SEABORN--FIFA Data Analysis Visualization

Table of Contents The table of contents for this tutorial is as follows - - Import libraries - Read dataset - Exploratory data analysis - Visualize distribution of Age variable with Seaborn distplot() function - Seaborn Kernel Density Estimation (KDE) plot - Histograms - Visualize distribution of values in Preferred Foot variable with Seaborn countplot() function - Seaborn catplot() function - Seaborn stripplot() function - Seaborn boxplot() function - Seaborn violinplot() function - Seaborn pointplot() function - Seaborn barplot() function - Visualizing statistical relationship with Seaborn relplot() function - Seaborn scatterplot() function - Seaborn lineplot() function - Seaborn regplot() function - Seaborn lmplot() function - Multi-plot grids - Seaborn Facetgrid() function - Seaborn Pairgrid() function - Seaborn Jointgrid() function - Controlling the size and shape of the plot - Seaborn figure styles

In [1]:
```
#Import Libraries
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
sns.set(style="whitegrid")
import matplotlib.pyplot as plt
from collections import Counter
%matplotlib inline
```

In [11]:
```
import os
for dirname, _, filenames in os.walk('E:\Data Science & AI\Dataset files\FIFA.cs
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [5]:
```
# ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

In [13]:
```
#Read Dataset
fifa = pd.read_csv(r'E:\Data Science & AI\Dataset files\FIFA.csv', index_col=0)
```

In [15]:
```
#Preview Dataset
fifa.head()
```

Out[15]:

	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]:
```
fifa
```

Out[19]:

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

18207 rows × 88 columns

In [21]:
```
# View summary of dataset
fifa.info()
```

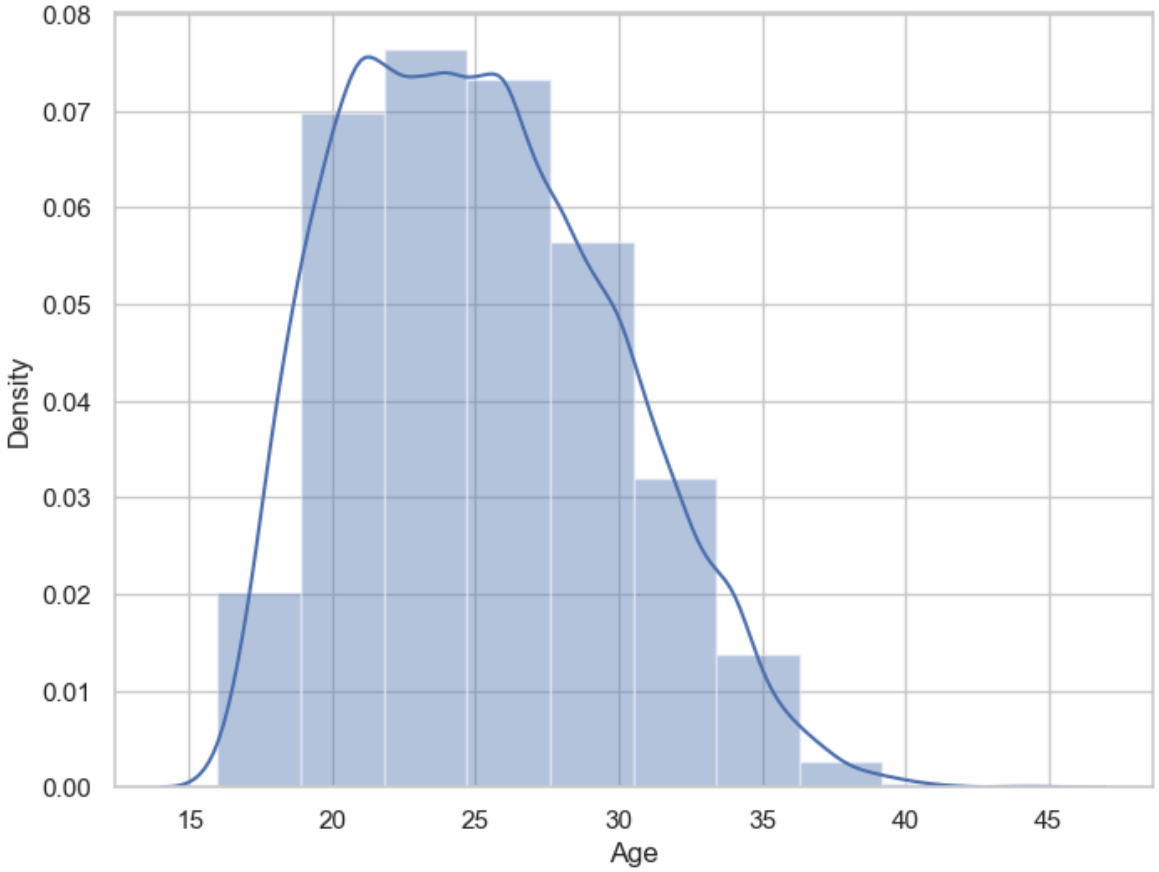
```
<class 'pandas.core.frame.DataFrame'>
Index: 18207 entries, 0 to 18206
Data columns (total 88 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   ID                        18207 non-null  int64
 1   Name                      18207 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: 12.4+ MB
```

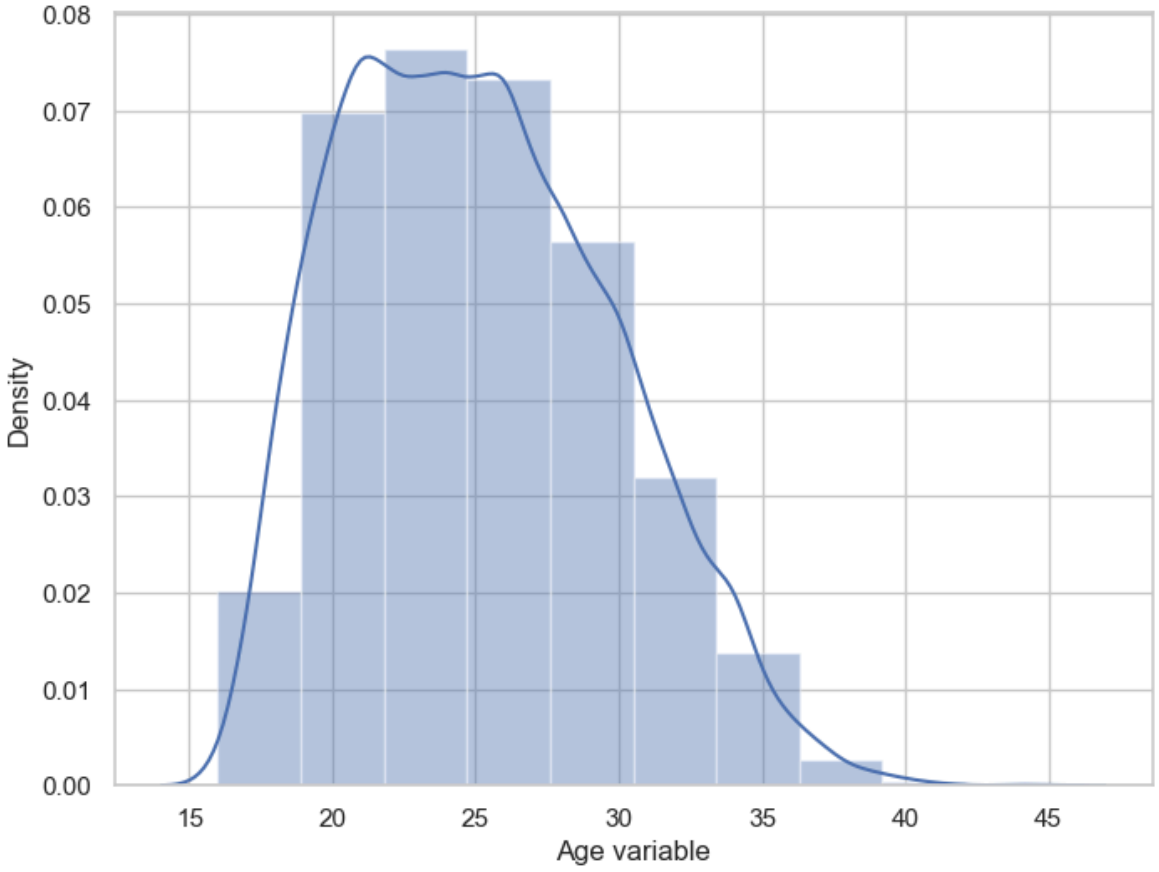
In [23]:
```
fifa['Body Type'].value_counts()
```

Out[23]:
```
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
```

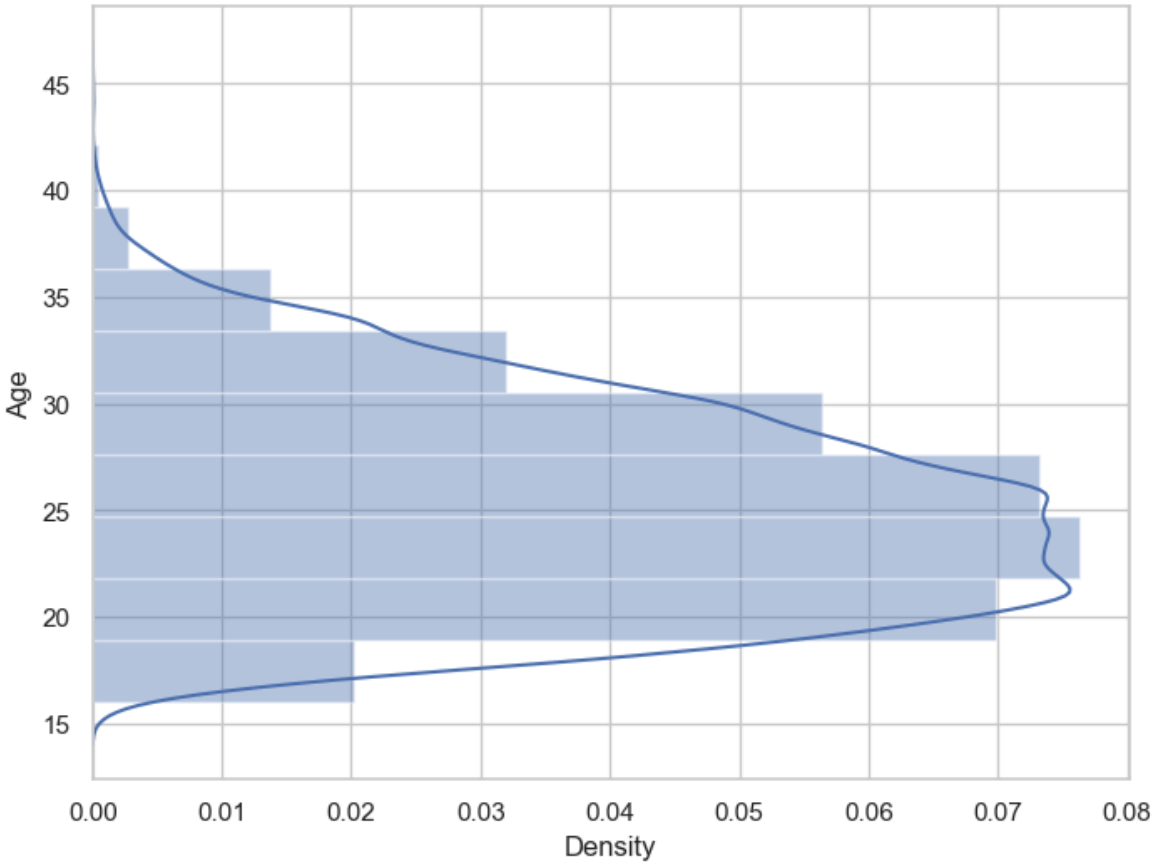
In [27]:
```
# Visualize distribution of `Age` variable with Seaborn `distplot()` function

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



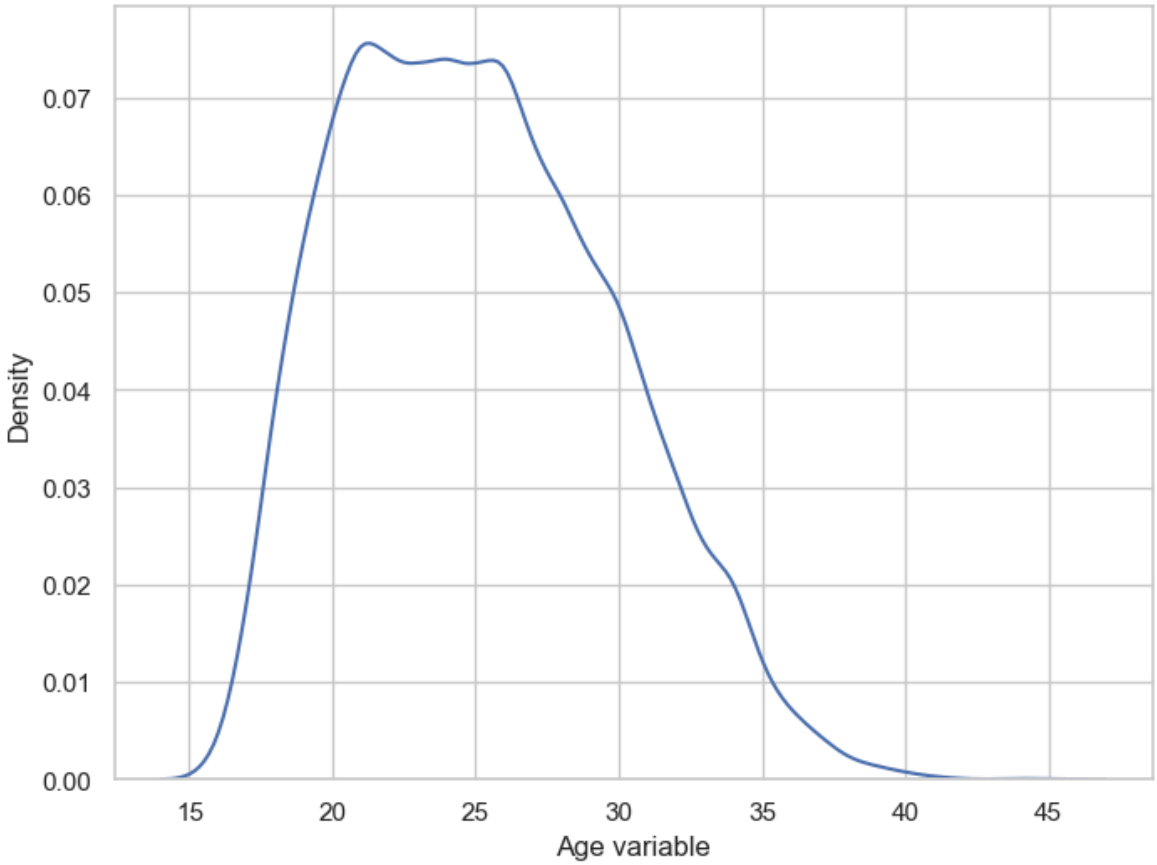
In [29]:
```
f, ax = plt.subplots(figsize=(8,6))
x = fifa['Age']
x = pd.Series(x, name="Age variable")
ax = sns.distplot(x, bins=10)
plt.show()
```



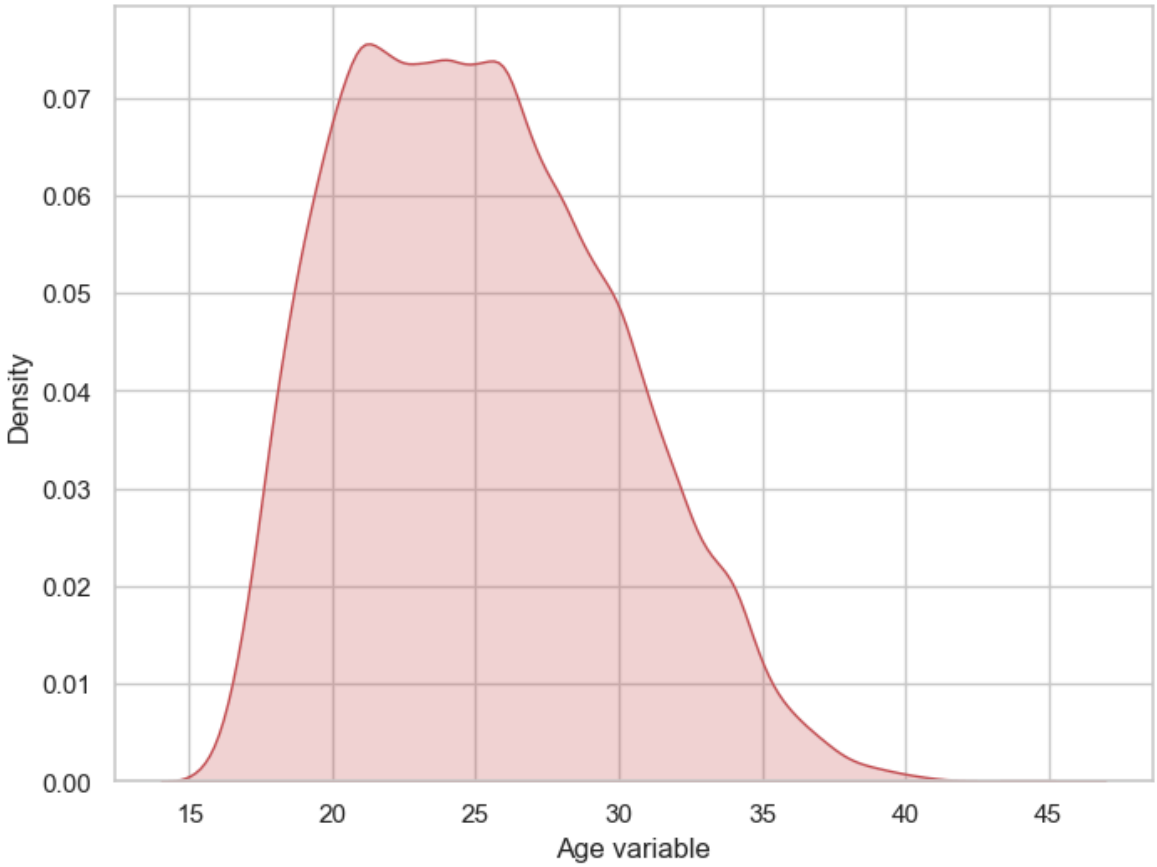
In [35]:
```
f, ax = plt.subplots(figsize=(8,6))
x = fifa['Age']
ax = sns.distplot(x, bins=10, vertical = True) #Vertical graph
plt.show()
```



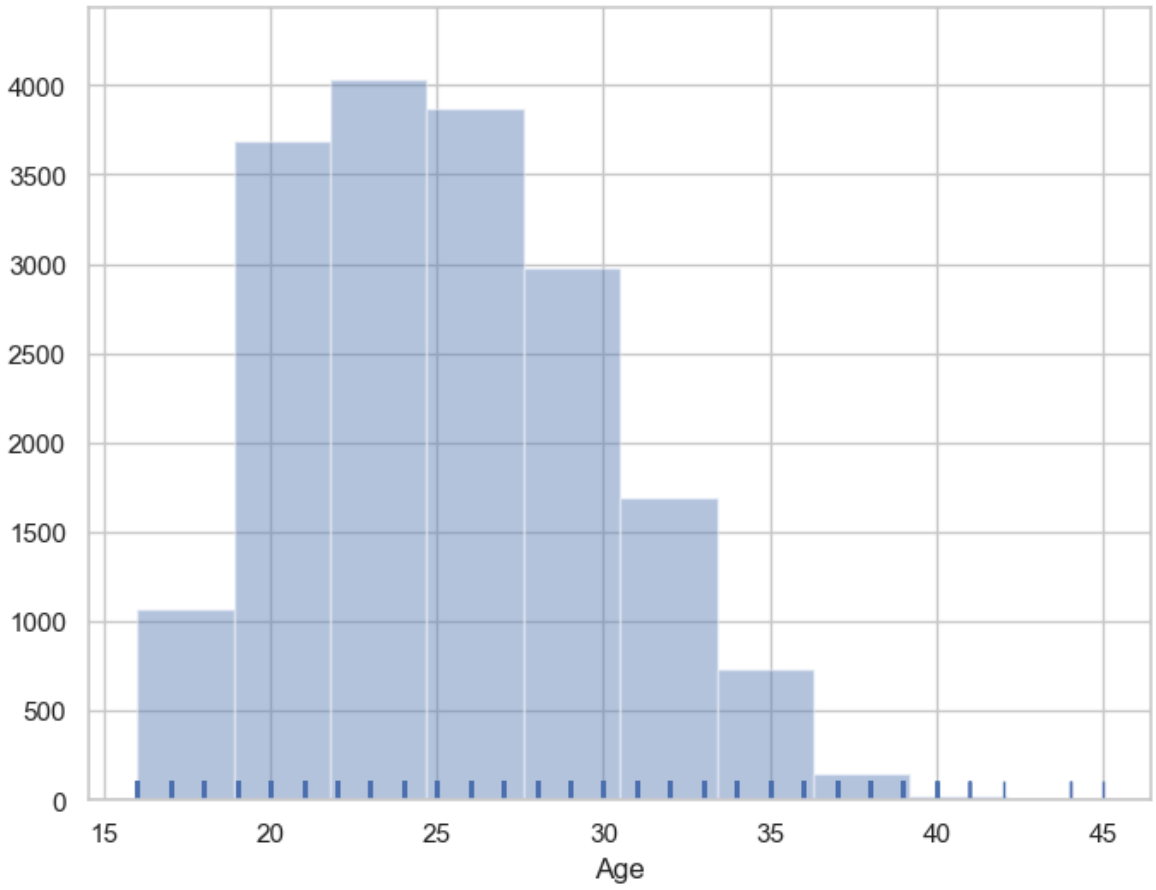
In [37]:
```
# Seaborn Kernel Density Estimation (KDE) Plot

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



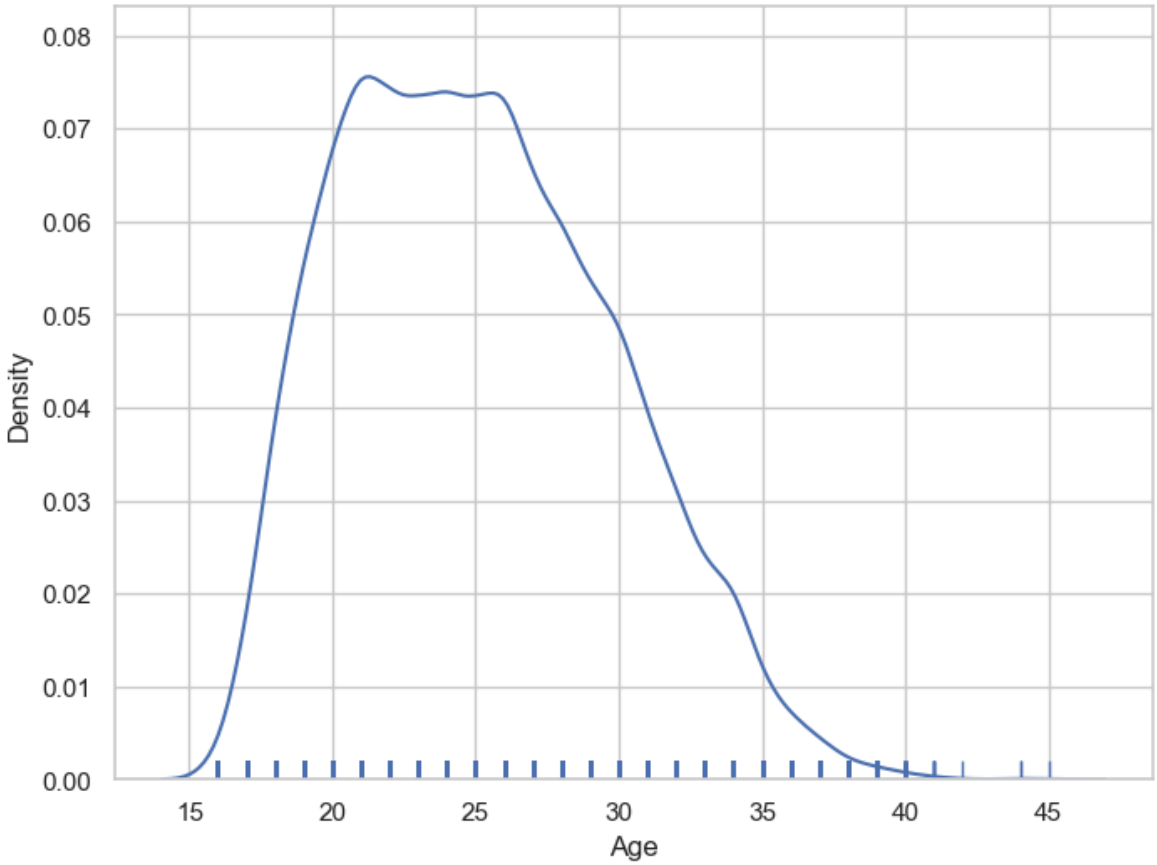
In [41]:
```
f, ax = plt.subplots(figsize=(8,6)) #Density Curve & Using a Differnt Colour
x = fifa['Age']
x = pd.Series(x, name="Age variable")
ax = sns.kdeplot(x, shade=True, color='r')
plt.show()
```



In [43]:
```
# Histograms--hist()

f, ax = plt.subplots(figsize=(8,6))
x = fifa['Age']
ax = sns.distplot(x, kde=False, rug=True, bins=10)
plt.show()
```



In [45]:
```
f, ax = plt.subplots(figsize=(8,6)) #KDE plot Alternatively
x = fifa['Age']
ax = sns.distplot(x, hist=False, rug=True, bins=10)
plt.show()
```



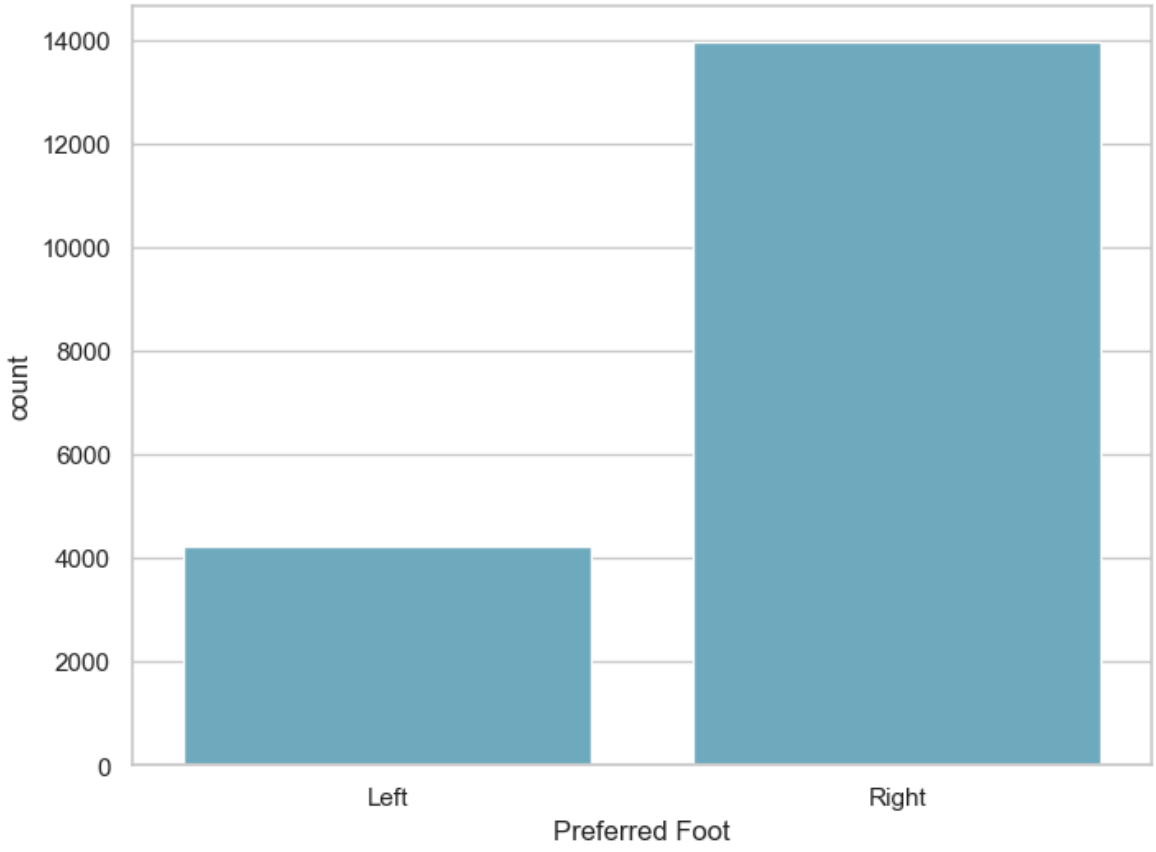
In [47]:
```
# Preferred Foot Variable
fifa['Preferred Foot'].nunique()
```

Out[47]:
```
2
```

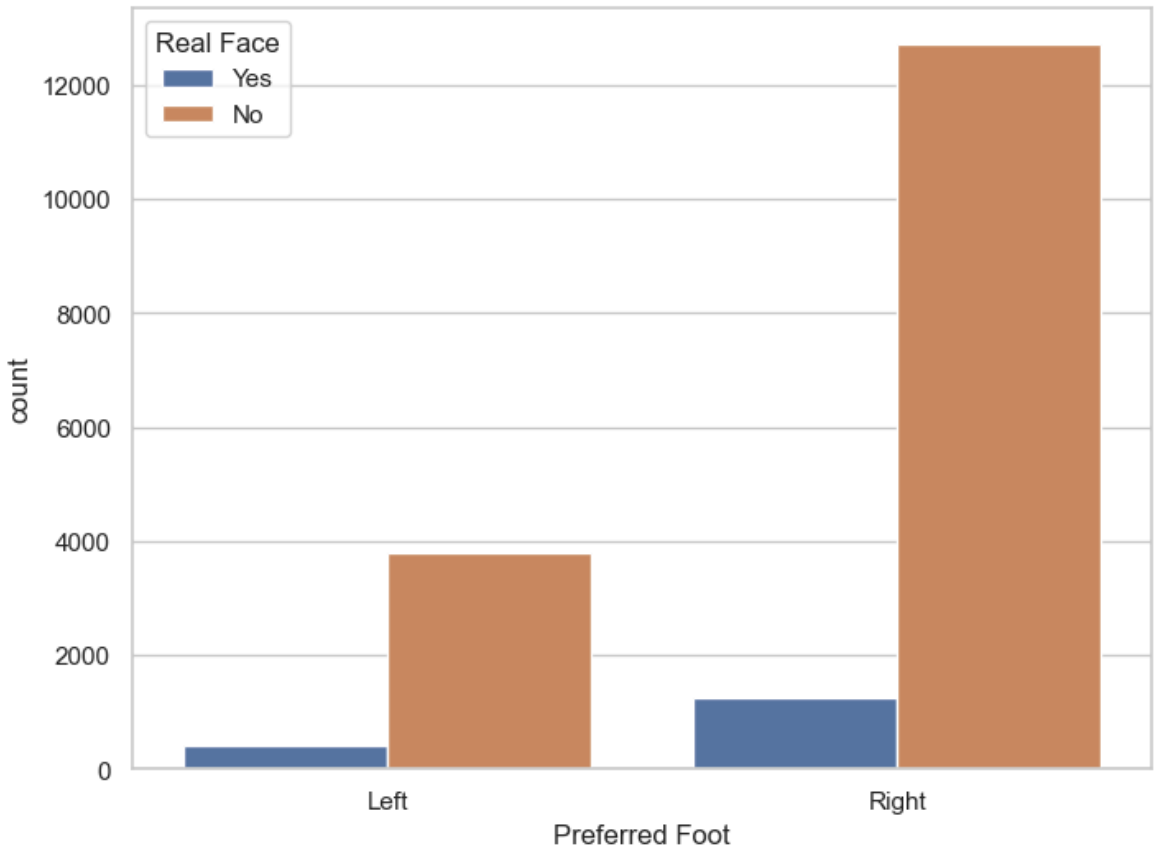
In [49]:
```
# Distribution of values--Preferred Foot Variable

fifa['Preferred Foot'].value_counts()
```

Out[49]:
```
Preferred Foot
Right    13948
Left      4211
Name: count, dtype: int64
```

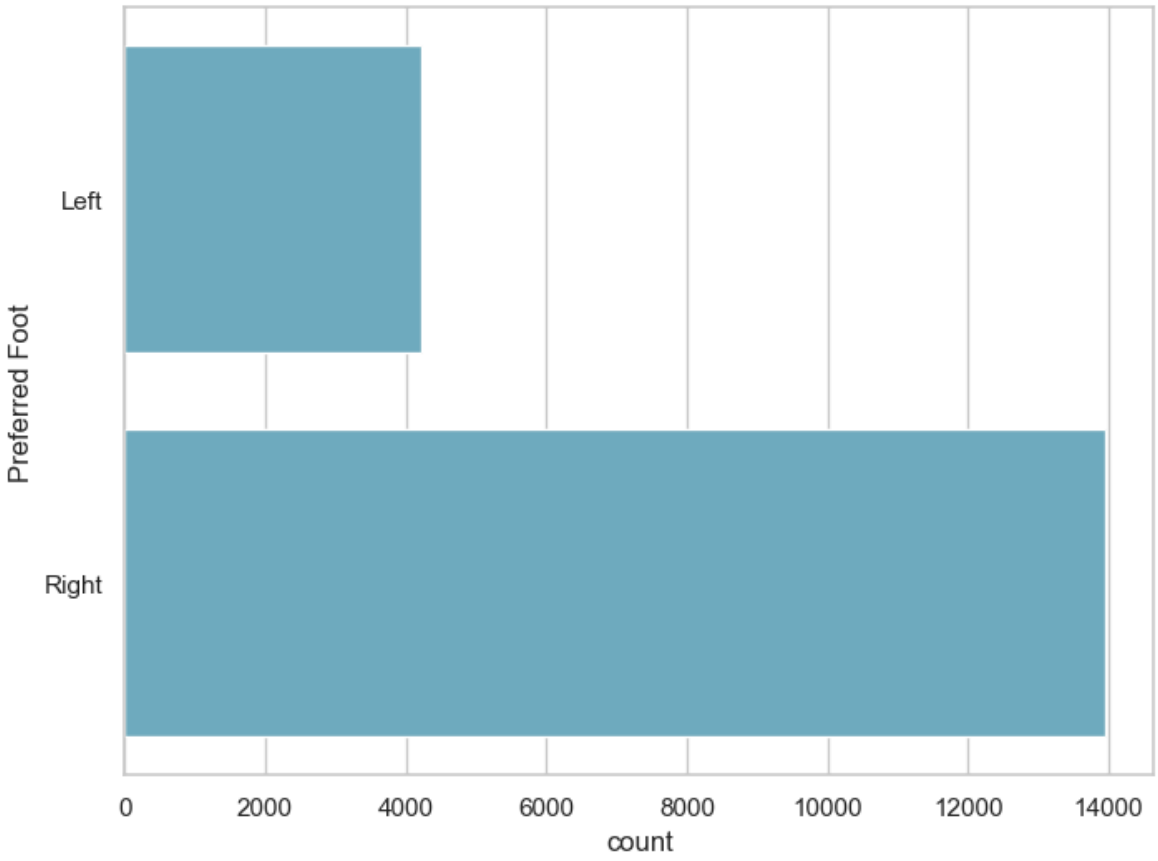
In [51]:
```
#countplot() function

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



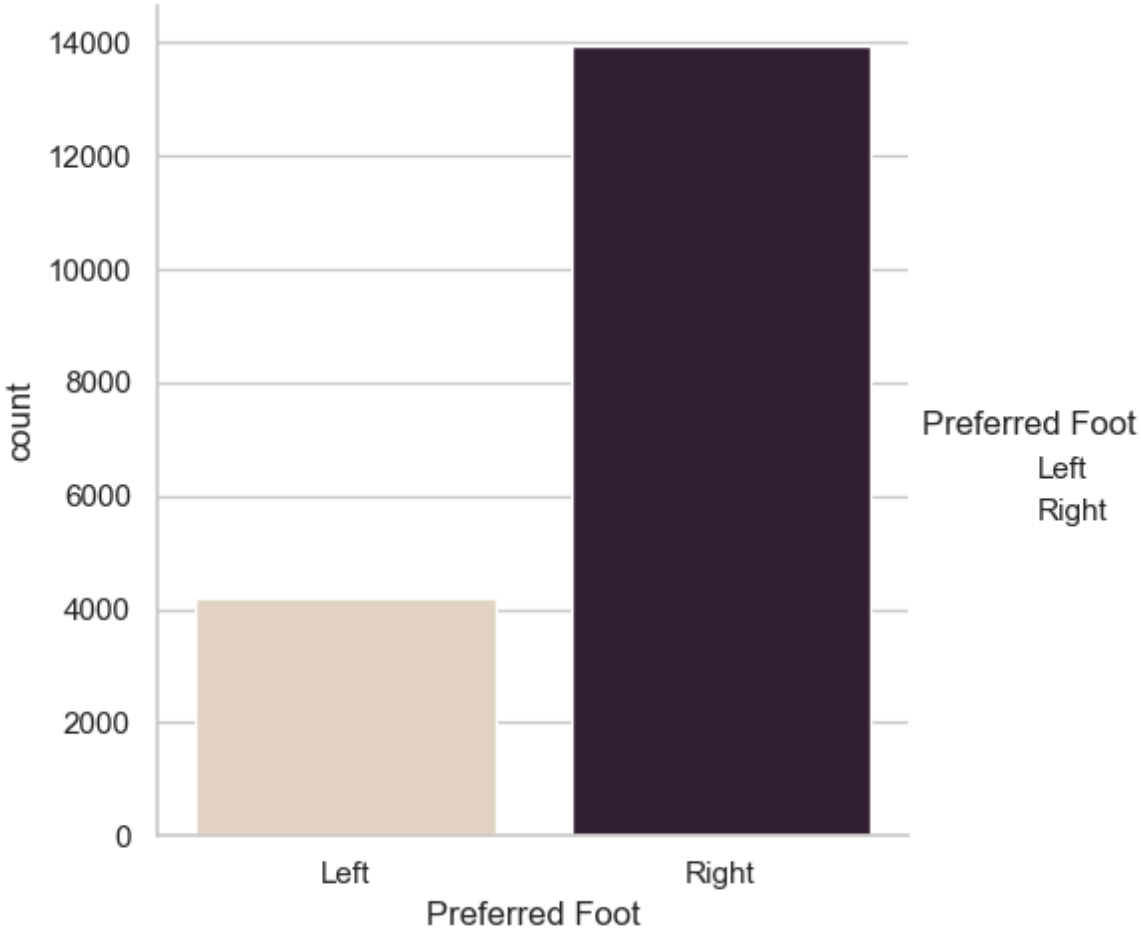
In [53]:
```
#value counts for two categorical variables

f, ax = plt.subplots(figsize=(8, 6))
sns.countplot(x="Preferred Foot", hue="Real Face", data=fifa)
plt.show()
```



In [55]:
```
#Plot Vertically

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



In [57]:
```
# Seaborn catplot() function to draw a countplot()

g = sns.catplot(x="Preferred Foot", kind="count", palette="ch:.25", data=fifa)
```



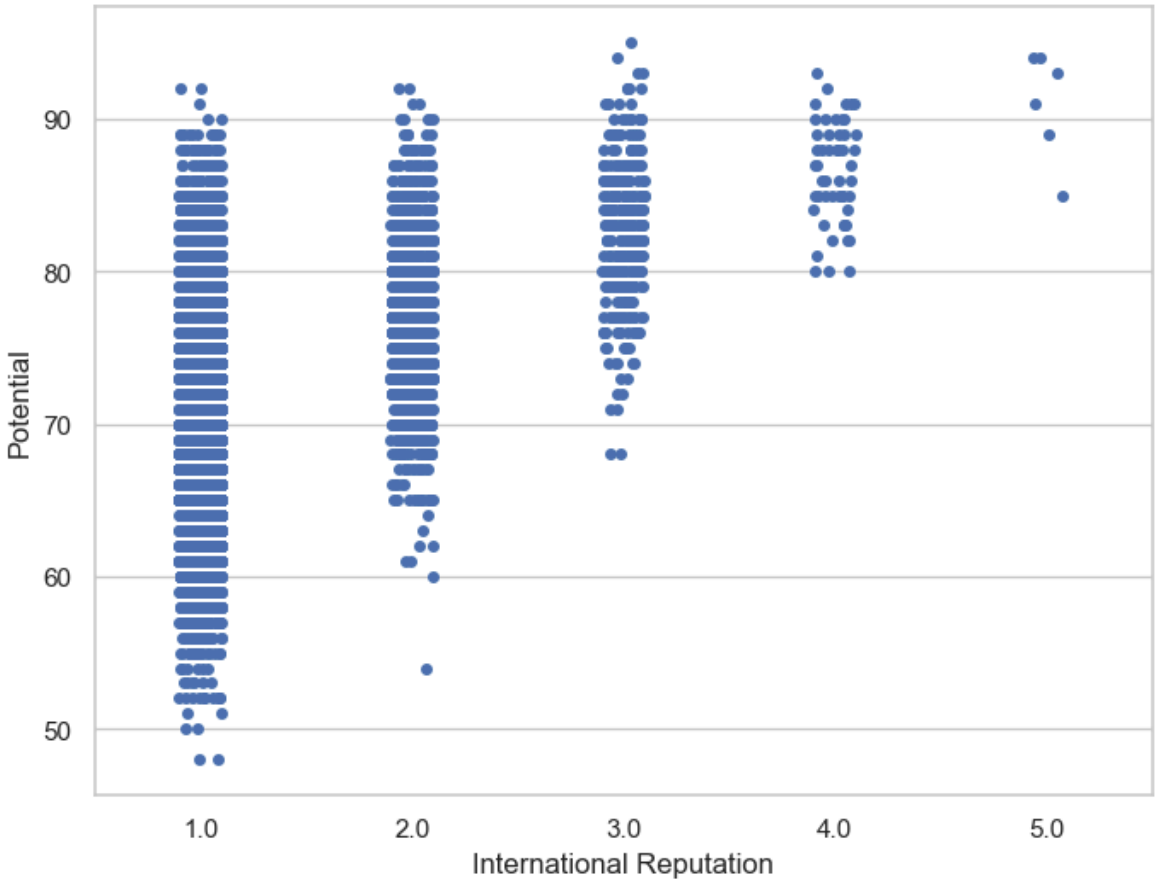
In [59]:
```
# International Reputation variable

fifa['International Reputation'].nunique()
```

Out[59]:
```
5
```

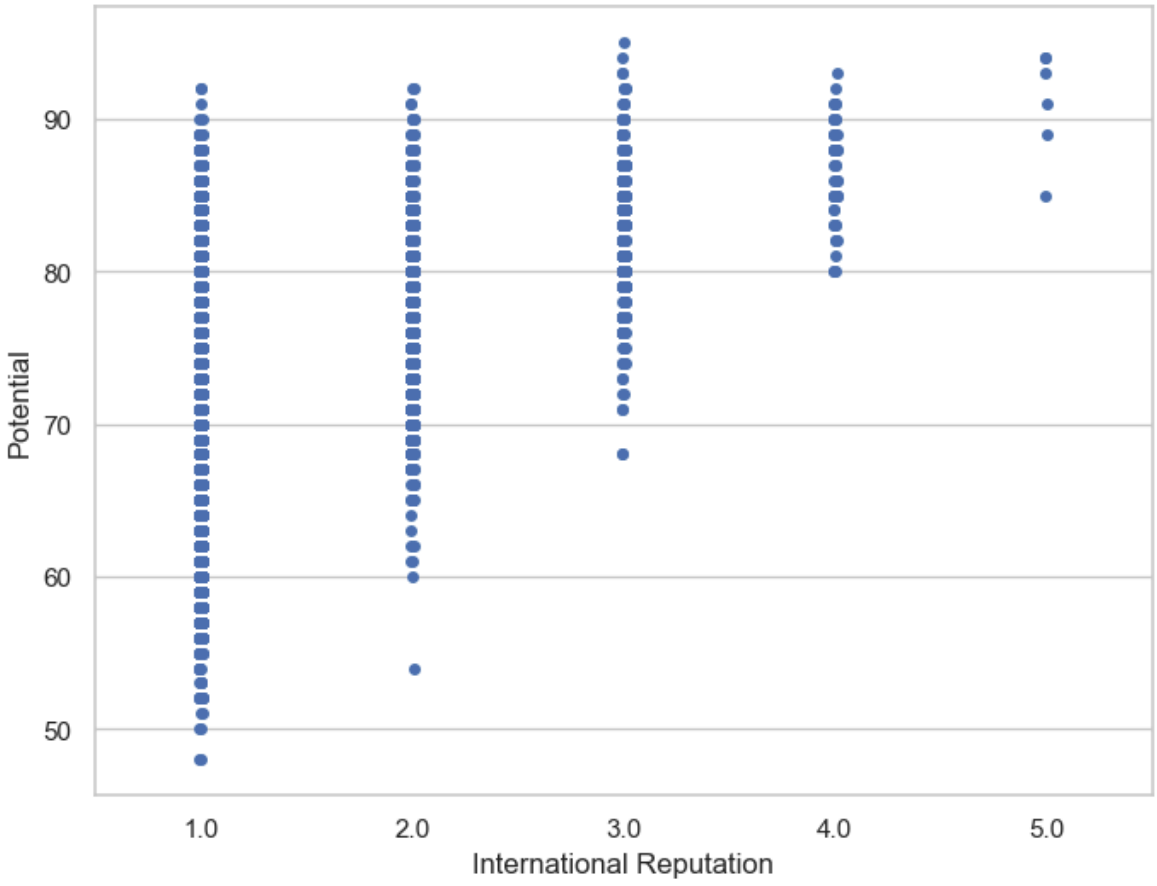
In [61]:
```
#Distribution of values in International Reputation variable

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

Out[61]:
```
International Reputation
1.0    16532
2.0     1261
3.0      309
4.0       51
5.0        6
Name: count, dtype: int64
```

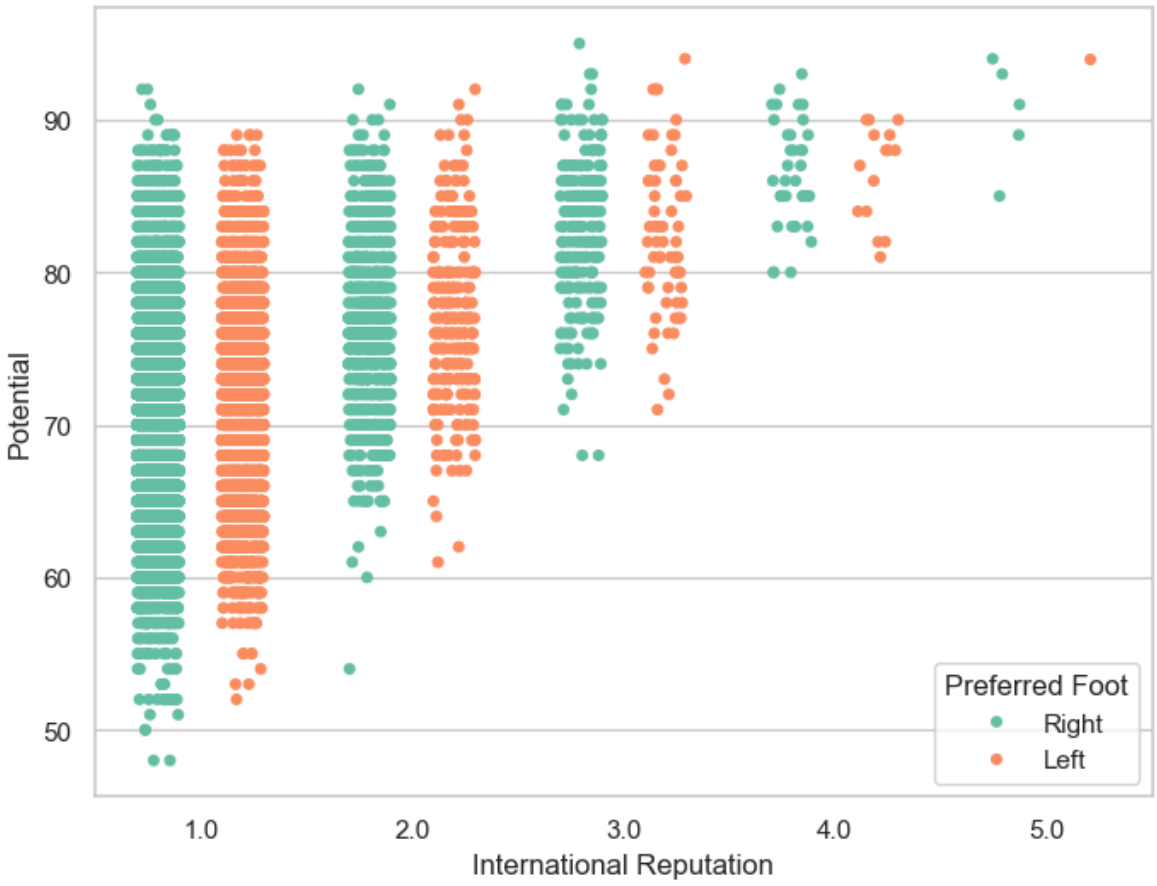
In [63]:
```
# seaborn stripplot() Function--Categorical variable and Potential

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



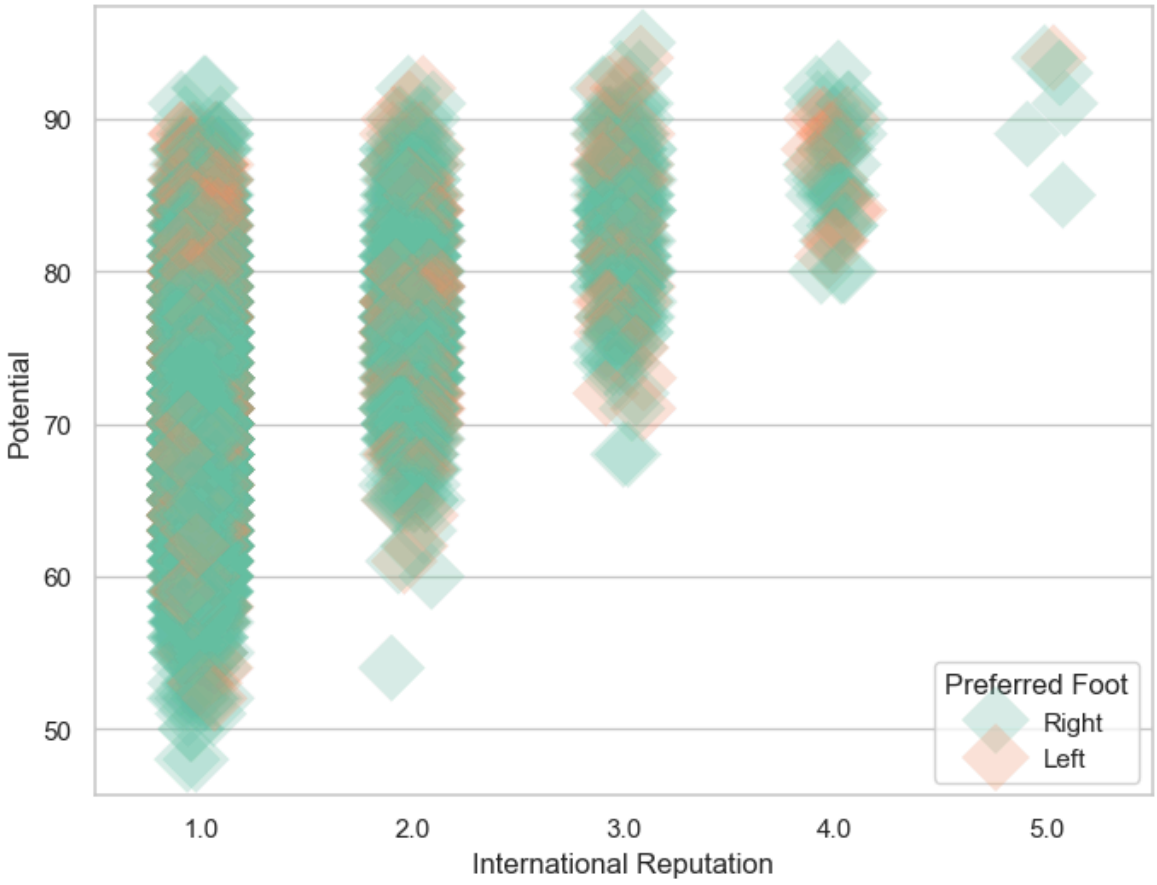
In [65]:
```
#jitter to bring out the distribution of values

f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", data=fifa, jitter=0.0
plt.show()
```



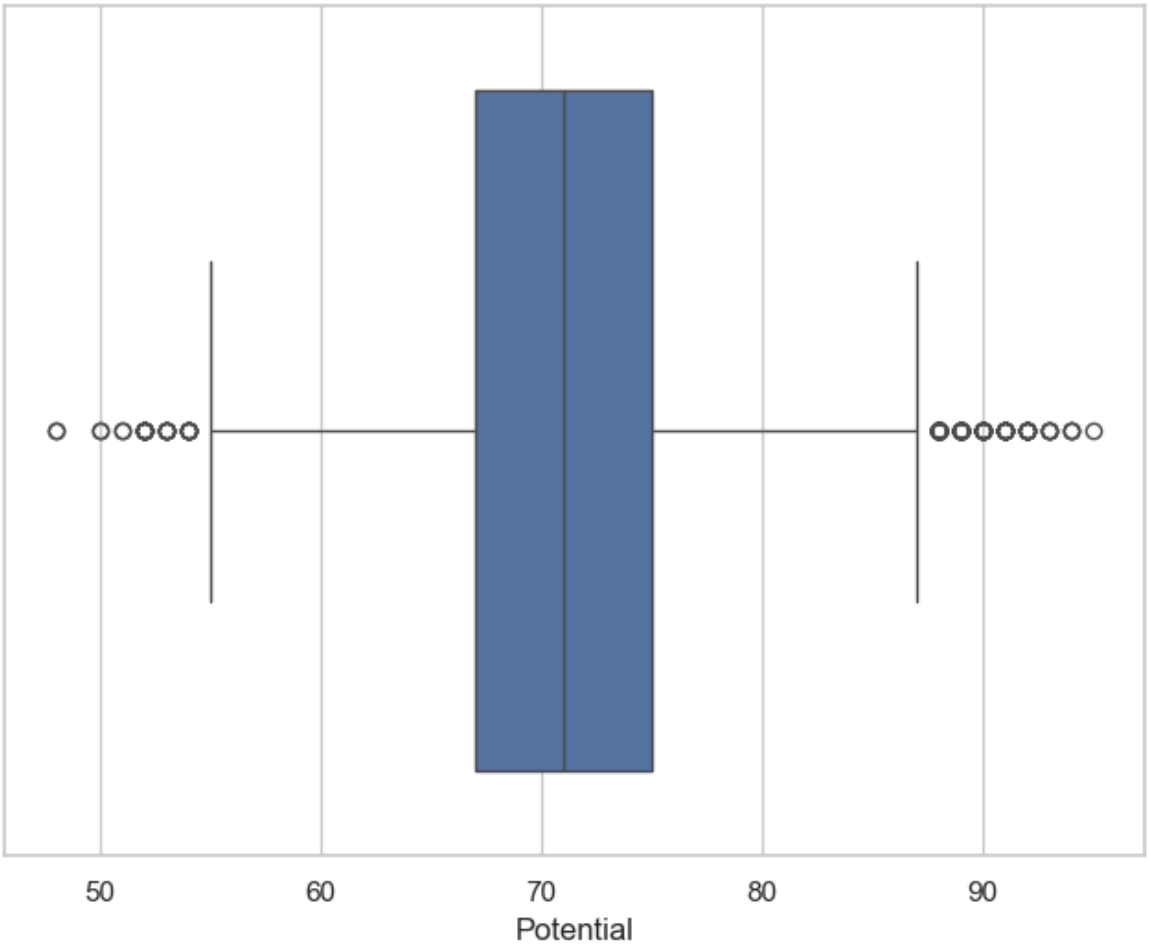
In [67]:
```
# Second Categorical Variable

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



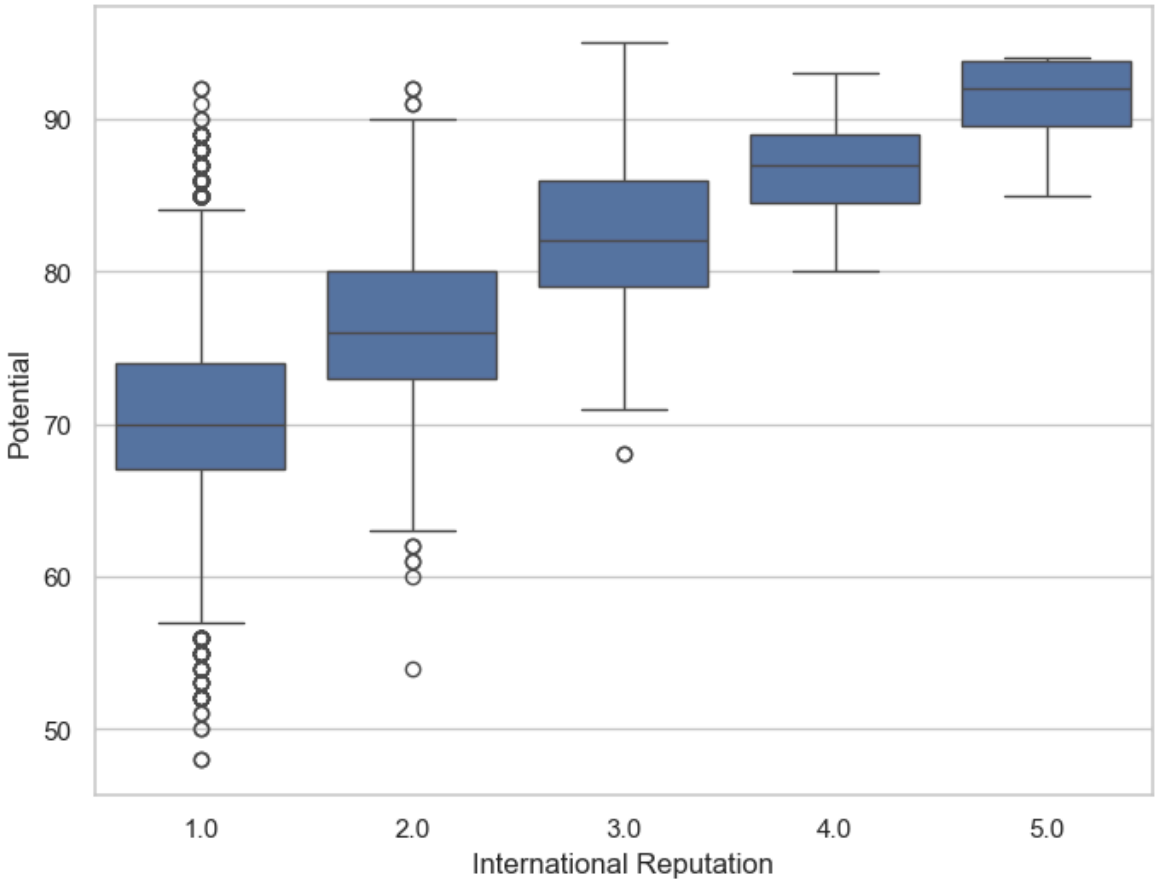
In [69]:
```
# Draw strips with large points and different aesthetics

f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", hue="Preferred Foot",
                    data=fifa, palette="Set2", size=20, marker="D",
                    edgecolor="gray", alpha=.25)
plt.show()
```



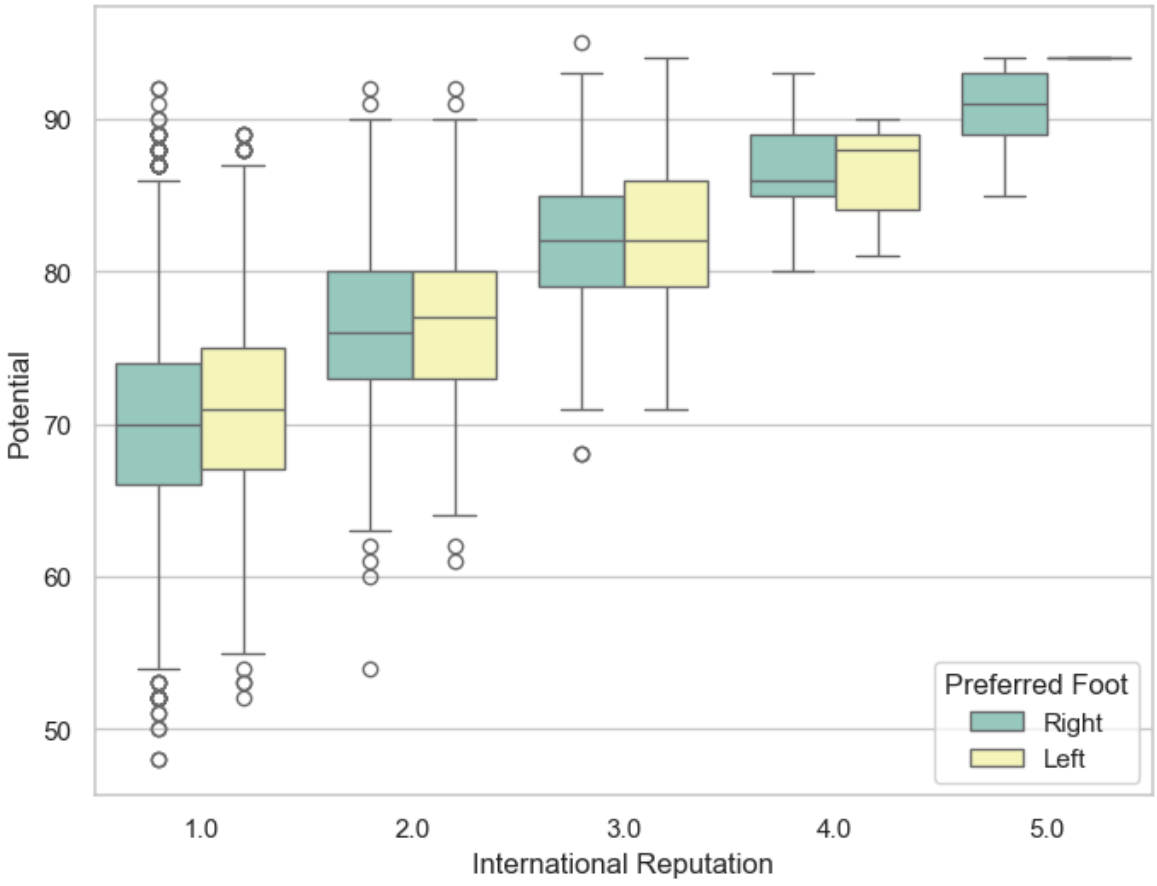
In [71]:
```
# boxplot()

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



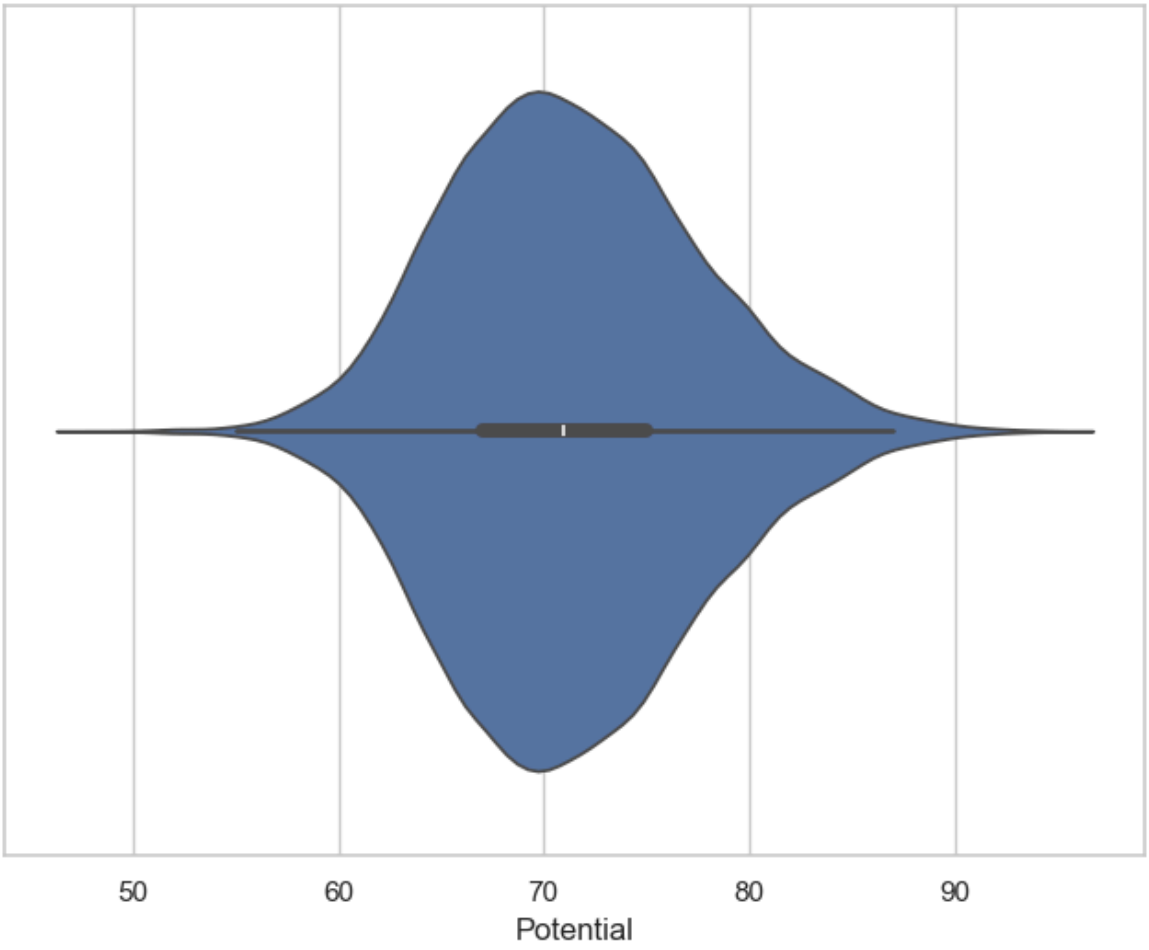
In [73]:
```
# Vertical boxplot()

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



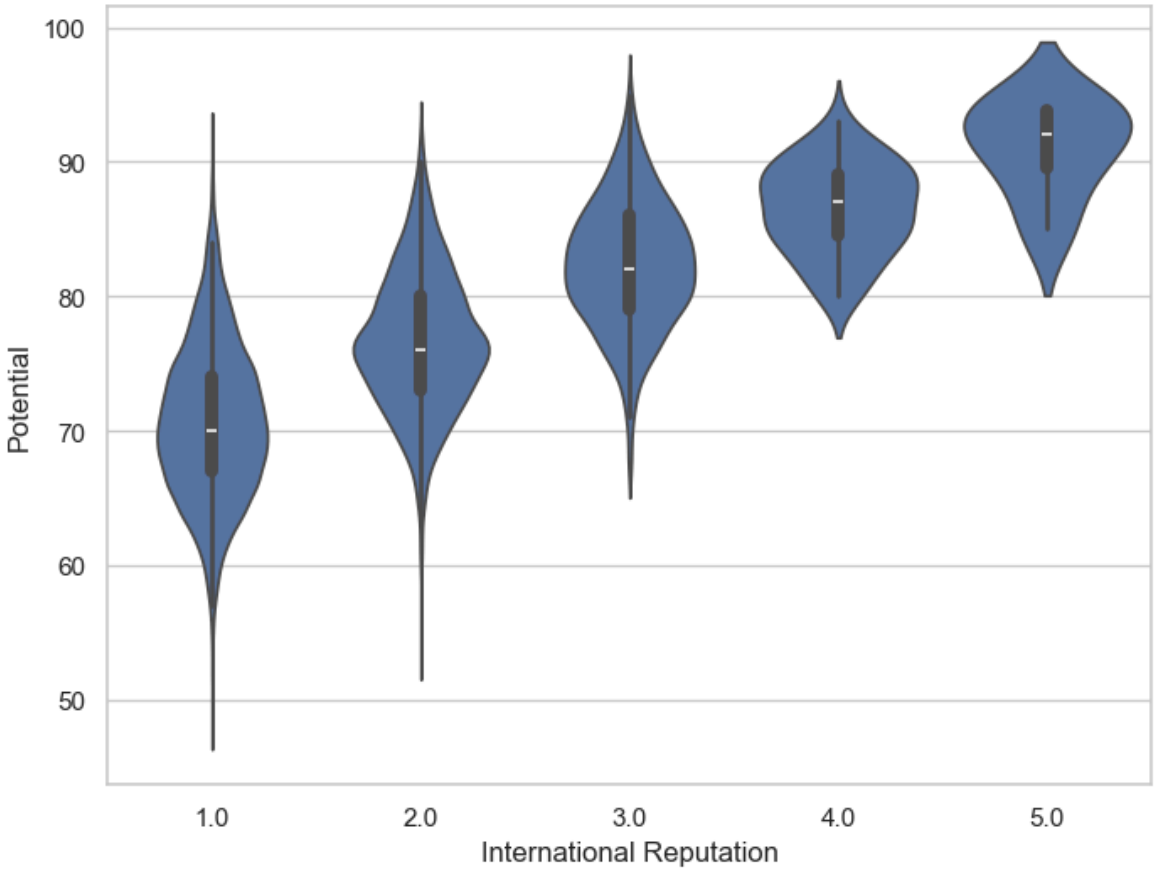
In [75]:
```
# two categorical Variable

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



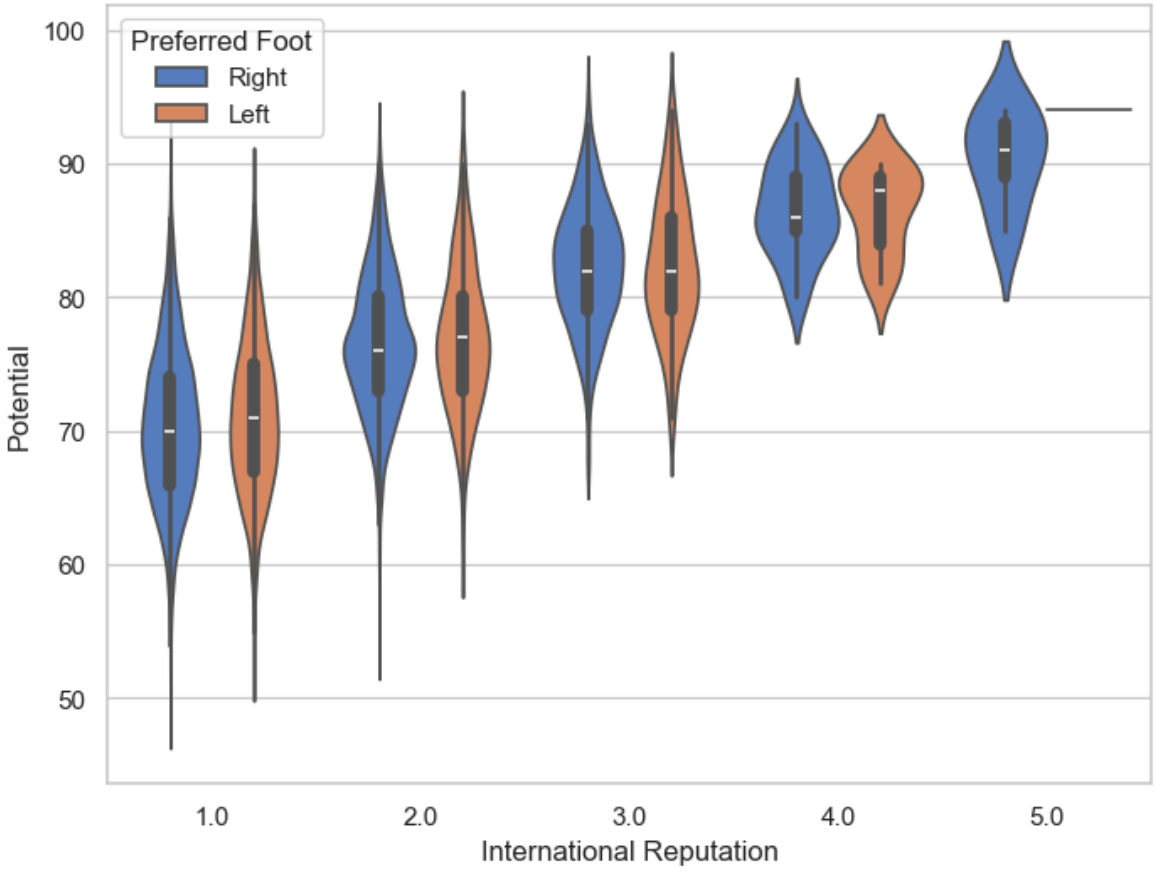
In [77]:
```
# Violinplot() Function

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



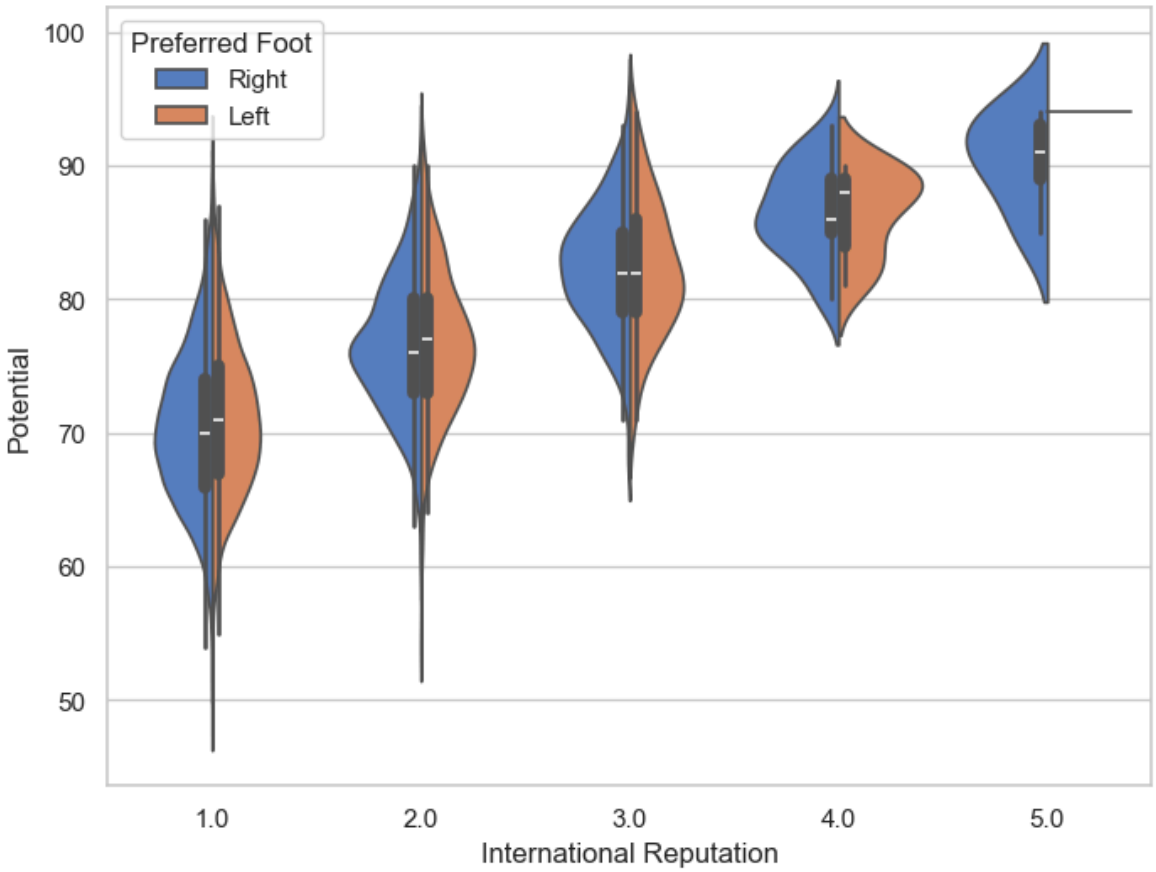
In [79]:
```
# Vertical violinplot by Categorical variable

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



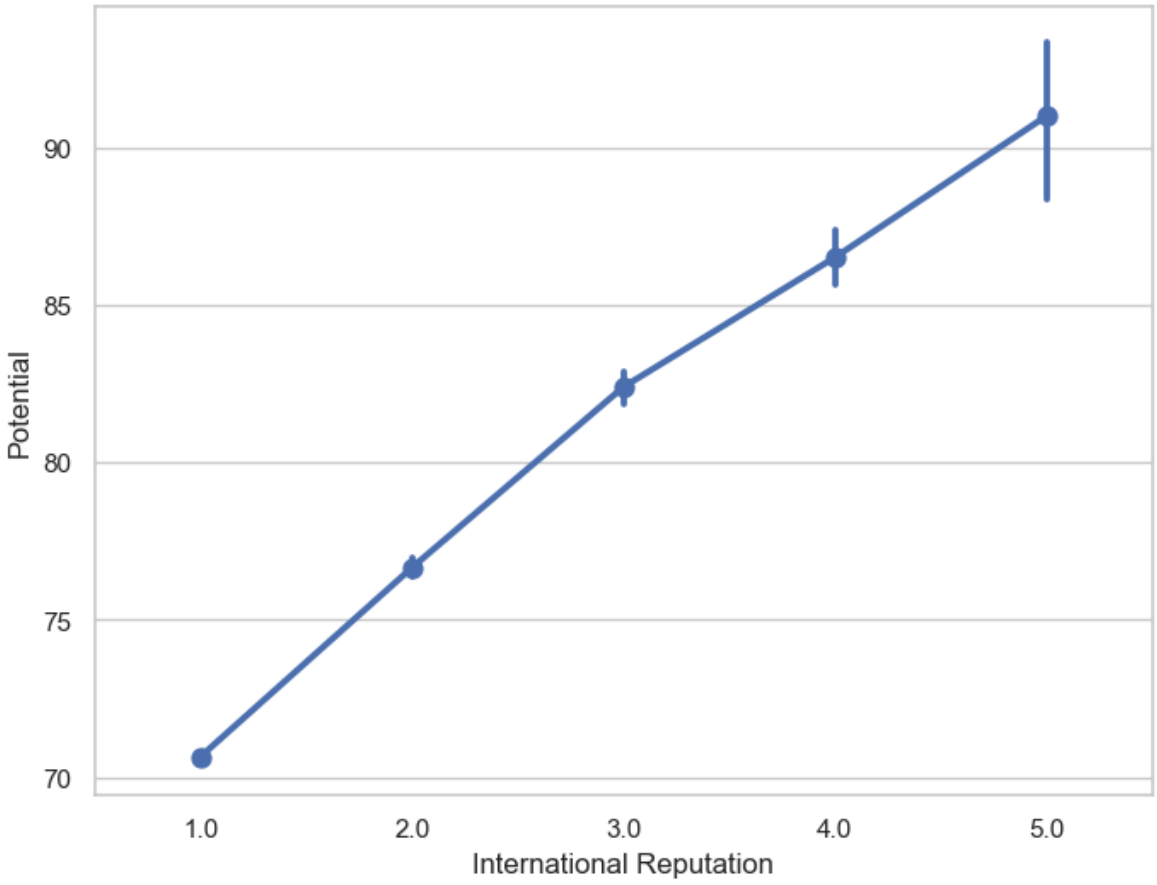
In [81]:
```
# Violinplot by two Categorical variable

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



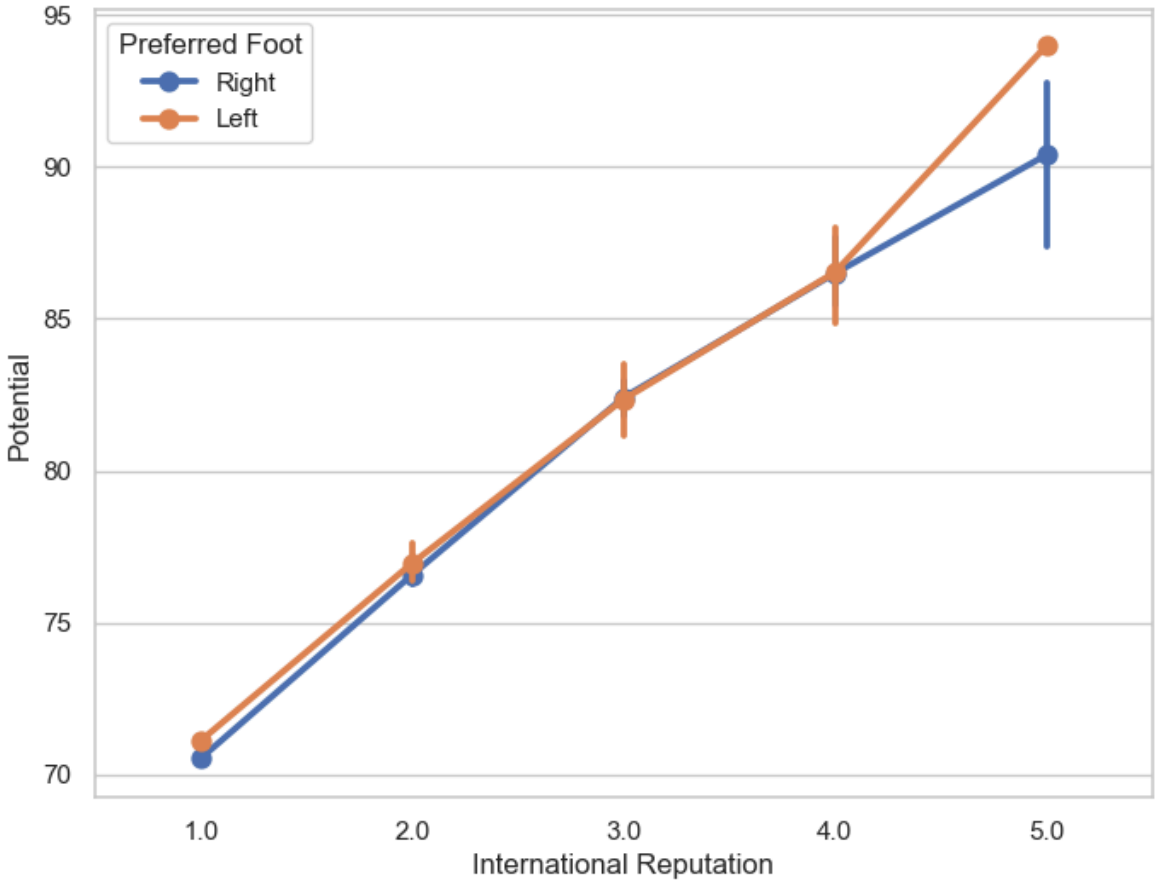
In [83]:
```
# split violins to compare the across the hue variable

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



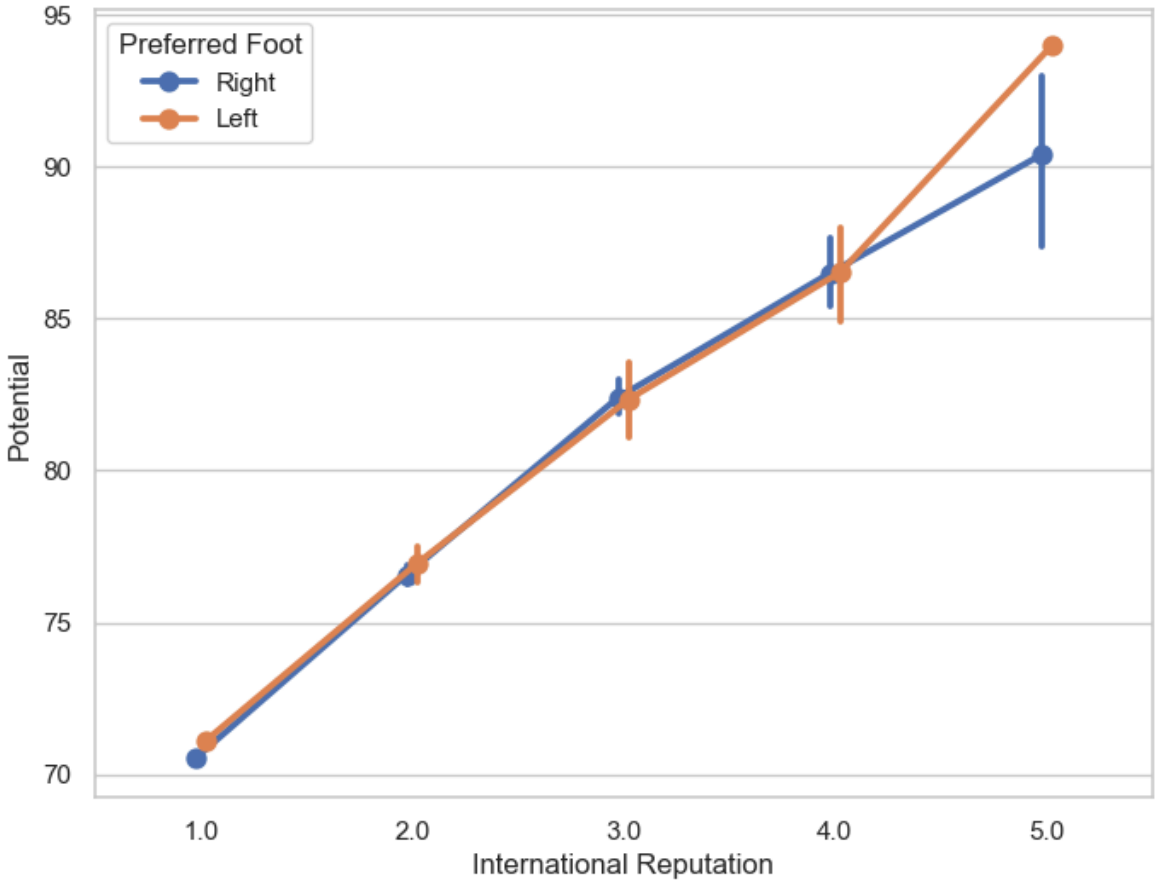
In [85]:
```
# pointplot() function

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



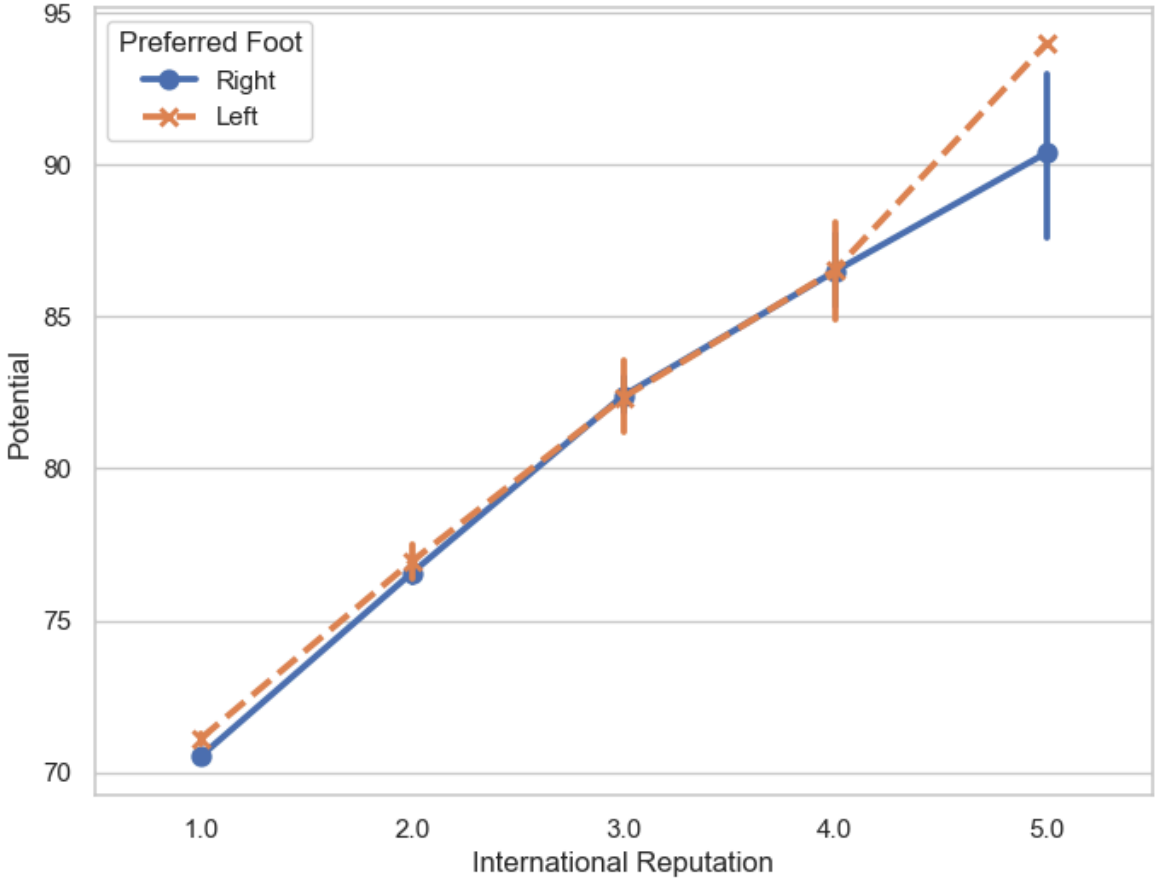
In [87]:
```
# set of vertical points with nested grouping by a two variables

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



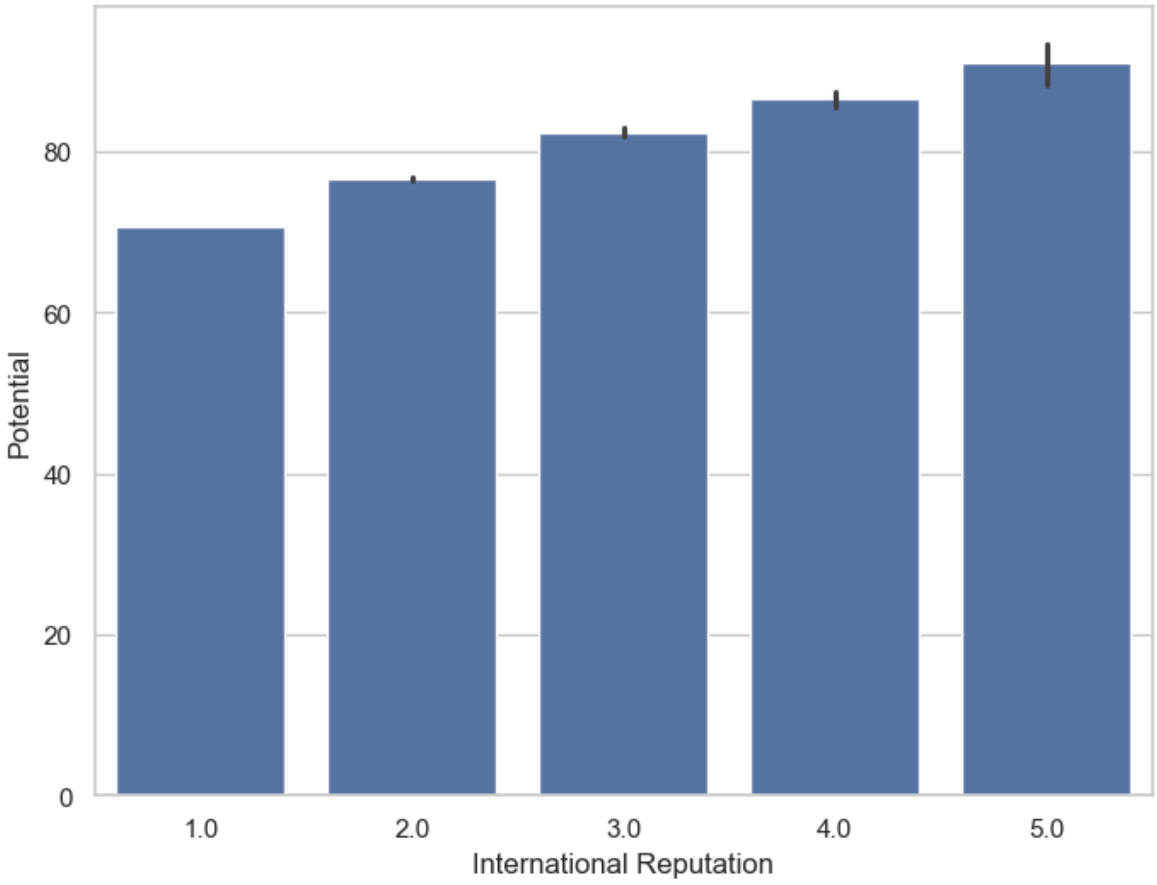
In [89]:
```
# the points for different hue levels along the categorical axis

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



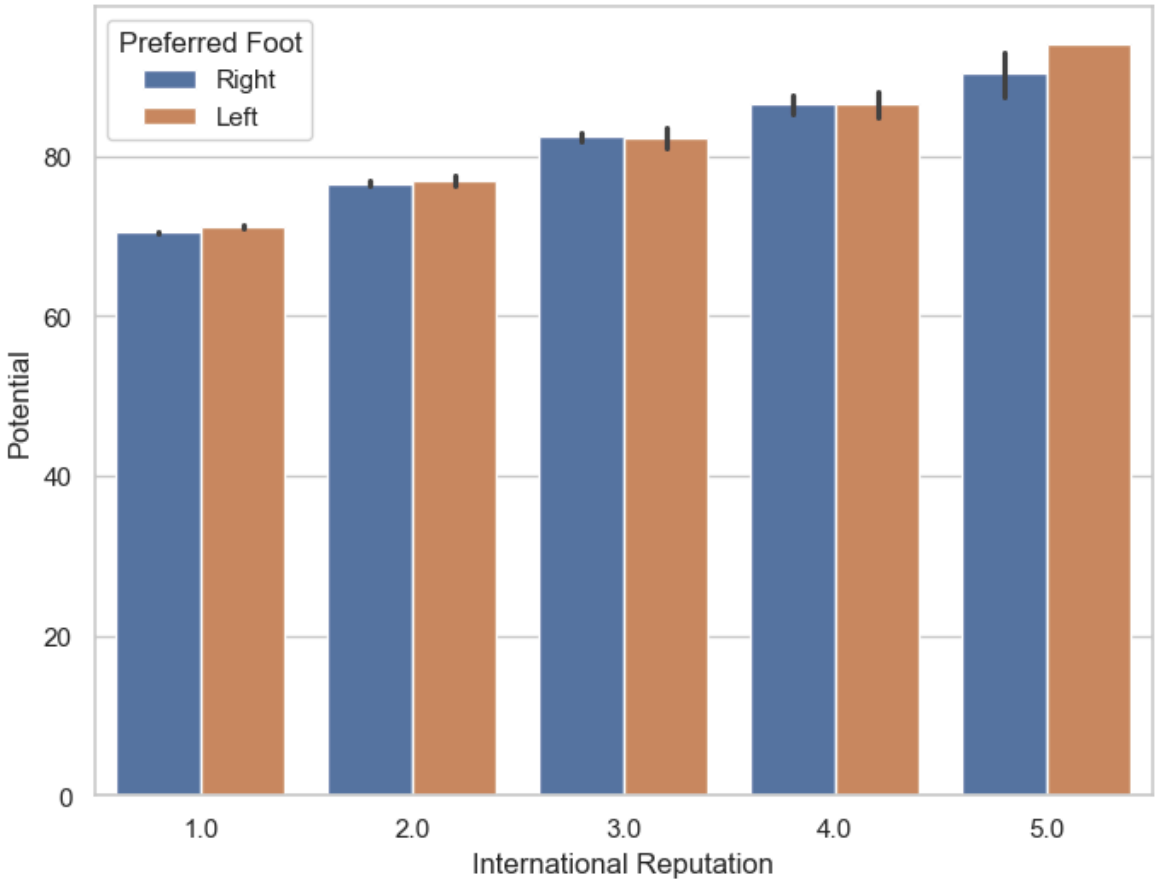
In [91]:
```
# different marker and line style for the hue levels

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



In [93]:
```
# barplot()

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

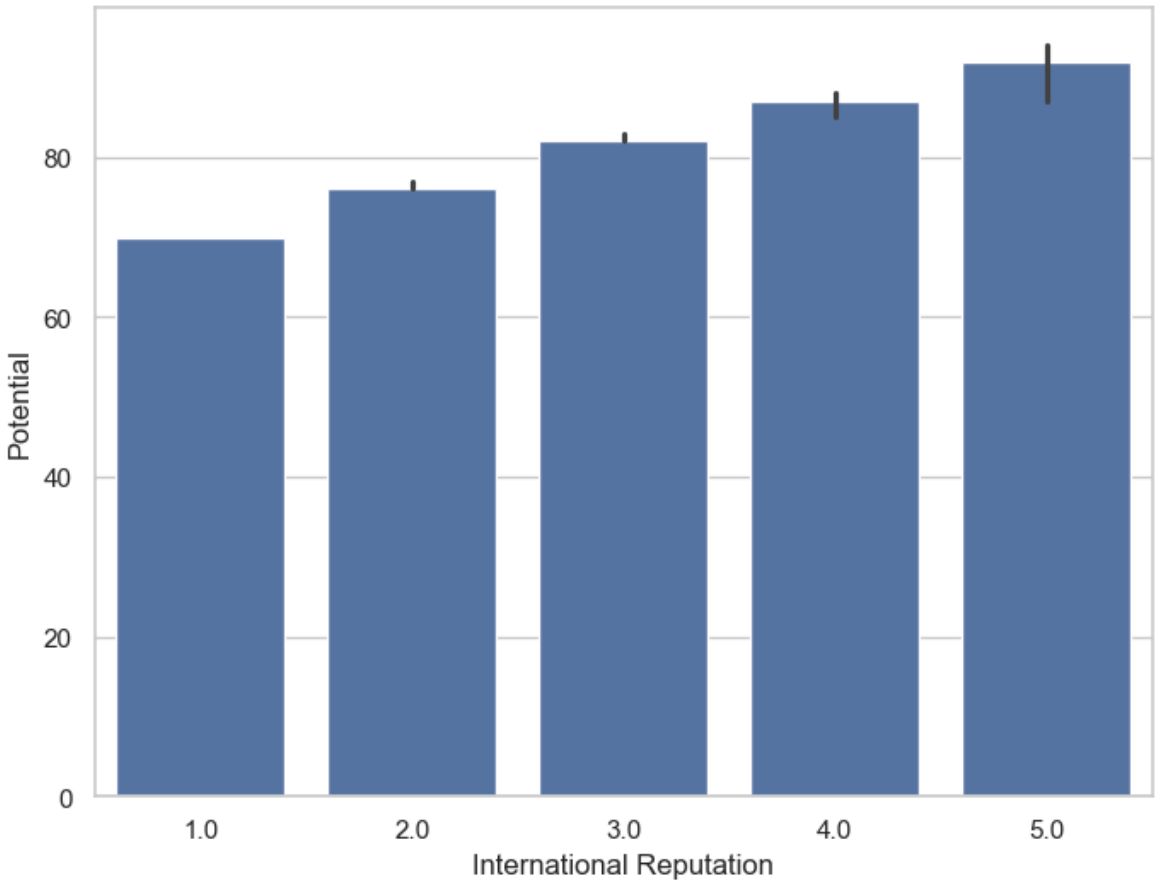


In [99]:
```
# set of vertical bars with nested grouping by a two variables

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



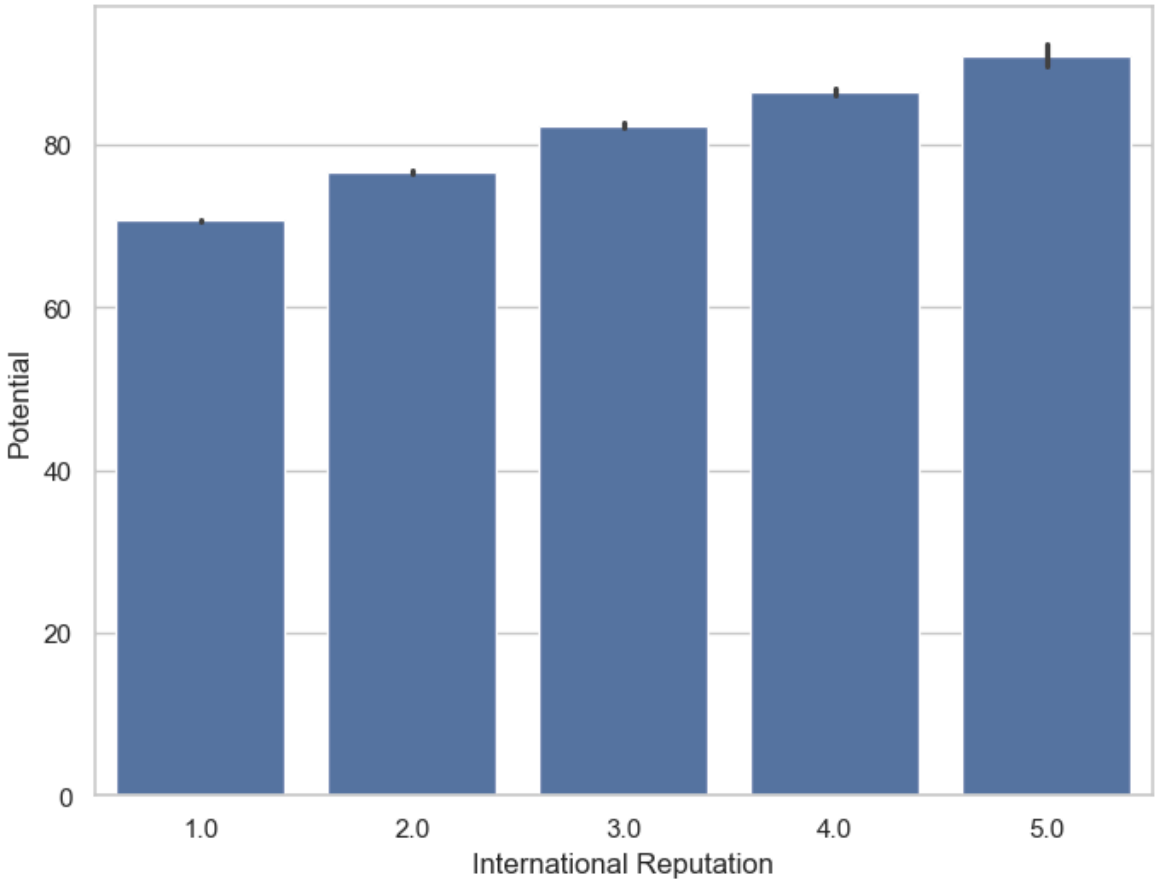
In [101…

```
# Median as the estimate of central tendency

from numpy import median
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa, estimator=me
plt.show()
```



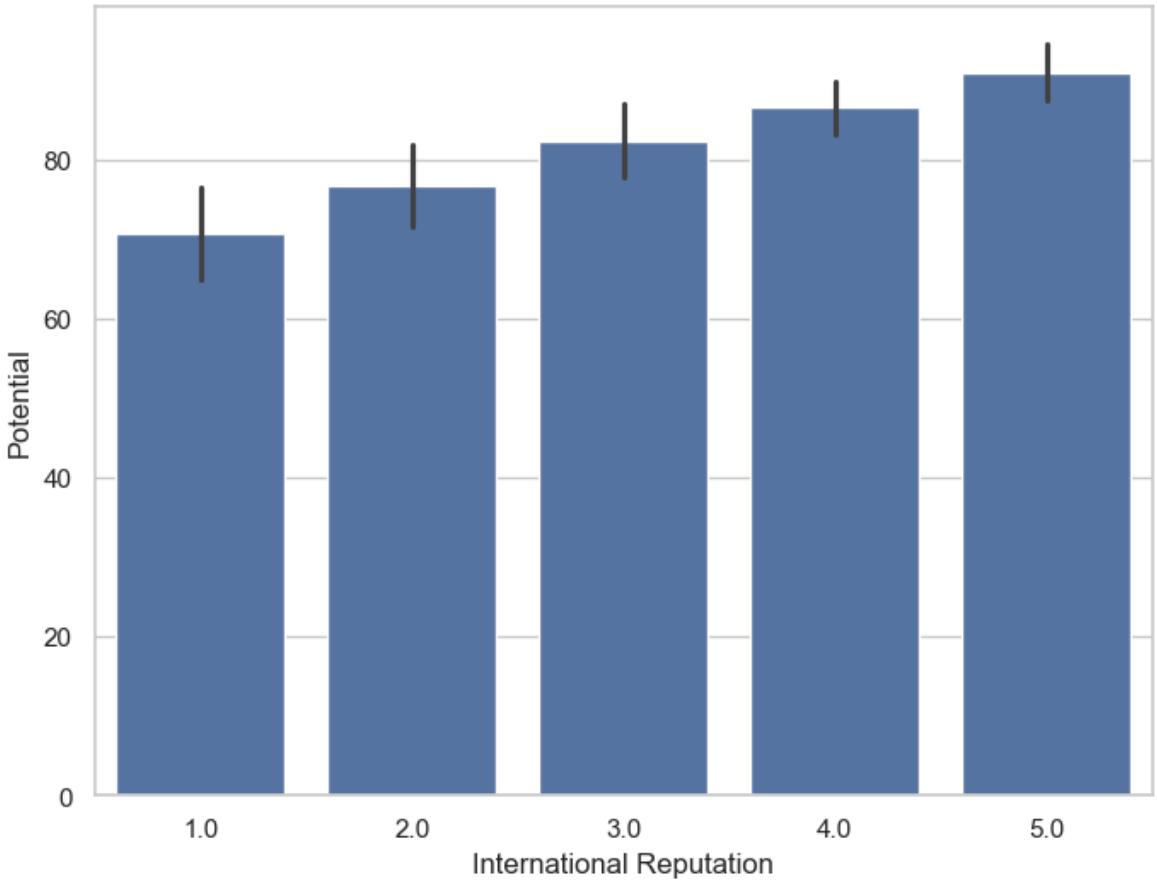
In [103…

```
#  standard error of the mean with the error bars

f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa, ci=68)
plt.show()
```



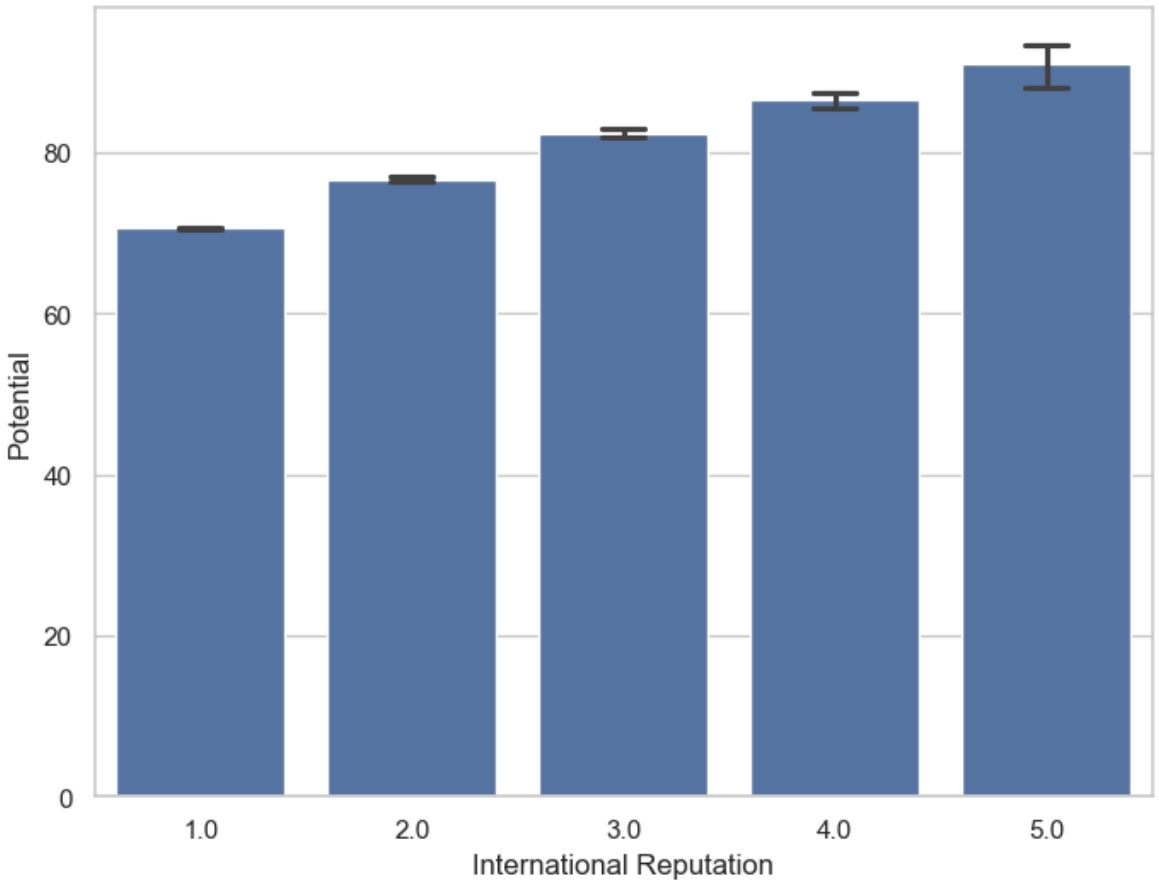
In [105…

```
# standard deviation of observations instead of a confidence interval

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



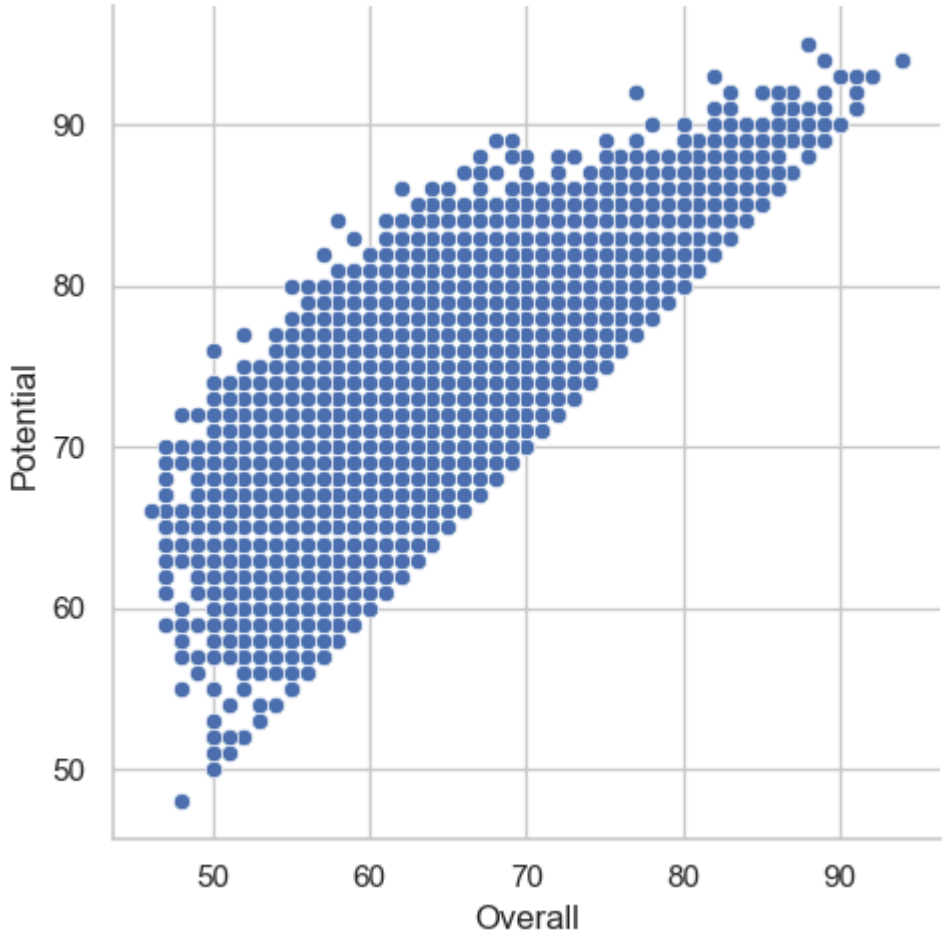
In [107…

```
#  add “caps” to the error bars

f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa, capsize=0.2)
plt.show()
```



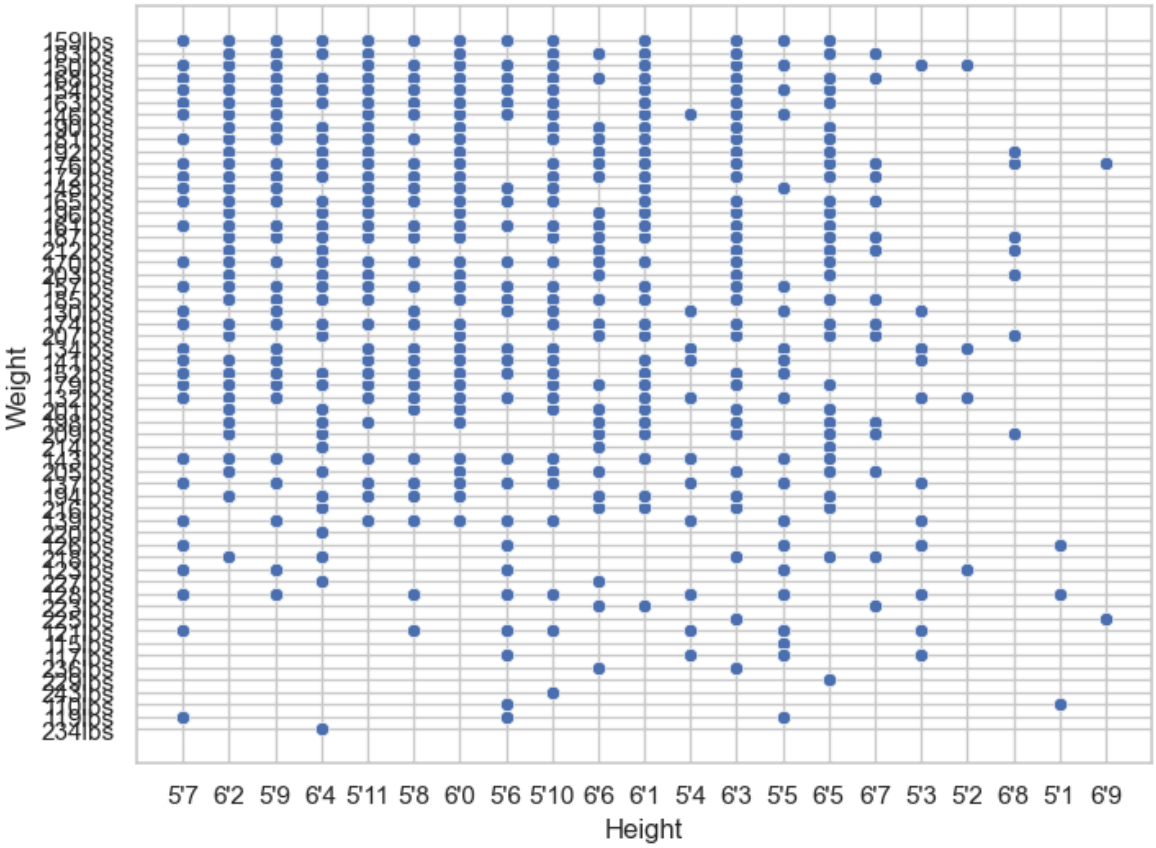
In [109…

```
# relplot()---Height and Weight

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



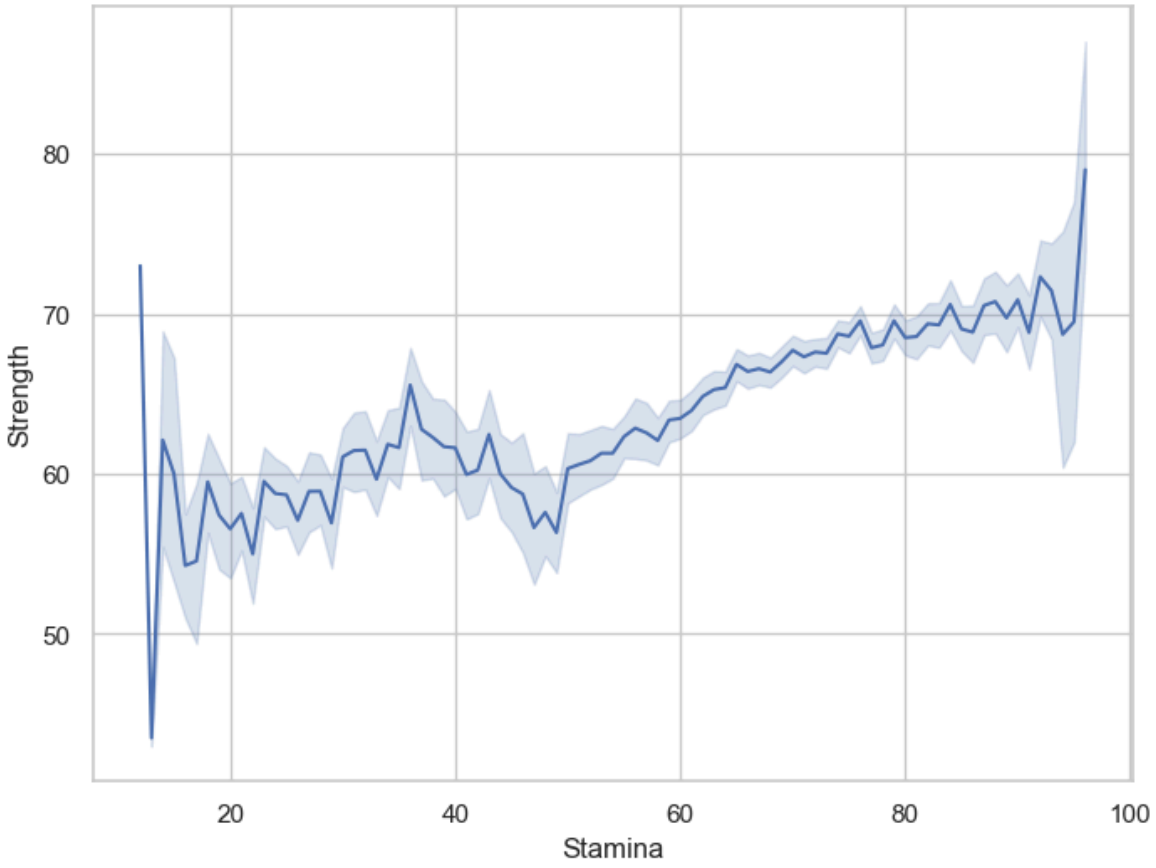
In [111…

```
# scatterplot() Function

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



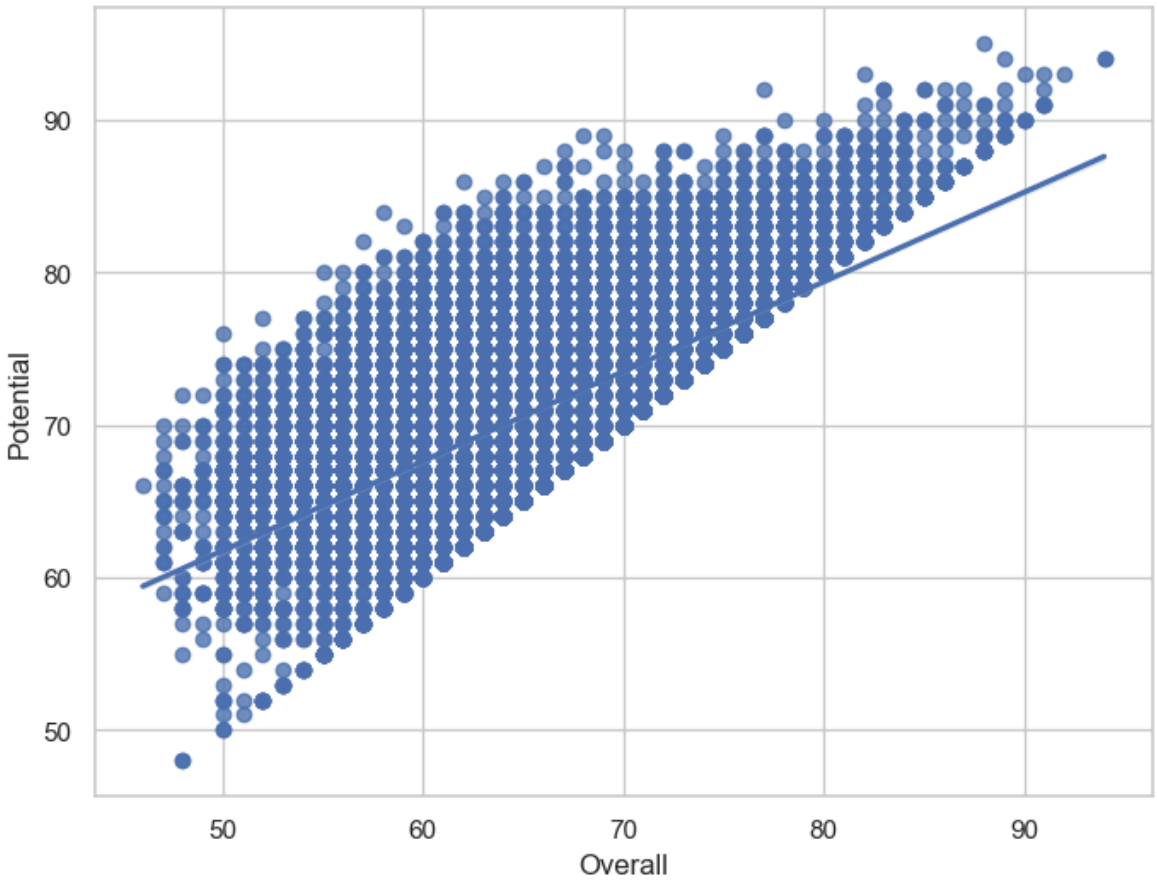
In [116…

```
#  lineplot() function

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



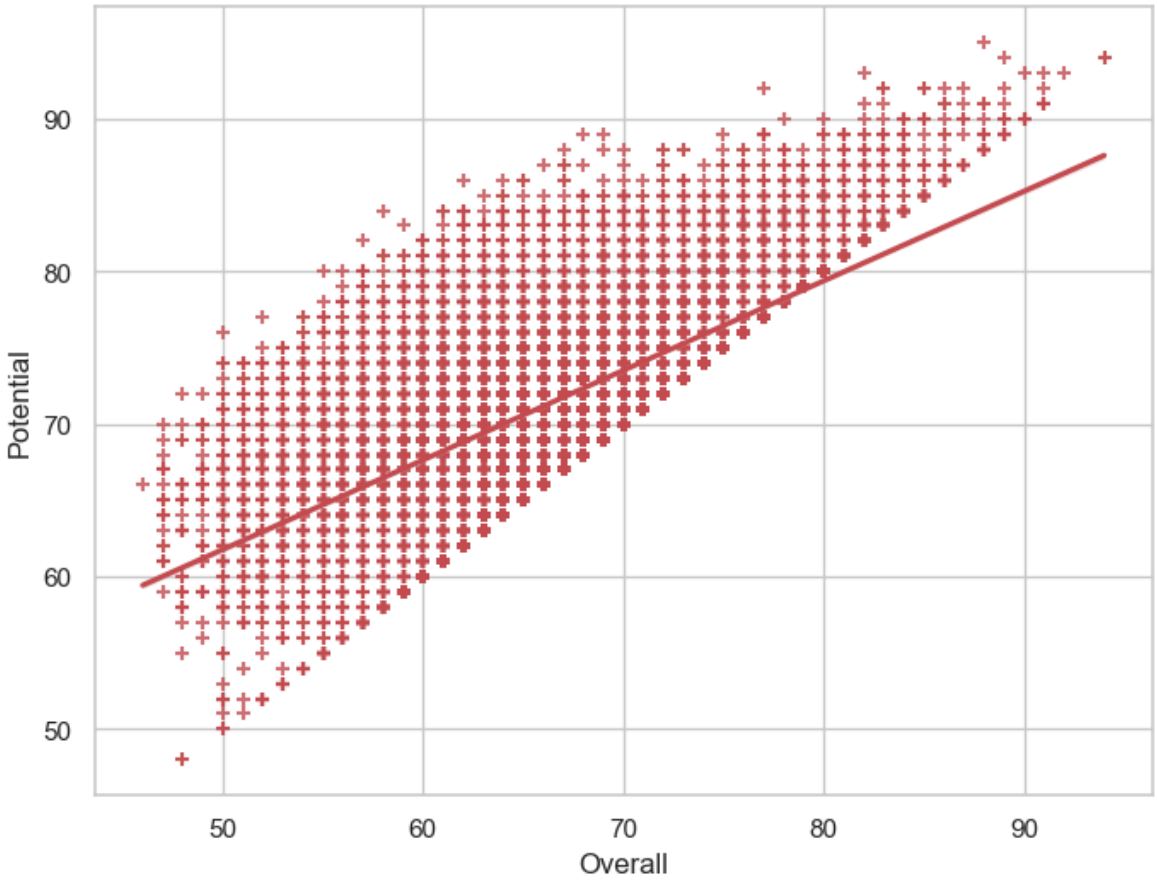
In [118…

```
#  regplot() function

f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa)
plt.show()
```



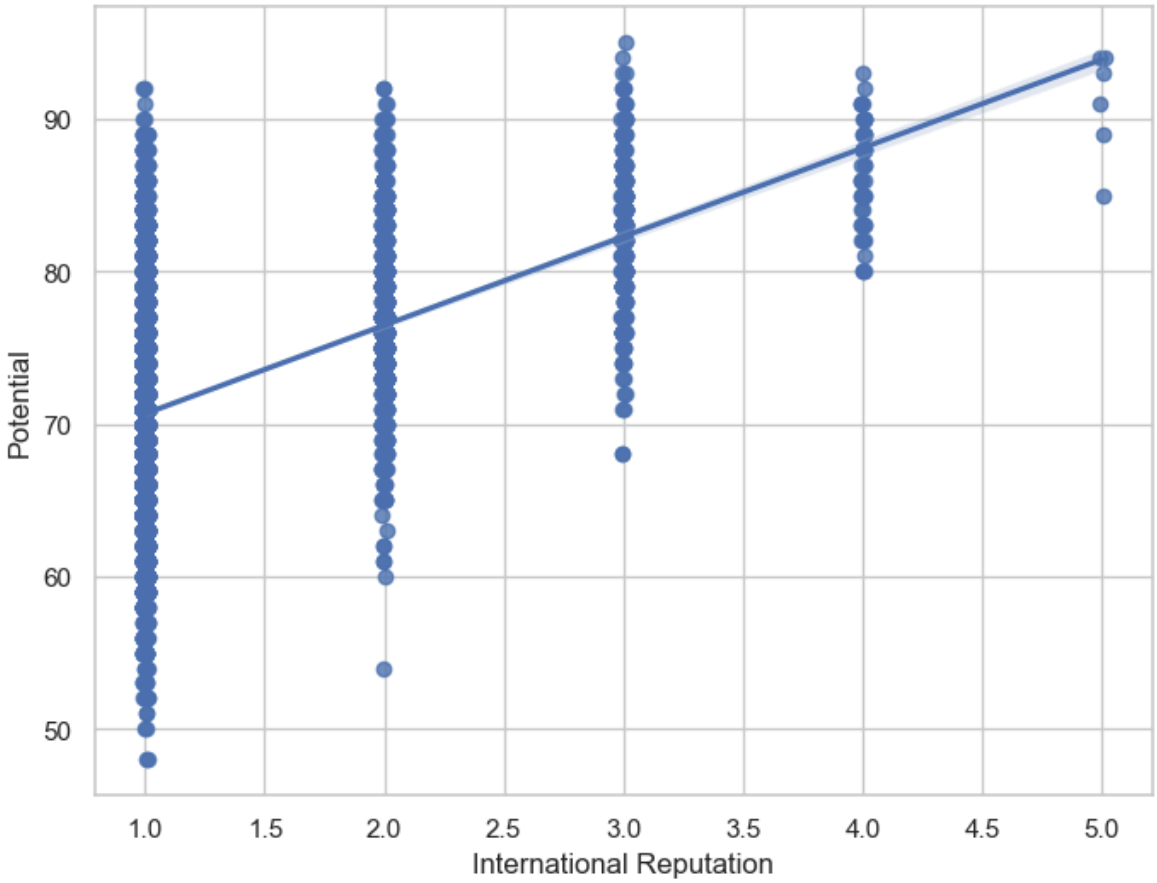
In [120…

```
#  different color and marker

f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa, color= "r", marker="+")
plt.show()
```



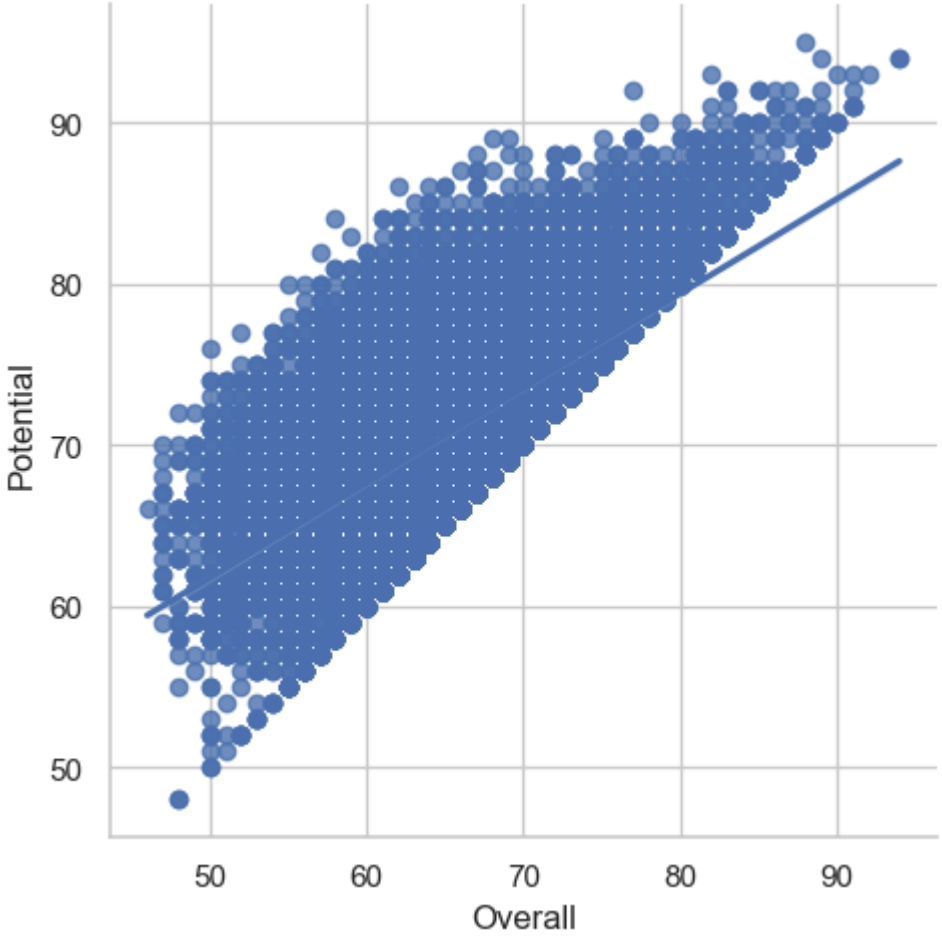
In [122…

```
#  discrete variable and add some jitter

f, ax = plt.subplots(figsize=(8, 6))
sns.regplot(x="International Reputation", y="Potential", data=fifa, x_jitter=.01
plt.show()
```



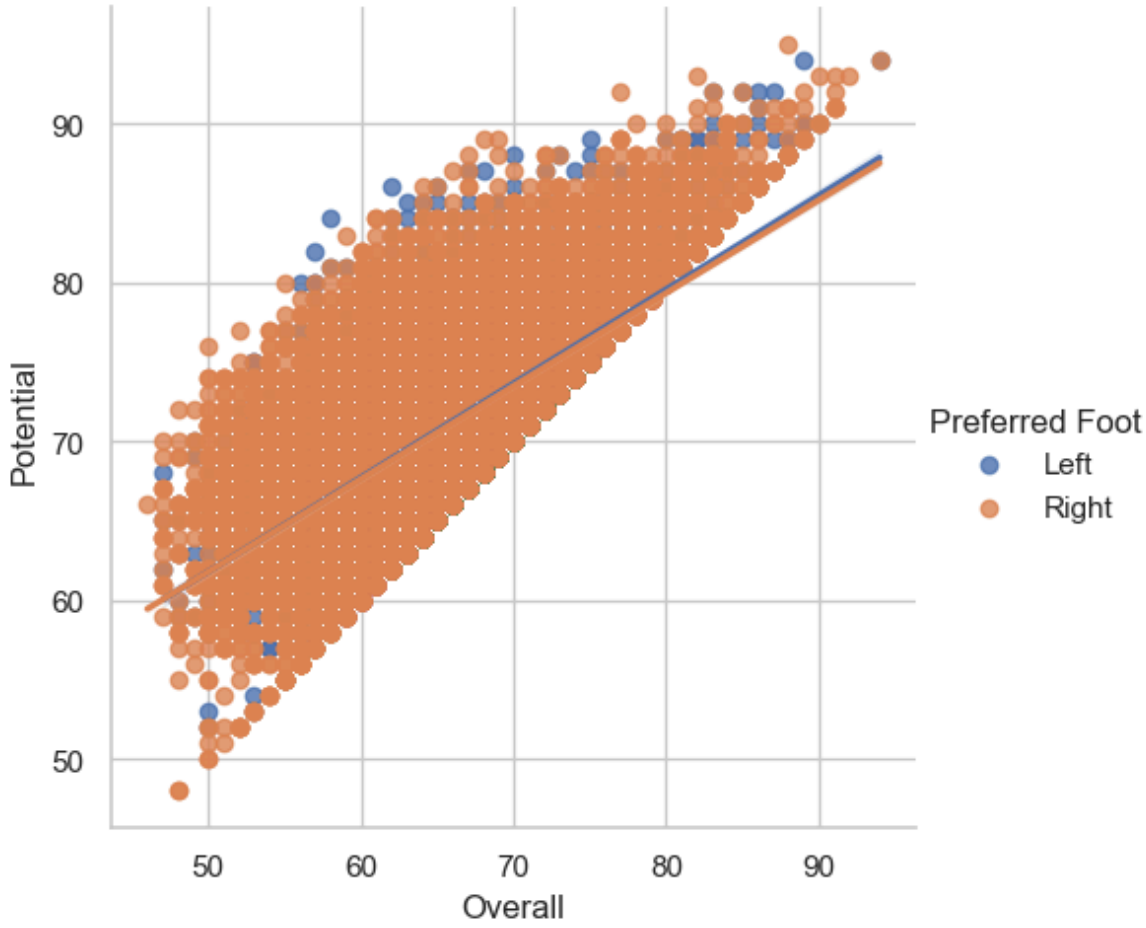
In [124…

```
# lmplot() function

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



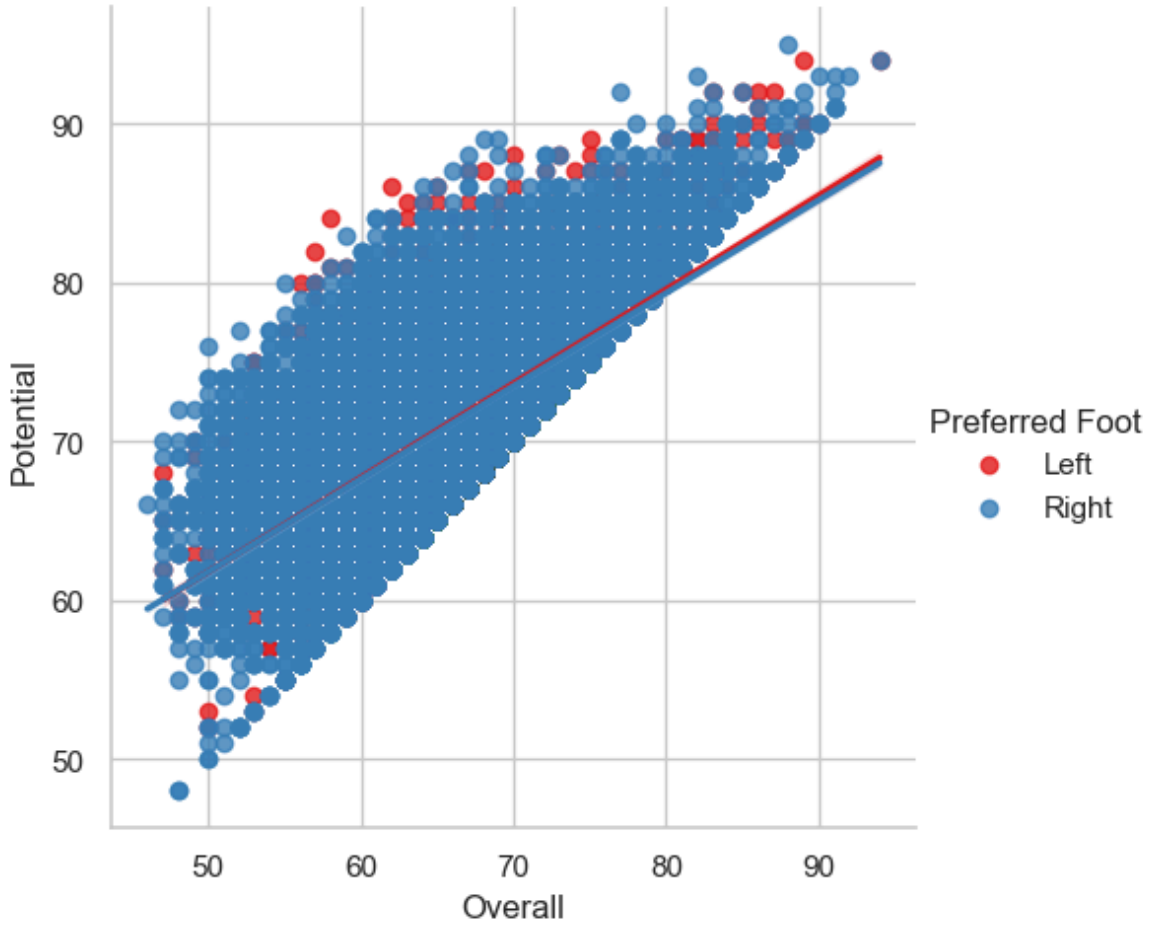
In [126…

```
#  a third variable and plot the levels in different colors

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



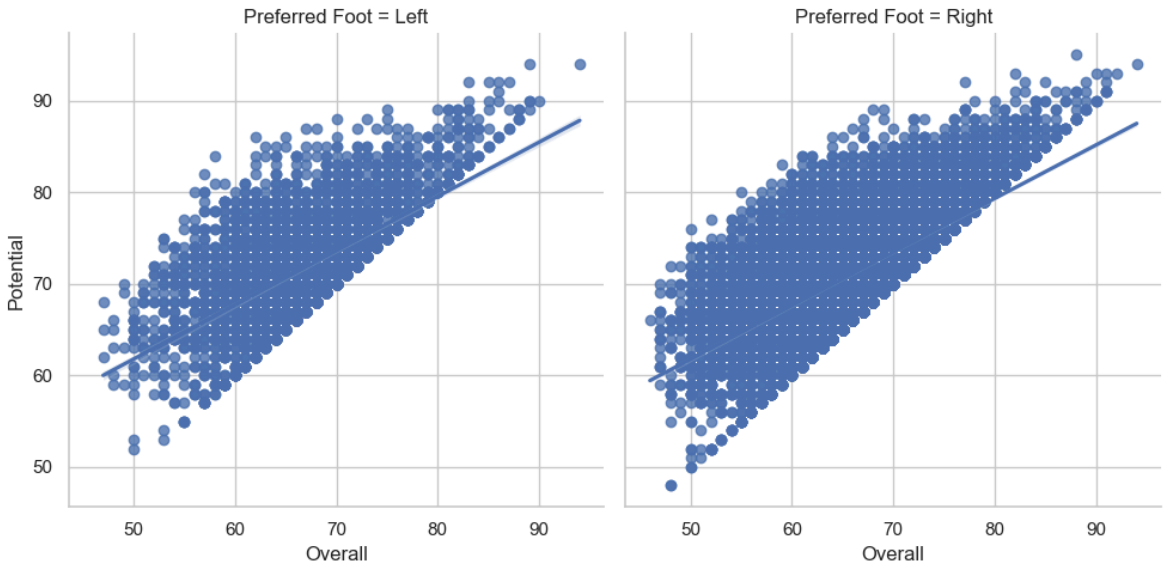
In [128…

```
#  different color palette

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



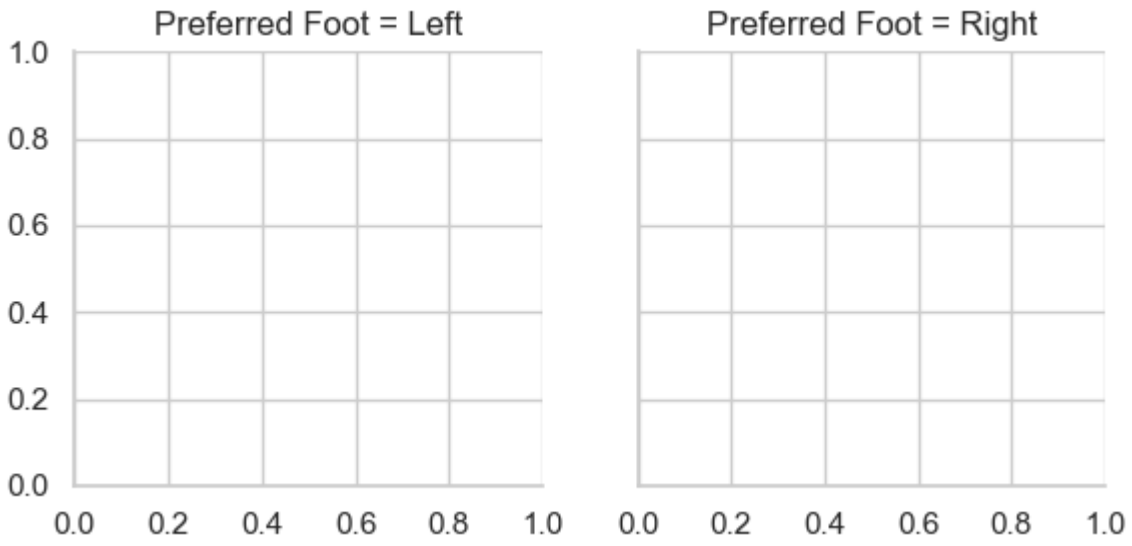
In [130…

```
#  evels of the third variable across different columns

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



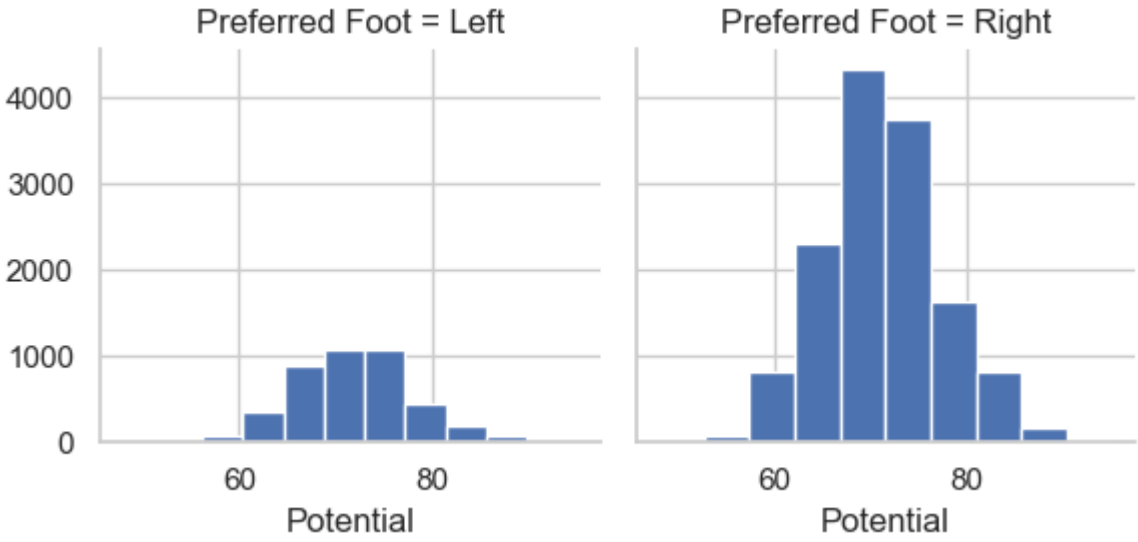
In [132…

```
# Multi-plot grids-- facetgrid()

g = sns.FacetGrid(fifa, col="Preferred Foot")
```



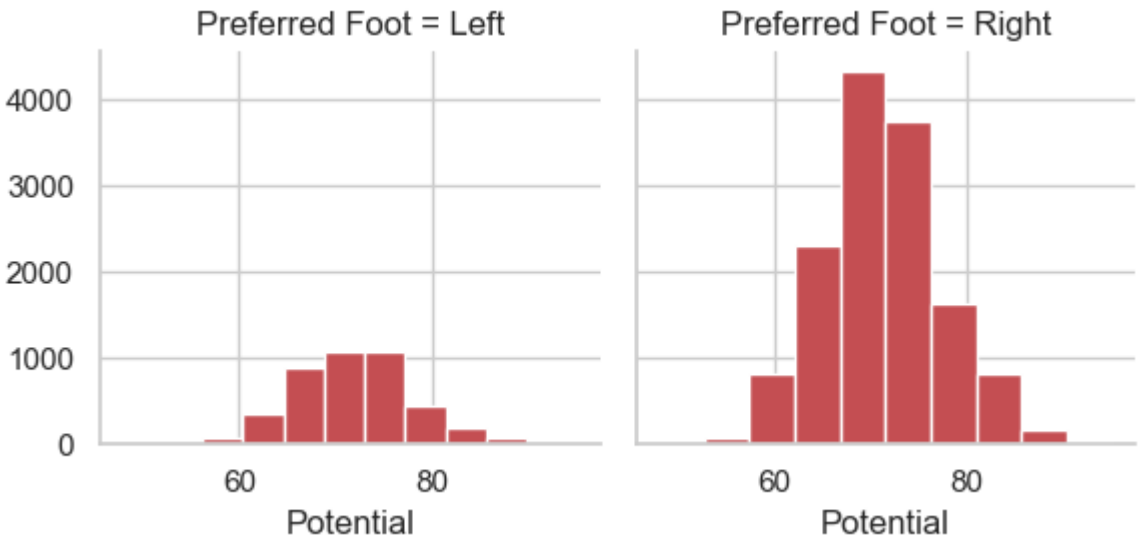
In [134…

```
#  univariate plot of `Potential` variable on each facet

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



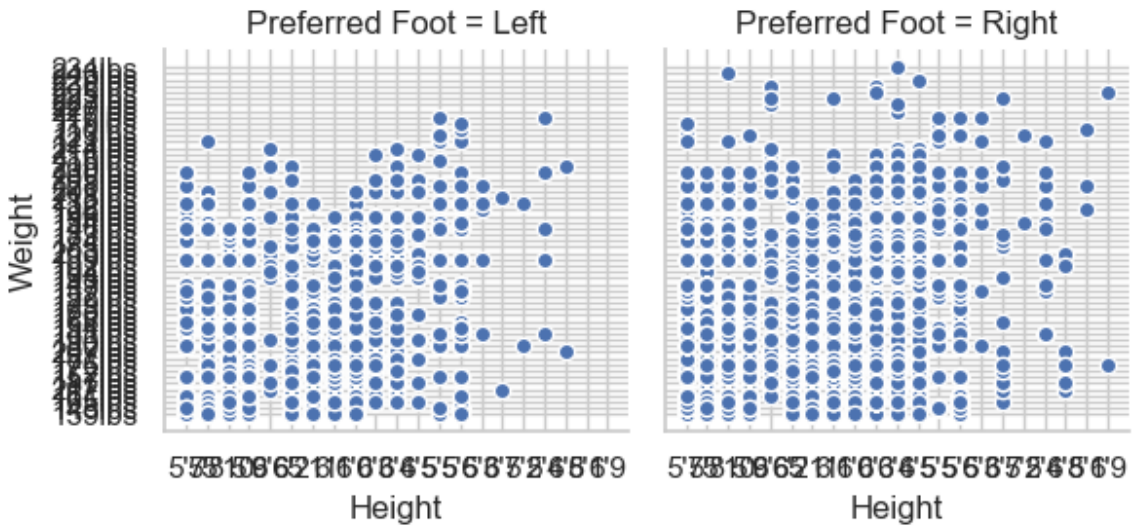
In [136…

```
g = sns.FacetGrid(fifa, col="Preferred Foot")
g = g.map(plt.hist, "Potential", bins=10, color="r")
```



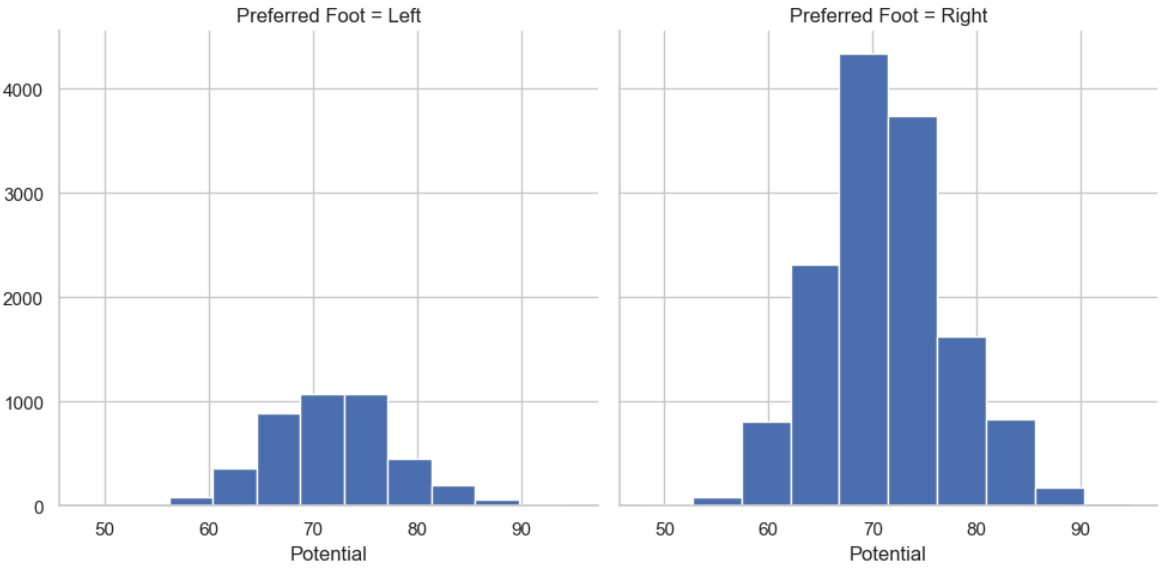
In [138…

```
#  plot a bivariate function on each facet

g = sns.FacetGrid(fifa, col="Preferred Foot")
g = (g.map(plt.scatter, "Height", "Weight", edgecolor="w").add_legend())
```



In [140…

```
#  size of the figure is set by providing the height of each facet

g = sns.FacetGrid(fifa, col="Preferred Foot", height=5, aspect=1)
g = g.map(plt.hist, "Potential")
```

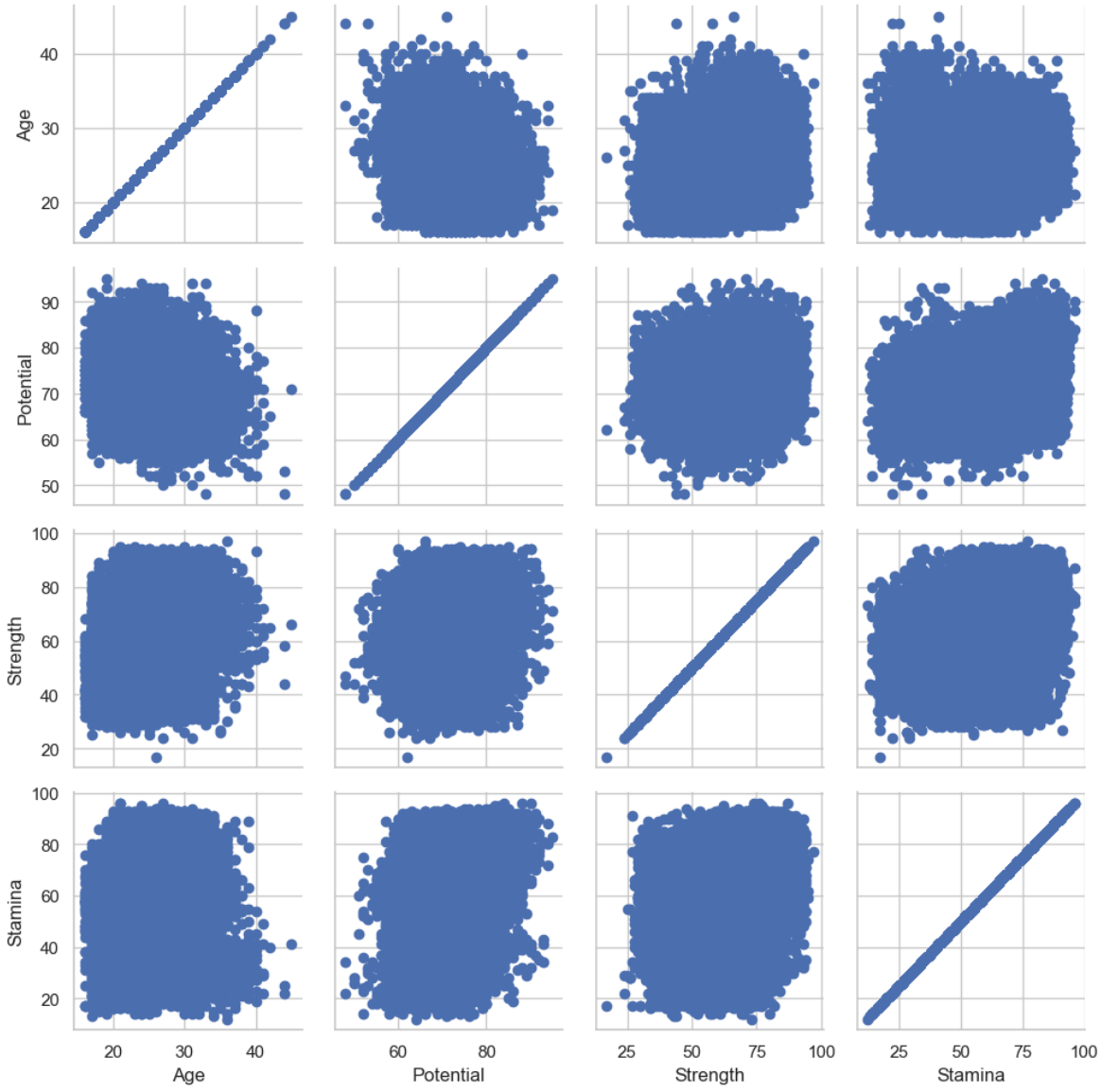


In [146…

```
#  pairgrid() function

fifa_new = fifa[['Age', 'Potential', 'Strength', 'Stamina', 'Preferred Foot']]
```

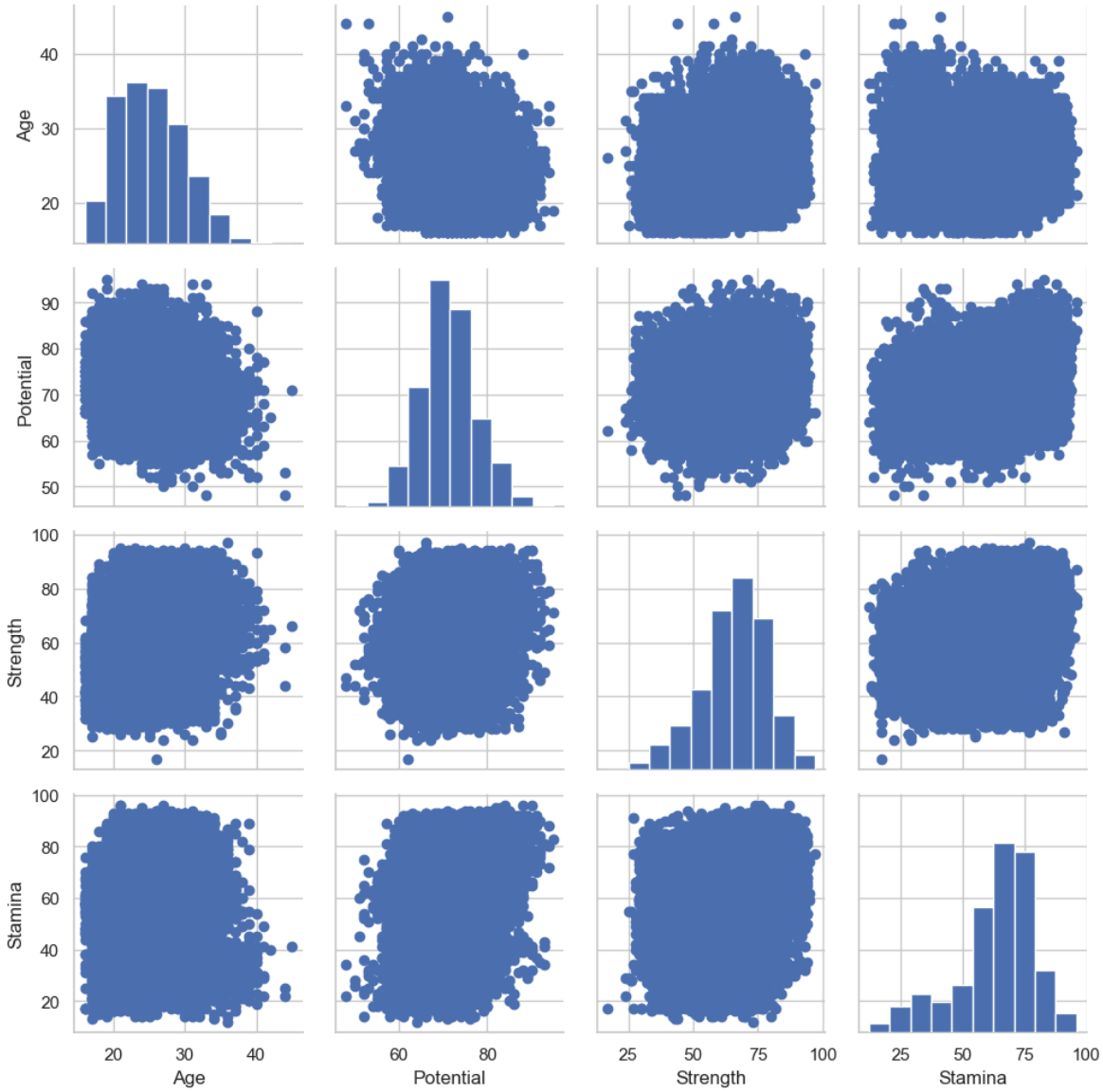
In [148…

```
g = sns.PairGrid(fifa_new)
g = g.map(plt.scatter)
```



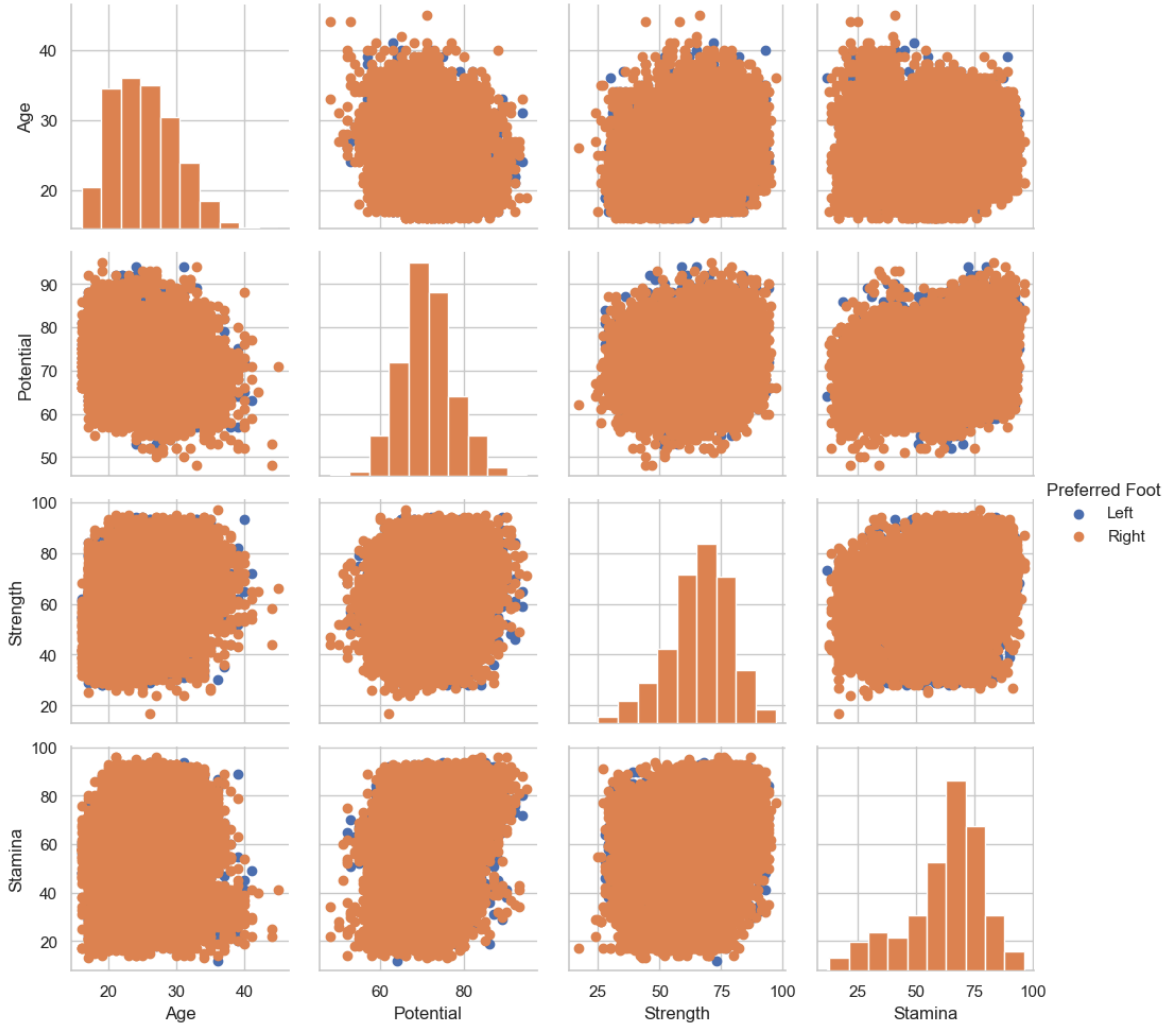
In [150…

```
#  univariate distribution on the diagonal

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



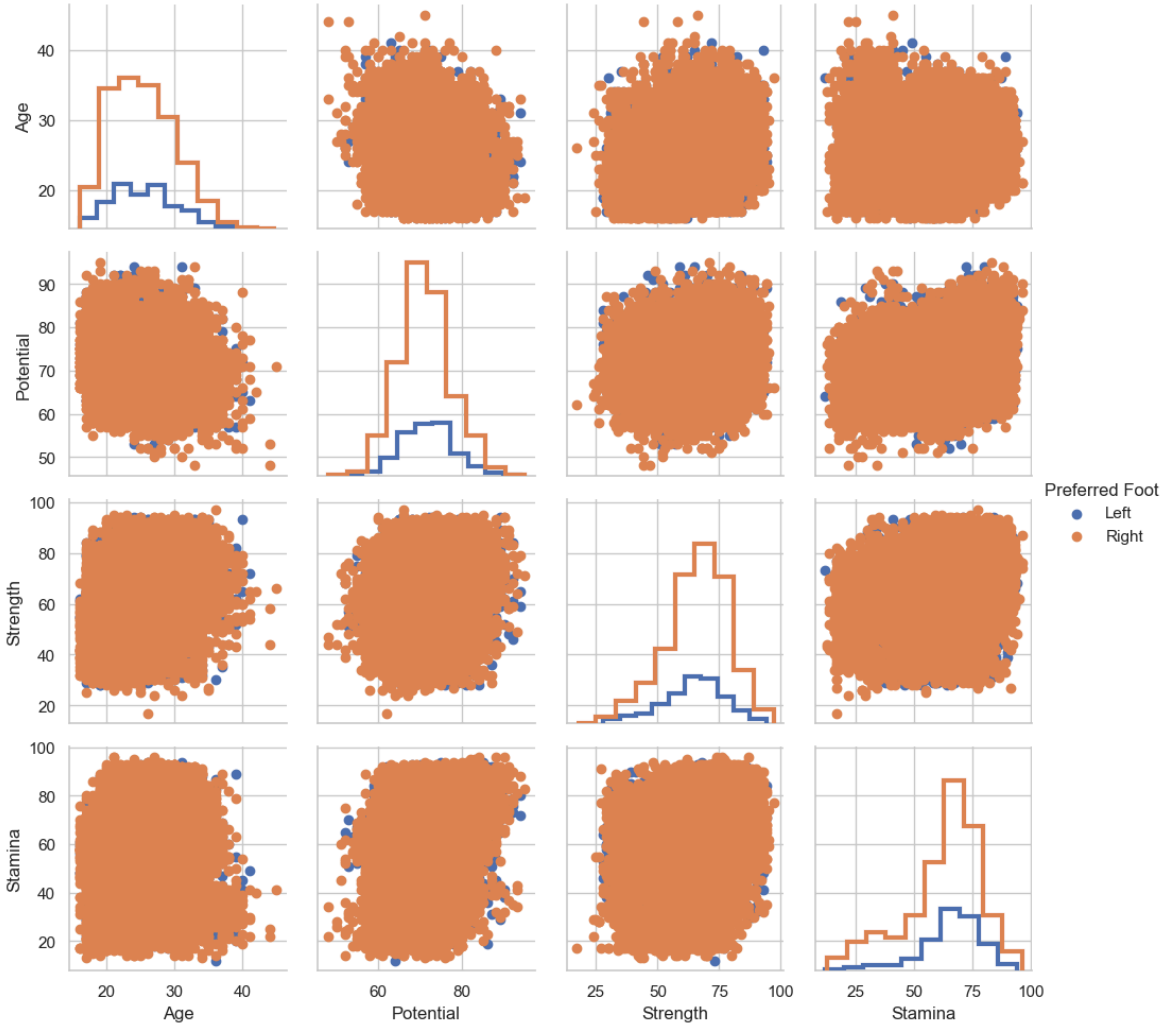
In [153…

```
#  points using the categorical variable Preferred Foot

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



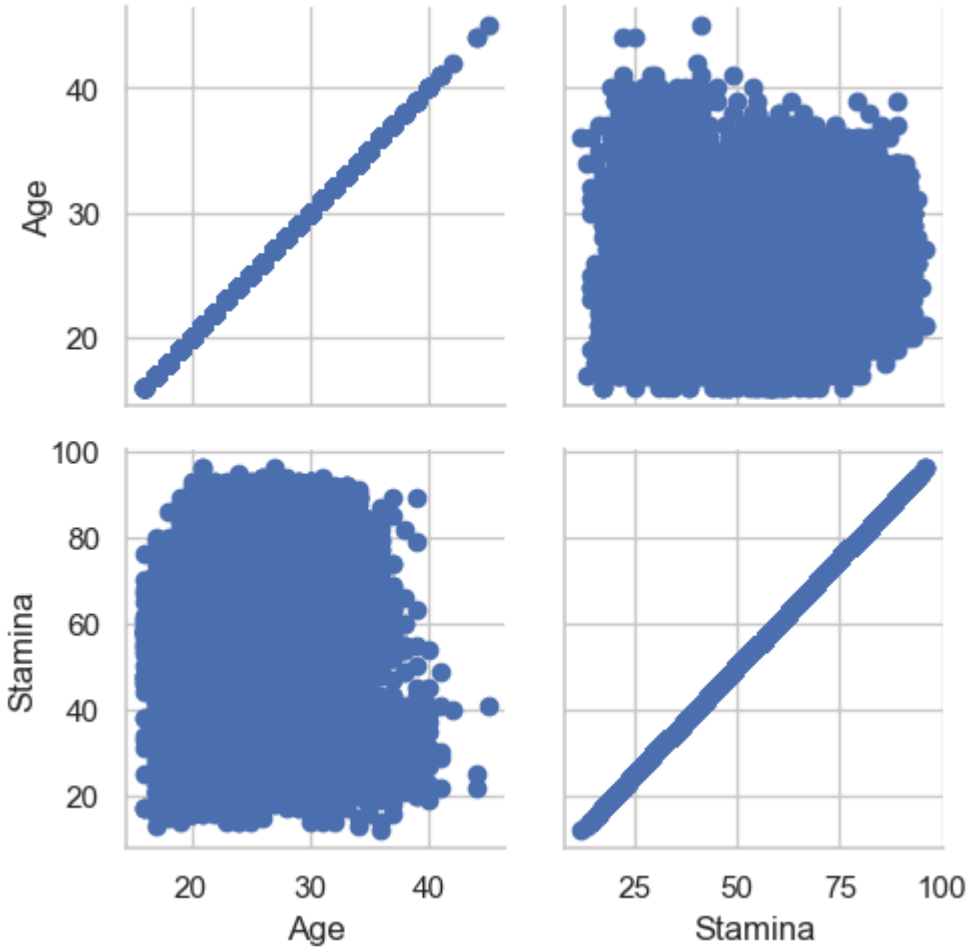
In [155…

```
#  different style to show multiple histograms

g = sns.PairGrid(fifa_new, hue="Preferred Foot")
g = g.map_diag(plt.hist, histtype="step", linewidth=3)
g = g.map_offdiag(plt.scatter)
g = g.add_legend()
```



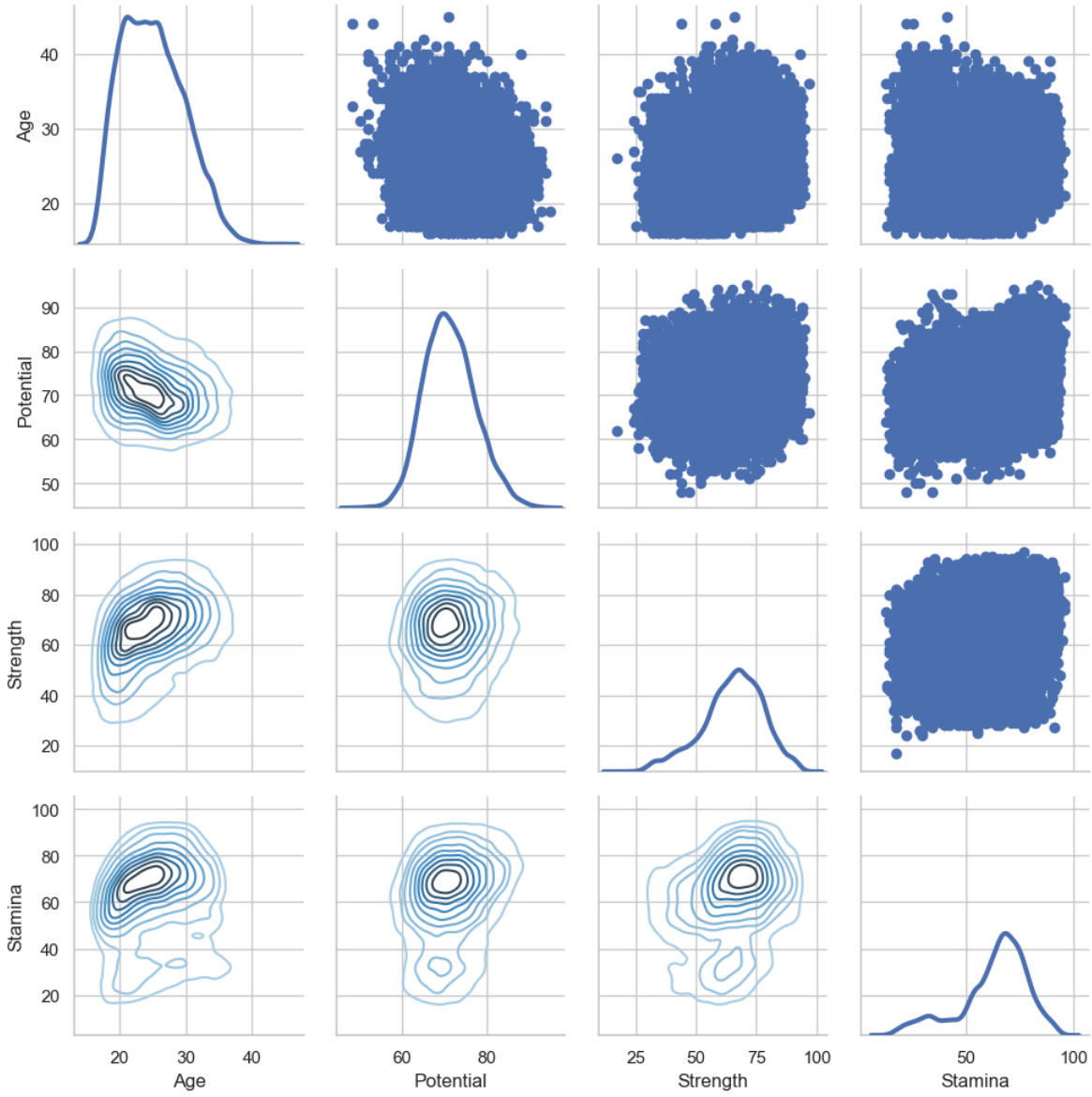
In [157…

```
#  plot a subset of variables

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



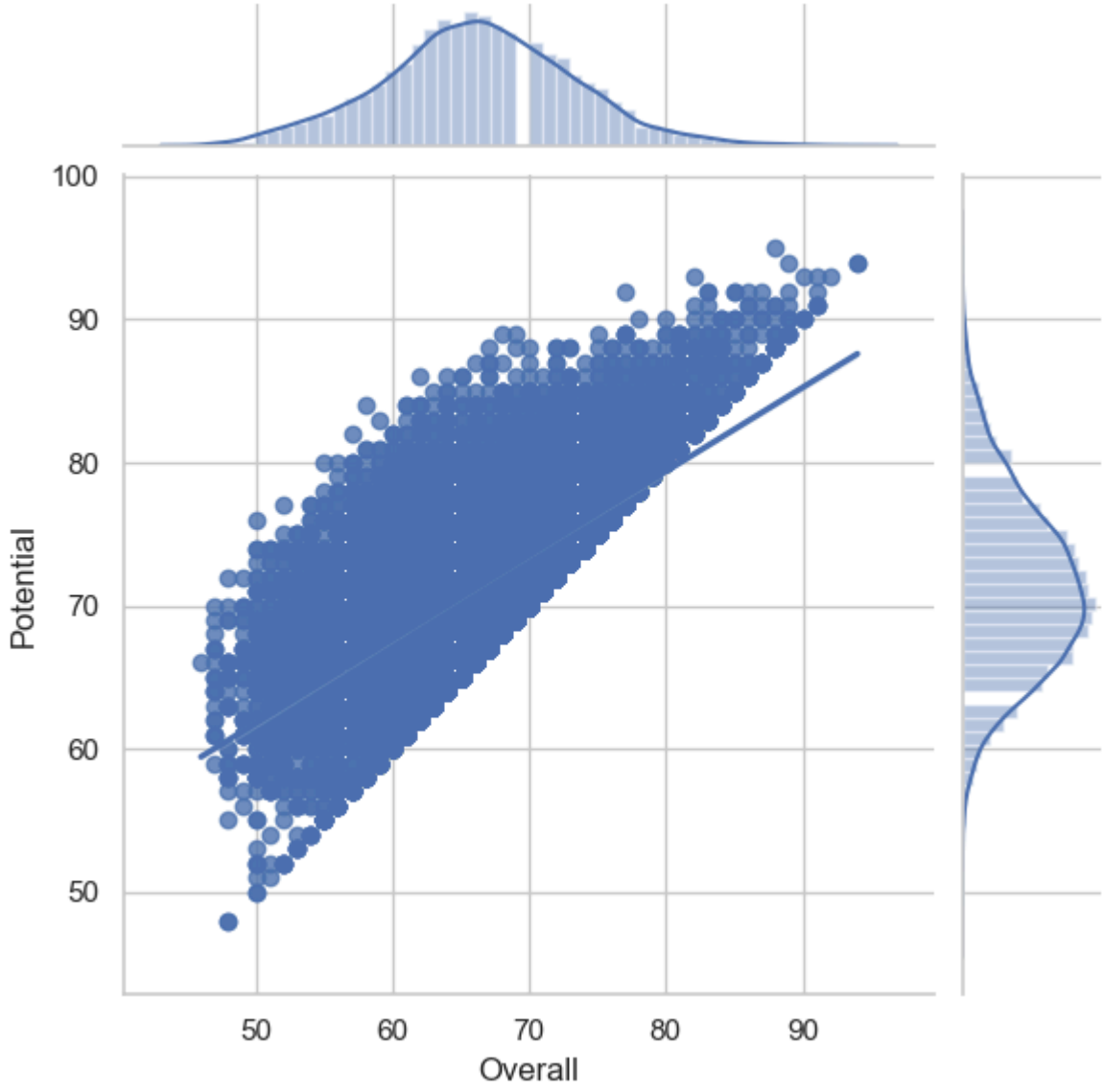
In [159…

```
#  different functions on the upper and lower triangles

g = sns.PairGrid(fifa_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)
```



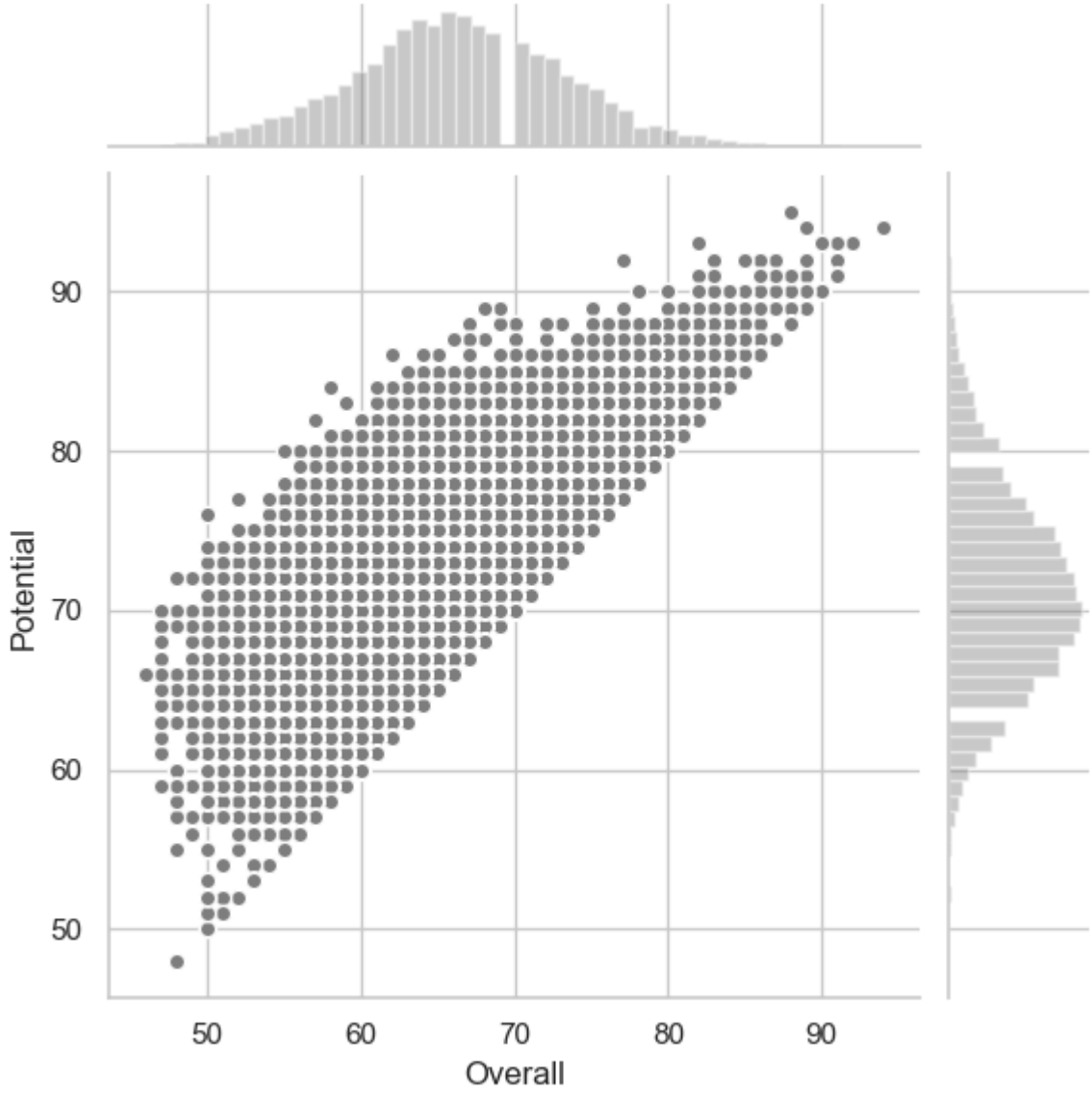
In [160…

```
#  jointgrid() function

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



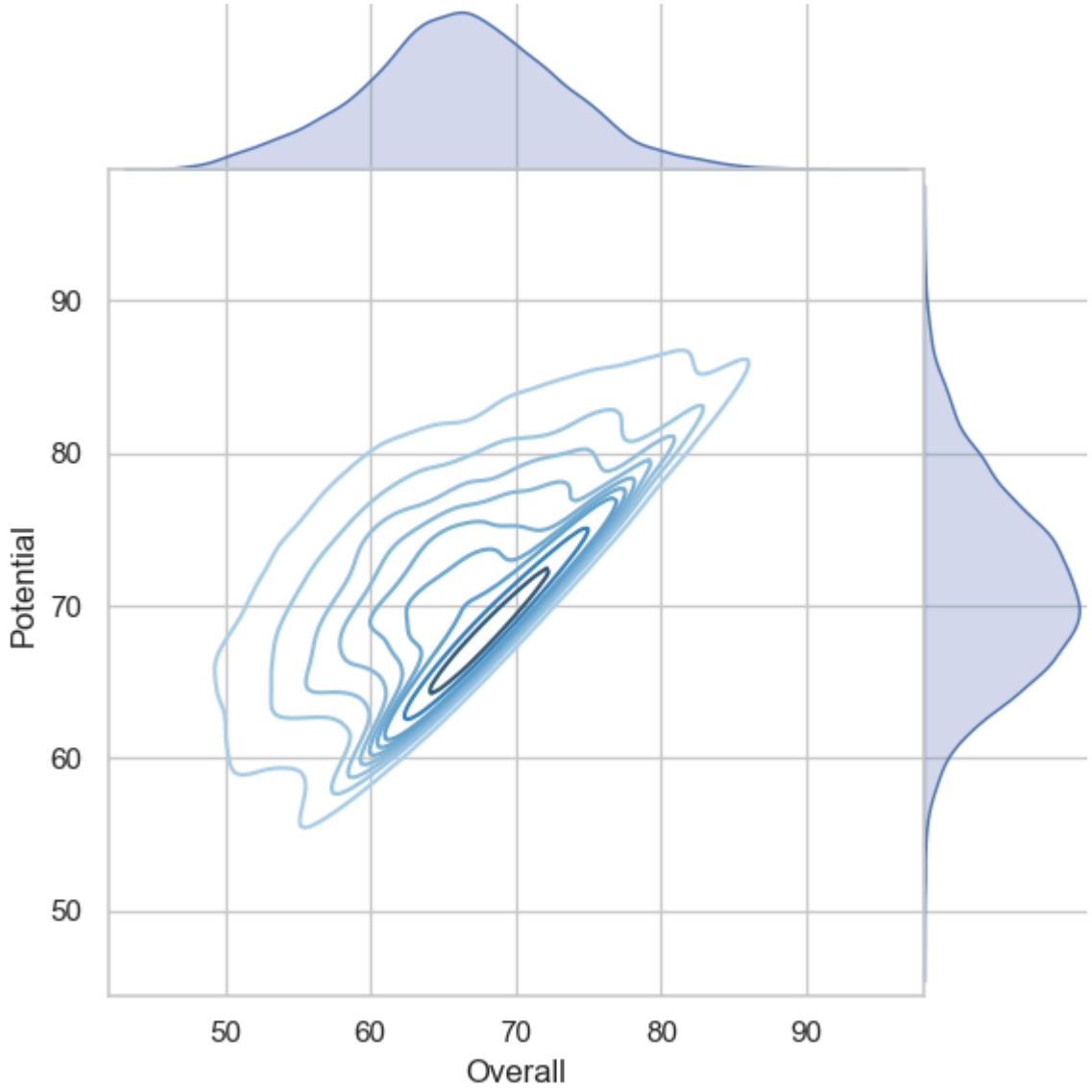
In [163…
```
import matplotlib.pyplot as plt
```

In [165…
```
#  he join and marginal plots separately, which allows finer-level control other

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



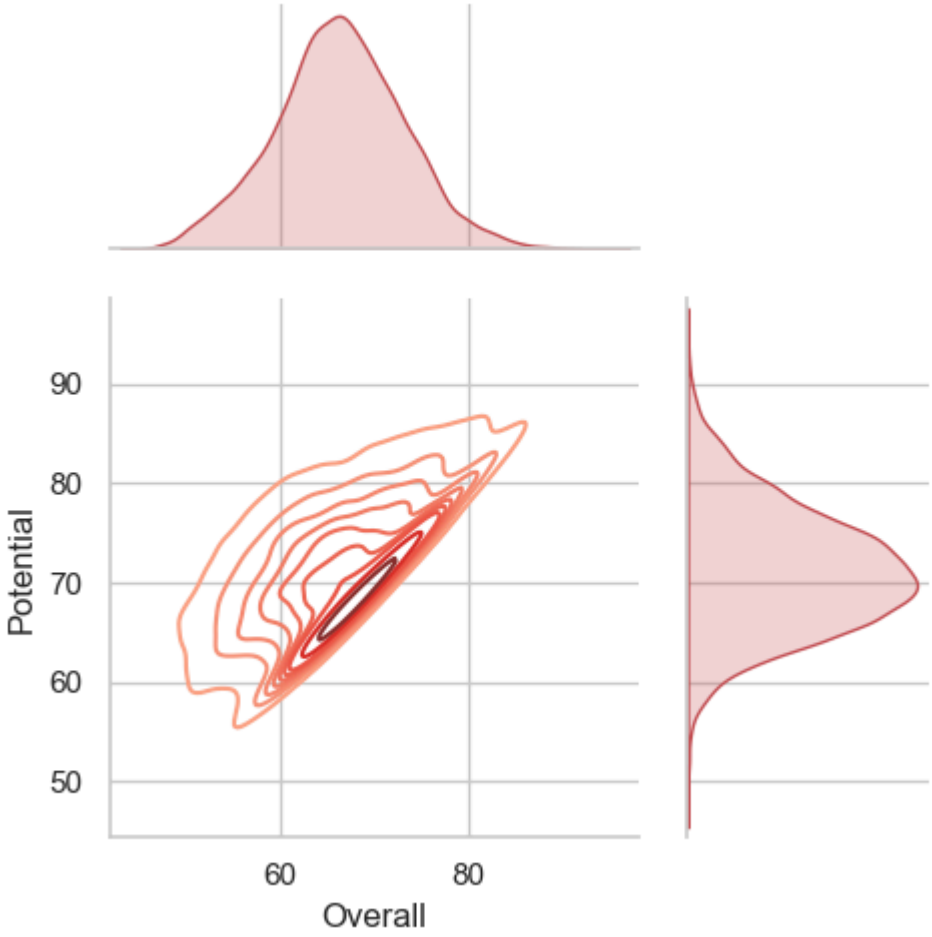
In [167…

```
#  remove the space between the joint and marginal axes

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



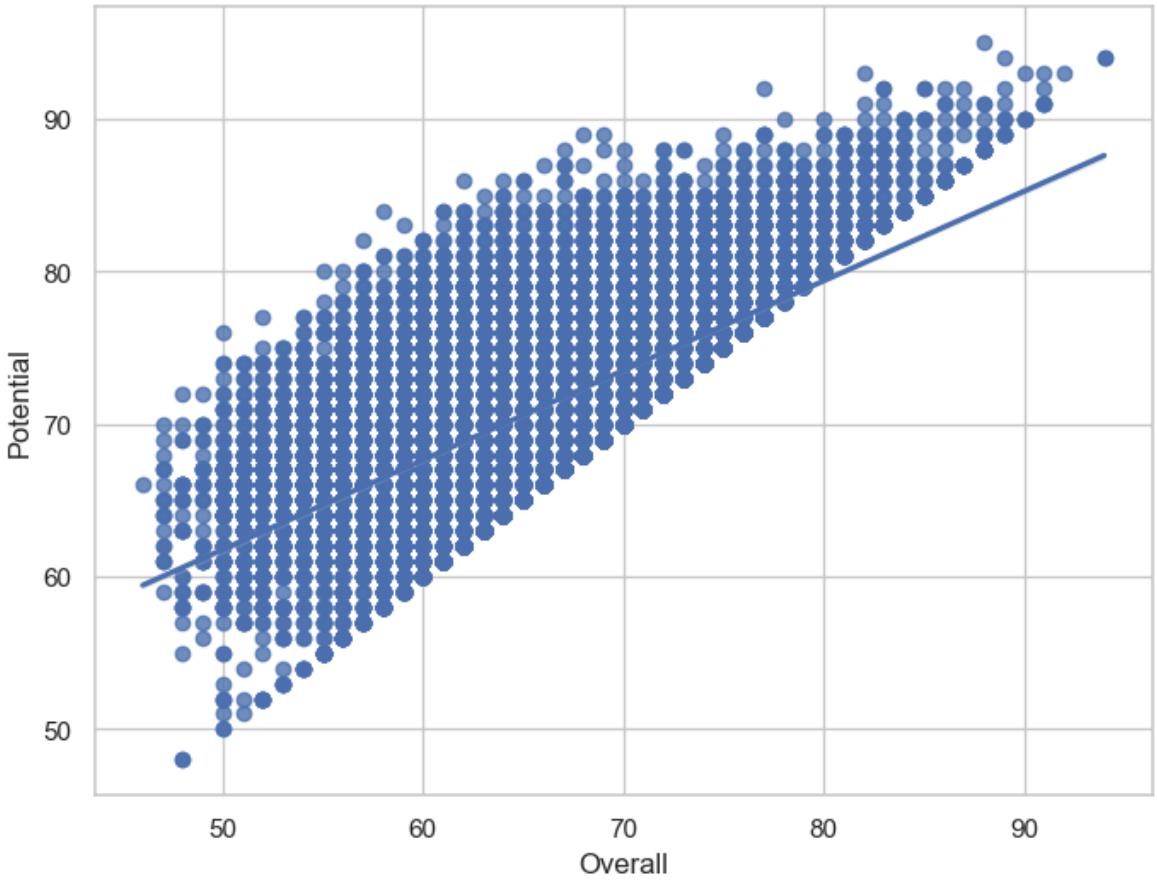
In [169…

```
#  Smaller plot with relatively larger marginal axes

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



In [175…

```
#  Controlling the size and shape of the plot
                # regplot() and lmplot()

f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa);
```

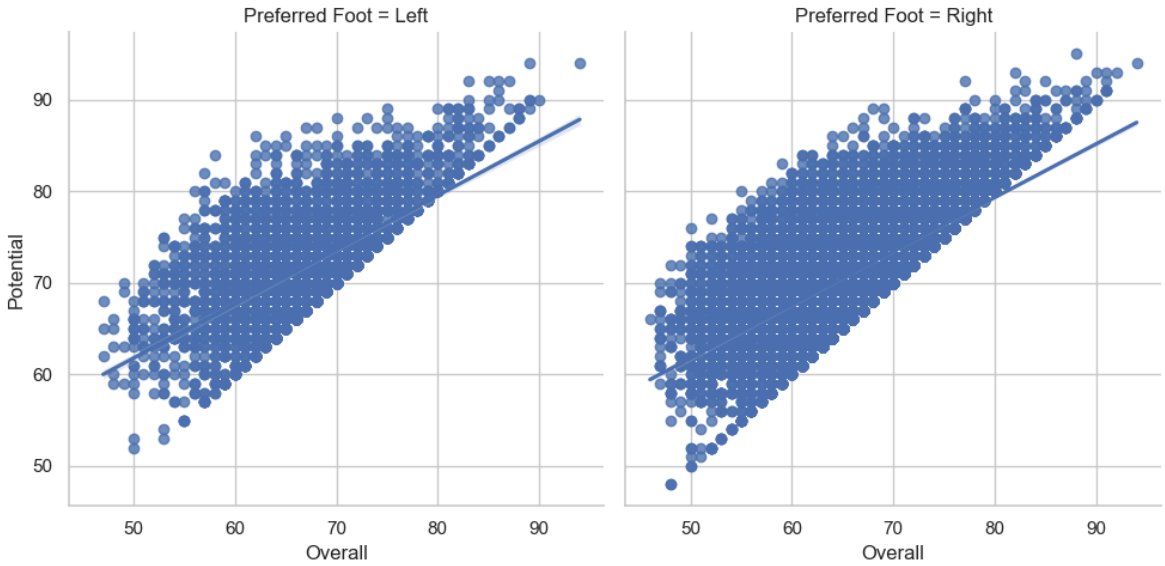


In [173…

```
#  lmplot()

sns.lmplot(x="Overall", y="Potential", col="Preferred Foot", data=fifa, col_wrap
```

Out[173…

```
<seaborn.axisgrid.FacetGrid at 0x21fe97e63c0>
```

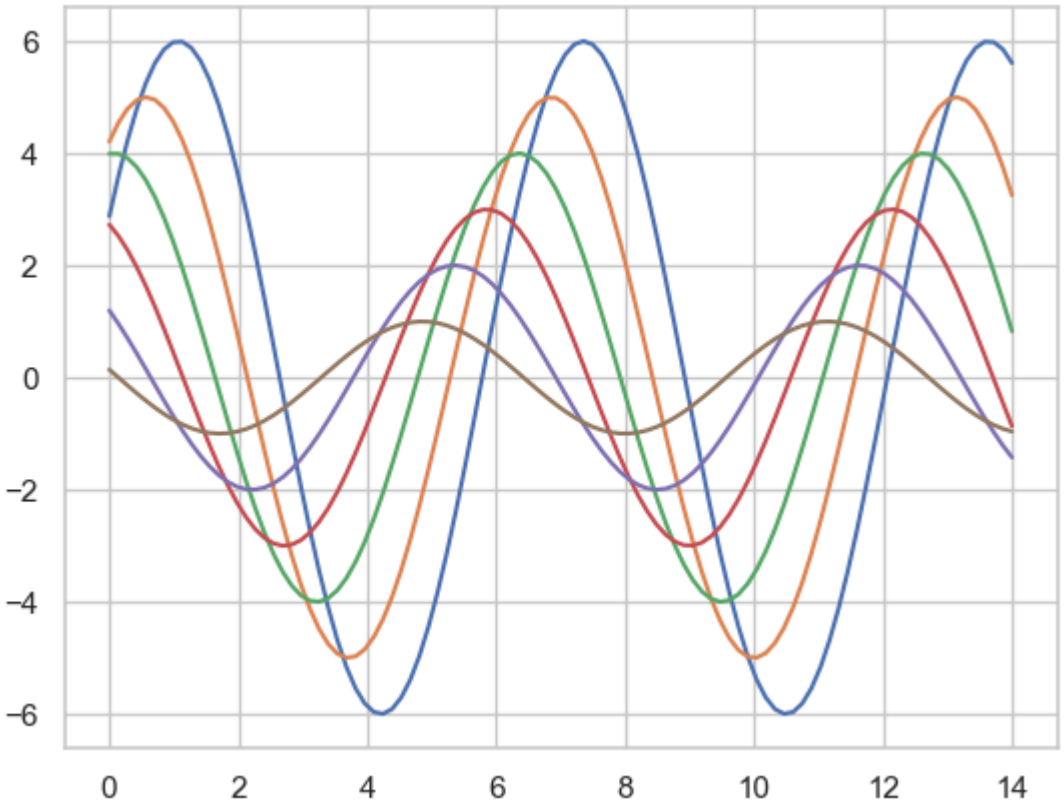


In [177…

```
#  figures styles
    #  darkgrid, hitegrid,dark,white and ticks

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)
```

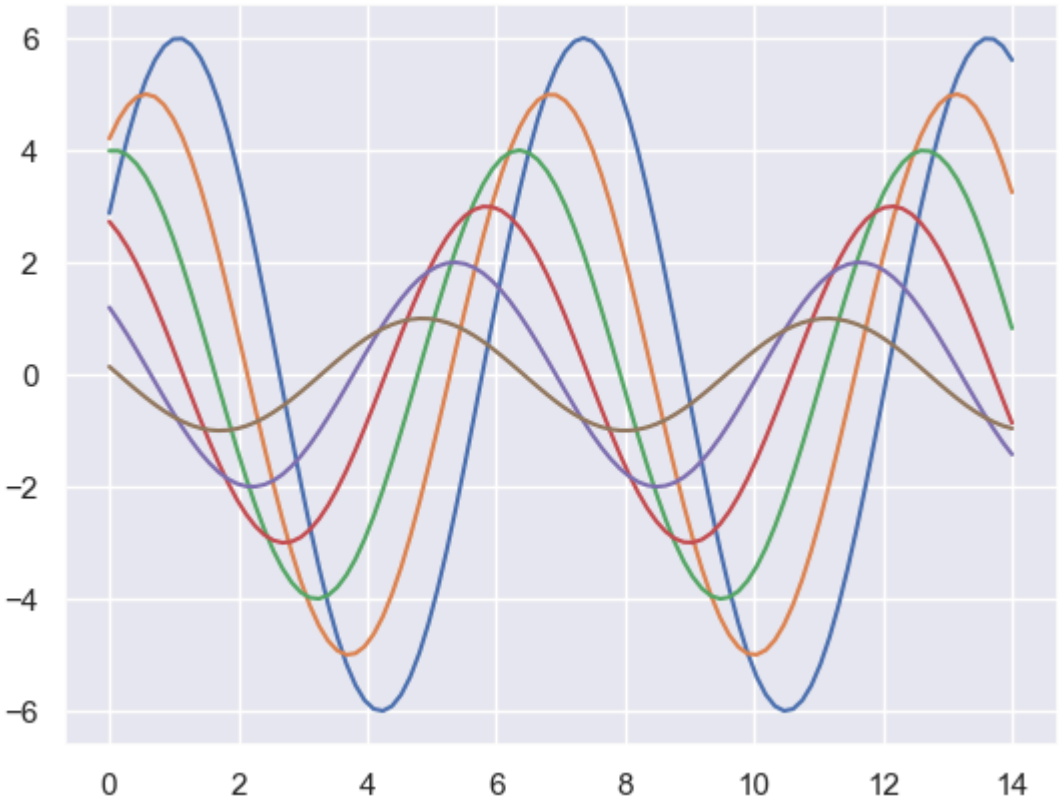
In [179…

```
#  plot looks like with matplotlib default parameters

sinplot()
```



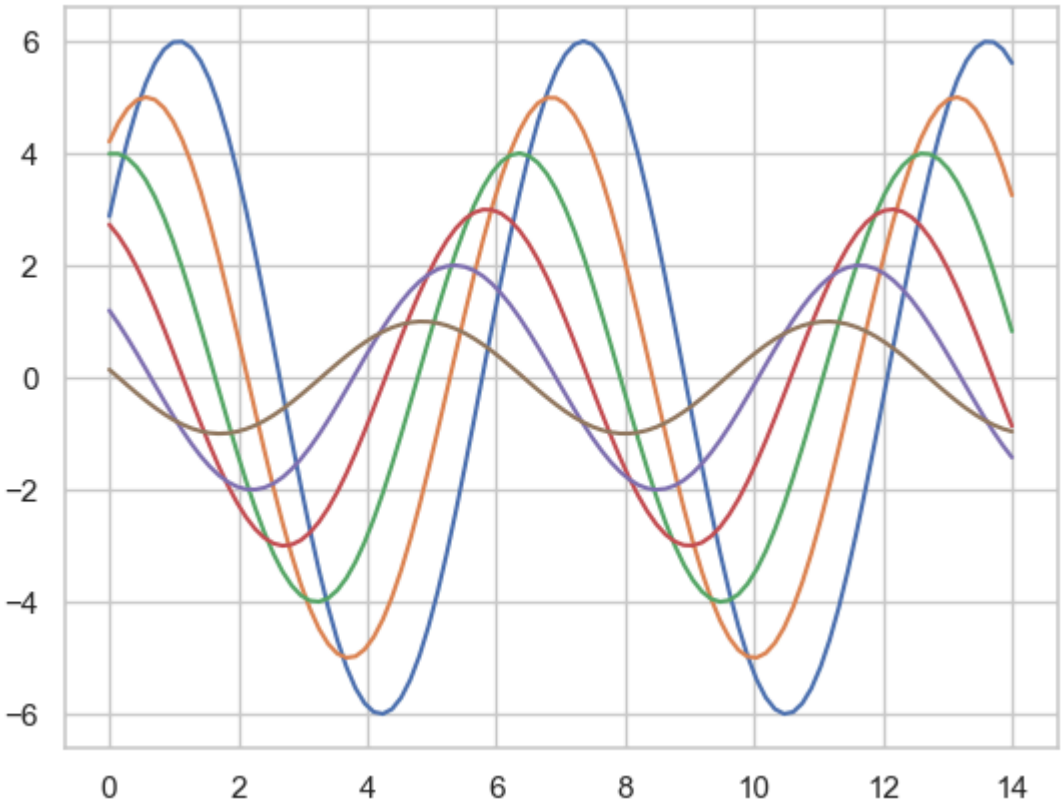
In [181…

```
#  set() function

sns.set()
sinplot()
```



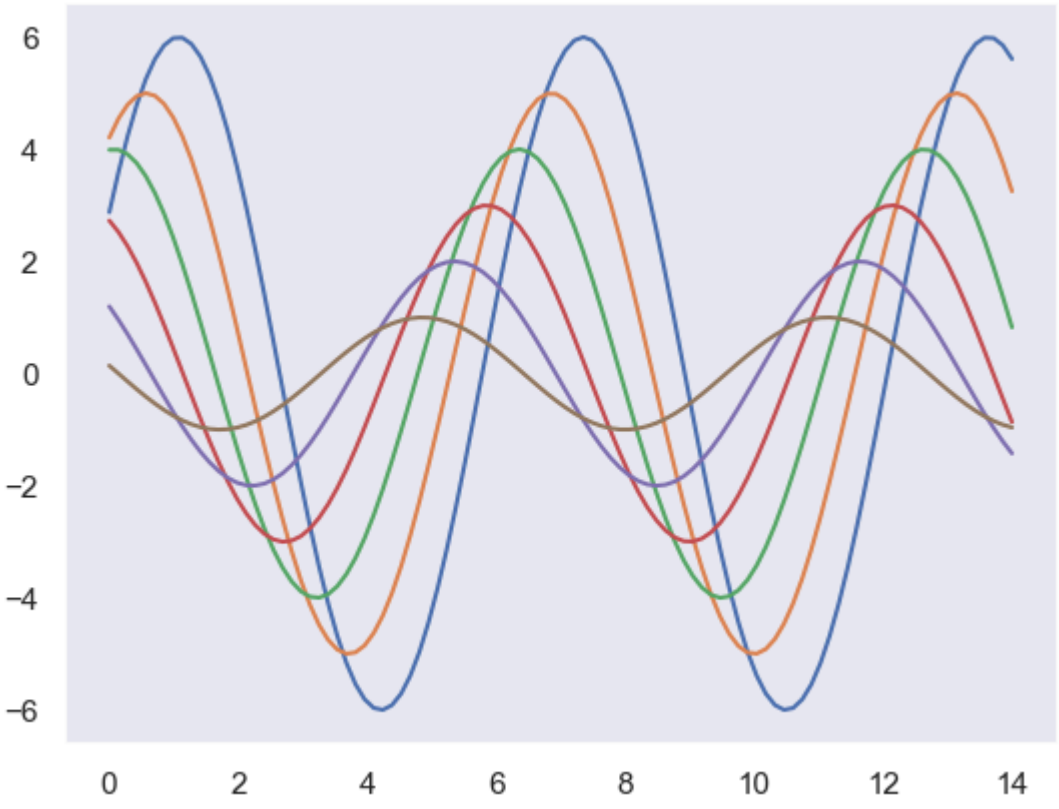
In [183…

```
#  different styles

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

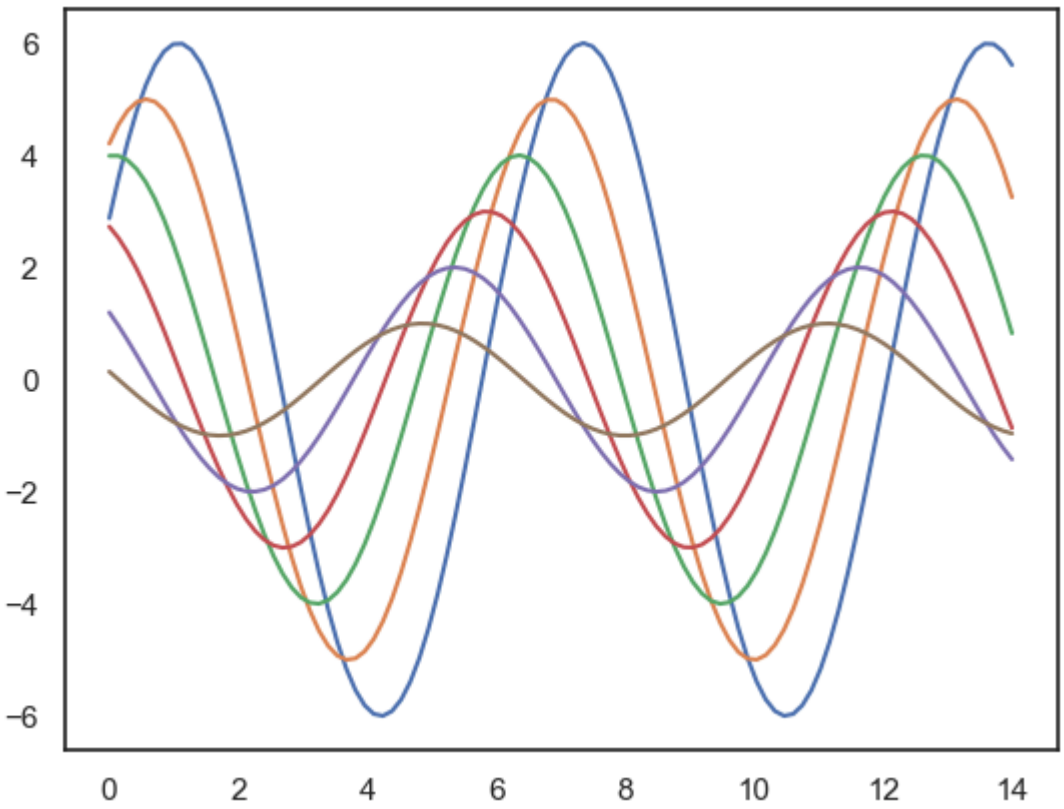


In [185…

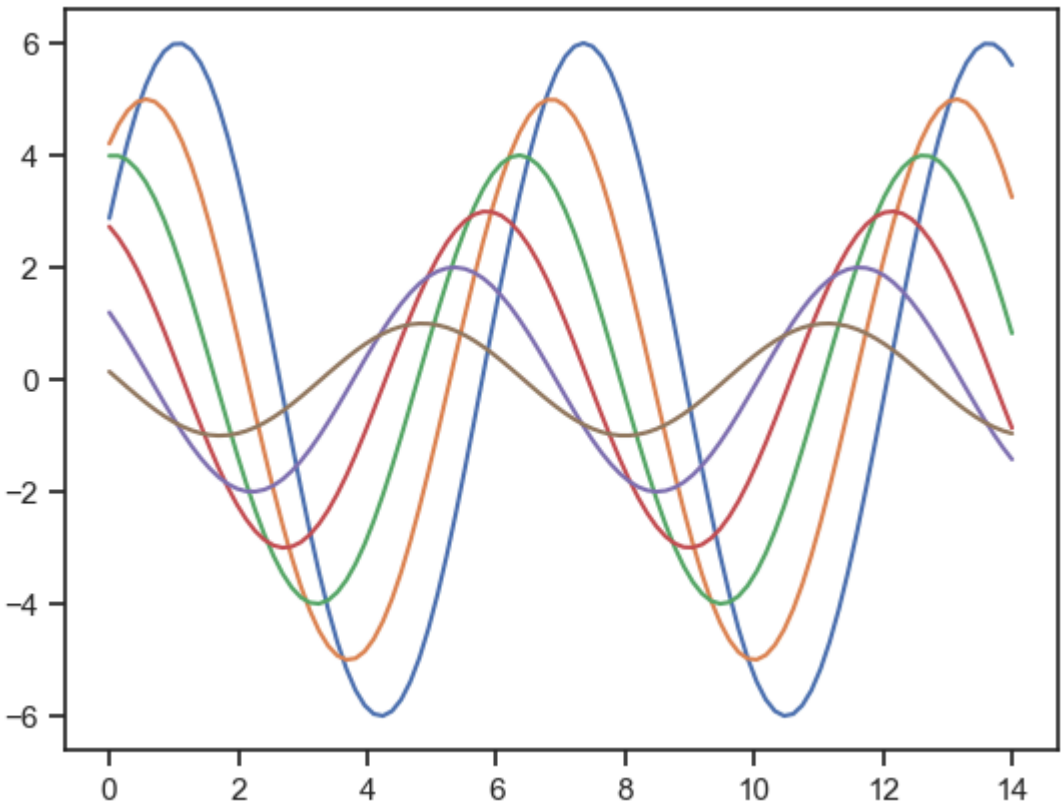
```
sns.set_style("dark")
sinplot()
```



In [187…
```
sns.set_style("white")
sinplot()
```



In [189…
```
sns.set_style("ticks")
sinplot()
```



In [ ]: