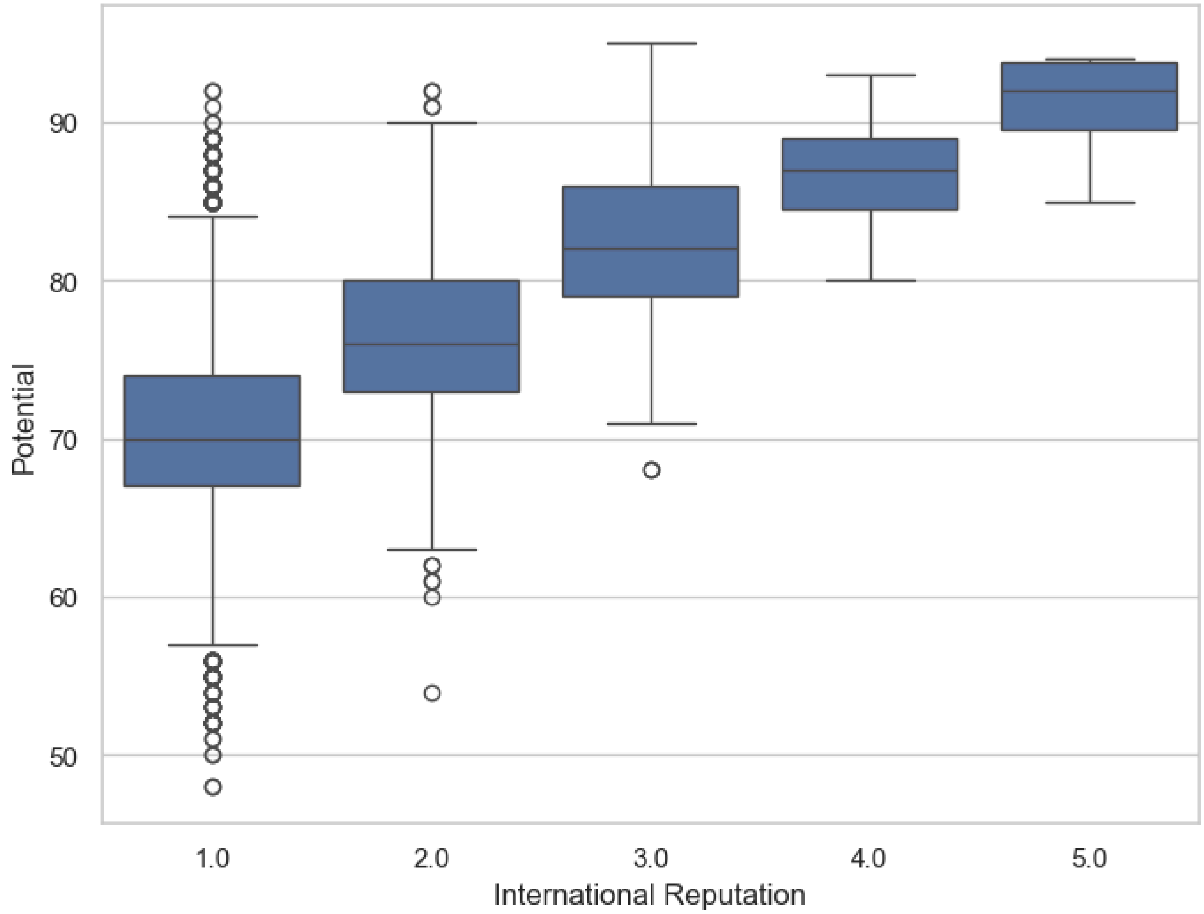
```
# BOX PLOT OF POTENTIAL VAR
f,ax=plt.subplots(figsize=(8,6))
```

```
ax=sns.boxplot(x=df['Potential'])
plt.show()
```



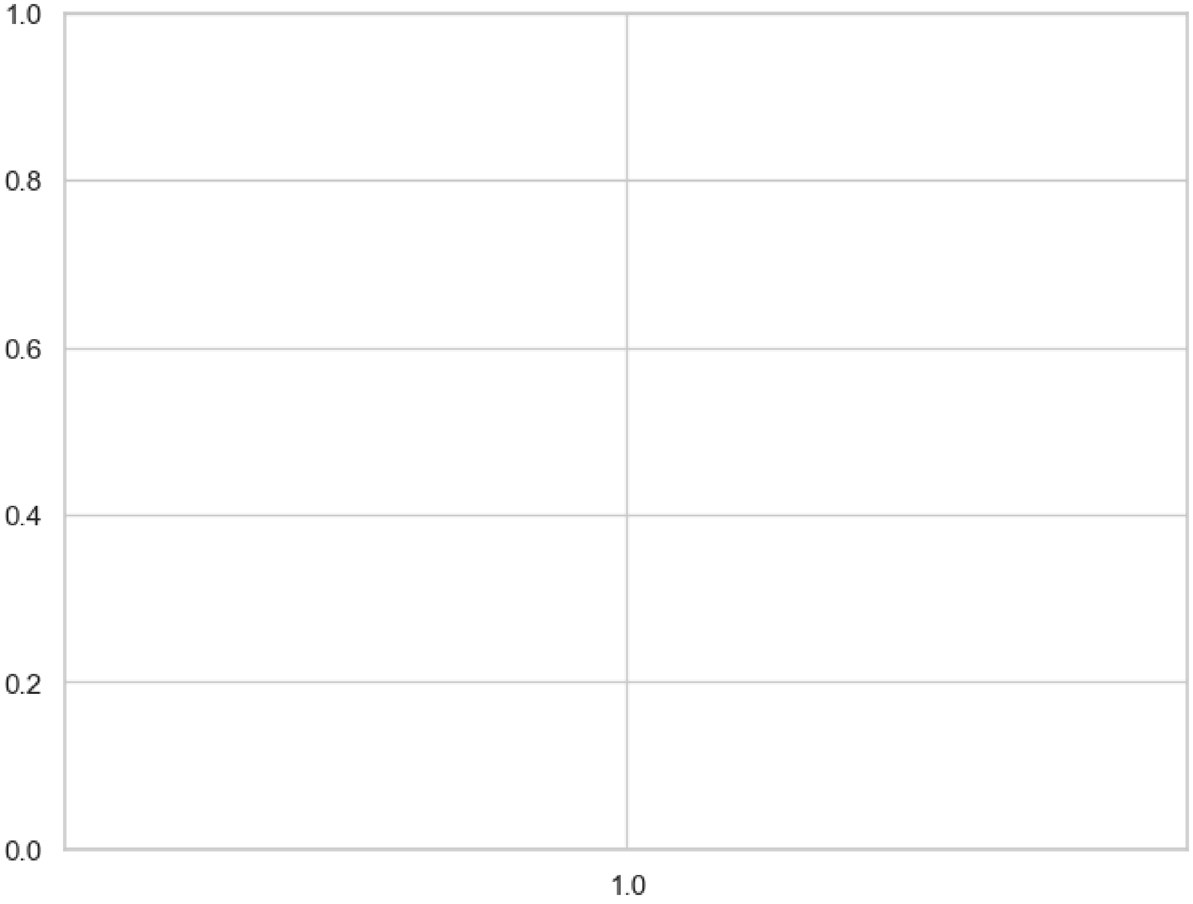
In [51]:

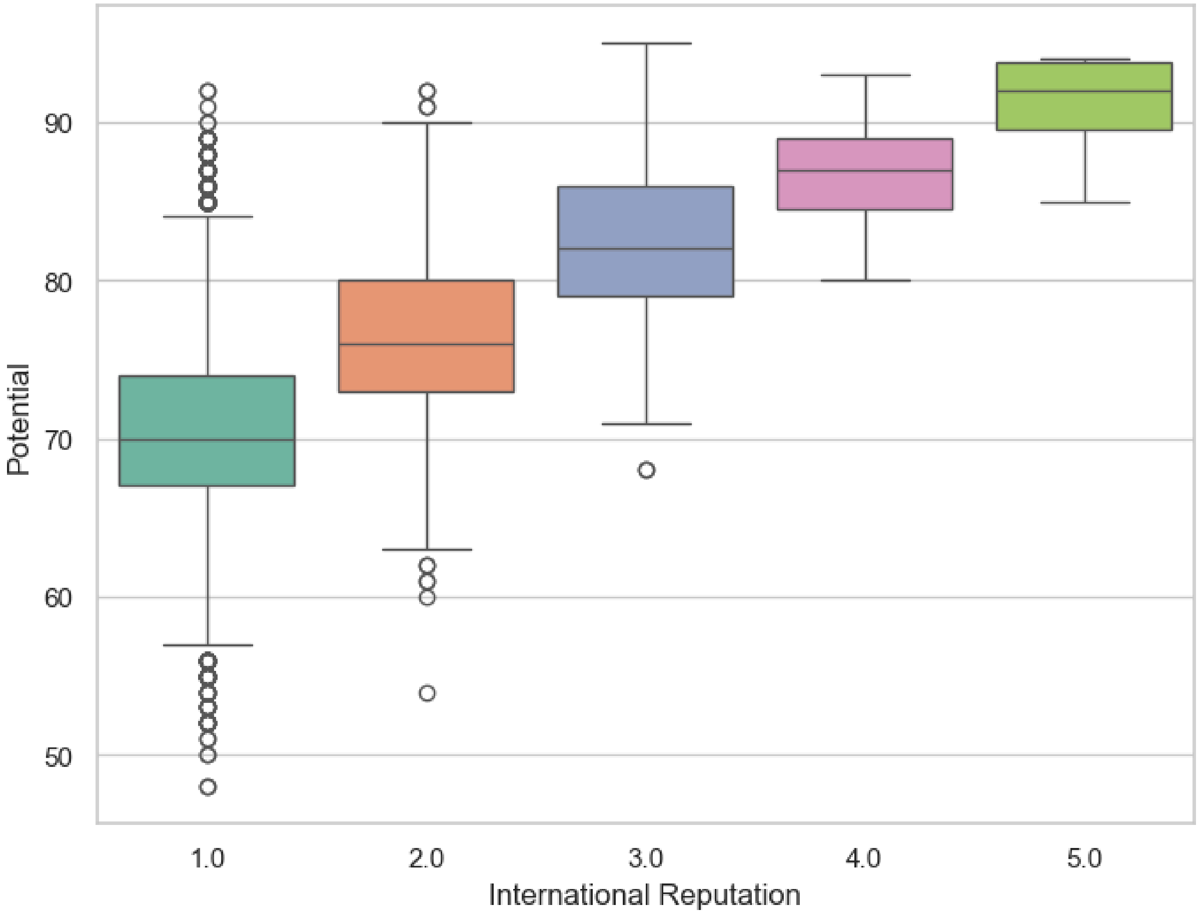
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.boxplot(x='International Reputation',y='Potential',data=df)
plt.show()
```



In [53]:

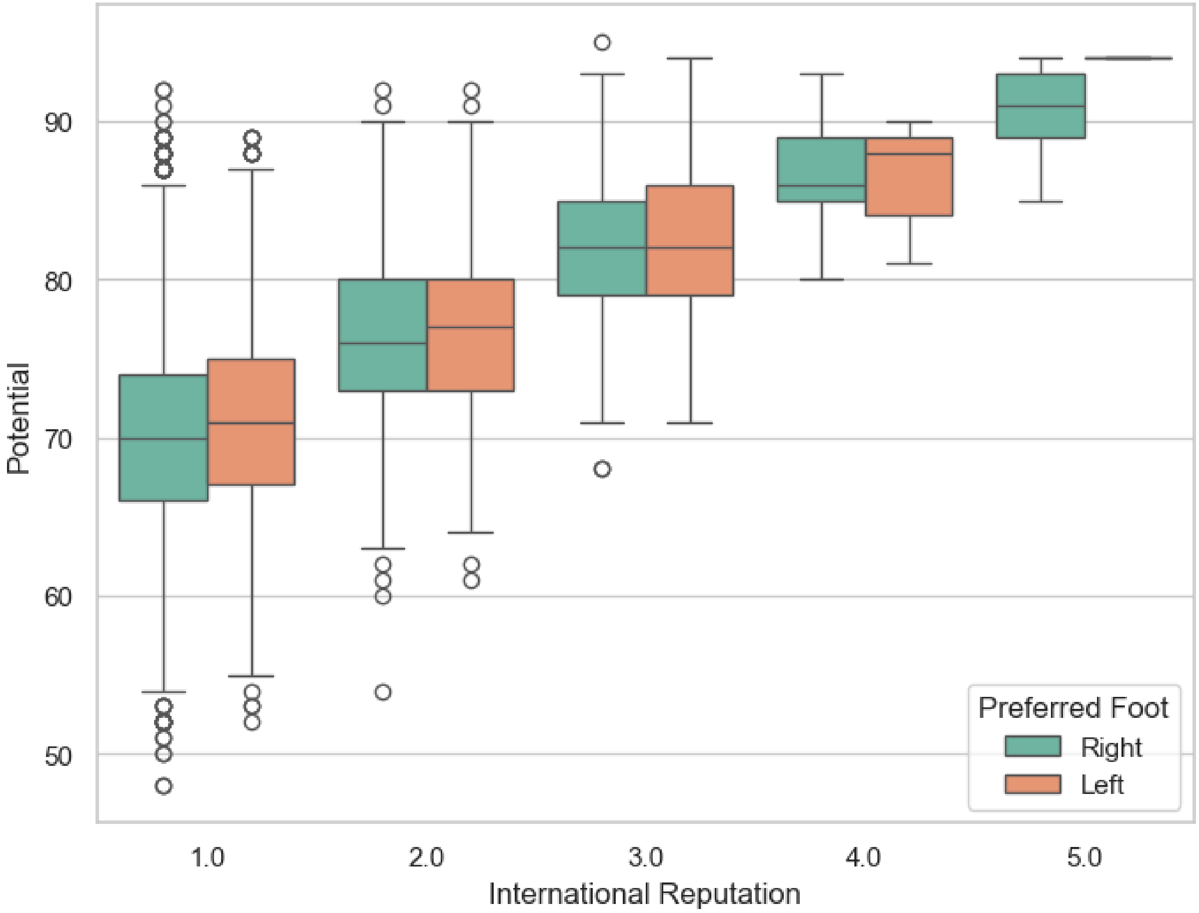
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.boxplot(x='International Reputation',y='Potential',data=df,palette='Set2')
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
1.0


90
80
70
60
50
Potential
1.0
2.0
3.0
4.0
5.0
International Reputation

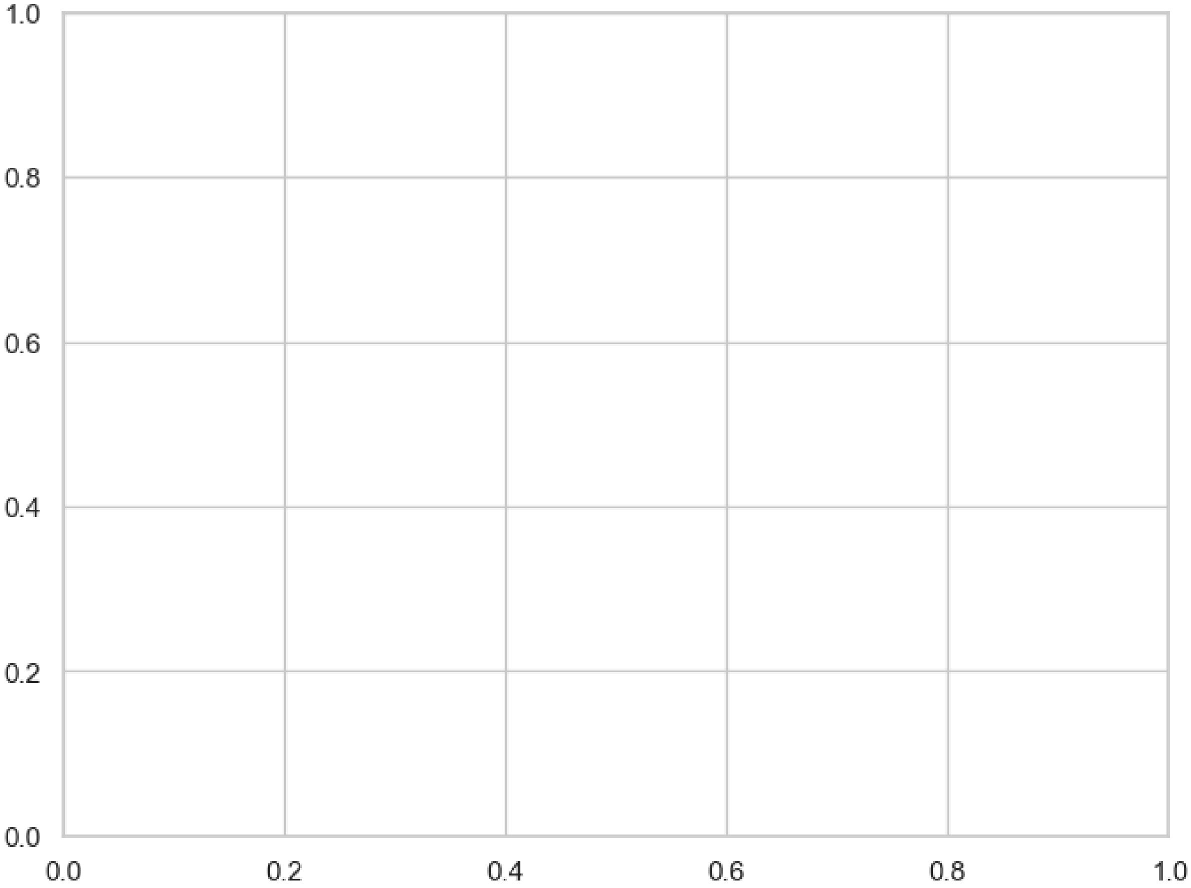
In [54]:

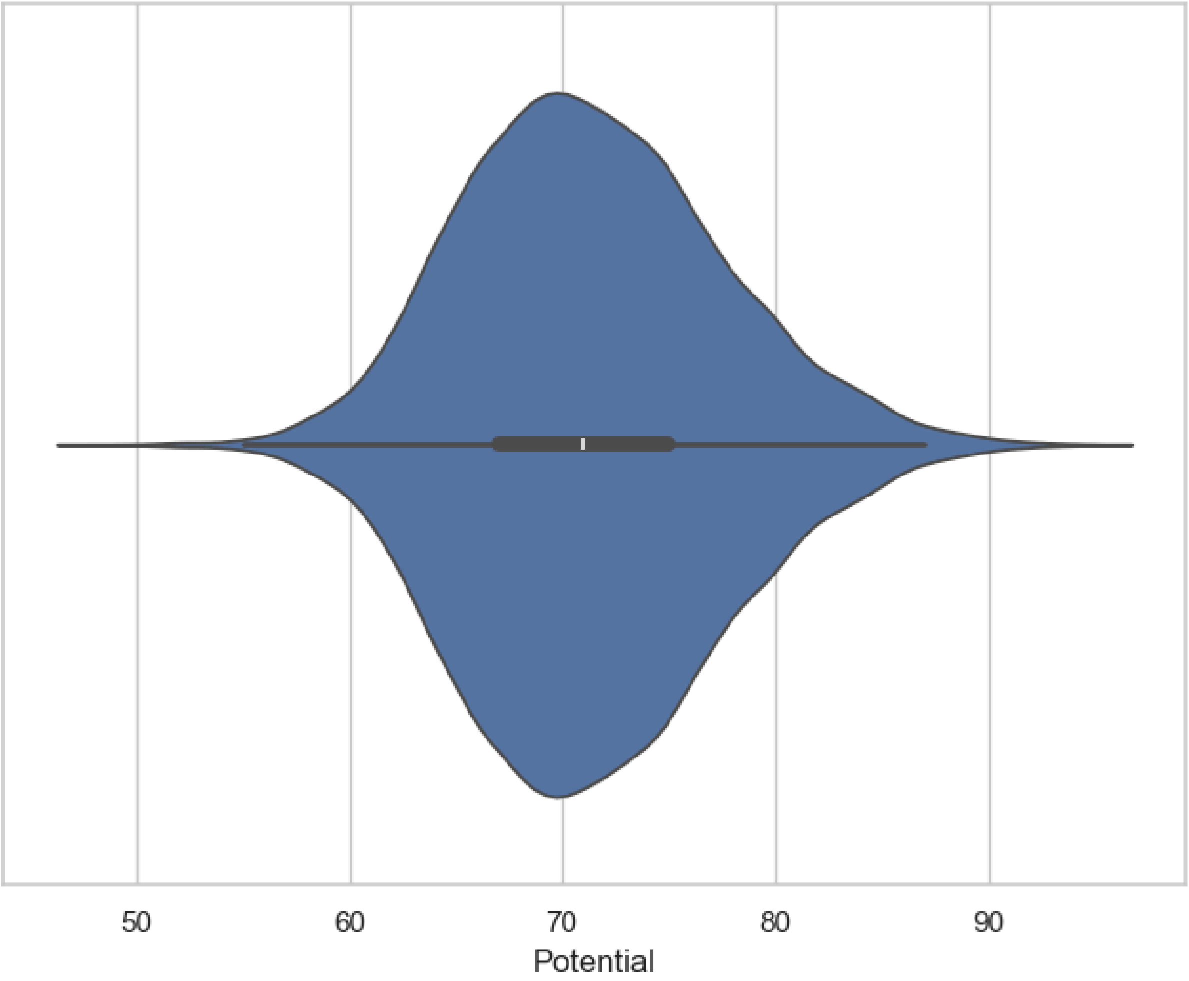
```python
f,ax=plt.subplots(figsize=(8,6))
ax=sns.boxplot(x='International Reputation',y='Potential',hue='Preferred Foot',data
plt.show()
```



In [56]:

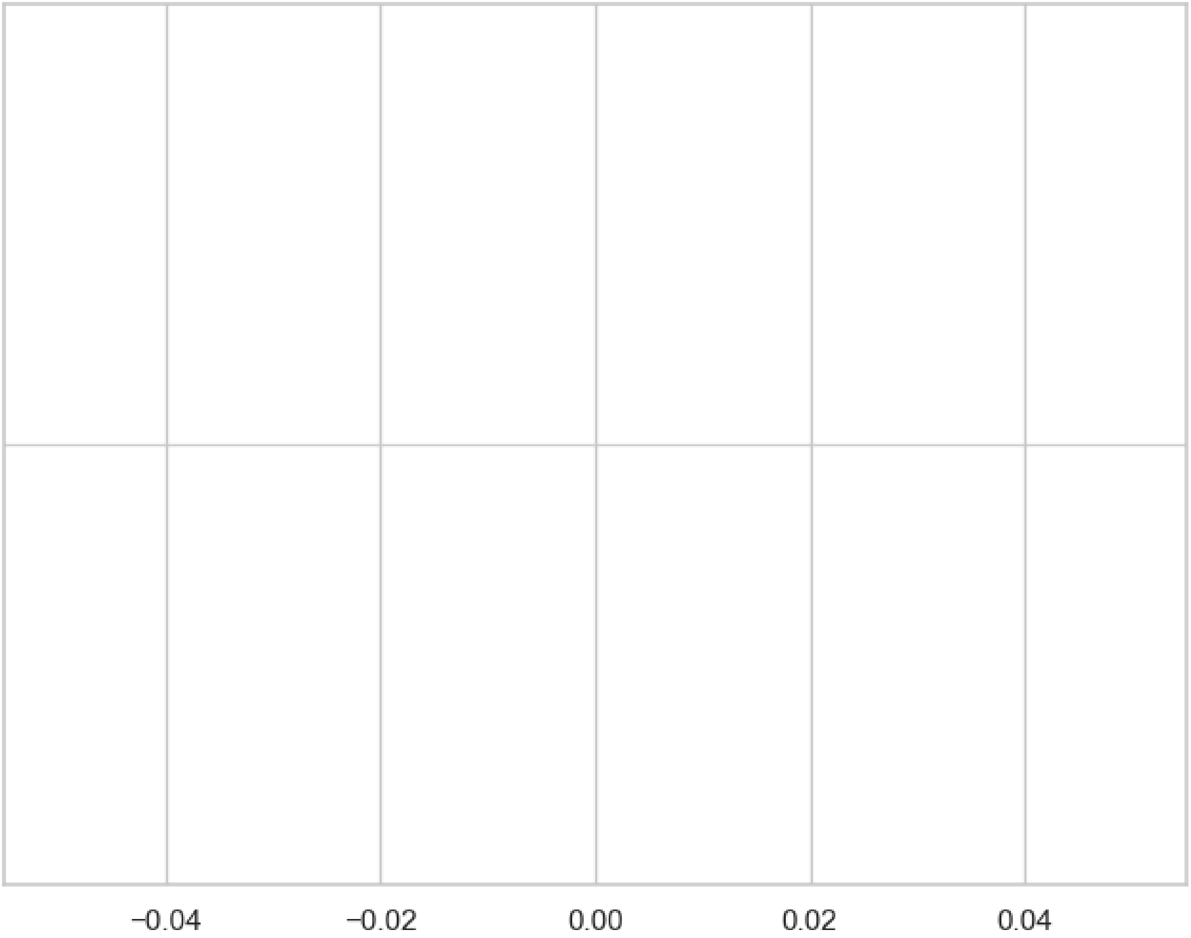
```python
f,ax=plt.subplots(figsize=(8,6))
ax=sns.violinplot(x=df['Potential'])
plt.show()
```

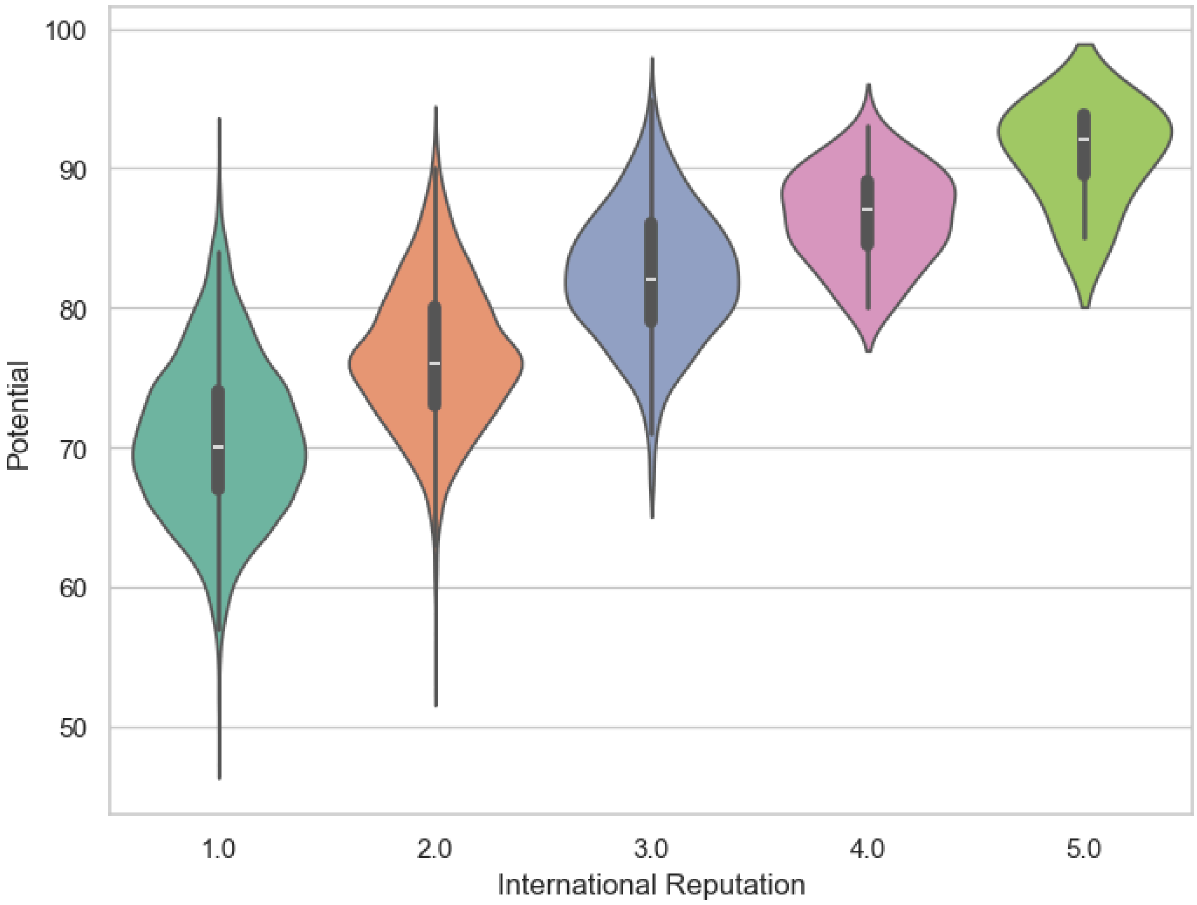

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



In [58]:

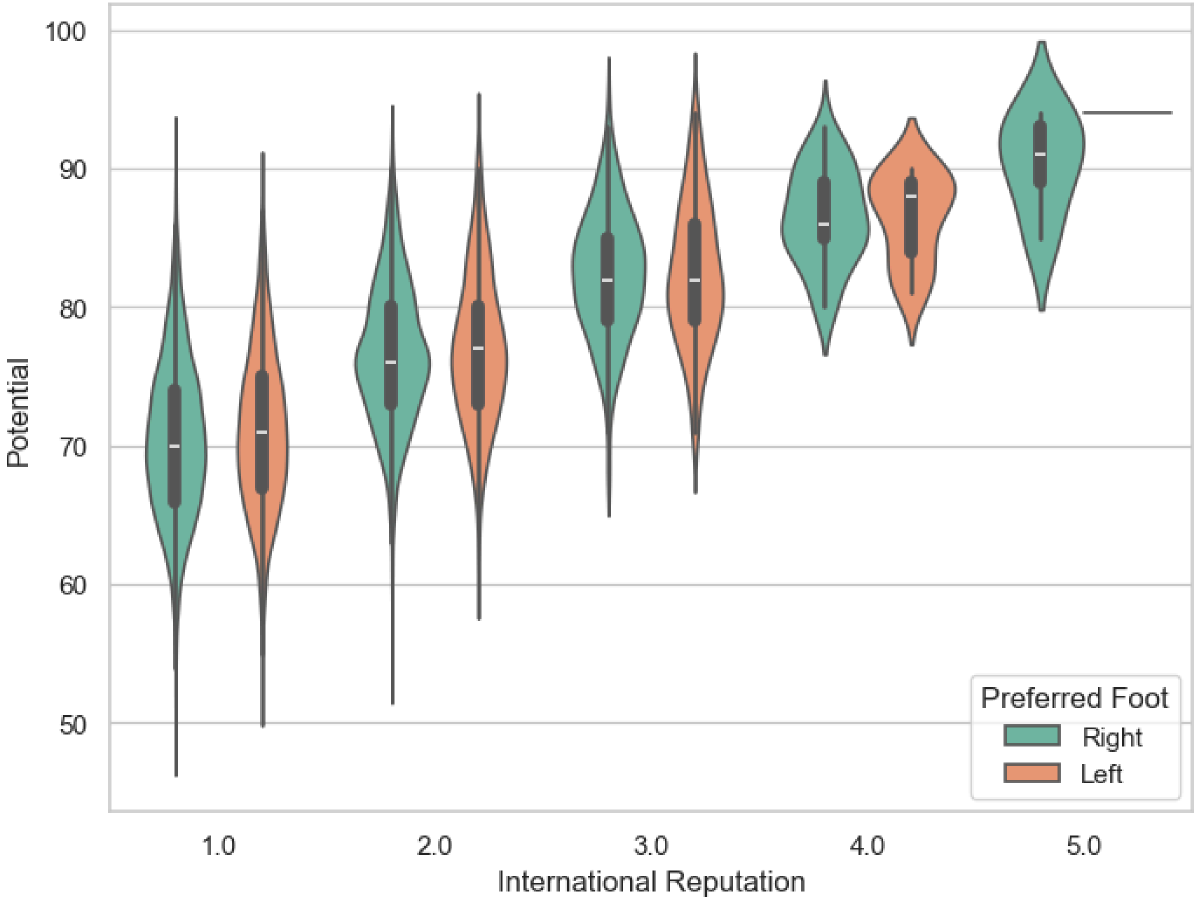
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.violinplot(x='International Reputation',y='Potential',data=df,palette='Set2'
plt.show()
```


−0.04
−0.02
0.00
0.02
0.04


100
90
80
70
60
50
Potential
1.0
2.0
3.0
4.0
5.0
International Reputation

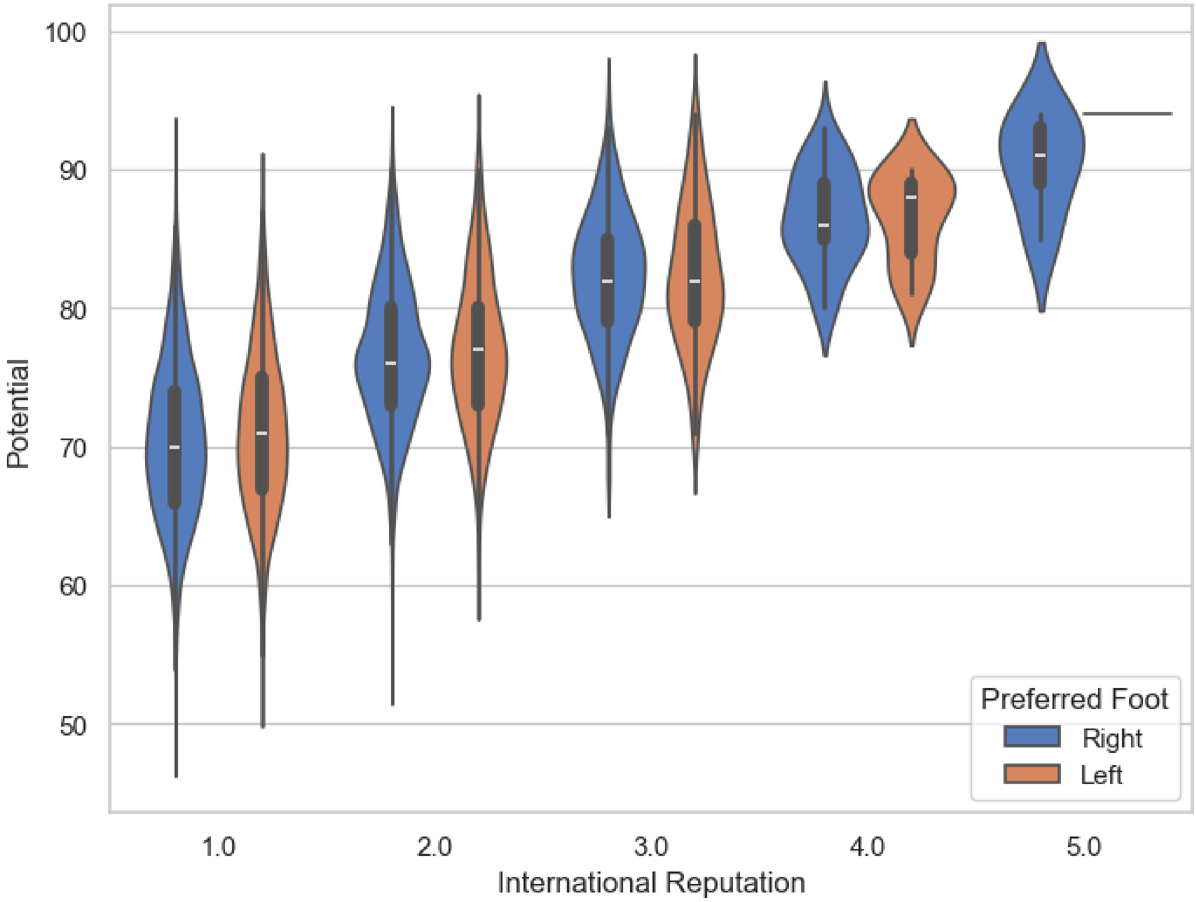
In [59]:

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.violinplot(x='International Reputation',y='Potential',data=df,palette='Set2'
plt.show()
```

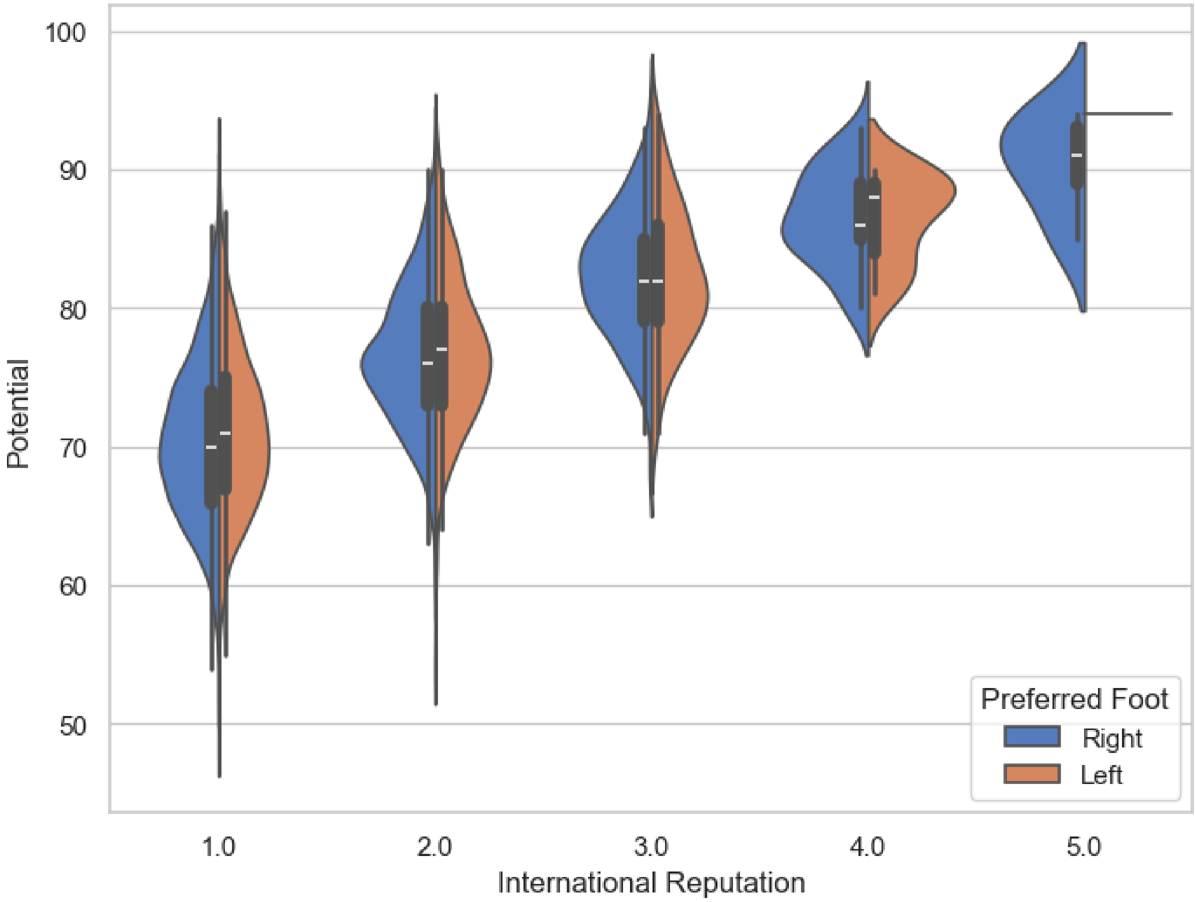


In [60]:

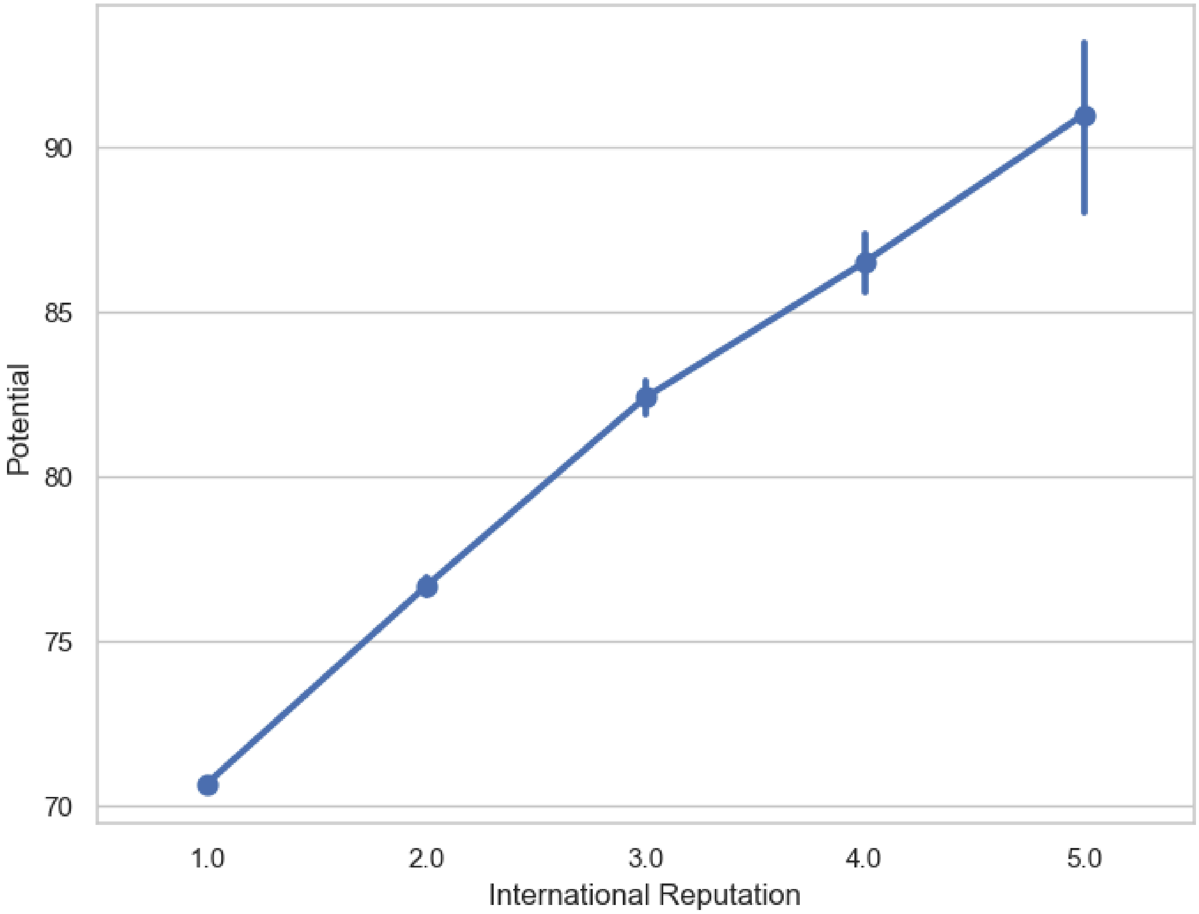
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.violinplot(x='International Reputation',y='Potential',data=df,palette='muted
plt.show()
```



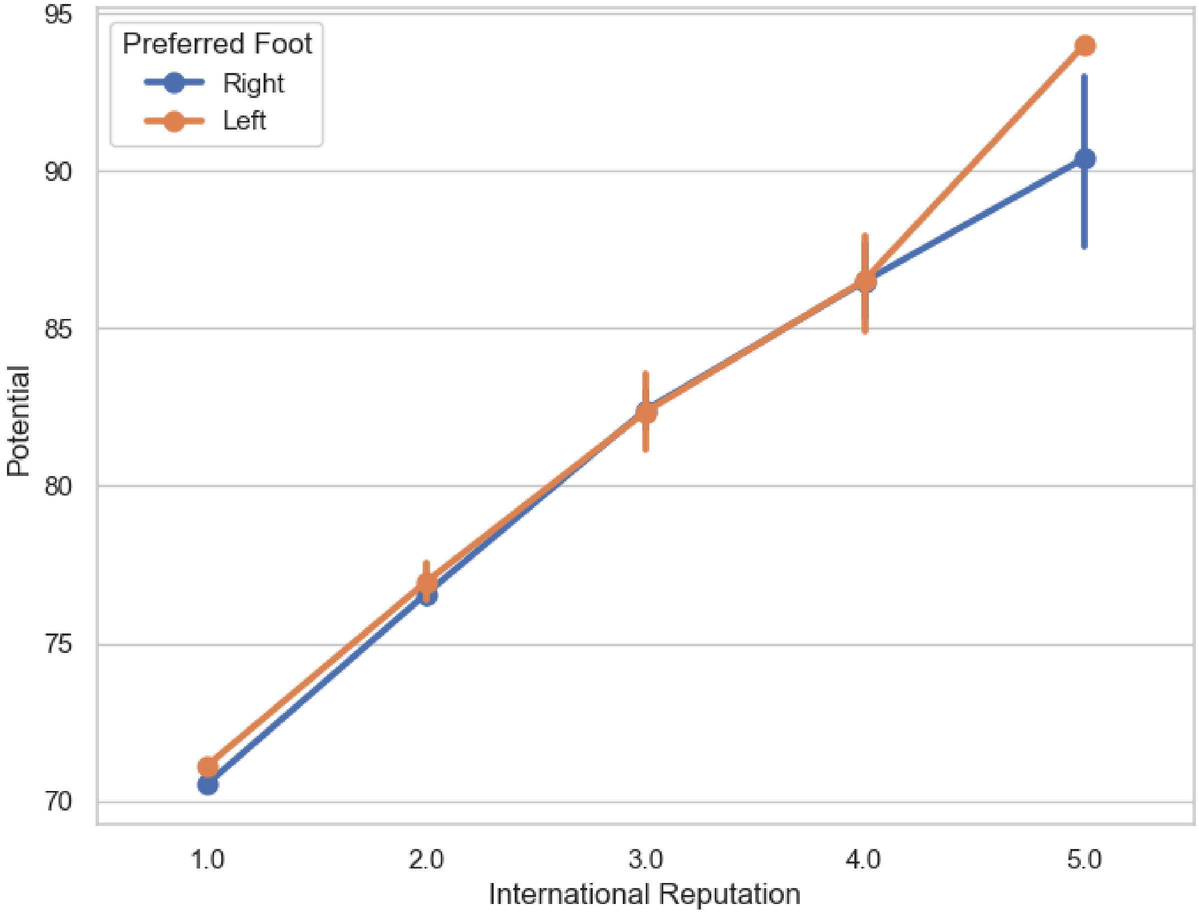
In [62]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.violinplot(x='International Reputation',y='Potential',data=df,palette='muted
plt.show()
```



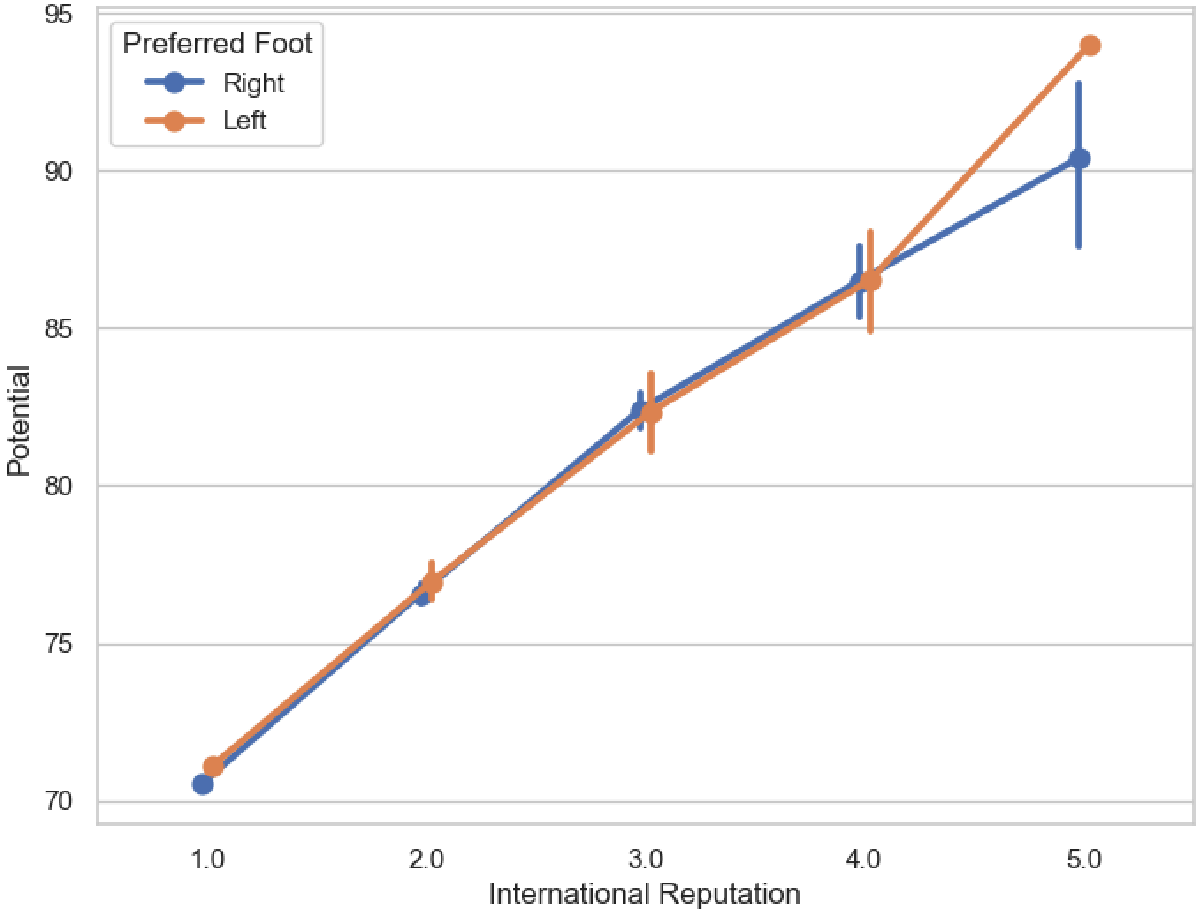
In [63]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.pointplot(x='International Reputation',y='Potential',data=df)
plt.show()
```



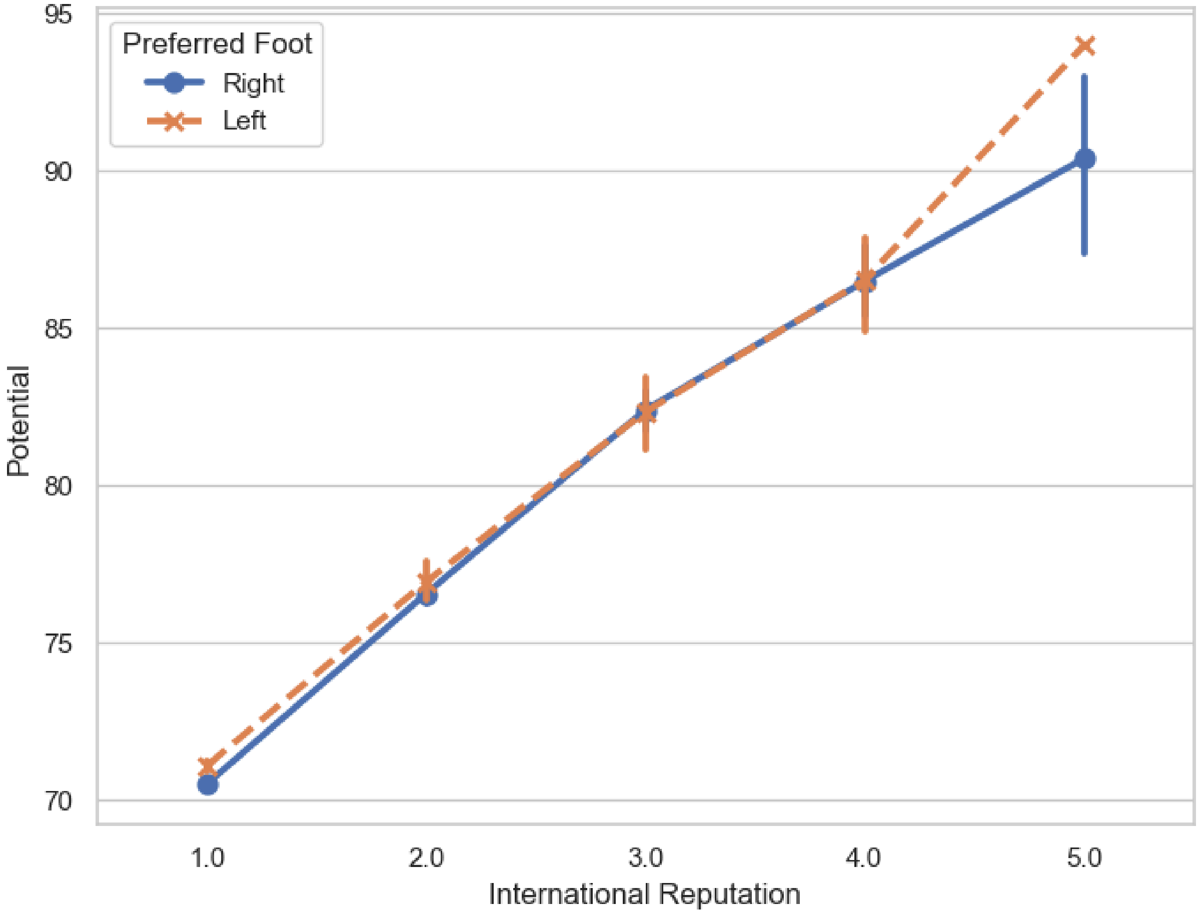
```
In [64]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.pointplot(x='International Reputation',y='Potential',hue='Preferred Foot',da
         plt.show()
```



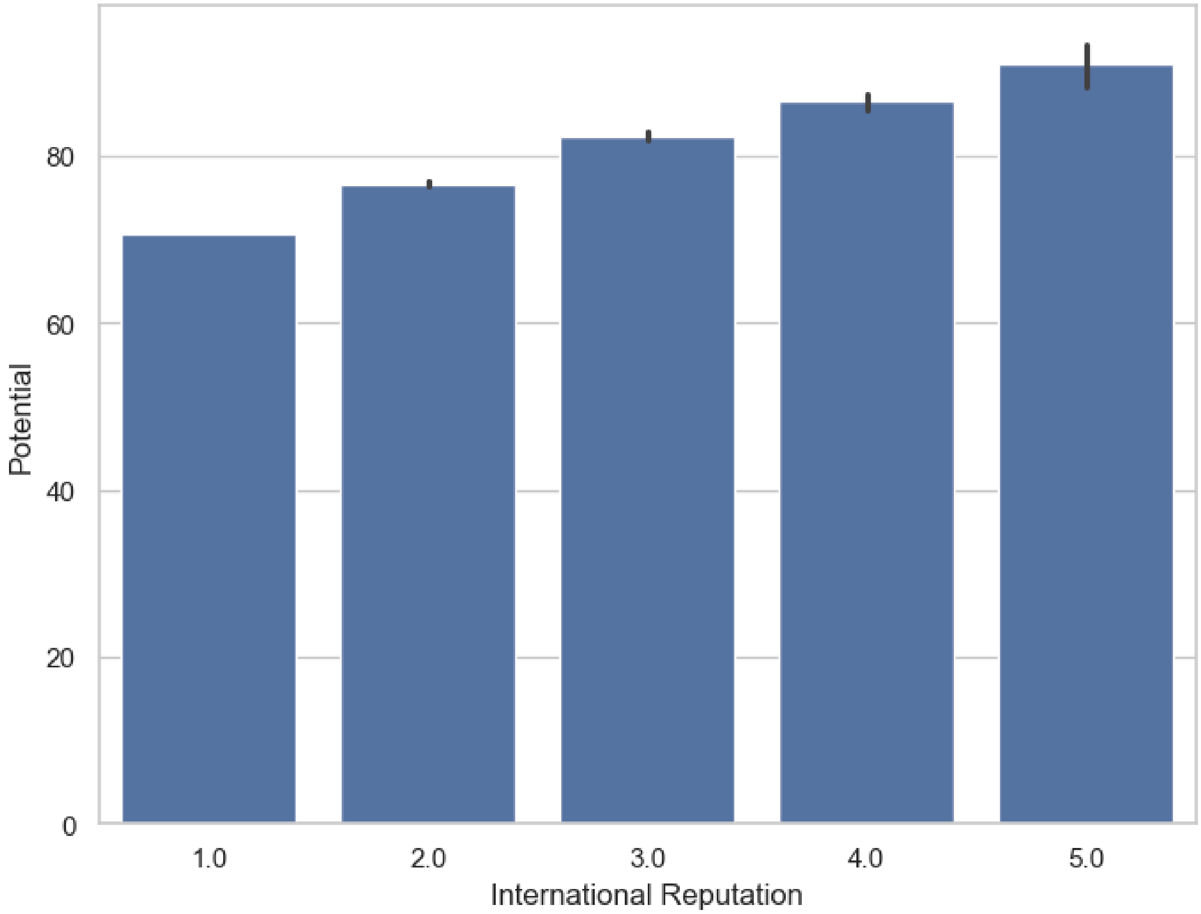
```
In [65]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.pointplot(x='International Reputation',y='Potential',hue='Preferred Foot',da
         plt.show()
```



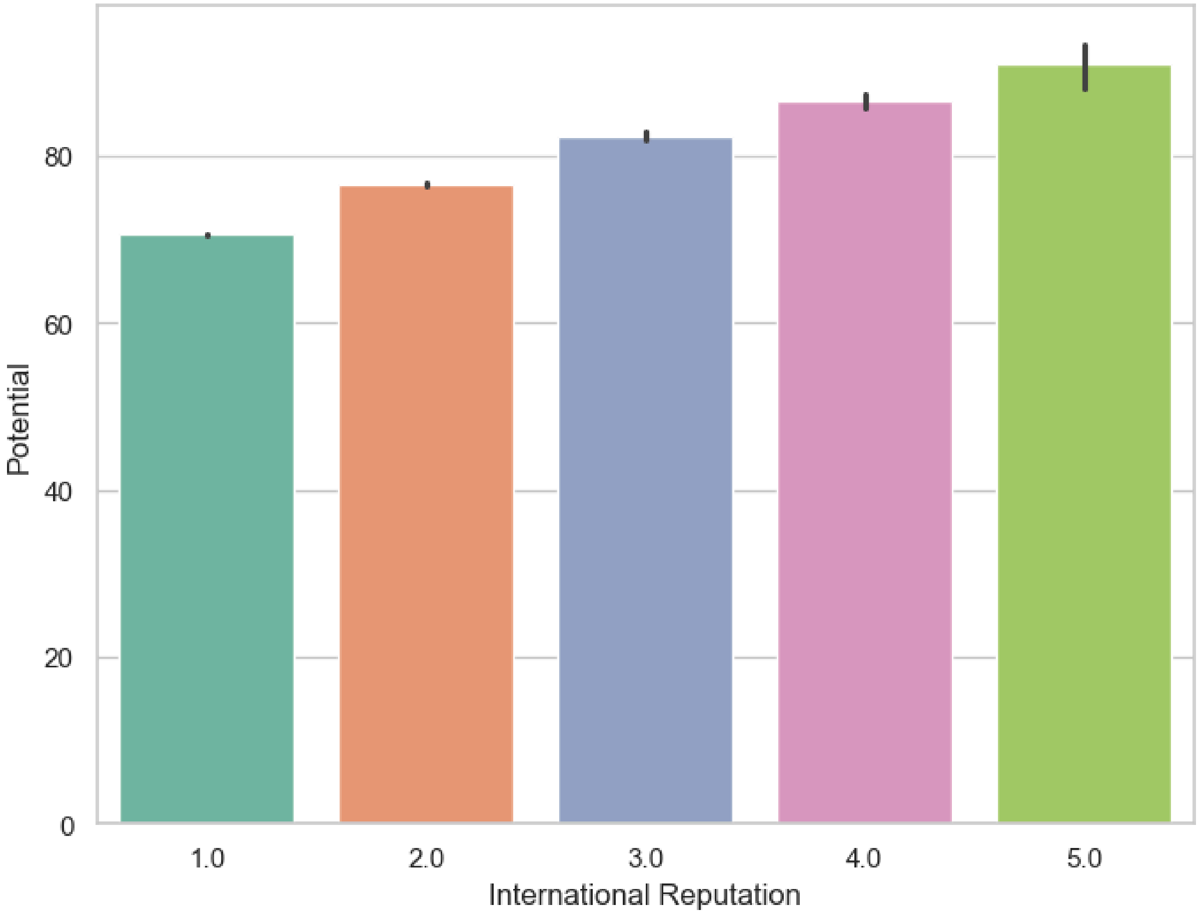
In [66]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.pointplot(x='International Reputation',y='Potential',hue='Preferred Foot',da
plt.show()
```



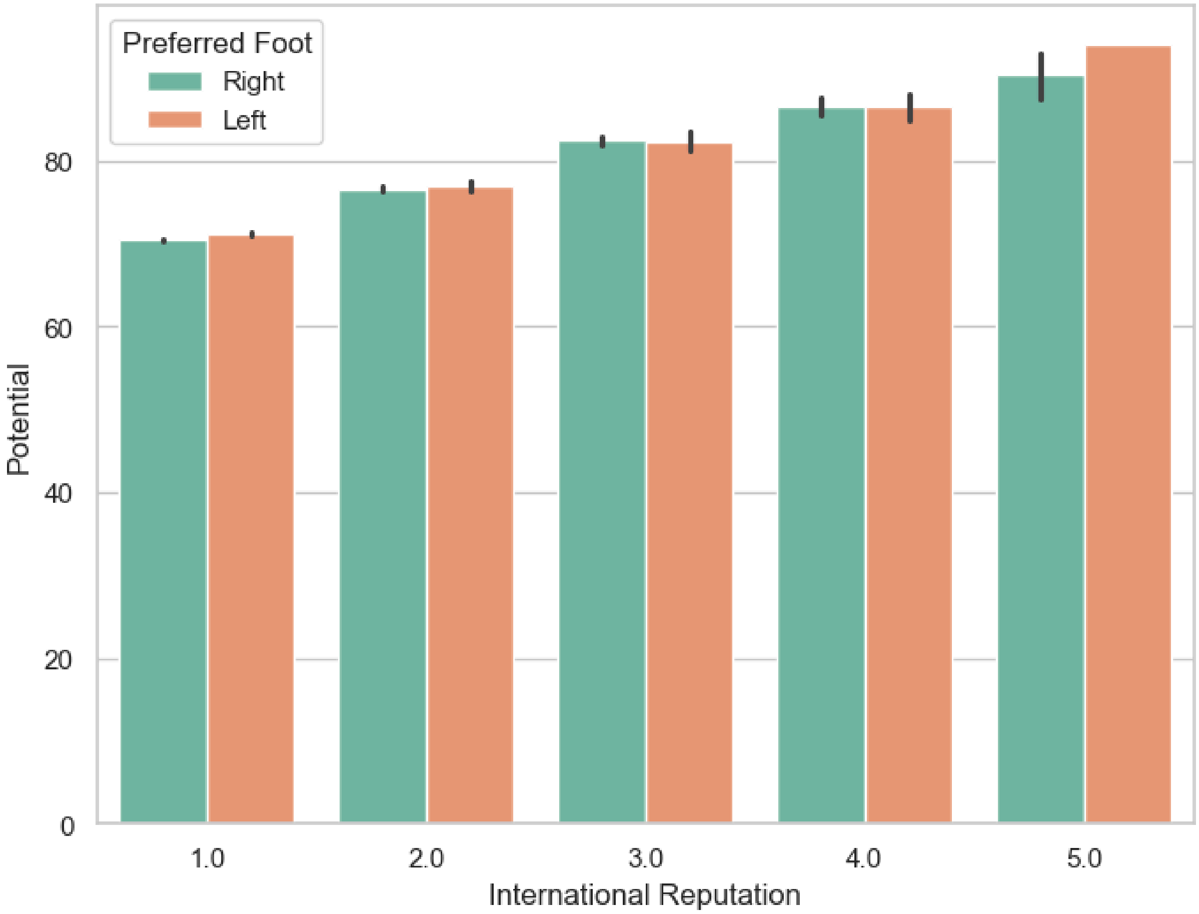
```
In [67]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.barplot(x='International Reputation',y='Potential',data=df)
         plt.show()
```



```
In [68]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.barplot(x='International Reputation',y='Potential',data=df,palette='Set2')
         plt.show()
```

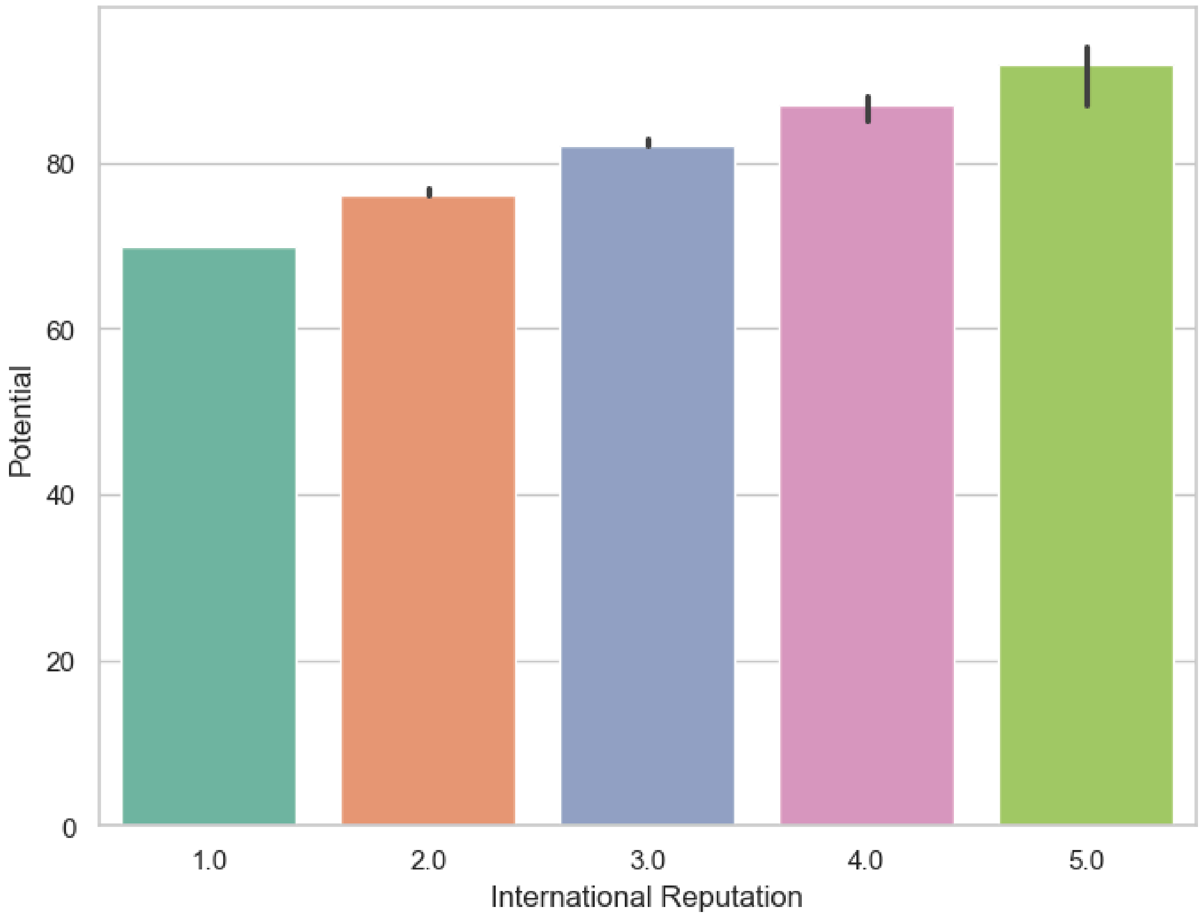


In [69]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.barplot(x='International Reputation',y='Potential',data=df,palette='Set2',hu
plt.show()
```

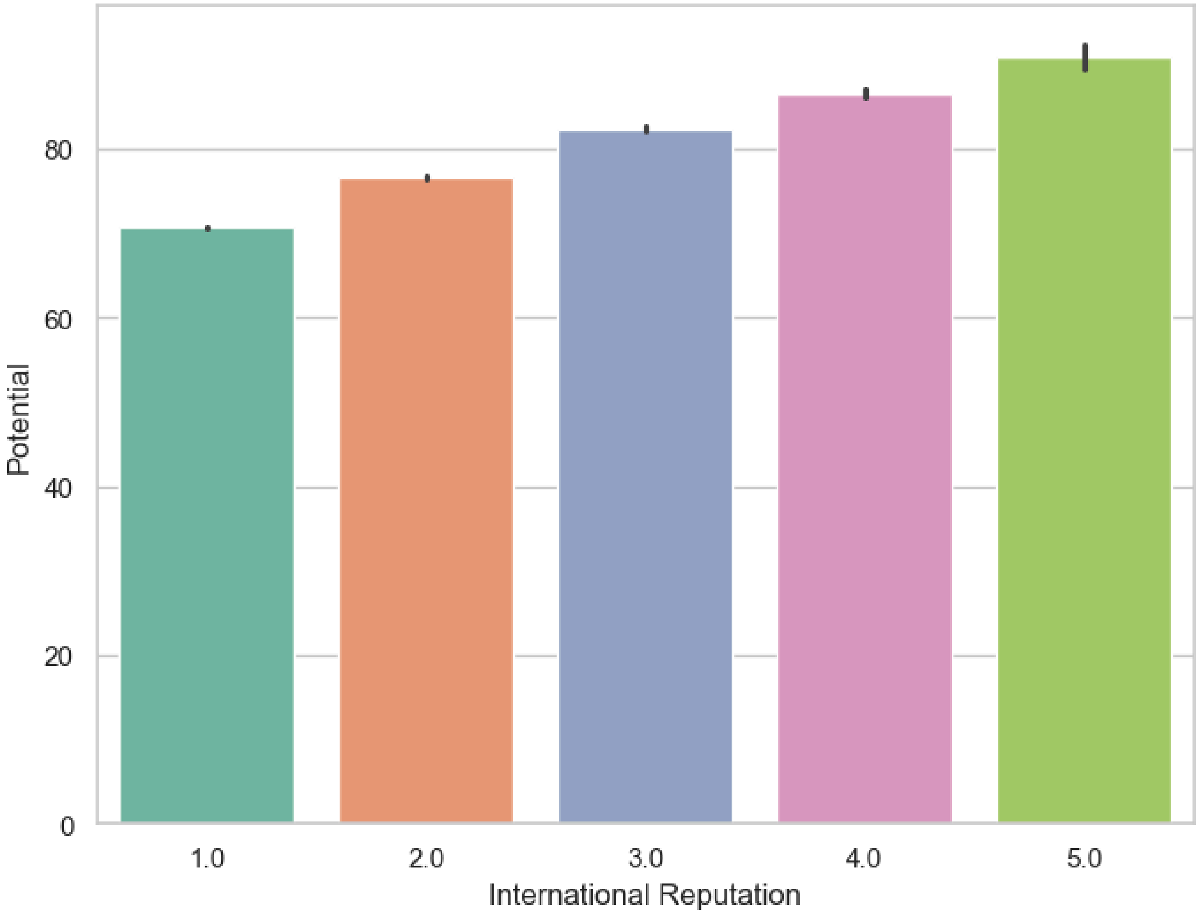


In [73]:

```
from numpy import median
f,ax=plt.subplots(figsize=(8,6))
ax=sns.barplot(x='International Reputation',y='Potential',data=df,palette='Set2', e
plt.show()
```



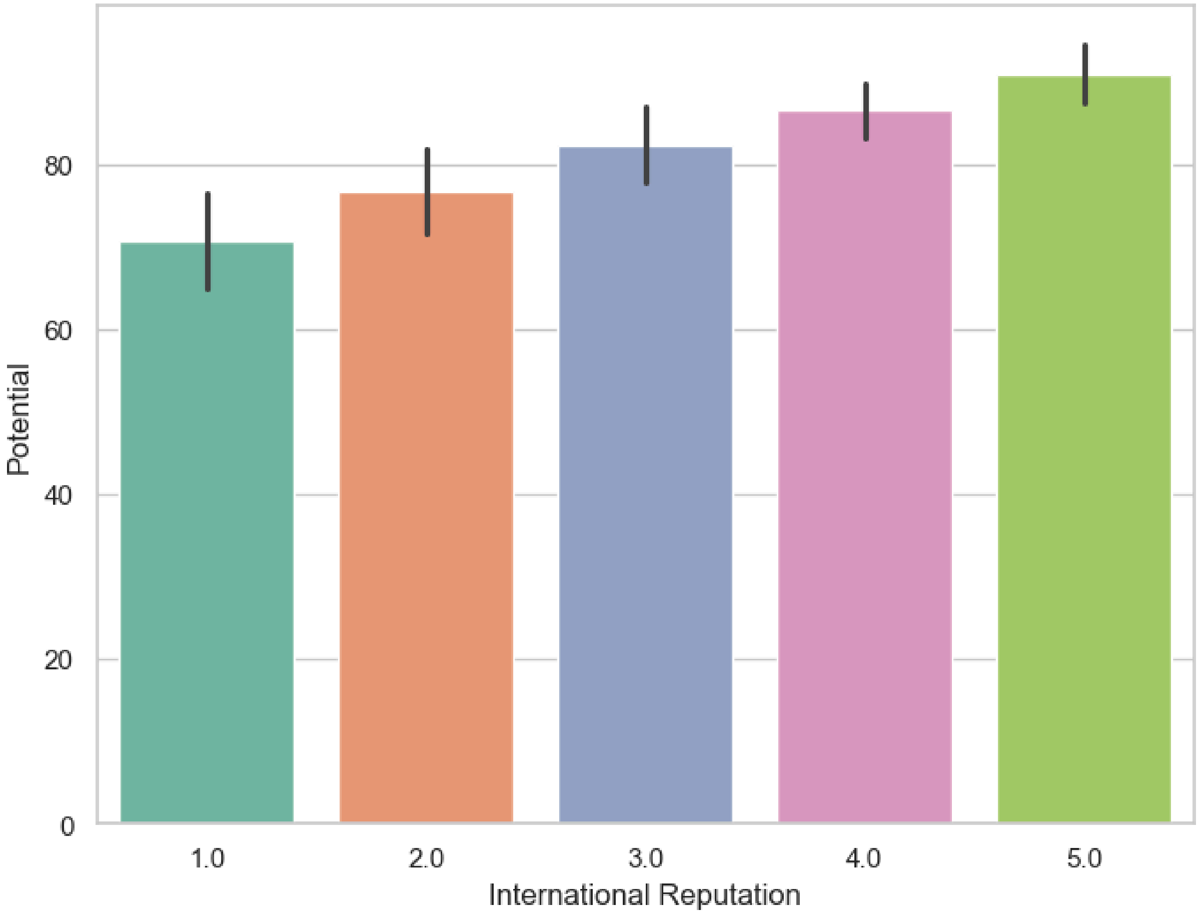
```
In [74]: from numpy import median
         f,ax=plt.subplots(figsize=(8,6))
         ax=sns.barplot(x='International Reputation',y='Potential',data=df,palette='Set2', c
         plt.show()
```



In [75]:

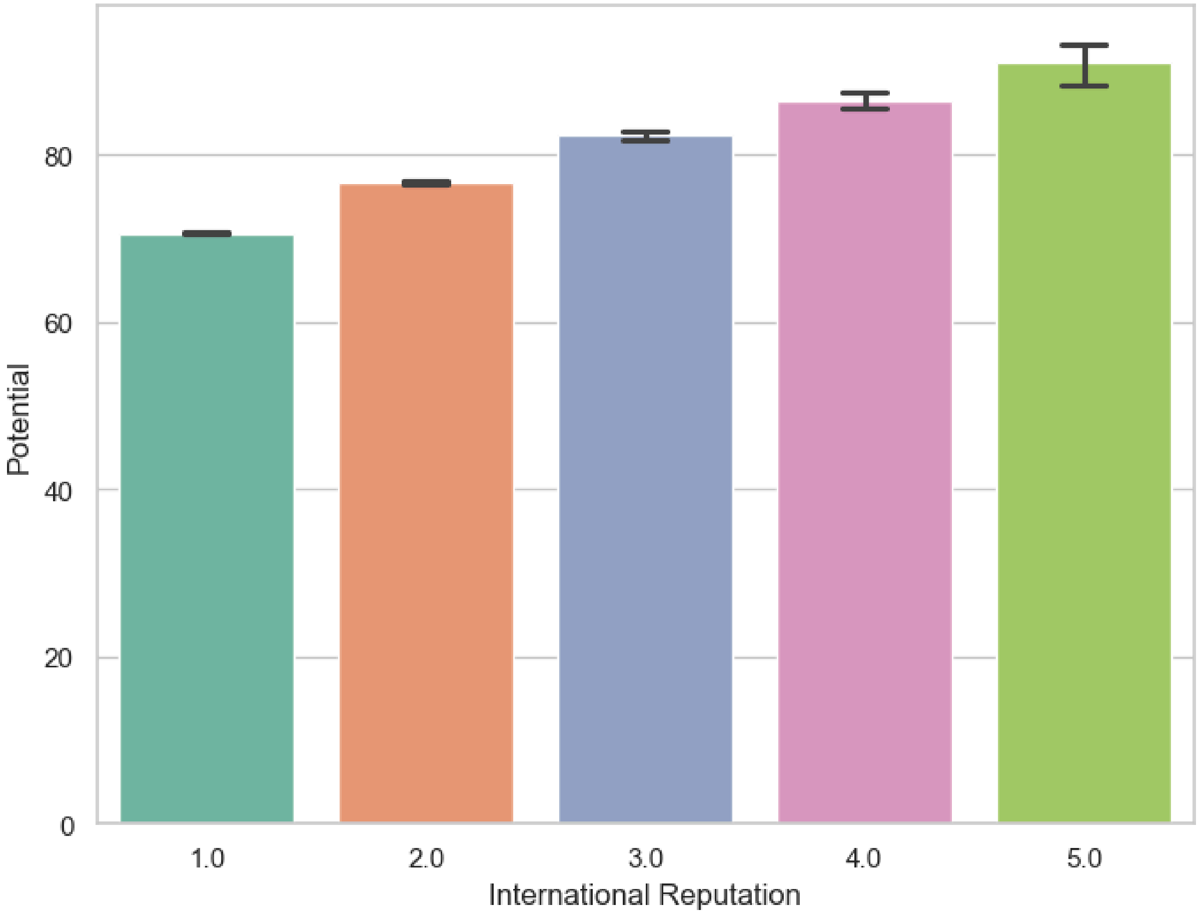
```
from numpy import median
f,ax=plt.subplots(figsize=(8,6))
ax=sns.barplot(x='International Reputation',y='Potential',data=df,palette='Set2', c

plt.show()
```

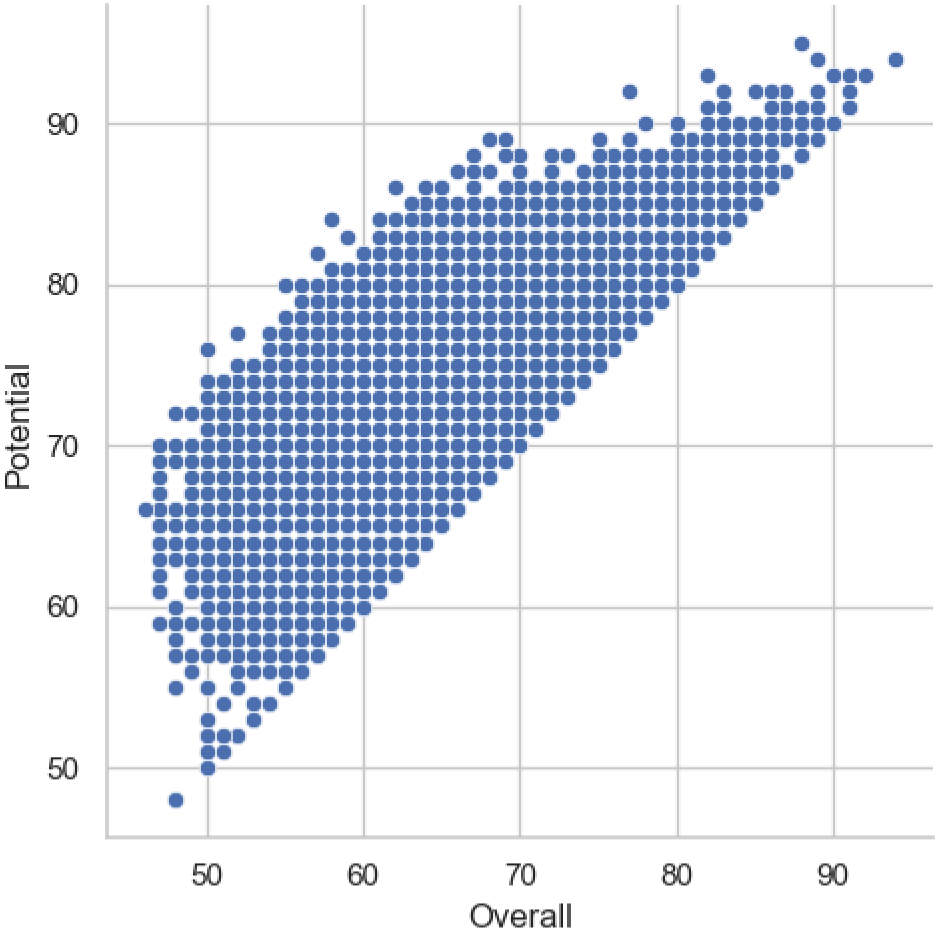


In [76]:

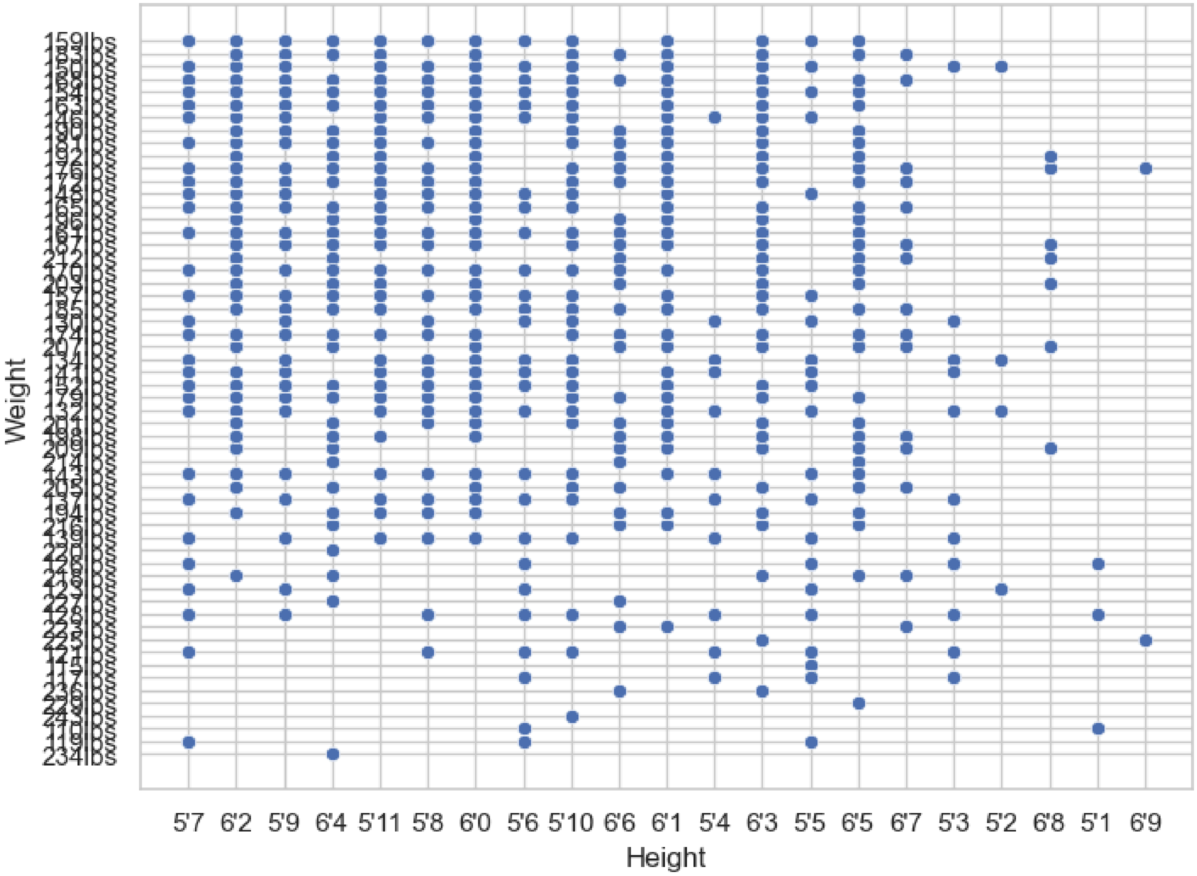
```
from numpy import median
f,ax=plt.subplots(figsize=(8,6))
ax=sns.barplot(x='International Reputation',y='Potential',data=df,palette='Set2',ca
plt.show()
```



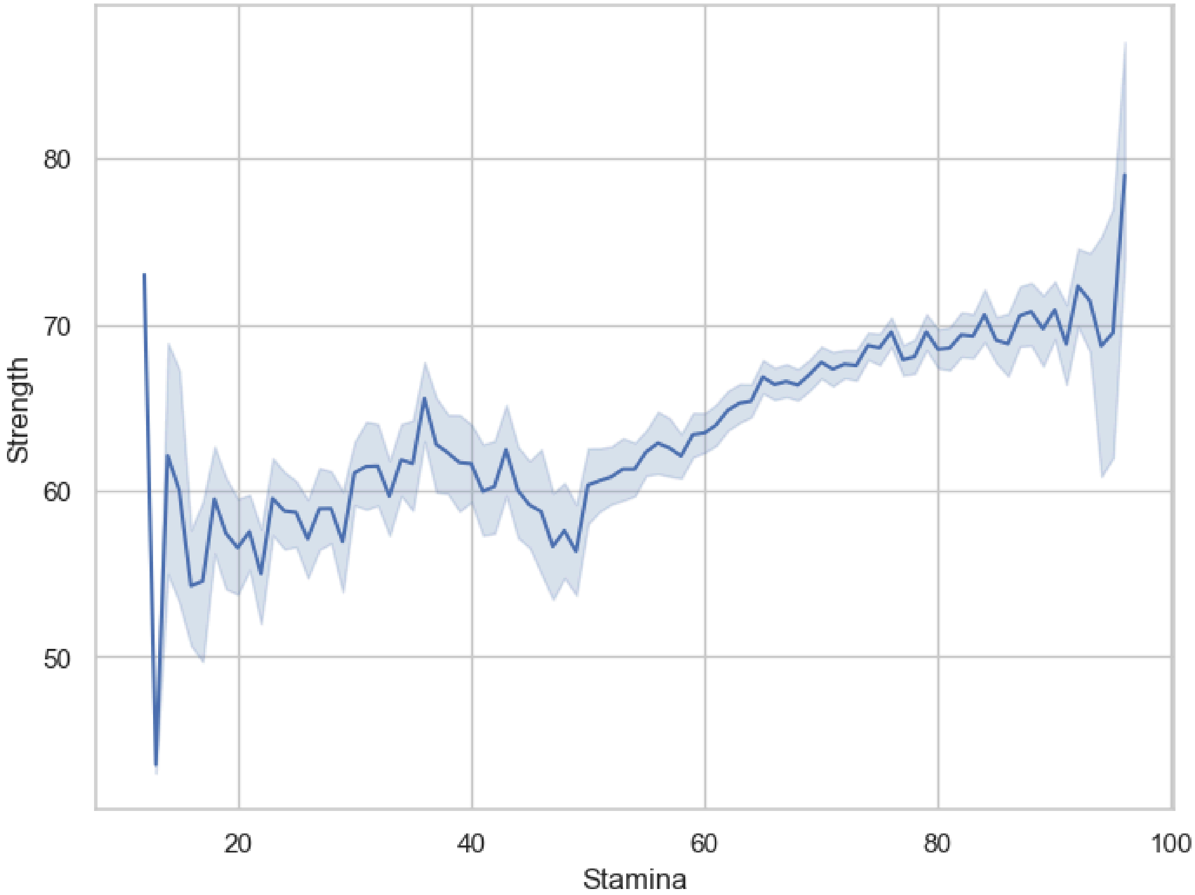
```
In [77]: h=sns.relplot(x='Overall',y='Potential',data=df)
         plt.show()                                                   # relational plot
```



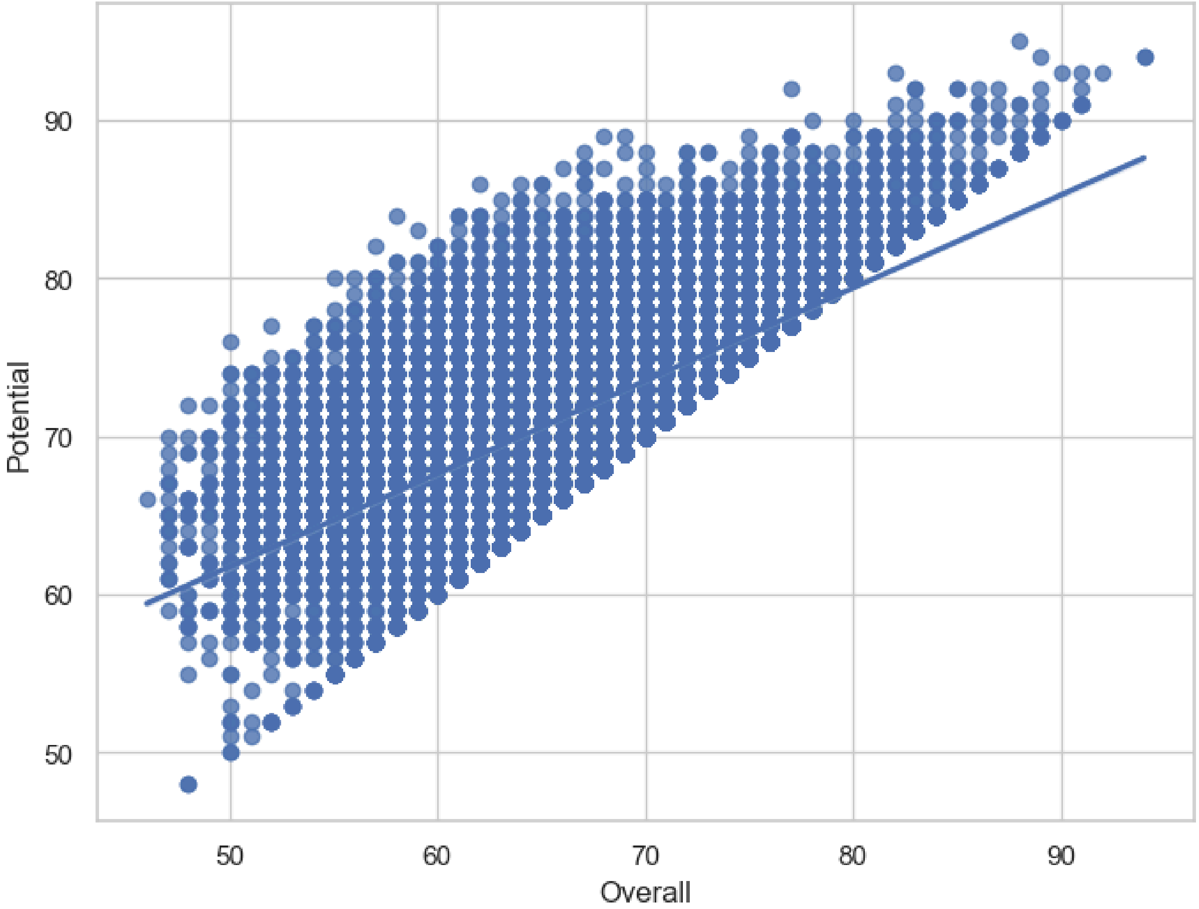
```
In [80]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.scatterplot(x='Height',y='Weight',data=df)       # scatter plot
         plt.show()
```



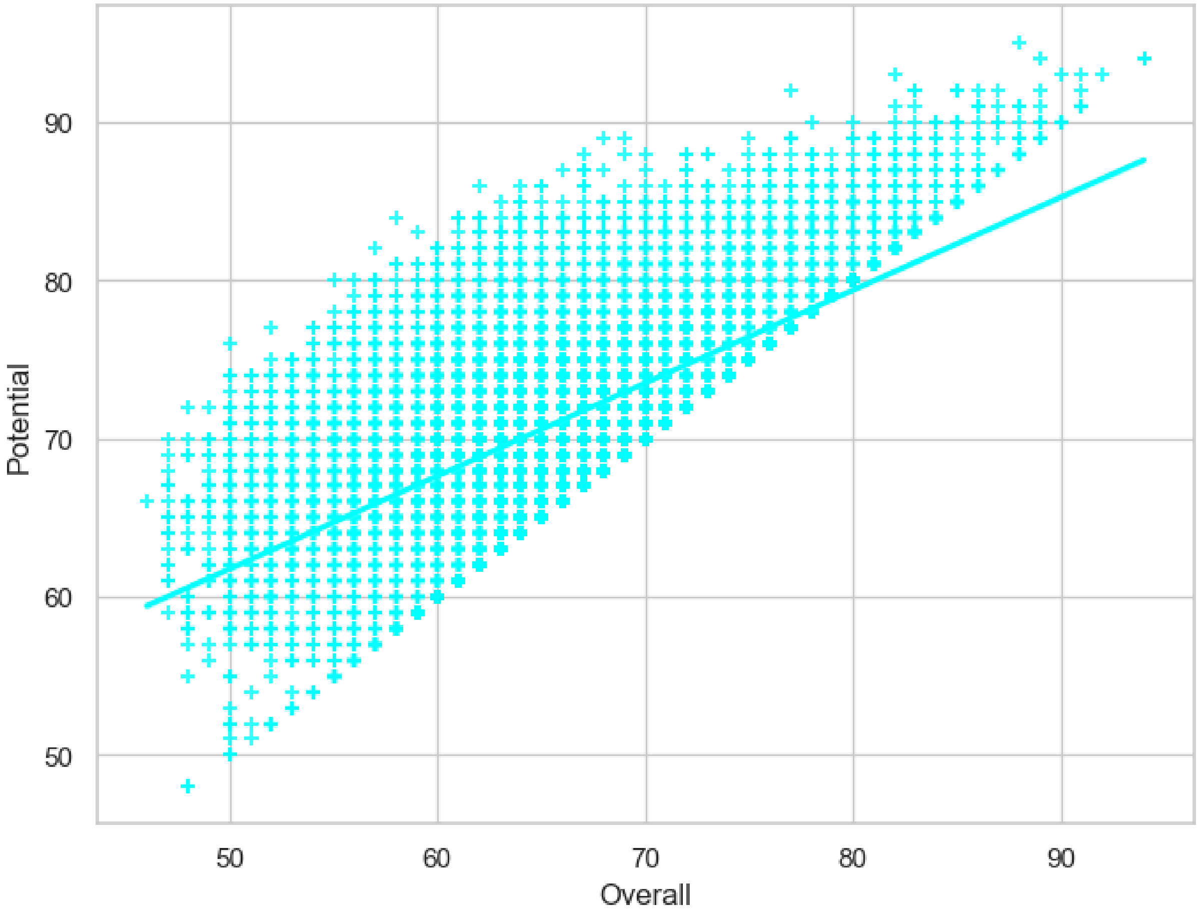
```
In [81]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.lineplot(x='Stamina',y='Strength',data=df)
         plt.show()                                              # lineplot function
```



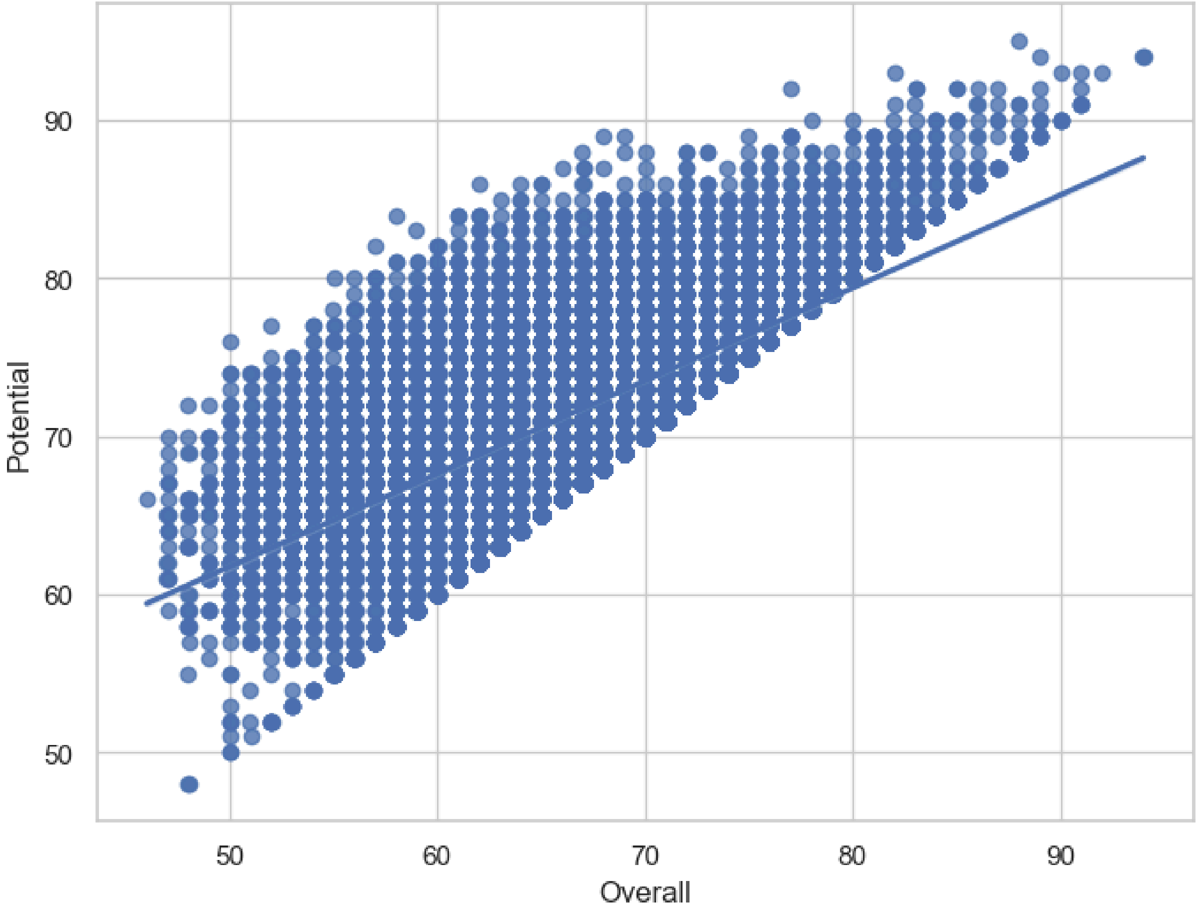
```
In [83]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.regplot(x='Overall',y='Potential',data=df)
         plt.show()
```



In [85]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.regplot(x='Overall',y='Potential',data=df,color='cyan',marker='+')
plt.show()
```

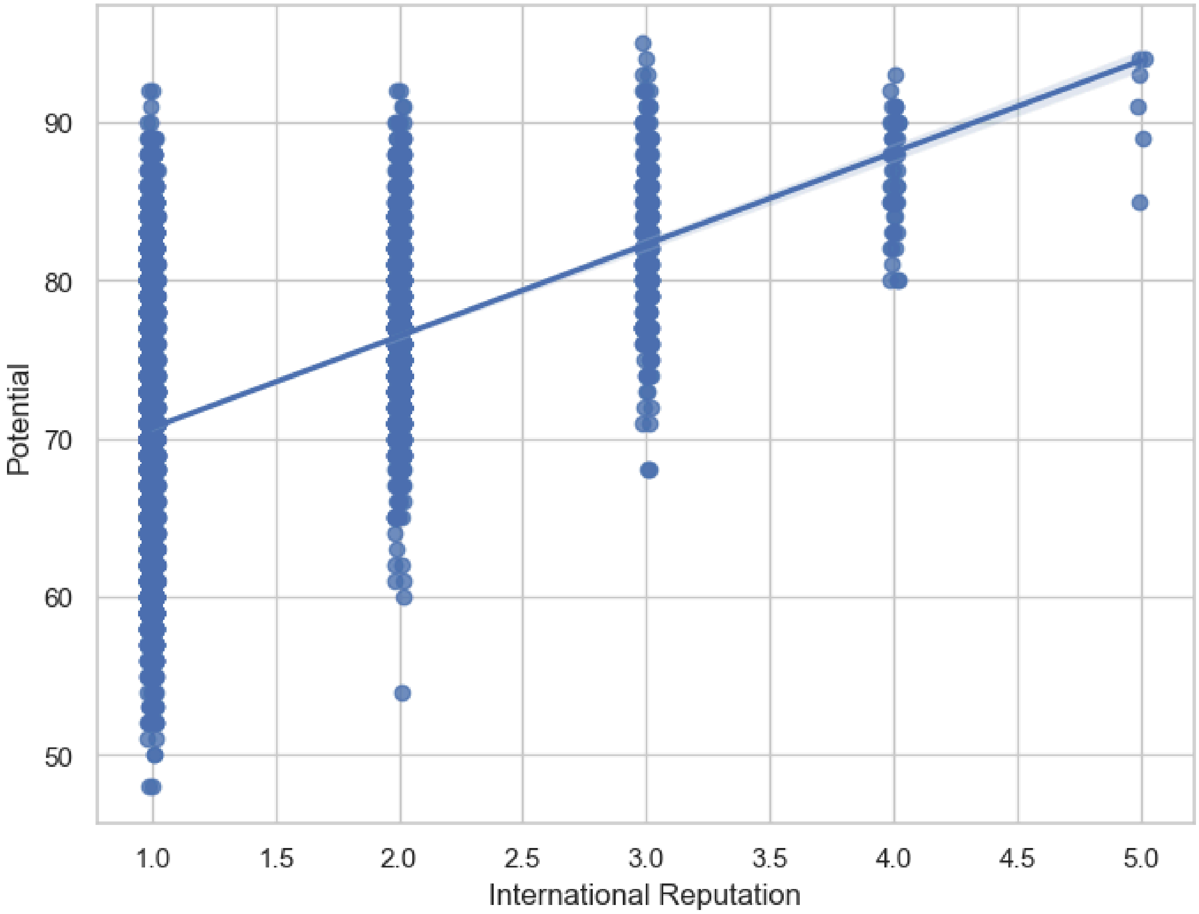


```
In [86]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.regplot(x='Overall',y='Potential',data=df,x_jitter=.02)
         plt.show()
```



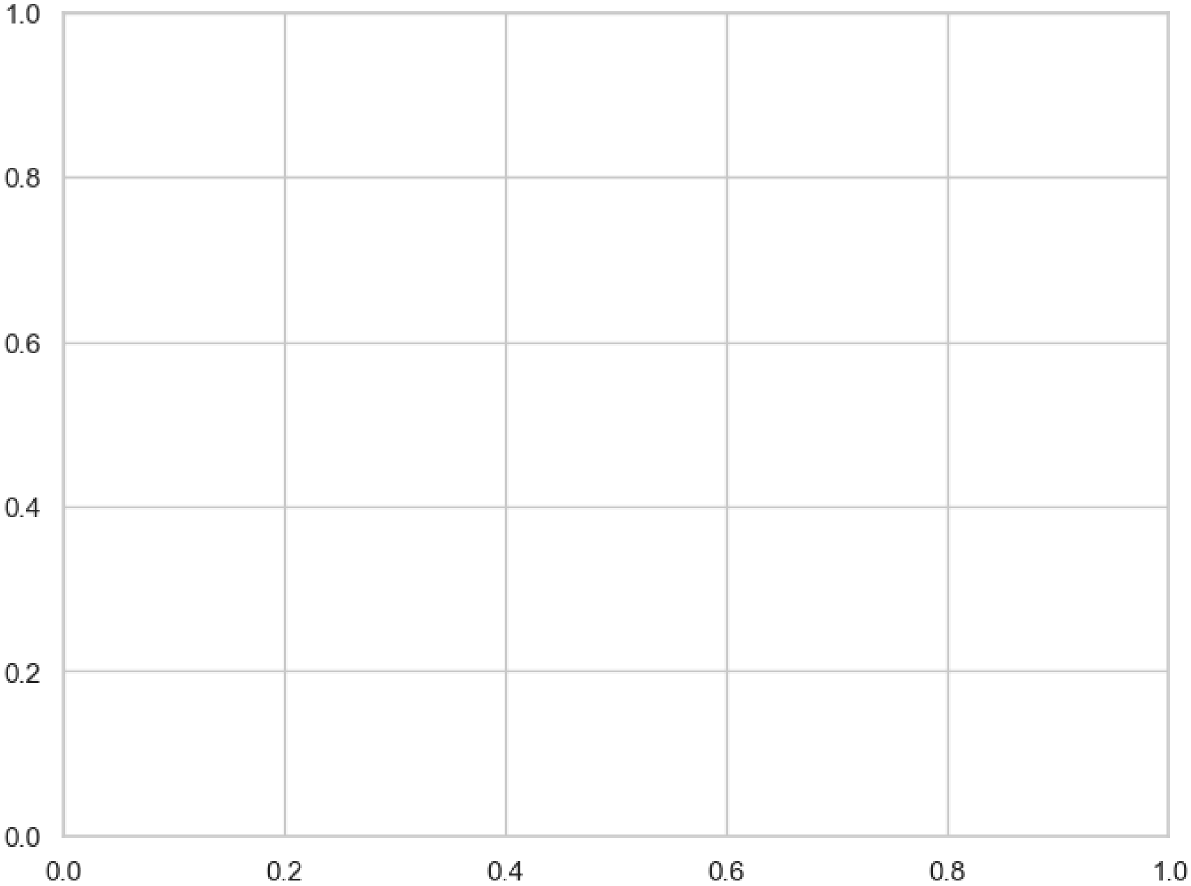
In [90]:

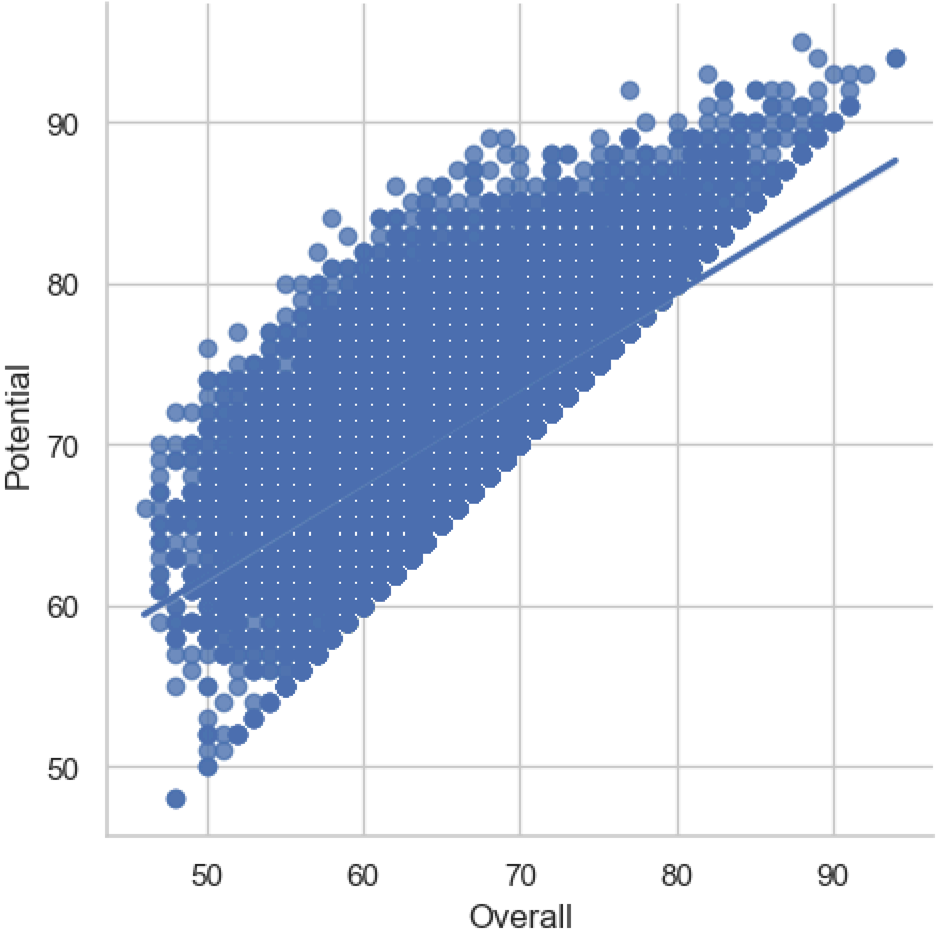
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.regplot(x='International Reputation',y='Potential',data=df,x_jitter=0.02)
plt.show()
```



In [92]:

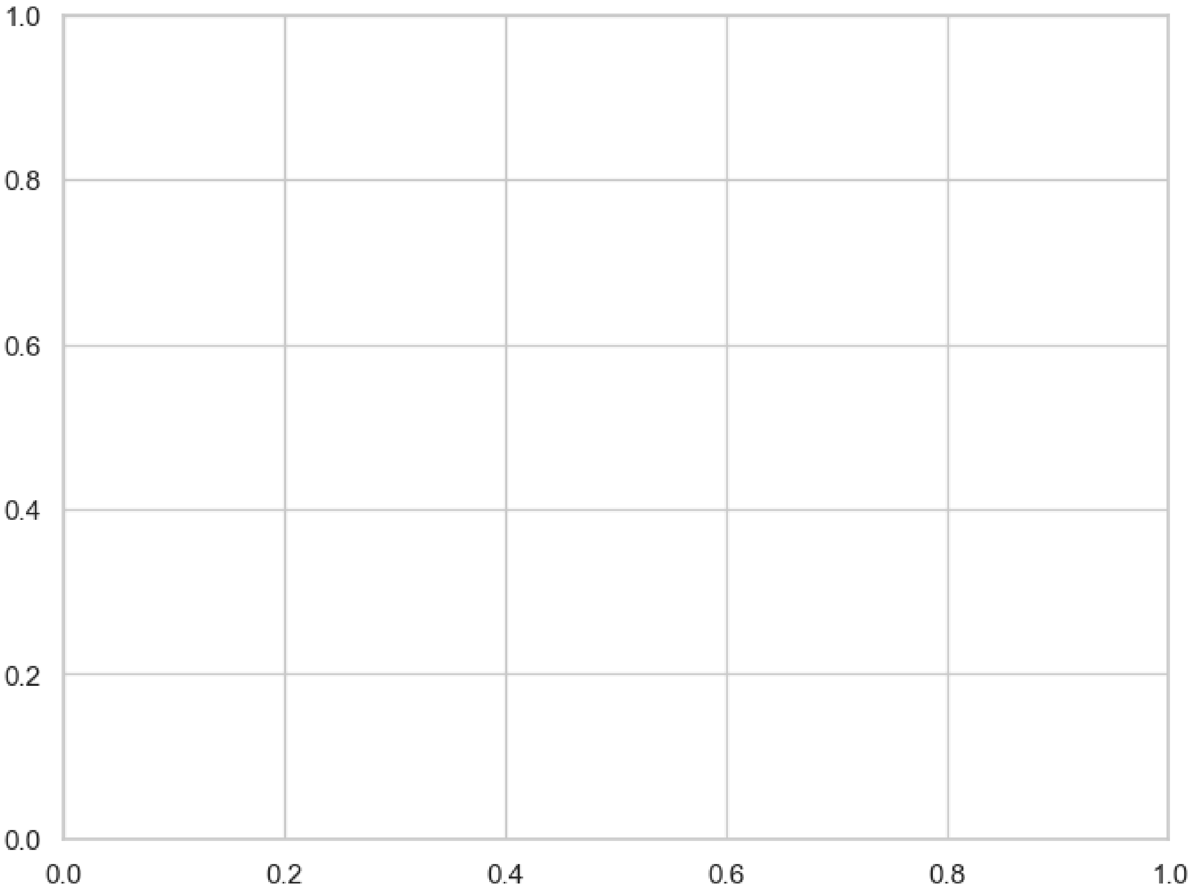
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.lmplot(x='Overall',y='Potential',data=df)
plt.show()
```

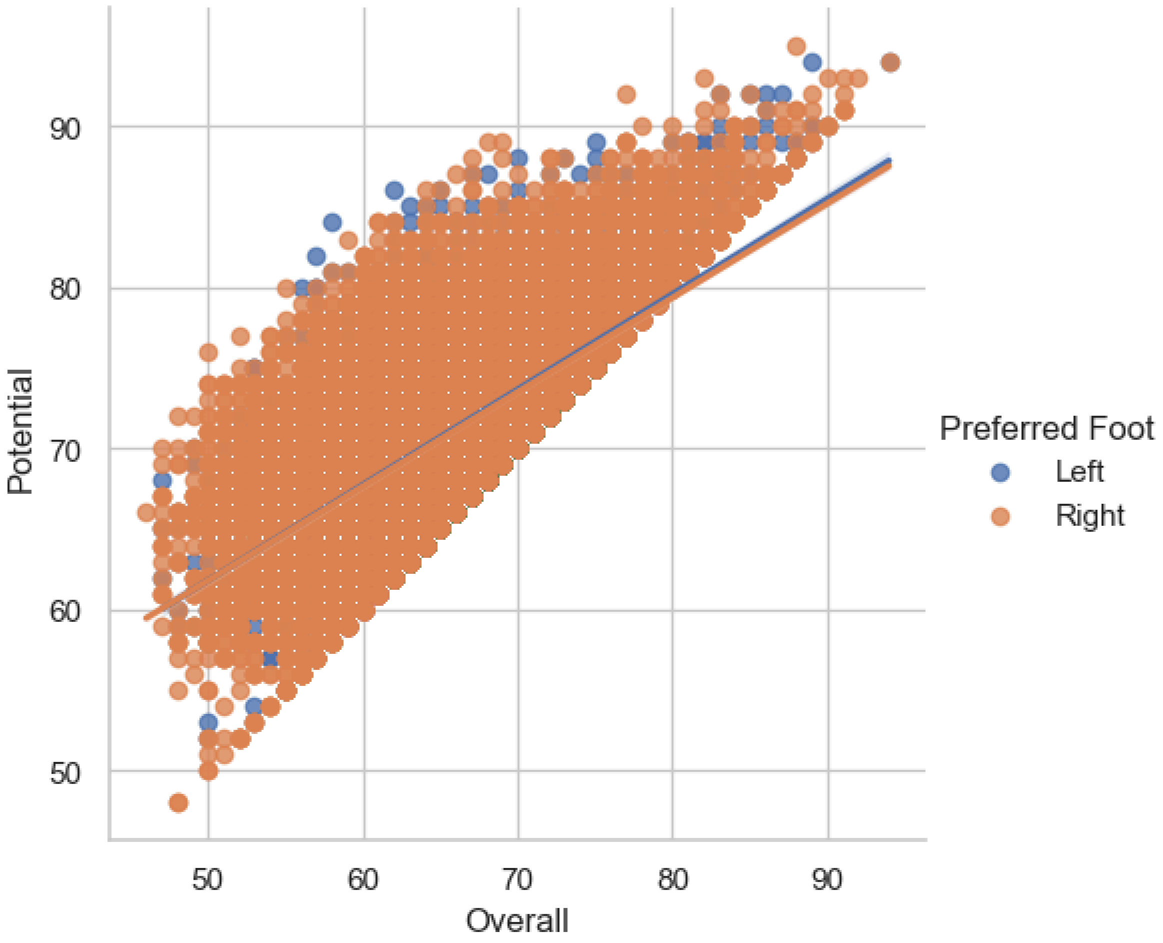

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


90
80
70
60
50
Potential
50
60
70
80
90
Overall

In [93]:

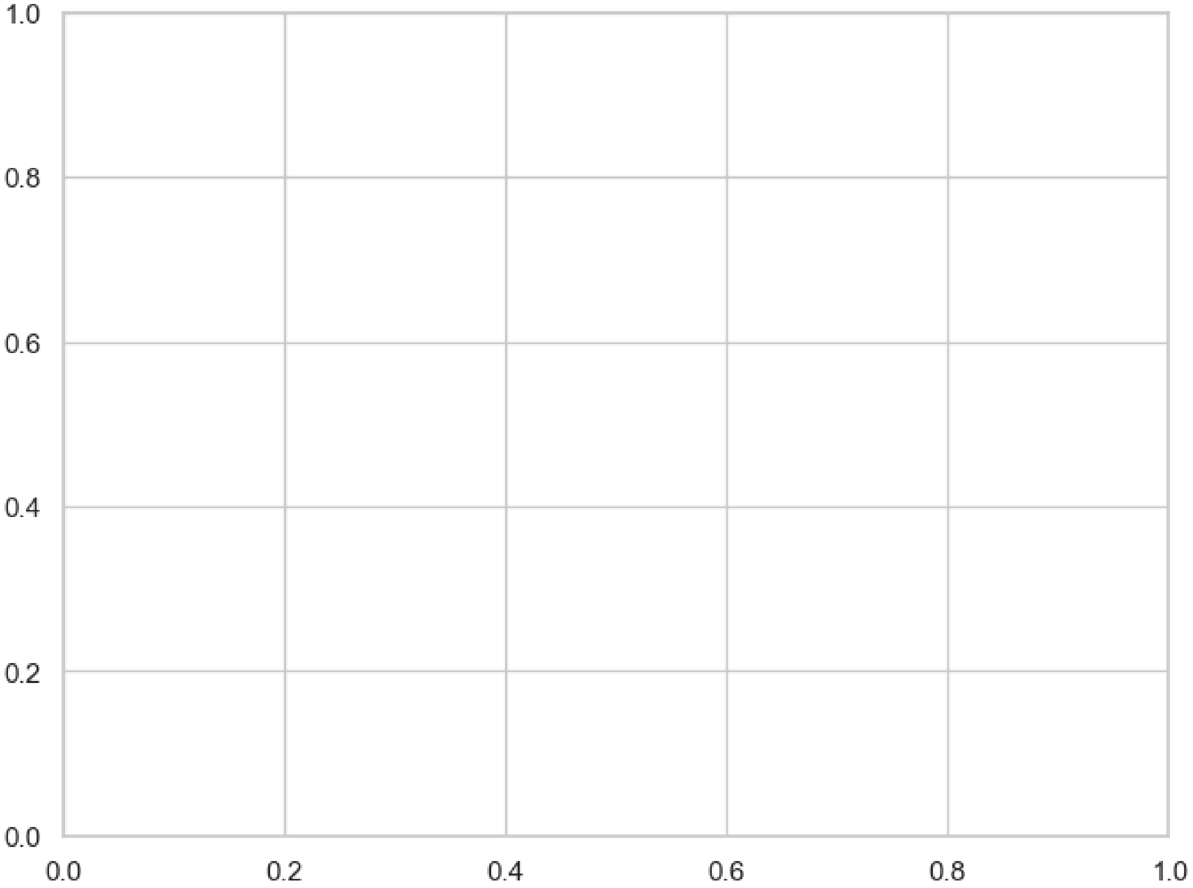
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.lmplot(x='Overall',y='Potential',data=df,hue='Preferred Foot')
plt.show()
```

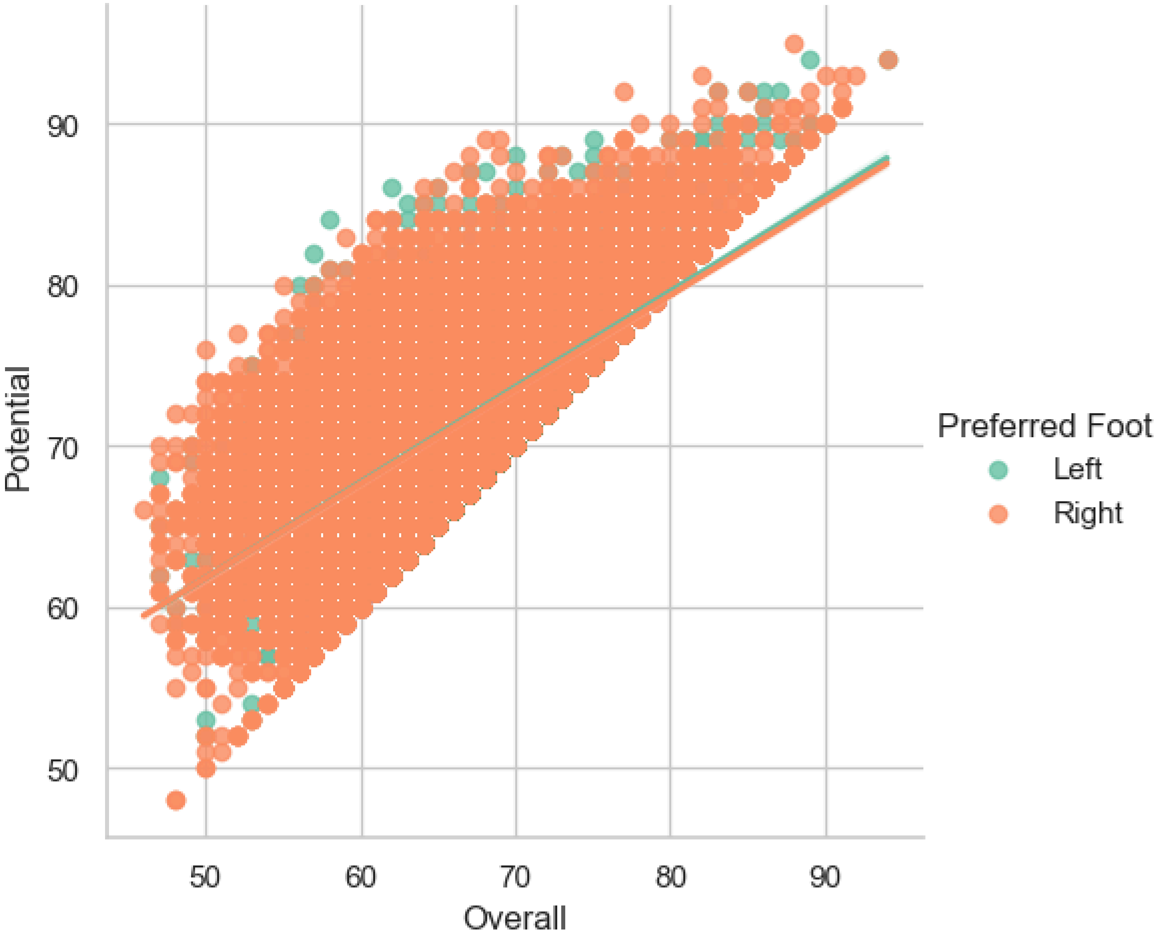




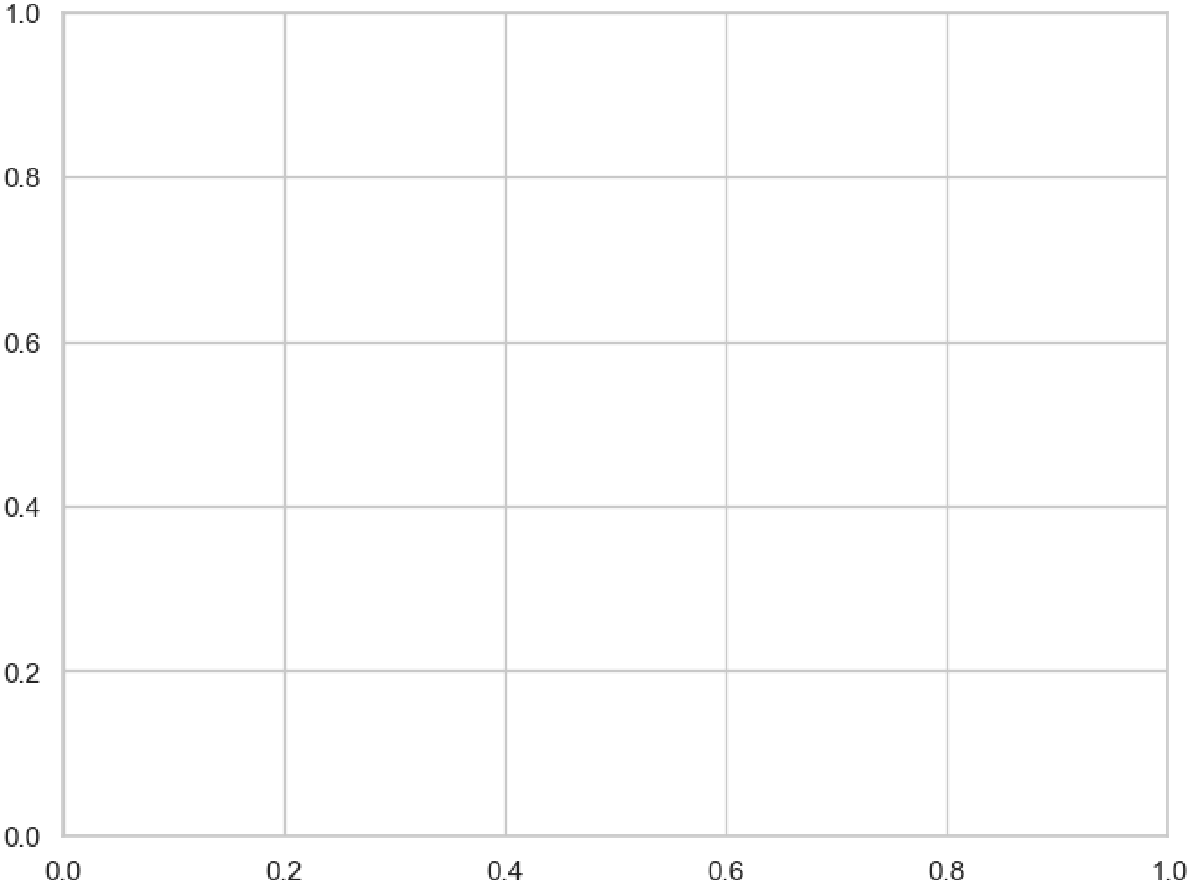
In [94]:

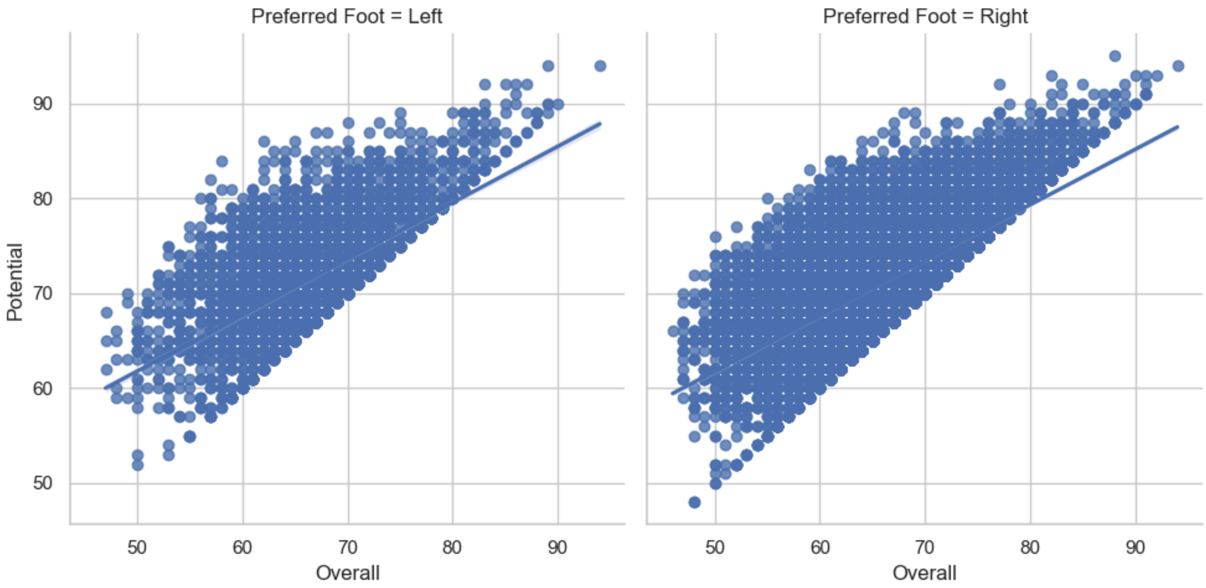
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.lmplot(x='Overall',y='Potential',data=df,hue='Preferred Foot',palette='Set2'
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

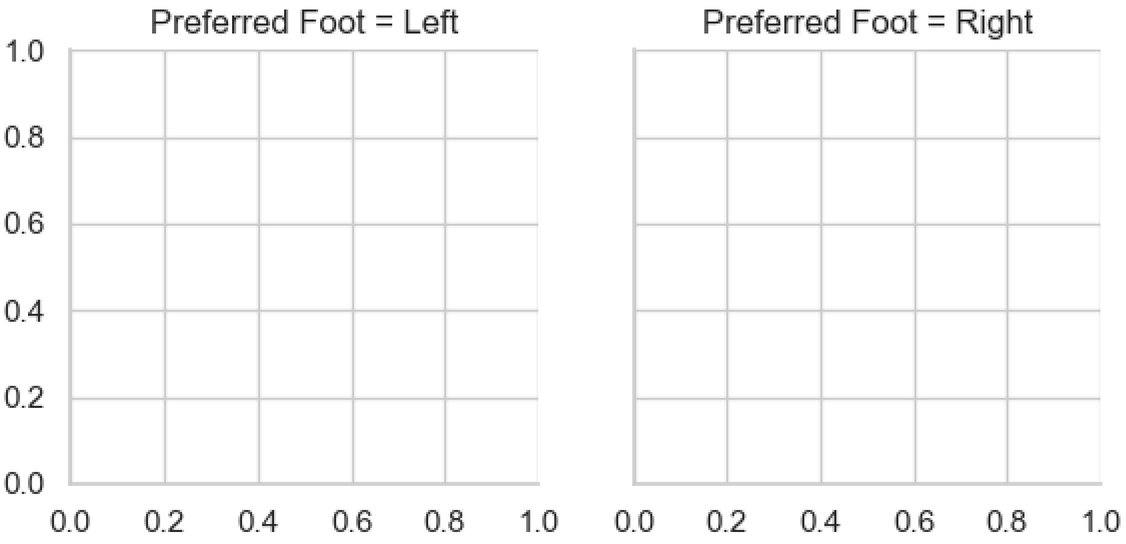

90
80
70
60
50
Potential
50
60
70
80
90
Overall
Preferred Foot
Left
Right

In [95]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.lmplot(x='Overall',y='Potential',data=df,col='Preferred Foot',palette='Set1'
plt.show()
```



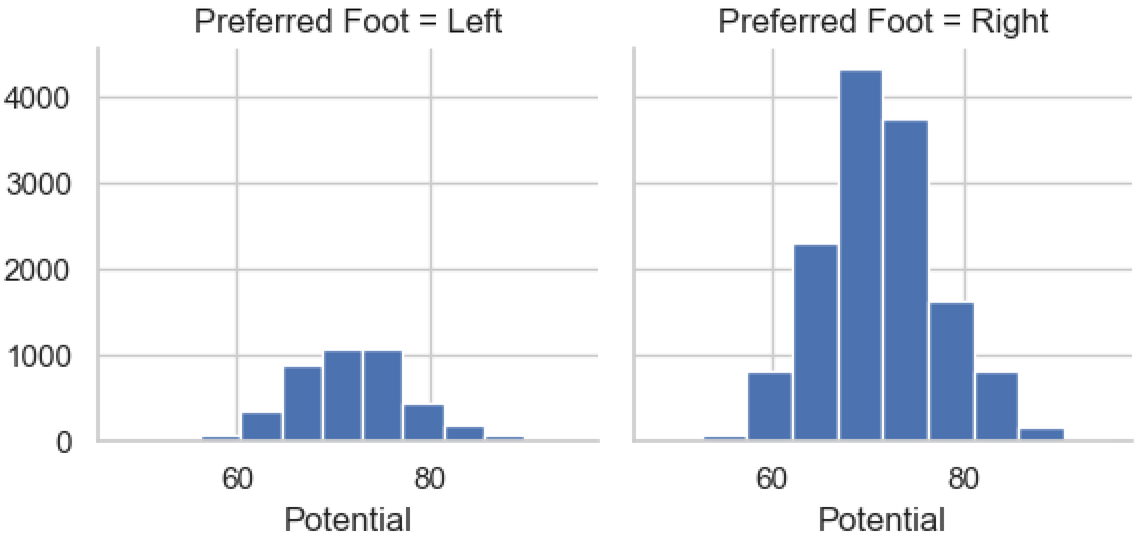


In [96]:
```
h=sns.FacetGrid(df,col='Preferred Foot')
plt.show()
```



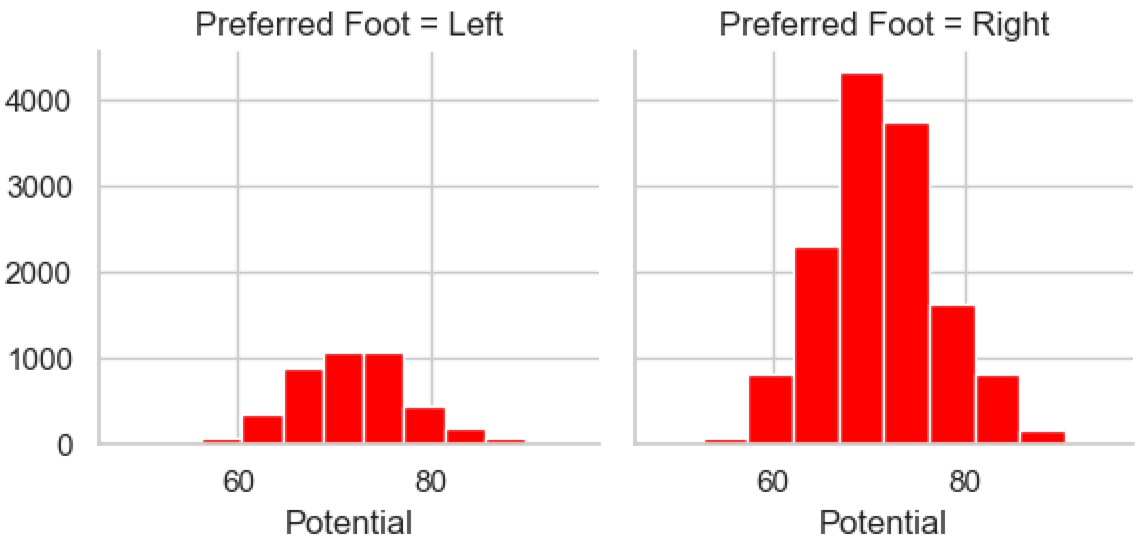
In [97]:

```
h=sns.FacetGrid(df,col='Preferred Foot')
h=h.map(plt.hist,'Potential')
plt.show()
```



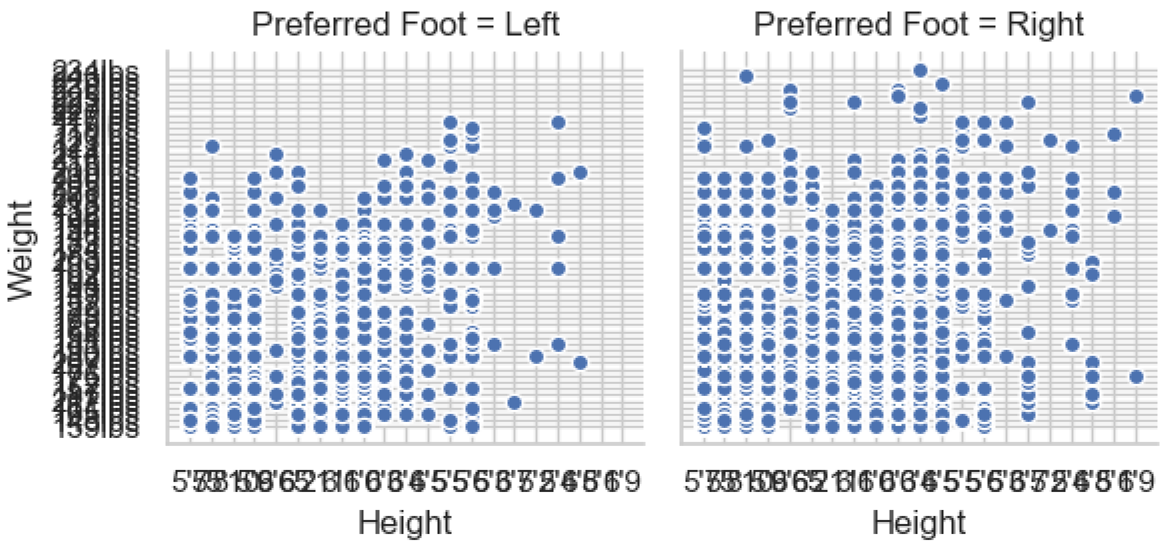
In [98]:

```
h=sns.FacetGrid(df,col='Preferred Foot')
h=h.map(plt.hist,'Potential',color='red')
plt.show()
```



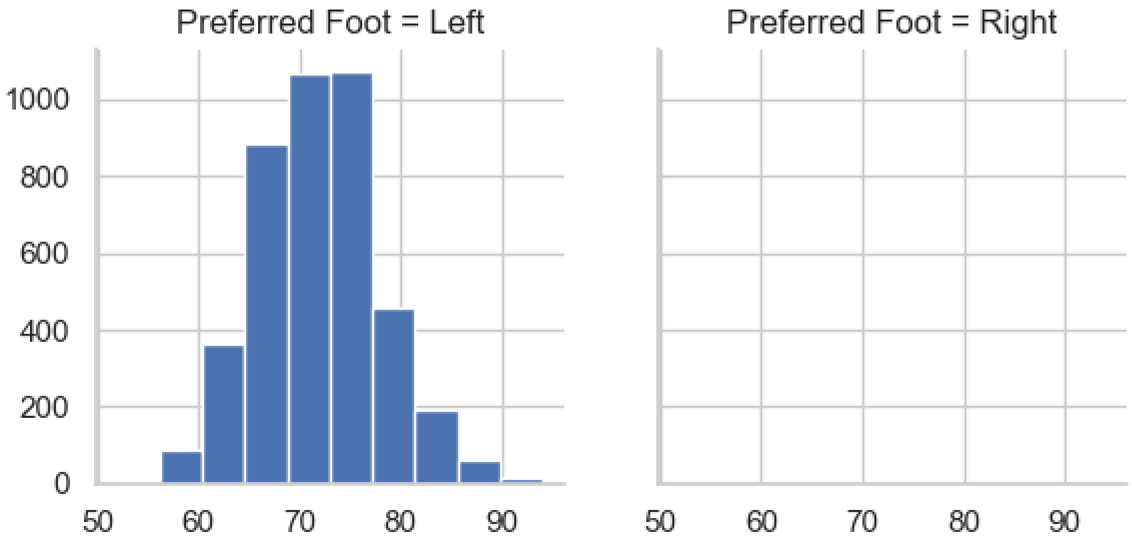
In [99]:

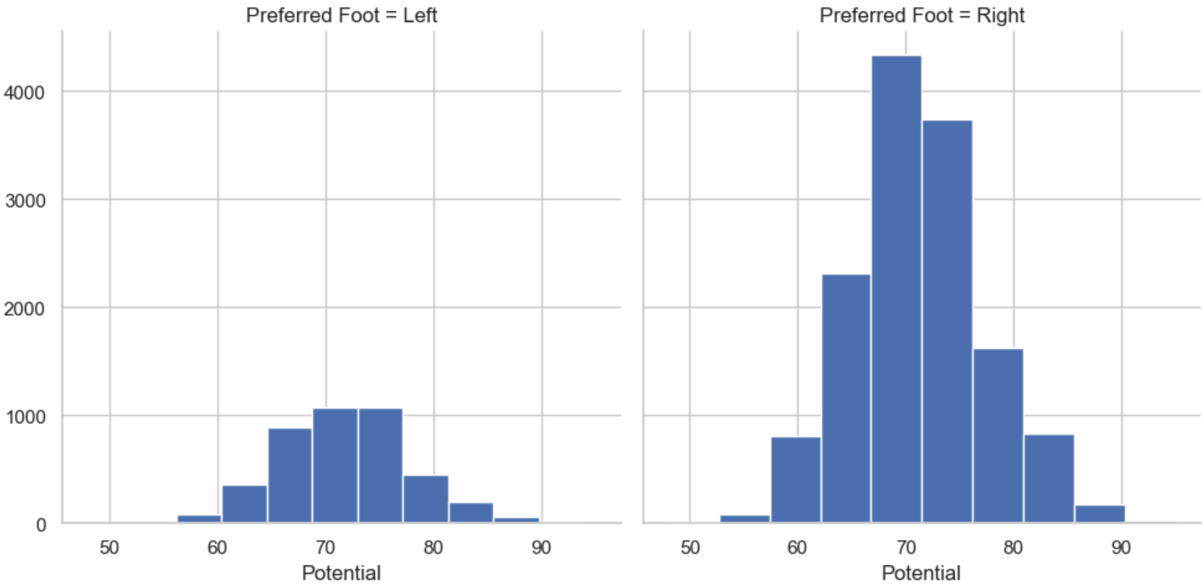
```
h=sns.FacetGrid(df,col='Preferred Foot')
h=(h.map(plt.scatter,'Height','Weight',edgecolor='w').add_legend())
plt.show()
```



In [101…

```
h=sns.FacetGrid(df,col='Preferred Foot',height=5,aspect=1)
h=h.map(plt.hist,'Potential')
plt.show()
```



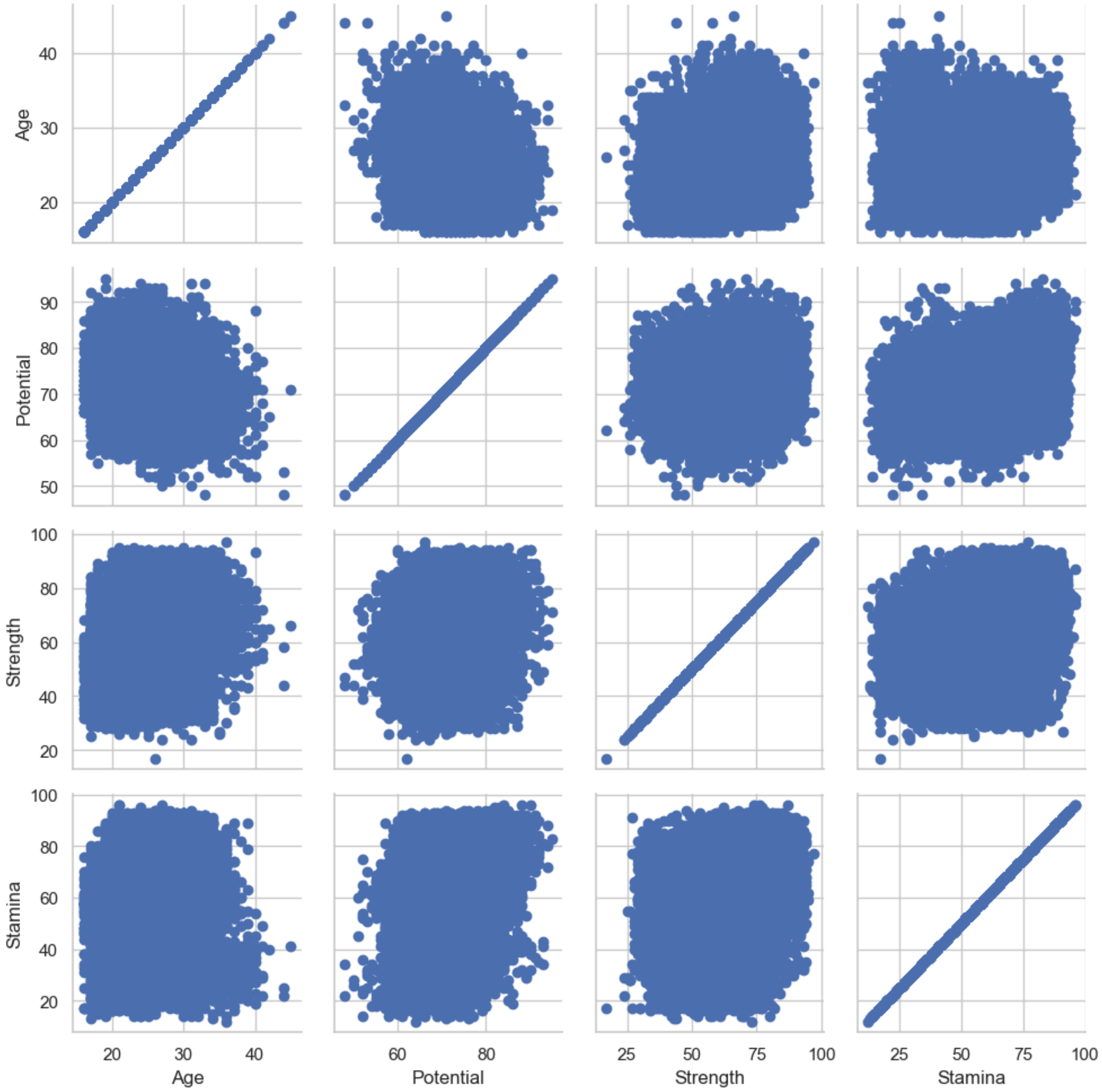


In [102…

```
df_new = df[['Age', 'Potential', 'Strength', 'Stamina', 'Preferred Foot']]
```

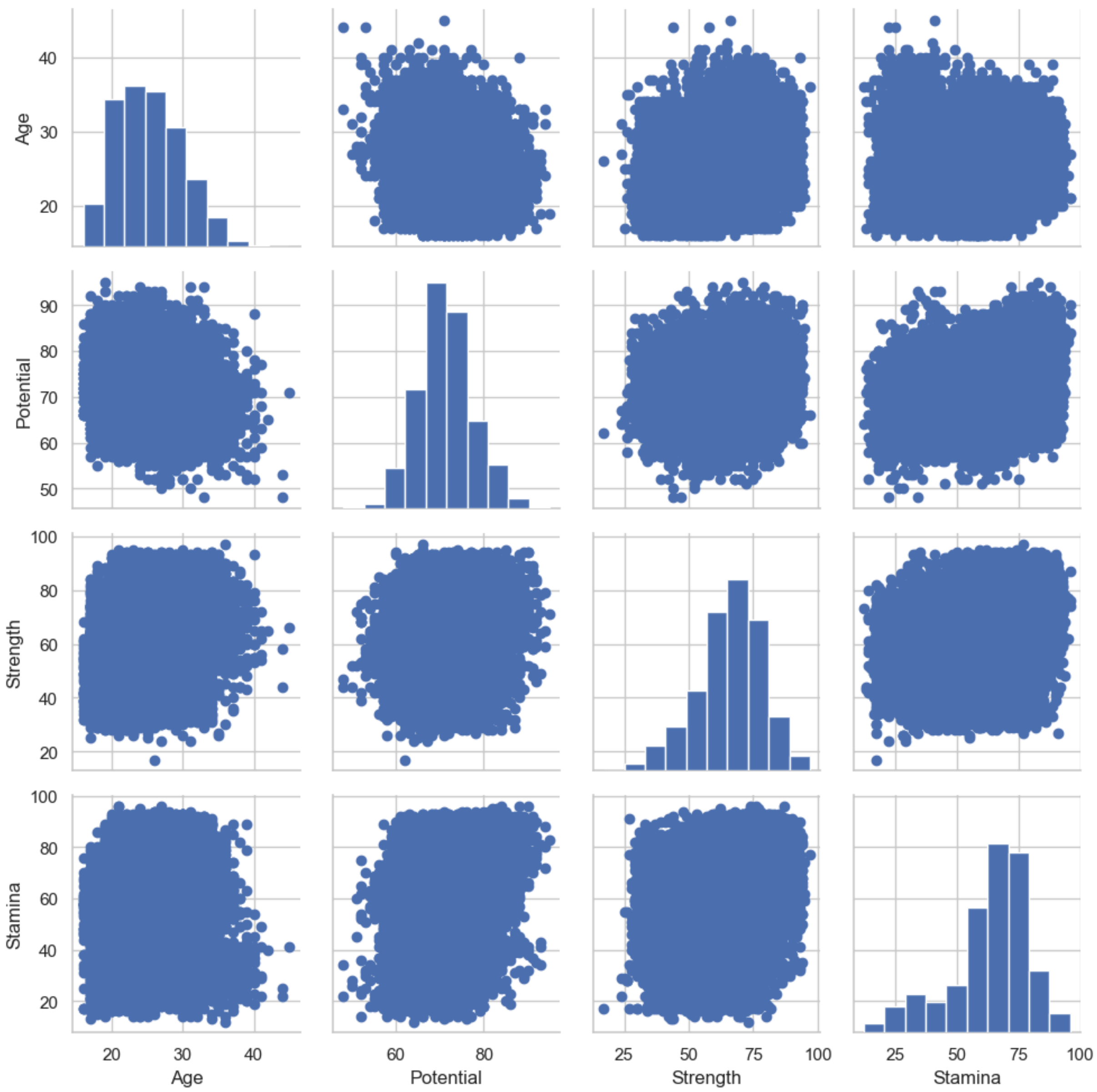
In [103…

```
h=sns.PairGrid(df_new)
h=h.map(plt.scatter)
plt.show()
```



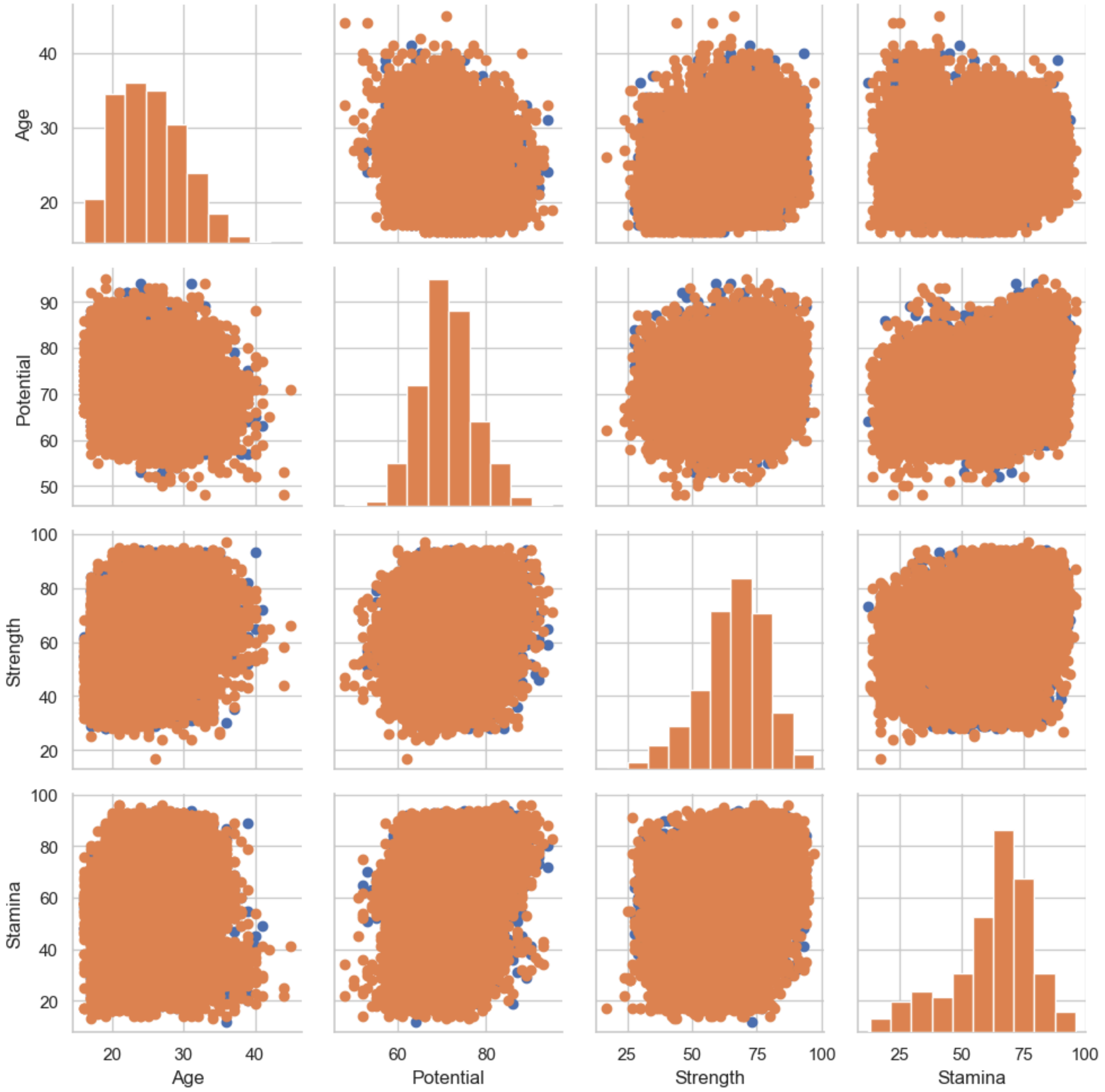
In [106…

```
h=sns.PairGrid(df_new)
h=h.map_diag(plt.hist)
h=h.map_offdiag(plt.scatter)
plt.show()
```



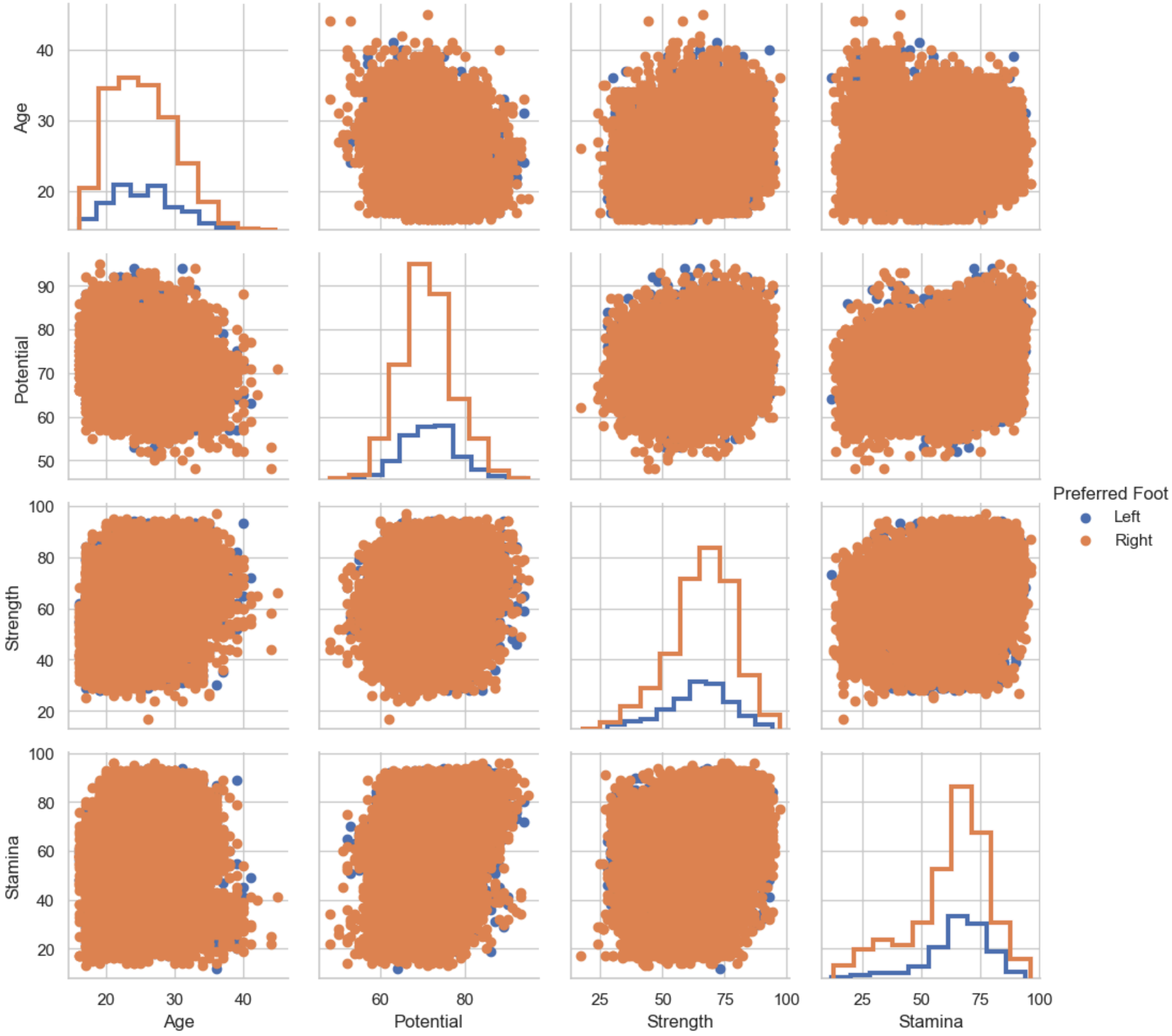
In [107…

```
h=sns.PairGrid(df_new,hue='Preferred Foot')
h=h.map_diag(plt.hist)
h=h.map_offdiag(plt.scatter)
plt.show()
```



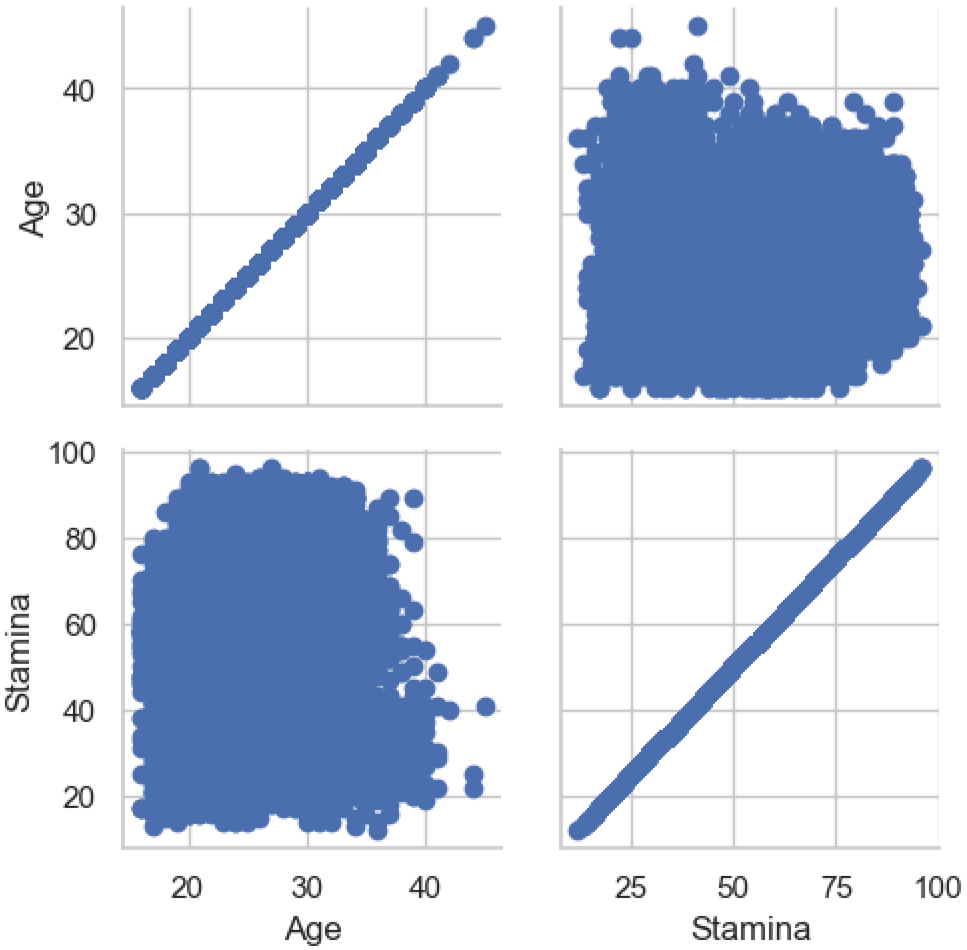
In [108…

```
h=sns.PairGrid(df_new,hue='Preferred Foot')
h=h.map_diag(plt.hist,histtype='step',linewidth=3)
h=h.map_offdiag(plt.scatter)
h=h.add_legend()
plt.show()
```



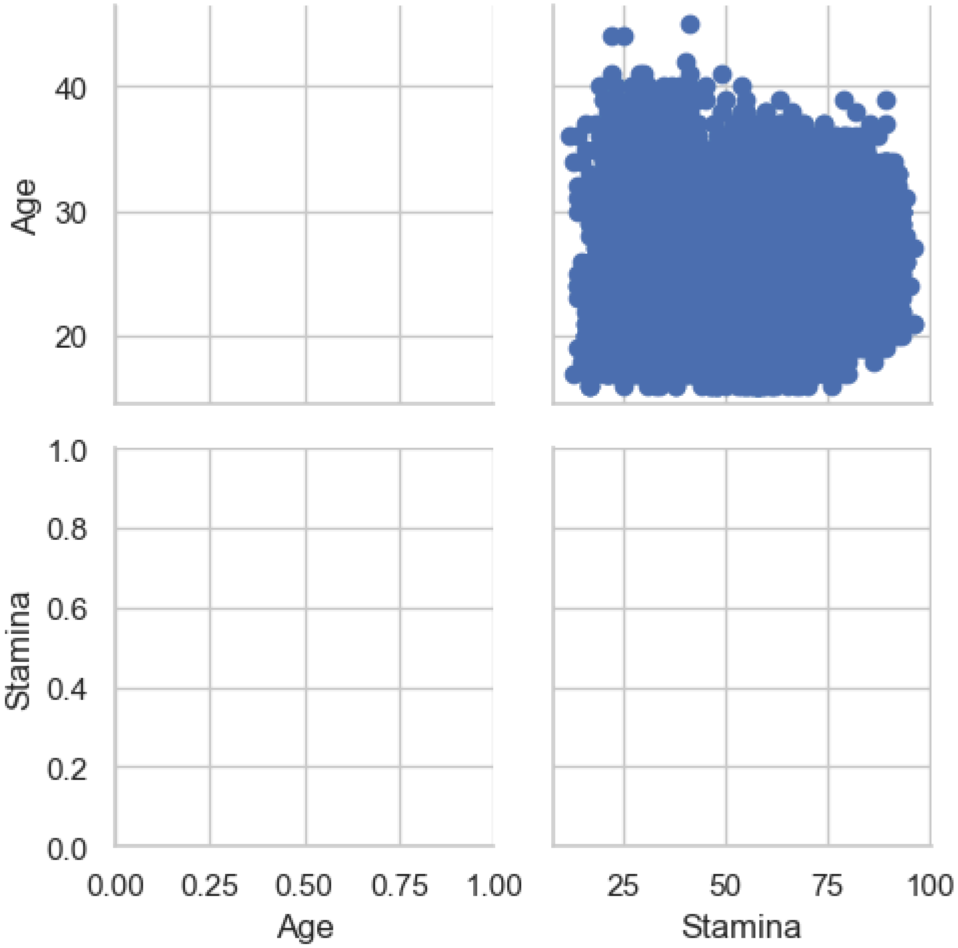
In [112…

```
c=sns.PairGrid(df_new,vars=['Age','Stamina'])
c=c.map(plt.scatter)
plt.show()
```



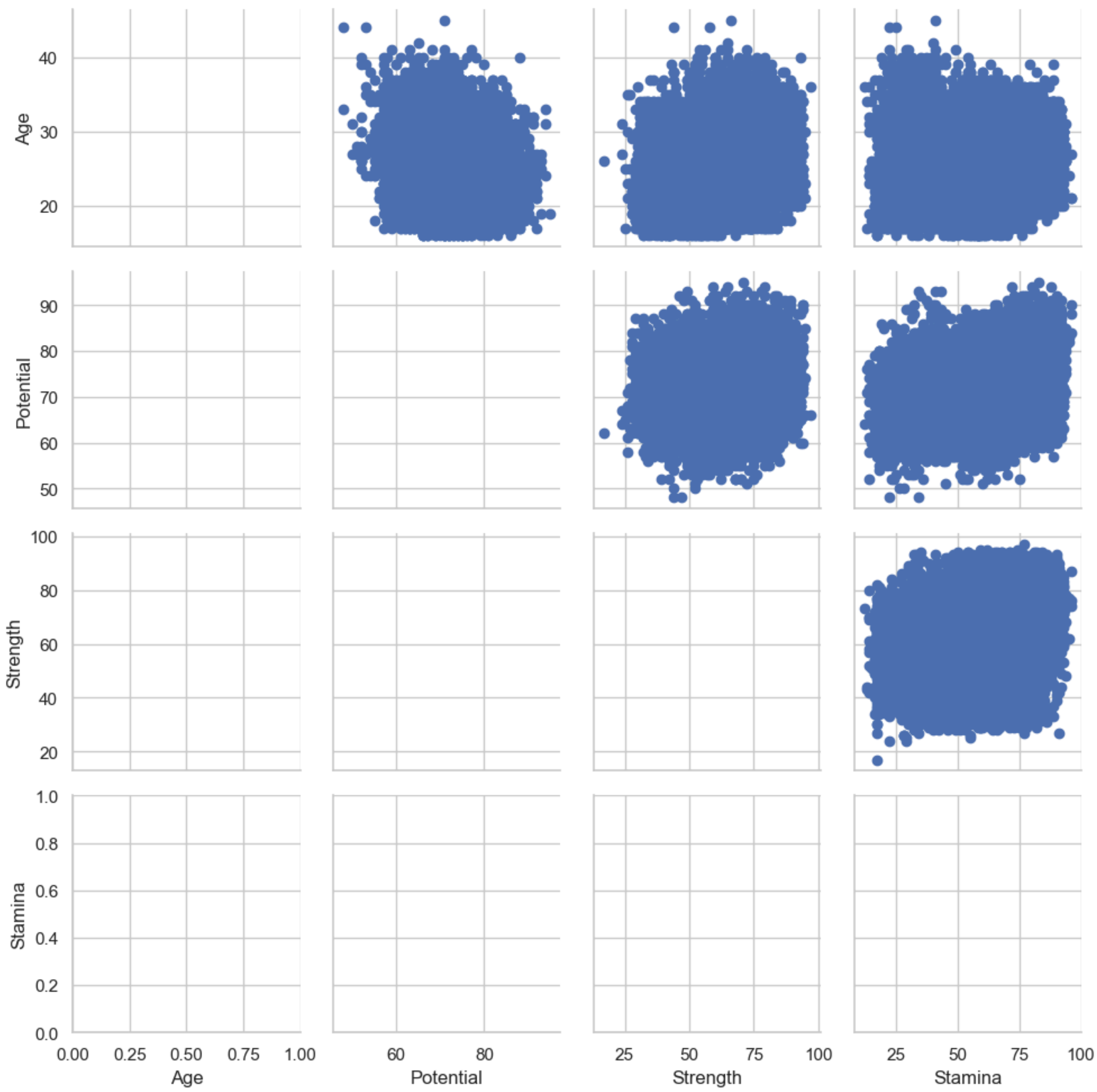
In [113…

```
c=sns.PairGrid(df_new,vars=['Age','Stamina'])
c=c.map_upper(plt.scatter)
plt.show()
```



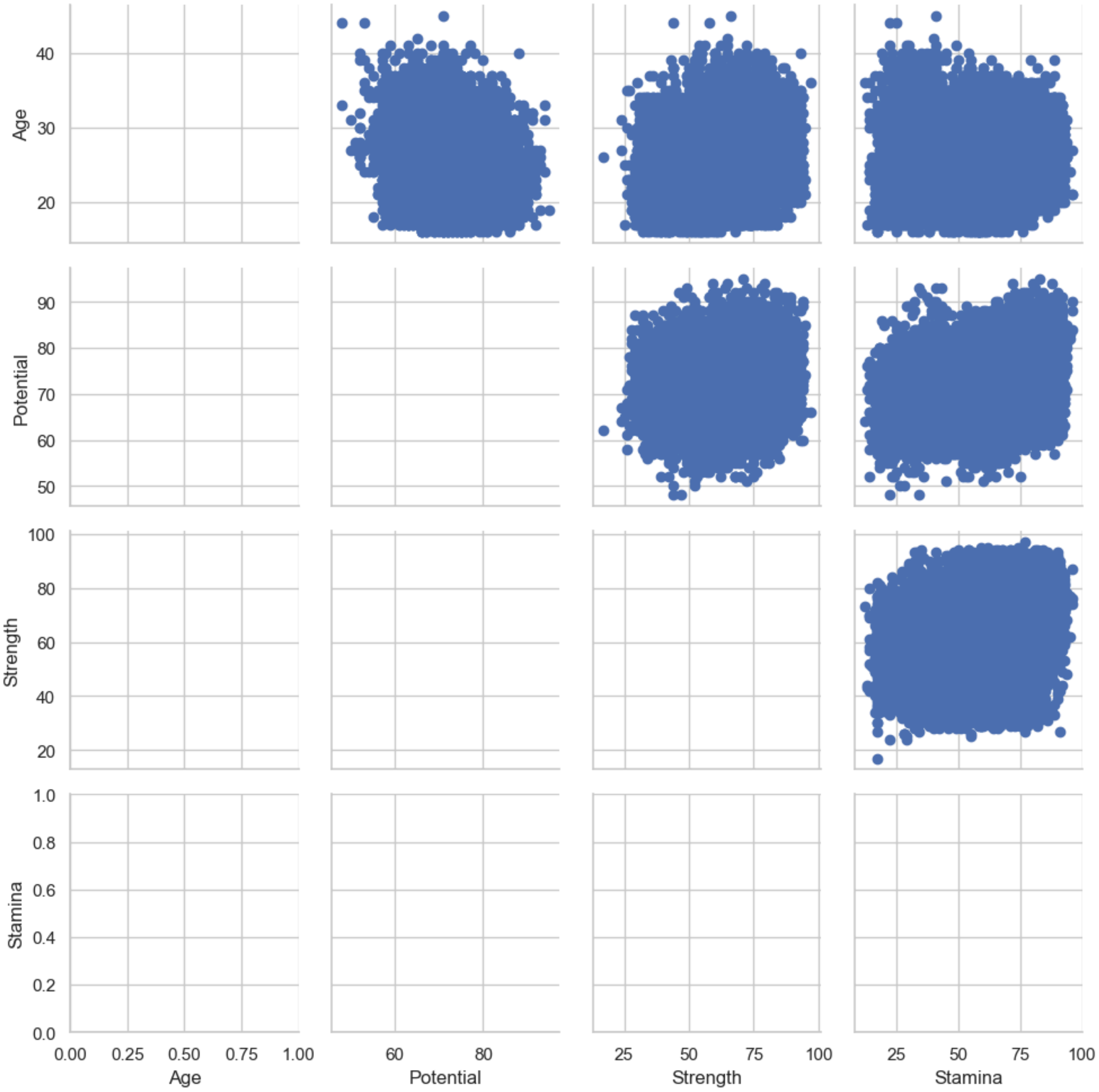
In [114…

```
c=sns.PairGrid(df_new)
c=c.map_upper(plt.scatter)
plt.show()
```

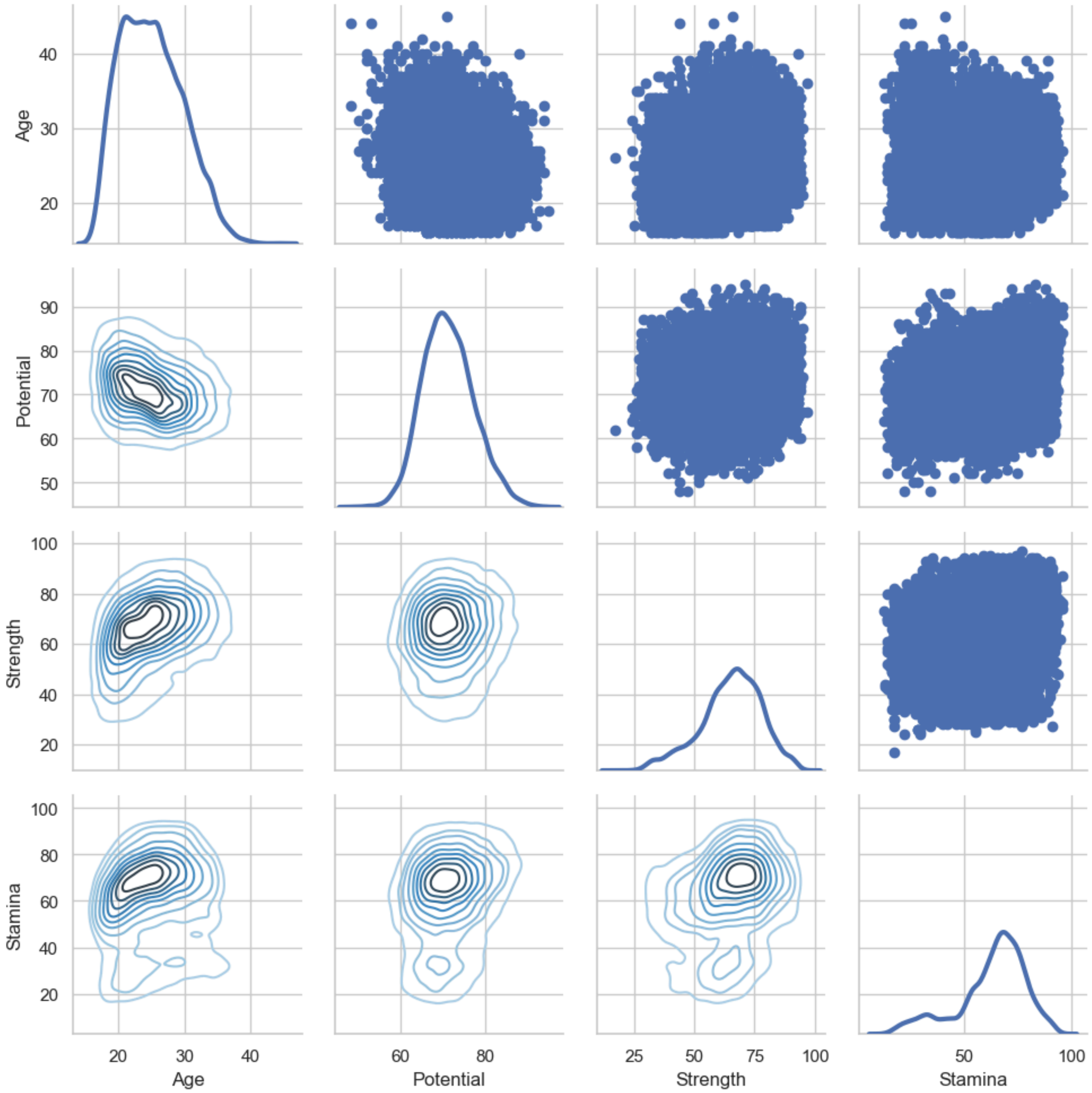


In [116…

```
c=sns.PairGrid(df_new)
c=c.map_upper(plt.scatter)
c=c.map_lower(sns.kdeplot,cmap='Blues_d')
c=c.map_diag(sns.kdeplot,lw=3,legend=False)
plt.show()
```

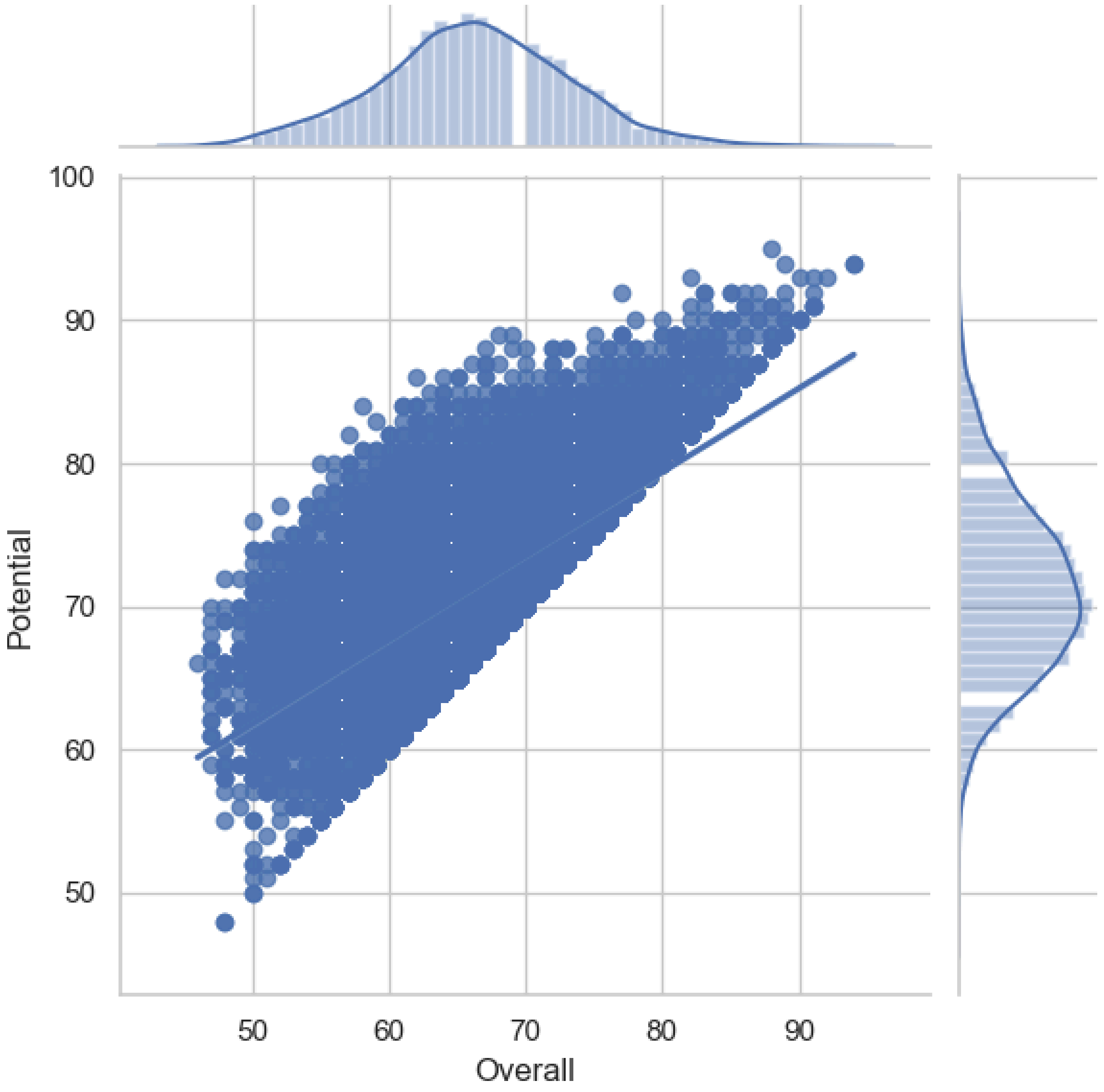


Age
40
30
20
Potential
90
80
70
60
50
Strength
100
80
60
40
20
Stamina
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.25
0.50
0.75
1.00
Age
60
80
Potential
25
50
75
100
Strength
25
50
75
100
Stamina



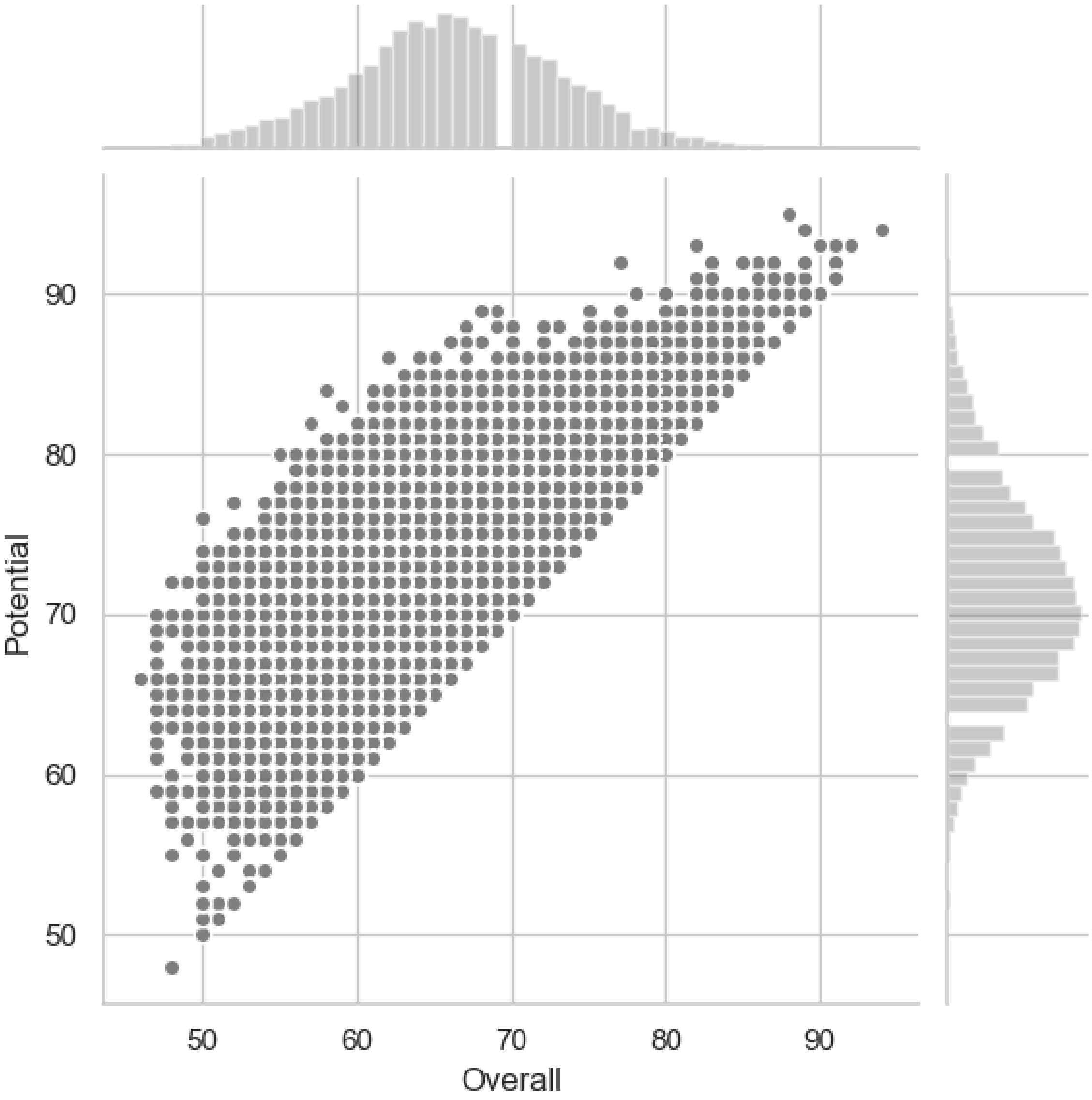
In [117…

```
l=sns.JointGrid(x='Overall',y='Potential',data=df)
l=l.plot(sns.regplot,sns.distplot)
plt.show()
```

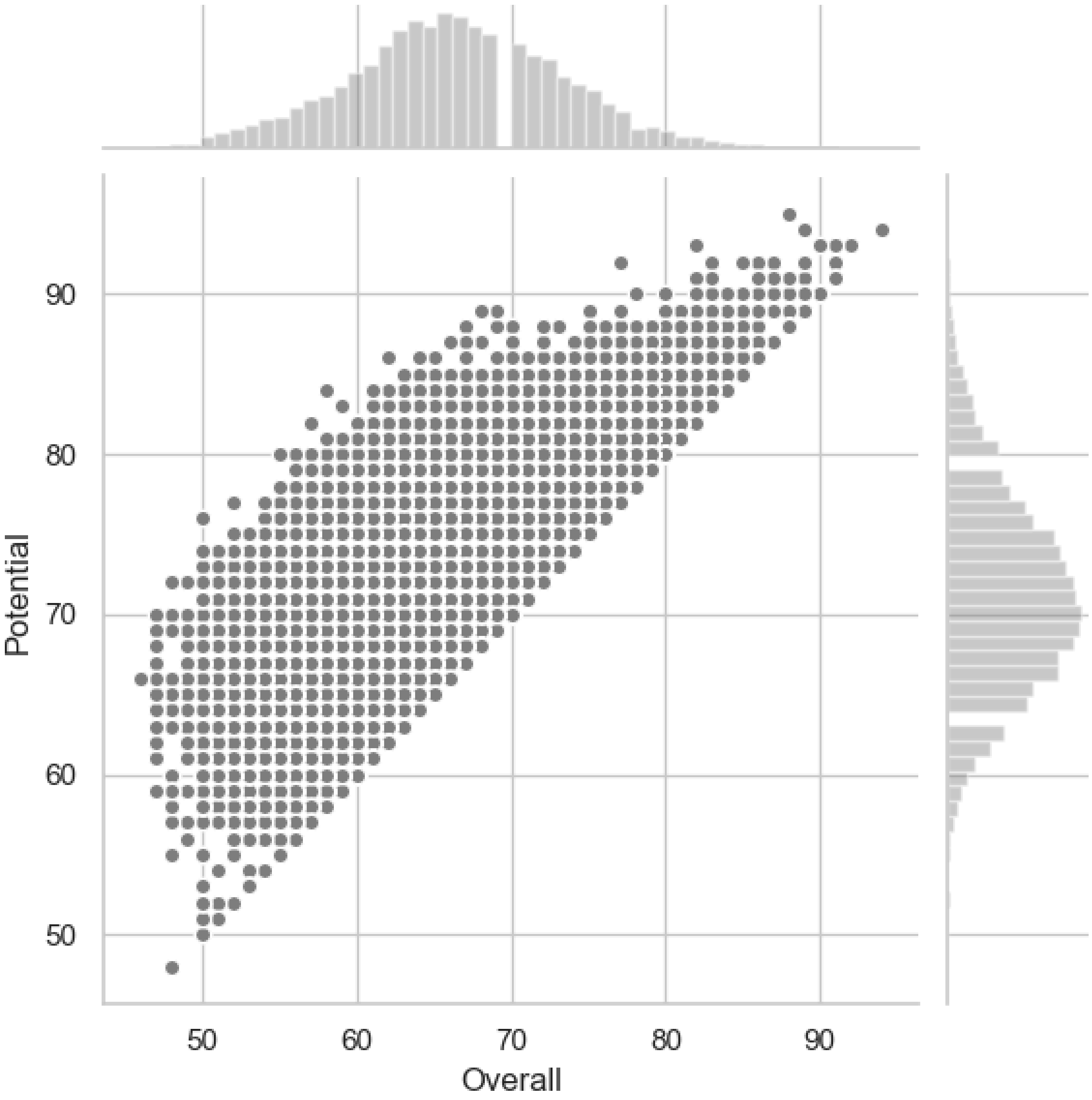


In [119…

```
g = sns.JointGrid(x="Overall", y="Potential", data=df)
g = g.plot_joint(plt.scatter, color=".5", edgecolor="white")
g = g.plot_marginals(sns.distplot, kde=False, color=".5")
plt.show()
```

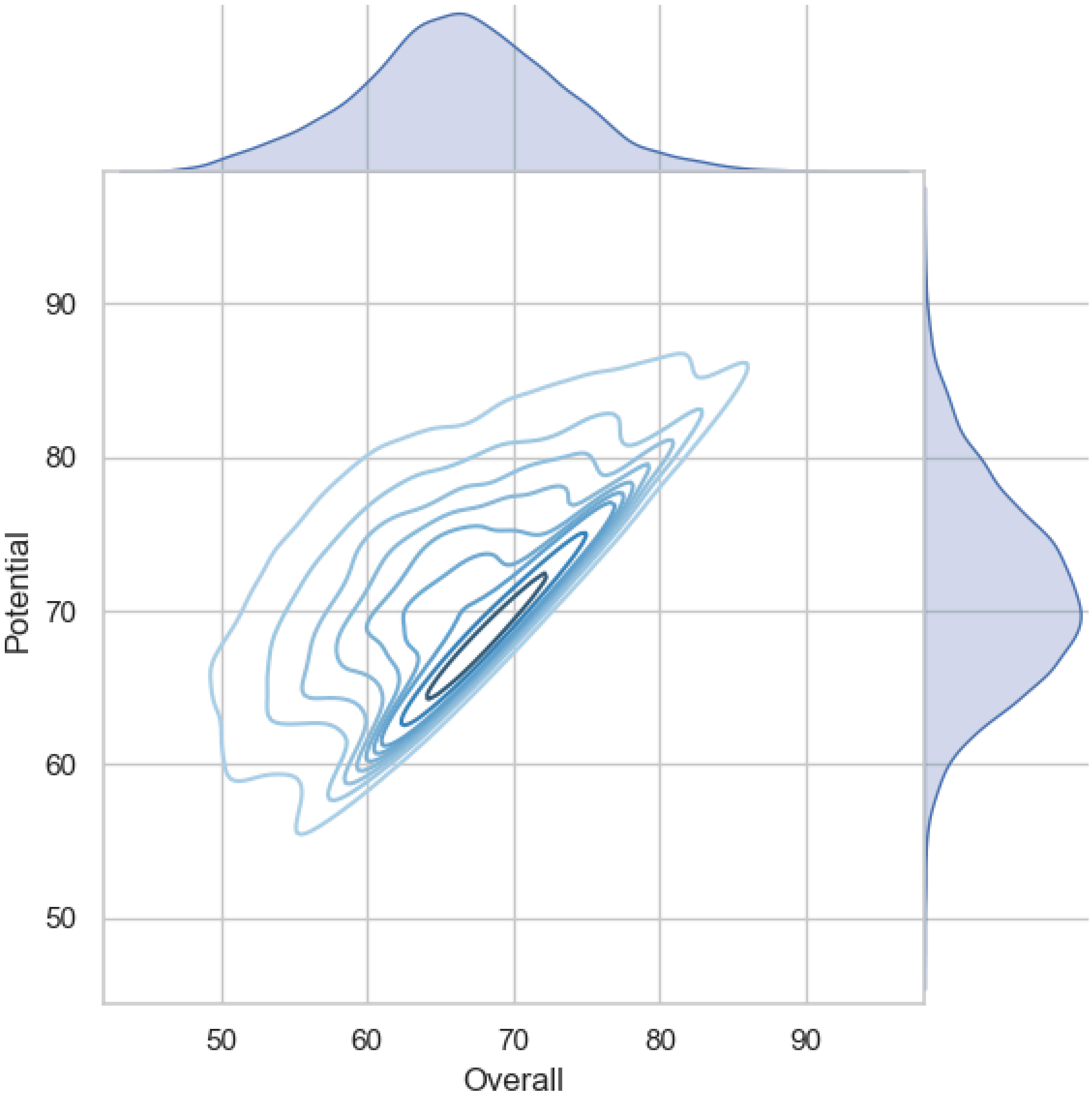


Potential
Overall
90
80
70
60
50
50
60
70
80
90



In [120…

```
g = sns.JointGrid(x="Overall", y="Potential", data=df, space=0)
g = g.plot_joint(sns.kdeplot, cmap="Blues_d")
g = g.plot_marginals(sns.kdeplot, shade=True)
plt.show()
```



```
In [121…  g = sns.JointGrid(x="Overall", y="Potential", data=df, height=5, ratio=2)
          g = g.plot_joint(sns.kdeplot, cmap="Reds_d")
          g = g.plot_marginals(sns.kdeplot, color="r", shade=True)
```

```
In [ ]:
```