FIFA Data Visualization - EDA

In [8]:
```
# import libraries
import numpy as np
import pandas as pd
import seaborn as sns
sns.set(style="whitegrid")
import matplotlib.pyplot as plt
from collections import Counter
%matplotlib inline
```

In [9]:
```
# ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

In [10]:
```
# read Dataset
fifa19 = pd.read_csv('FIFA.csv')
```

In [12]:
```
# Explorary Data Analysis

fifa19.head()
```

Out[12]:

	Unnamed: 0	ID	Name	Age	Photo	Natio
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Arge
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Po
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	Be

5 rows × 89 columns

In [17]:
```
fifa19.drop('Unnamed: 0',axis=1,inplace=True)
```

In [18]:
```
# Preview the dataset
fifa19.head()
```

Out[18]:

	ID	Name	Age	Photo	Nationality	
0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https
1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https
2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https
3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https
4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	http

5 rows × 88 columns

In [19]:
```
# View summary of dataset
fifa19.info()
```

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

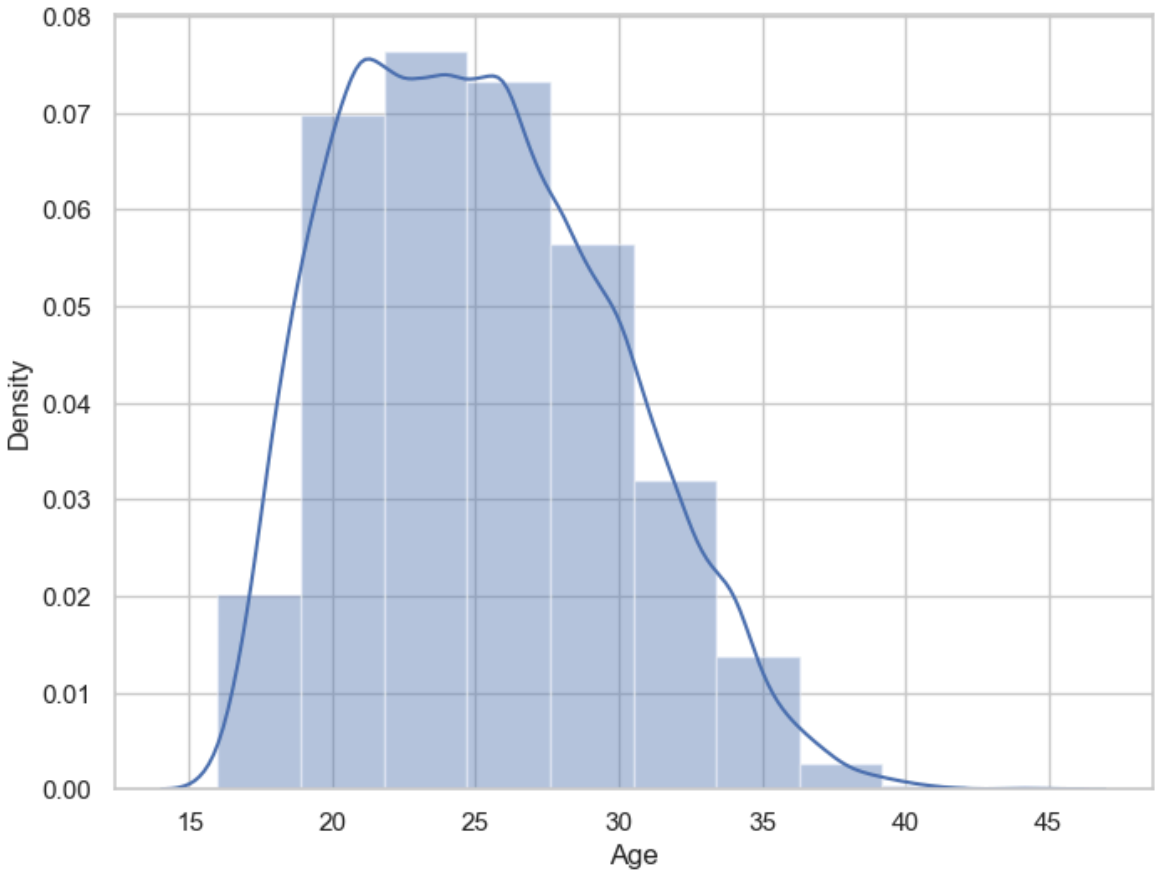
```
 55  HeadingAccuracy           18159 non-null  float64
 56  ShortPassing              18159 non-null  float64
 57  Volleys                   18159 non-null  float64
 58  Dribbling                 18159 non-null  float64
 59  Curve                     18159 non-null  float64
 60  FKAccuracy                18159 non-null  float64
 61  LongPassing               18159 non-null  float64
 62  BallControl               18159 non-null  float64
 63  Acceleration              18159 non-null  float64
 64  SprintSpeed               18159 non-null  float64
 65  Agility                   18159 non-null  float64
 66  Reactions                 18159 non-null  float64
 67  Balance                   18159 non-null  float64
 68  ShotPower                 18159 non-null  float64
 69  Jumping                   18159 non-null  float64
 70  Stamina                   18159 non-null  float64
 71  Strength                  18159 non-null  float64
 72  LongShots                 18159 non-null  float64
 73  Aggression                18159 non-null  float64
 74  Interceptions             18159 non-null  float64
 75  Positioning               18159 non-null  float64
 76  Vision                    18159 non-null  float64
 77  Penalties                 18159 non-null  float64
 78  Composure                 18159 non-null  float64
 79  Marking                   18159 non-null  float64
 80  StandingTackle            18159 non-null  float64
 81  SlidingTackle             18159 non-null  float64
 82  GKDiving                  18159 non-null  float64
 83  GKHandling                18159 non-null  float64
 84  GKKicking                 18159 non-null  float64
 85  GKPositioning             18159 non-null  float64
 86  GKReflexes                18159 non-null  float64
 87  Release Clause            16643 non-null  object
dtypes: float64(38), int64(5), object(45)
memory usage: 12.2+ MB
```

In [20]:
```
fifa19['Body Type'].value_counts()
```

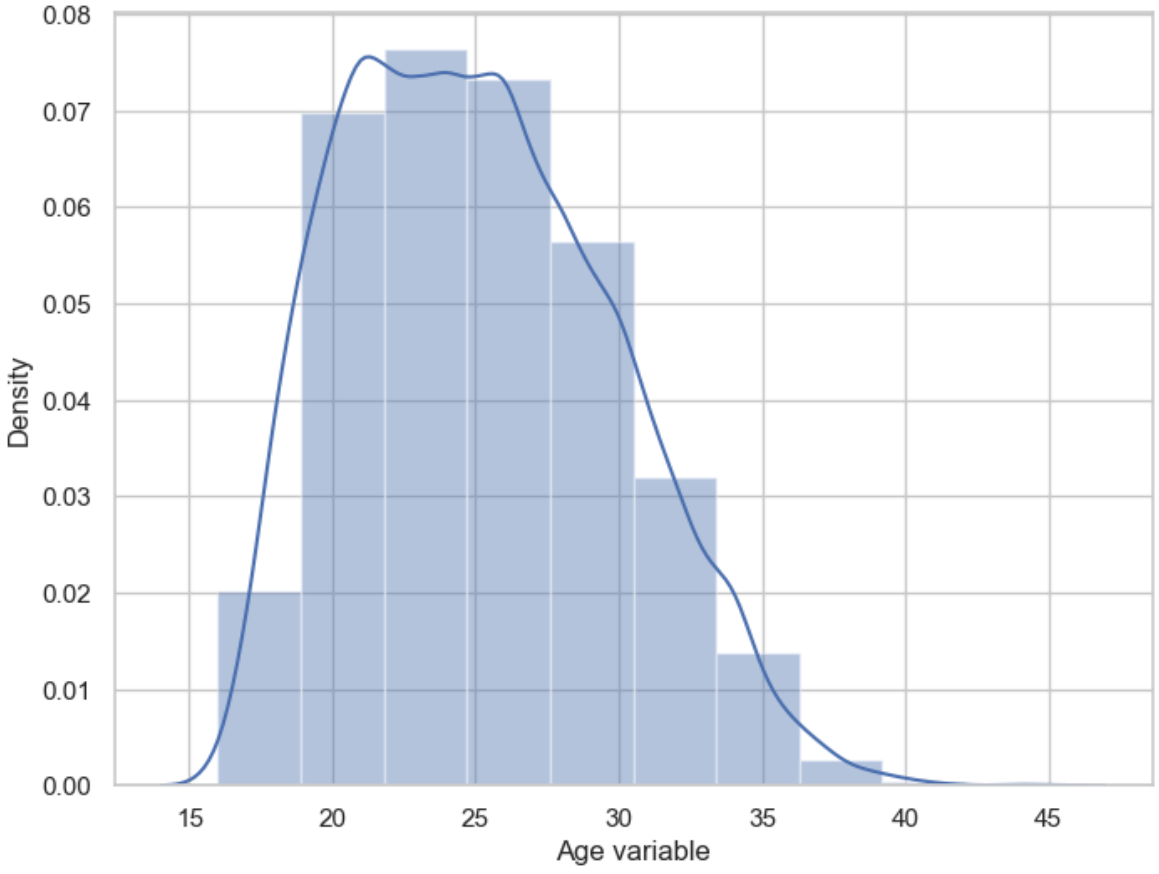
Out[20]:
```
Body Type
Normal                10595
Lean                   6417
Stocky                 1140
Messi                     1
C. Ronaldo                1
Neymar                    1
Courtois                  1
PLAYER_BODY_TYPE_25       1
Shaqiri                   1
Akinfenwa                 1
Name: count, dtype: int64
```

In [ ]:
```
# Explore Age Vaiable
```

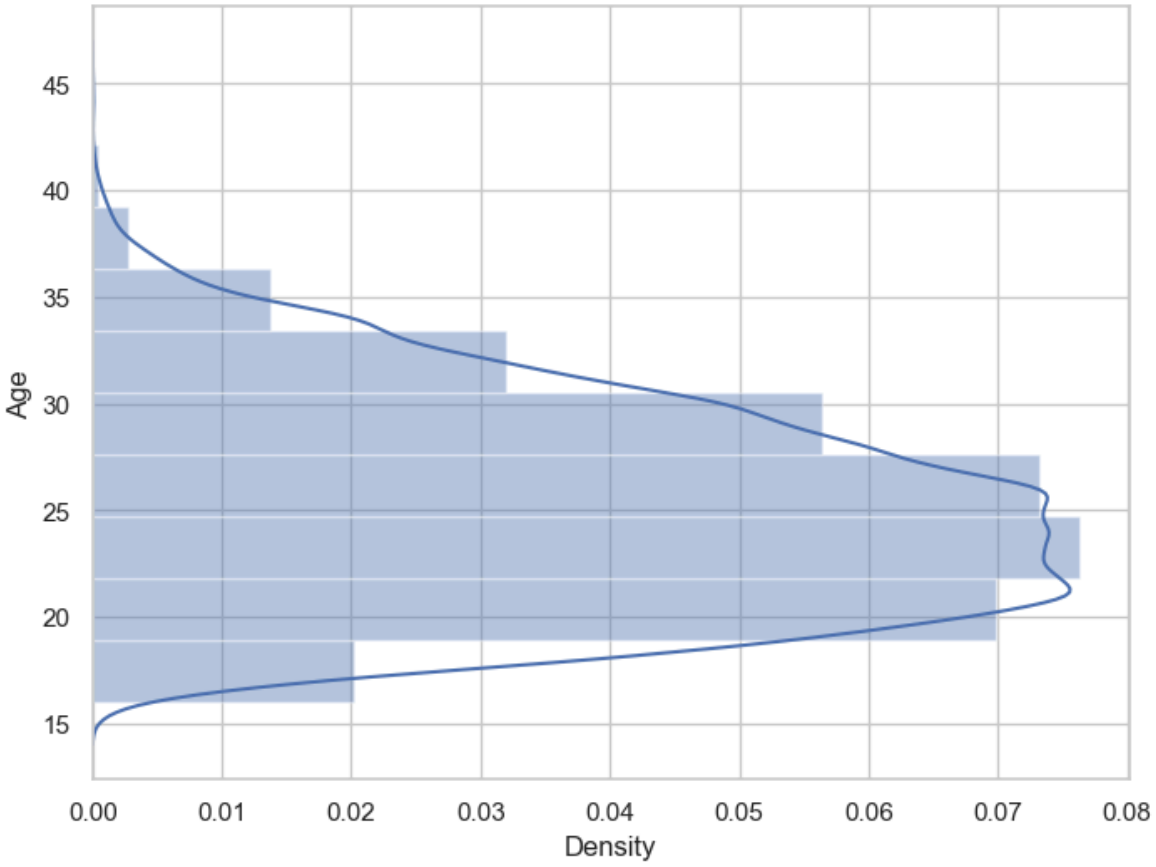
In [21]:
```
# Visualise distribution of Age Variable with Seaborn distplot() function
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
ax = sns.distplot(x, bins=10)
plt.show()
```



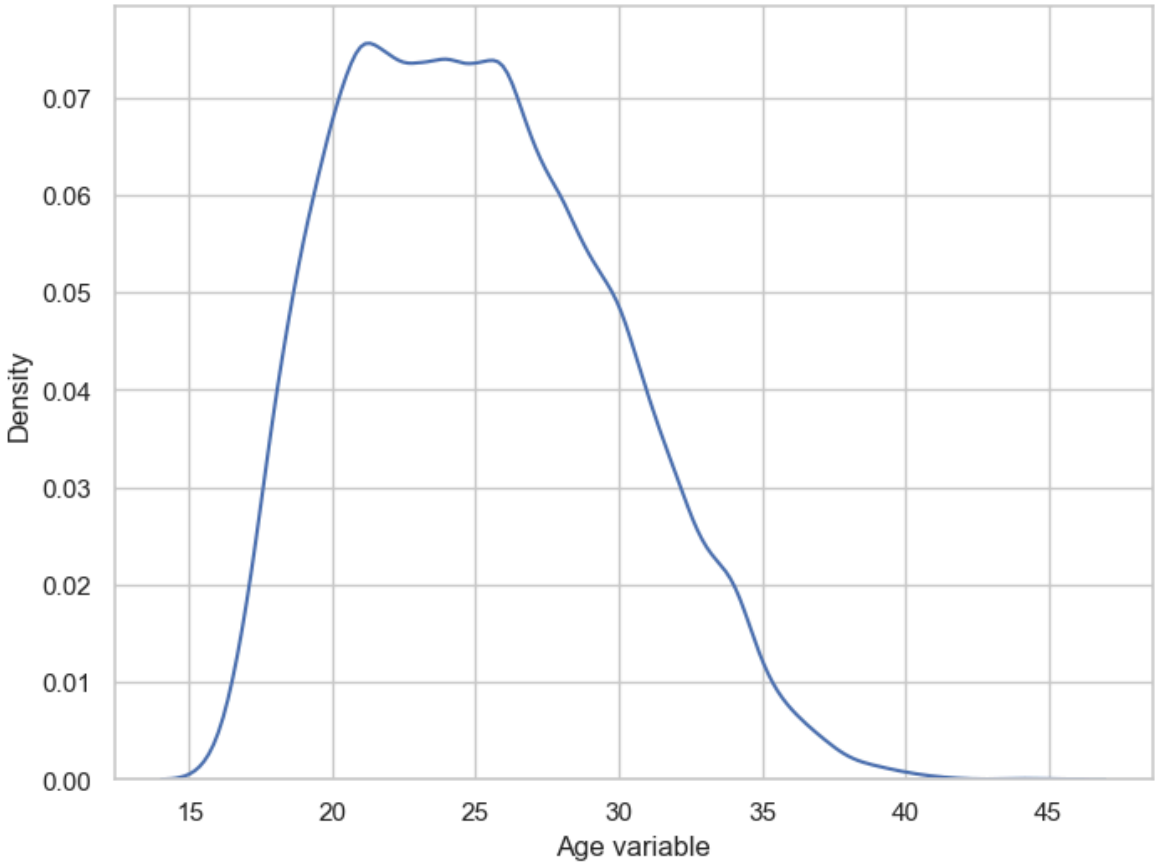
In [22]:
```
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
x = pd.Series(x, name="Age variable")
ax = sns.distplot(x, bins=10)
plt.show()
```



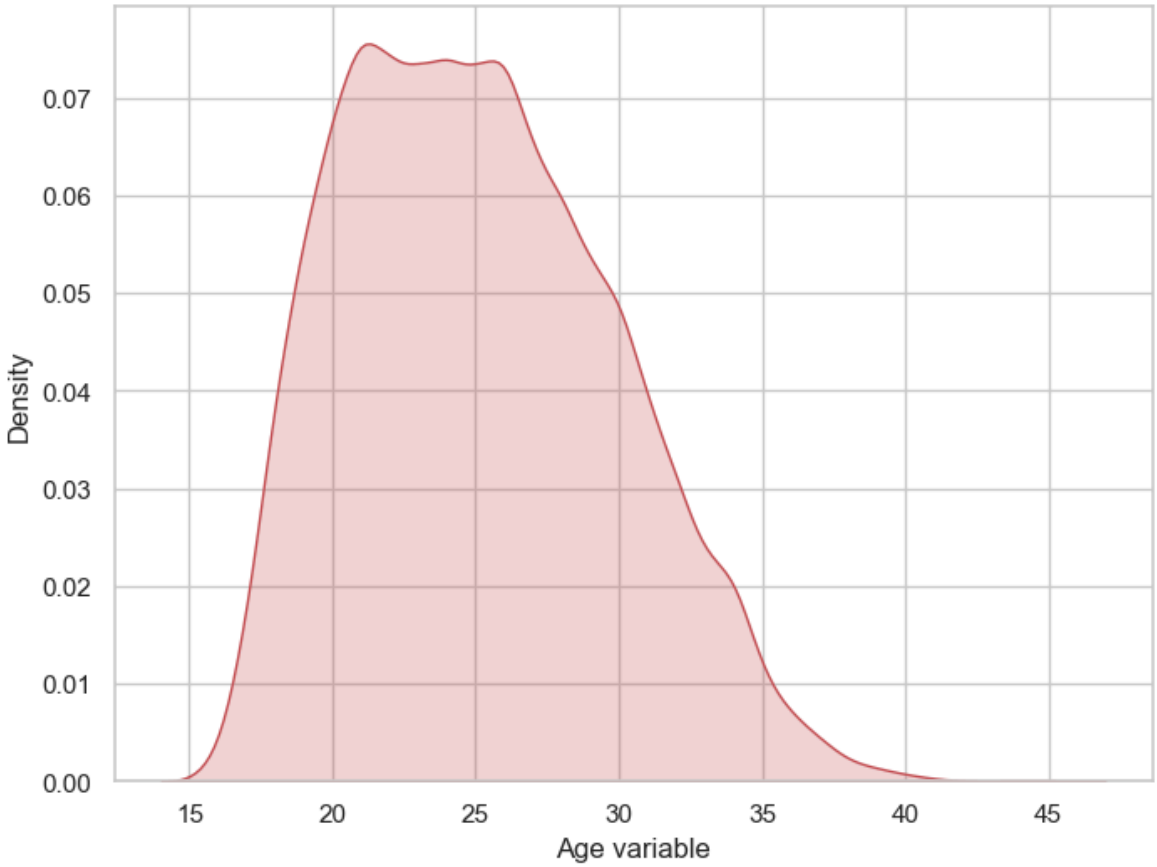
In [23]:
```
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
ax = sns.distplot(x, bins=10, vertical = True)
plt.show()
```



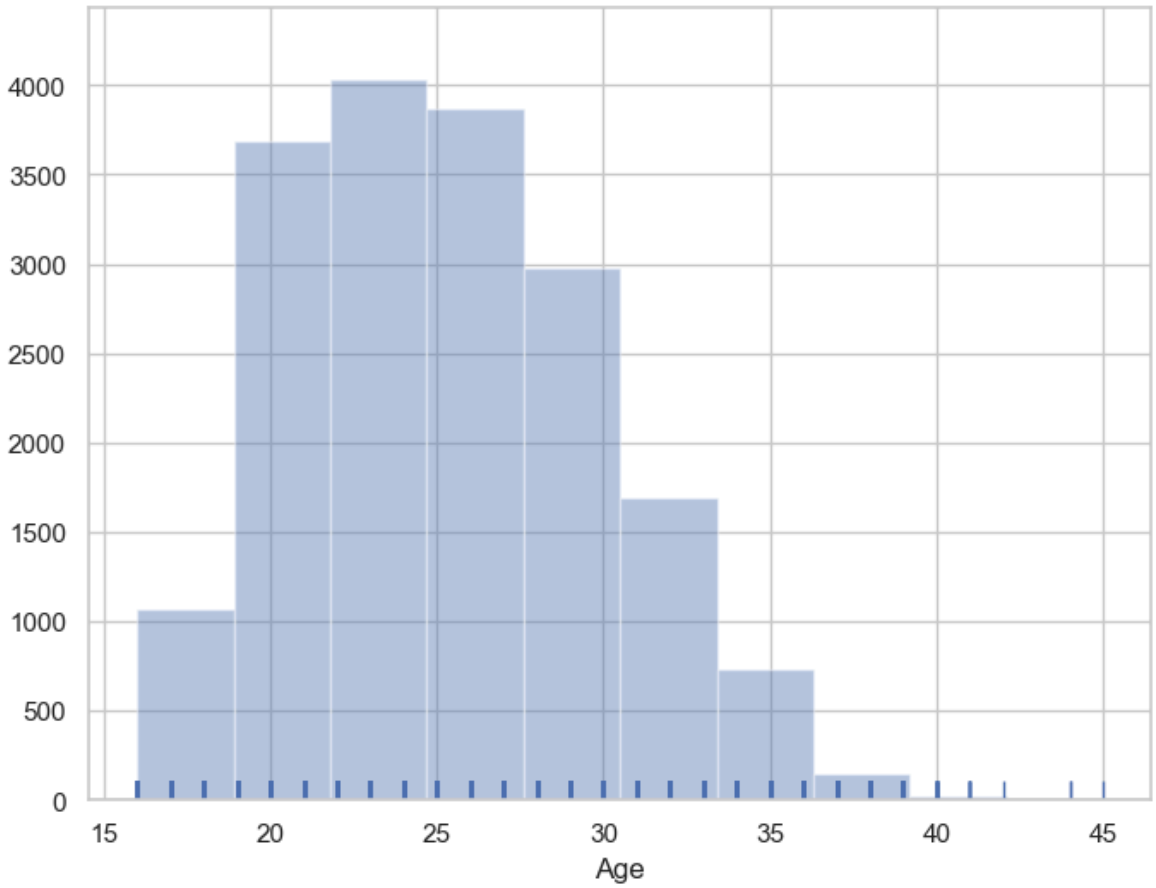
In [24]:
```
# Seaborn KDE plot

f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
x = pd.Series(x, name="Age variable")
ax = sns.kdeplot(x)
plt.show()
```



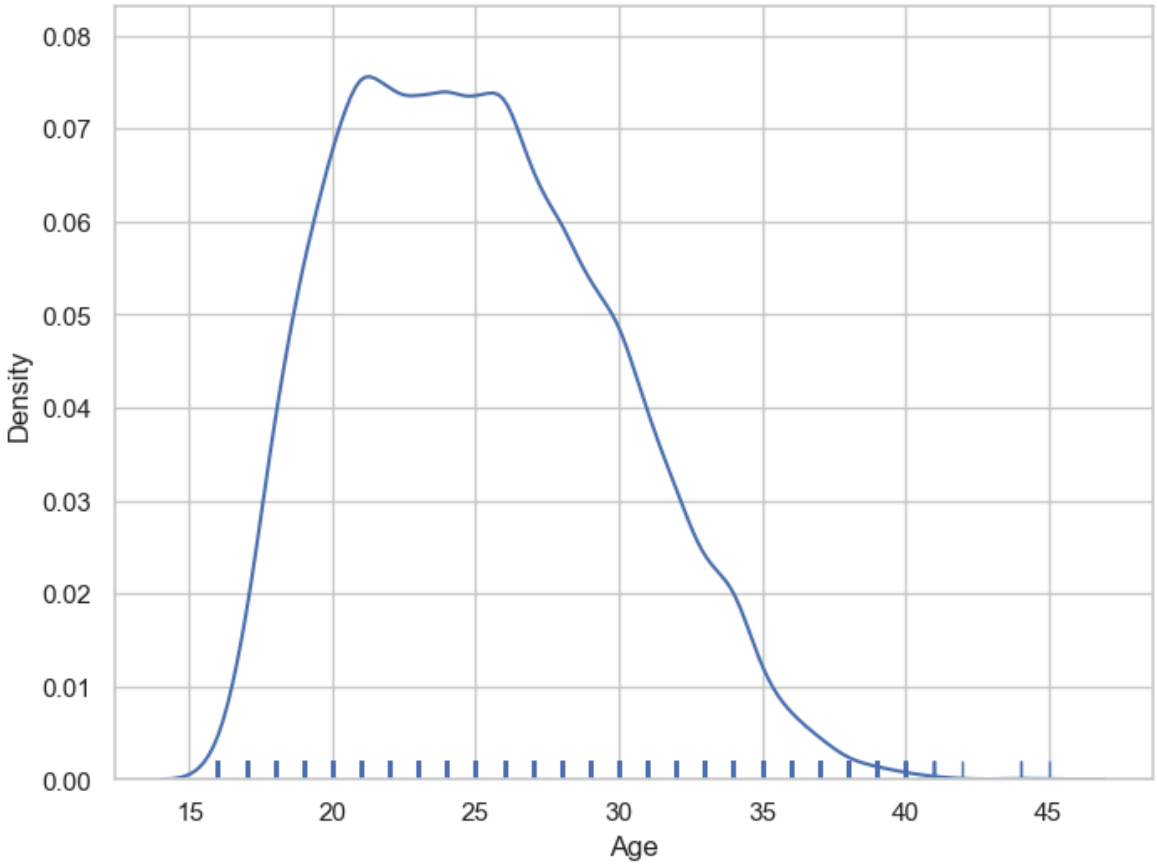
In [25]:
```
# shade under the density curve and use a different color
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
x = pd.Series(x, name="Age variable")
ax = sns.kdeplot(x, shade=True, color='r')
plt.show()
```



In [26]:
```
# Histograms
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
ax = sns.distplot(x, kde=False, rug=True, bins=10)
plt.show()
```



In [27]:
```
# Plot a KDE Plot
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
ax = sns.distplot(x, hist=False, rug=True, bins=10)
plt.show()
```



In [28]:
```
# Explore Preferred Foot Variable
```

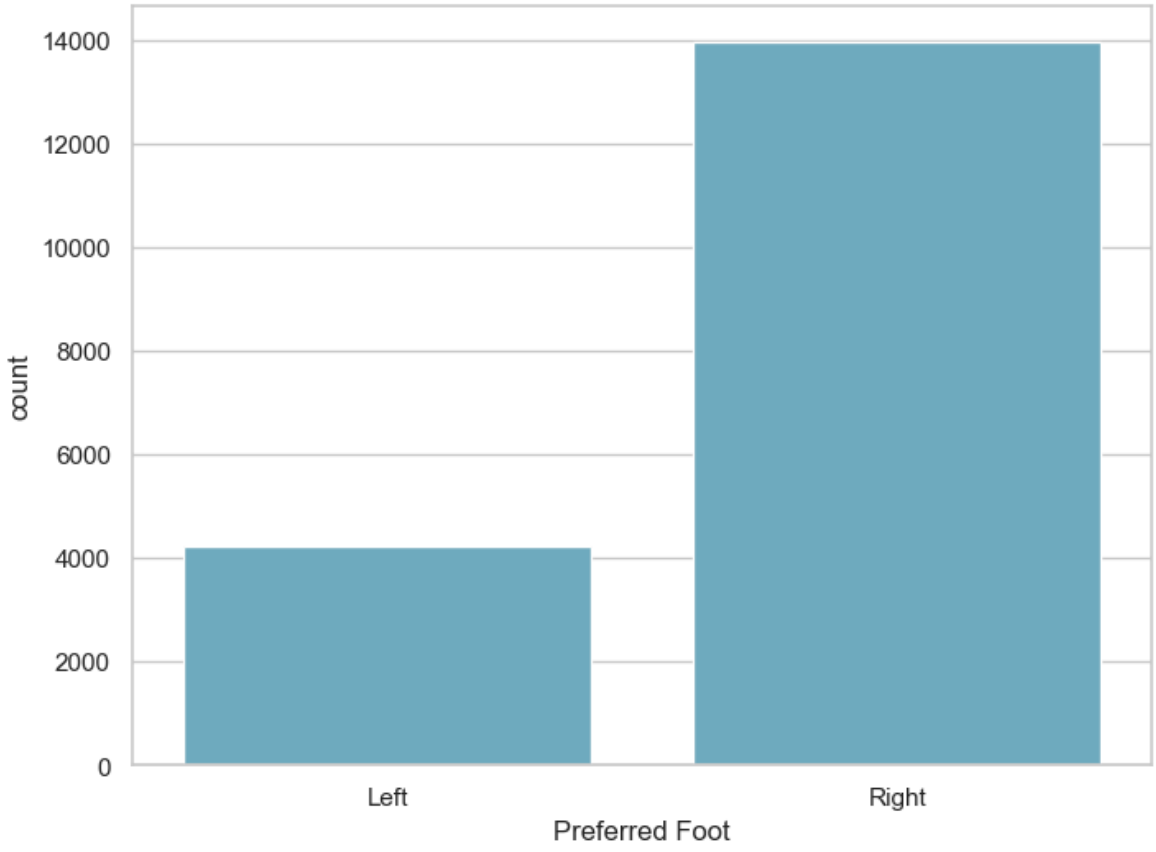
In [30]:
```
# Check number of unique values in Preferred Foot Variable
fifa19['Preferred Foot'].nunique()
```

Out[30]:
```
2
```

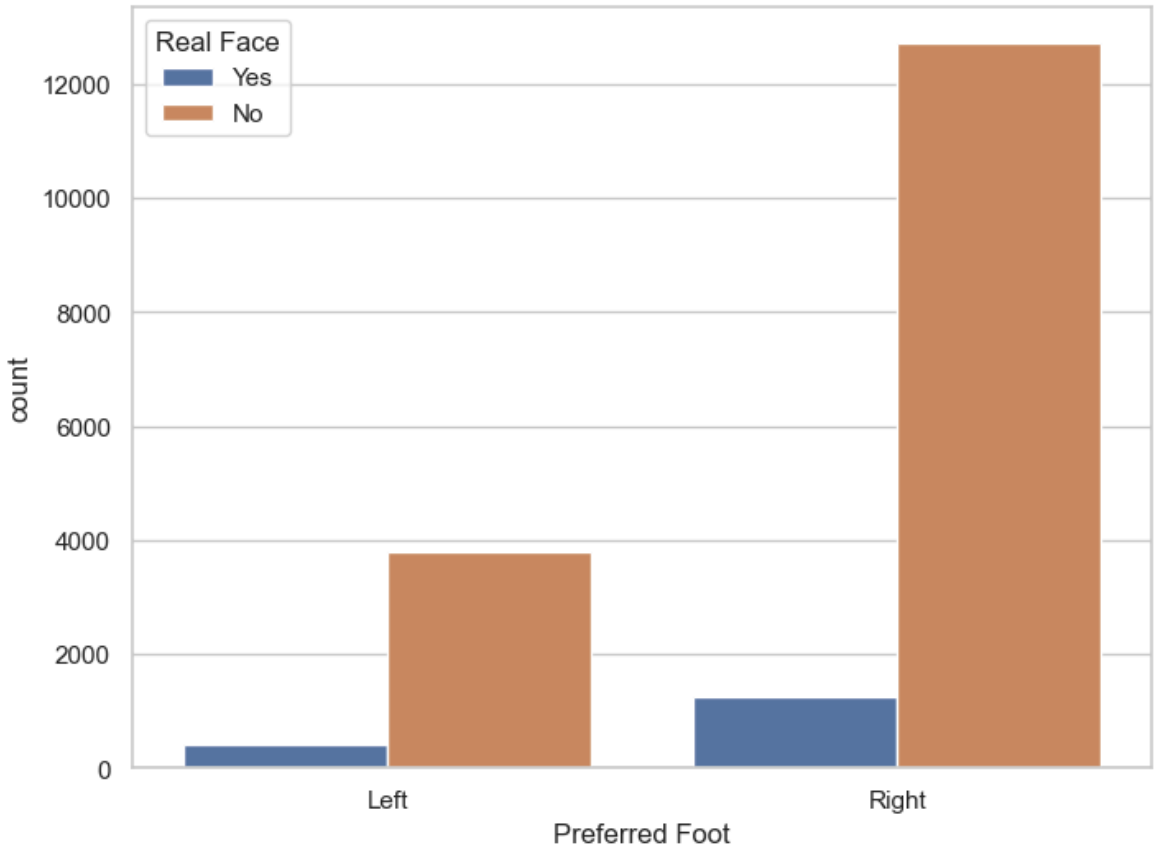
In [31]:
```
# Check the distribution of values in Preferred Foot Variable
fifa19['Preferred Foot'].value_counts()
```

Out[31]:
```
Preferred Foot
Right    13948
Left      4211
Name: count, dtype: int64
```

In [32]:
```
# Visualise distribution of values with Seaborn countplot() function

f, ax = plt.subplots(figsize=(8, 6))
sns.countplot(x="Preferred Foot", data=fifa19, color="c")
plt.show()
```

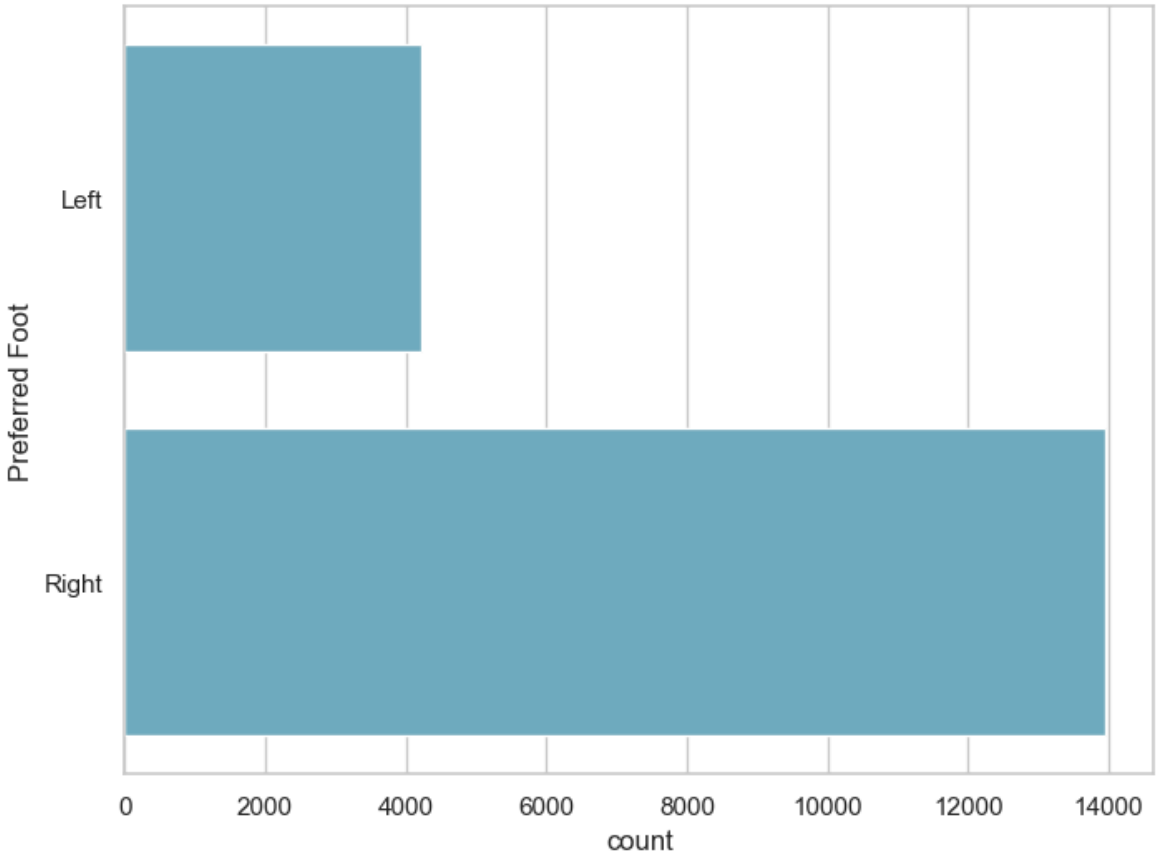


In [34]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.countplot(x="Preferred Foot", data=fifa19, hue="Real Face")
plt.show()
```

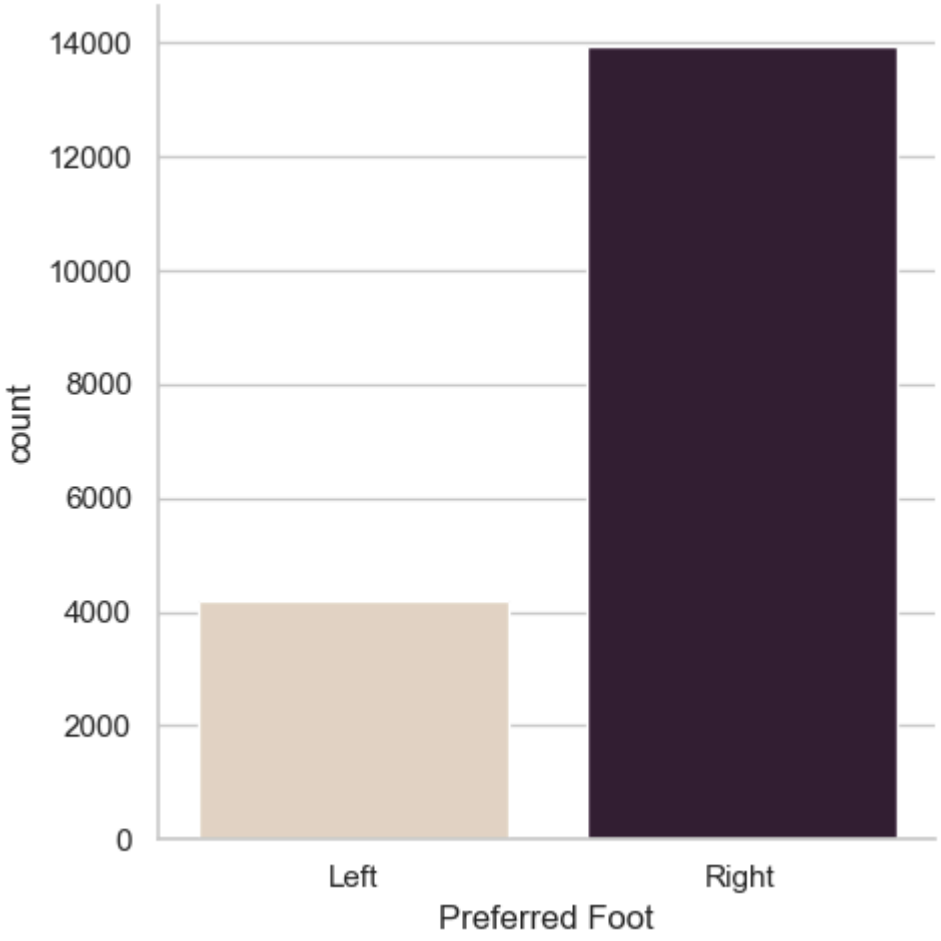


In [35]:
```
# draw plot vertically
f, ax = plt.subplots(figsize=(8, 6))
```

```
sns.countplot(y="Preferred Foot", data=fifa19, color="c")
plt.show()
```



In [36]:
```
# Seaborn Catplot() function
g = sns.catplot(x="Preferred Foot", kind="count", palette="ch:.25", data=fifa19)
```

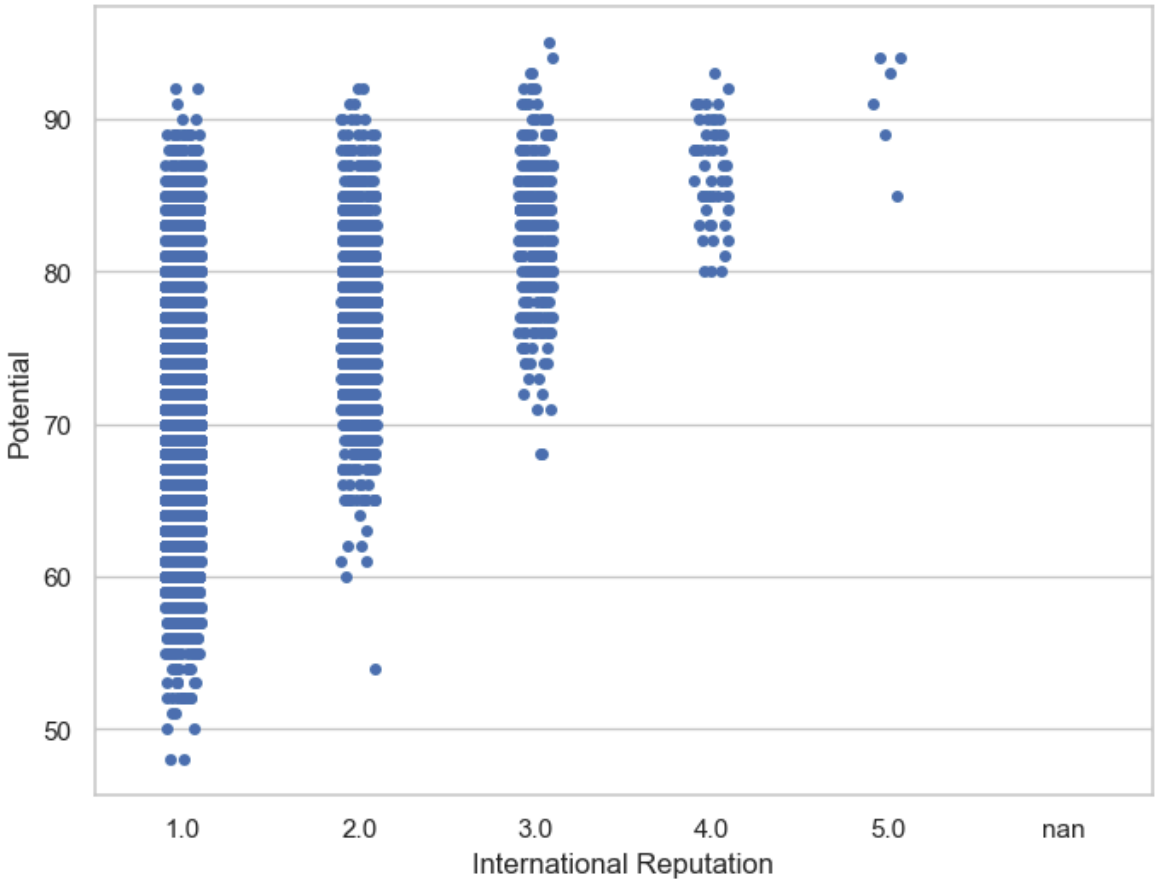


In [37]:
```
# Explore International Reputation Variable
```

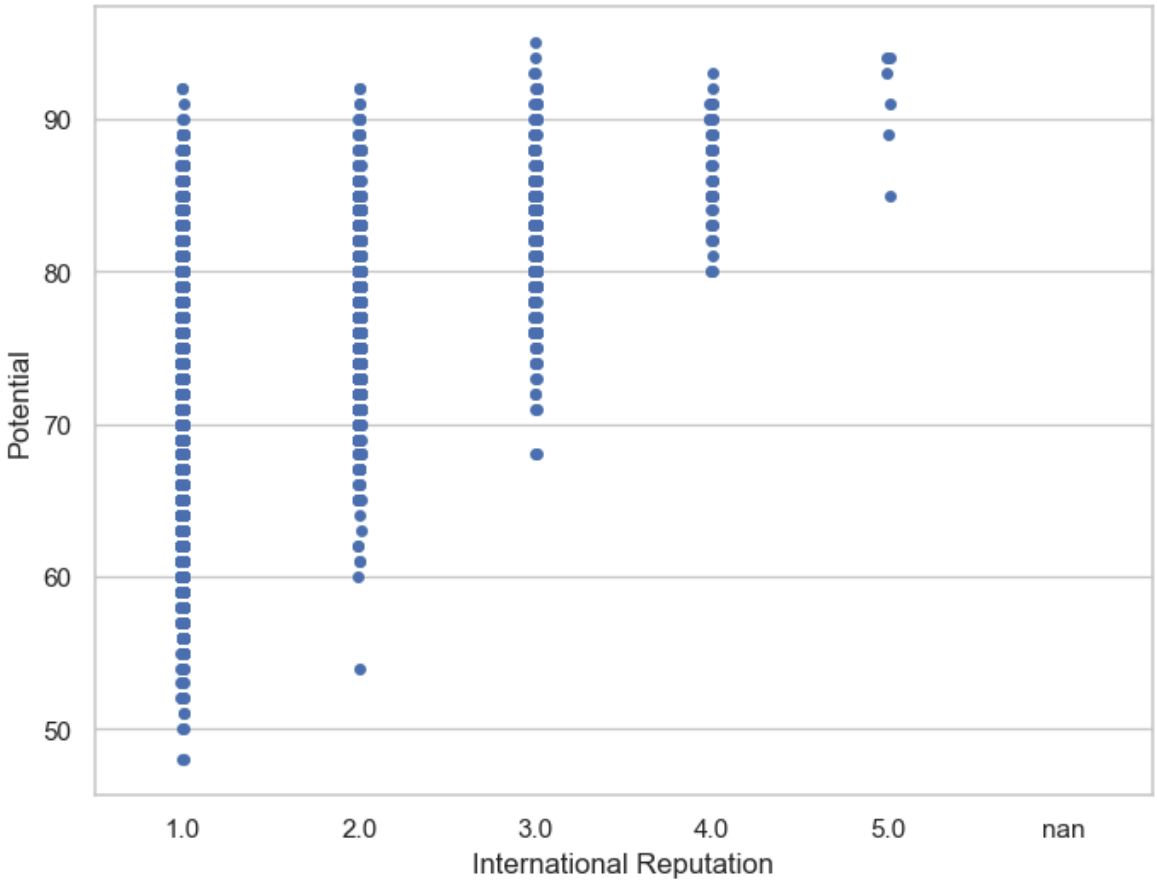
In [38]:
```
# Check the number of unique values in International Reputation Varible

fifa19['International Reputation'].value_counts()
```

Out[38]:
```
International Reputation
1.0    16532
2.0     1261
3.0      309
4.0       51
5.0        6
Name: count, dtype: int64
```

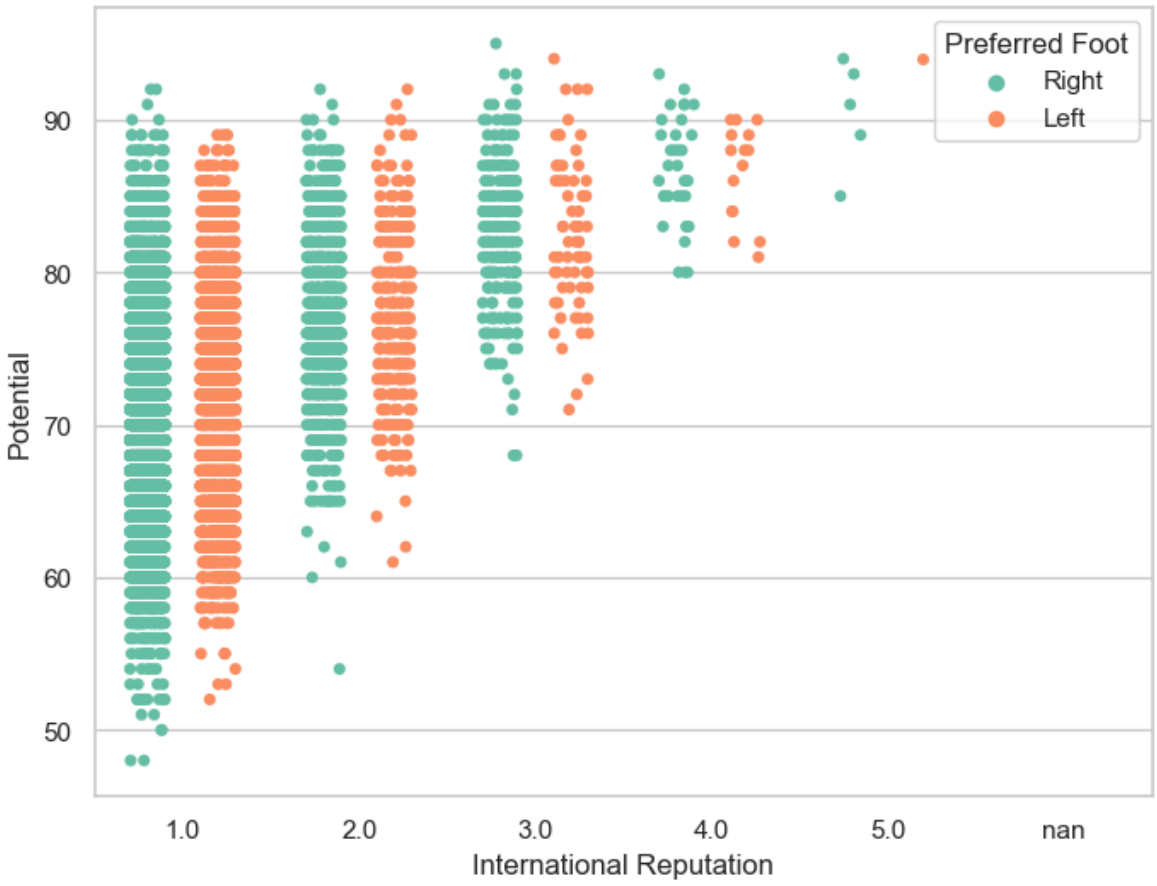
In [39]:
```
# Seaborn StripPlot() function
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", data=fifa19)
plt.show()
```



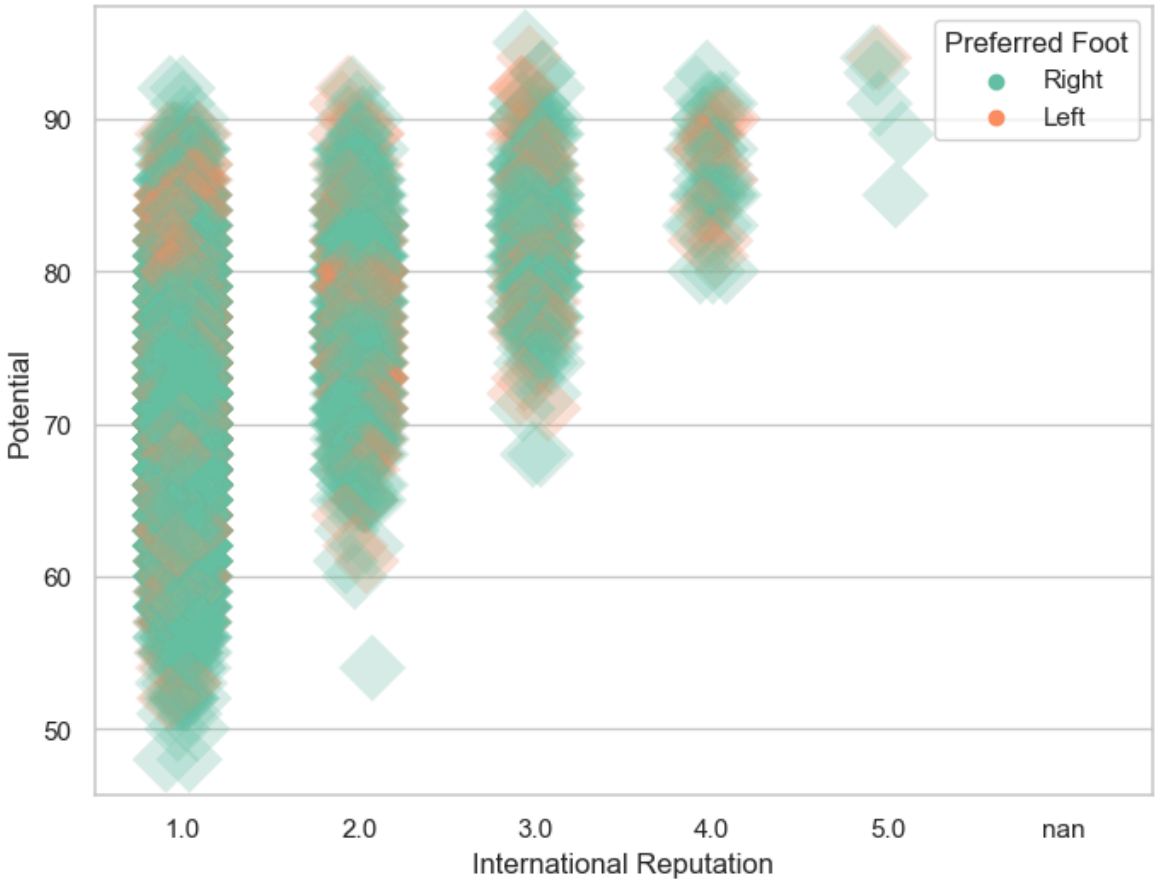
In [40]:
```
# we can add jitter to bring out the distribution of values

f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", data=fifa19, jitter=0
plt.show()
```



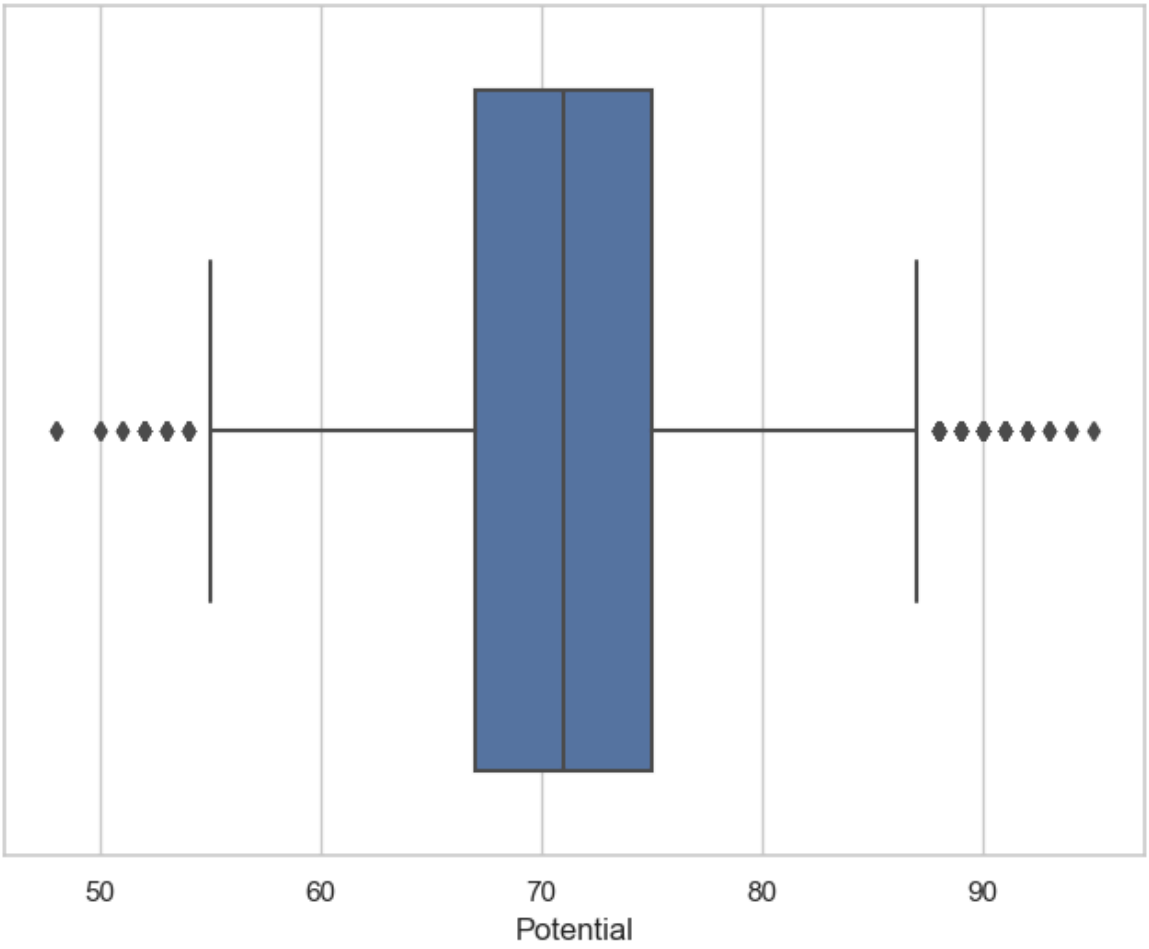
In [41]:
```
# we can nest the strips within a second categorical variable - preferred foot

f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", hue="Preferred Foot",
                    data=fifa19, jitter=0.2, palette="Set2", dodge=True)
plt.show()
```



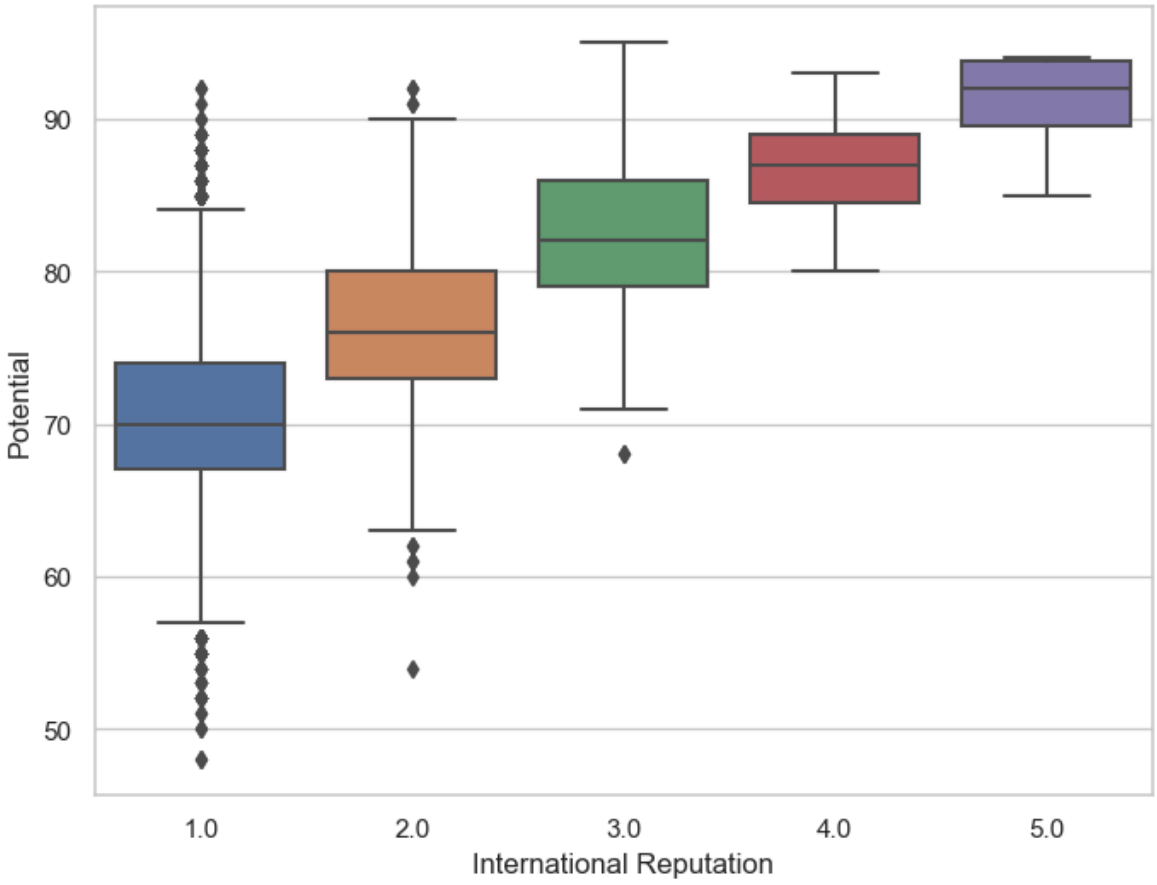
In [42]:
```
# draw strips with large points and different aesthetics
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", hue="Preferred Foot",
                    data=fifa19, palette="Set2", size=20, marker="D",
                    edgecolor="gray", alpha=.25)
plt.show()
```



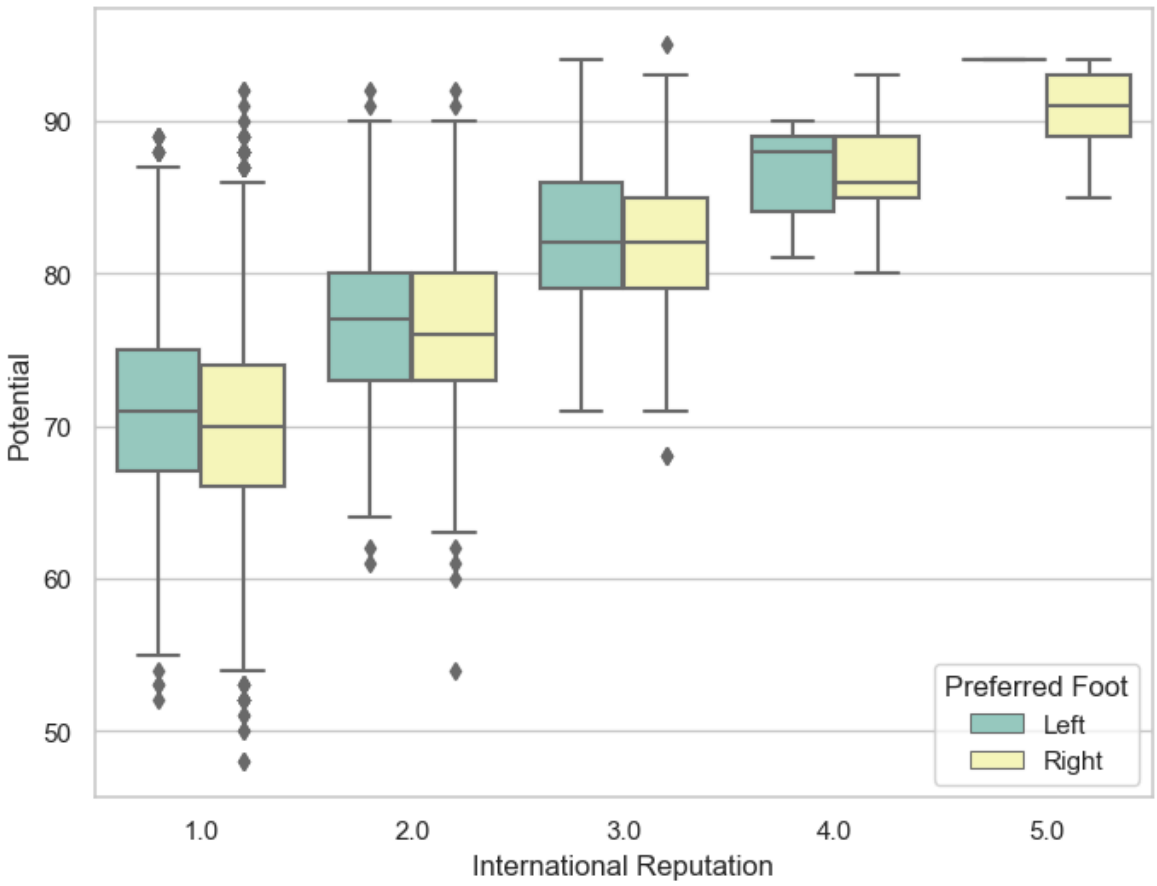
In [43]:
```
# Seaborn Boxplot()
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=fifa19["Potential"])
plt.show()
```



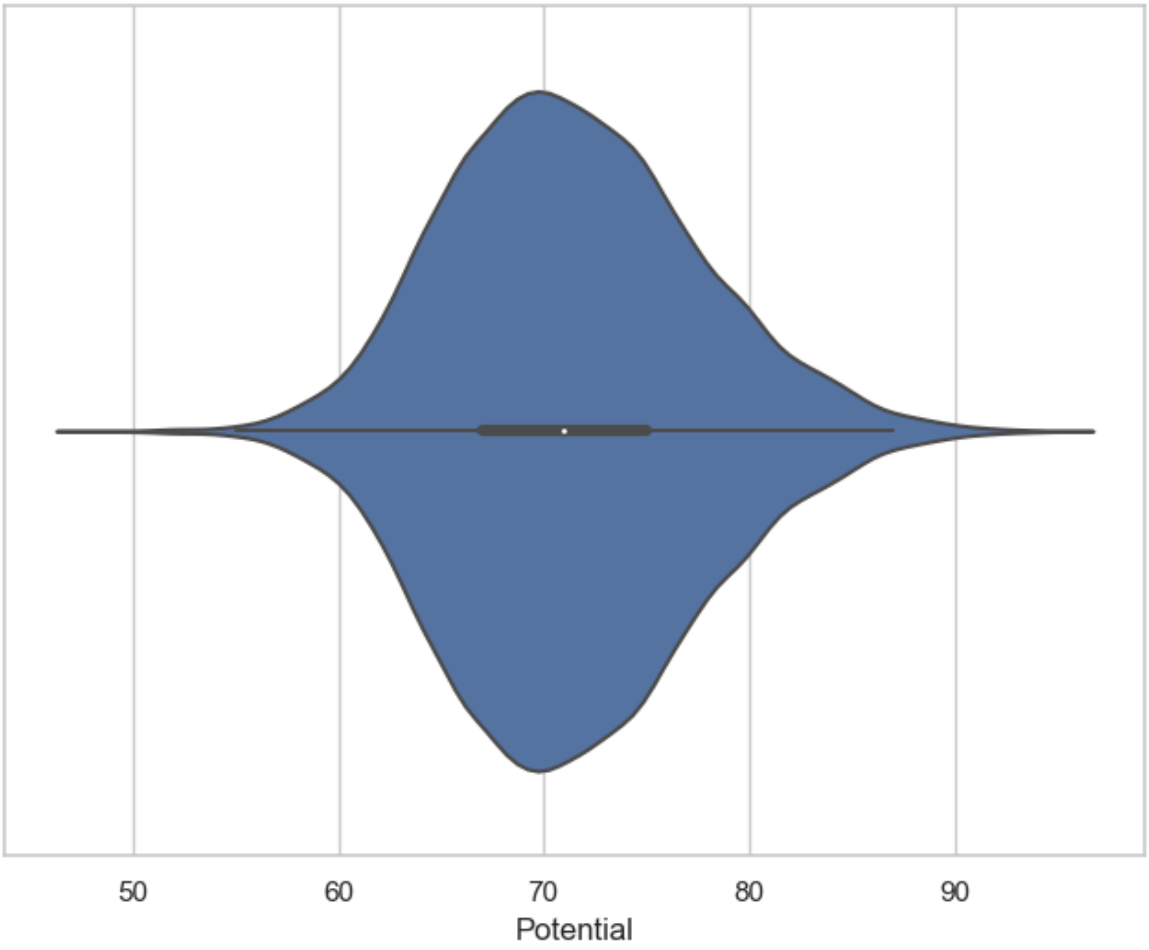
In [44]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x="International Reputation", y="Potential", data=fifa19)
plt.show()
```



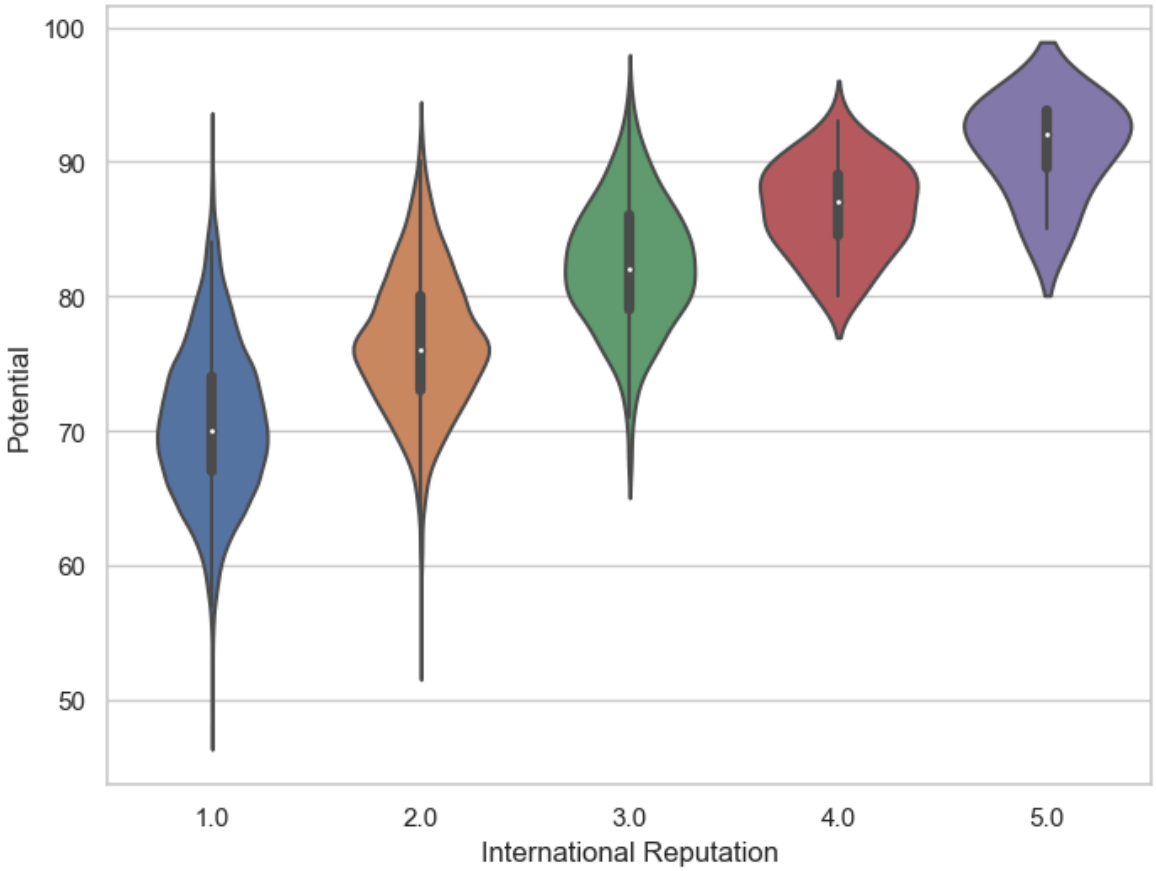
In [45]:
```
# we can draw a boxplot with nested grouping by two categorical variable
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x="International Reputation", y="Potential", hue="Preferred Foot", d
plt.show()
```



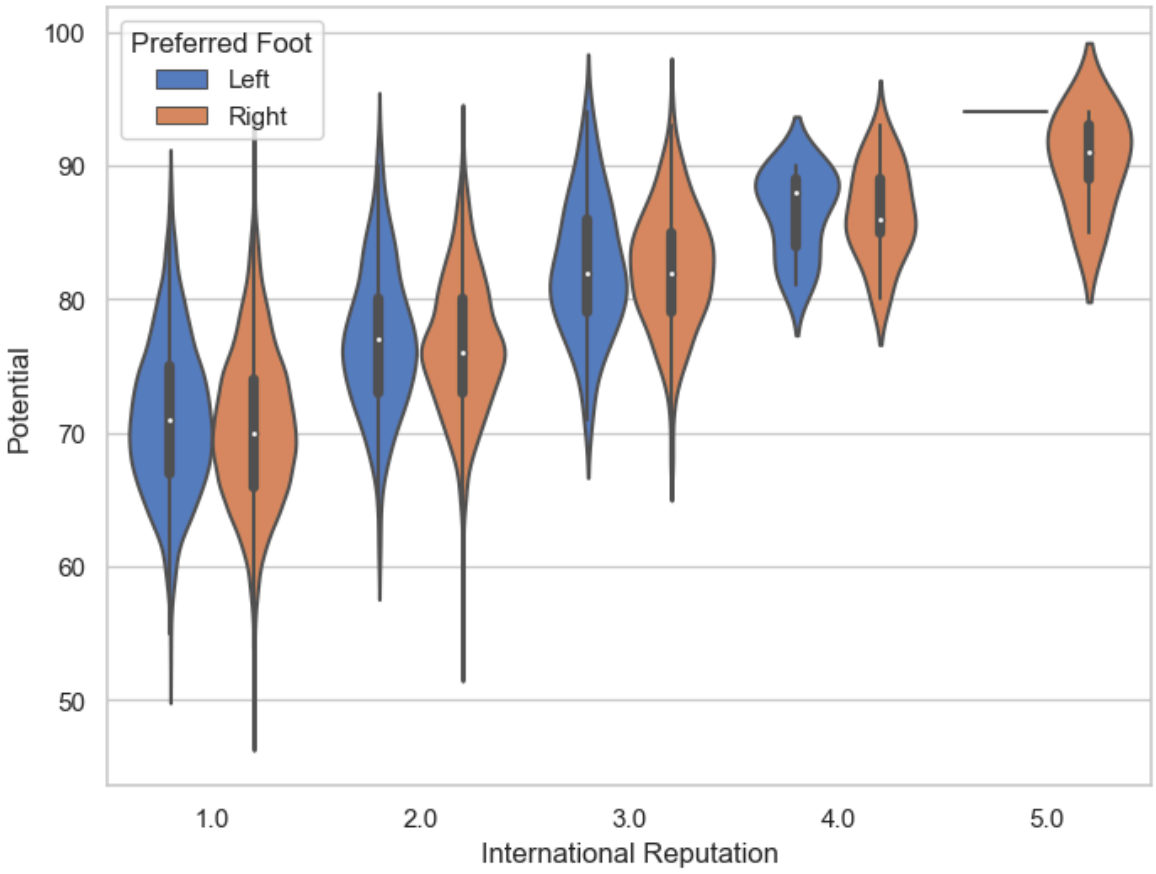
In [46]:
```
# Seaborn Violin Plot() function

f, ax = plt.subplots(figsize=(8, 6))
sns.violinplot(x=fifa19["Potential"])
plt.show()
```



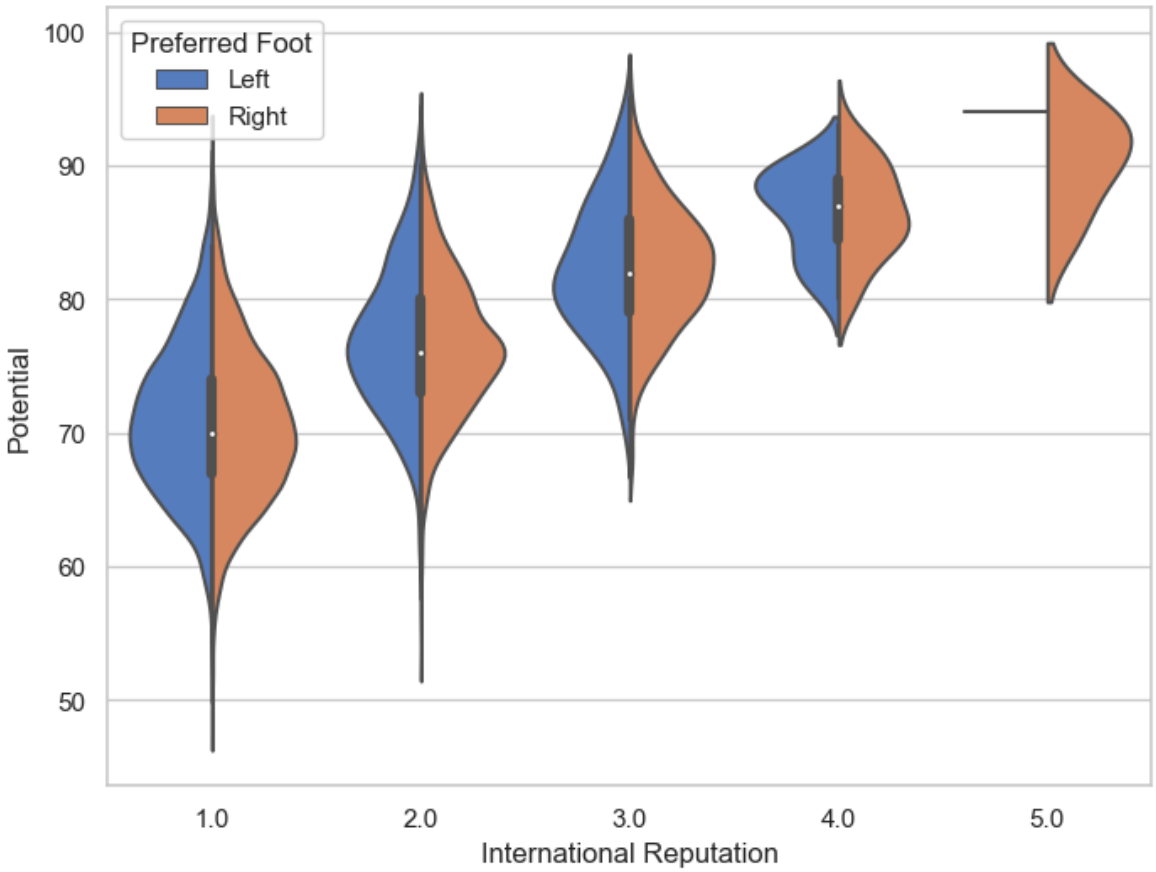
In [47]:
```
# draw the vertical violinplot grouped by the categorical variable International
f, ax = plt.subplots(figsize=(8, 6))
sns.violinplot(x="International Reputation", y="Potential", data=fifa19)
plt.show()
```



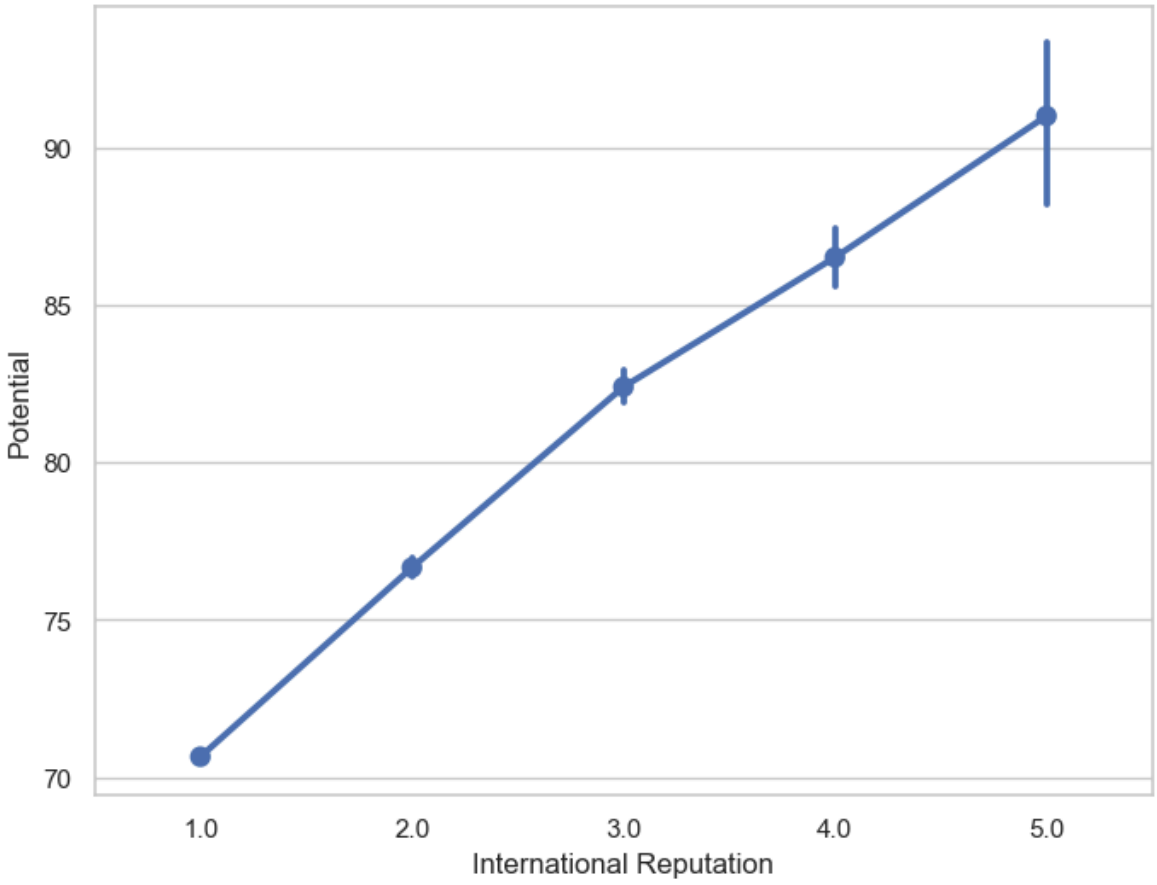
In [48]:
```
# draw a violinplot with nested grouing by two categorical variable

f, ax = plt.subplots(figsize=(8, 6))
sns.violinplot(x="International Reputation", y="Potential", hue="Preferred Foot"
plt.show()
```



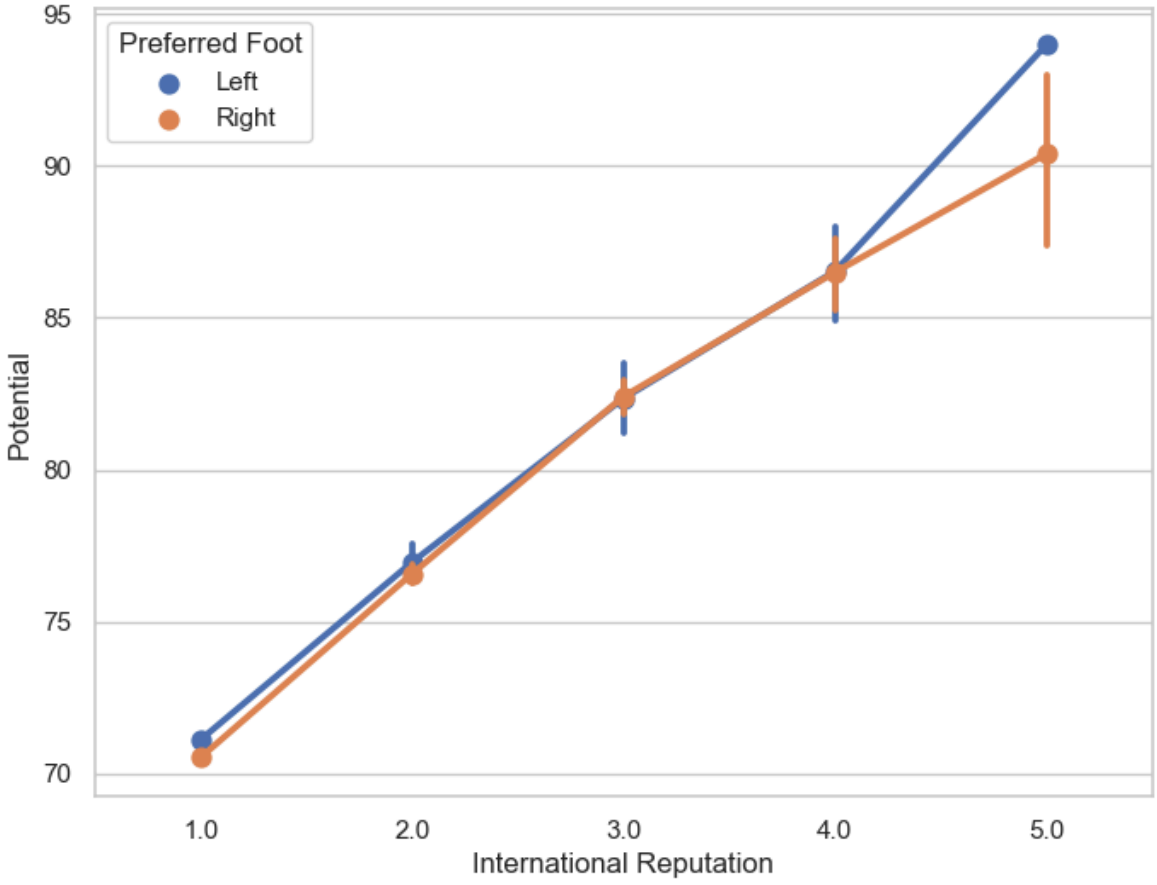
In [49]:
```
# Draw split violins to compare the across the hue variable as follows:

f, ax = plt.subplots(figsize=(8, 6))
sns.violinplot(x="International Reputation", y="Potential", hue="Preferred Foot"
               data=fifa19, palette="muted", split=True)
plt.show()
```



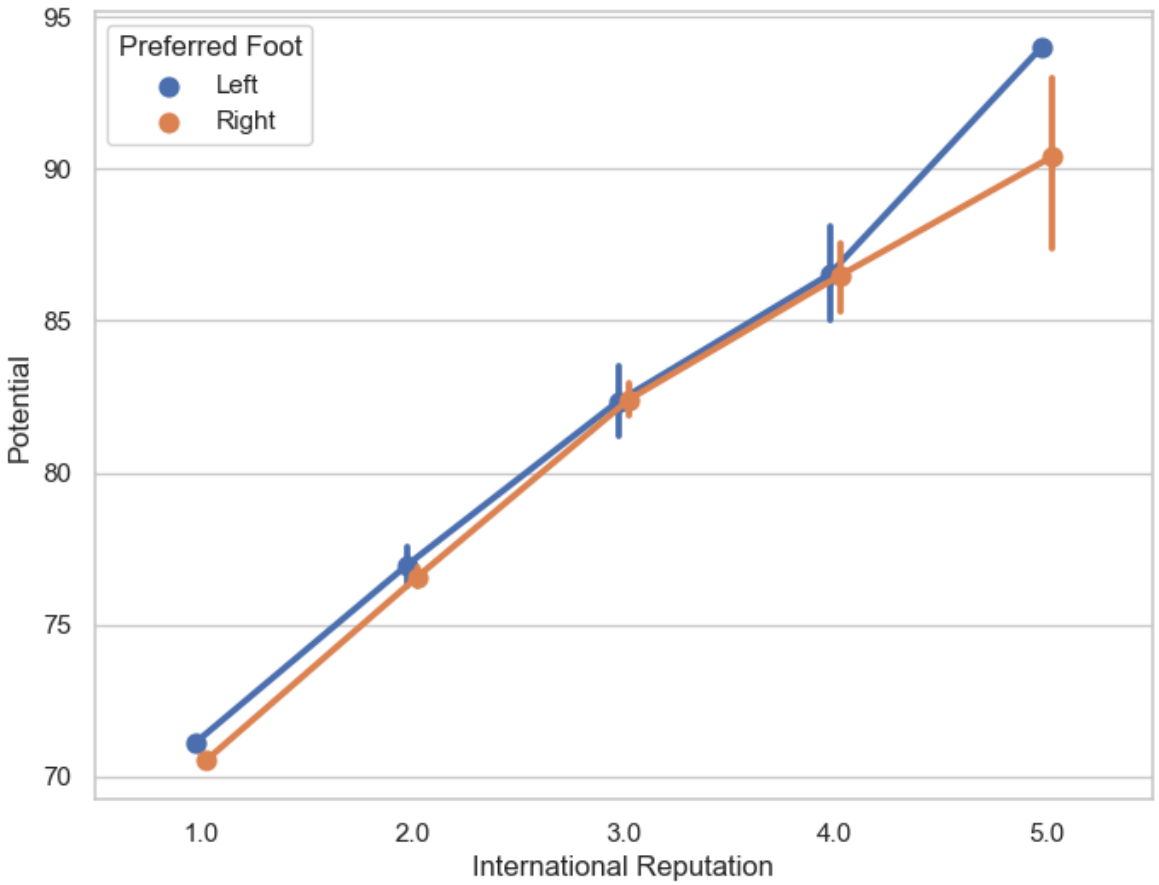
In [50]:
```
# Seaborn pointplot()
f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", data=fifa19)
plt.show()
```



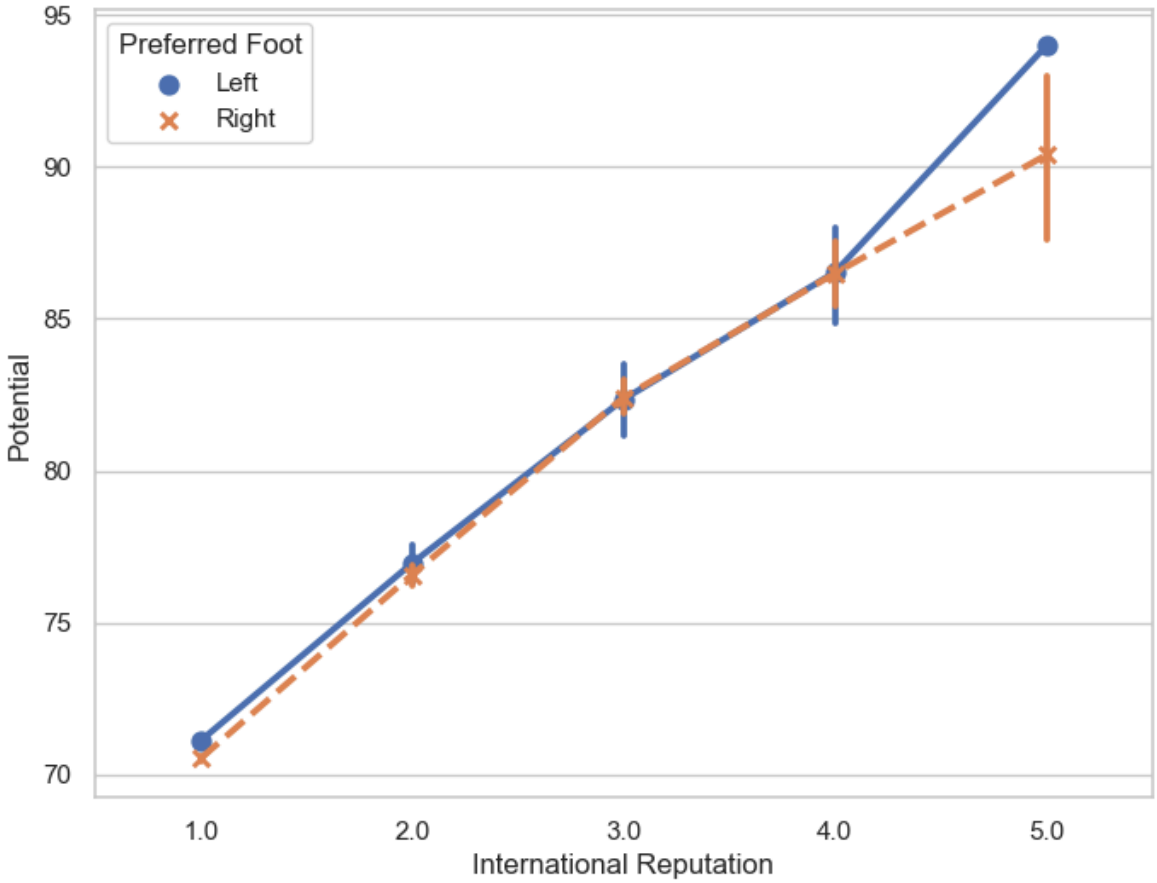
In [51]:
```
# we can draw a set of vertical points with nested grouping by a two variable
f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", hue="Preferred Foot",
plt.show()
```



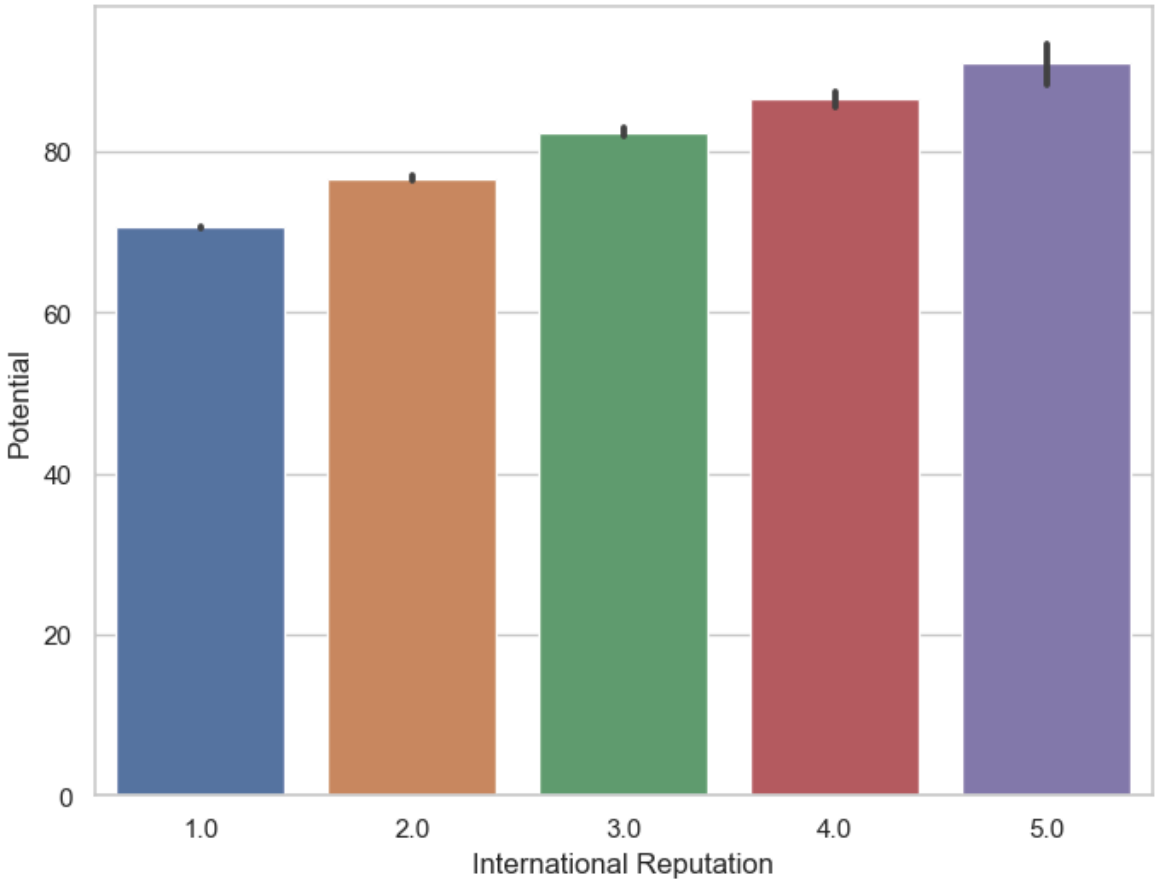
In [52]:
```
# We can separate the points for different hue levels along the categorical axis
f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", hue="Preferred Foot",
plt.show()
```



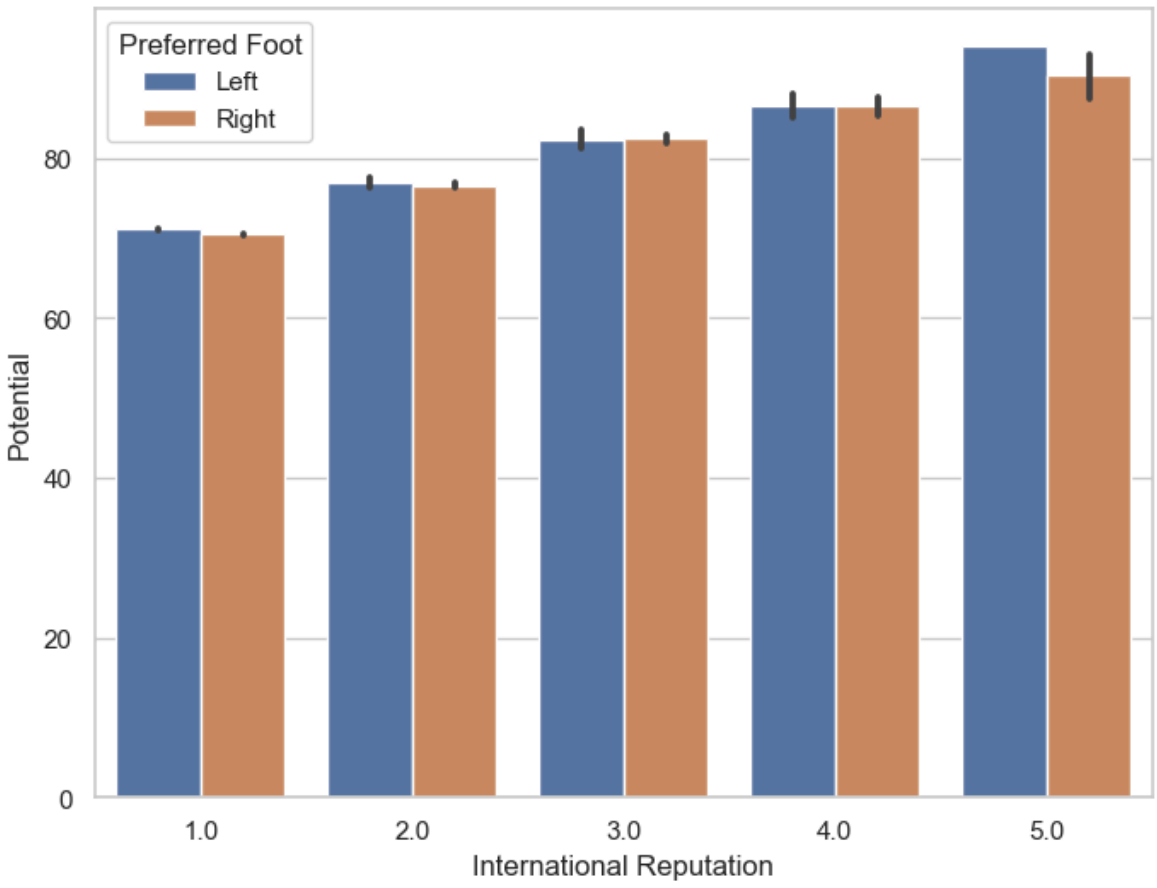
In [53]:
```
# Use a different marker and line style for the hue levels

f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", hue="Preferred Foot",
              data=fifa19, markers=["o", "x"], linestyles=["-", "--"])
plt.show()
```



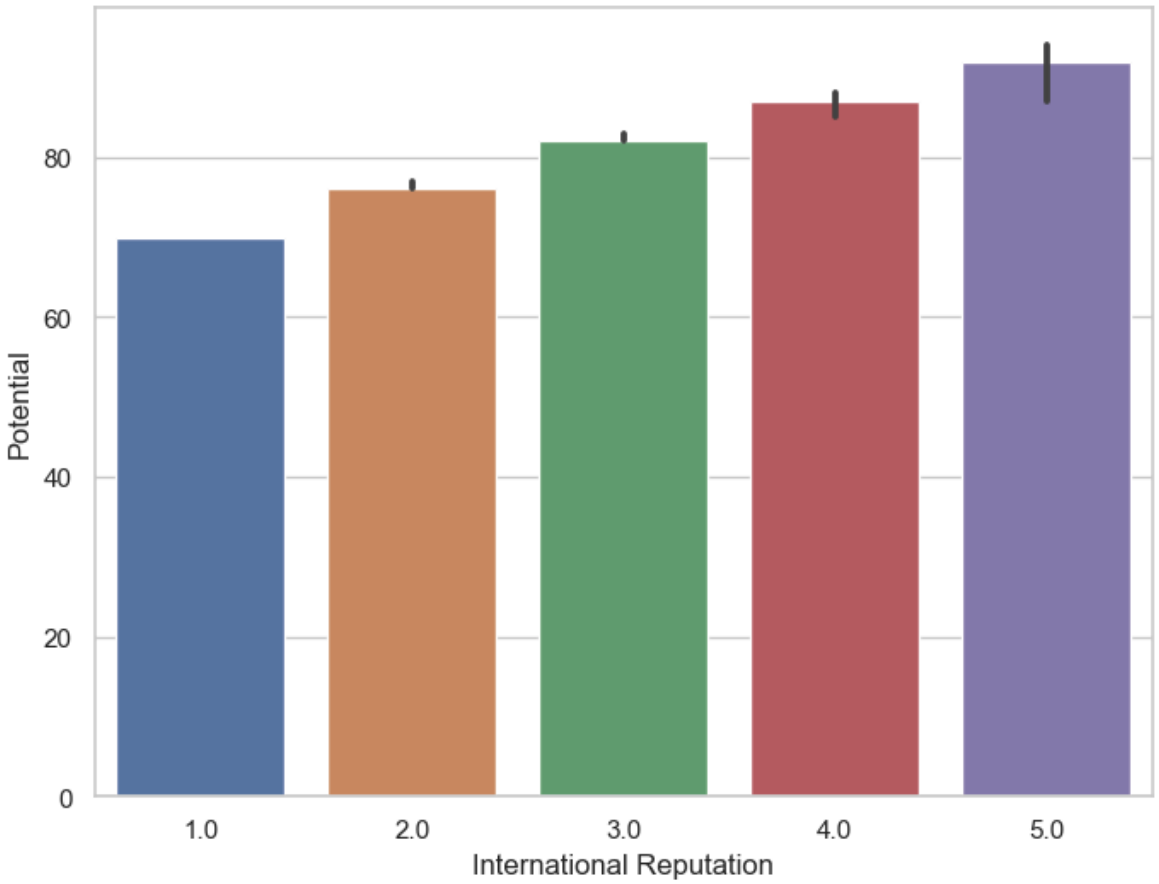
In [54]:
```
# Seaborn Boxplot()

f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19)
plt.show()
```



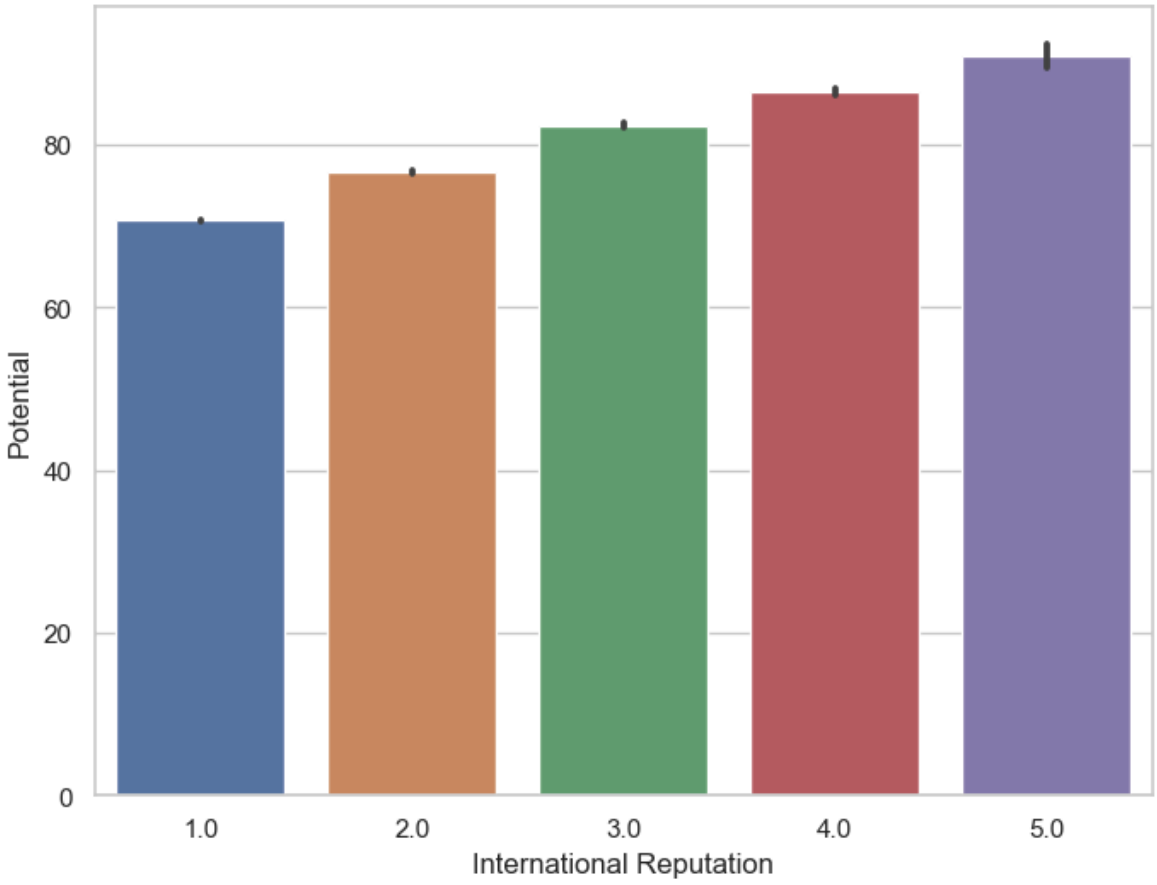
In [56]:
```
# draw a set of vertcal bars with nested grouping by a two variable
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential",hue="Preferred Foot", da
plt.show()
```



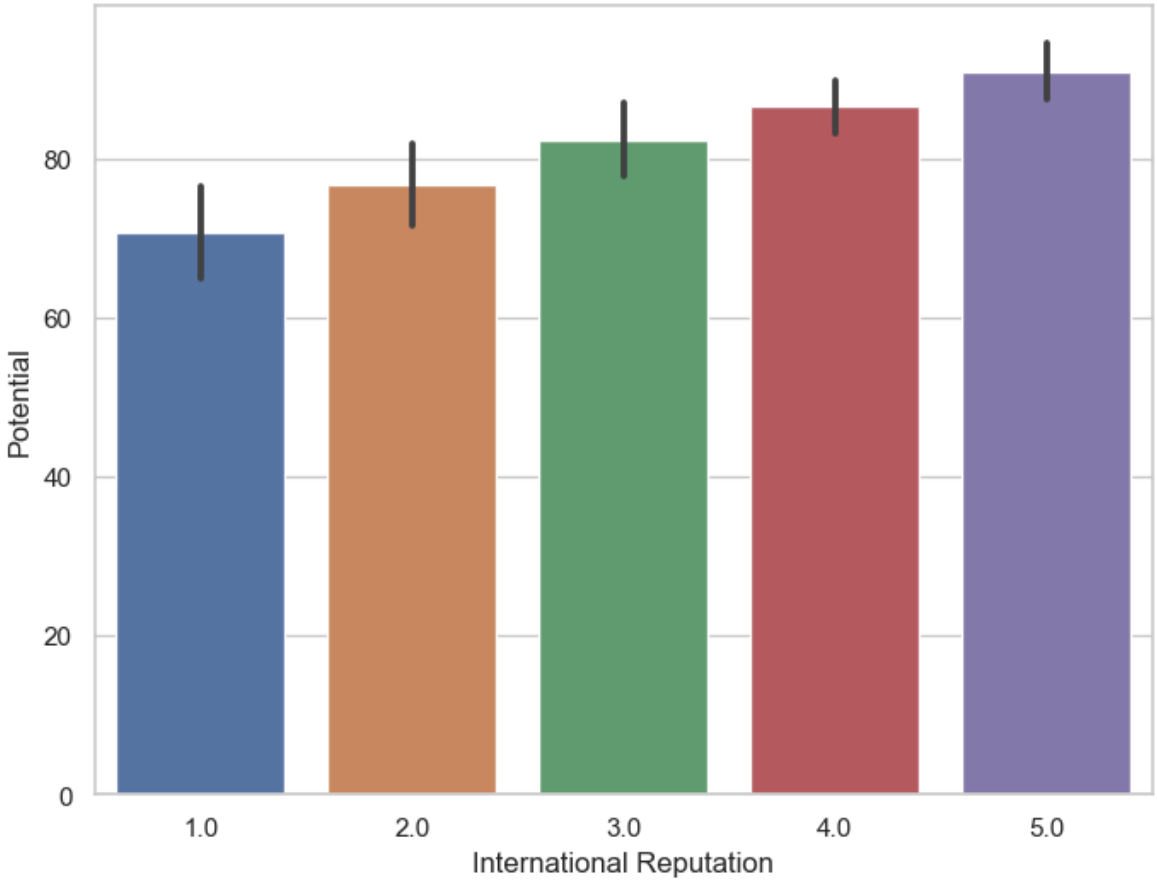
In [60]:
```
# We can use median as the estimation of central tendency
from numpy import median
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19, estimator=
plt.show()
```



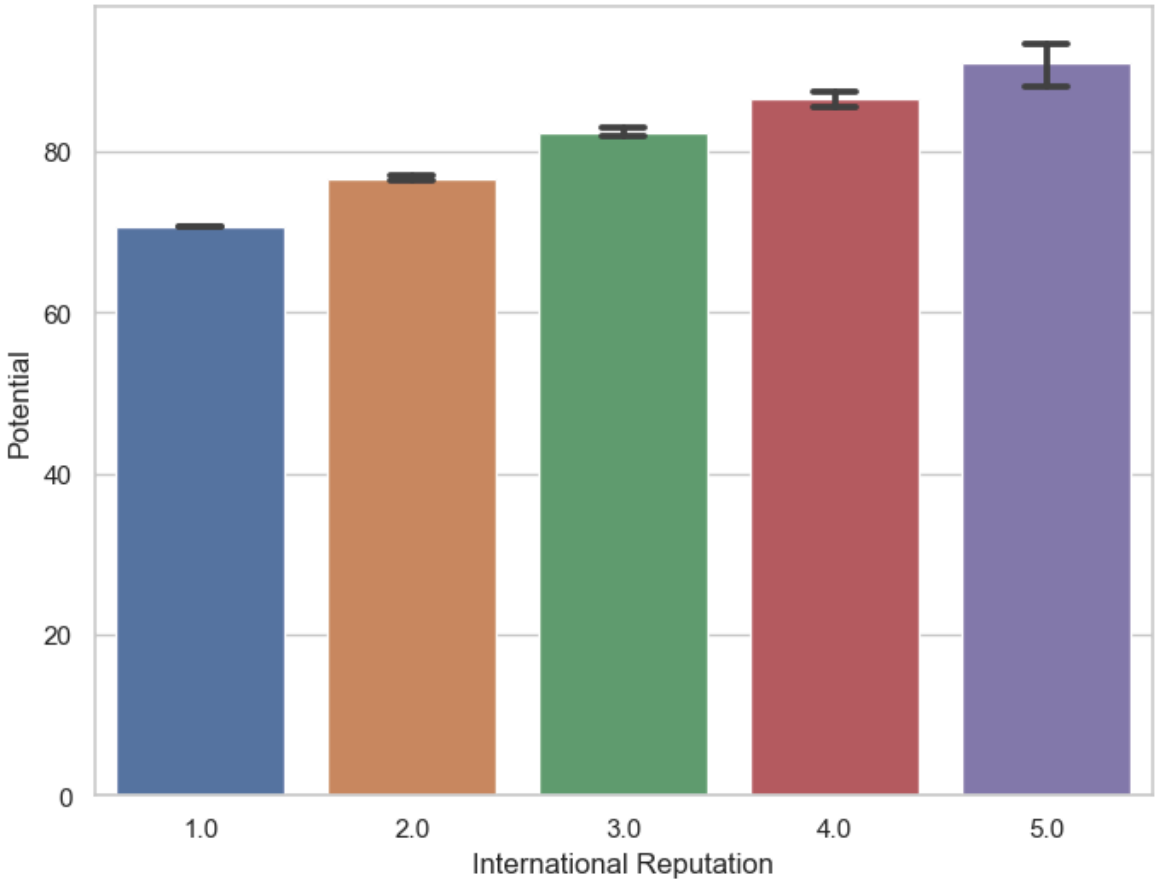
In [61]:
```
# we can show the standard erroe of the mean with the error bars as follows:
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19, ci=68)
plt.show()
```



In [62]:
```
# We can show standars deviation of observations instead of a confideance interv
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19, ci="sd")
plt.show()
```

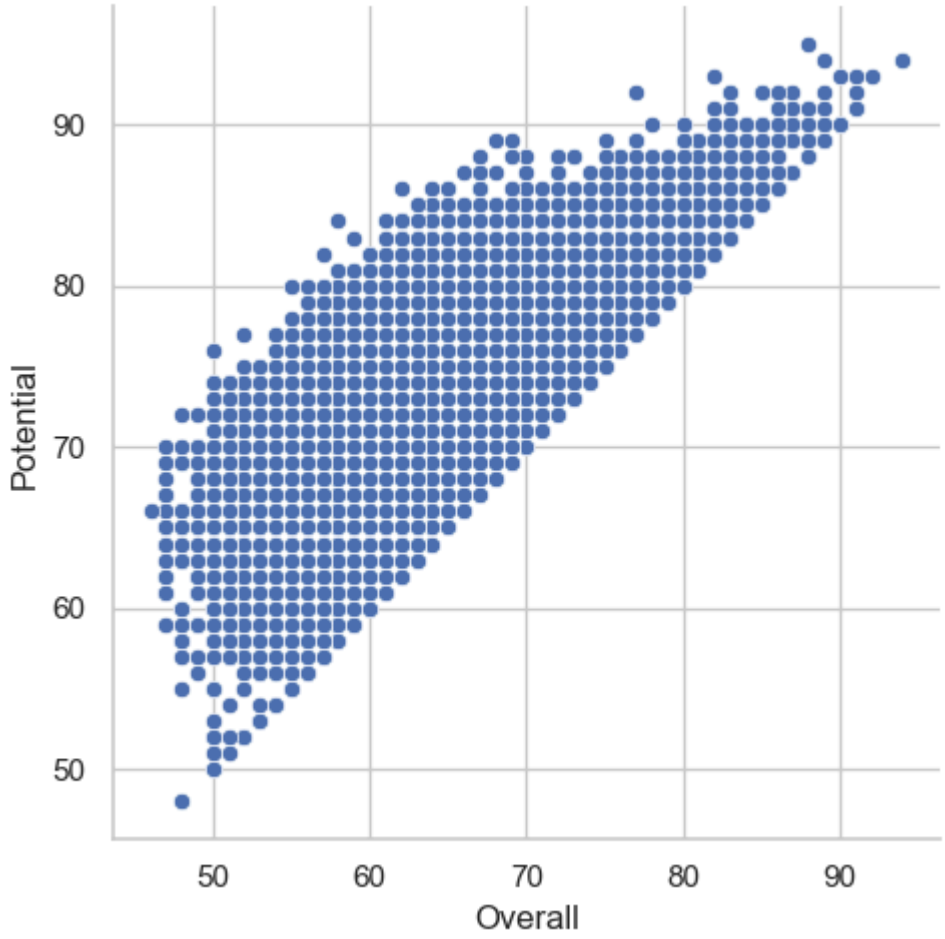


In [63]:
```
# we can add "caps" to the error bars:
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19, capsize=0.
plt.show()
```

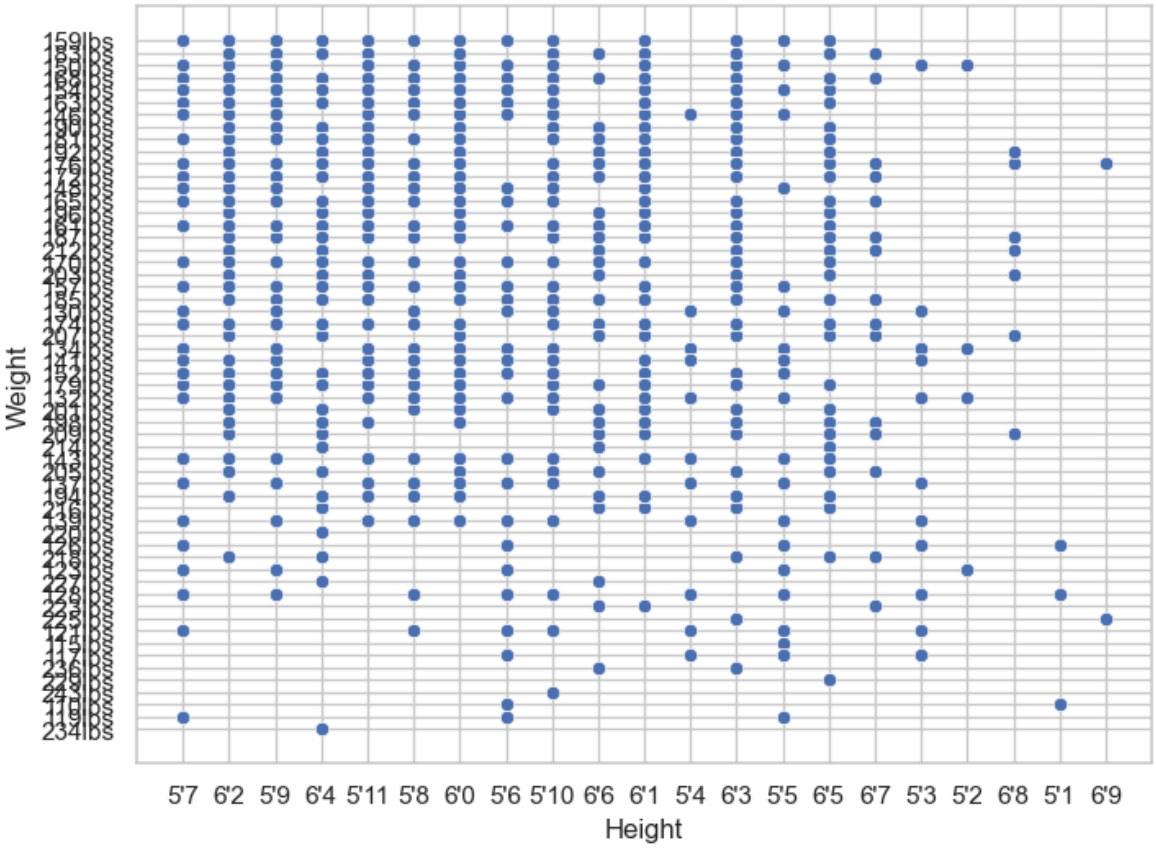


In [64]:
```
# Visualizing statistical relationship with seaborn relplot() function
```

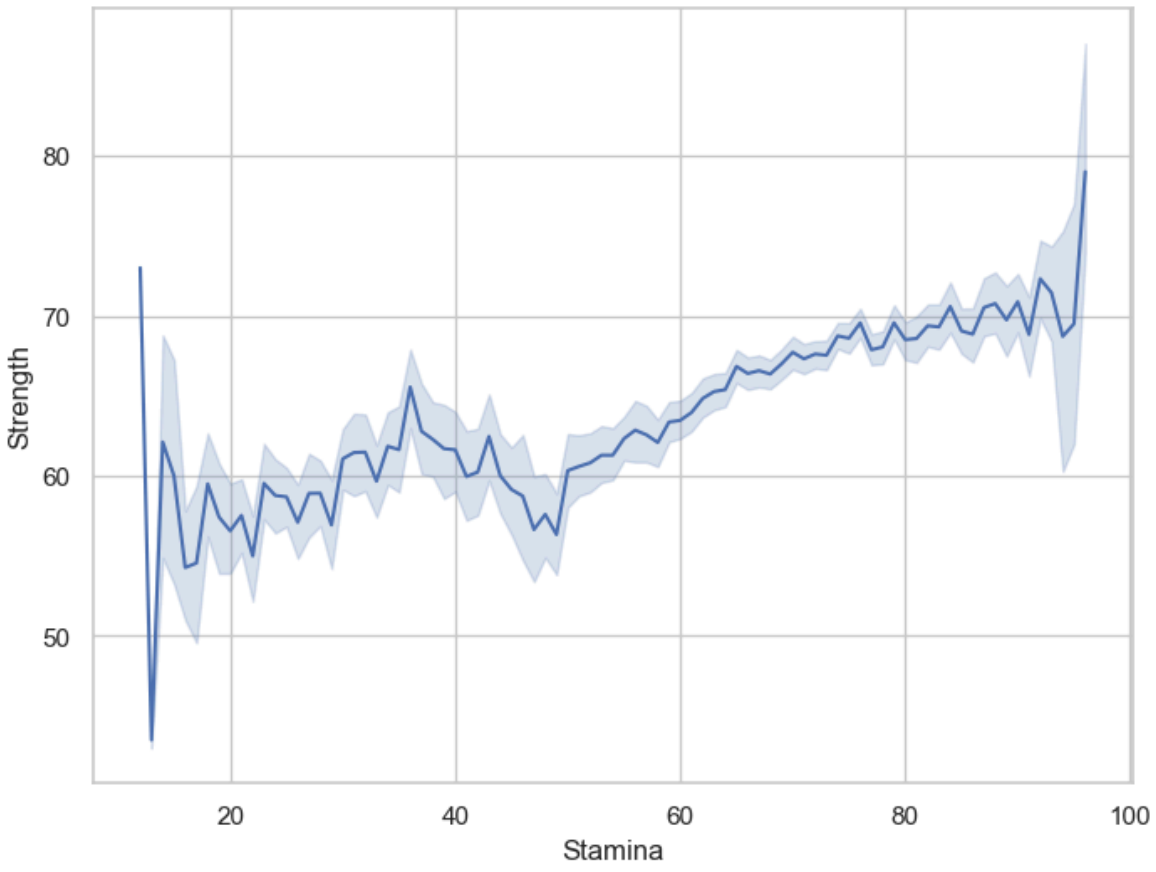
In [65]:
```
# Seaborn relplot() function

g = sns.relplot(x="Overall", y="Potential", data=fifa19)
```



In [66]:
```
# Seaborn scatterplot()

f, ax = plt.subplots(figsize=(8, 6))
sns.scatterplot(x="Height", y="Weight", data=fifa19)
plt.show()
```

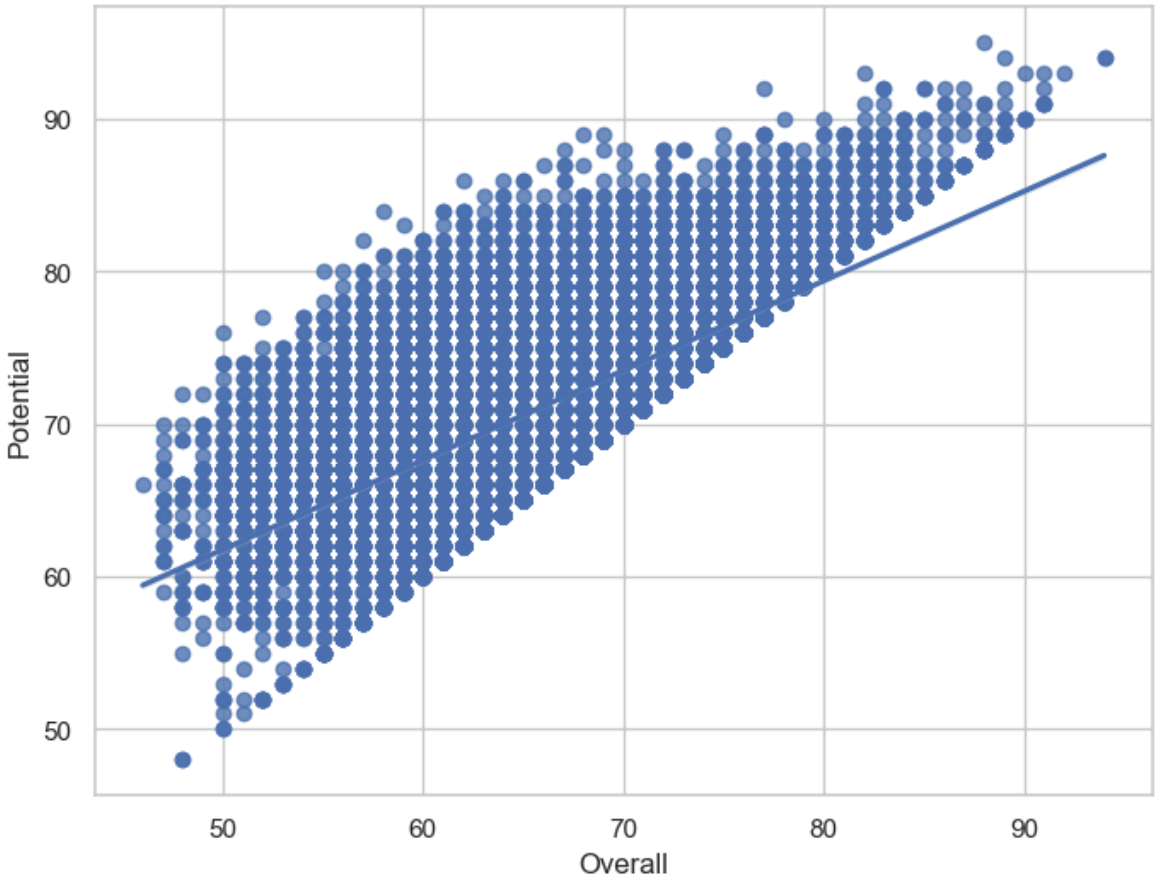


In [67]:
```
# Seaborn lineplot() function

f, ax = plt.subplots(figsize=(8, 6))
ax = sns.lineplot(x="Stamina", y="Strength", data=fifa19)
plt.show()
```

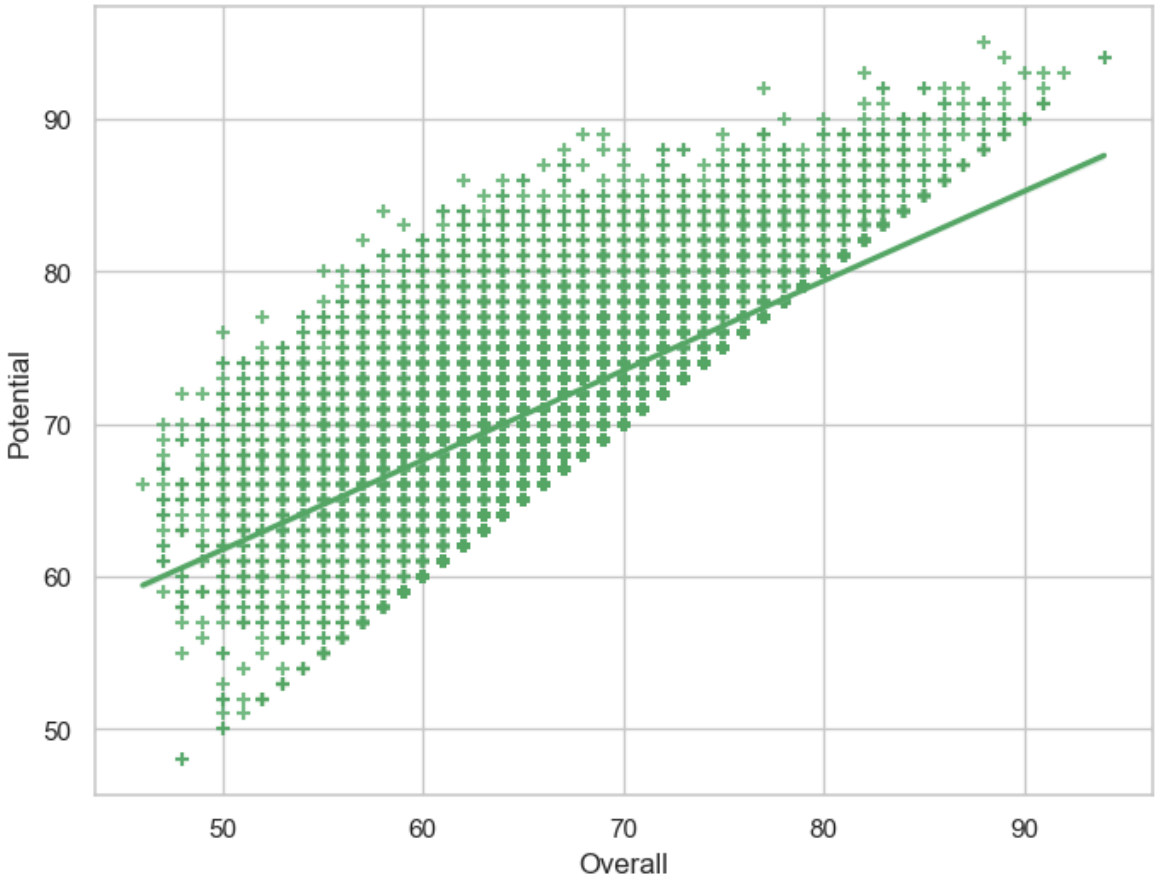


In [68]:
```
# Visualise linear relationship with Seaborn regplot() function
```

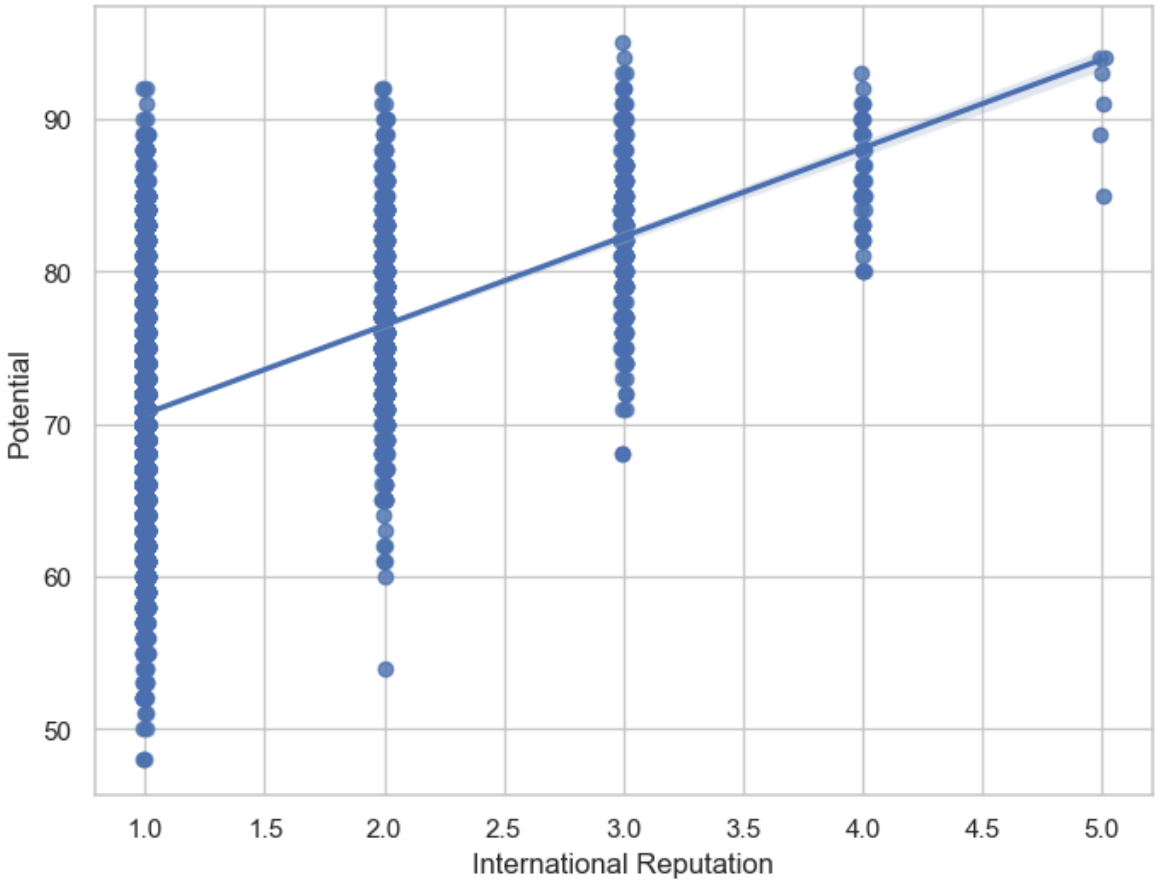
In [69]:
```
# Seaborn regplot() function
f, ax = plt.subplots(figsize=(8,6))
ax = sns.regplot(x="Overall",y="Potential",data=fifa19)
plt.show()
```



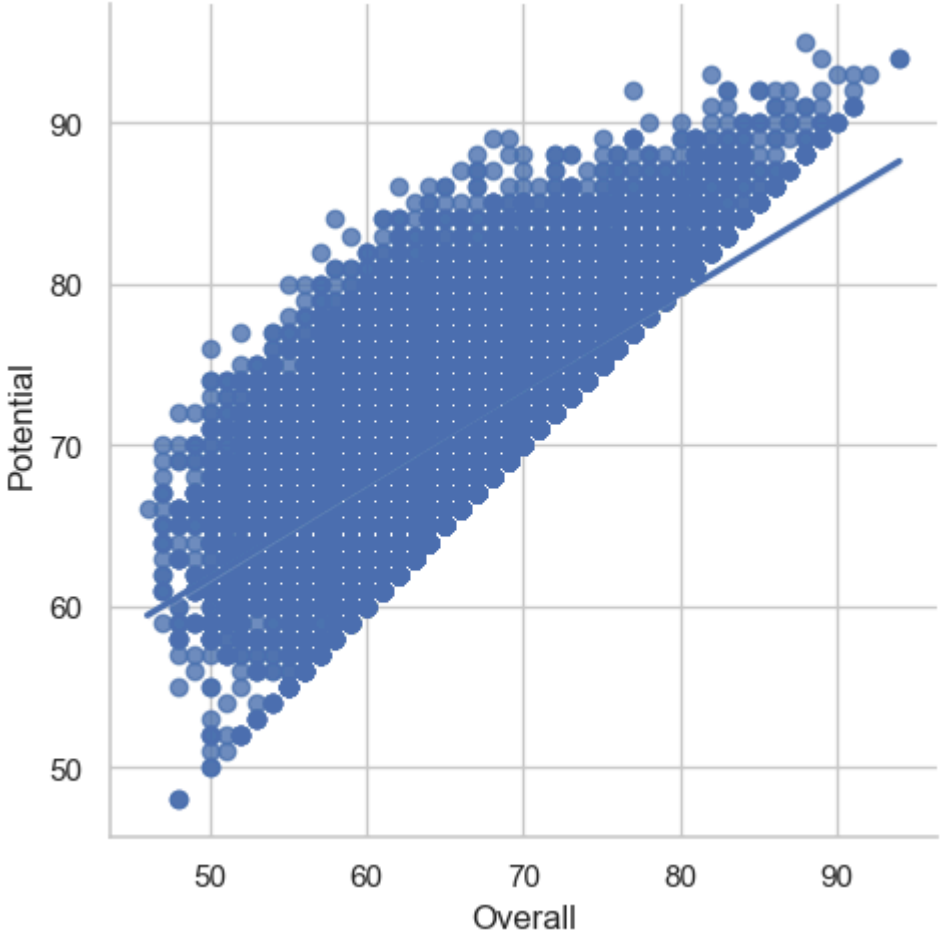
In [70]:
```
# we can use different color and marker
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa19, color= "g", marker="+"
plt.show()
```



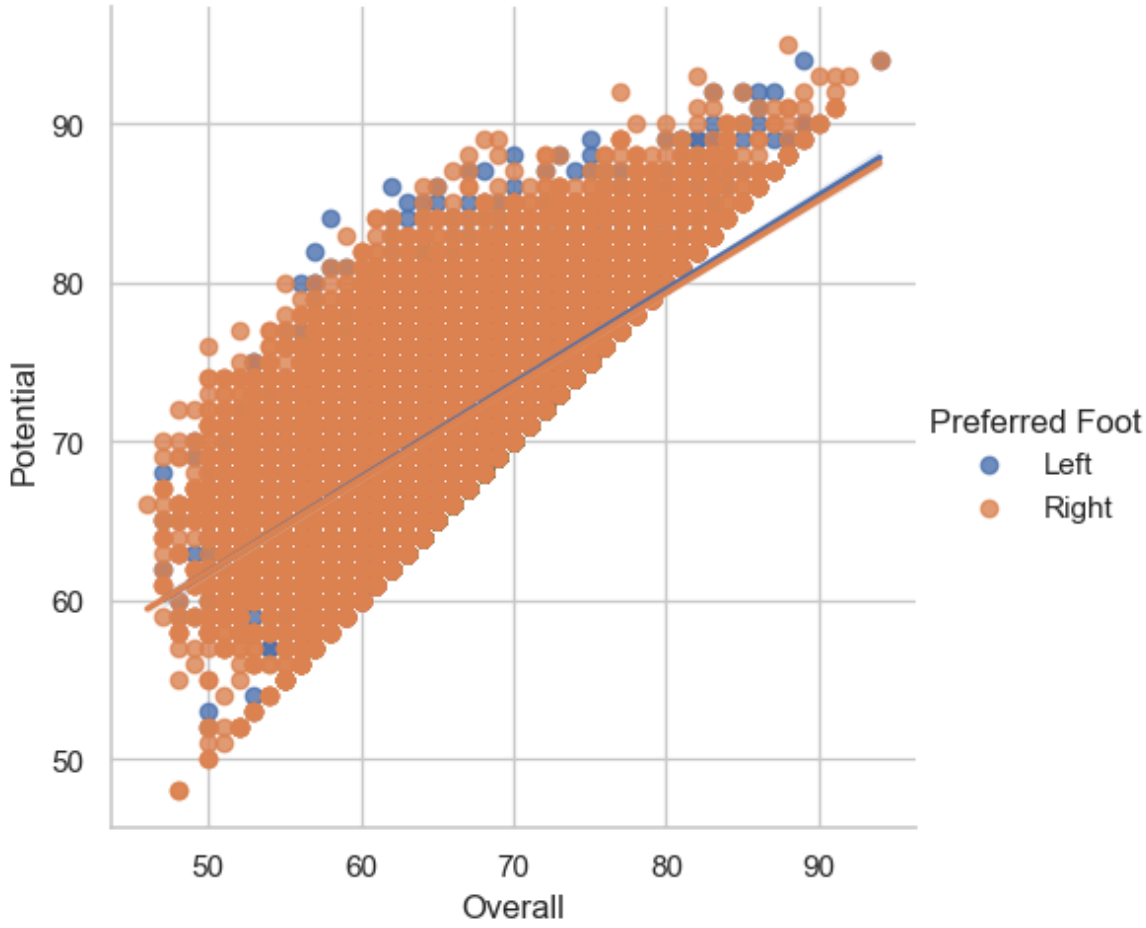
In [71]:
```
# we can plot with a discrete variable and add some jitter as follows:

f,ax=plt.subplots(figsize=(8,6))
sns.regplot(x="International Reputation", y="Potential", data=fifa19, x_jitter=.
plt.show()
```



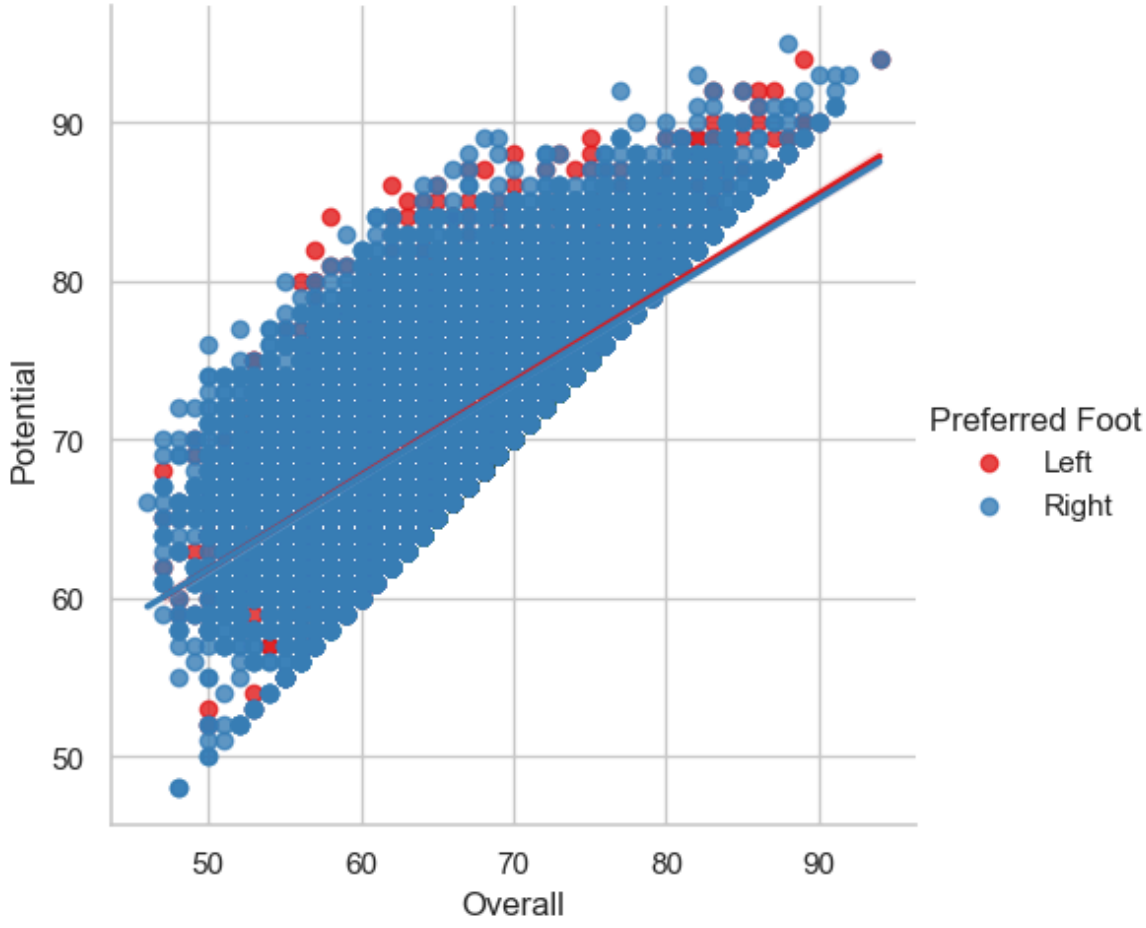
In [72]:
```
# Seaborn lmplot() function

g = sns.lmplot(x="Overall", y="Potential", data=fifa19)
```



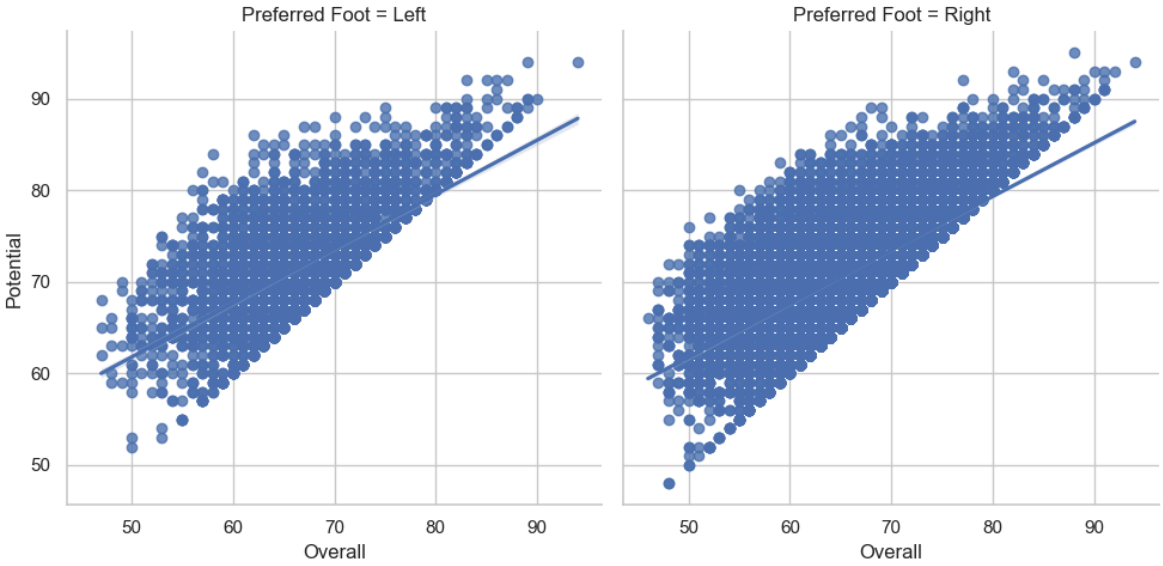
In [73]:
```
# we can comdition on a third variable and plot the levels in different colors a

g = sns.lmplot(x="Overall", y="Potential", hue="Preferred Foot", data=fifa19)
```



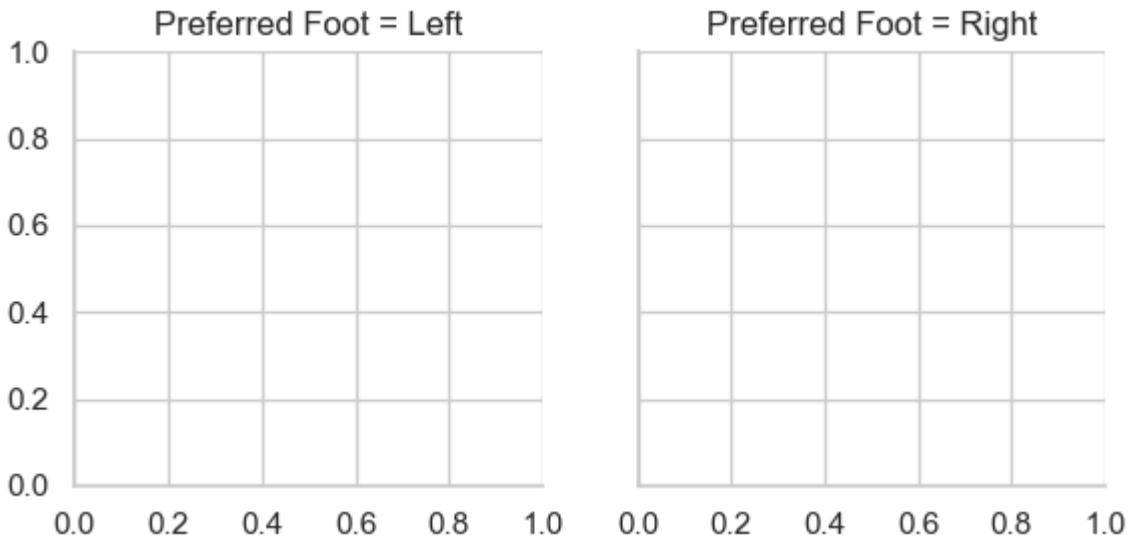
In [74]:
```
# using different color palette
g= sns.lmplot(x="Overall", y="Potential", hue="Preferred Foot", data=fifa19, pal
```



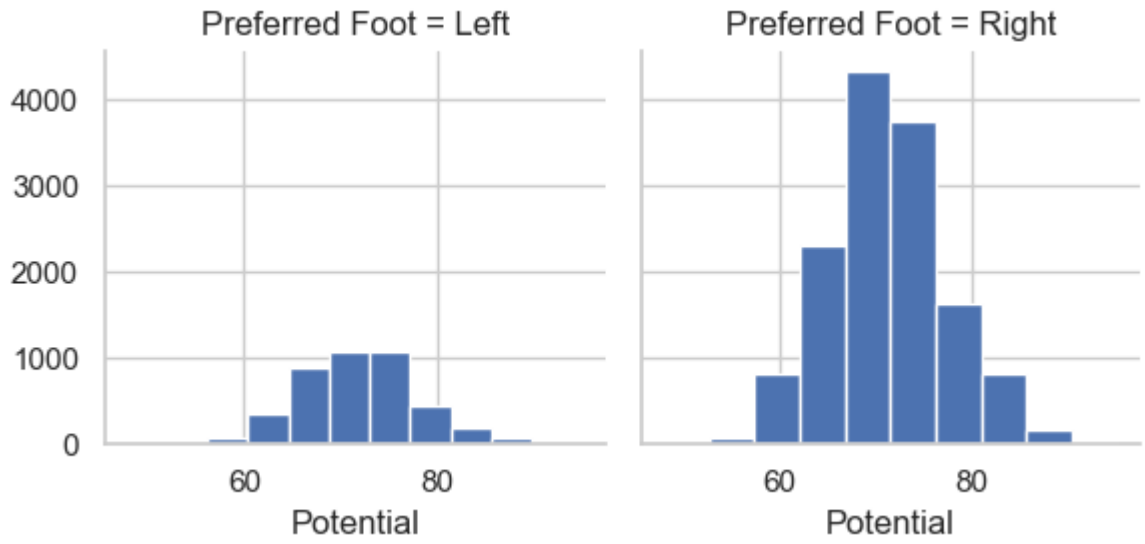
In [75]:
```
# we can plot the levels of the third variable across different columns as follo

g= sns.lmplot(x="Overall", y="Potential", col="Preferred Foot", data=fifa19)
```



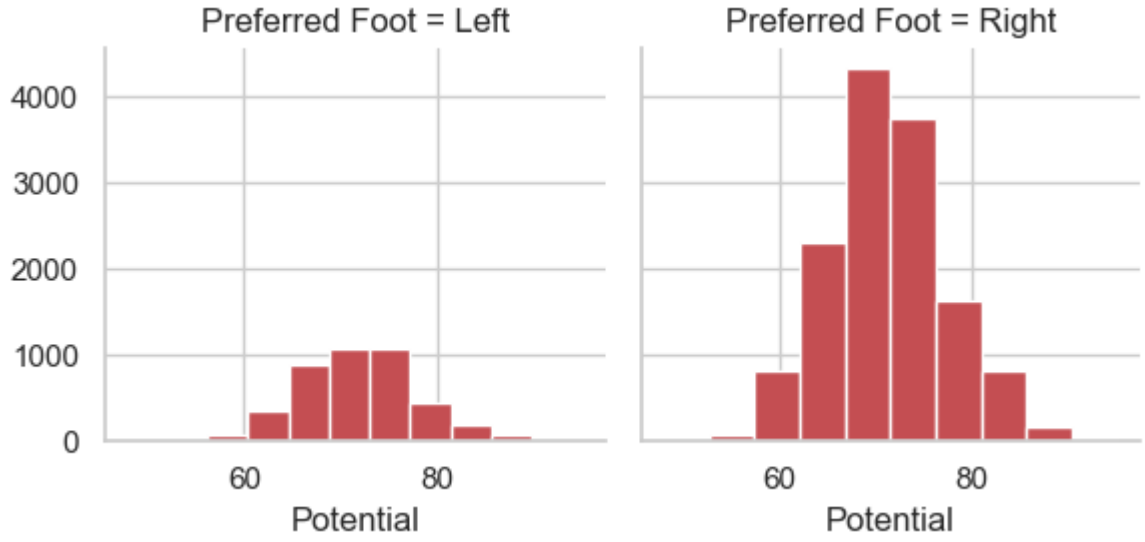
In [76]:
```
# Multi-plot Grids
g = sns.FacetGrid(fifa19, col="Preferred Foot")
```



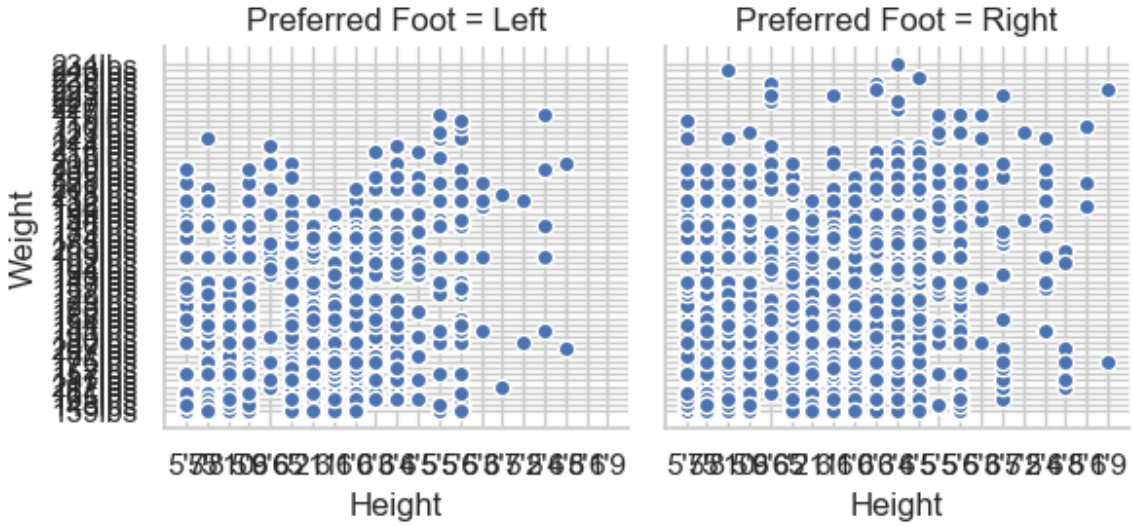
In [77]:
```
# we can draw a univariate plot of potential variable on each facet as follows-

g = sns.FacetGrid(fifa19, col="Preferred Foot")
g = g.map(plt.hist, "Potential")
```



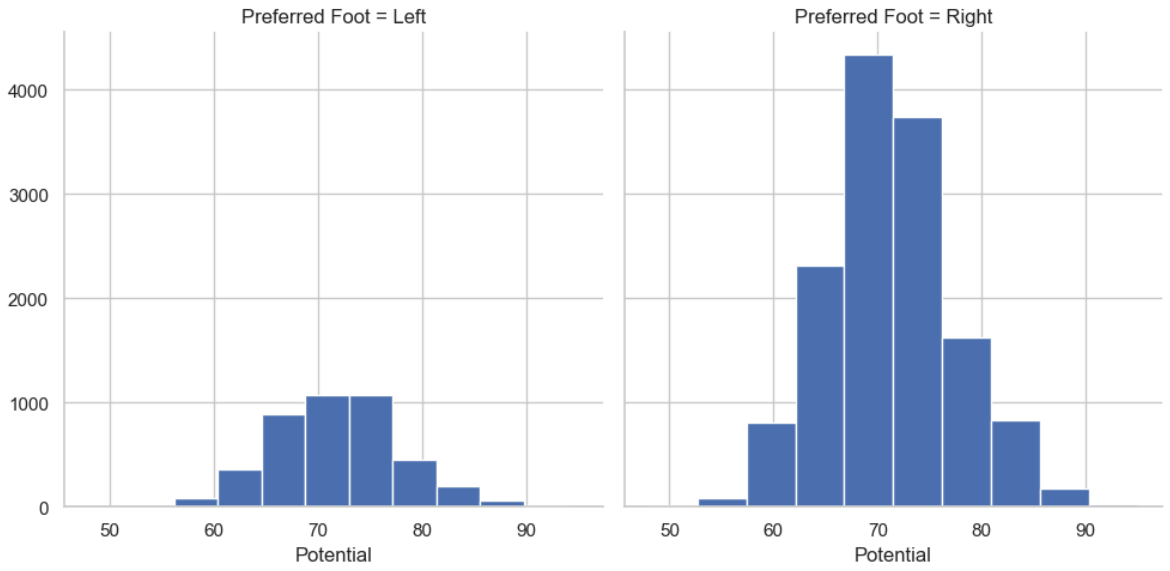
In [78]:
```
g = sns.FacetGrid(fifa19, col="Preferred Foot")
g = g.map(plt.hist, "Potential", bins=10, color="r")
```



In [79]:
```
# We can plot a bivariate function on each facet as follows-
g = sns.FacetGrid(fifa19, col="Preferred Foot")
g = (g.map(plt.scatter, "Height", "Weight", edgecolor="w").add_legend())
```

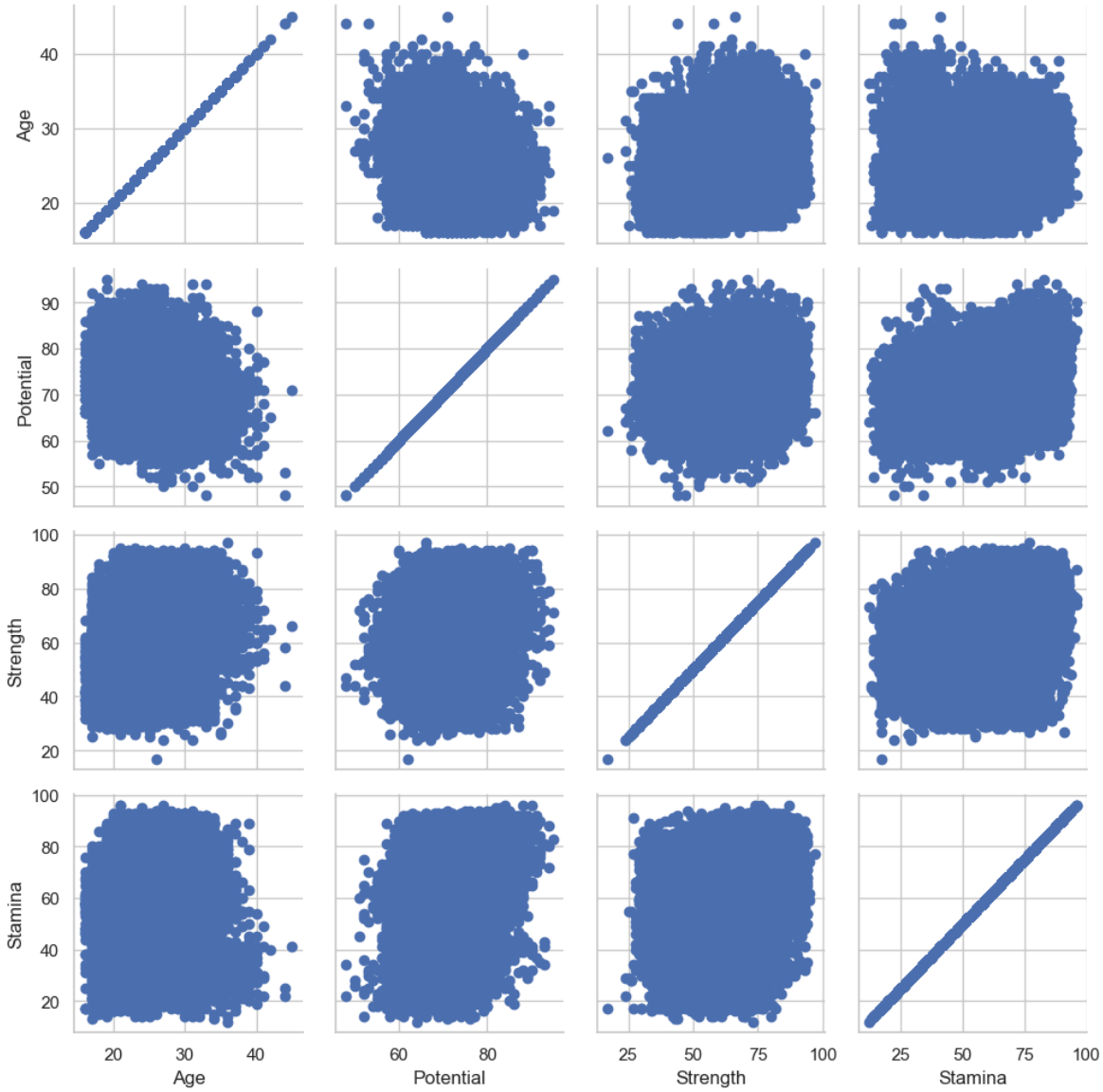


In [80]:
```
g = sns.FacetGrid(fifa19, col="Preferred Foot", height=5, aspect=1)
g = g.map(plt.hist, "Potential")
```

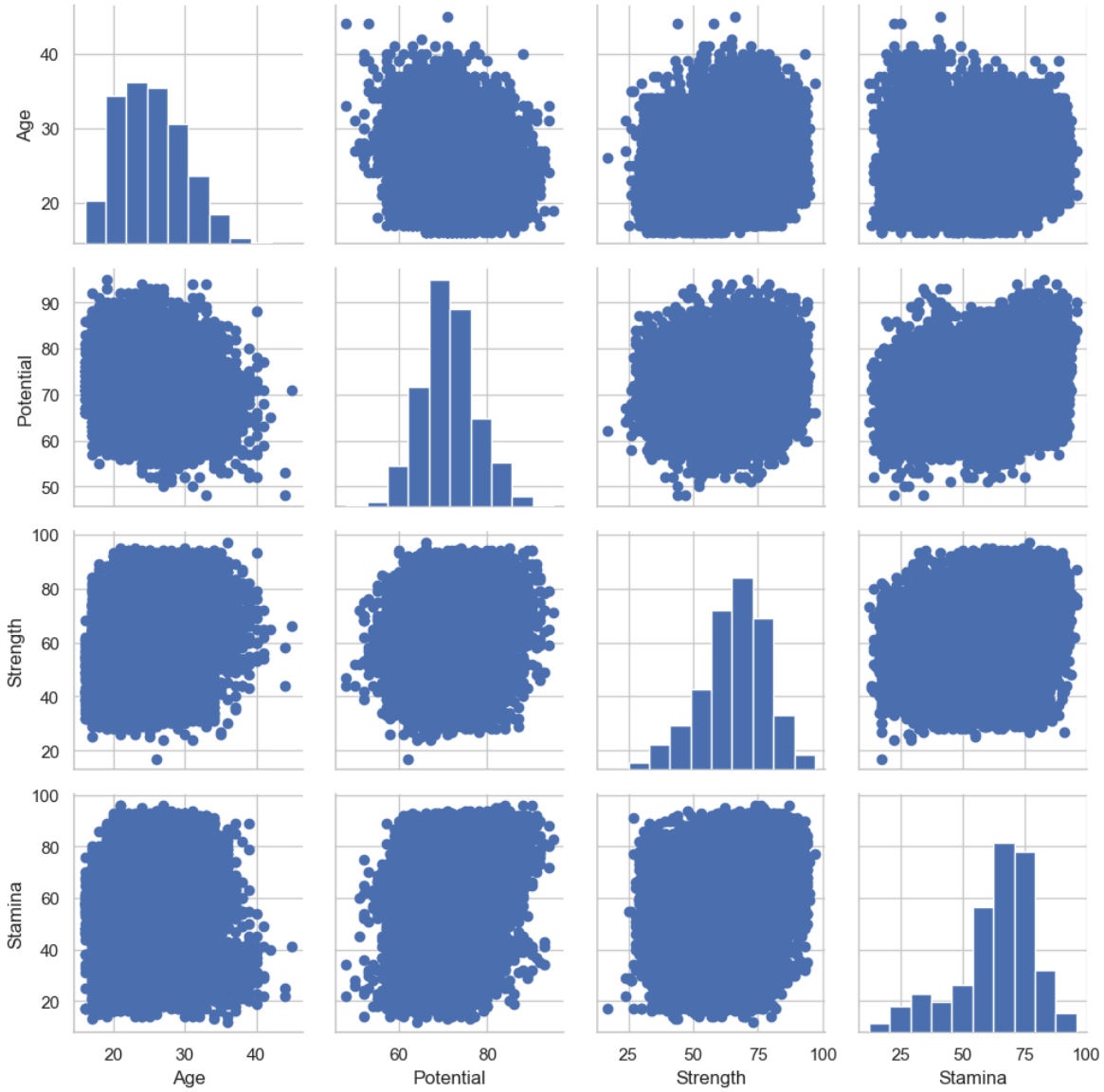


In [81]:
```
# Seaborn Pairgrid()
fifa19_new = fifa19[['Age', 'Potential', 'Strength', 'Stamina', 'Preferred Foot'
```

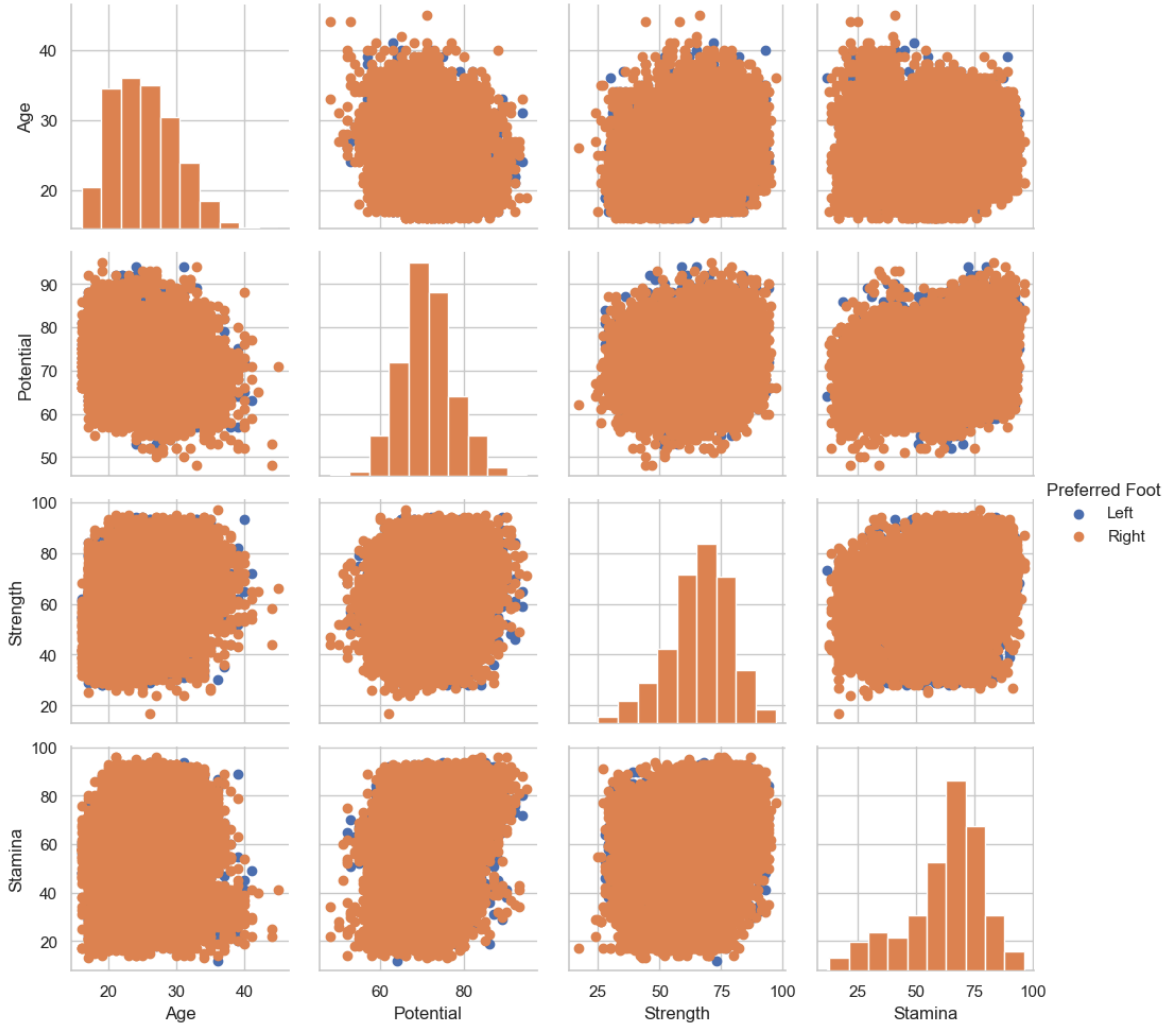
In [82]:
```
g = sns.PairGrid(fifa19_new)
g = g.map(plt.scatter)
```



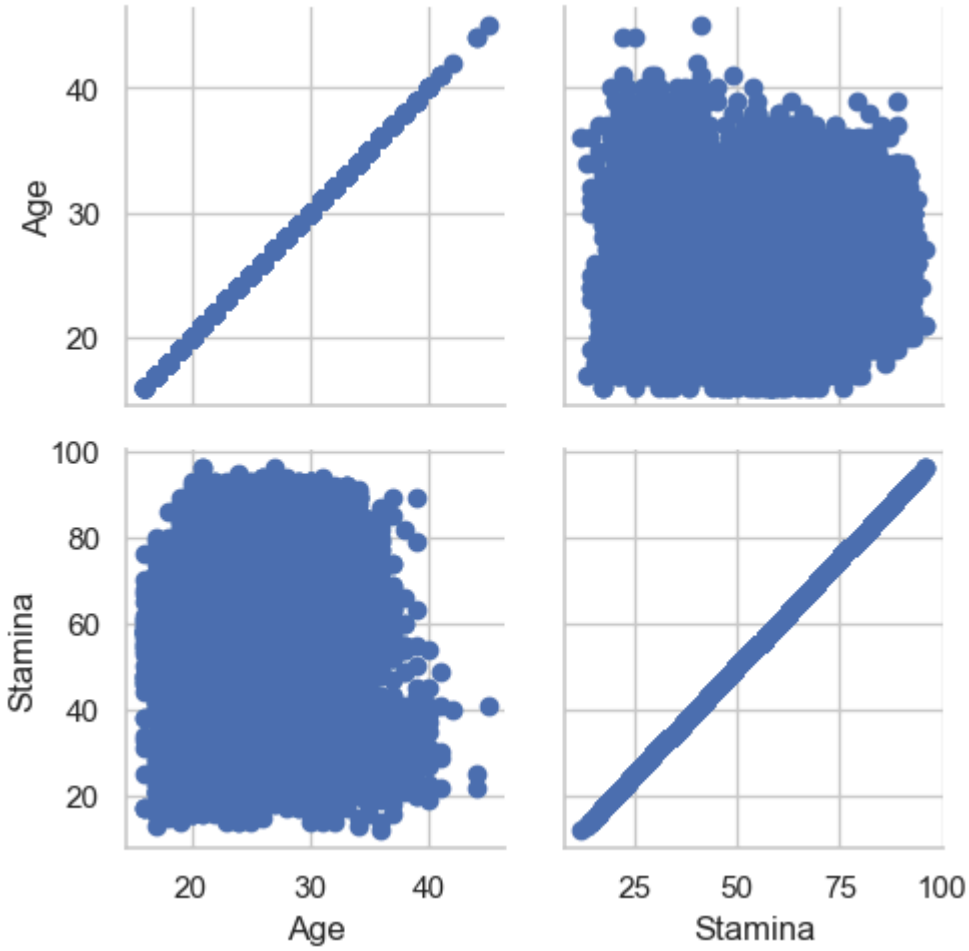
In [83]:
```
# we can show a univariate distribution on the diagonal as follows-

g = sns.PairGrid(fifa19_new)
g = g.map_diag(plt.hist)
g = g.map_offdiag(plt.scatter)
```



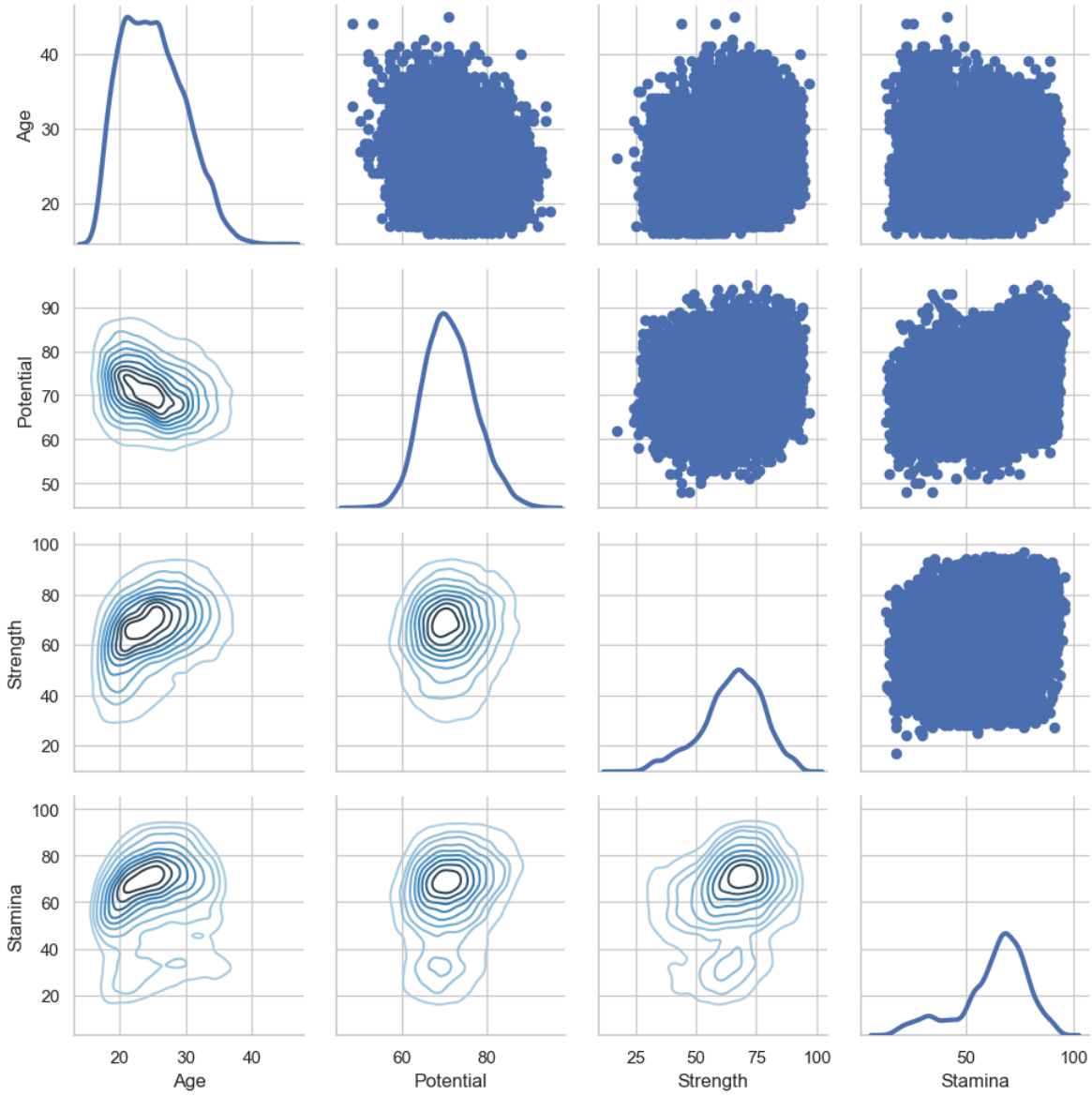
In [84]:
```
# we can color the points using the categorical variable Preferred Foot as follo

g = sns.PairGrid(fifa19_new, hue="Preferred Foot")
g = g.map_diag(plt.hist)
g = g.map_offdiag(plt.scatter)
g = g.add_legend()
```



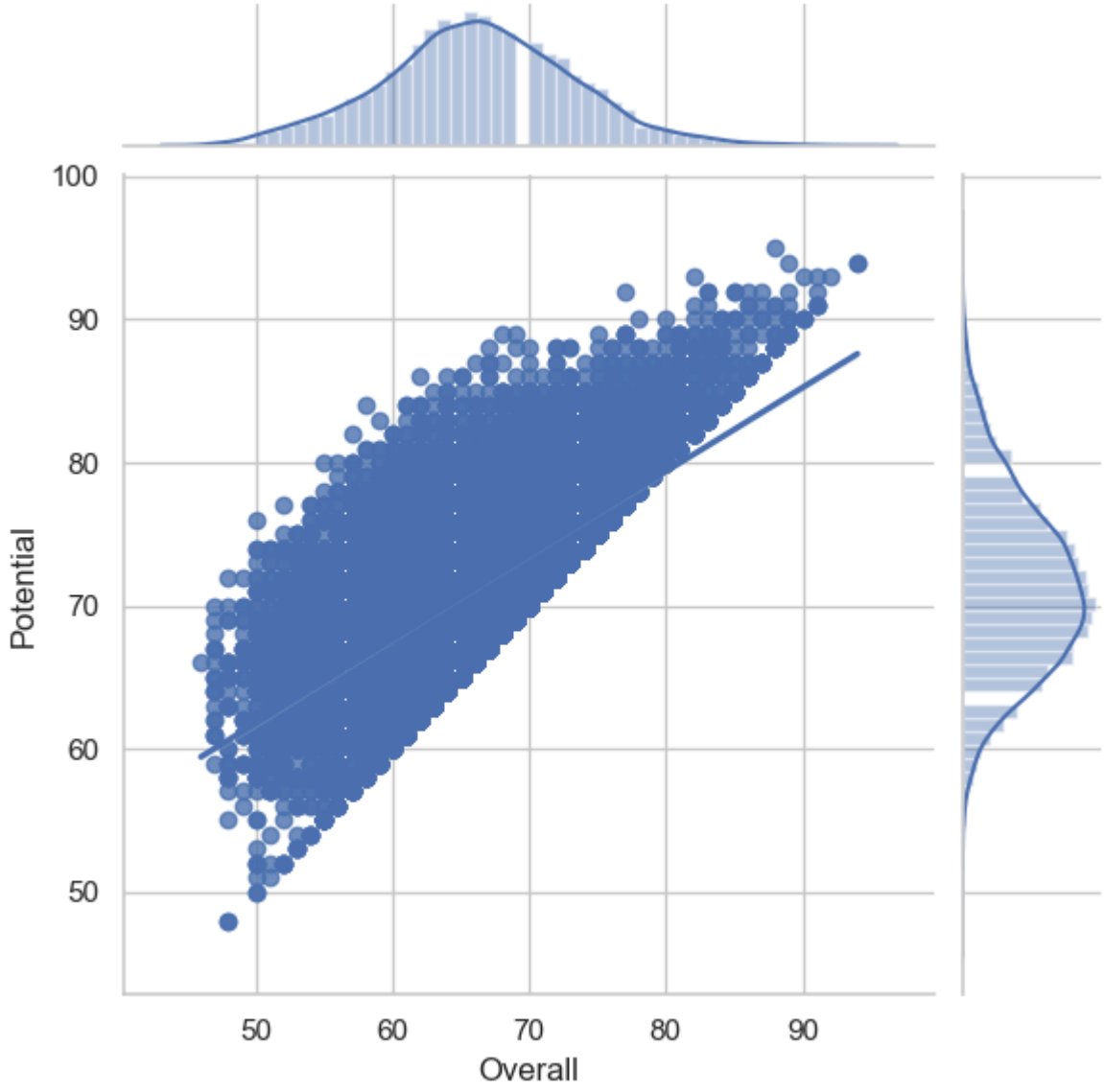
In [85]:
```
# we can plot a subset of variables as follows-

g = sns.PairGrid(fifa19_new, vars=['Age', 'Stamina'])
g = g.map(plt.scatter)
```



In [86]:
```
# We can use different functions on the upper and lower triangles as follows:

g = sns.PairGrid(fifa19_new)
g = g.map_upper(plt.scatter)
g = g.map_lower(sns.kdeplot, cmap="Blues_d")
g = g.map_diag(sns.kdeplot, lw=3, legend=False)
```

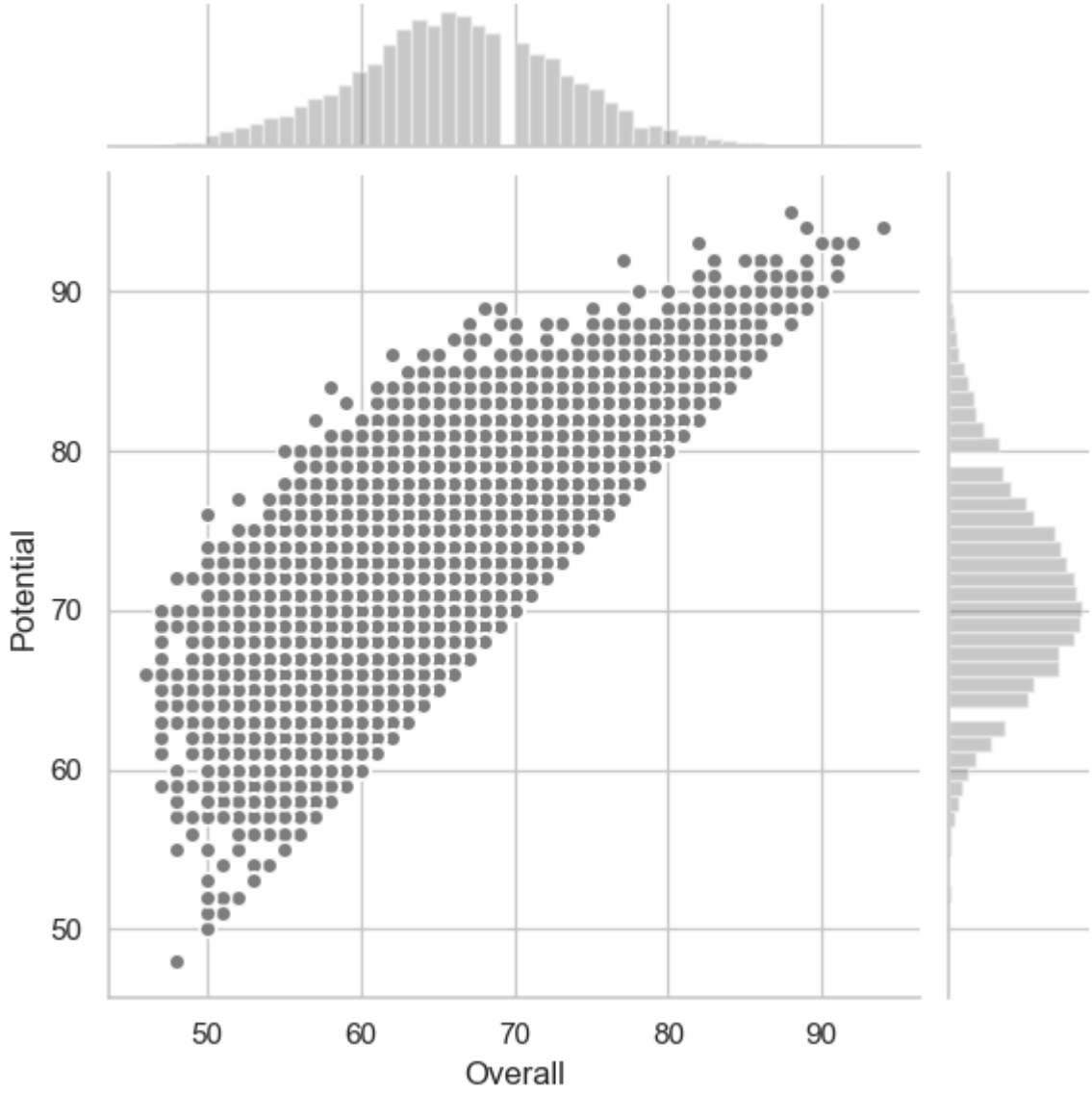


In [87]:
```
# Seaborn Jointgrid()

g = sns.JointGrid(x="Overall", y="Potential", data=fifa19)
g = g.plot(sns.regplot, sns.distplot)
```

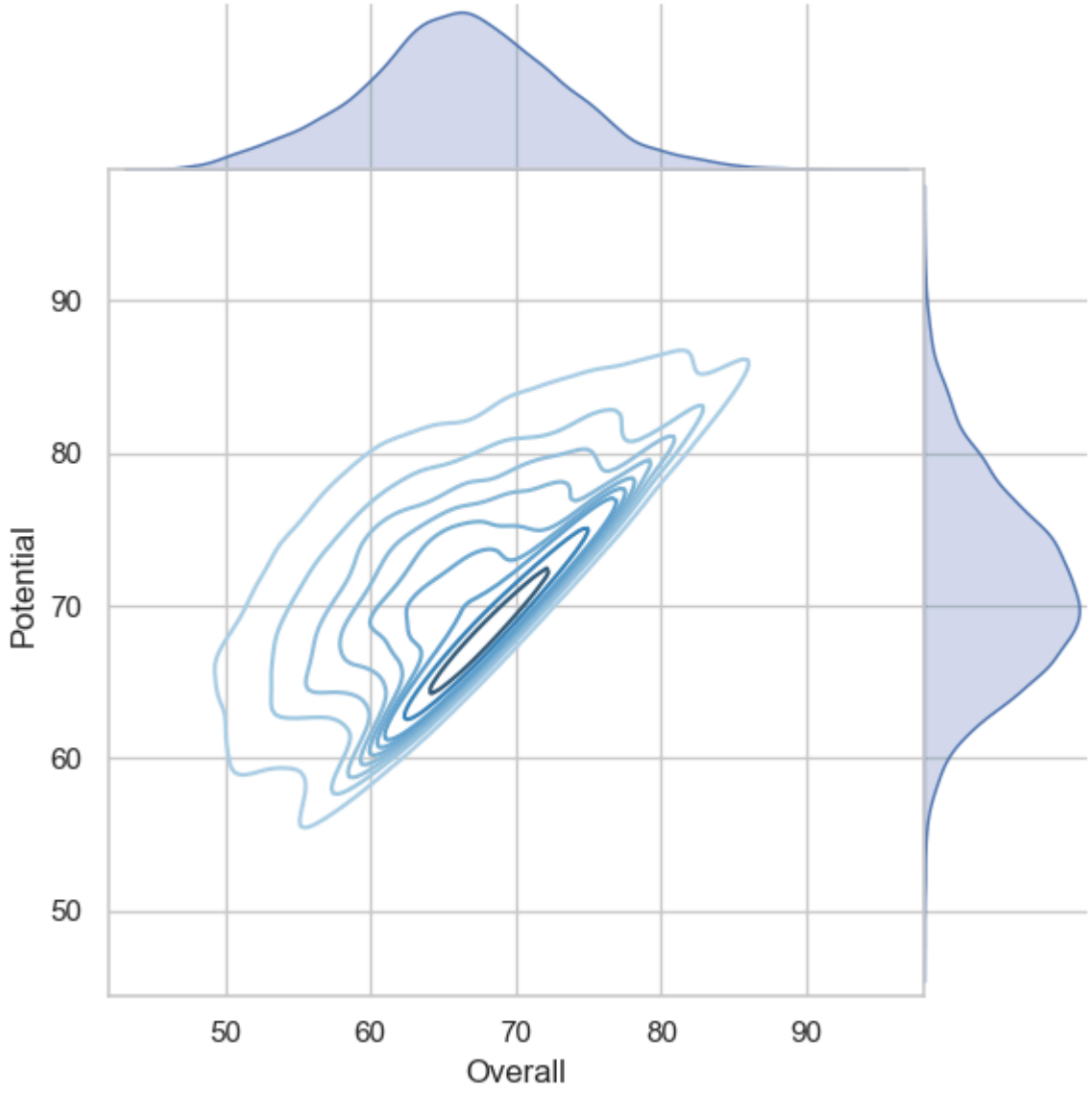


In [89]:
```
import matplotlib.pyplot as plt
```

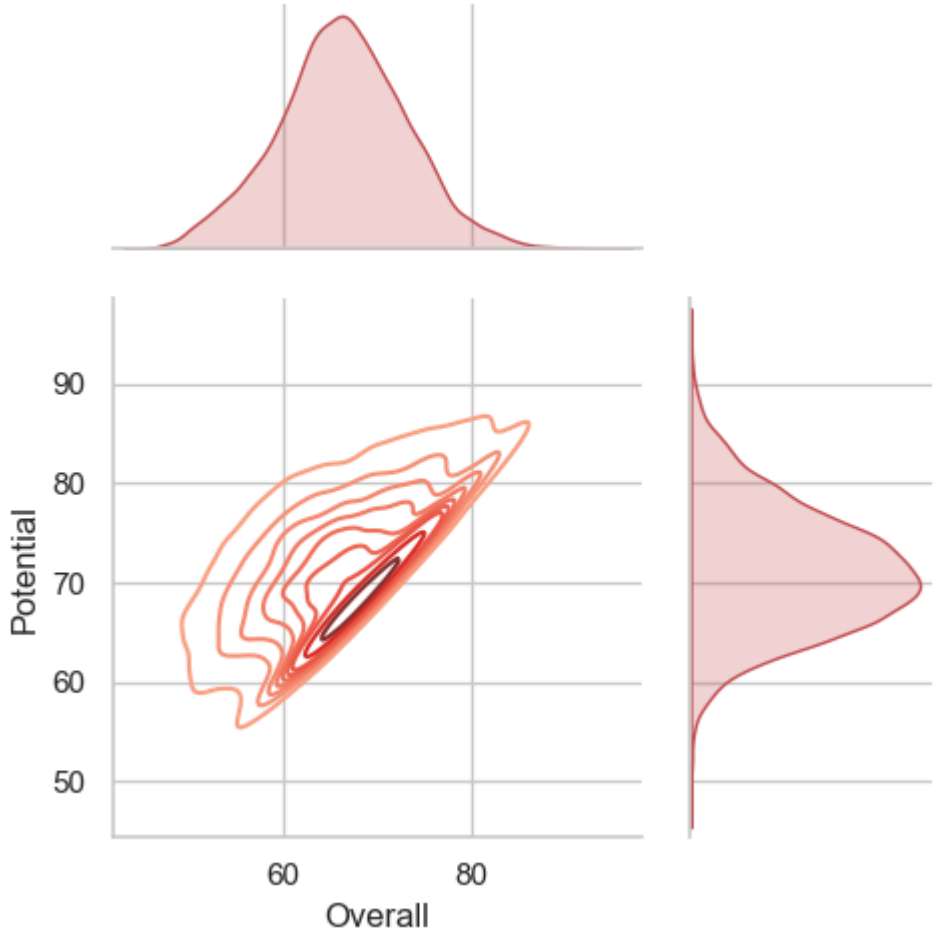
In [90]:
```
g = sns.JointGrid(x="Overall", y="Potential", data=fifa19)
g = g.plot_joint(plt.scatter, color=".5", edgecolor="white")
g = g.plot_marginals(sns.distplot, kde=False, color=".5")
```



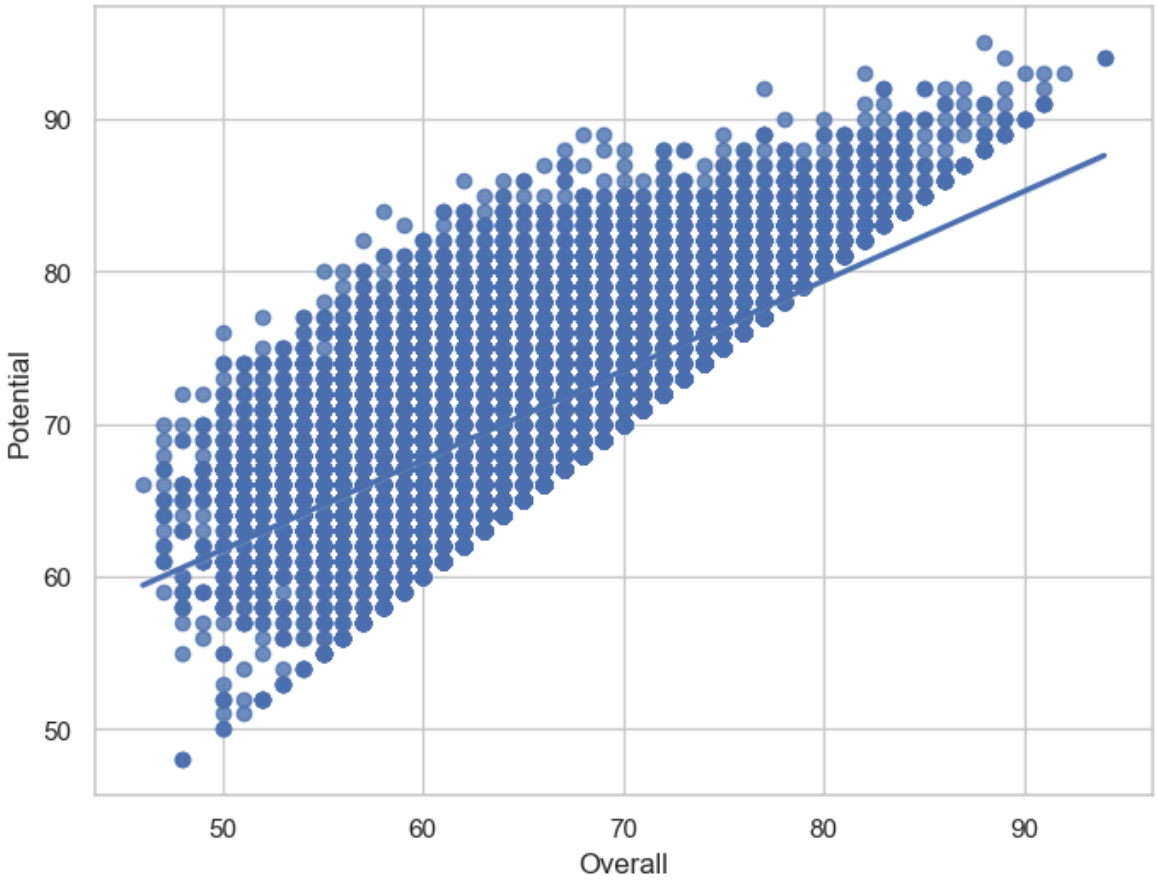
In [91]:
```
# we can remove the space between the joint and marginal axes as follows-

g = sns.JointGrid(x="Overall", y="Potential", data=fifa19, space=0)
g = g.plot_joint(sns.kdeplot, cmap="Blues_d")
g = g.plot_marginals(sns.kdeplot, shade=True)
```



In [92]:
```
g = sns.JointGrid(x="Overall", y="Potential", data=fifa19, height=5, ratio=2)
g = g.plot_joint(sns.kdeplot, cmap="Reds_d")
g = g.plot_marginals(sns.kdeplot, color="r", shade=True)
```

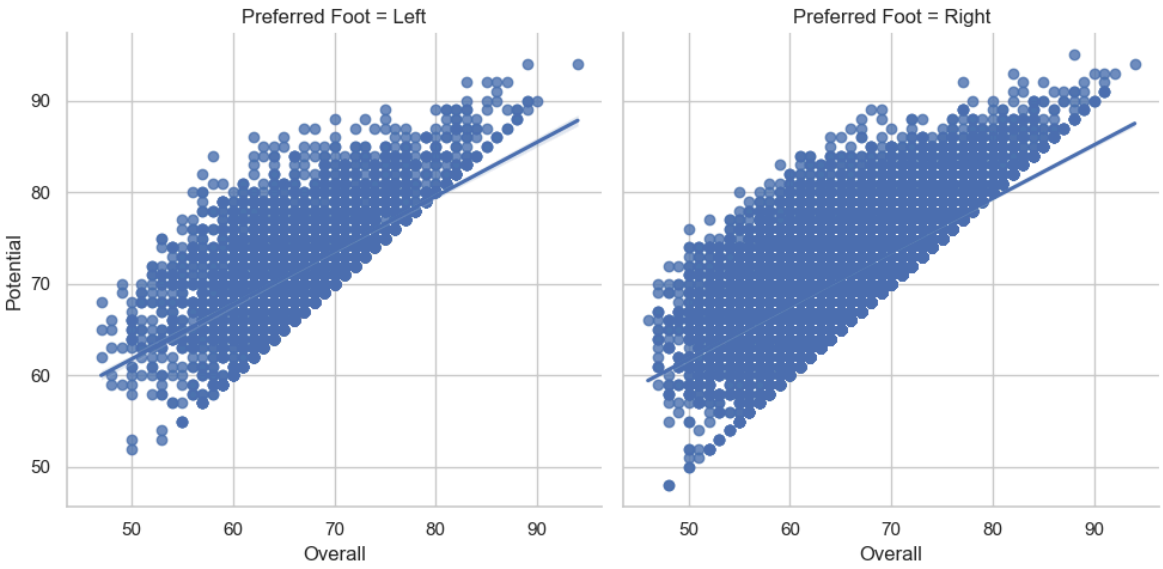


In [93]:
```
f,ax = plt.subplots(figsize=(8,6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa19);
```



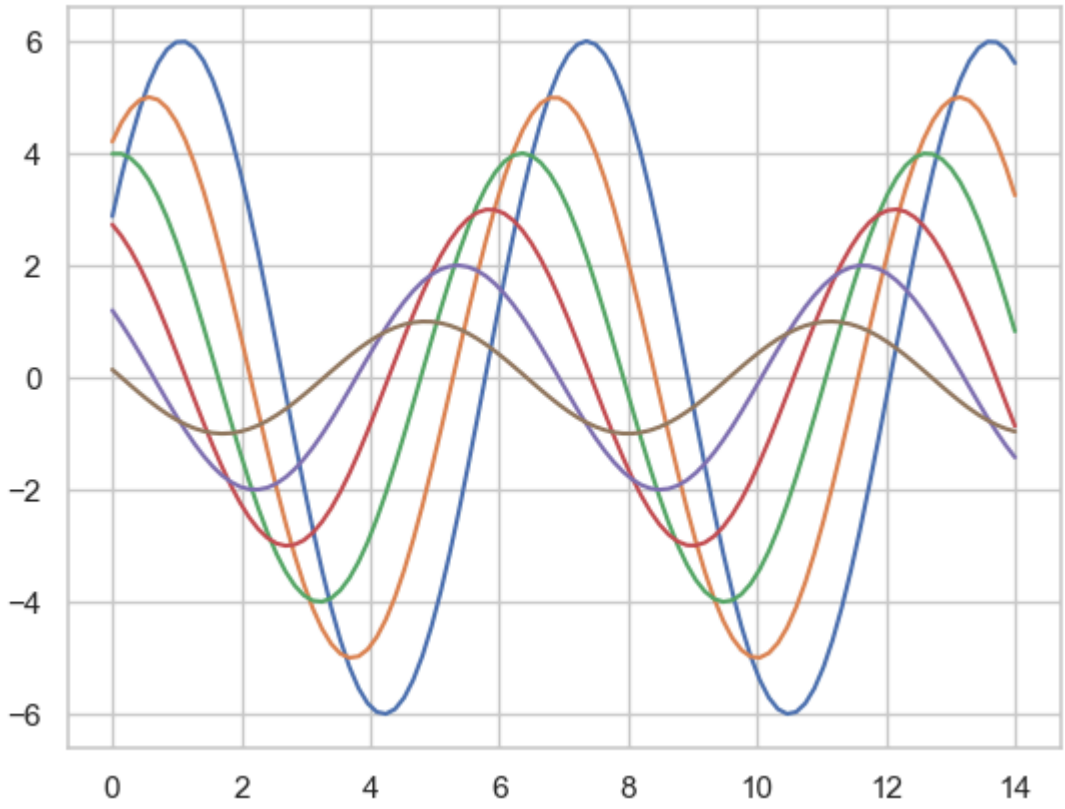
In [94]:
```
sns.lmplot(x="Overall", y="Potential", col="Preferred Foot", data=fifa19, col_wr
```

Out[94]: <seaborn.axisgrid.FacetGrid at 0x1d74fbda250>



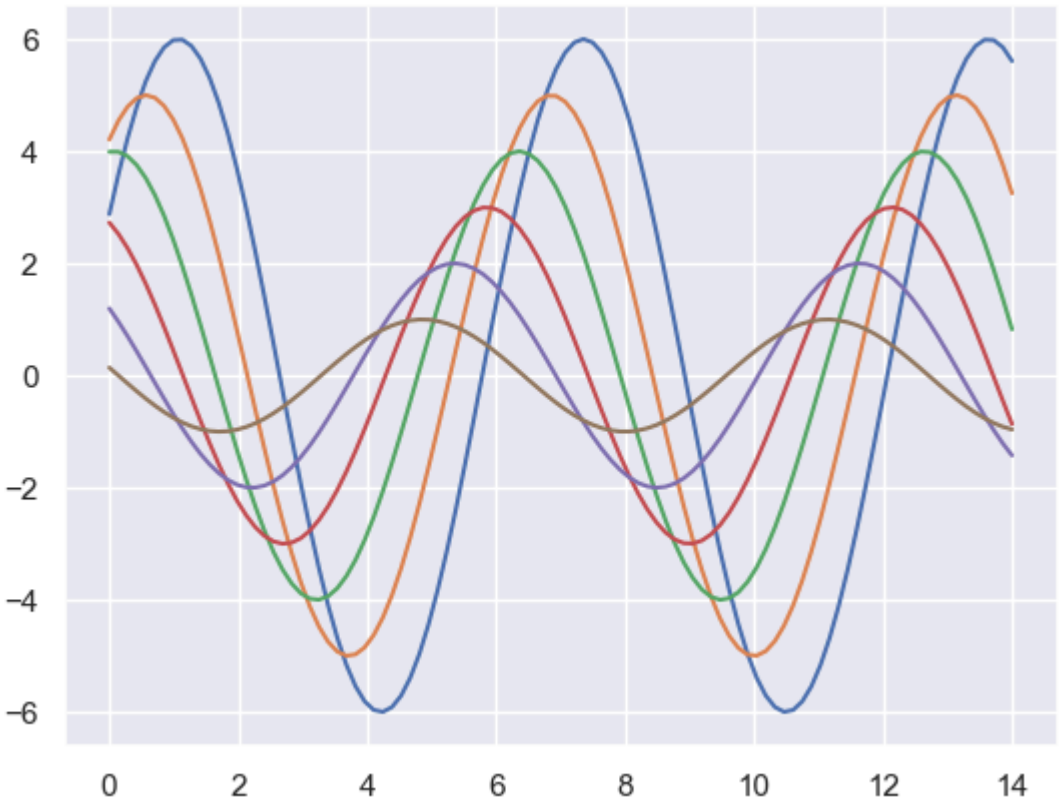
In [95]:
```
# Seaborn figure styles
def sinplot(flip=1):
    x = np.linspace(0, 14, 100)
    for i in range(1, 7):
        plt.plot(x, np.sin(x + i * .5) * (7 - i) * flip)
```

In [96]:
```
# This is what the plot looks like with matplotlib default parameters.

sinplot()
```

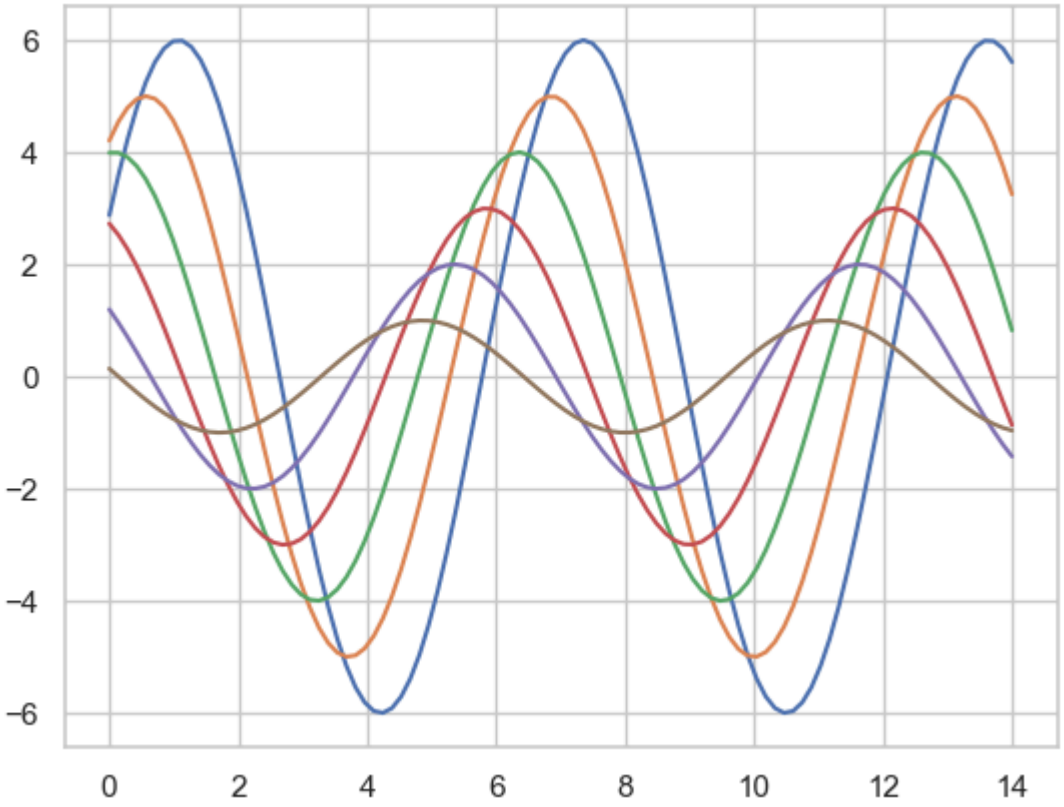


In [97]:
```
# To switch to seaborn deafaults, we need to call the set() function as follows-
```

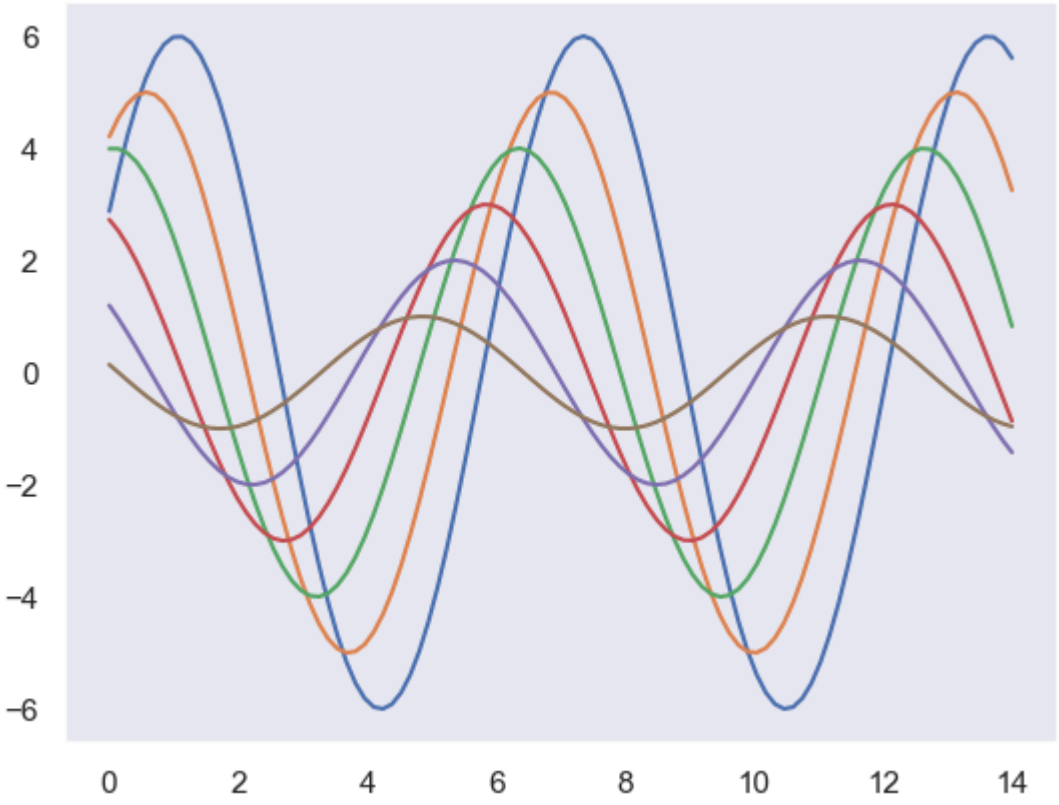
```
sns.set()
sinplot()
```



In [98]:
```
# we can set different styles as follows-

sns.set_style("whitegrid")
sinplot()
```



In [99]:
```
sns.set_style("dark")
sinplot()
```



In [ ]:
```
sns.set_style("whitegrid")
sinplot()
```