FIFA Sports Analysis and Visualization

In [1]:
```
import pandas as pd #for linear algebre
import numpy as np #data processing, csv file
import seaborn as sns
sns.set_style=('whitegrid')
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

FIFA-Players Analysis

In [2]:
```
fifa19 = pd.read_csv(r'C:\Users\Affan\OneDrive\Desktop\FSDS Course NIT\Prakash S
fifa19
```

Out[2]:

	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 [3]:
```
fifa19.head()
```

Out[3]:

	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 [4]:
```
fifa19.tail()
```

Out[4]:

	ID	Name	Age	Photo	Nationa
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

5 rows × 88 columns

In [5]:
```
fifa19.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 [6]:
```
fifa19.describe()
```

Out[6]:

	ID	Age	Overall	Potential	Special	Internat Reputa
count	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18159.00
mean	214298.338606	25.122206	66.238699	71.307299	1597.809908	1.11
std	29965.244204	4.669943	6.908930	6.136496	272.586016	0.39
min	16.000000	16.000000	46.000000	48.000000	731.000000	1.00
25%	200315.500000	21.000000	62.000000	67.000000	1457.000000	1.00
50%	221759.000000	25.000000	66.000000	71.000000	1635.000000	1.00
75%	236529.500000	28.000000	71.000000	75.000000	1787.000000	1.00
max	246620.000000	45.000000	94.000000	95.000000	2346.000000	5.00

8 rows × 43 columns

In [7]:
```
fifa19.columns
```

Out[7]:
```
Index(['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 [8]:
```
fifa19['Nationality'].unique()
```

Out[8]:
```
array(['Argentina', 'Portugal', 'Brazil', 'Spain', 'Belgium', 'Croatia',
       'Uruguay', 'Slovenia', 'Poland', 'Germany', 'France', 'England',
       'Italy', 'Egypt', 'Colombia', 'Denmark', 'Gabon', 'Wales',
       'Senegal', 'Costa Rica', 'Slovakia', 'Netherlands',
       'Bosnia Herzegovina', 'Morocco', 'Serbia', 'Algeria', 'Austria',
       'Greece', 'Chile', 'Sweden', 'Korea Republic', 'Finland', 'Guinea',
       'Montenegro', 'Armenia', 'Switzerland', 'Norway', 'Czech Republic',
       'Scotland', 'Ghana', 'Central African Rep.', 'DR Congo',
       'Ivory Coast', 'Russia', 'Ukraine', 'Iceland', 'Mexico', 'Jamaica',
       'Albania', 'Venezuela', 'Japan', 'Turkey', 'Ecuador', 'Paraguay',
       'Mali', 'Nigeria', 'Cameroon', 'Dominican Republic', 'Israel',
       'Kenya', 'Hungary', 'Republic of Ireland', 'Romania',
       'United States', 'Cape Verde', 'Australia', 'Peru', 'Togo',
       'Syria', 'Zimbabwe', 'Angola', 'Burkina Faso', 'Iran', 'Estonia',
       'Tunisia', 'Equatorial Guinea', 'New Zealand', 'FYR Macedonia',
       'United Arab Emirates', 'China PR', 'Guinea Bissau', 'Bulgaria',
       'Kosovo', 'South Africa', 'Madagascar', 'Georgia', 'Tanzania',
       'Gambia', 'Cuba', 'Belarus', 'Uzbekistan', 'Benin', 'Congo',
       'Mozambique', 'Honduras', 'Canada', 'Northern Ireland', 'Cyprus',
       'Saudi Arabia', 'Curacao', 'Moldova', 'Bolivia',
       'Trinidad & Tobago', 'Sierra Leone', 'Zambia', 'Chad',
       'Philippines', 'Haiti', 'Comoros', 'Libya', 'Panama',
       'São Tomé & Príncipe', 'Eritrea', 'Oman', 'Iraq', 'Burundi',
       'Fiji', 'New Caledonia', 'Lithuania', 'Luxembourg', 'Korea DPR',
       'Liechtenstein', 'St Kitts Nevis', 'Latvia', 'Suriname', 'Uganda',
       'El Salvador', 'Bermuda', 'Kuwait', 'Antigua & Barbuda',
       'Thailand', 'Mauritius', 'Guatemala', 'Liberia', 'Kazakhstan',
       'Niger', 'Mauritania', 'Montserrat', 'Namibia', 'Azerbaijan',
       'Guam', 'Faroe Islands', 'India', 'Nicaragua', 'Barbados',
       'Lebanon', 'Palestine', 'Guyana', 'Sudan', 'St Lucia', 'Ethiopia',
       'Puerto Rico', 'Grenada', 'Jordan', 'Rwanda', 'Qatar',
       'Afghanistan', 'Hong Kong', 'Andorra', 'Malta', 'Belize',
       'South Sudan', 'Indonesia', 'Botswana'], dtype=object)
```

In [9]:
```
fifa19['Body Type'].value_counts()
```

Out[9]:
```
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 [10]:
```
fifa19.columns.isna()
```

Out[10]:
```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False])
```

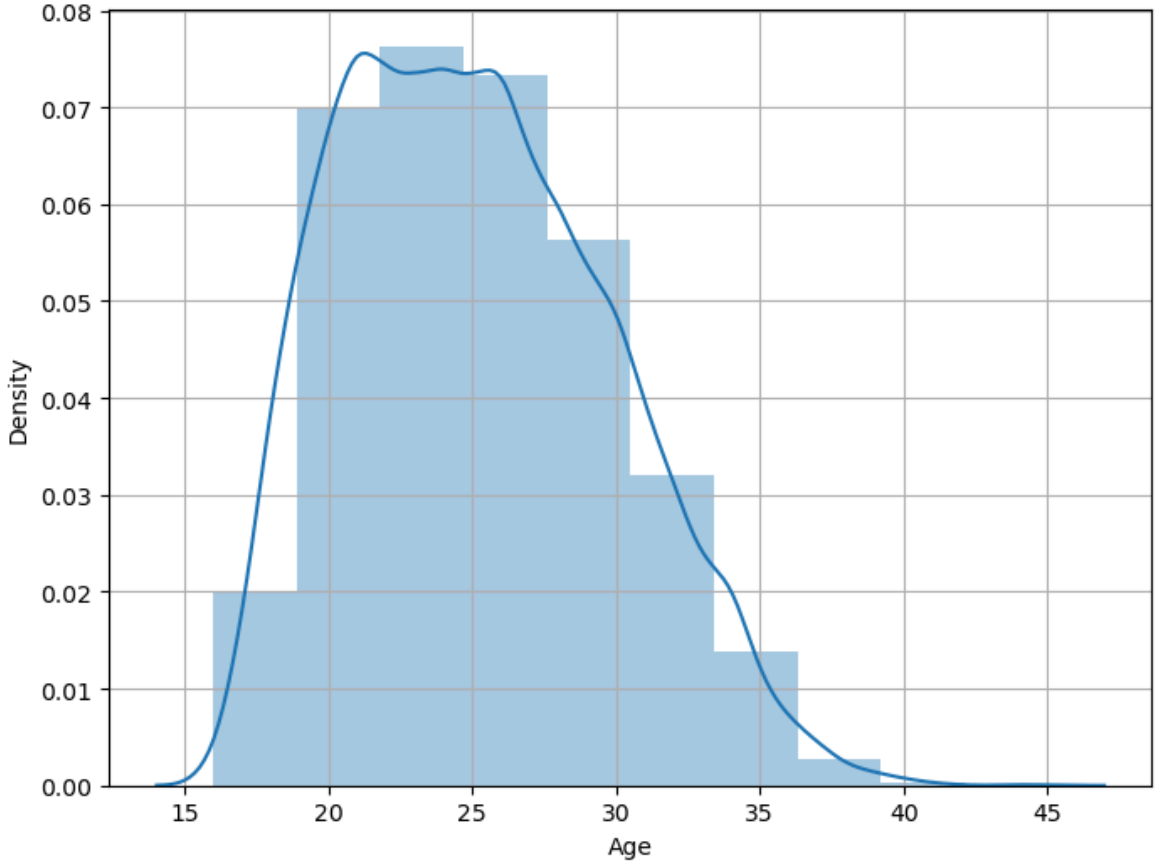
In [11]:
```
fifa19.info()
```

```
<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
```

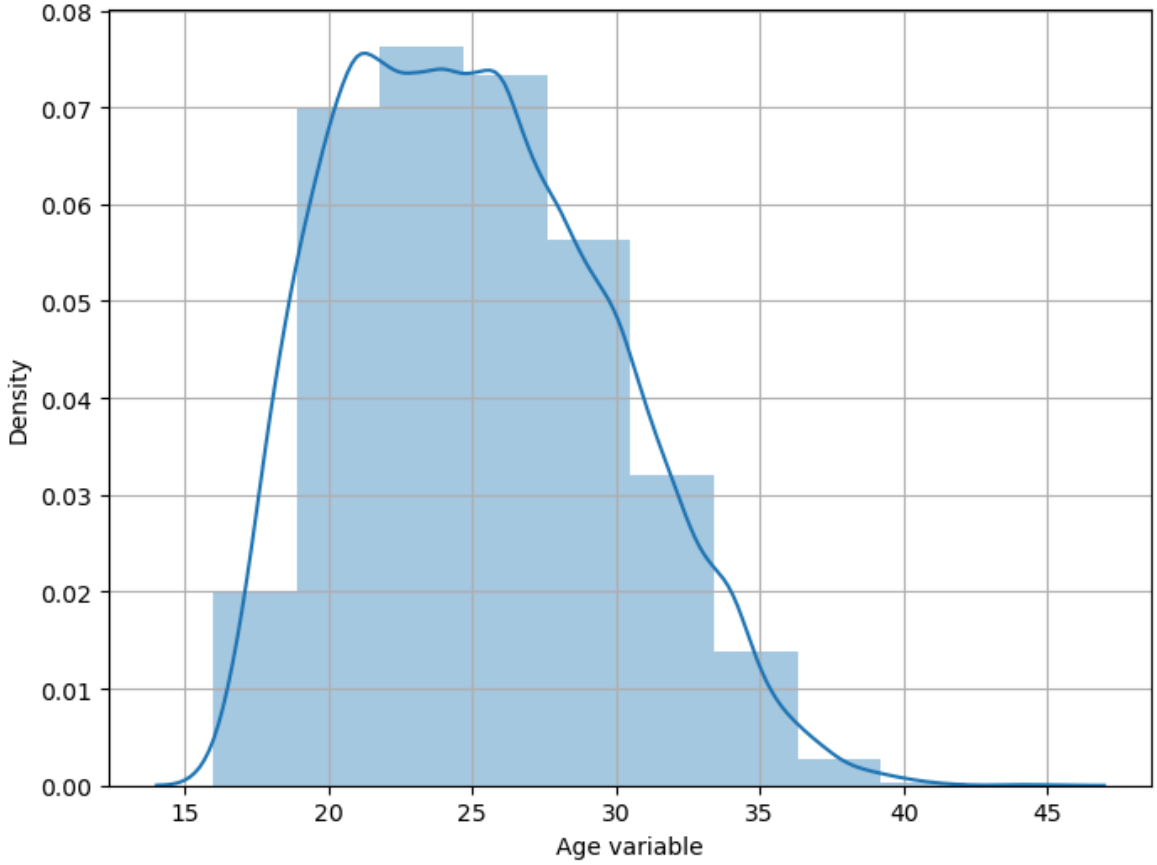
```
 55  HeadingAccuracy           18159 non-null  float64
 56  ShortPassing              18159 non-null  float64
 57  Volleys                   18159 non-null  float64
 58  Dribbling                 18159 non-null  float64
 59  Curve                     18159 non-null  float64
 60  FKAccuracy                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
```

from the above info we can say that this dataset has 89 total variables out of which 44 are numerical var from which 38 are float, 6 are int dtype and remaning 45 out of 89 var is char dtype ----Below we are going to explore about AGE variable----

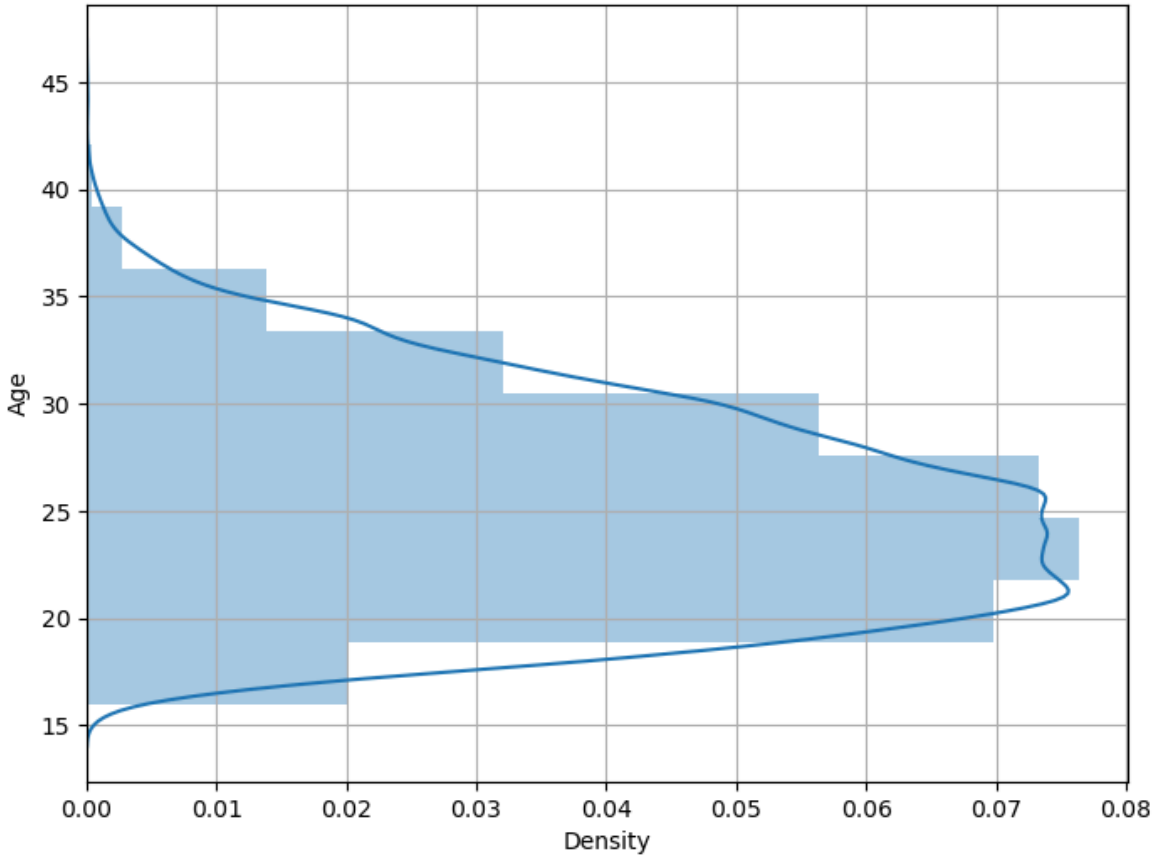
In [12]:
```
f,ax = plt.subplots(figsize=(8,6))
x=fifa19['Age']
ax=sns.distplot(x,bins=10)
plt.grid(ax)
plt.show()
```



In [13]:
```
f,ax = plt.subplots(figsize=(8,6))
x=fifa19['Age']
x=pd.Series(x,name='Age variable')
ax=sns.distplot(x,bins=10)
plt.grid(ax)
plt.show()
```



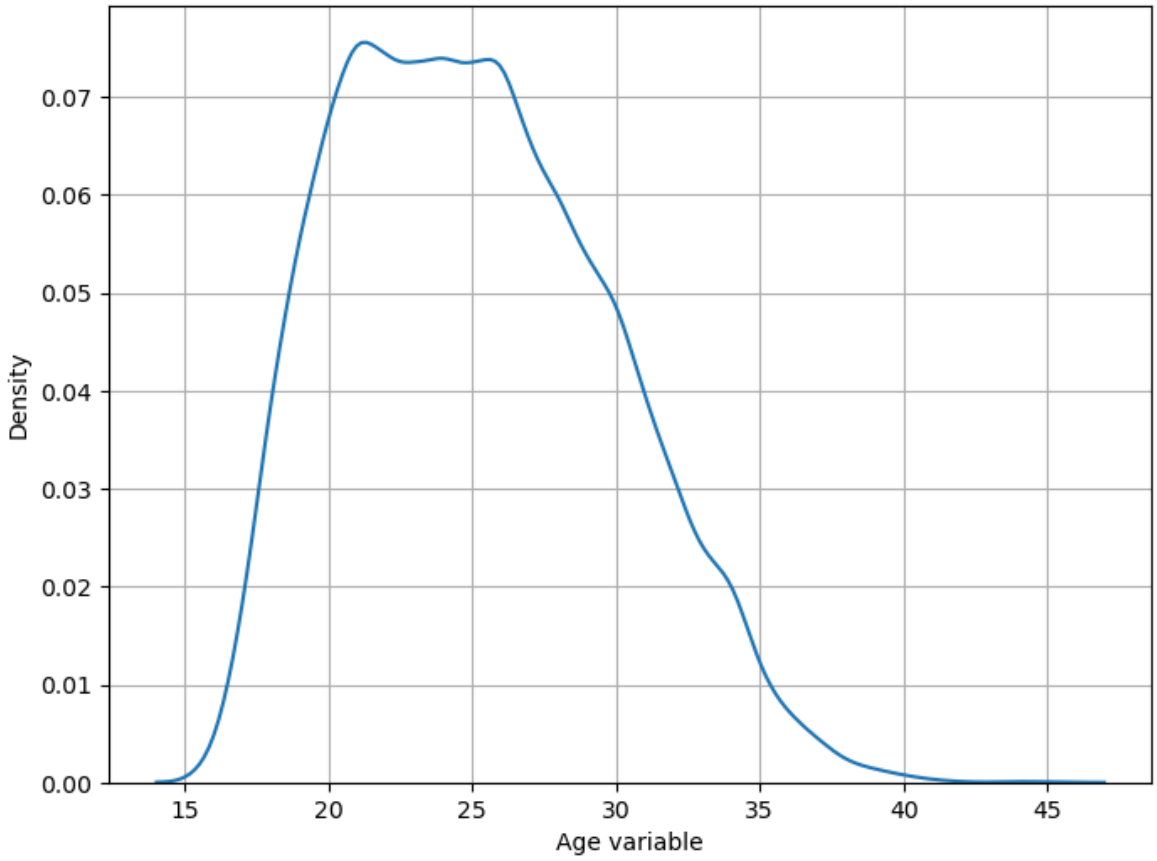
In [14]:
```
f,ax = plt.subplots(figsize=(8,6))
x=fifa19['Age']
ax=sns.distplot(x,bins=10,vertical=True)
plt.grid(ax)
plt.show()
```



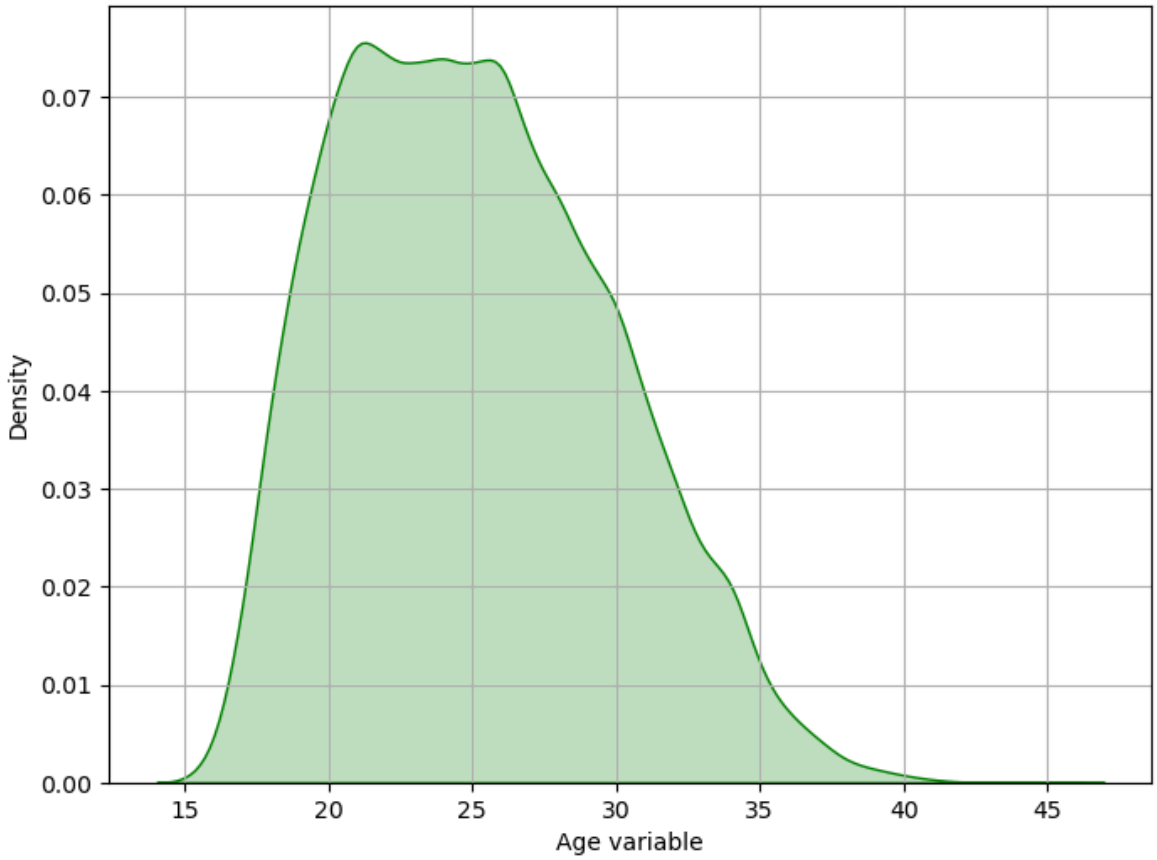
KDE PLOT

kde plot is used for plotting univariate and bivariate analysis

In [15]:
```
f,ax = plt.subplots(figsize=(8,6))
x=fifa19['Age']
x=pd.Series(x,name='Age variable')
ax=sns.kdeplot(x)
plt.grid(ax)
plt.show()
```

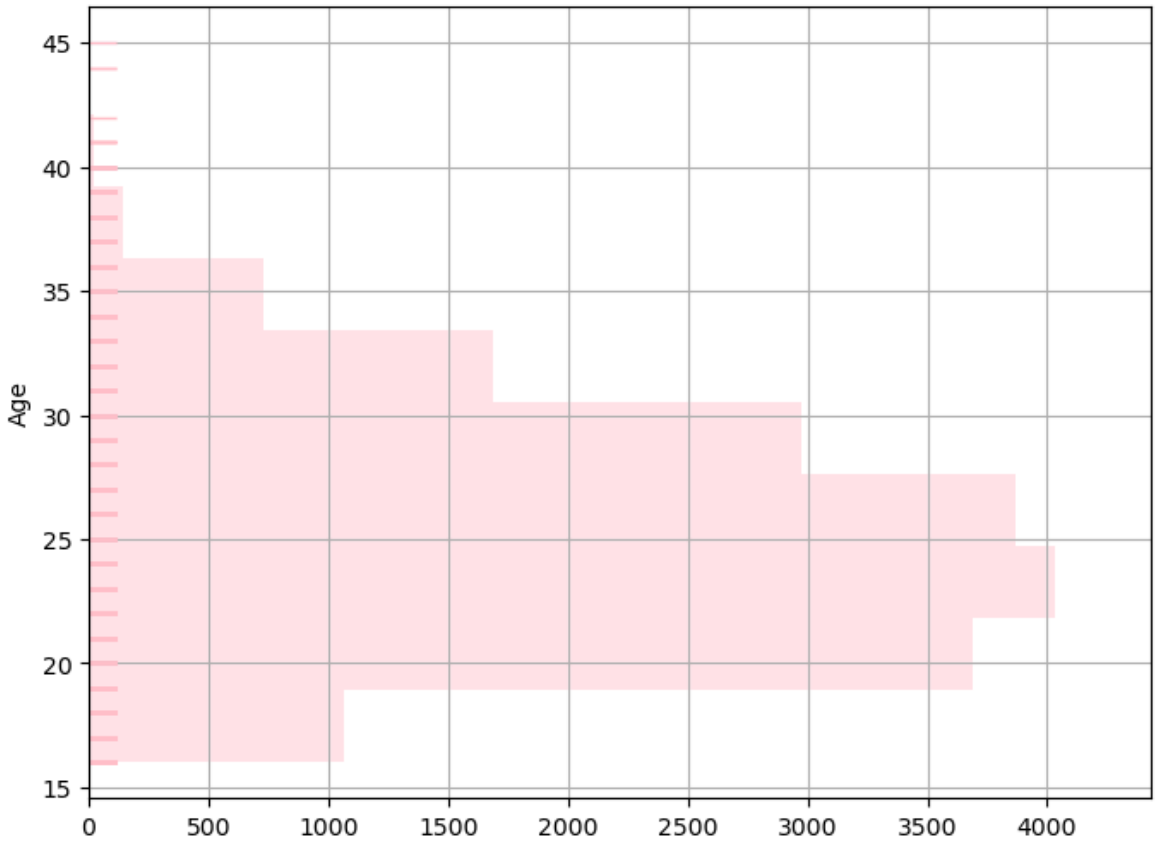


In [16]:
```
f,ax = plt.subplots(figsize=(8,6))
x=fifa19['Age']
x=pd.Series(x,name='Age variable')
ax=sns.kdeplot(x,shade=True,color='g')
plt.grid(ax)
plt.show()
```



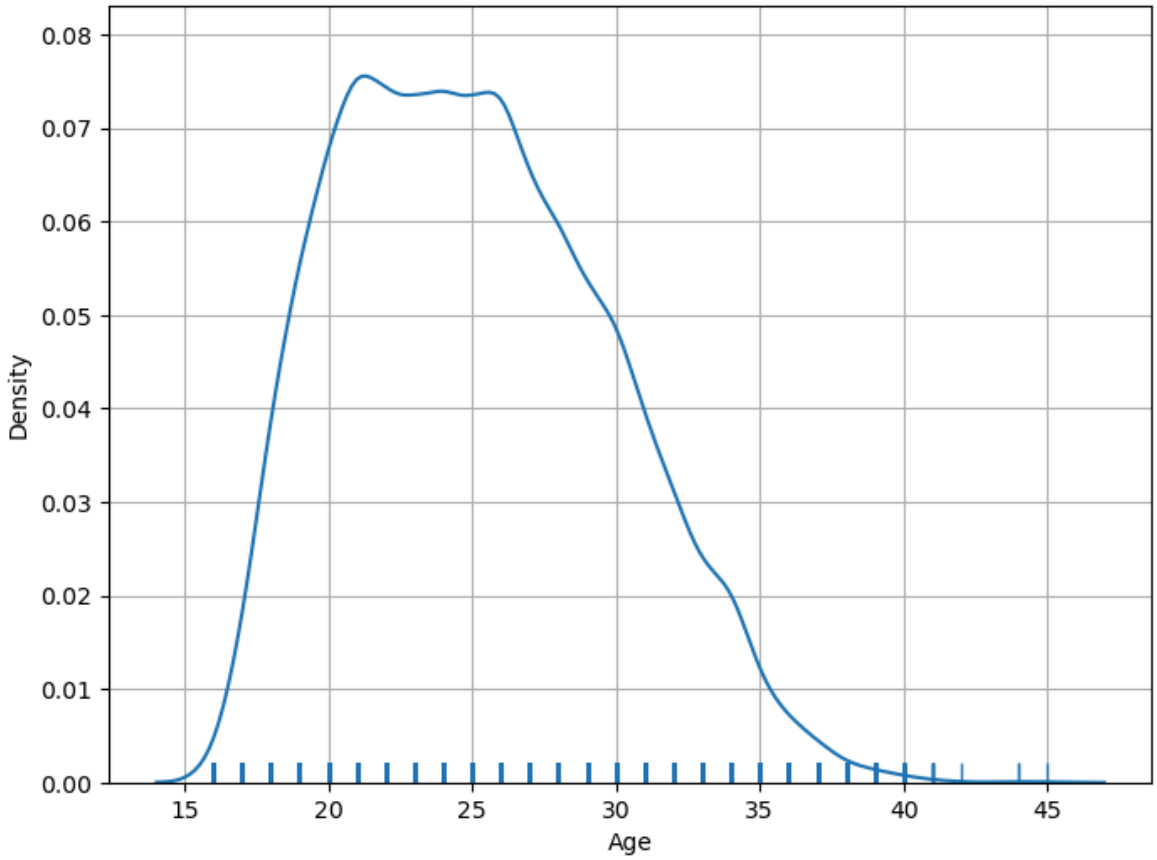
histogram

In [17]:
```
f,ax = plt.subplots(figsize=(8,6))
x=fifa19['Age']
ax=sns.distplot(x,kde=False,rug=True,bins=10,color='pink',vertical=True)
plt.grid(ax)
plt.show()
```



alternate way to plot KDE as follows:

In [18]:
```
f, ax = plt.subplots(figsize=(8,6))
x = fifa19['Age']
ax = sns.distplot(x, hist=False, rug=True, bins=10)
plt.grid(ax)
plt.show()
```



----Below we are going to expore about 'PREFERRED FOOT' variable----

In [19]:
```
fifa19['Preferred Foot'].nunique() #no. of unique val
```

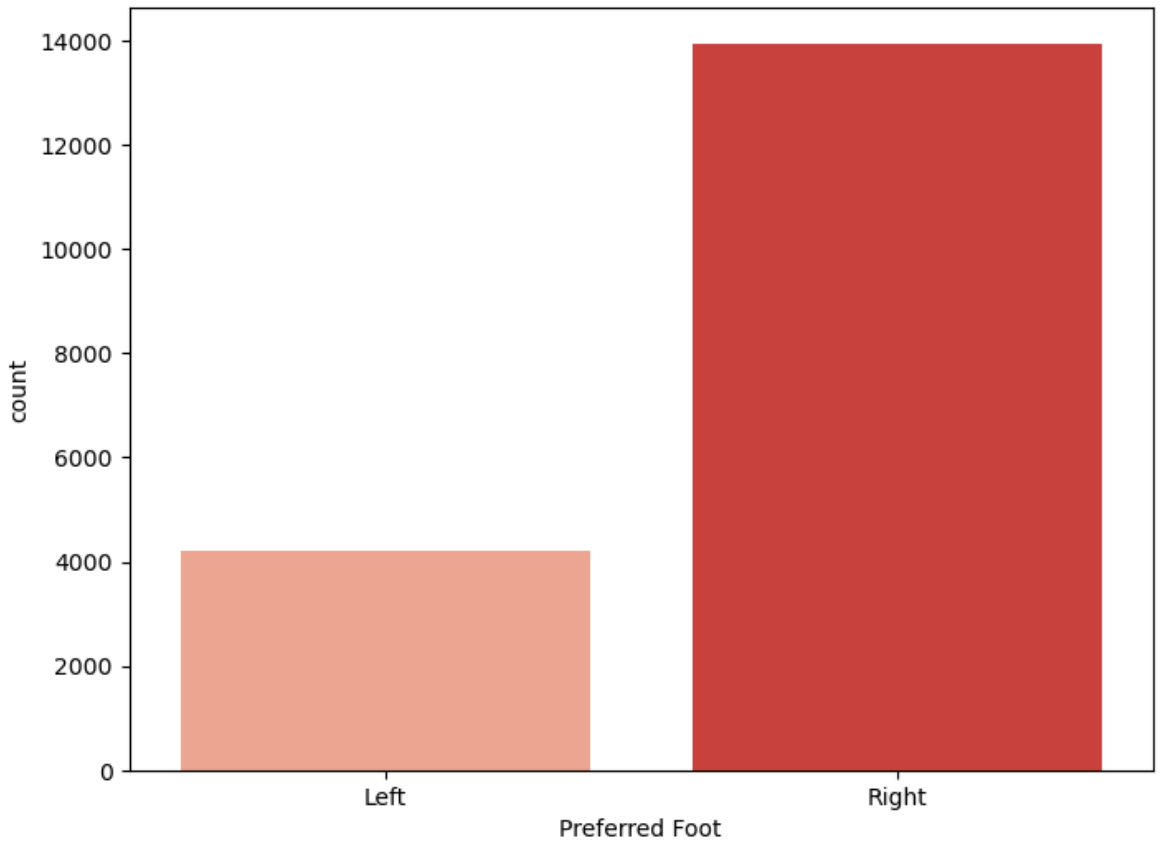
Out[19]: 2

In [20]:
```
fifa19['Preferred Foot'].value_counts() #the var contains two value LEFT n RIGHT
```

Out[20]:
```
Preferred Foot
Right    13948
Left      4211
Name: count, dtype: int64
```

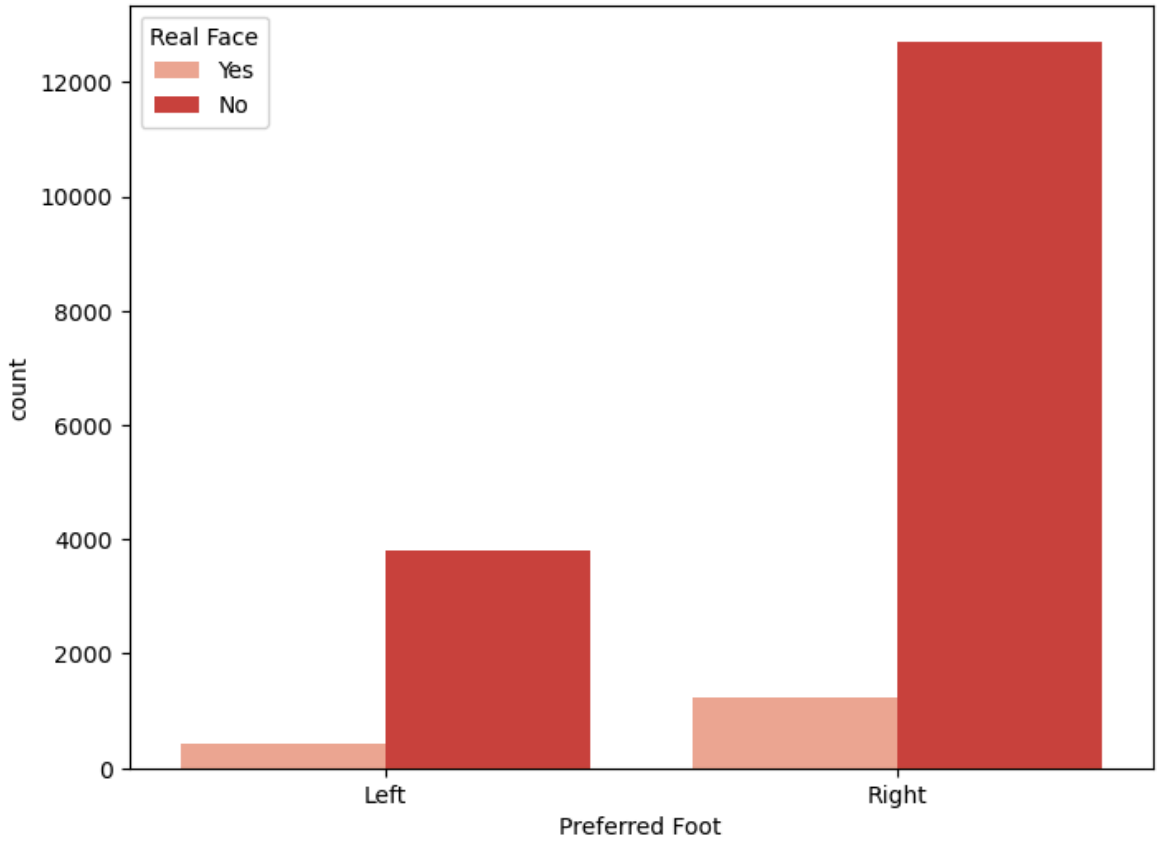
-----Here we are visualizing distribution of values with countplot()

In [21]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.countplot(x='Preferred Foot', data=fifa19,color='c',palette='Reds',fill=True
plt.show()
```

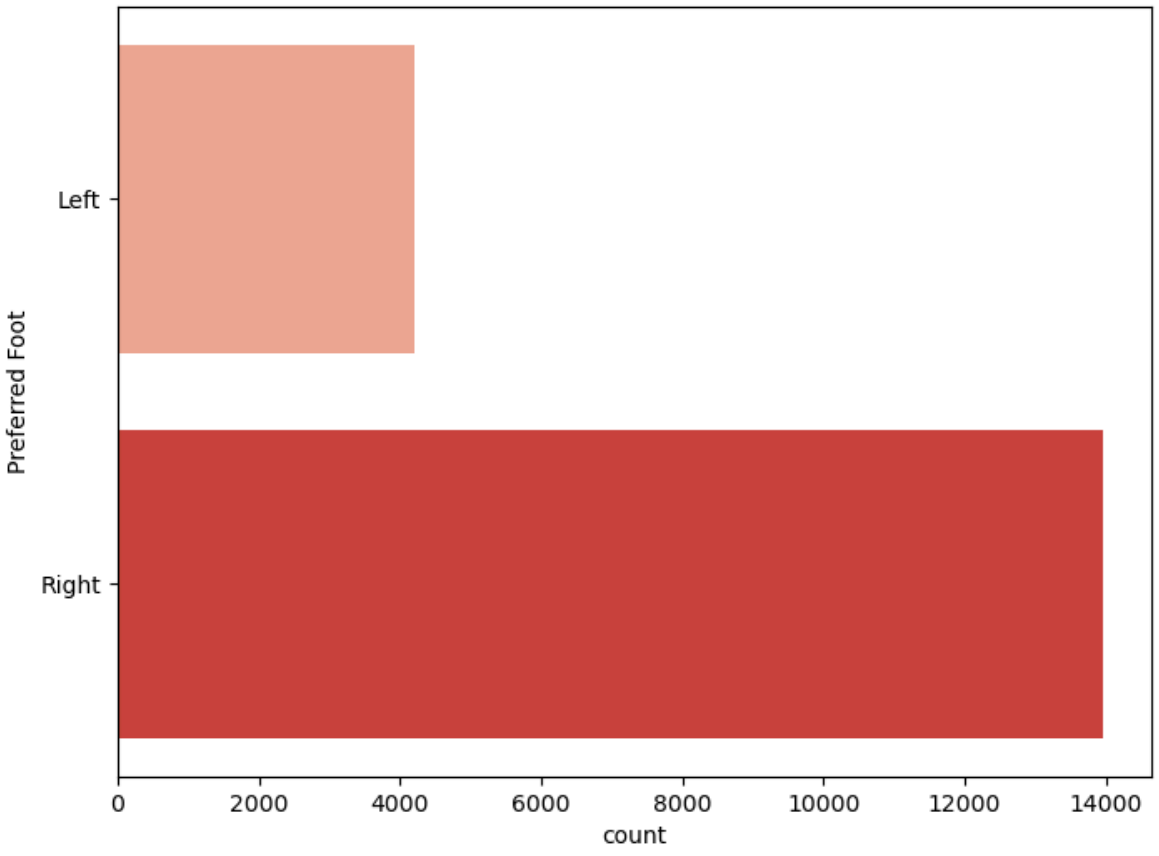


----we can also show value counts for two categorical vars as follows----

In [22]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.countplot(x='Preferred Foot',hue='Real Face', data=fifa19,color='c',palette=
plt.show()
```

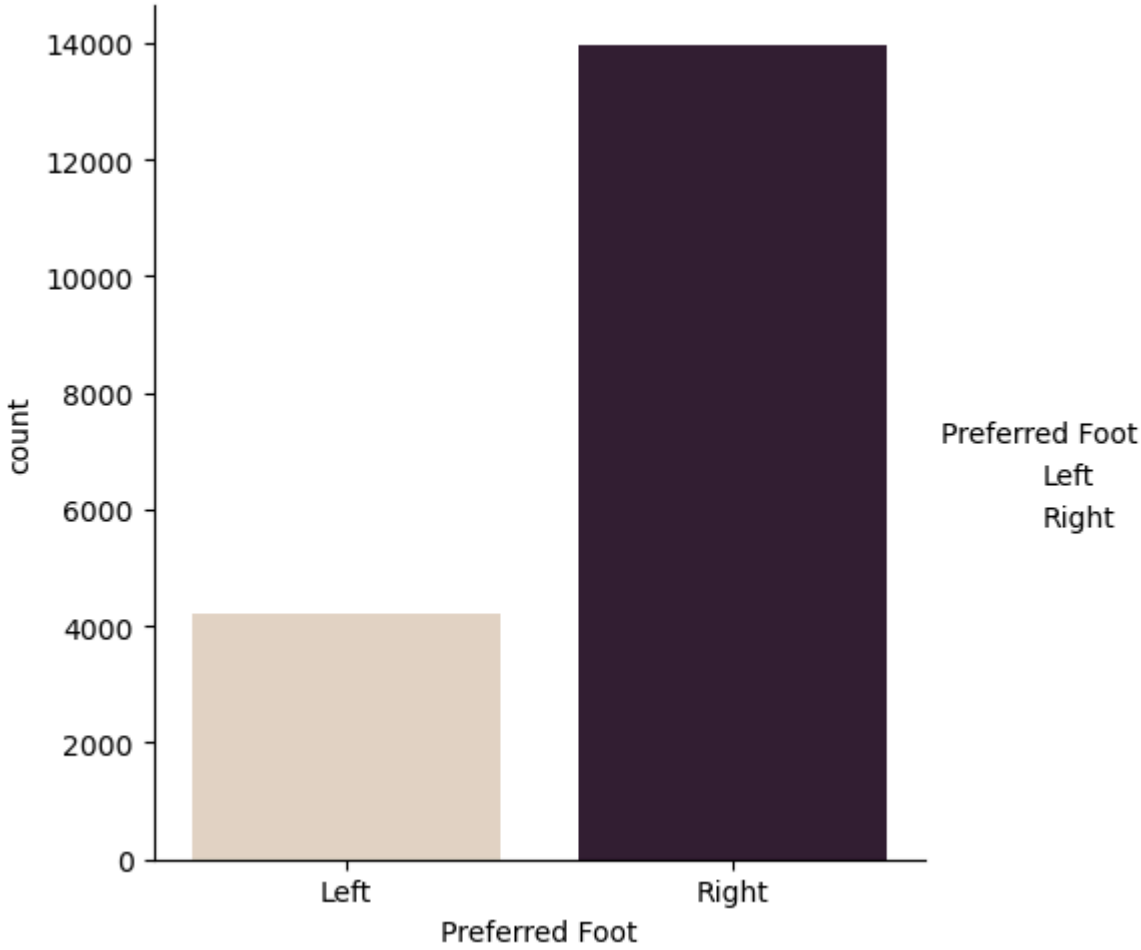


In [23]:
```
#if i put 'y=' instead of 'x=' i can get the graph in vertical format
f,ax=plt.subplots(figsize=(8,6))
sns.countplot(y='Preferred Foot',data=fifa19,color='c',palette='Reds',fill=True,
plt.show()
```



catplot() function

In [24]:
```
g=sns.catplot(x='Preferred Foot',kind='count',palette='ch:.25',data=fifa19)
plt.show()
```



----exploring 'International Reputation' variable now----

In [25]:
```
fifa19['International Reputation'].nunique()
```

Out[25]:
```
5
```

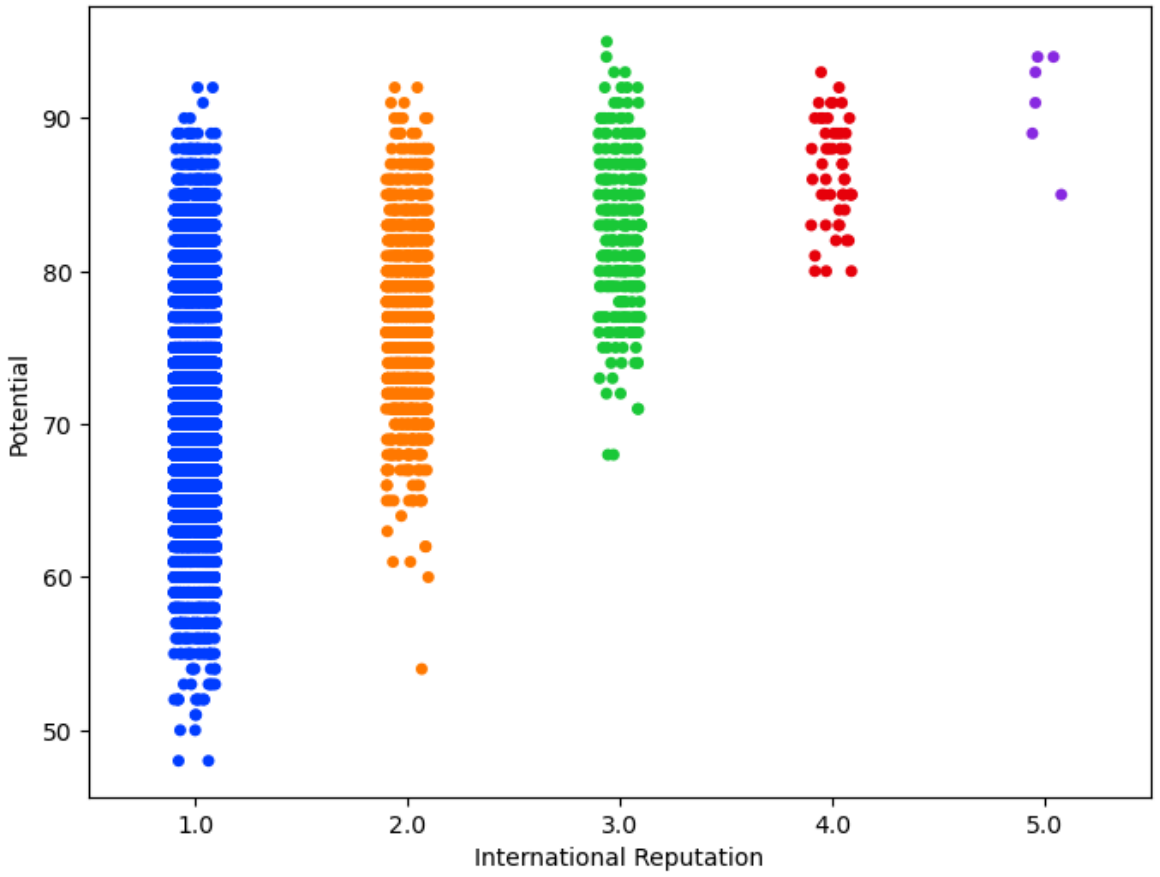
In [26]:
```
fifa19['International Reputation'].value_counts()
```

Out[26]:
```
International Reputation
1.0    16532
2.0     1261
3.0      309
4.0       51
5.0        6
Name: count, dtype: int64
```

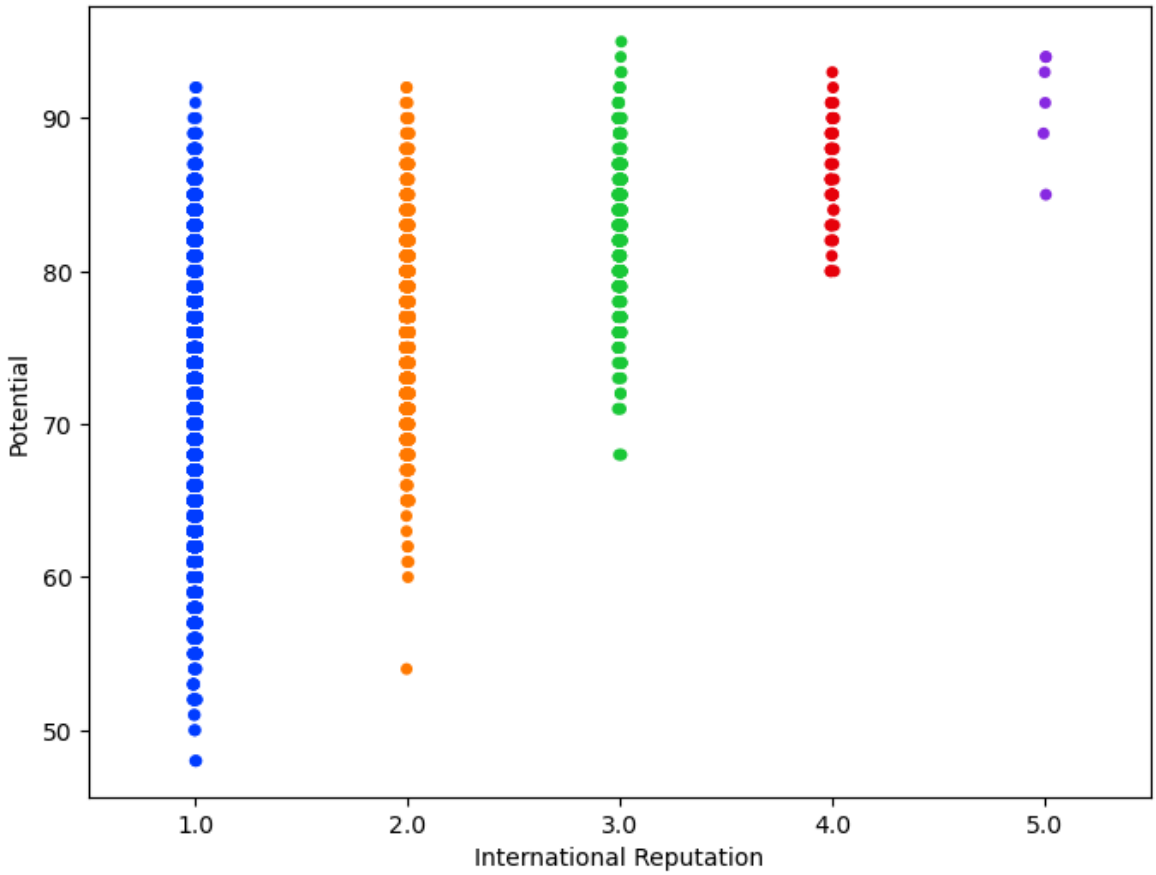
using Stripplot() function

this function draws a scatterplot where one var is categorical

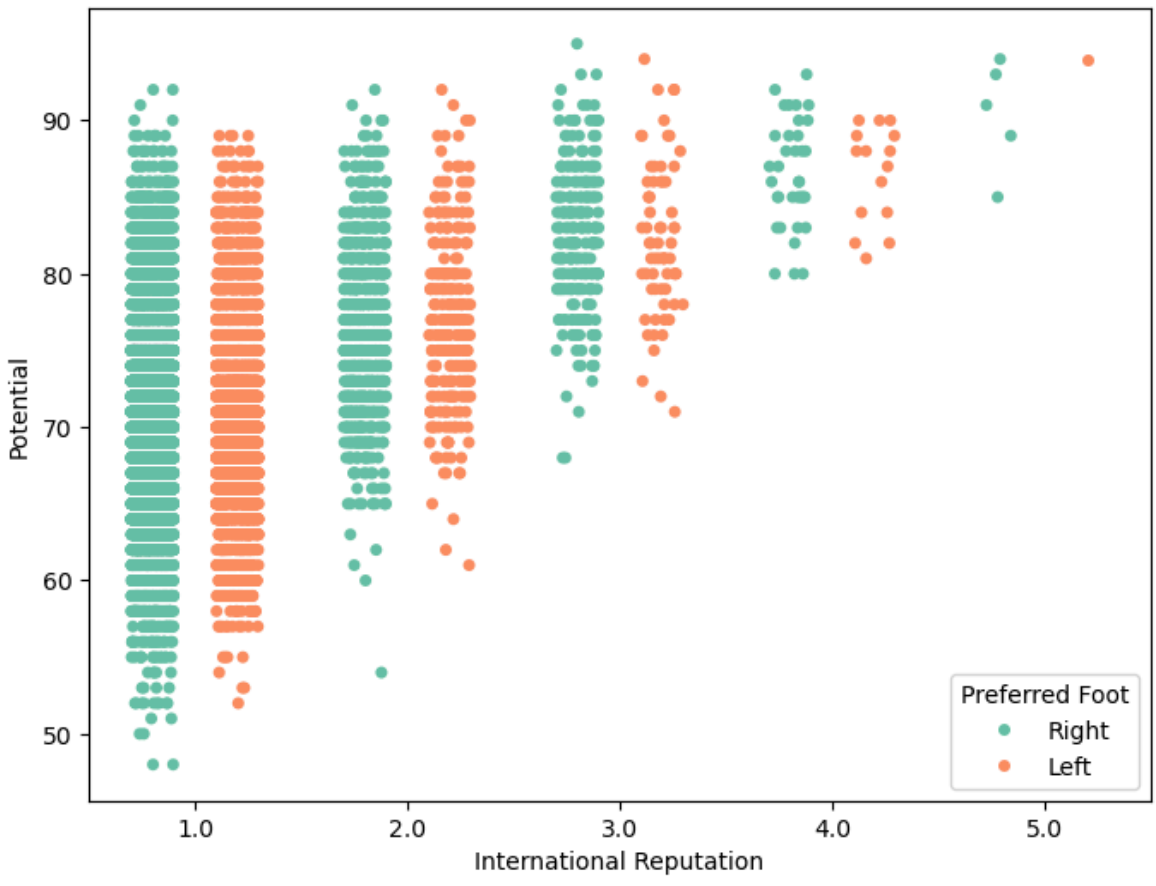
In [27]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.stripplot(x='International Reputation',y='Potential',data=fifa19,palette='br
plt.show()
```



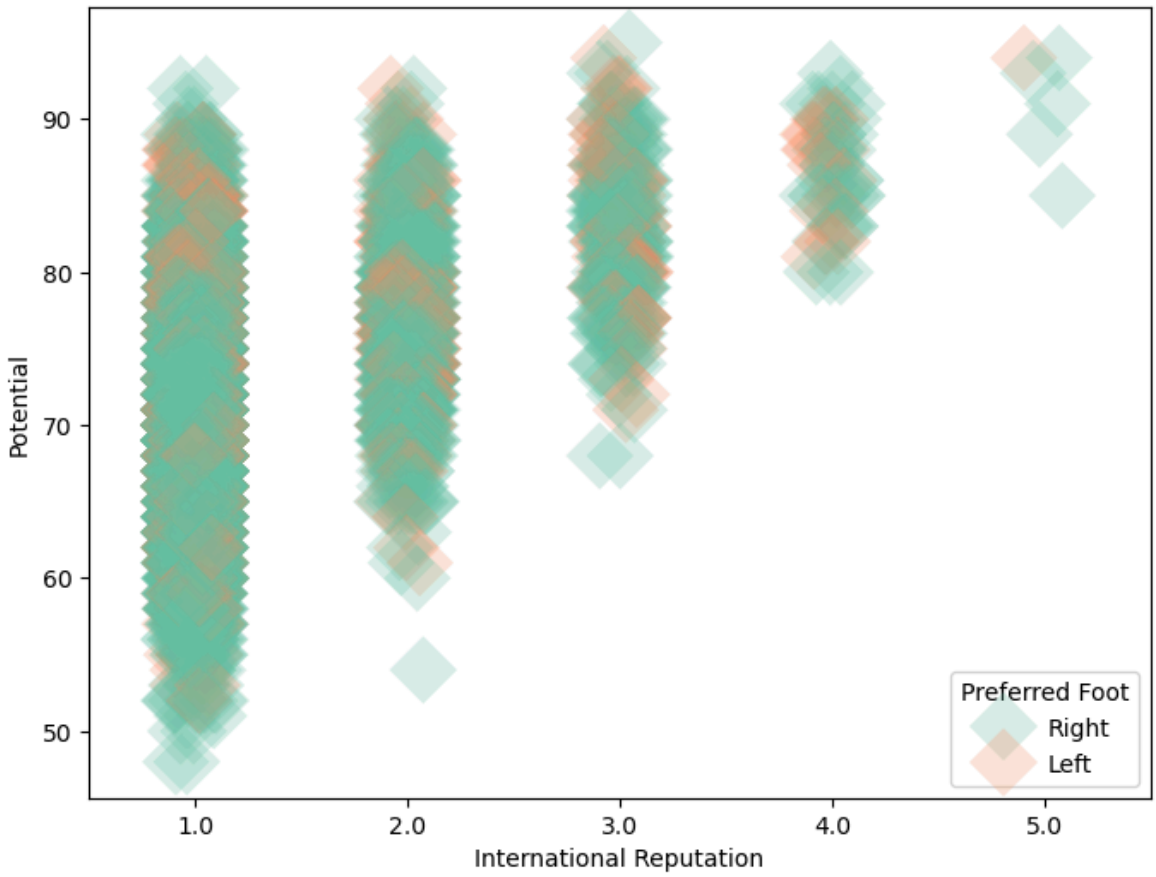
In [28]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", data=fifa19, jitter=0
plt.show()
```



In [29]:
```
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()
```

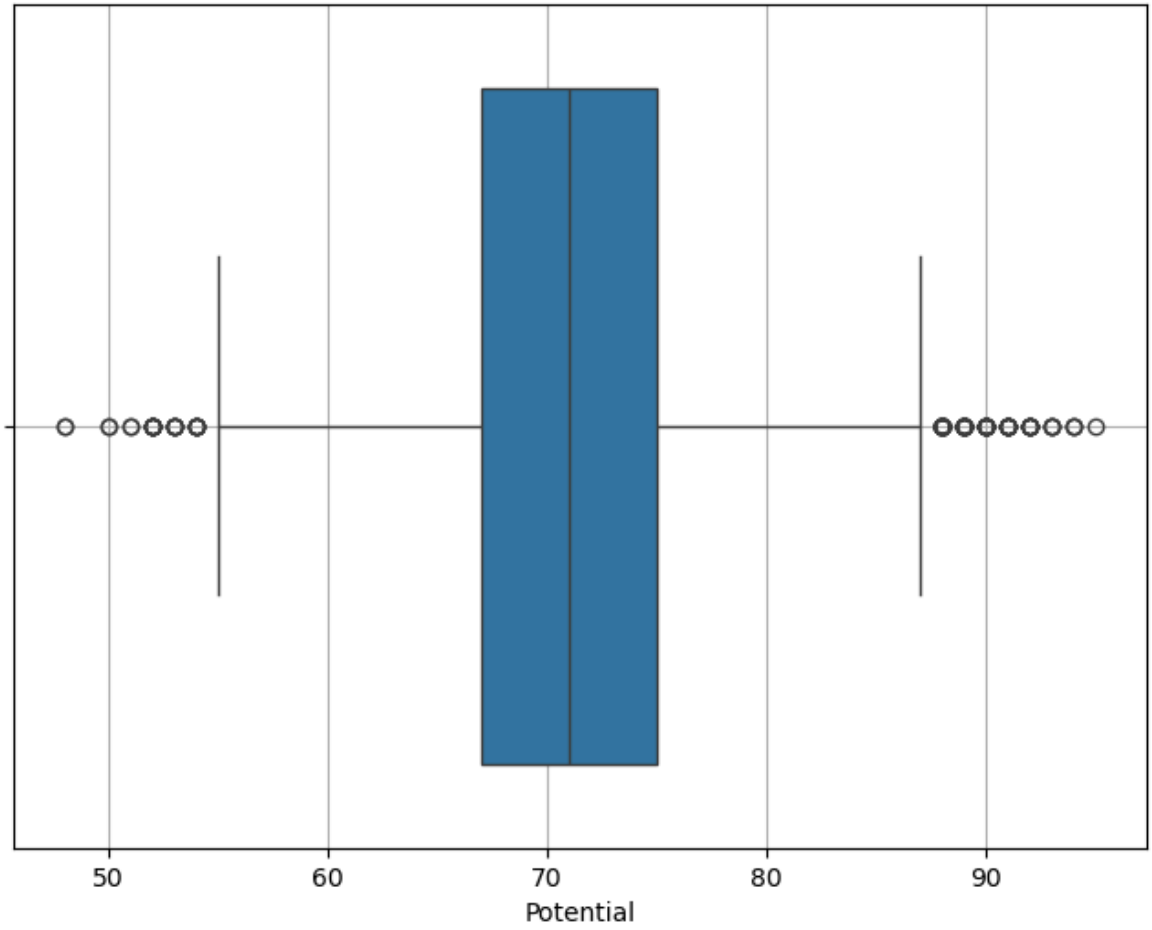


In [30]:
```
#we can draw diff strips with large poinys and diff asthetics as follows
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="International Reputation", y="Potential", hue="Preferred Foot",
                   data=fifa19, palette="Set2", size=20, marker="D",
                   edgecolor="gray", alpha=.25)
plt.show()
```

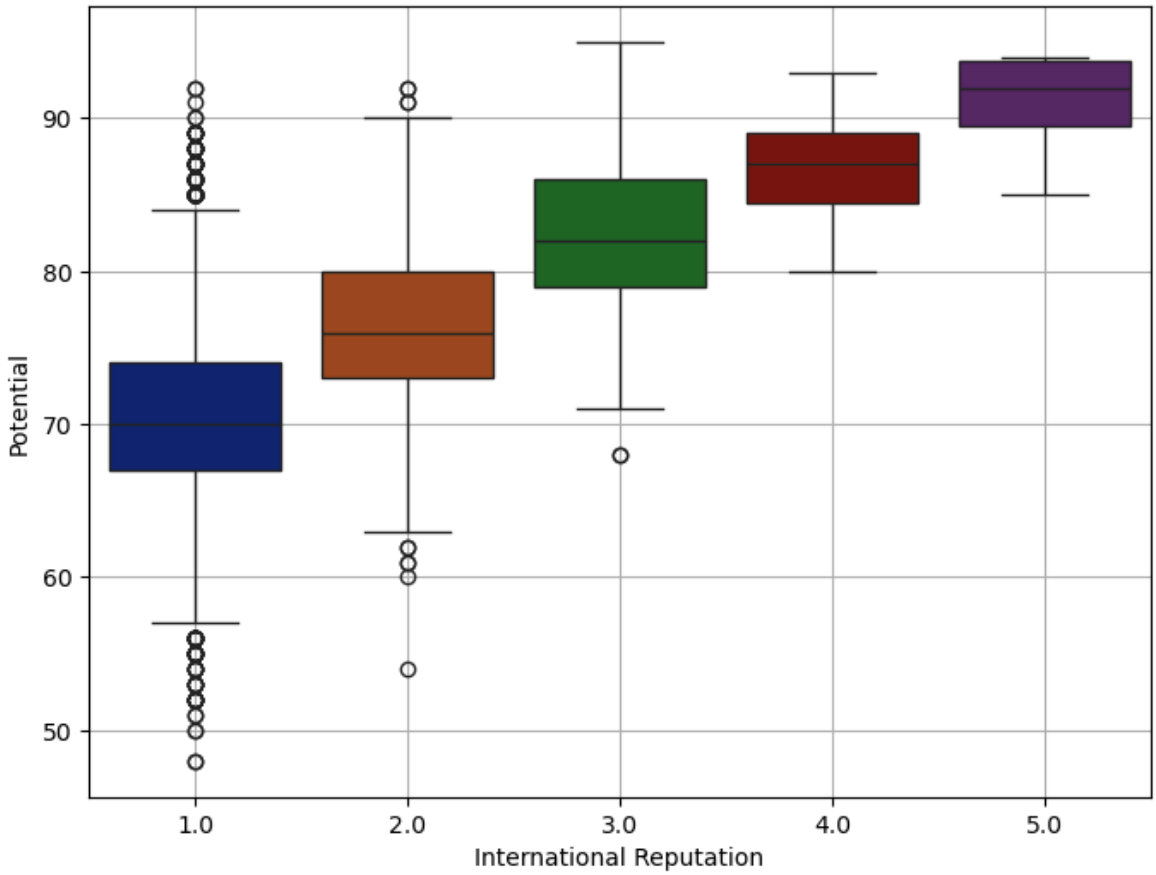


boxplot function

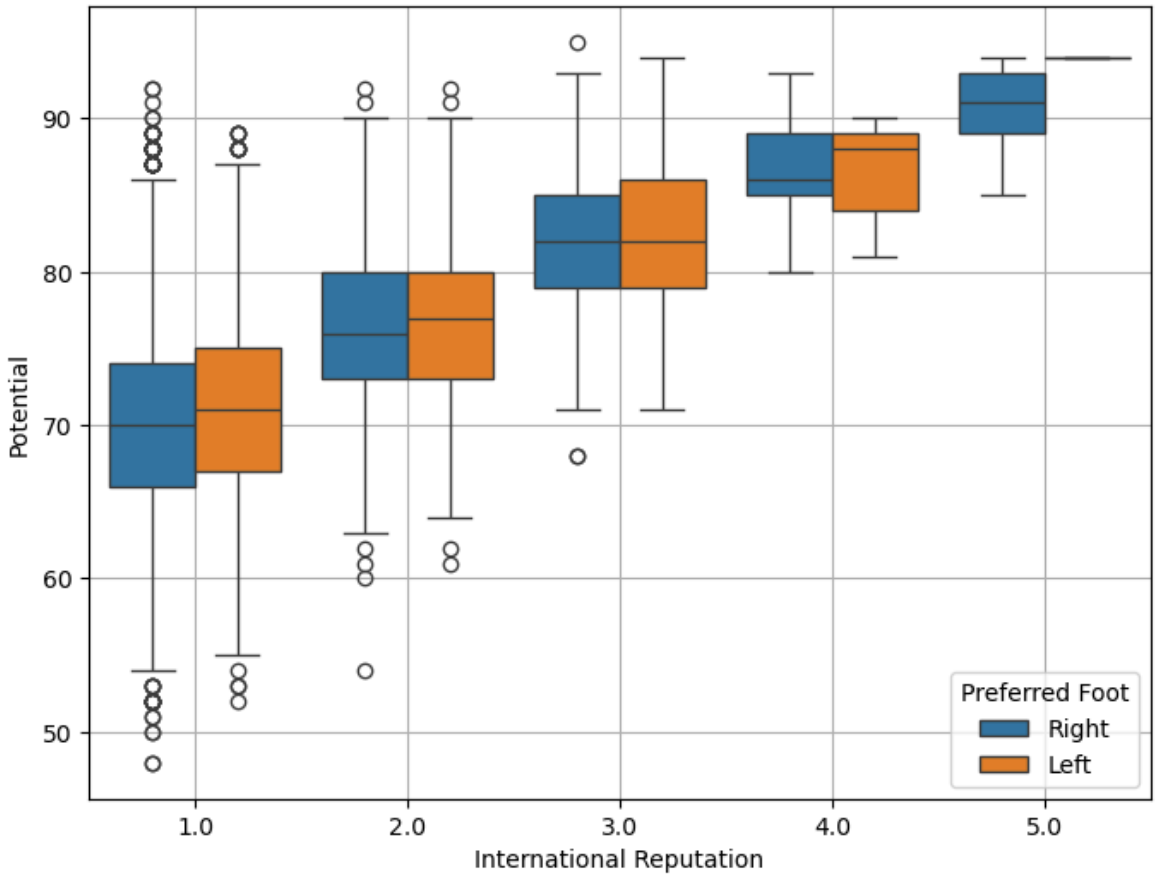
In [31]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(x=fifa19['Potential'])
plt.grid()
plt.show()
```



In [32]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(data=fifa19,x='International Reputation',y='Potential',palette='dark
plt.grid()
plt.show()
```

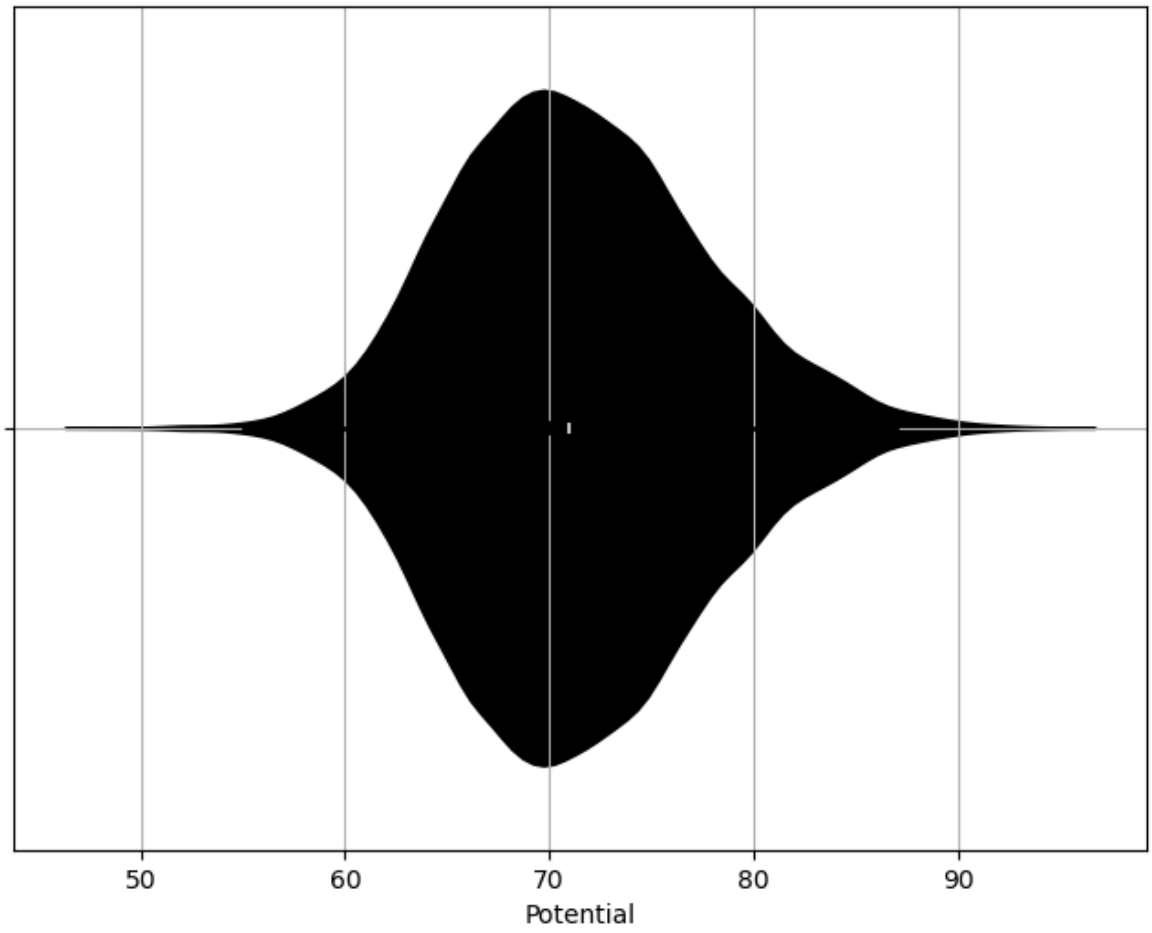


In [33]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(data=fifa19,x='International Reputation',y='Potential',hue='Preferre
plt.grid()
plt.show()
```

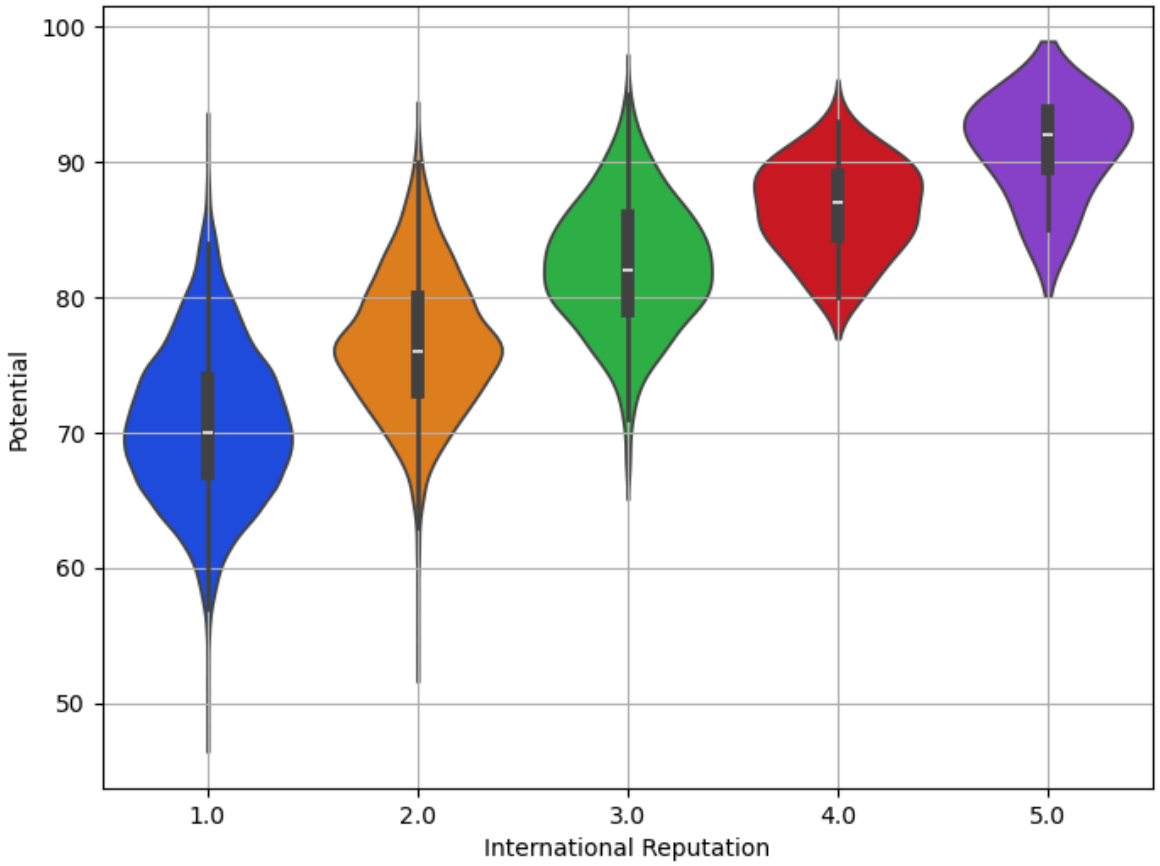


using violinplot() function

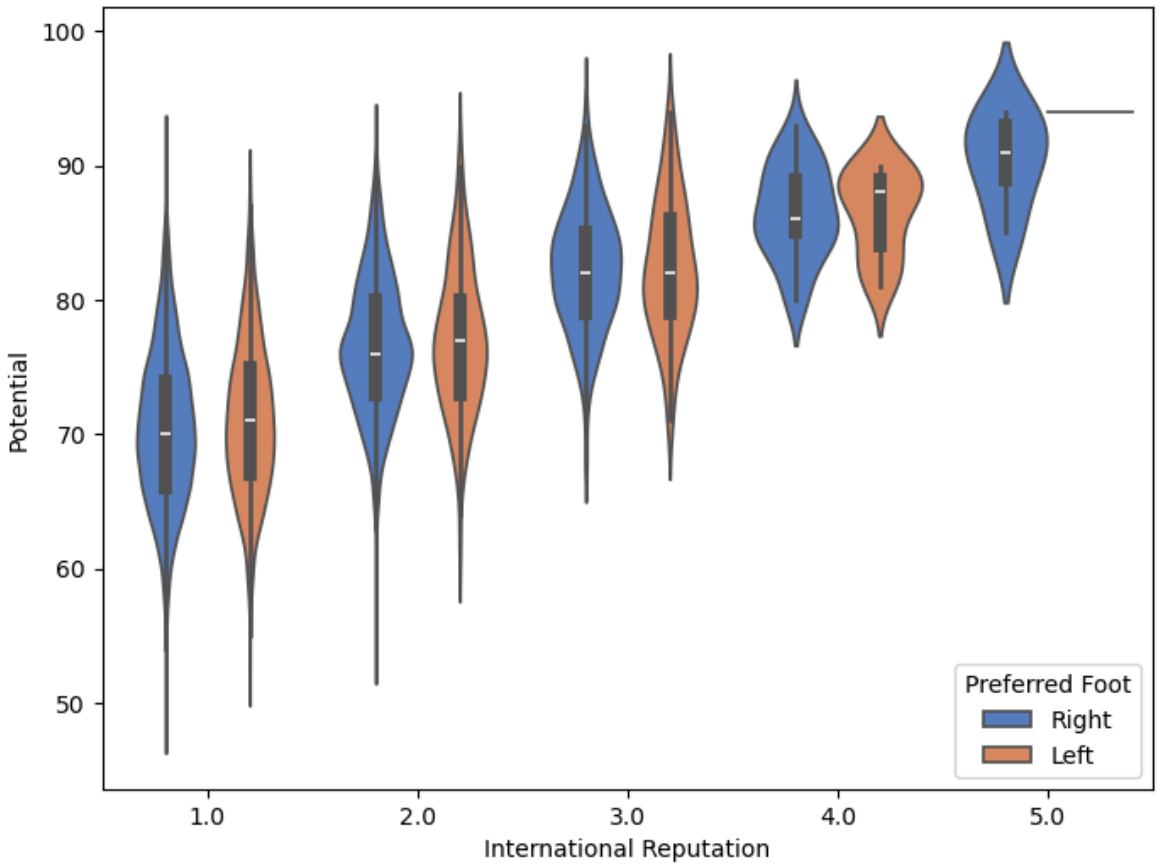
In [34]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.violinplot(x=fifa19['Potential'],color='black')
plt.grid()
plt.show()
```



In [35]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.violinplot(data=fifa19,x='International Reputation',y='Potential',palette='b
plt.grid()
plt.show()
```

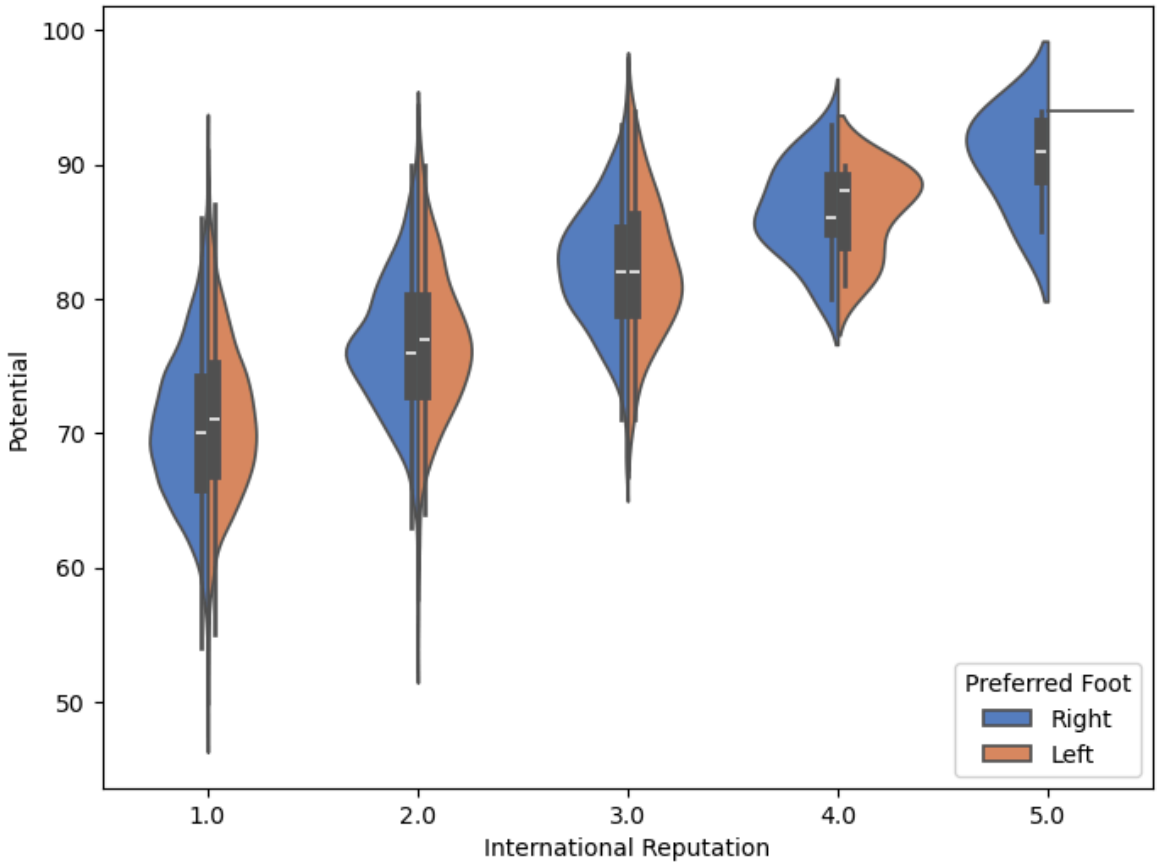


In [36]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.violinplot(x="International Reputation", y="Potential", hue="Preferred Foot"
plt.show()
```



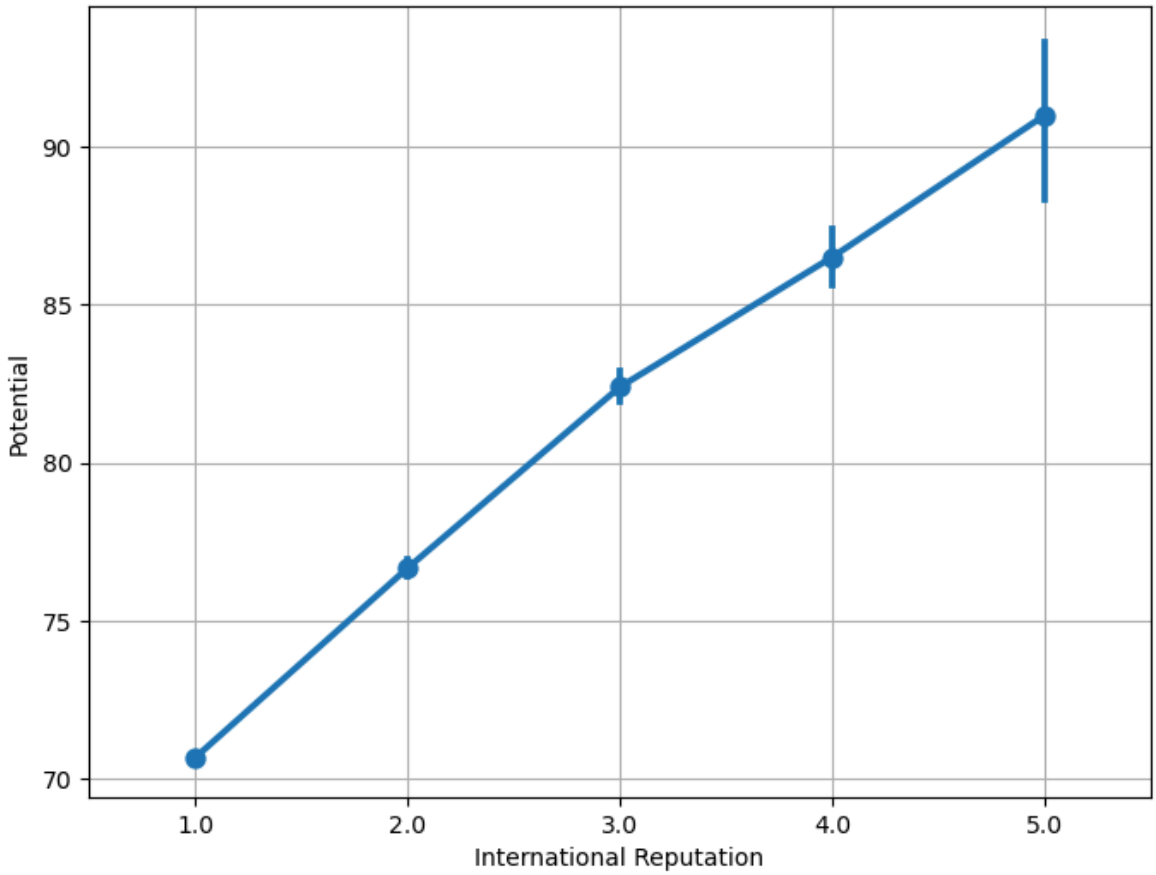
In [37]:
```
#we can aslo draw split violine to compare the hue var as follows using split:
f, ax = plt.subplots(figsize=(8, 6))
```

```
sns.violinplot(x="International Reputation", y="Potential", hue="Preferred Foot"
plt.show()
```

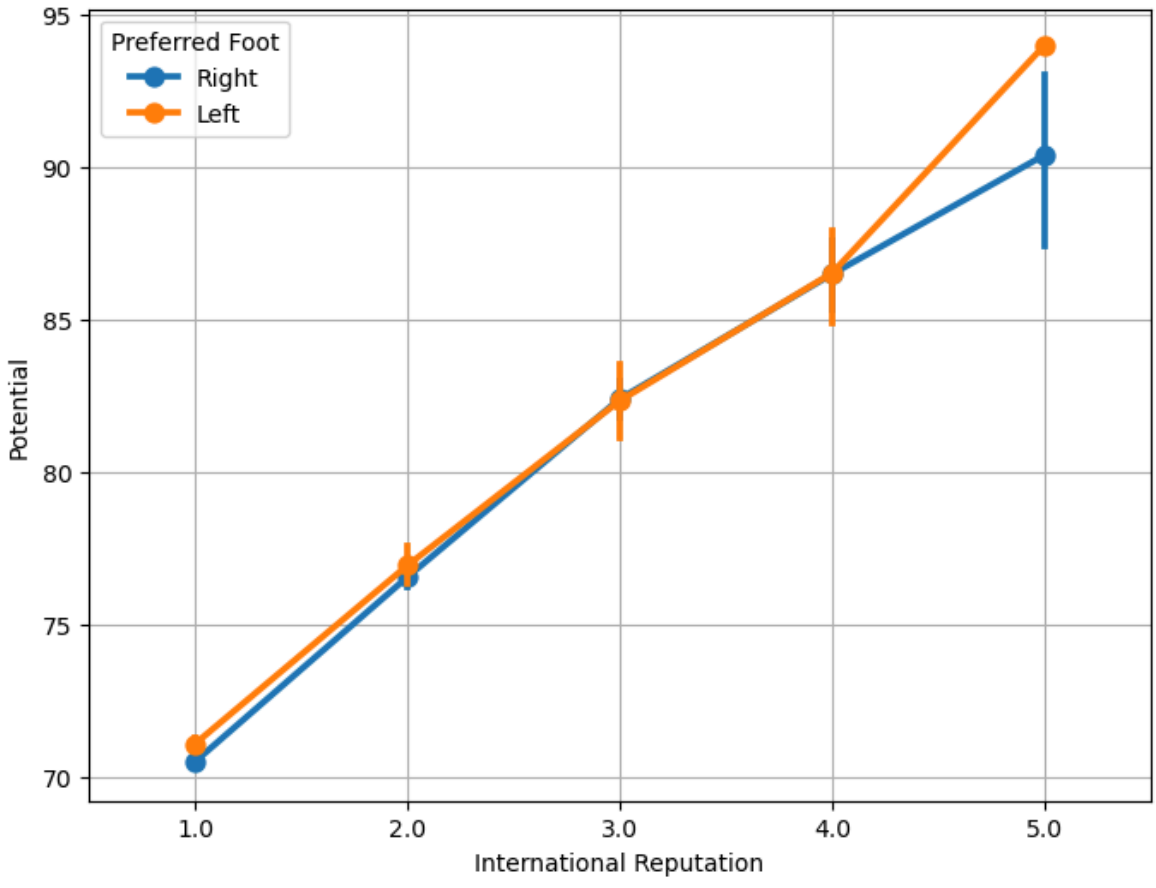


pointplot() function

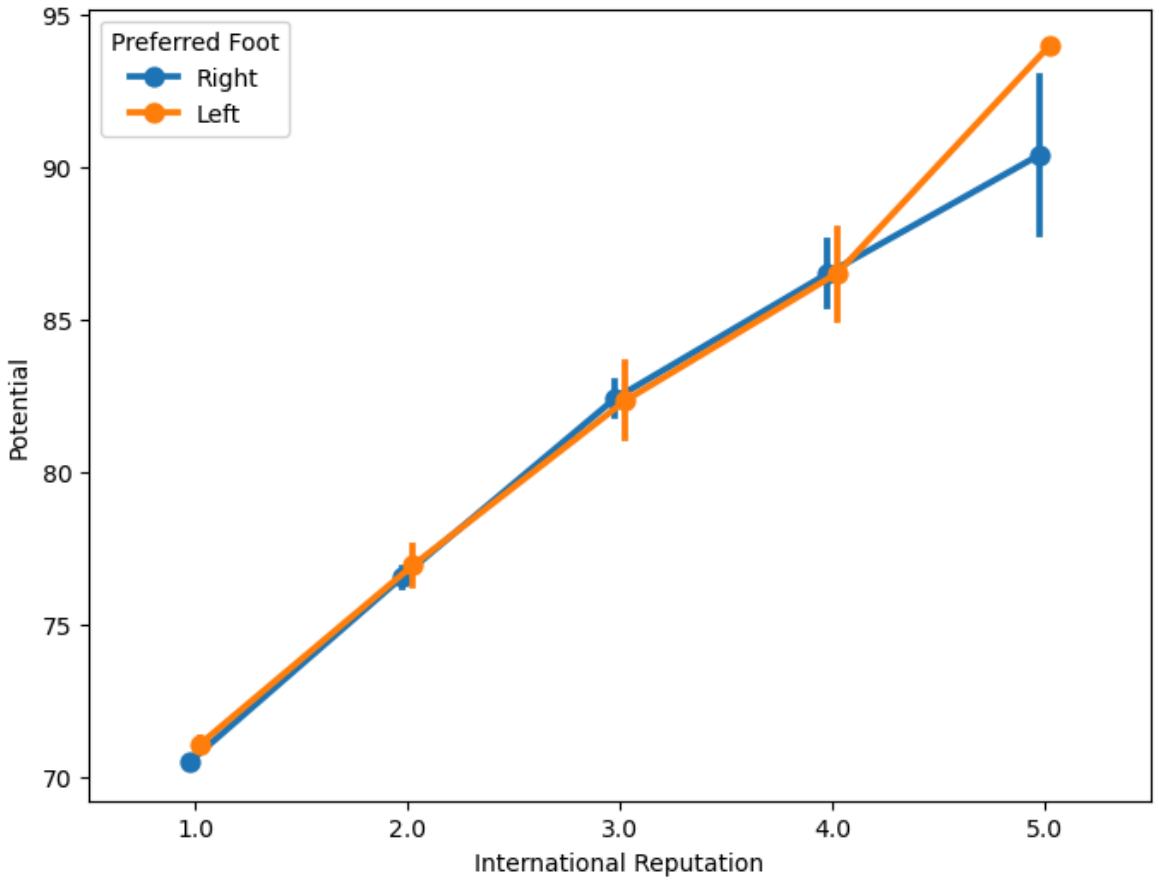
In [38]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", data=fifa19)
plt.grid()
plt.show()
```



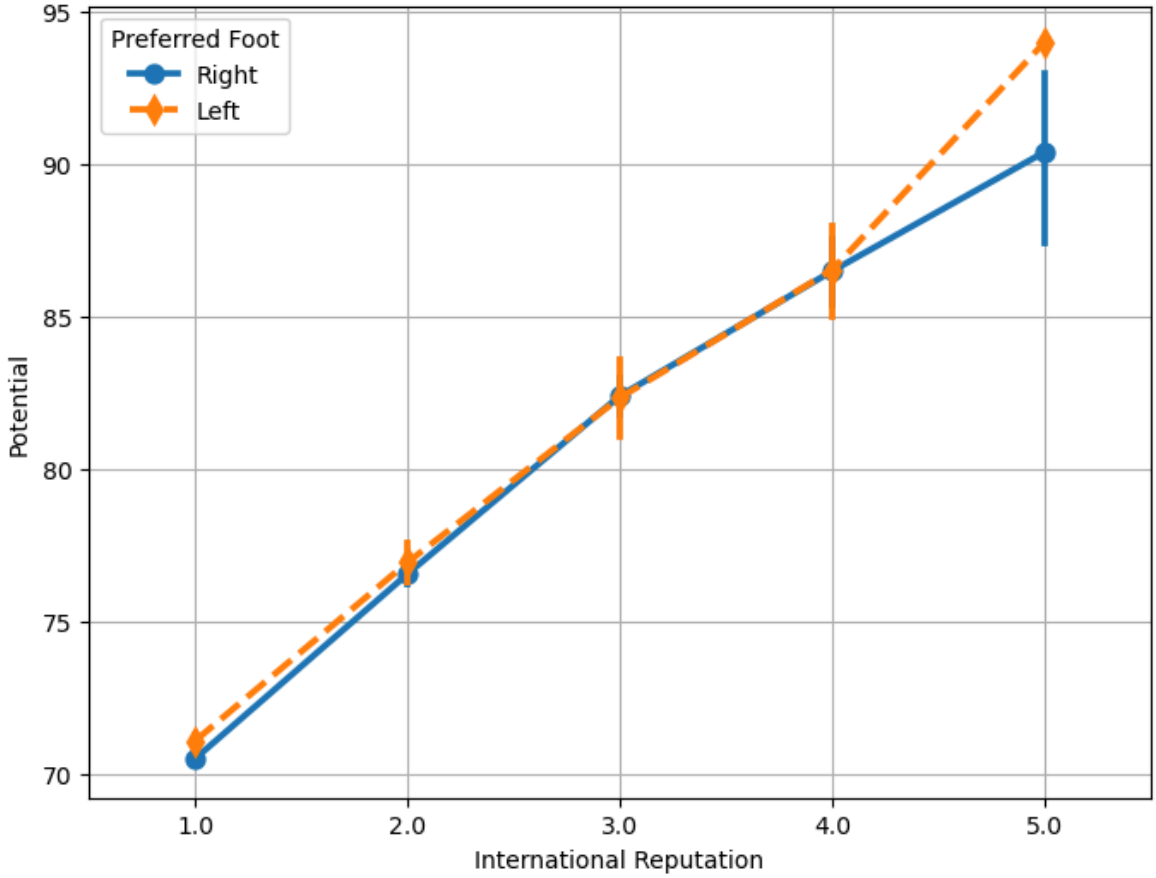
In [39]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", hue="Preferred Foot",
plt.grid()
plt.show()
```



In [40]:
```
#we can separate the points of diff hue lvls along the catgorical axis as follow
f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", hue="Preferred Foot",
plt.show()
```

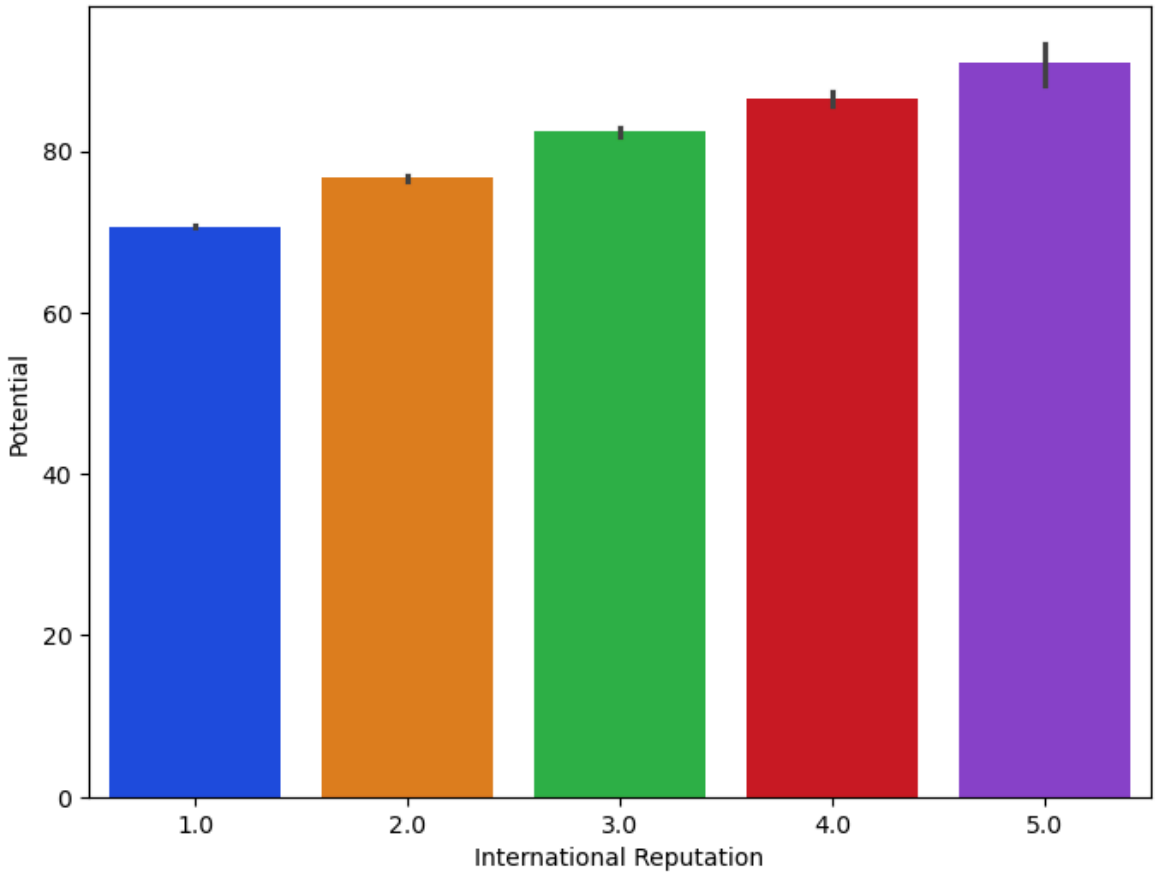


In [41]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.pointplot(x="International Reputation", y="Potential", hue="Preferred Foot",
plt.grid()
plt.show()
```

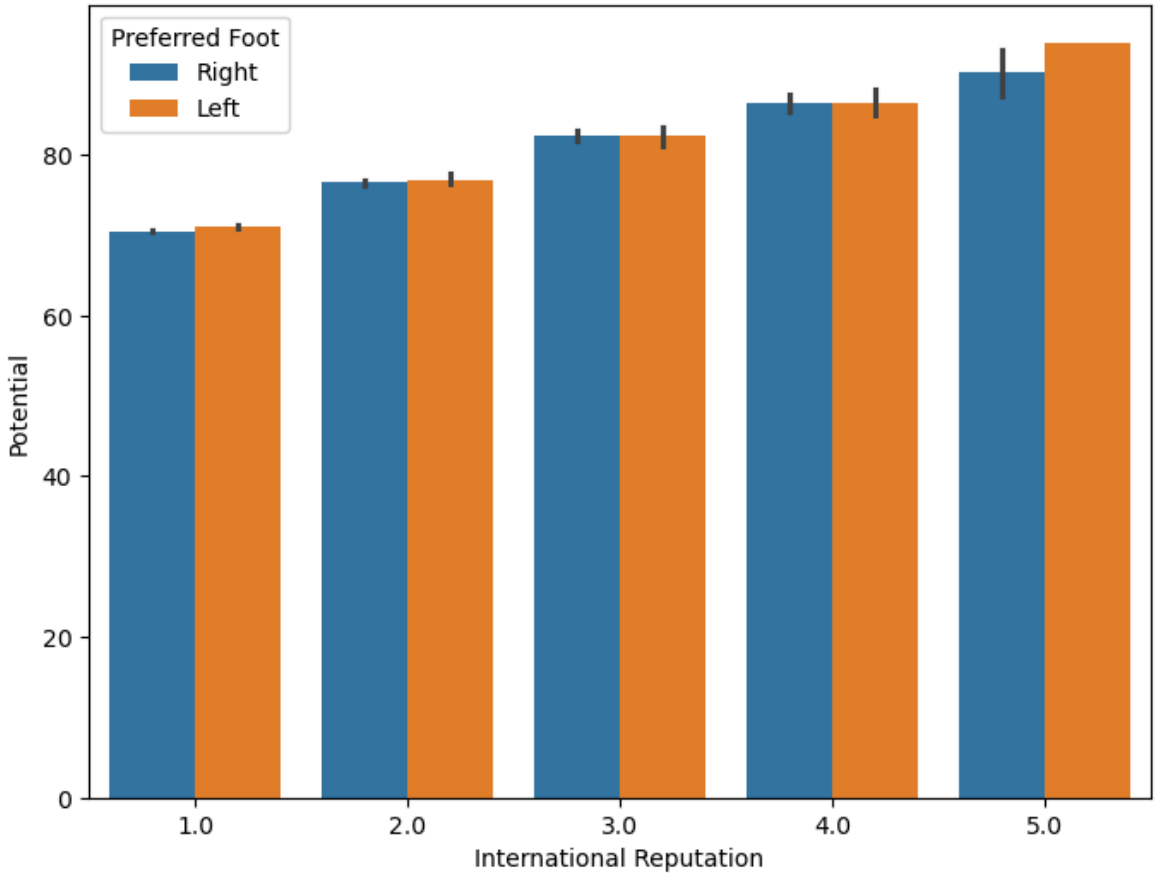


Barplot() function

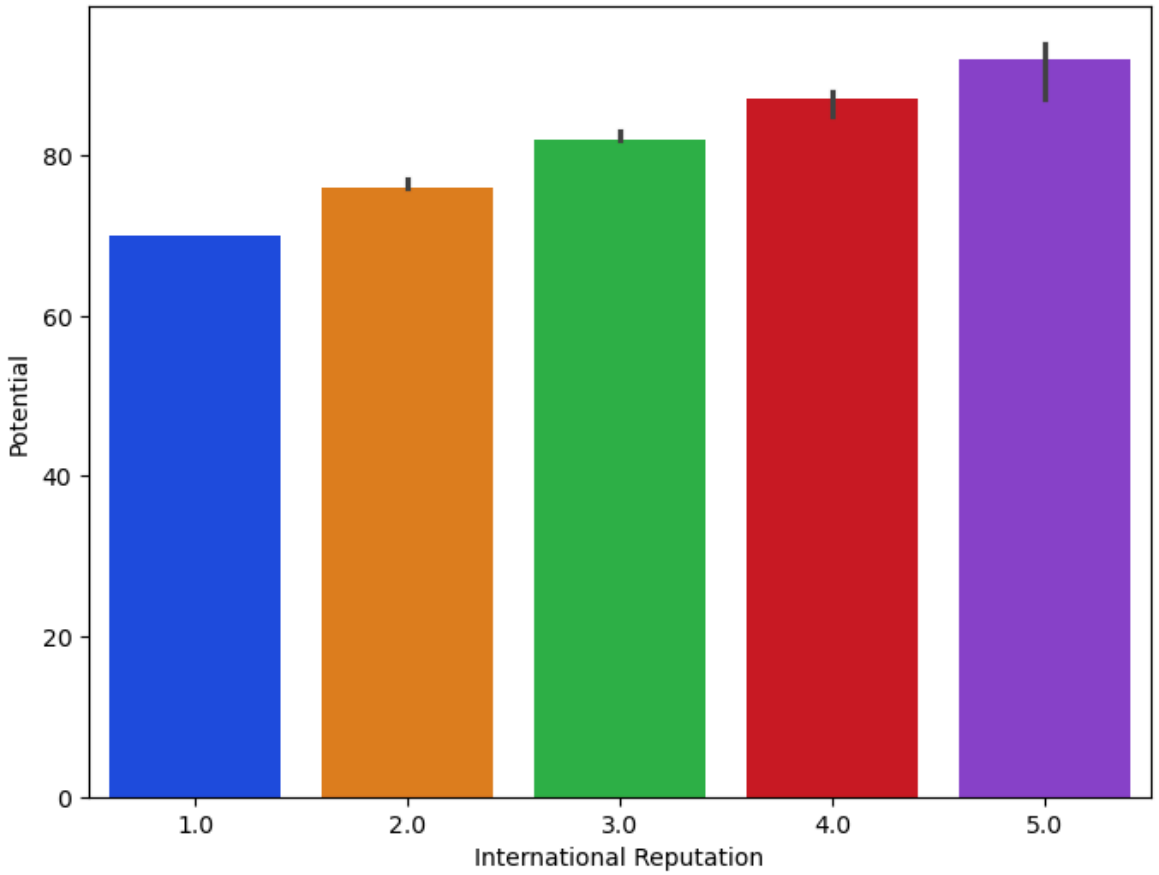
In [42]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19,palette='br
plt.show()
```



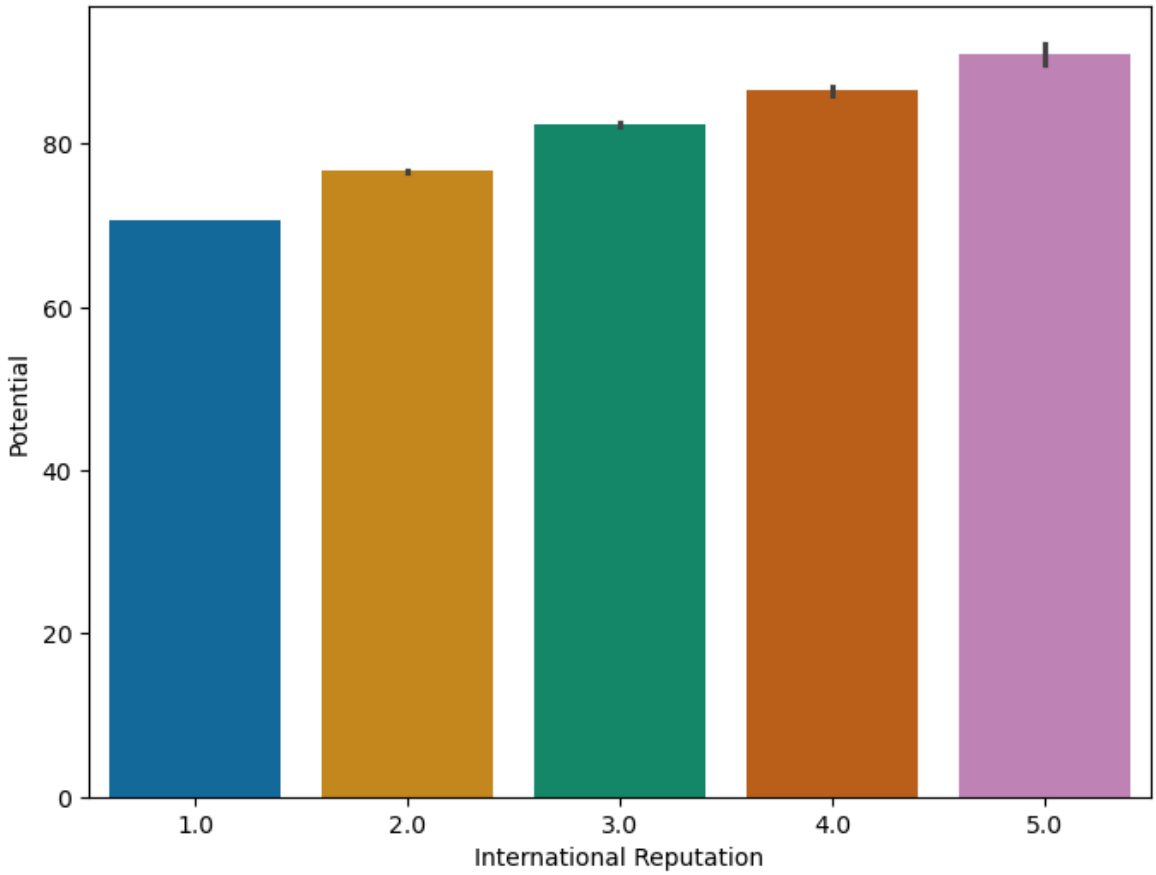
In [43]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", hue='Preferred Foot', d
plt.show()
```



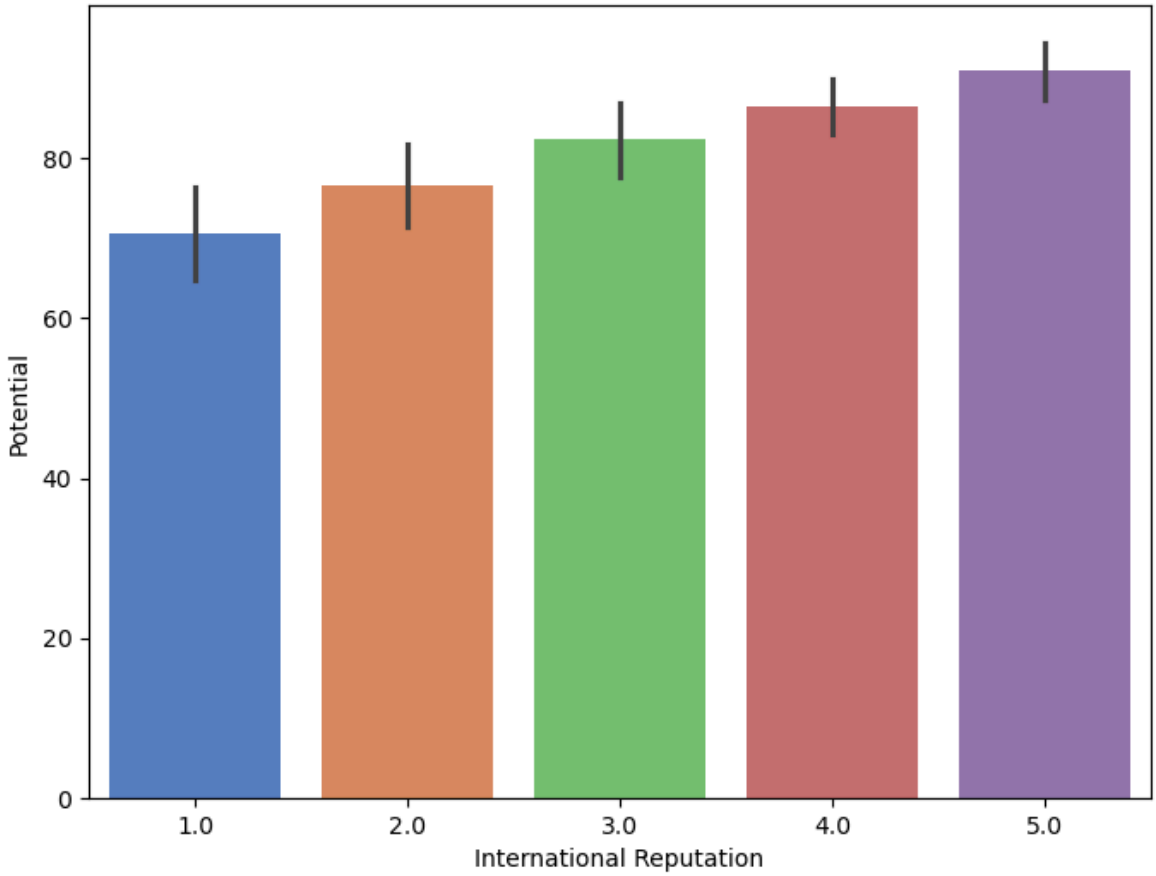
In [44]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19,palette='br
plt.show()
```



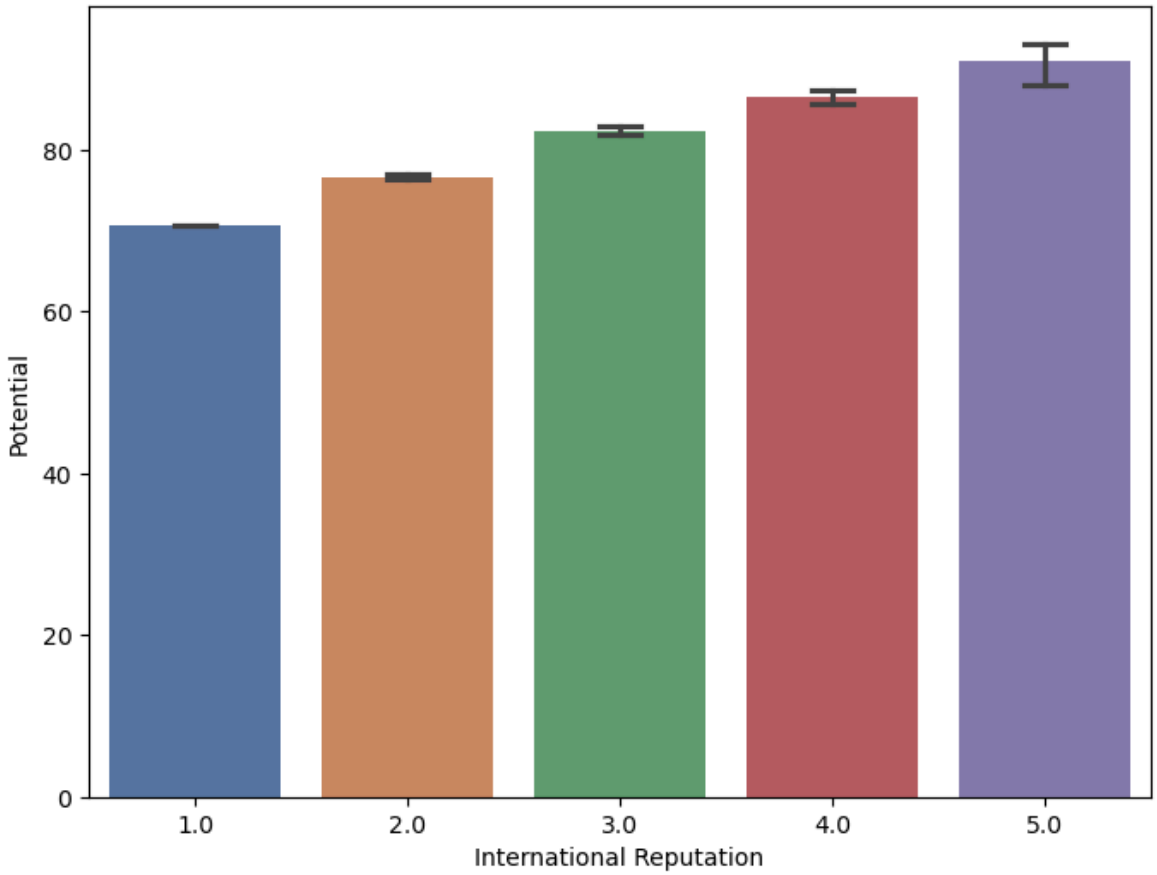
In [45]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19,ci=68,palet
plt.show()
```



In [46]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19, ci="sd",pa
plt.show()
```



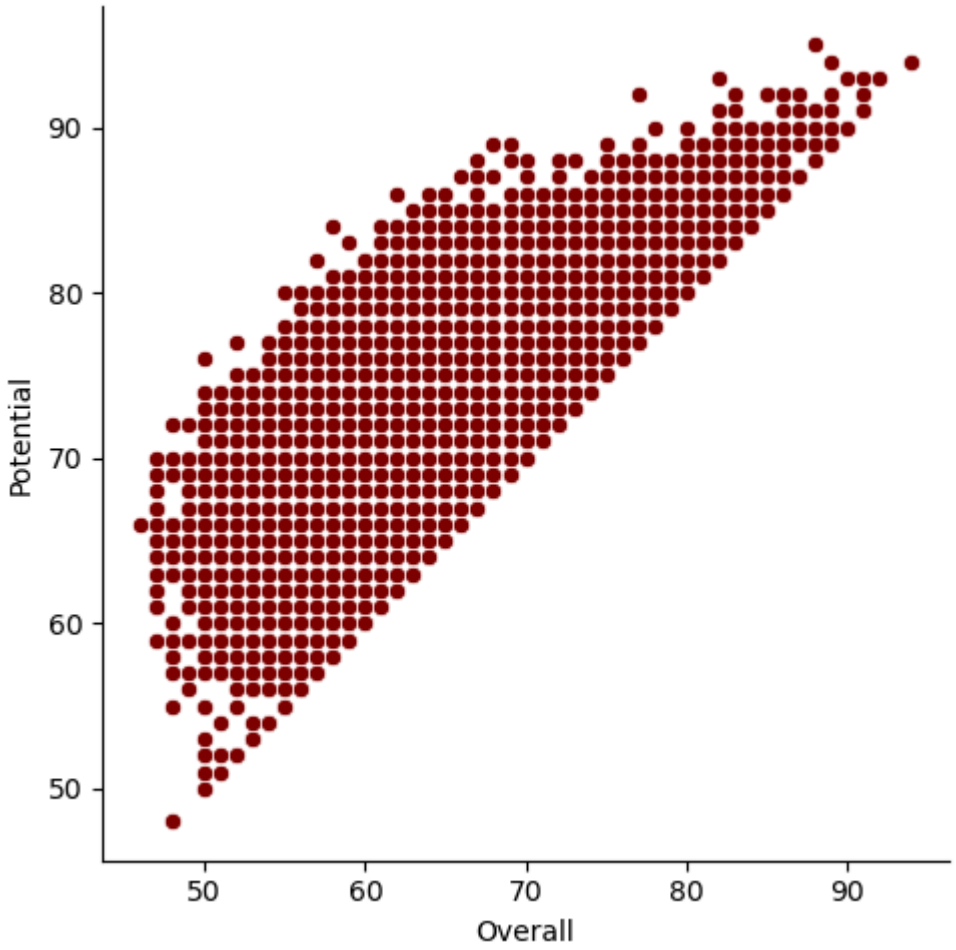
In [47]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.barplot(x="International Reputation", y="Potential", data=fifa19, palette='d
plt.show()
```



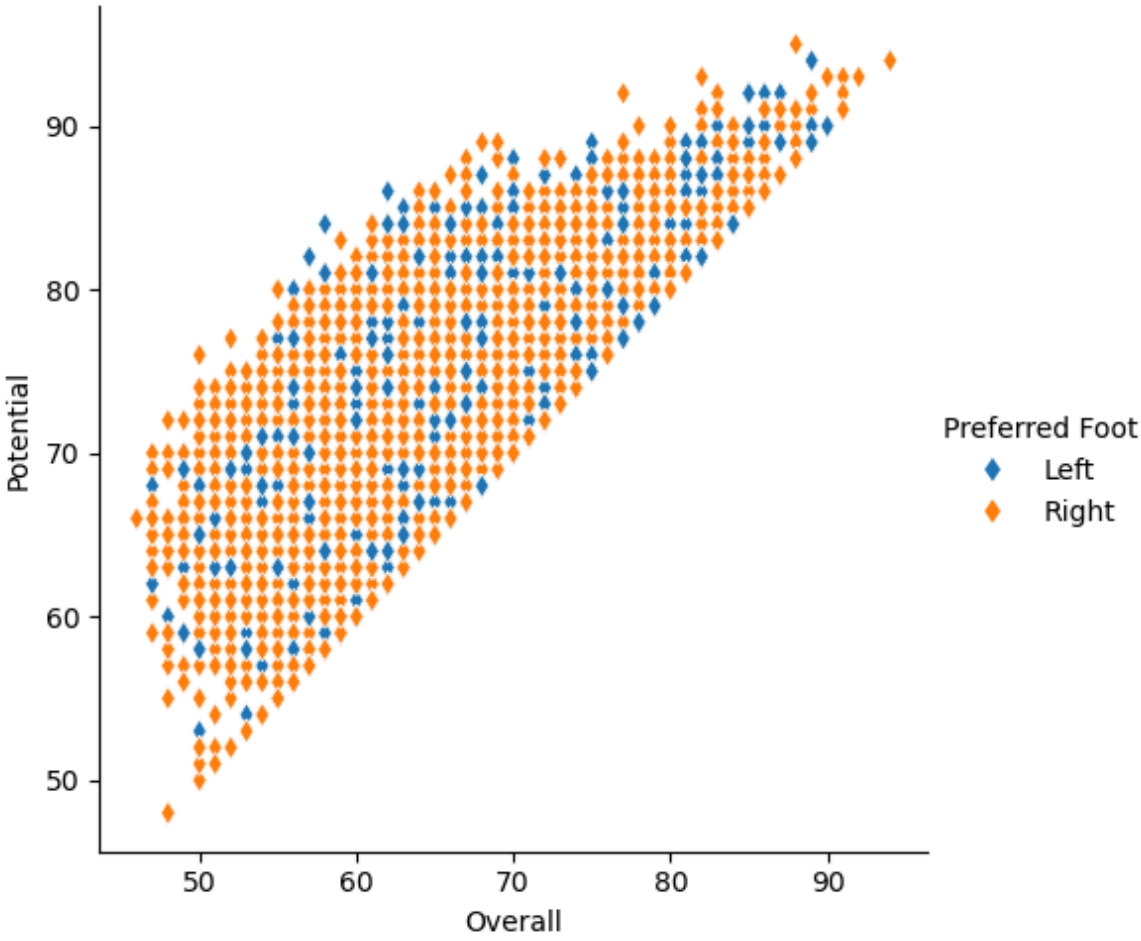
relplot() function

Seaborn `relplot()` function - Seaborn `relplot()` function helps us to draw figure-level interface for drawing relational plots onto a FacetGrid. - This function provides access to several different axes-level functions that show the relationship between two variables with semantic mappings of subsets. - The `kind` parameter selects the underlying axes-level function to use- - scatterplot() (with kind="scatter"; the default) - lineplot() (with kind="line") ---we can plit a scatterplot with hight, weight variables using relplot() as follows---

In [48]:
```
g= sns.relplot(x='Overall',y='Potential',data=fifa19,color='Maroon')
plt.show()
```

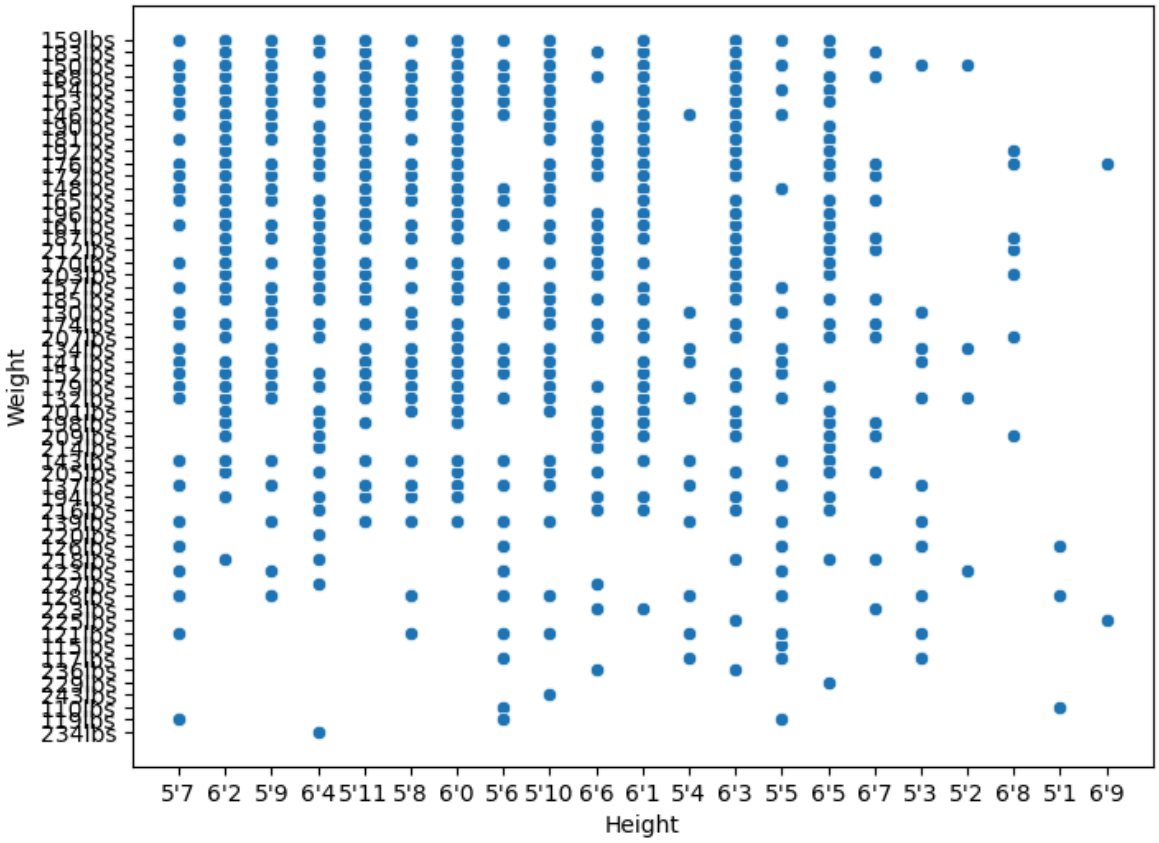


In [49]:
```
g= sns.relplot(x='Overall',y='Potential',hue='Preferred Foot',marker='d',data=fi
plt.show()
```

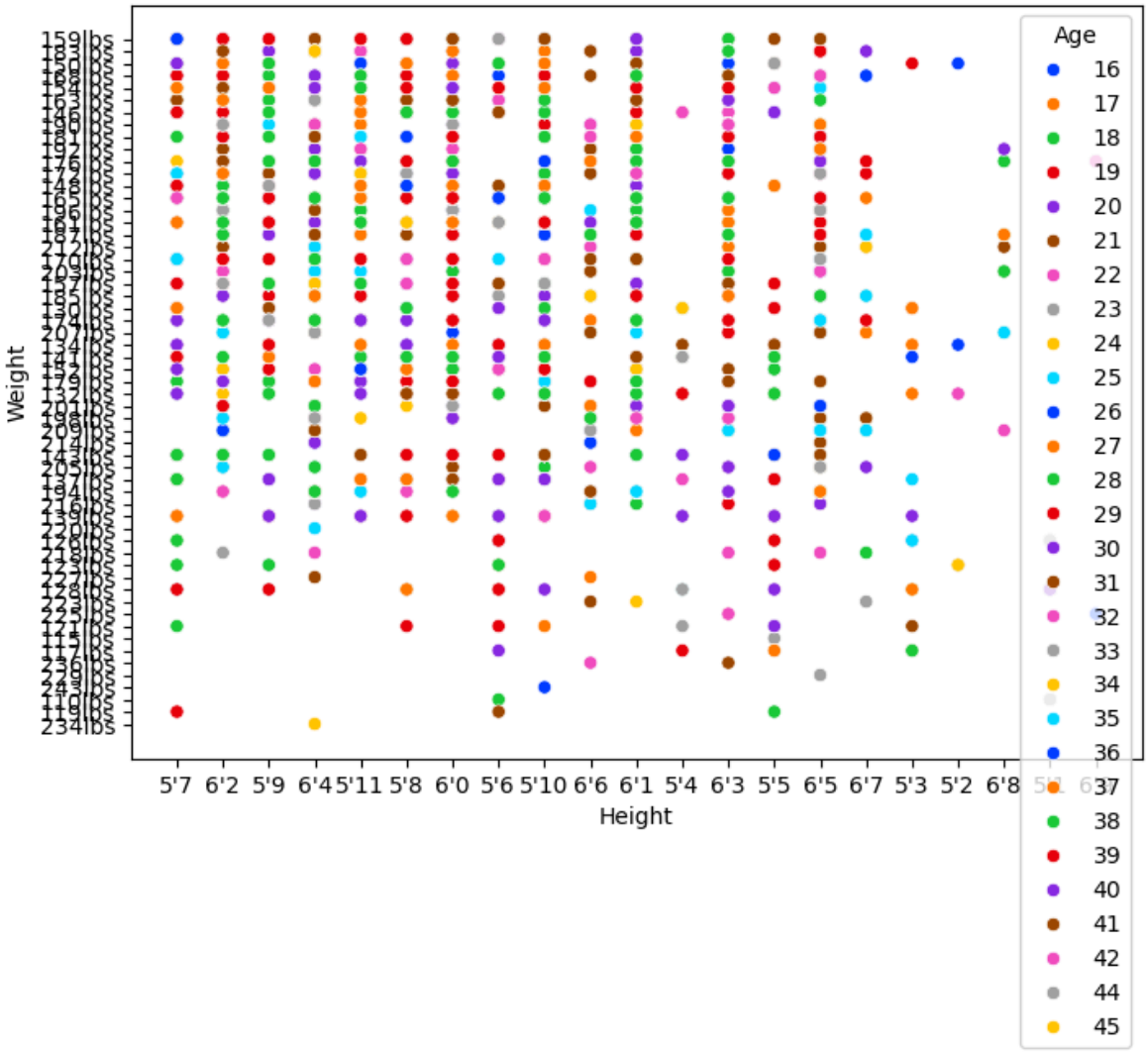


scatter plot

In [50]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.scatterplot(x="Height", y="Weight", data=fifa19)
plt.show()
```

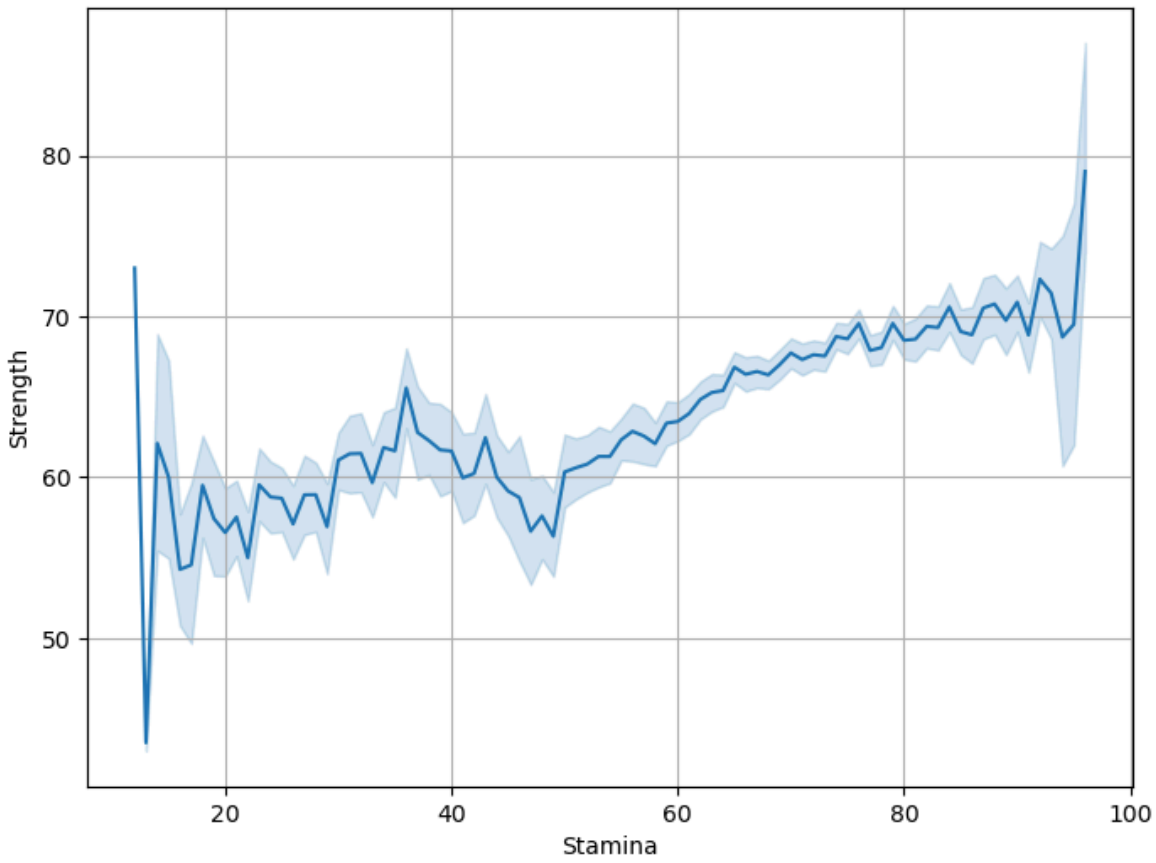


In [51]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.scatterplot(x="Height", y="Weight", hue='Age', data=fifa19,palette='bright')
plt.show()
```



lineplot

In [52]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.lineplot(x='Stamina',y='Strength',data=fifa19)
plt.grid()
plt.show()
```

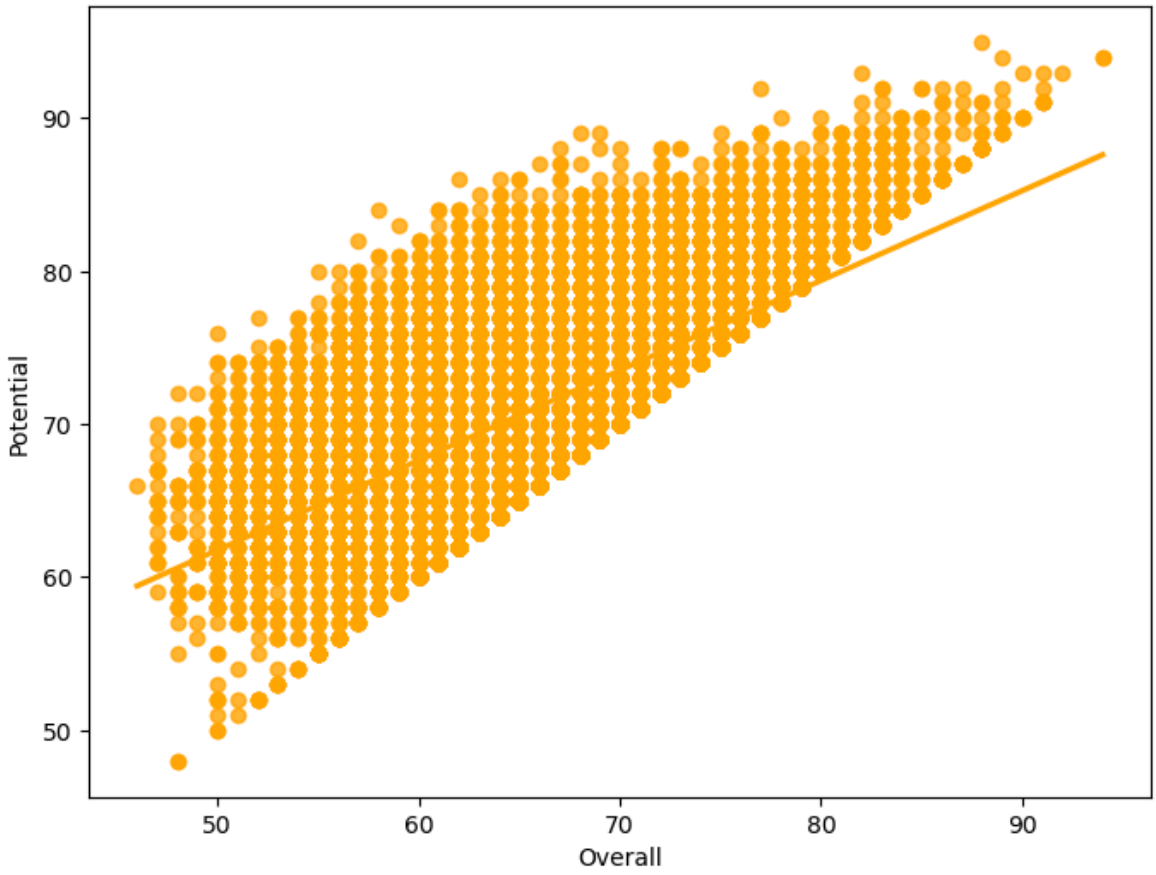


In [53]:
```
#f,ax=plt.subplots(figsize=(8,6))
#sns.lineplot(x='Stamina',y='Strength',hue='Age',data=fifa19)
#plt.grid()
#plt.show()
```

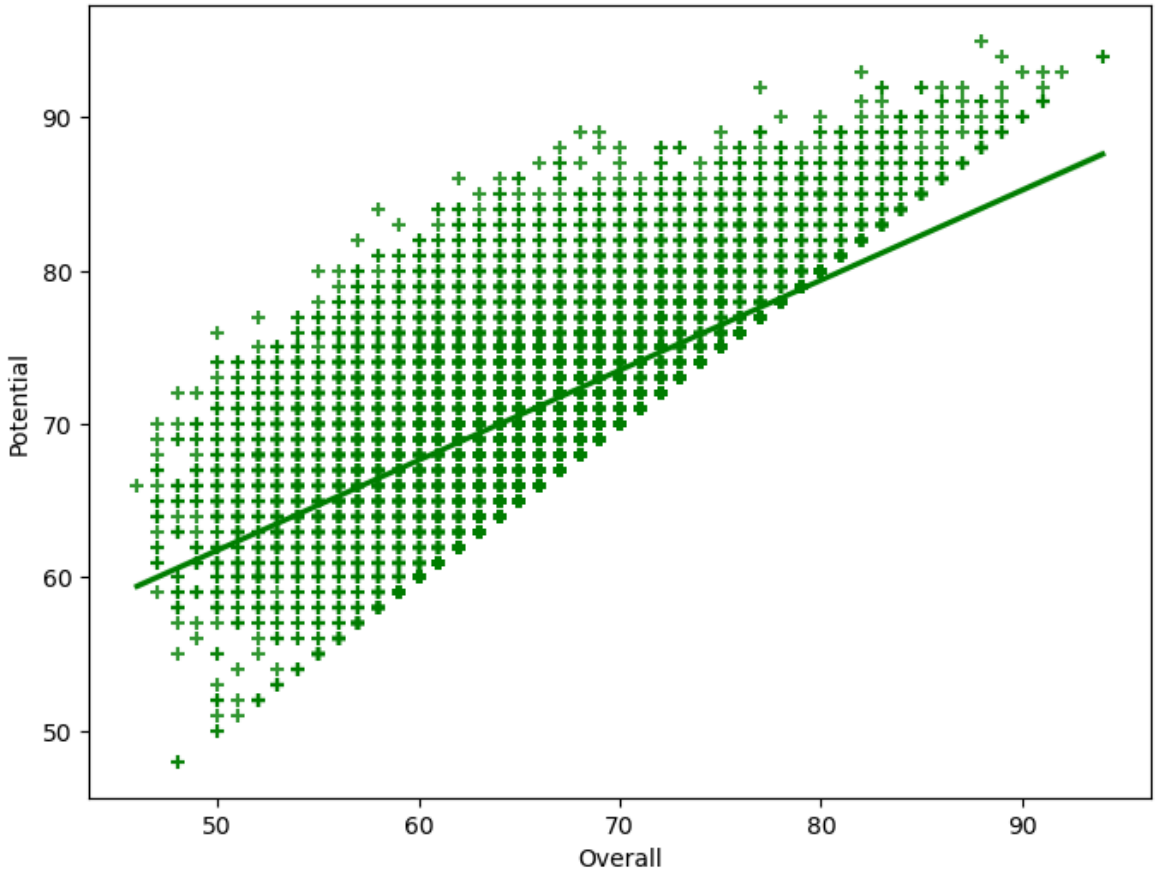
Regression plot

we can plot or model between Overall and Potential var with regplot() as follows-

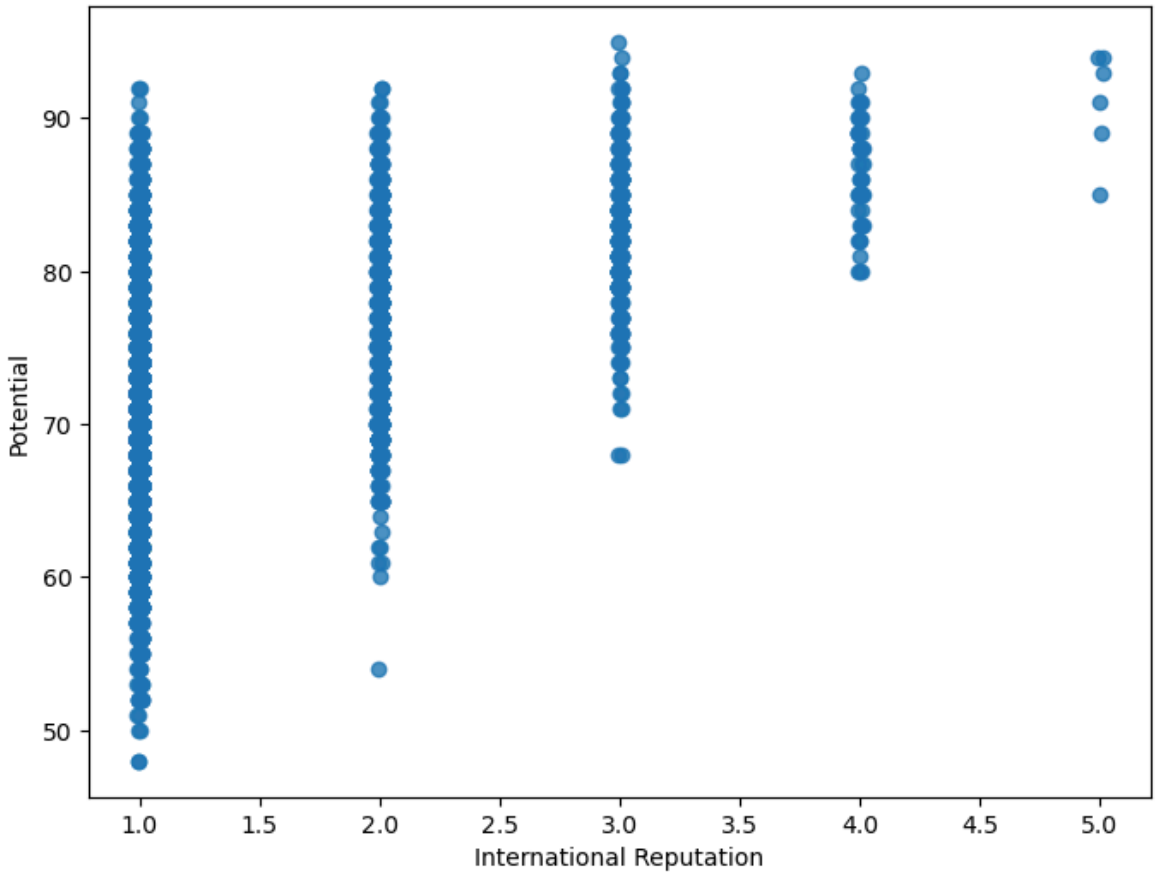
In [54]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa19,color='orange')
plt.show()
```



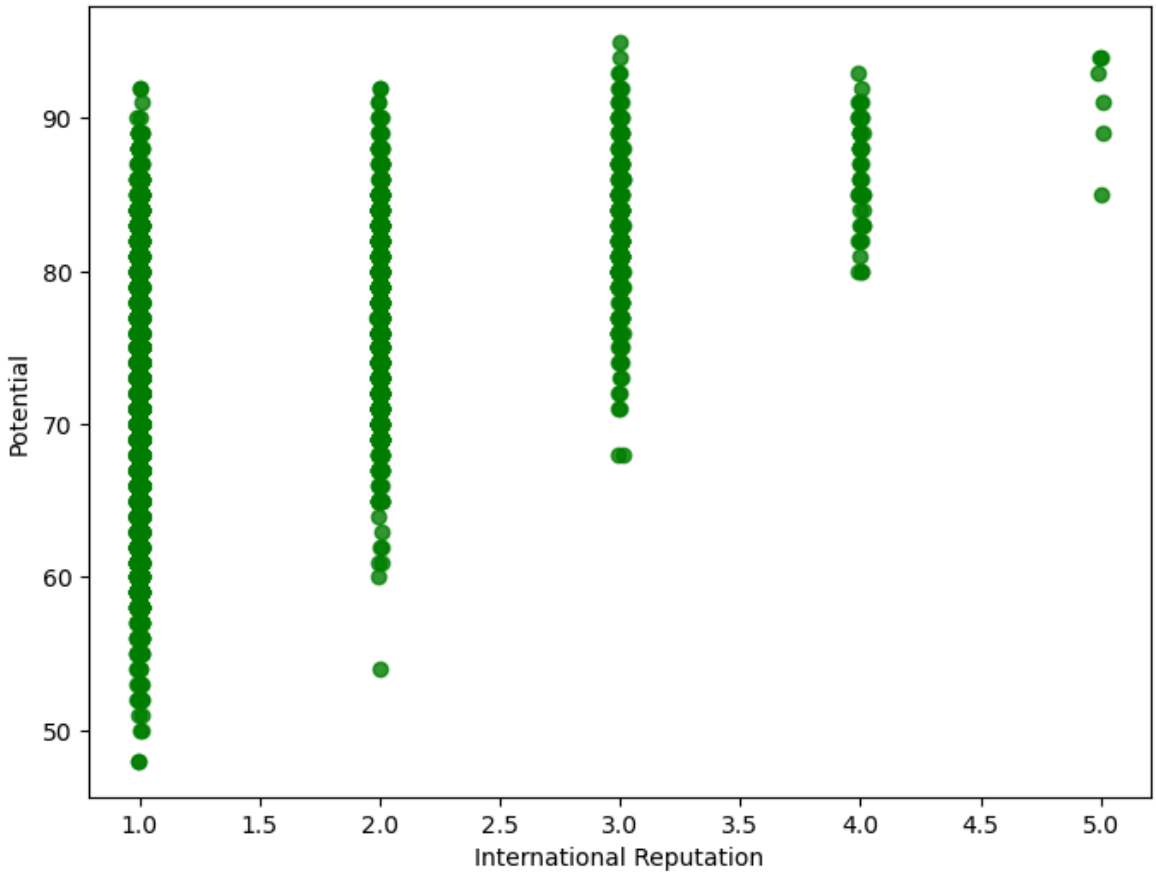
In [55]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa19, color= "g", marker="+"
plt.show()
```



In [56]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.regplot(x="International Reputation", y="Potential", data=fifa19, x_jitter=.
plt.show()
```



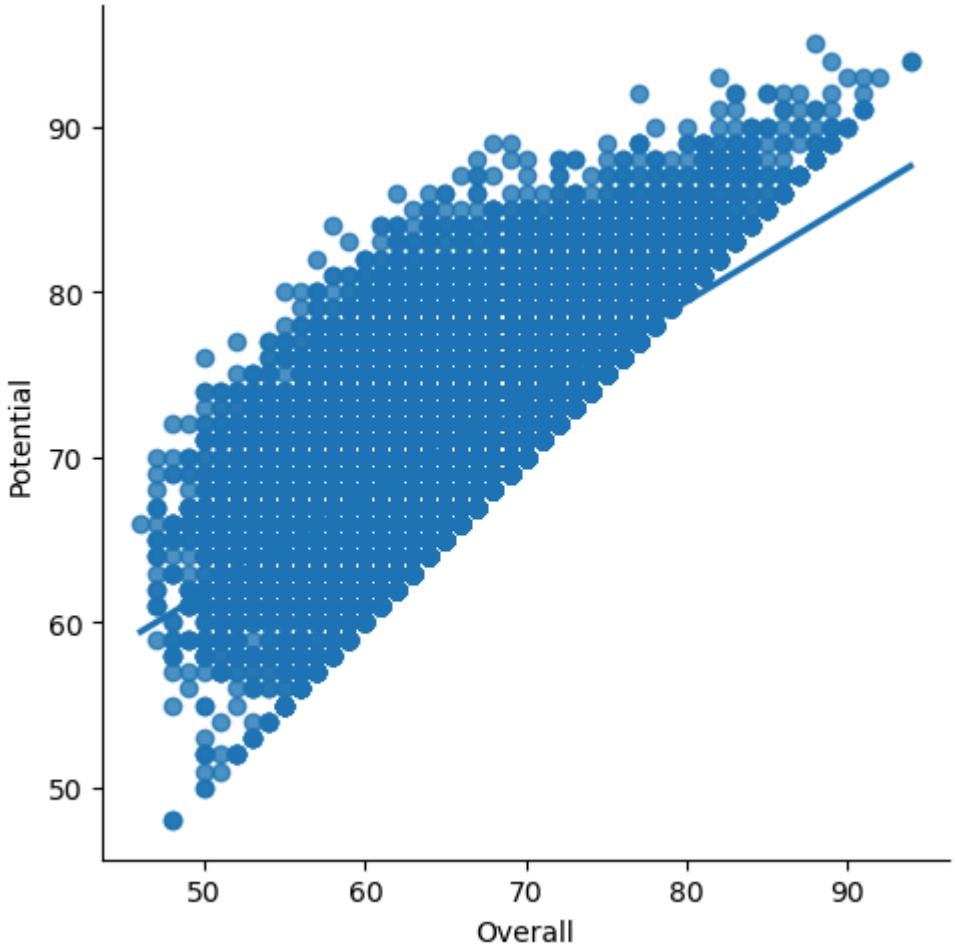
In [57]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.regplot(x="International Reputation", y="Potential", data=fifa19, x_jitter=.
plt.show()
```



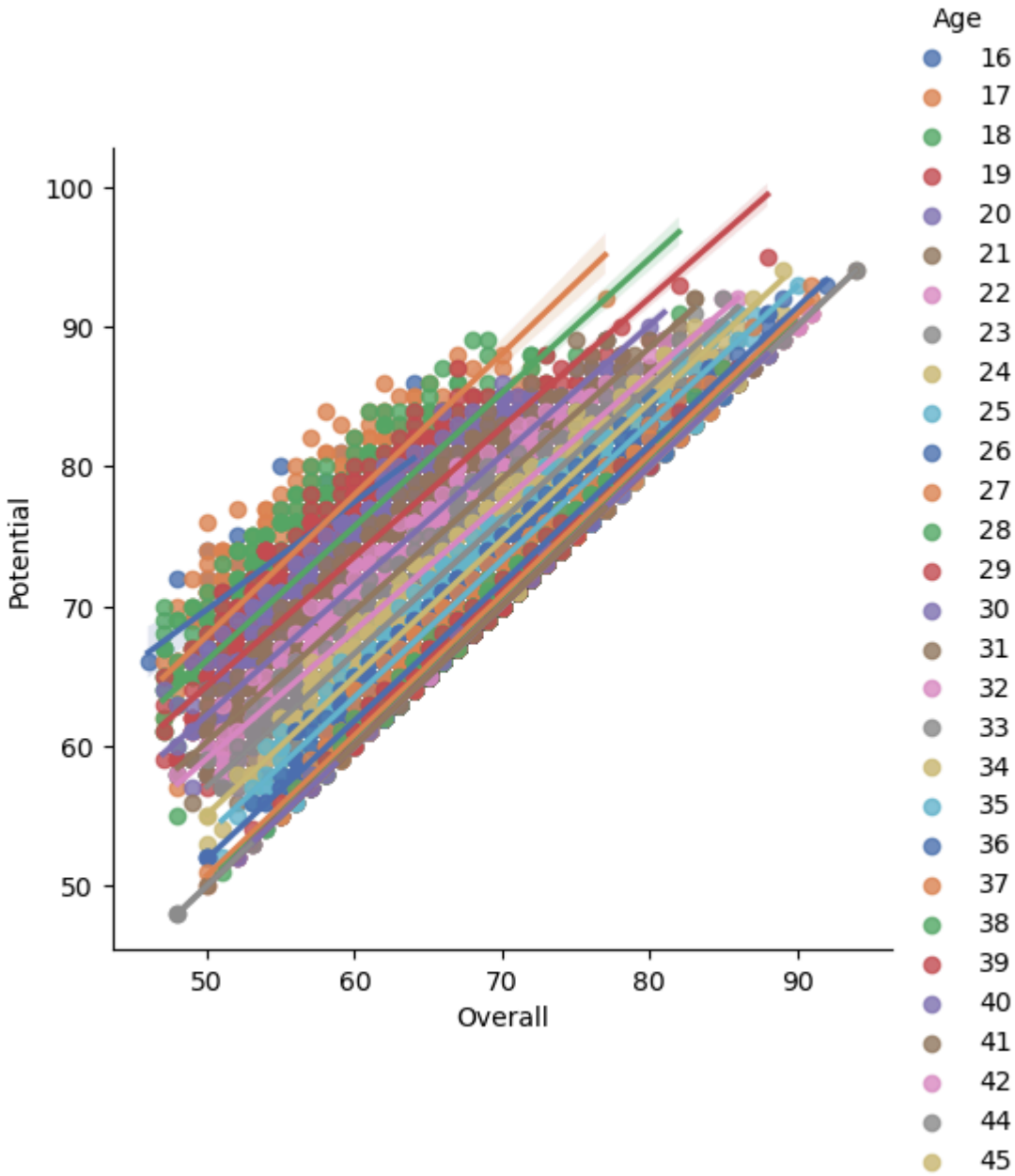
lmplot() funvction

---this function combines regplot() and facetgird---

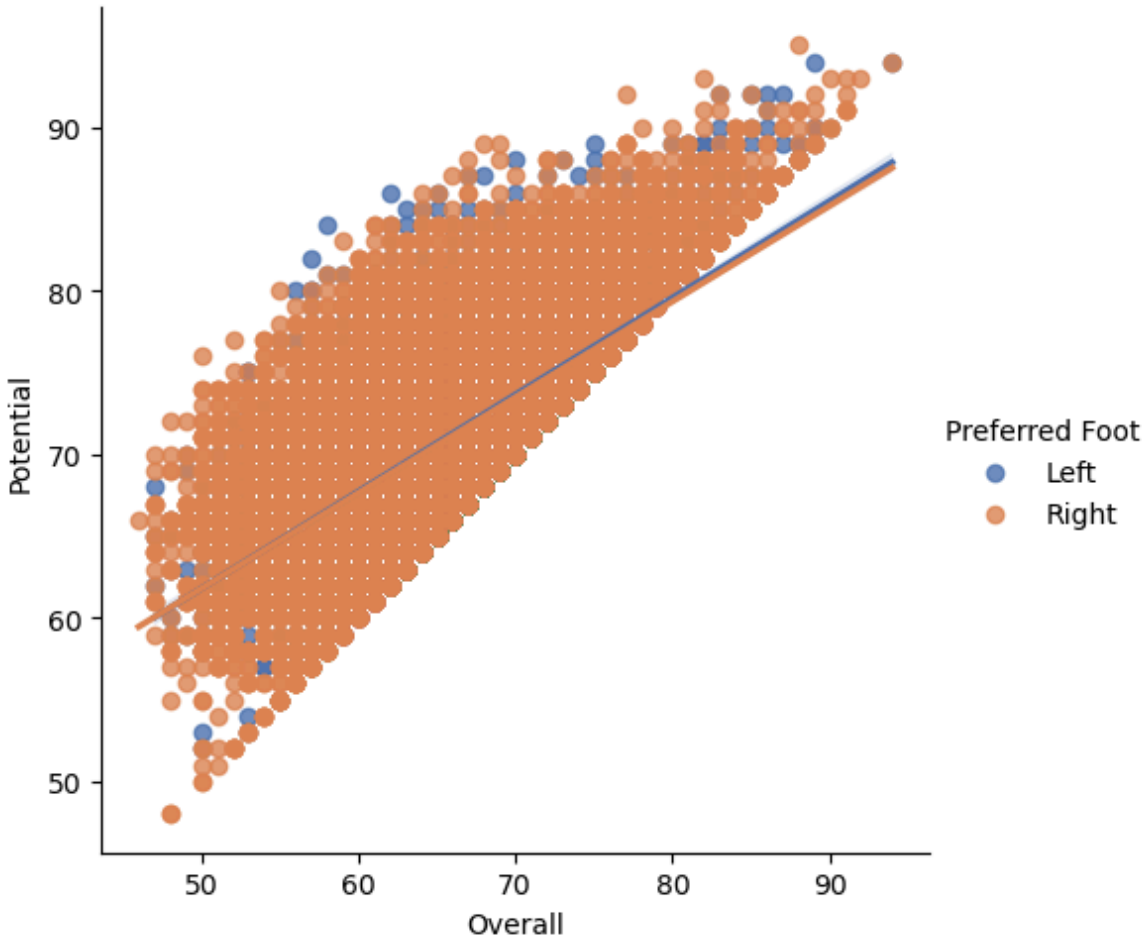
In [58]:
```
g=sns.lmplot(x='Overall',y='Potential',data=fifa19,palette='deep')
plt.show()
```



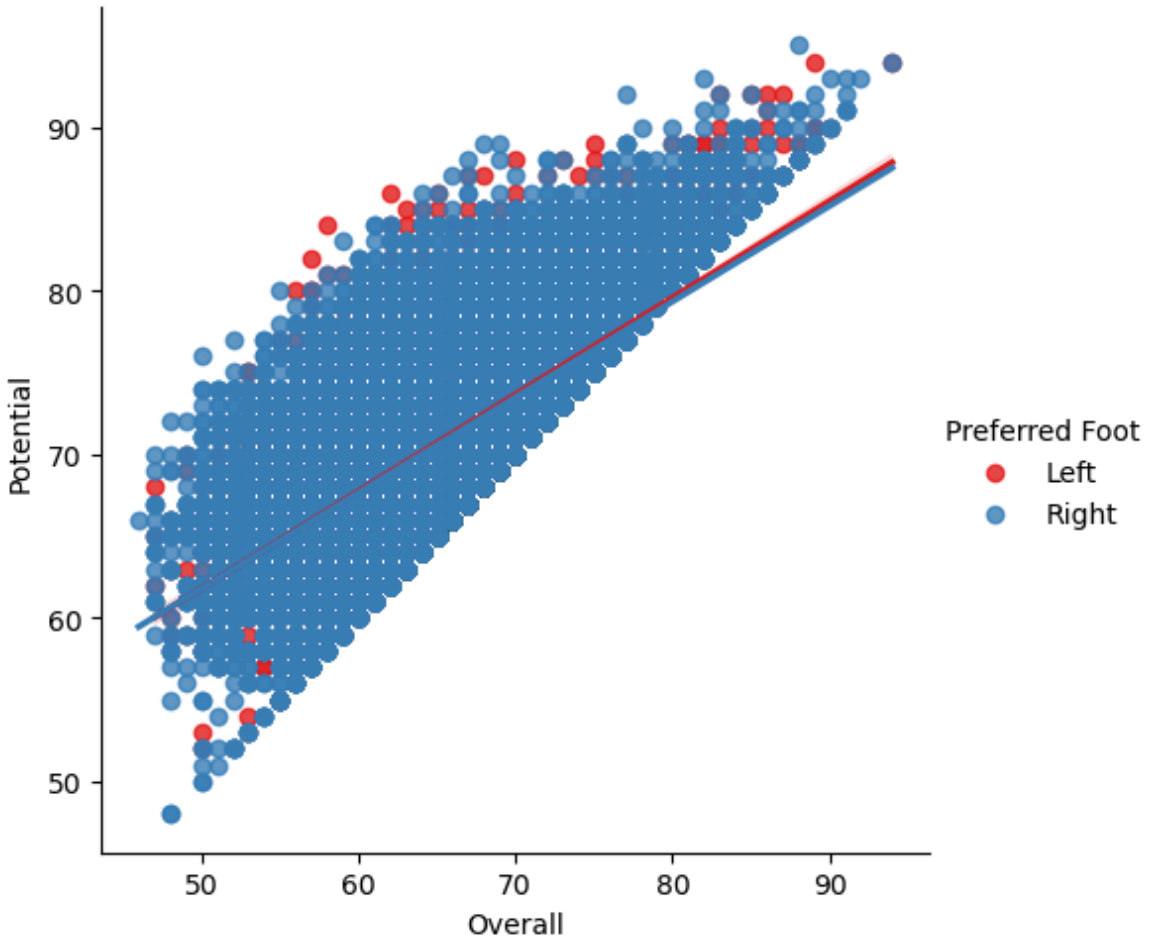
In [59]:
```
g1=sns.lmplot(x='Overall',y='Potential',data=fifa19,hue='Age',palette='deep')
plt.show()
```



In [60]:
```
g2=sns.lmplot(x='Overall',y='Potential',data=fifa19,hue='Preferred Foot',palette
plt.show()
```



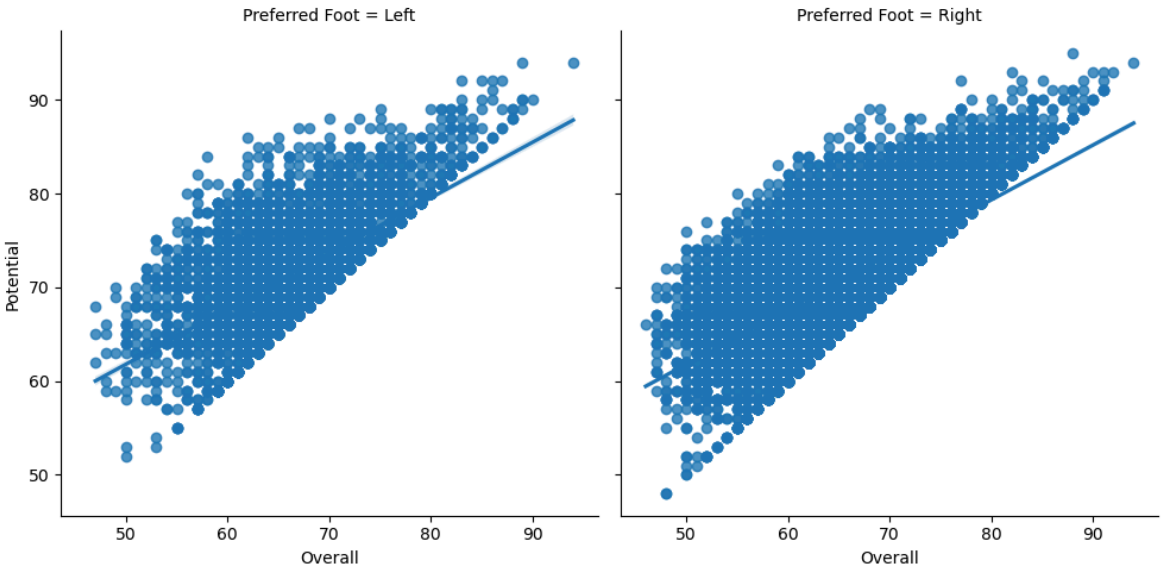
In [61]:
```
g= sns.lmplot(x="Overall", y="Potential", hue="Preferred Foot", data=fifa19, pal
plt.show()
```



In [ ]:

In [62]:
```
#we can plot the lvls of the third var across diff cols as follows--
```

In [63]:
```
g= sns.lmplot(x="Overall", y="Potential", col="Preferred Foot", data=fifa19,pale
plt.show()
```

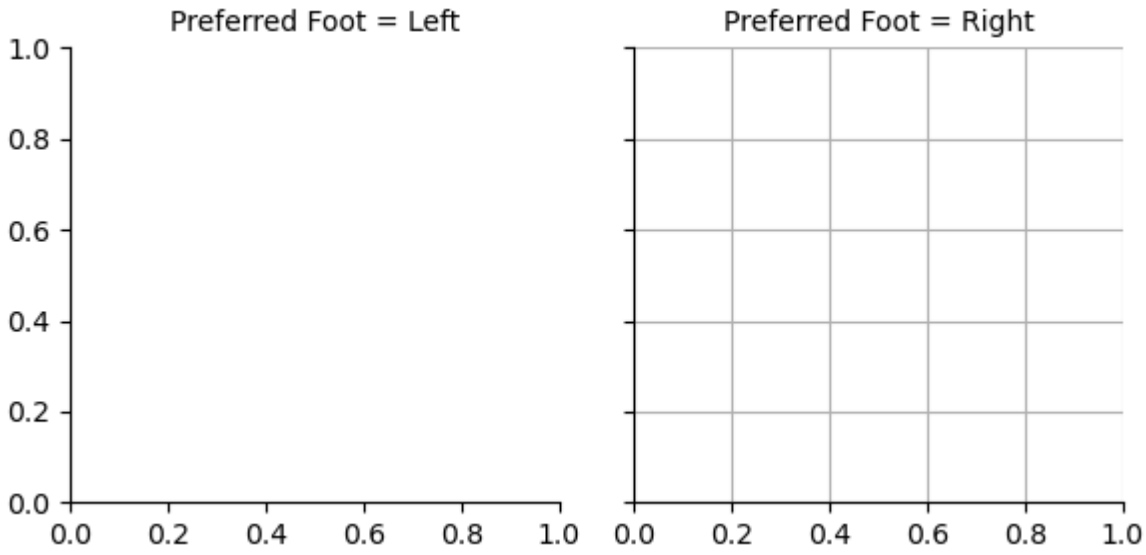


Multi-plot grids

Seaborn `FacetGrid()` function

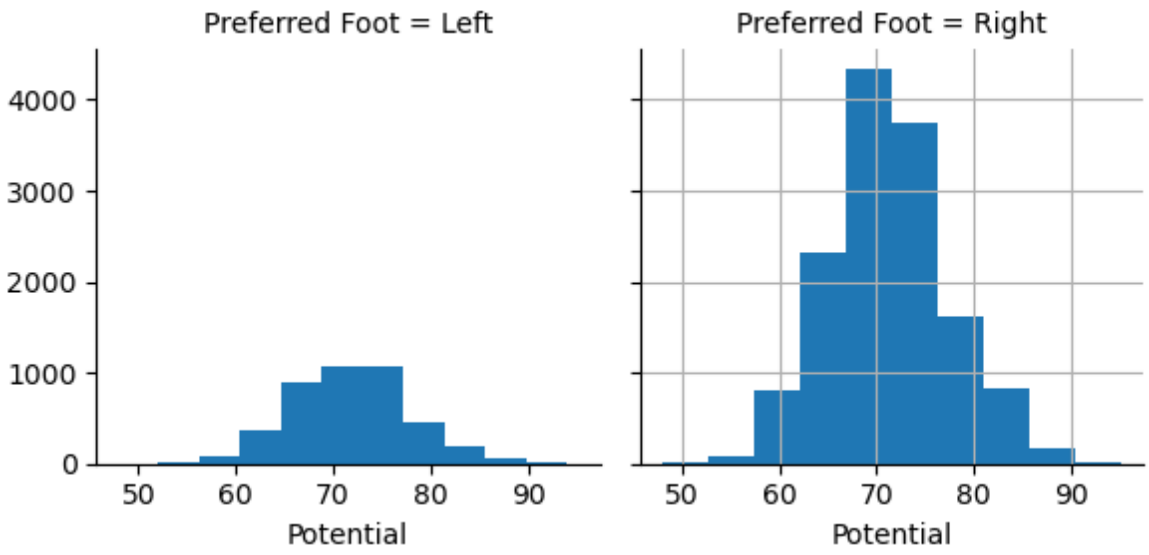
- The FacetGrid class is useful when you want to visualize the distribution of a variable or the relationship between multiple variables separately within subsets of your dataset.
- A FacetGrid can be drawn with up to three dimensions - `row`, `col` and `hue`. The first two have obvious correspondence with the resulting array of axes - the `hue` variable is a third dimension along a depth axis, where different levels are plotted with different colors.
- The class is used by initializing a FacetGrid object with a dataframe and the names of the variables that will form the `row`, `column` or `hue` dimensions of the grid.
- These variables should be categorical or discrete, and then the data at each level of the variable will be used for a facet along that axis.

In [64]:
```
g=sns.FacetGrid(fifa19,col='Preferred Foot')
plt.grid(g)
plt.show()
```

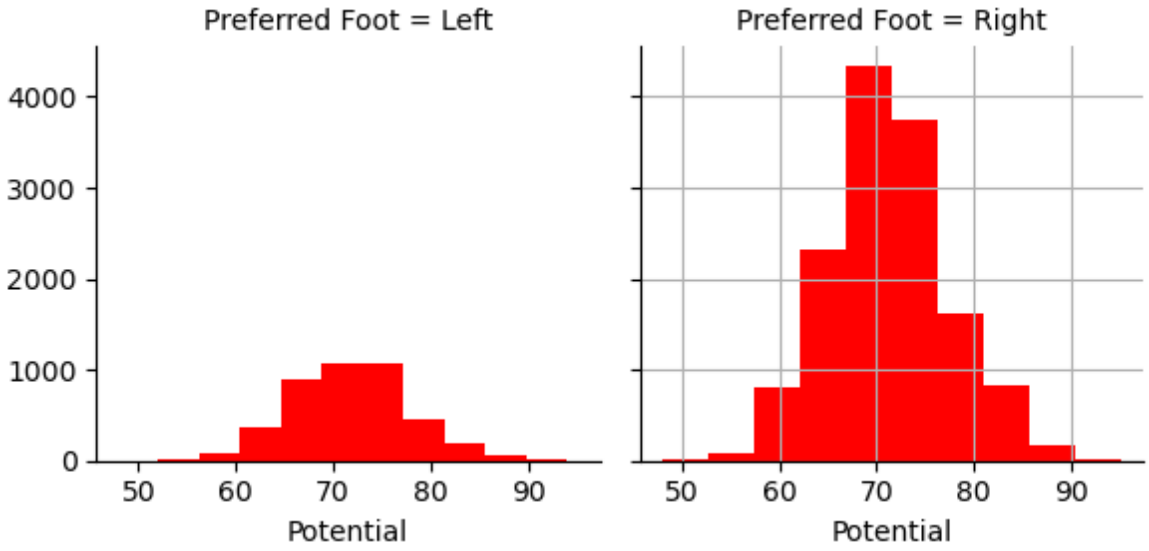


In [65]:
```
# we can also draw a univariate plot of potential var on each facet as follows-
```

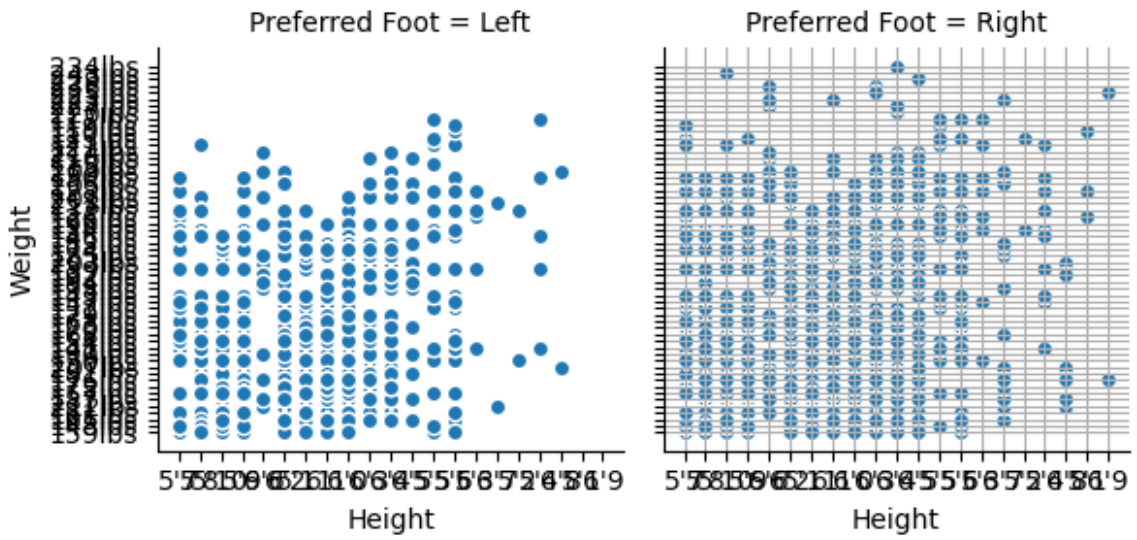
In [66]:
```
g = sns.FacetGrid(fifa19, col="Preferred Foot")
g = g.map(plt.hist, "Potential")
plt.grid(g)
plt.show()
```



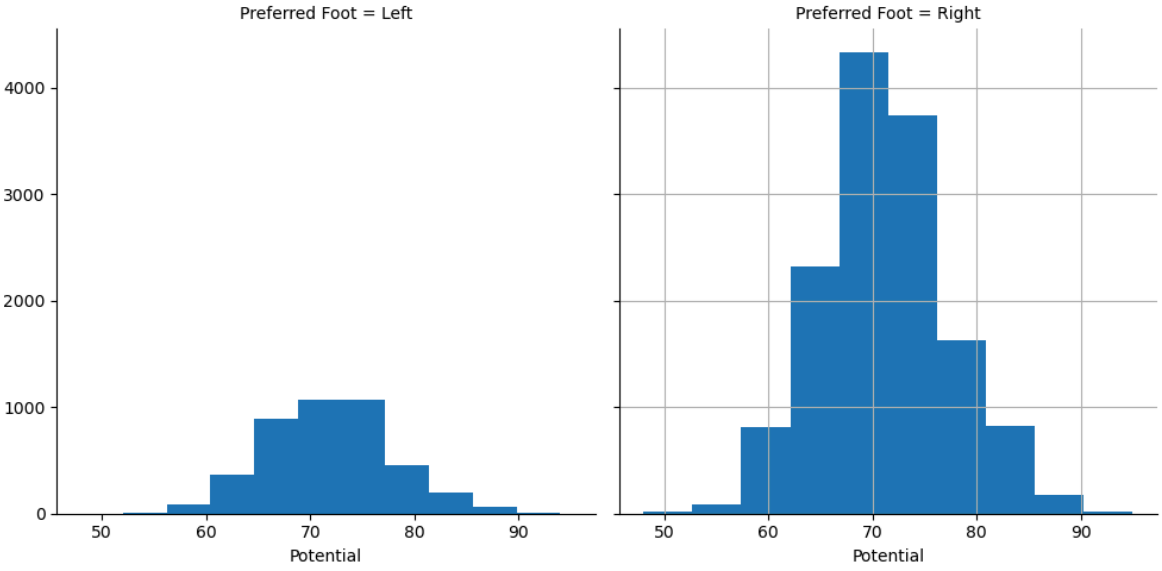
In [67]:
```
g = sns.FacetGrid(fifa19, col="Preferred Foot")
g = g.map(plt.hist, "Potential",bins=10,color='r')
plt.grid(g)
plt.show()
```



In [68]:
```
# here we have plotted bivariate function on each facet
g = sns.FacetGrid(fifa19, col="Preferred Foot")
g = g.map(plt.scatter,'Height','Weight',edgecolor='w').add_legend()
plt.grid(g)
plt.show()
```



In [69]:
```
g = sns.FacetGrid(fifa19, col="Preferred Foot",height=5,aspect=1)
g = g.map(plt.hist, "Potential")
plt.grid(g)
plt.show()
```



Seaborn `Pairgrid()` function

- This function plots subplot grid for plotting pairwise relationships in a dataset.
- This class maps each variable in a dataset onto a column and row in a grid of multiple axes.
- Different axes-level plotting functions can be used to draw bivariate plots in the upper and lower triangles, and the the marginal distribution of each variable can be shown on the diagonal.
- It can also represent an additional level of conditionalization with the hue parameter, which plots different subets of data in different colors.
- This uses color to resolve elements on a third dimension, but only draws subsets on top of each other and will not tailor the hue parameter for the specific visualization

the way that axes-level functions that accept hue will.

In [70]:
```
fifa19_new = fifa19[['Age', 'Potential', 'Strength', 'Stamina', 'Preferred Foot'
```

In [71]:
```
fifa19_new
```

Out[71]:

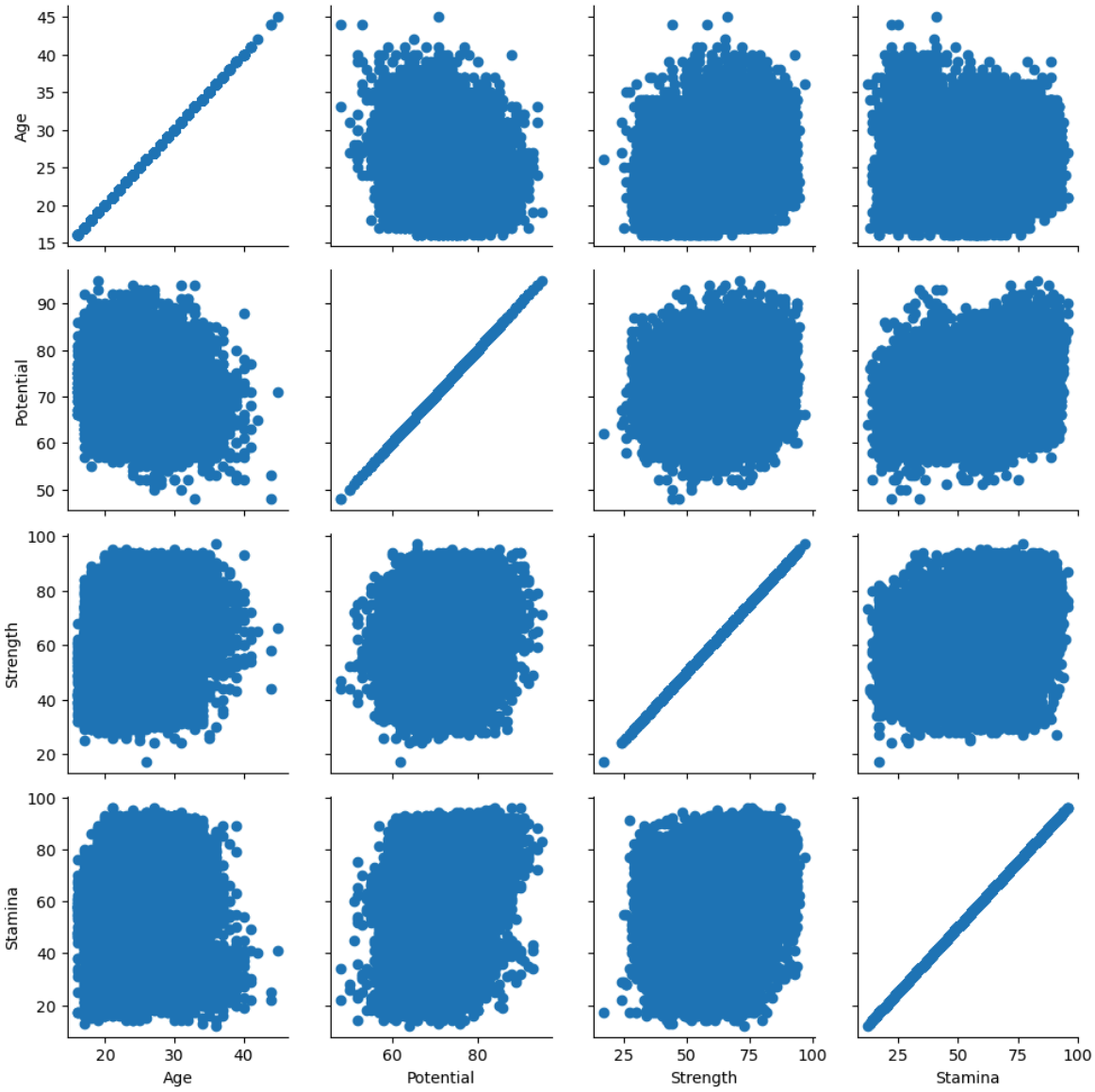
	Age	Potential	Strength	Stamina	Preferred Foot
0	31	94	59.0	72.0	Left
1	33	94	79.0	88.0	Right
2	26	93	49.0	81.0	Right
3	27	93	64.0	43.0	Right
4	27	92	75.0	90.0	Right
...	...	...	...	...	...
18202	19	65	47.0	40.0	Right
18203	19	63	67.0	43.0	Right
18204	16	67	32.0	55.0	Right
18205	17	66	48.0	40.0	Right
18206	16	66	60.0	47.0	Right

18207 rows × 5 columns

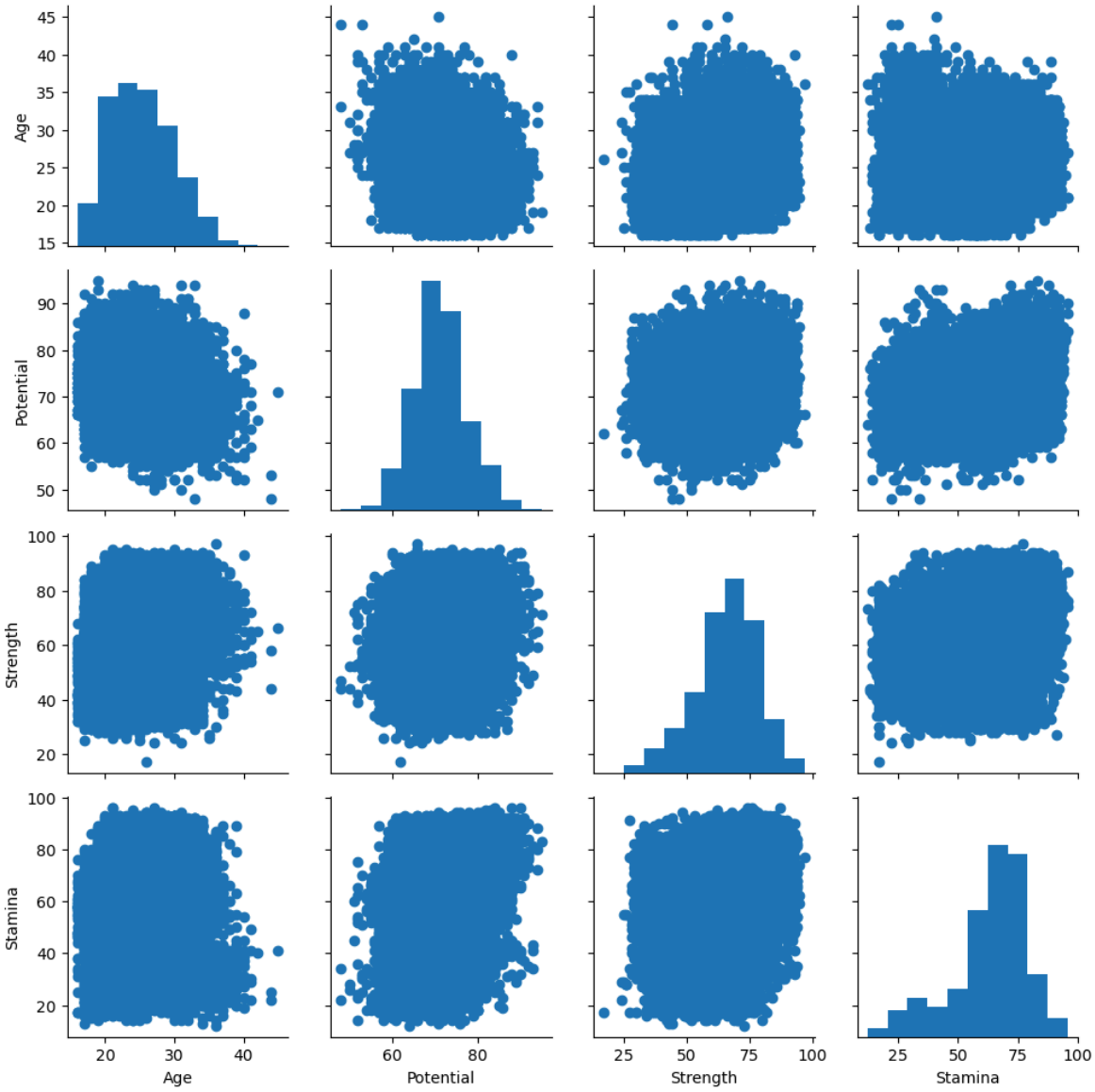
In [72]:
```
type(fifa19_new)
```

Out[72]:
```
pandas.core.frame.DataFrame
```

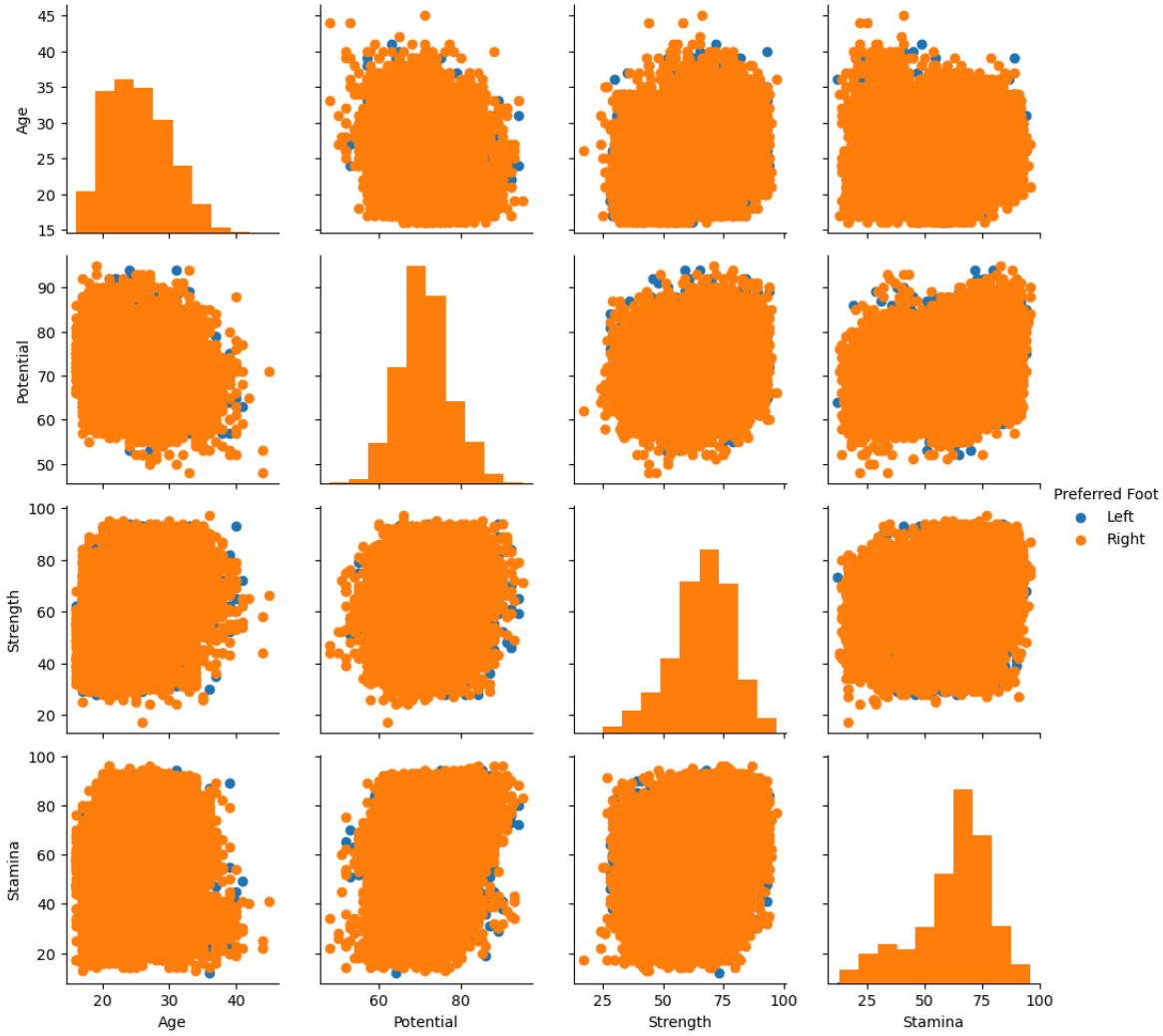
In [73]:
```
g = sns.PairGrid(fifa19_new)
g= g.map(plt.scatter)
plt.show()
```



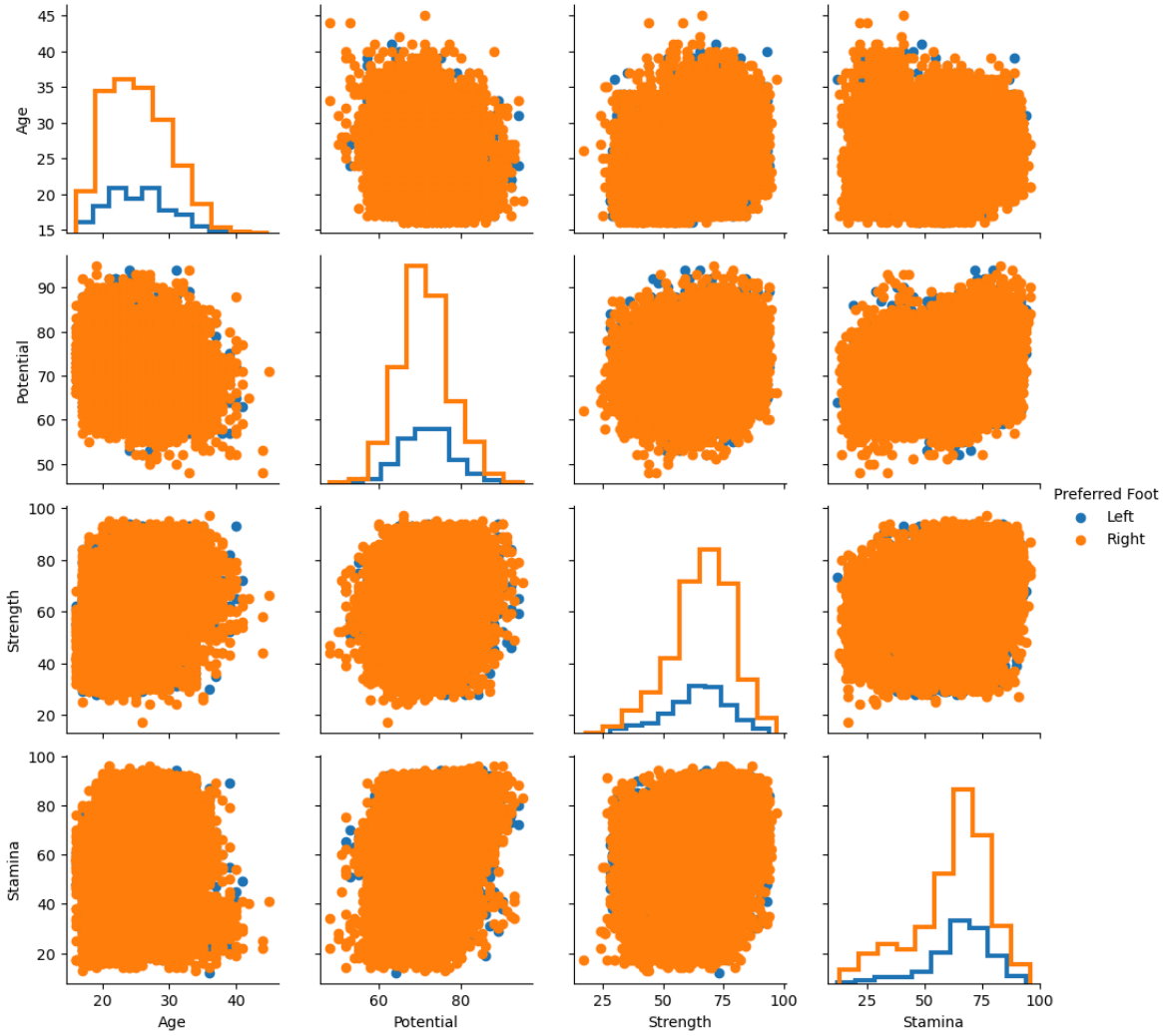
In [74]:
```
g = sns.PairGrid(fifa19_new)
g= g.map_diag(plt.hist)
g = g.map_offdiag(plt.scatter)
plt.show()
```



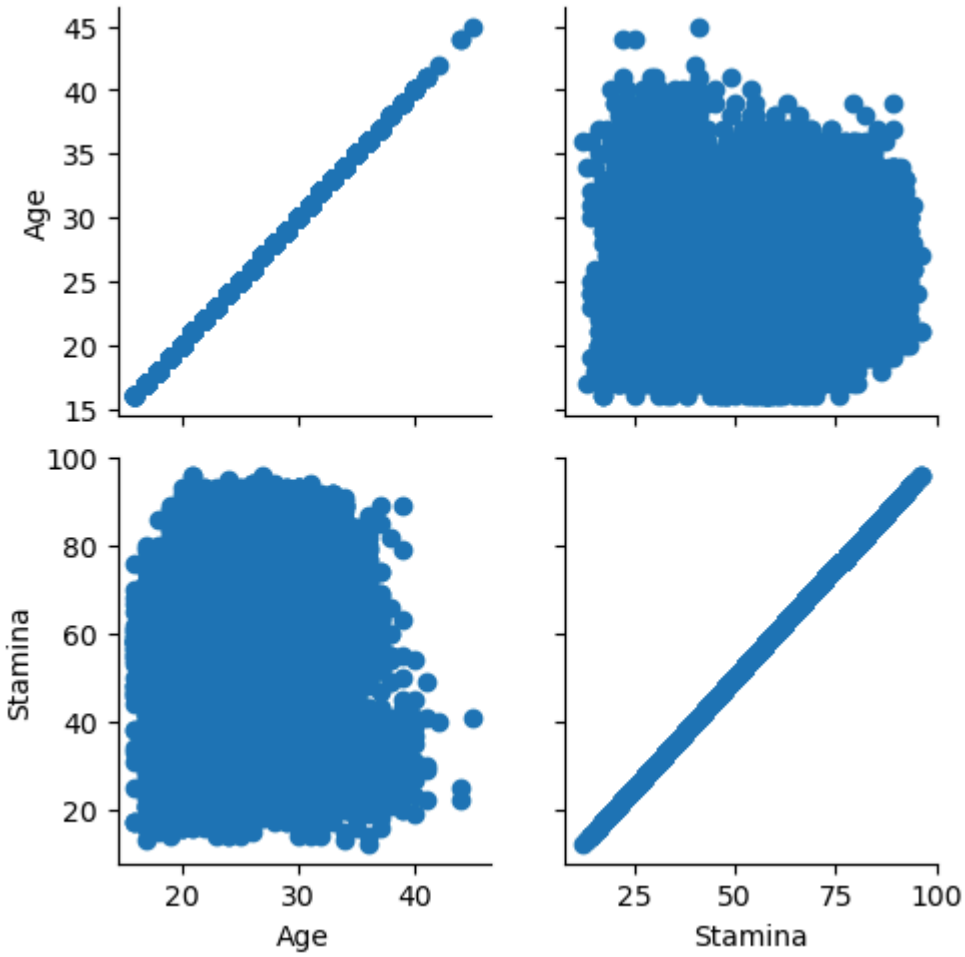
In [75]:
```
g = sns.PairGrid(fifa19_new, hue="Preferred Foot")
g = g.map_diag(plt.hist)
g = g.map_offdiag(plt.scatter)
g = g.add_legend()
plt.show()
```



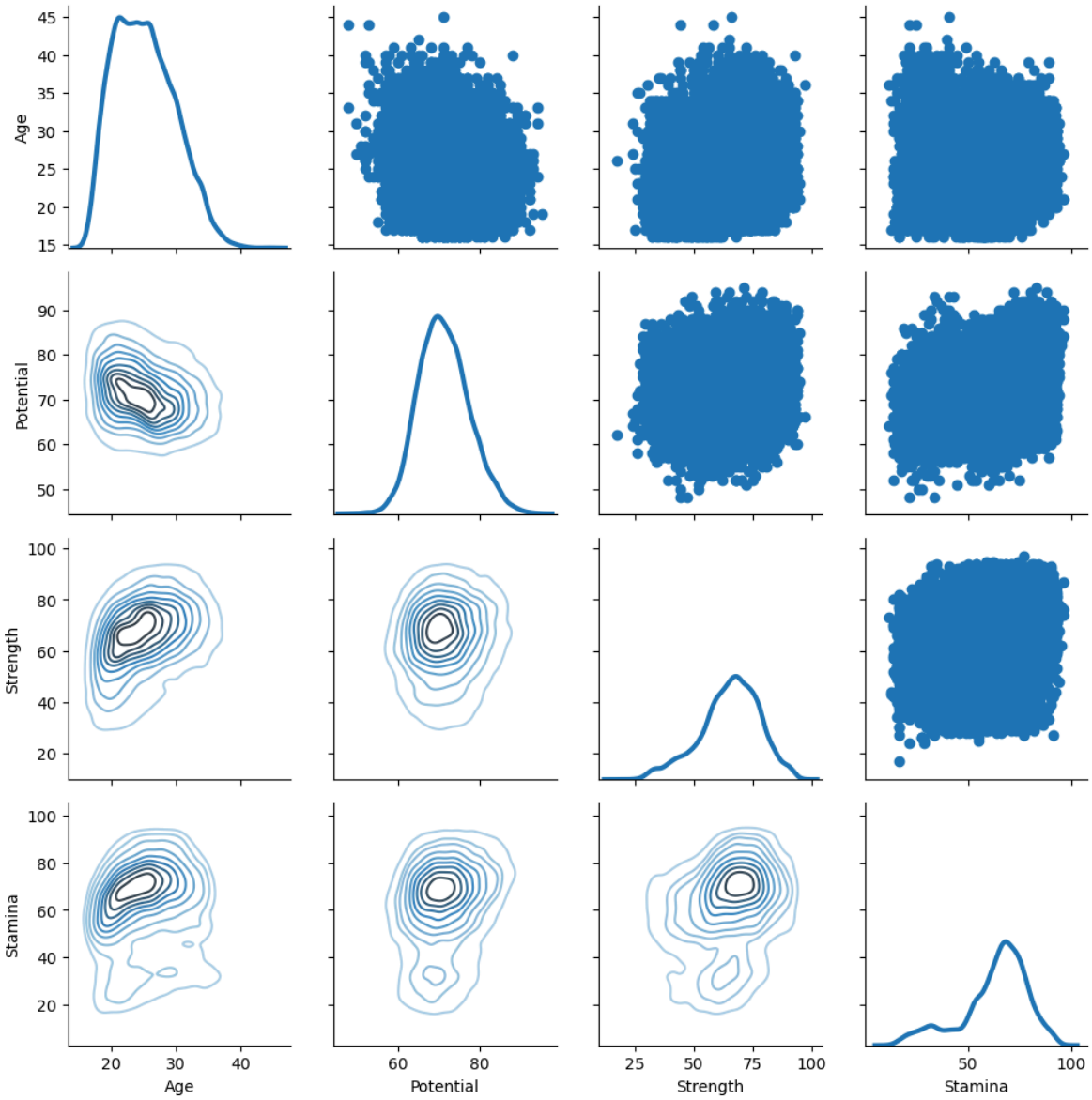
In [76]:
```
#we can use diff style to show mul histograms as follows:
g= sns.PairGrid(fifa19_new,hue='Preferred Foot')
g=g.map_diag(plt.hist, histtype='step',linewidth=3)
g=g.map_offdiag(plt.scatter)
g=g.add_legend()
plt.show()
```



In [77]:
```
#plotting subset of vars
g=sns.PairGrid(fifa19_new,vars=['Age','Stamina'])
g=g.map(plt.scatter)
plt.show()
```



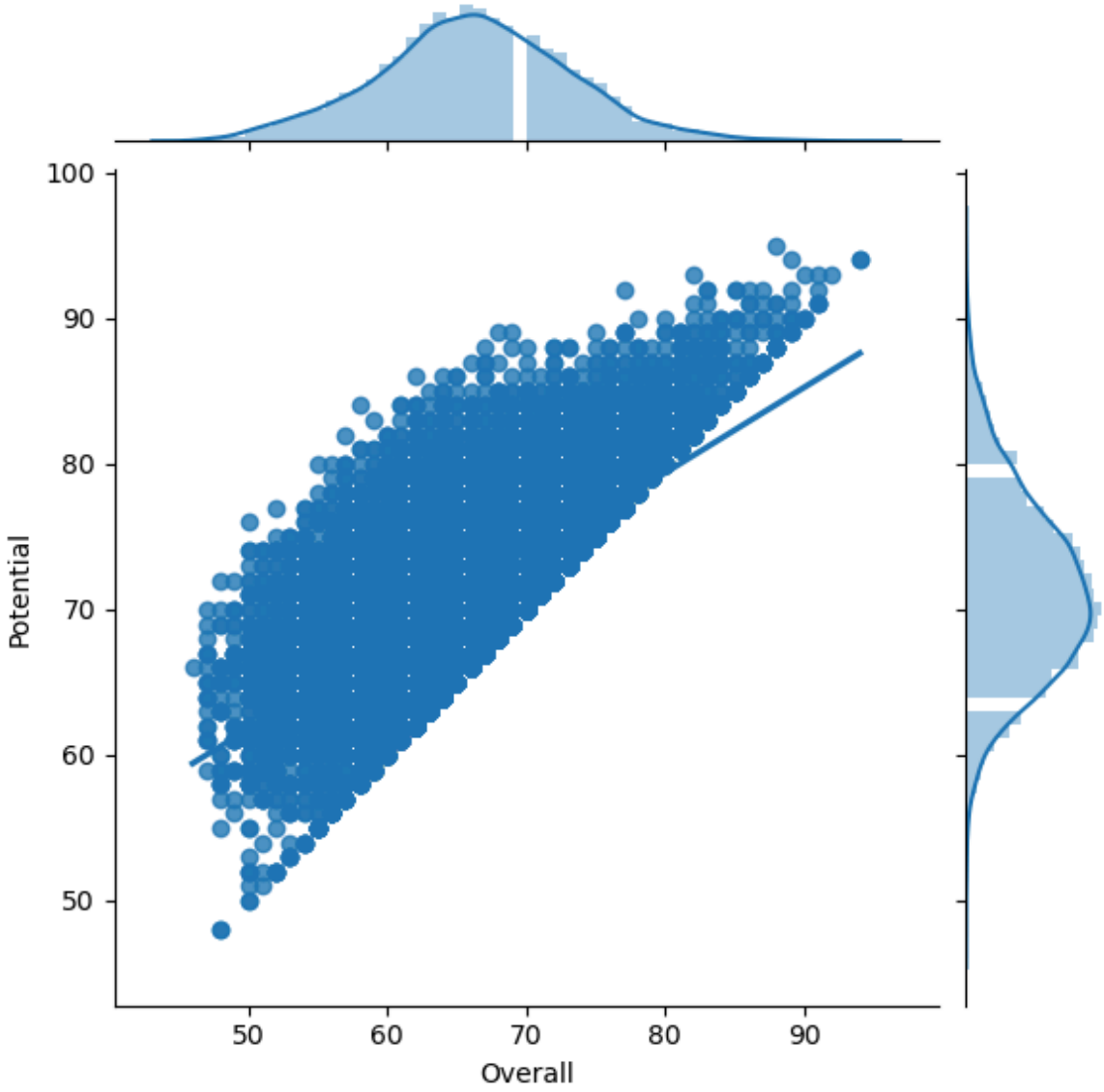
In [78]:
```
g=sns.PairGrid(fifa19_new)
g=g.map_upper(plt.scatter)
g=g.map_lower(sns.kdeplot,cmap='Blues_d')
g=g.map_diag(sns.kdeplot,lw=3,legend=False)
plt.show()
```



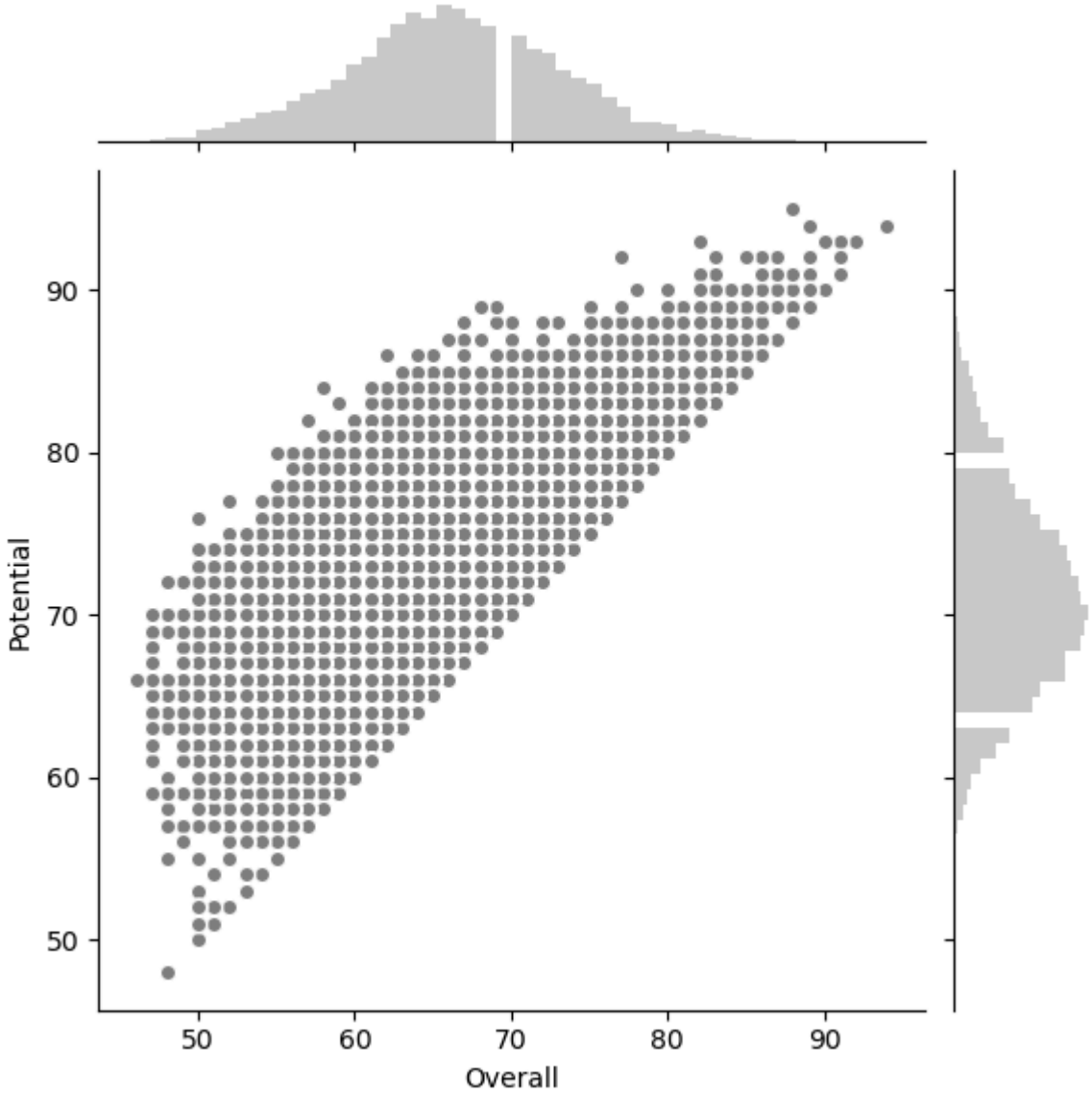
jointgrid() function

-- this func allows to plot bivariate graph with marginal univariate plot, it setup the grid of sublots

In [79]:
```
g = sns.JointGrid(x="Overall", y="Potential", data=fifa19)
g = g.plot(sns.regplot, sns.distplot)
plt.show()
```

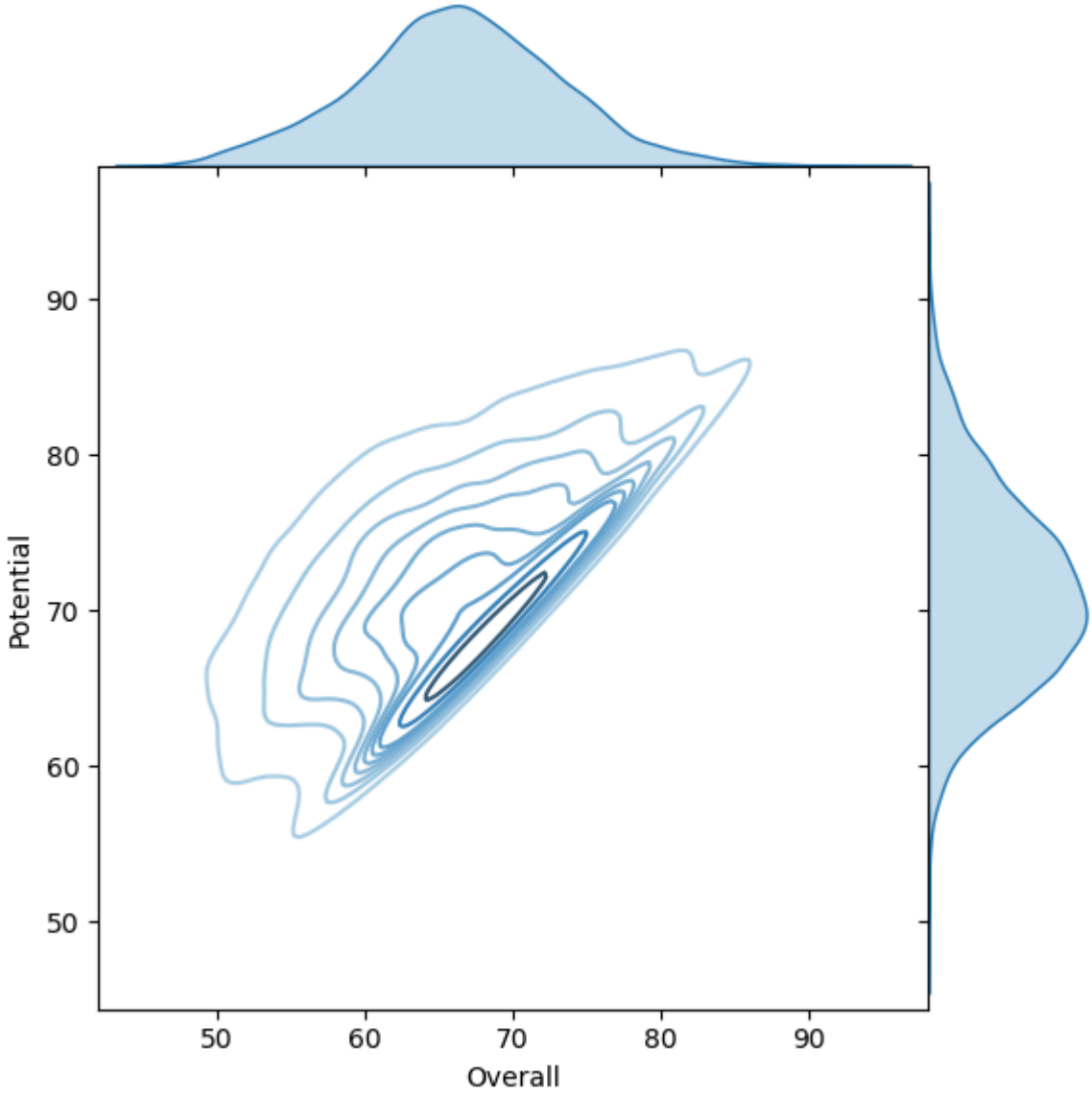


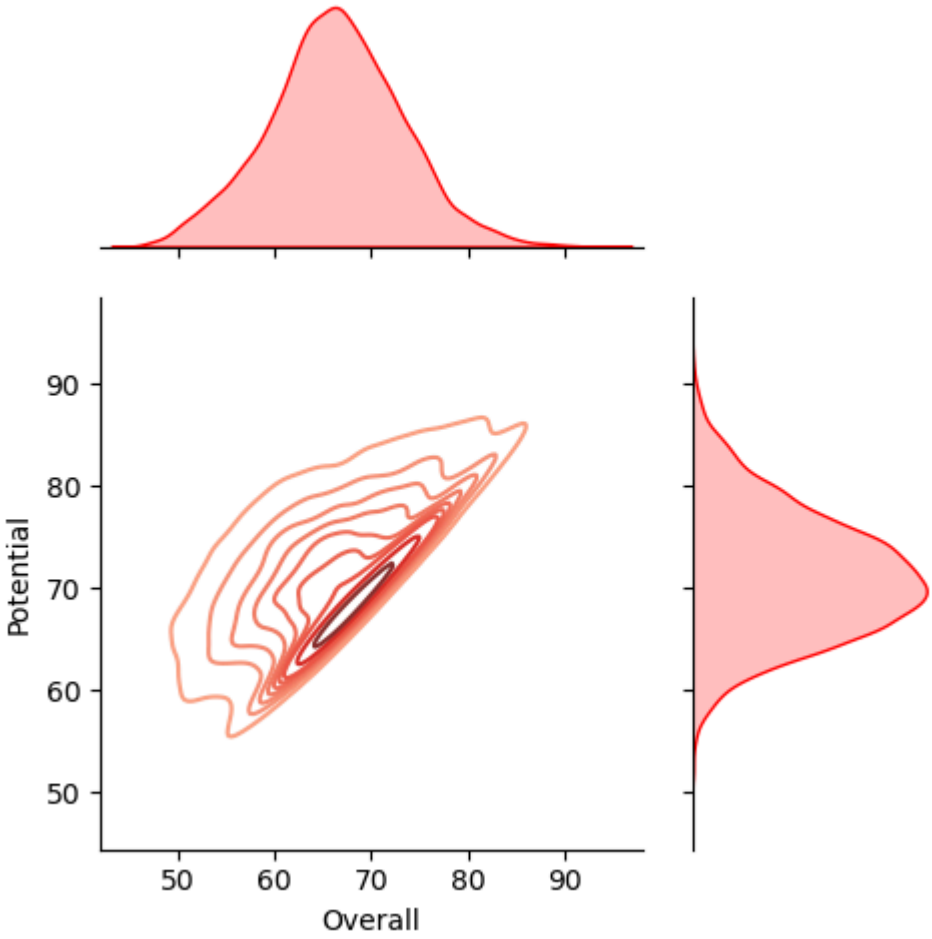
In [80]:
```
g = sns.JointGrid(x="Overall", y="Potential", data=fifa19)
g = g.plot_joint(plt.scatter, color=".5", edgecolor="white")
g = g.plot_marginals(sns.distplot, kde=False, color=".5")
plt.show()
```



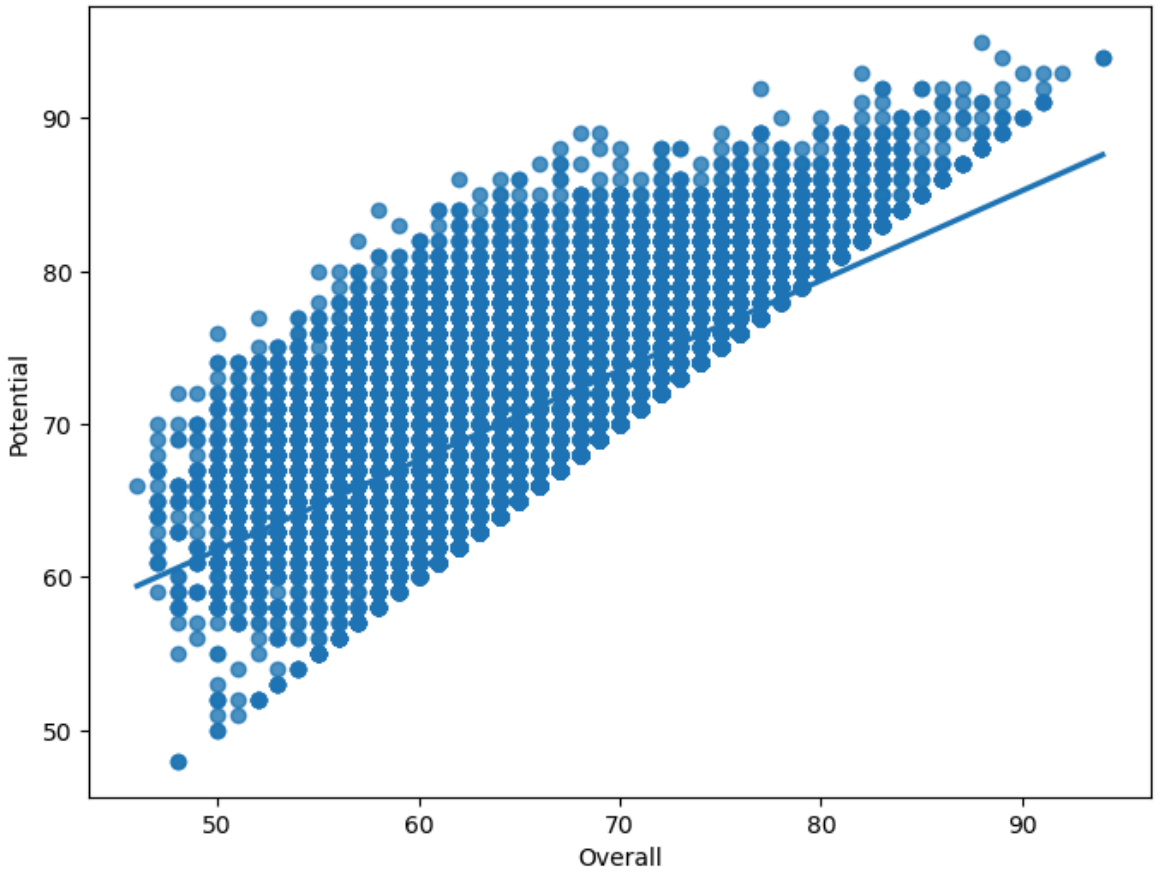
In [81]:
```
g = sns.JointGrid(x="Overall", y="Potential", data=fifa19, space=0)
g = g.plot_joint(sns.kdeplot, cmap="Blues_d")
g = g.plot_marginals(sns.kdeplot, shade=True)
plt.show()
```

In [82]:
```
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)
plt.show()
```

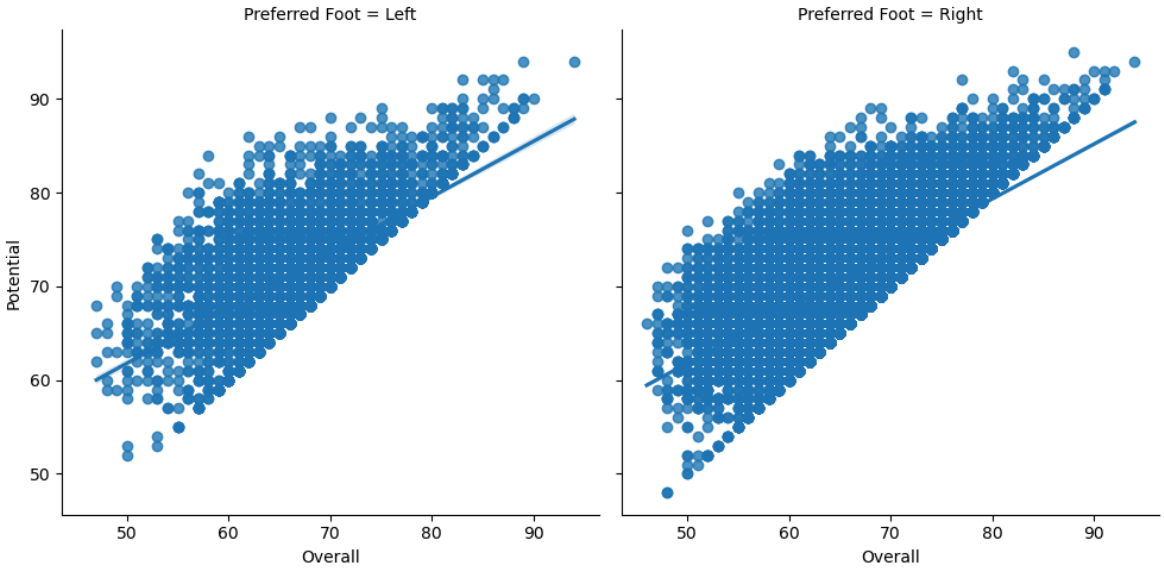

Potential
90
80
70
60
50
50
60
70
80
90
Overall

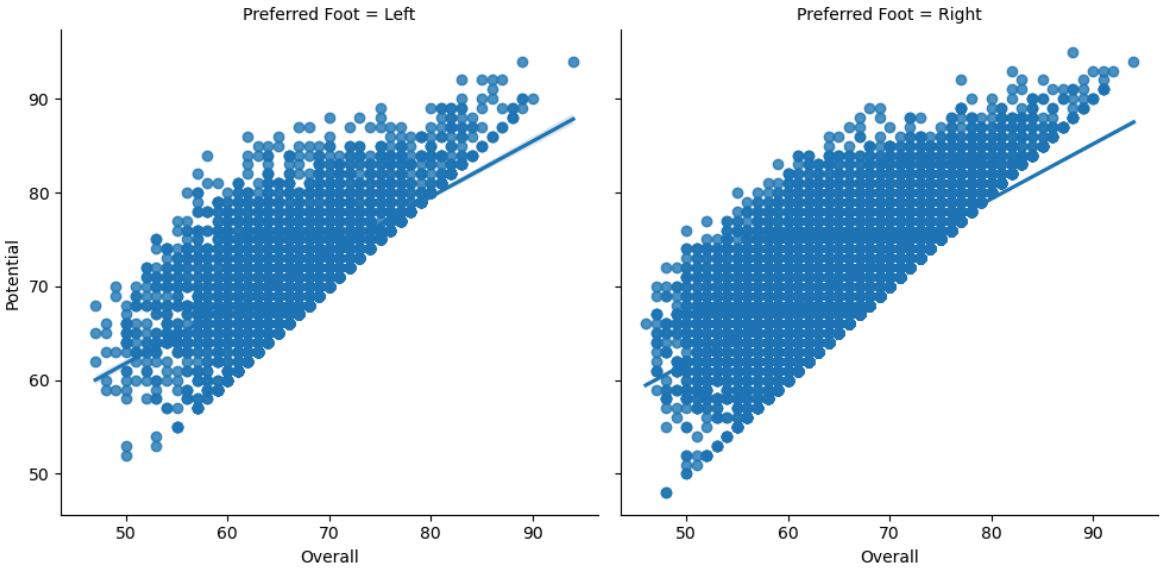


In [83]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="Overall", y="Potential", data=fifa19);
plt.show()
```



In [85]:
```
sns.lmplot(x="Overall", y="Potential", col="Preferred Foot", data=fifa19, col_wr
plt.show()
```

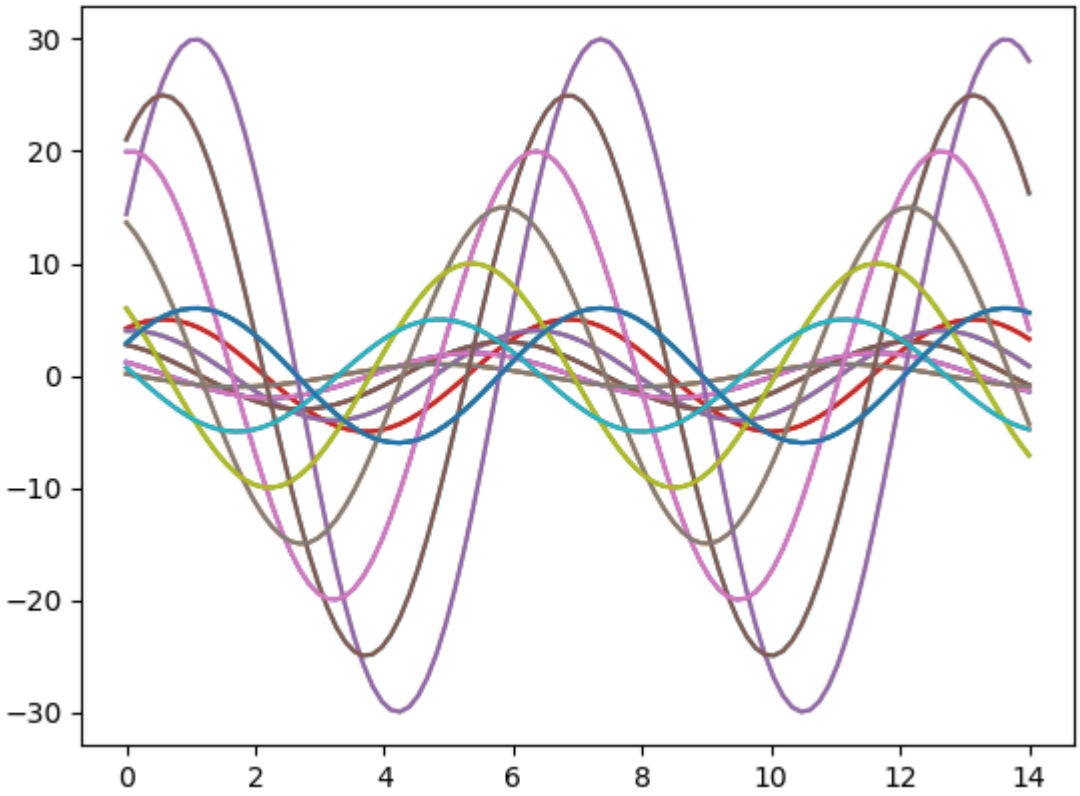


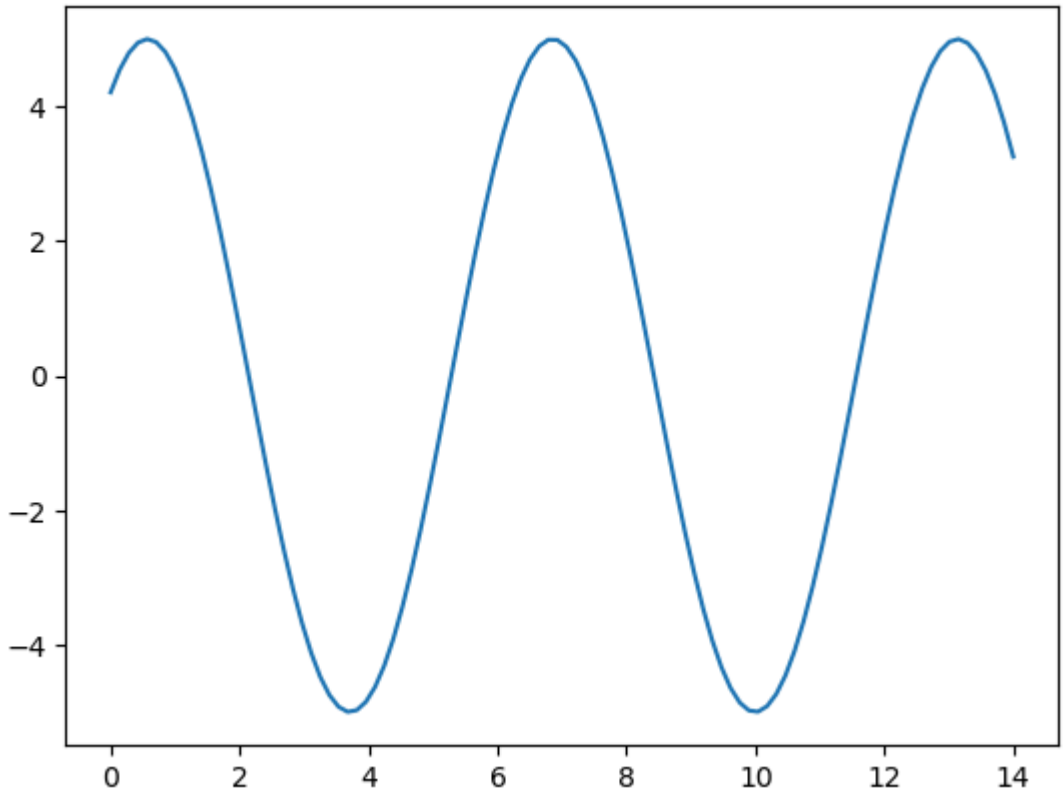


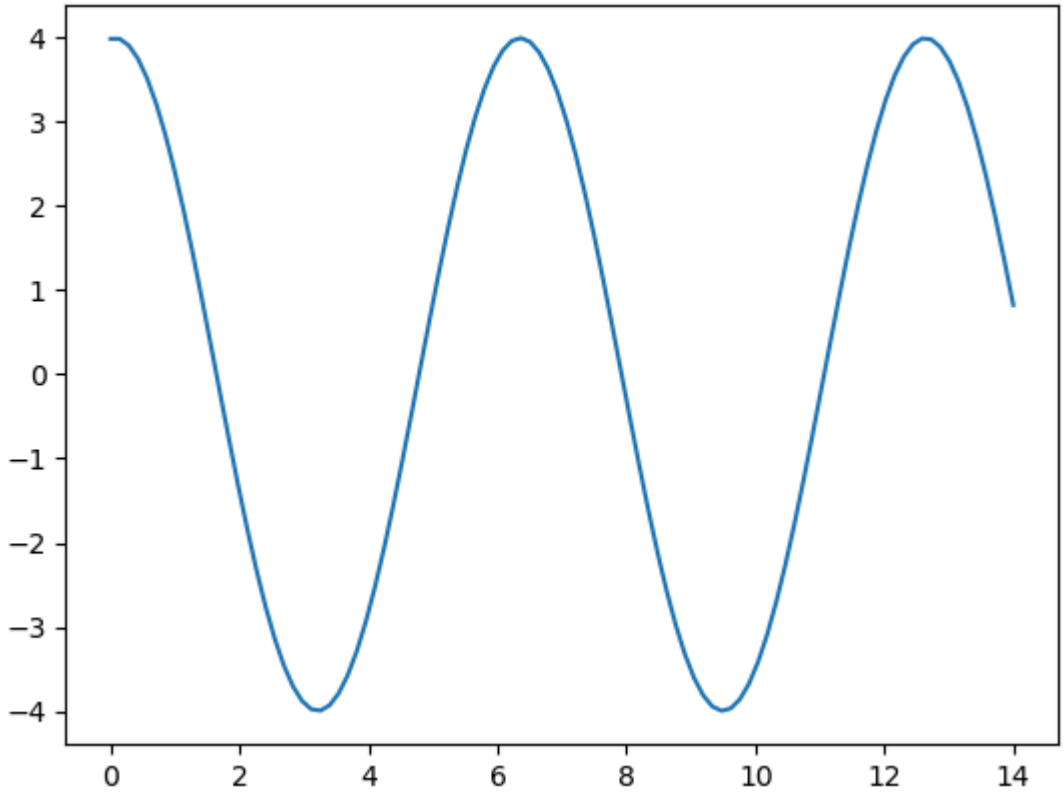
I will define a simple function to plot some offset sine waves, which will help us see the different stylistic parameters as follows -

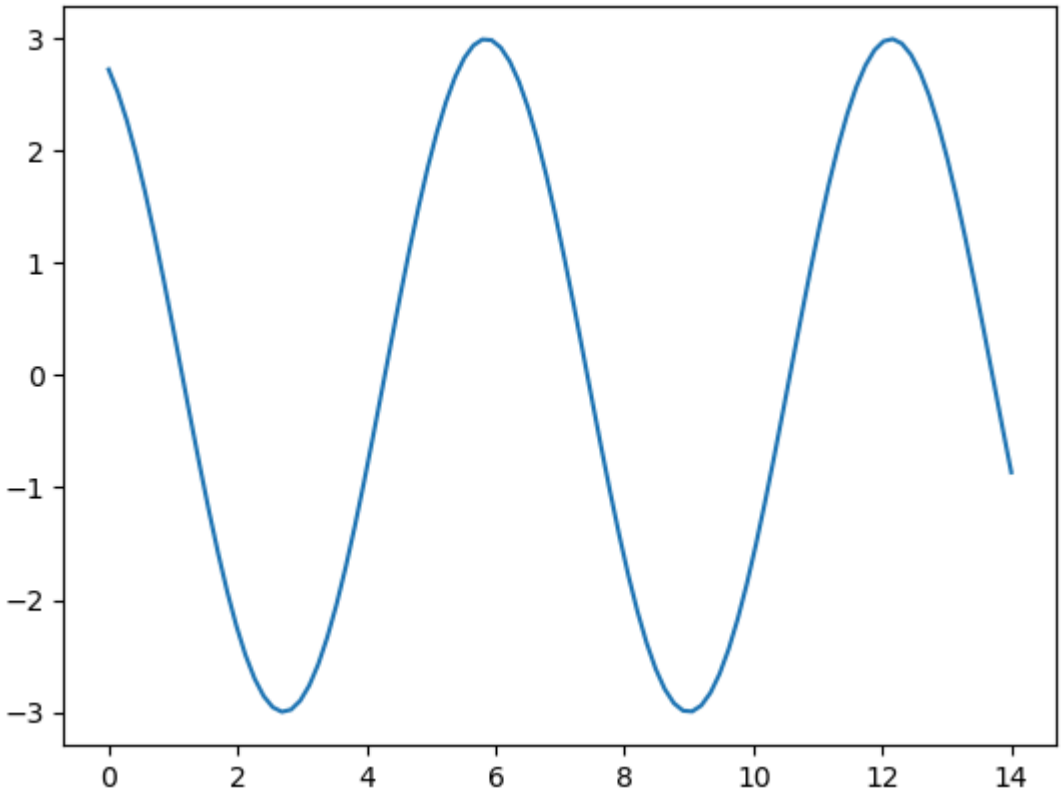
In [96]:
```
def sinplot(flip=1):
    x=np.linspace(0,14,100)
    for i in range(1,7):
        plt.plot(x,np.sin(x+i*.5)*(7-i)*flip)
        plt.show()
```

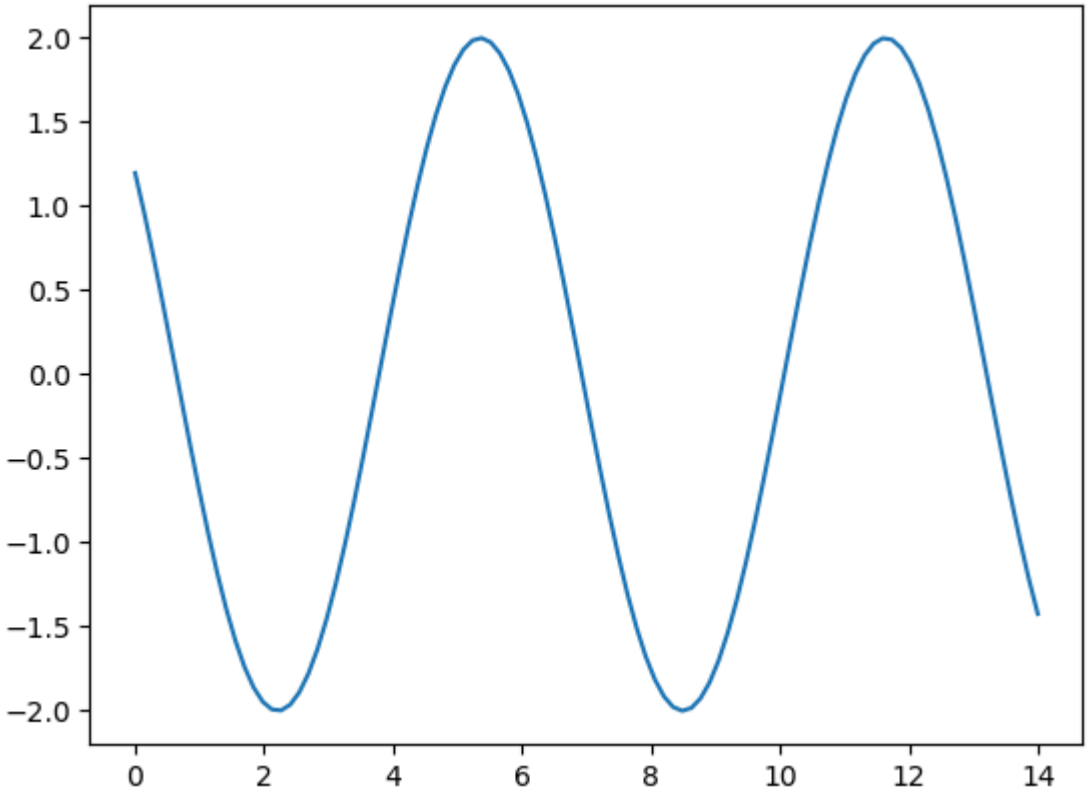
In [97]:
```
sinplot() #this is what plot with default matplotplib parameters looks like
```

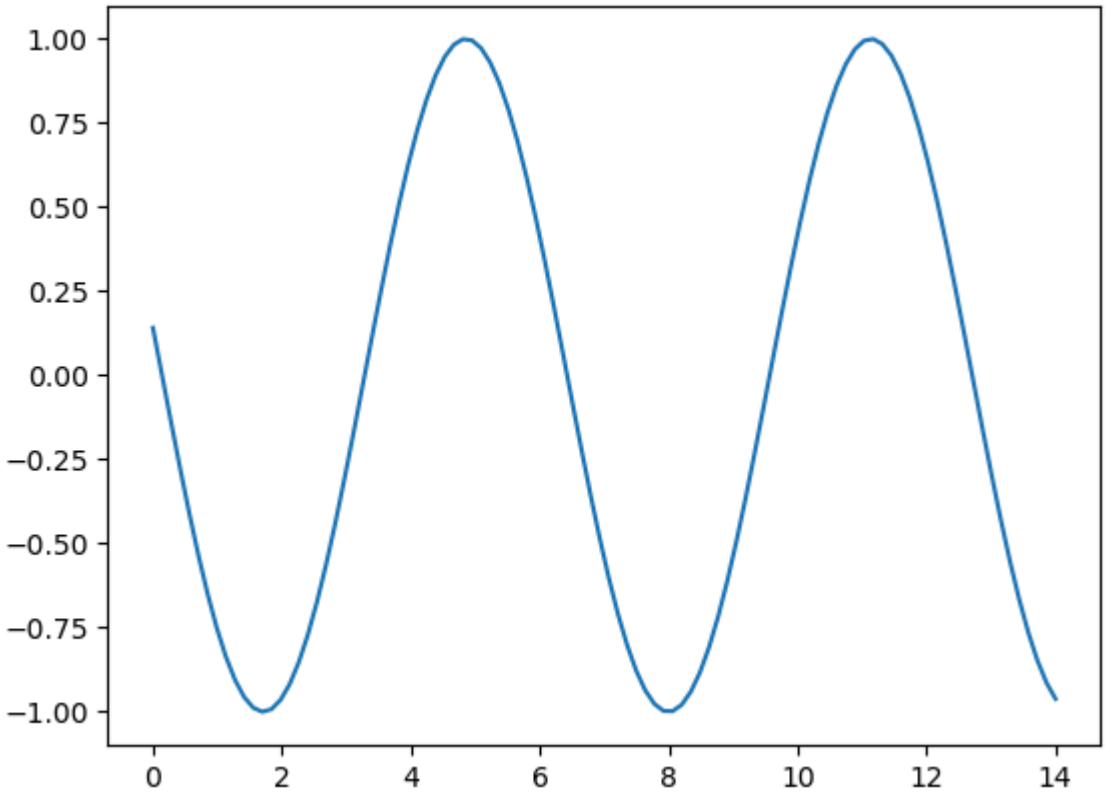



4
2
0
−2
−4
0
2
4
6
8
10
12
14


4
3
2
1
0
−1
−2
−3
−4
0
2
4
6
8
10
12
14

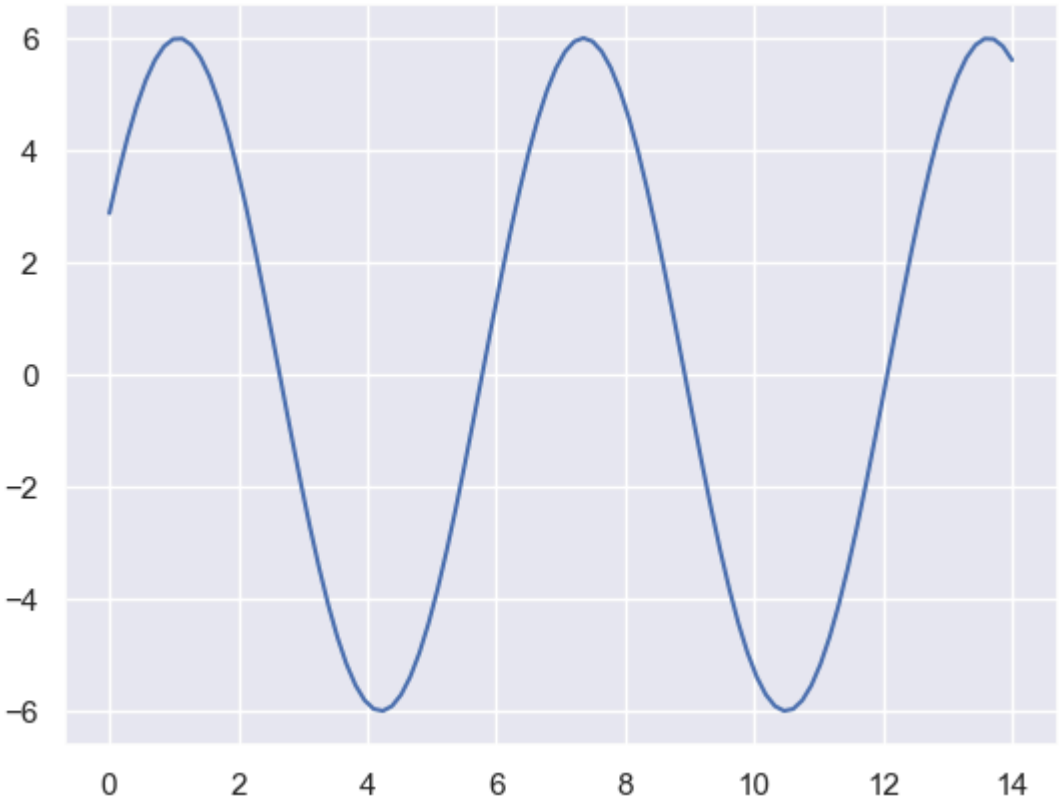

3
2
1
0
−1
−2
−3
0
2
4
6
8
10
12
14

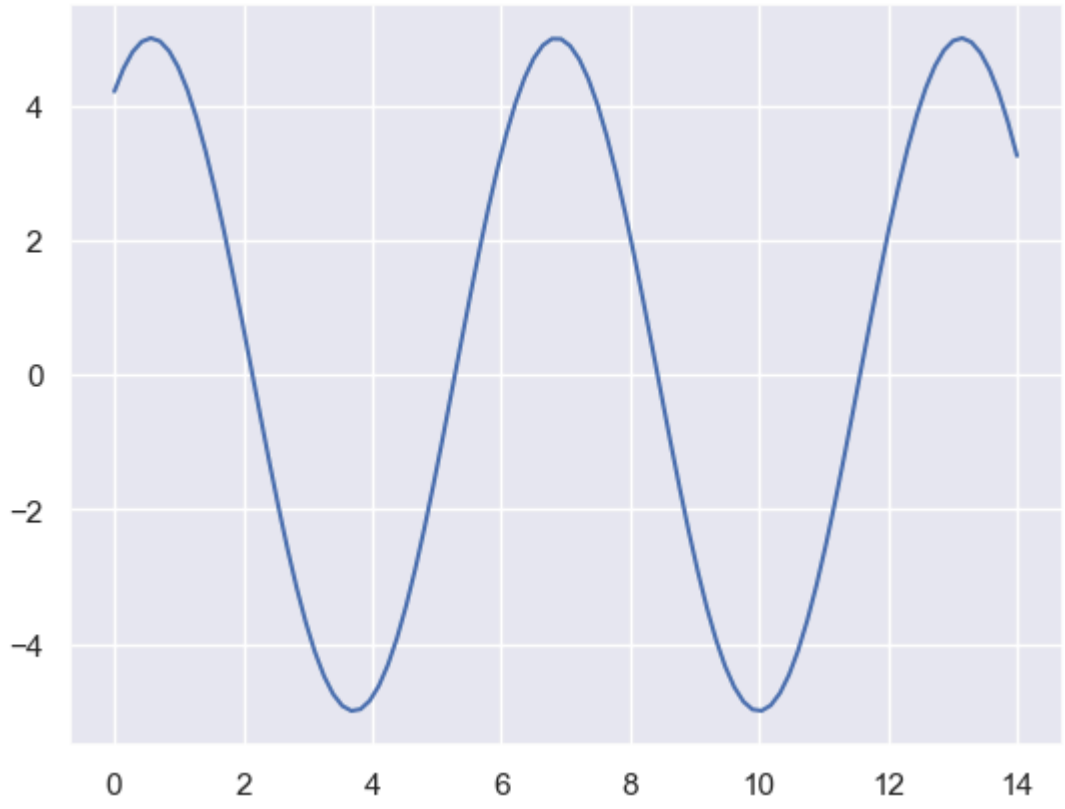

2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
2
4
6
8
10
12
14

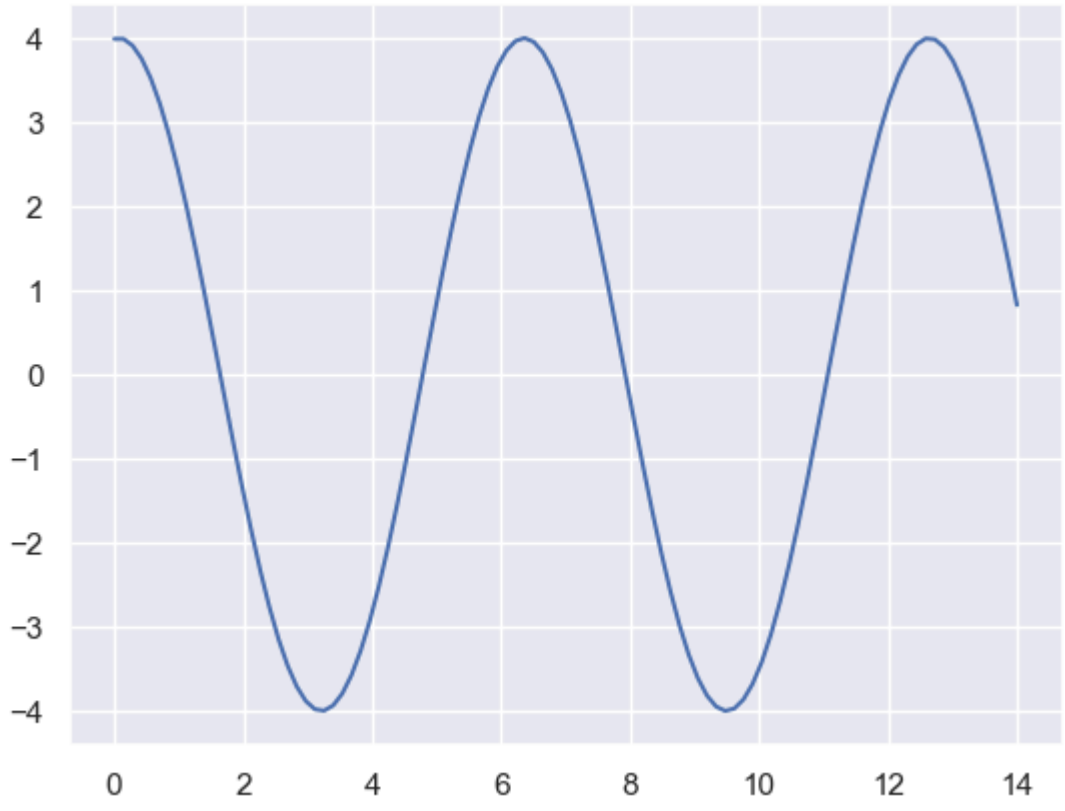


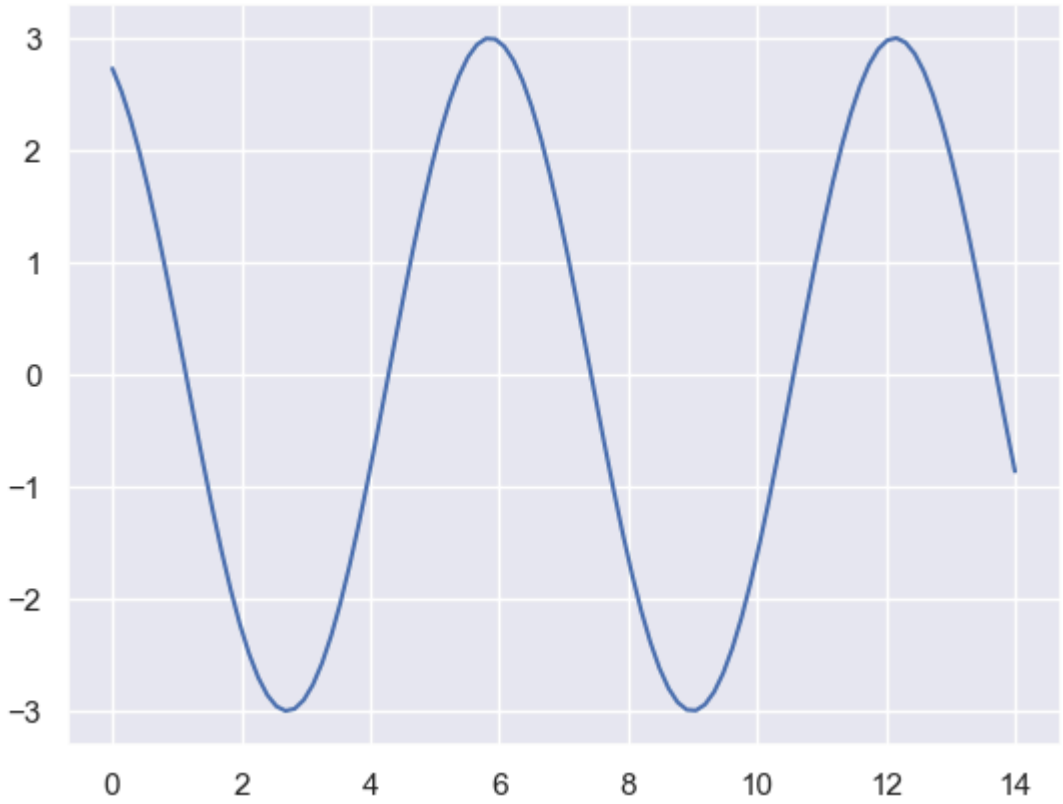
In [100…

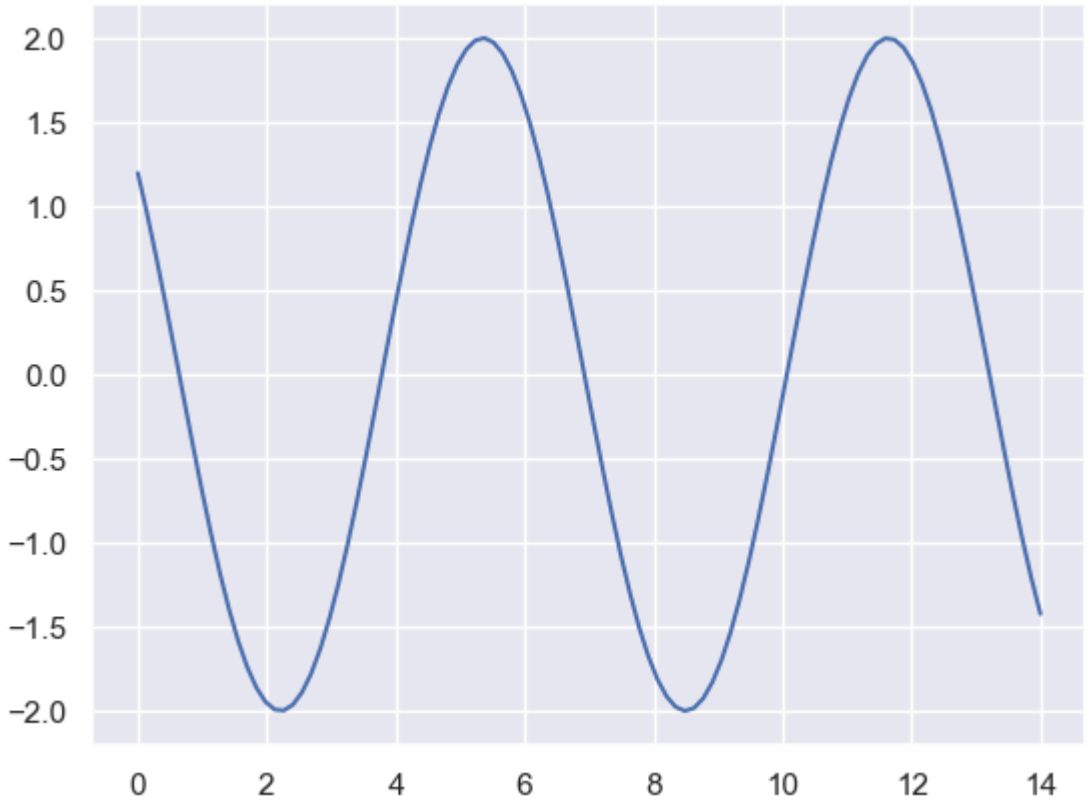
```
sns.set()
sinplot()
```

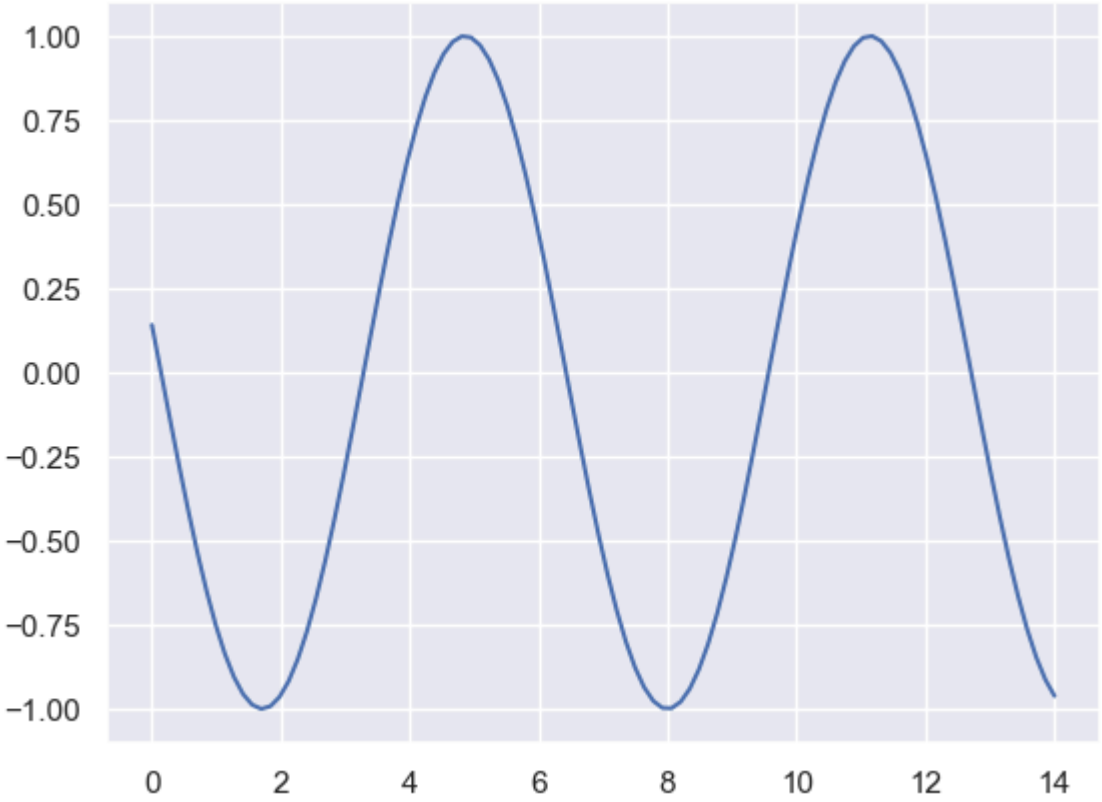



4
2
0
−2
−4
0
2
4
6
8
10
12
14


4
3
2
1
0
−1
−2
−3
−4
0
2
4
6
8
10
12
14

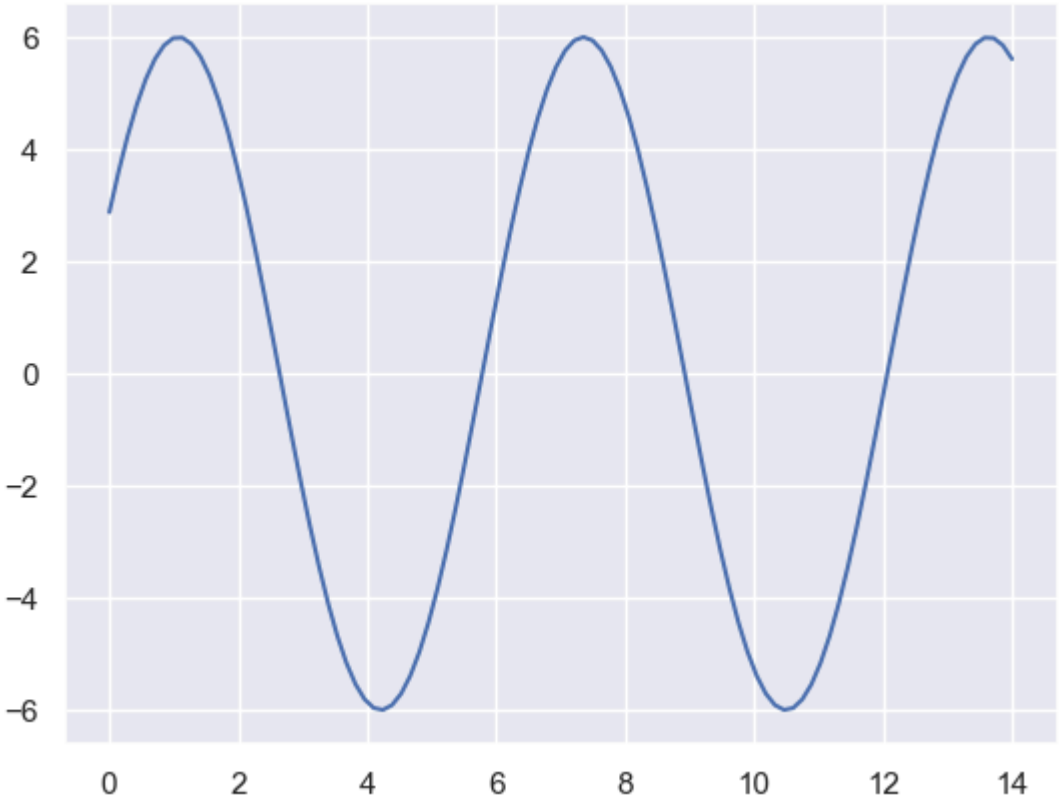

3
2
1
0
−1
−2
−3
0
2
4
6
8
10
12
14

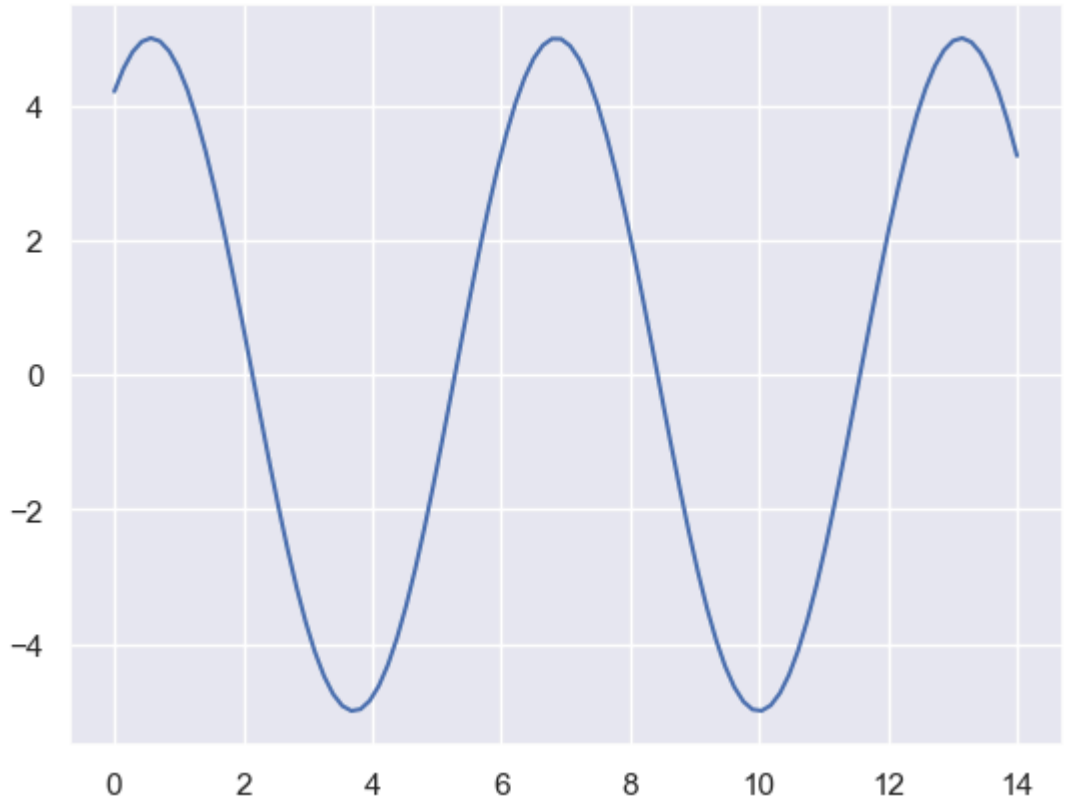

2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
2
4
6
8
10
12
14

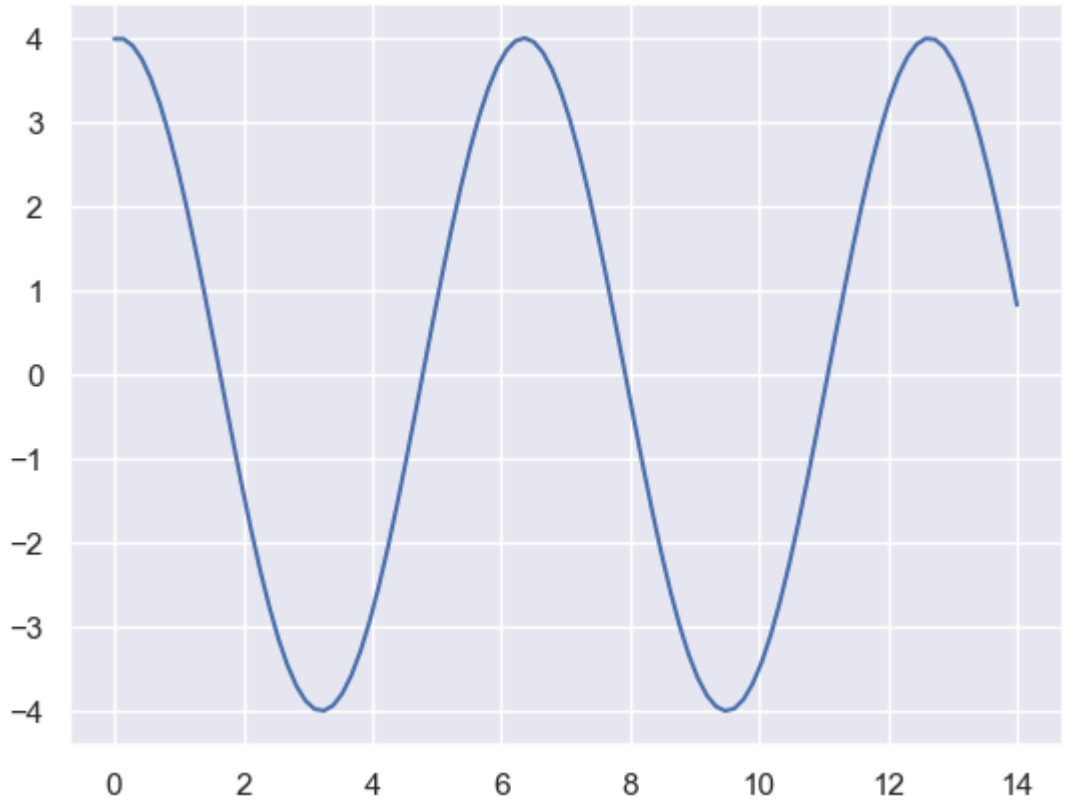


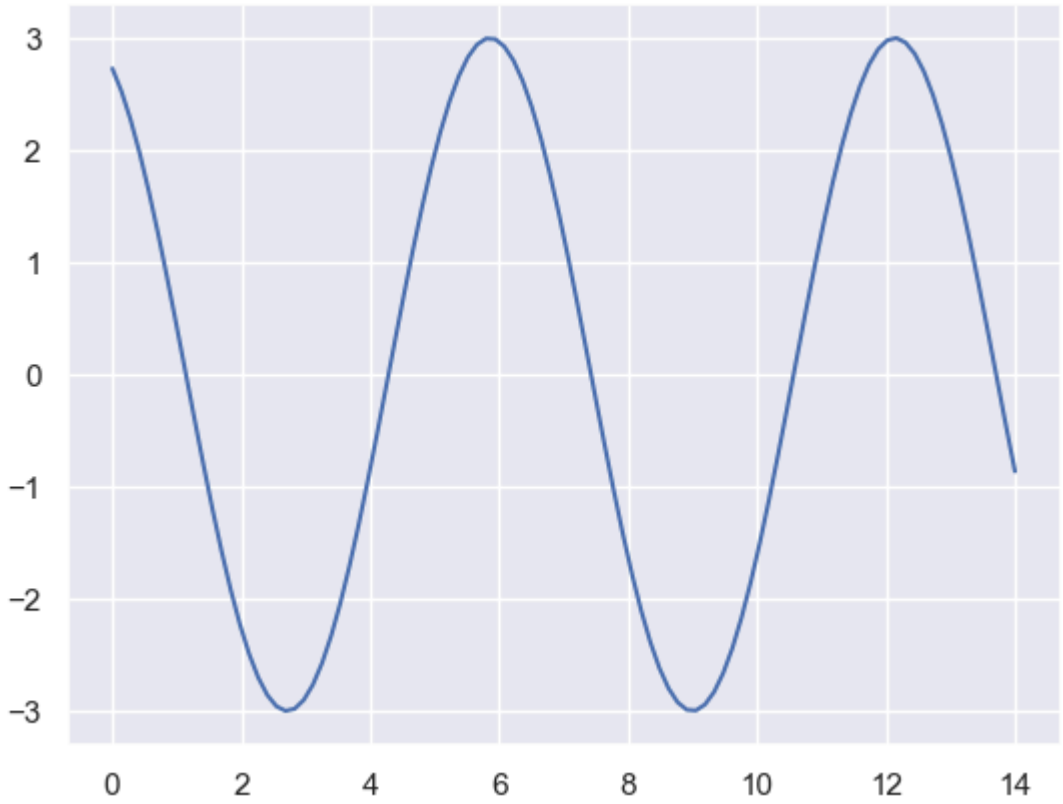
In [107…

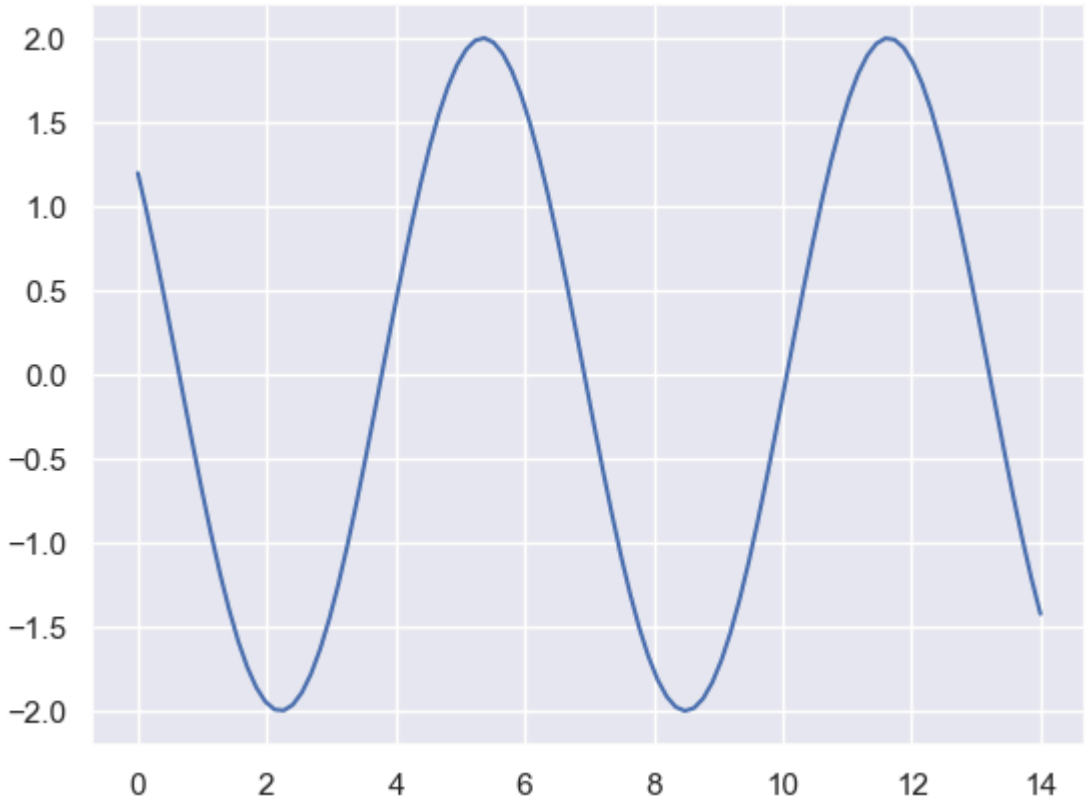
```
sns.set_style="whitegrid"
sinplot()
```

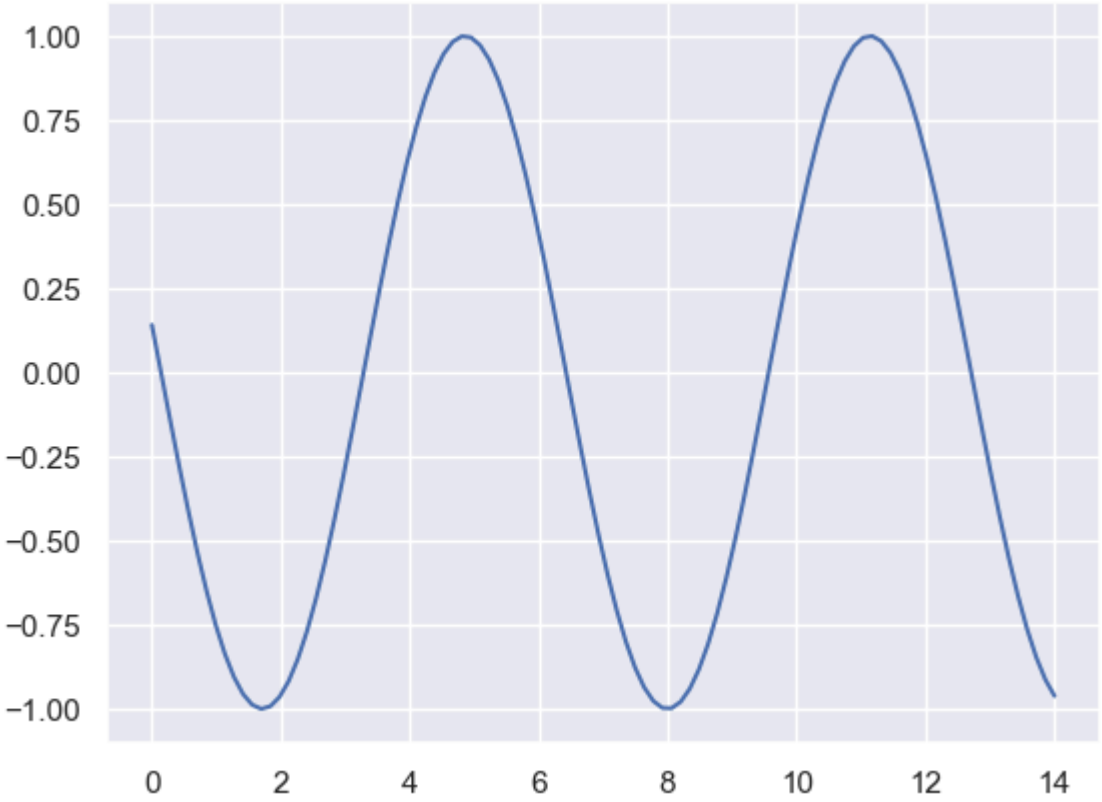



4
2
0
−2
−4
0
2
4
6
8
10
12
14


4
3
2
1
0
−1
−2
−3
−4
0
2
4
6
8
10
12
14

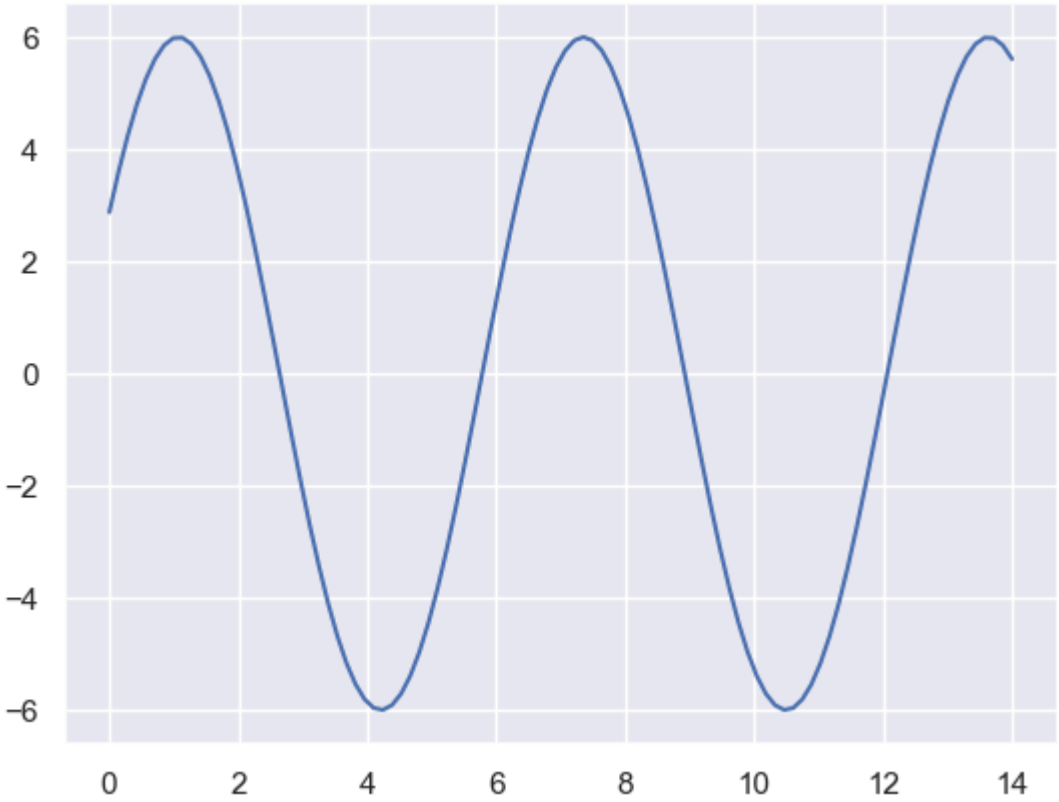

3
2
1
0
−1
−2
−3
0
2
4
6
8
10
12
14

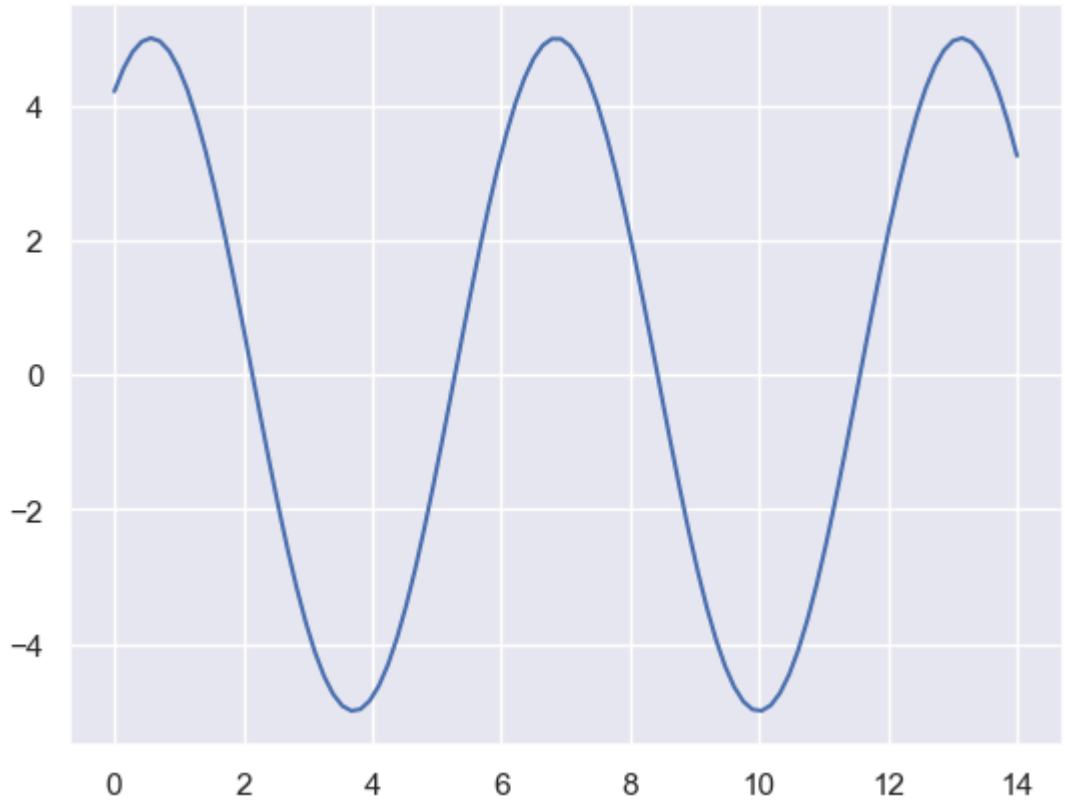

2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
2
4
6
8
10
12
14

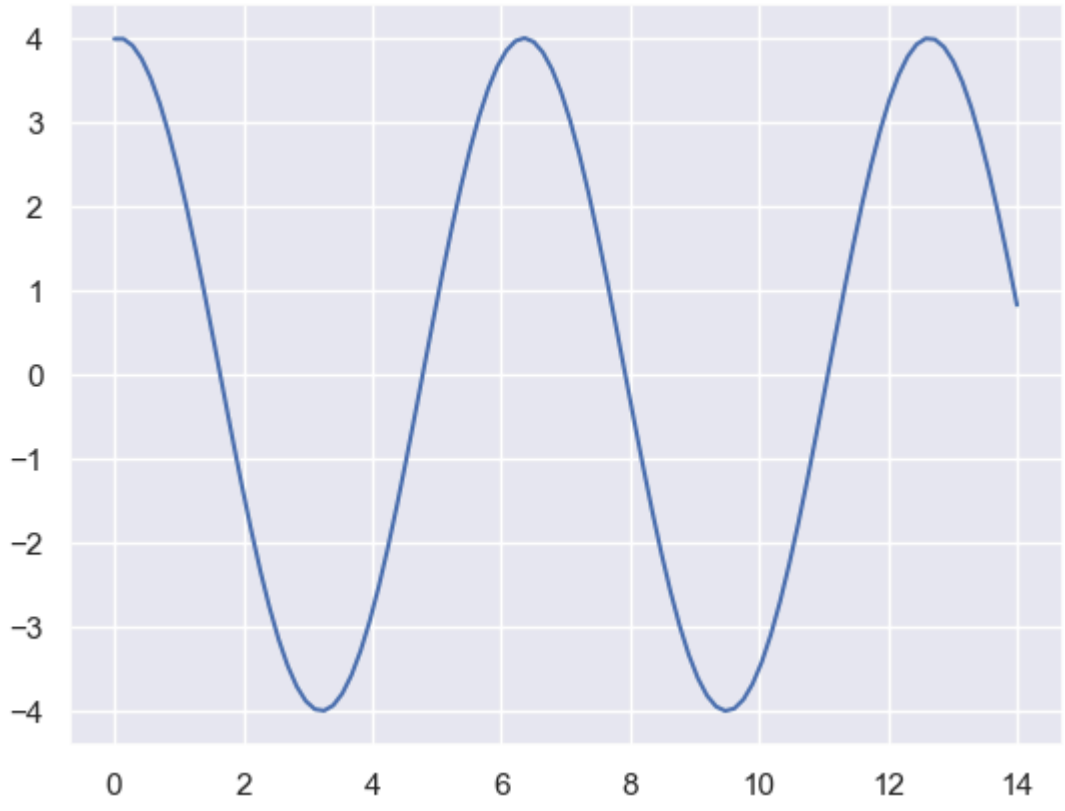


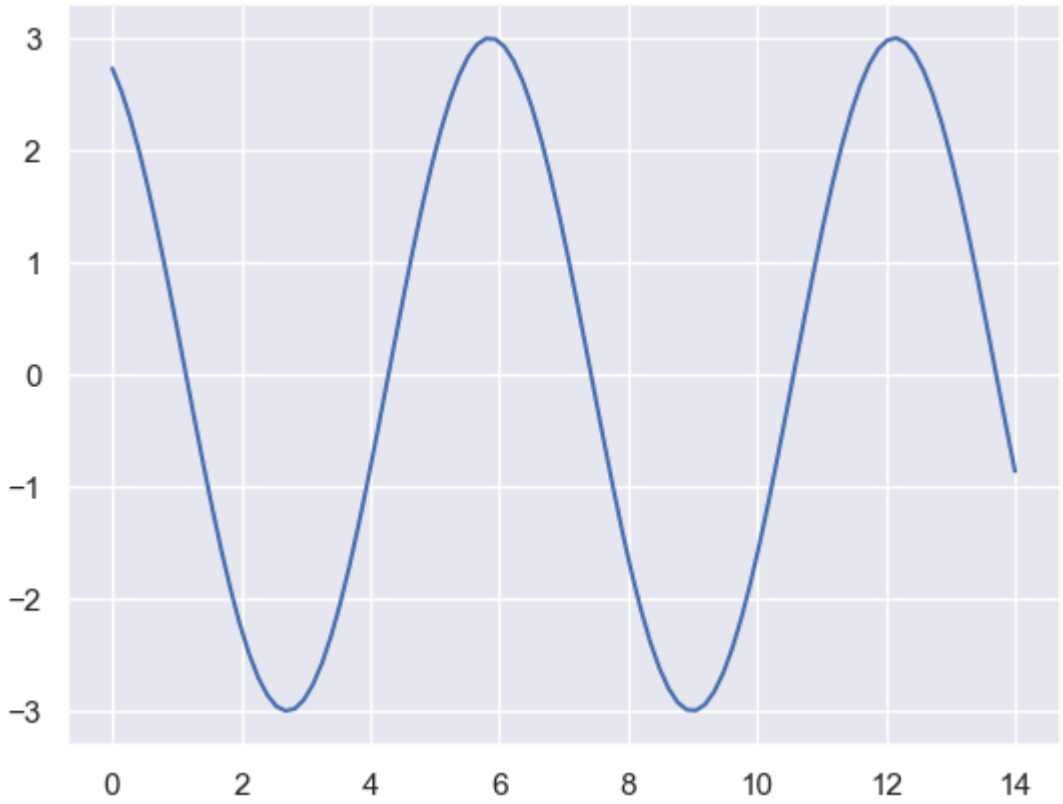
In [109…

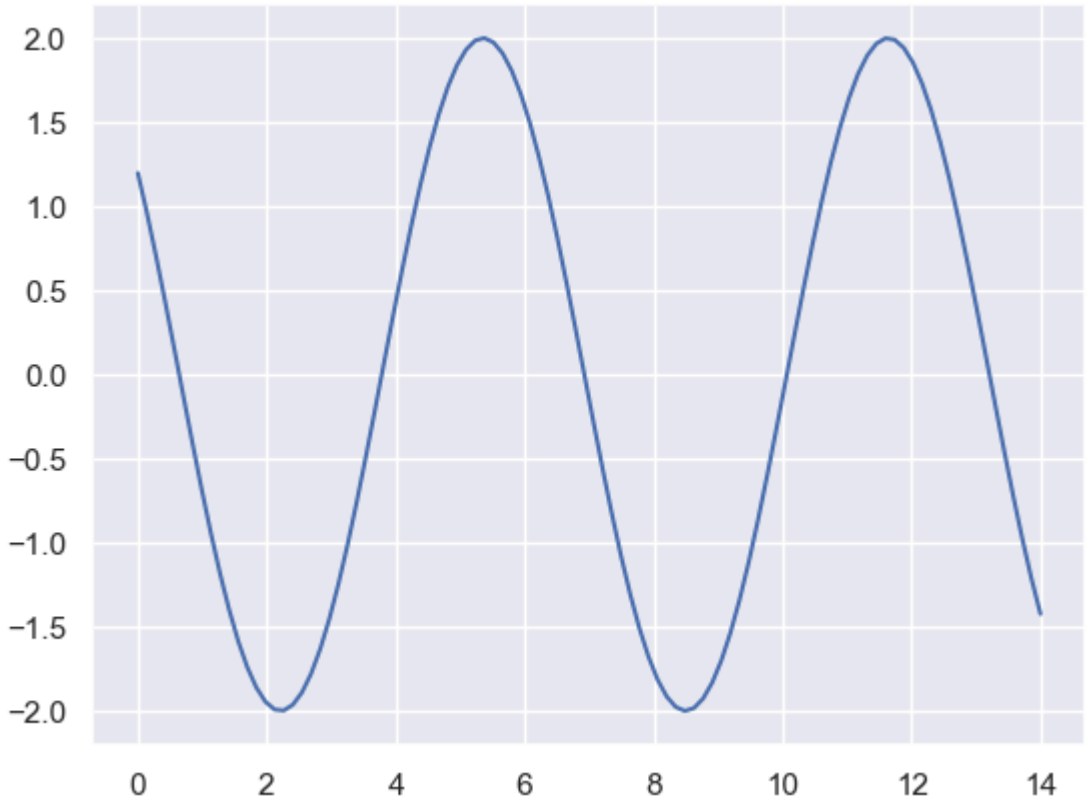
```
sns.set_style='dark'
sinplot()
```

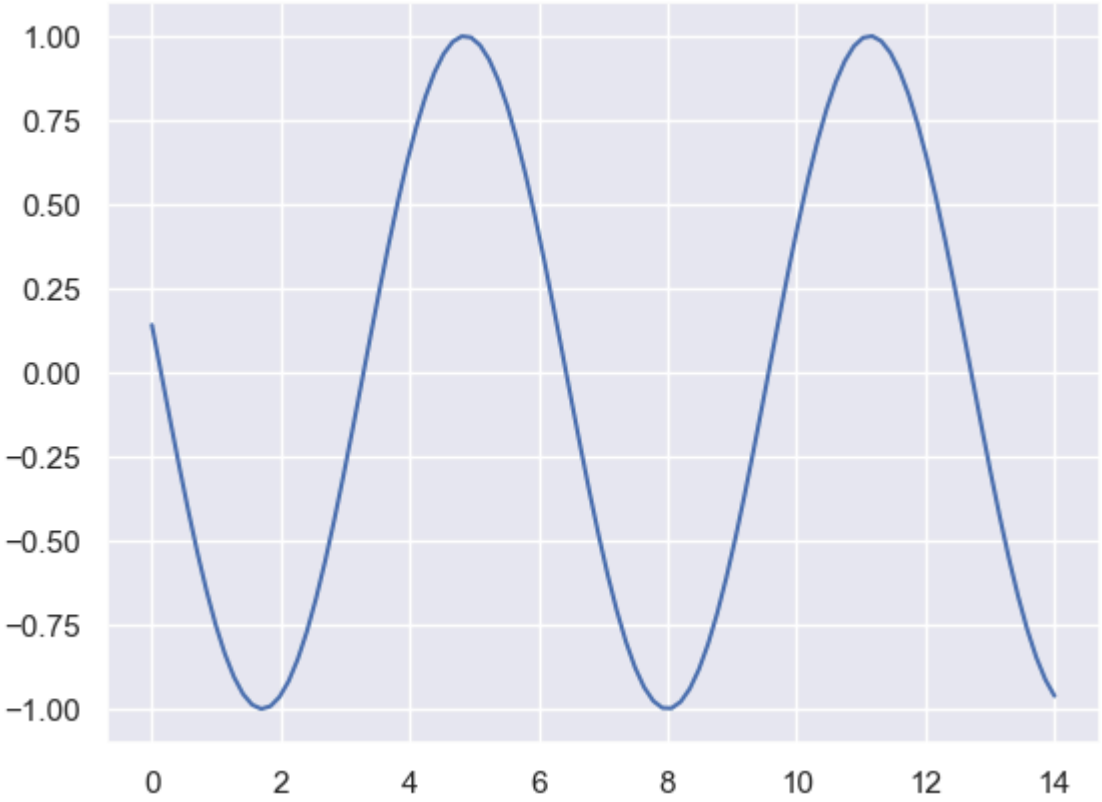



4
2
0
−2
−4
0
2
4
6
8
10
12
14


4
3
2
1
0
−1
−2
−3
−4
0
2
4
6
8
10
12
14

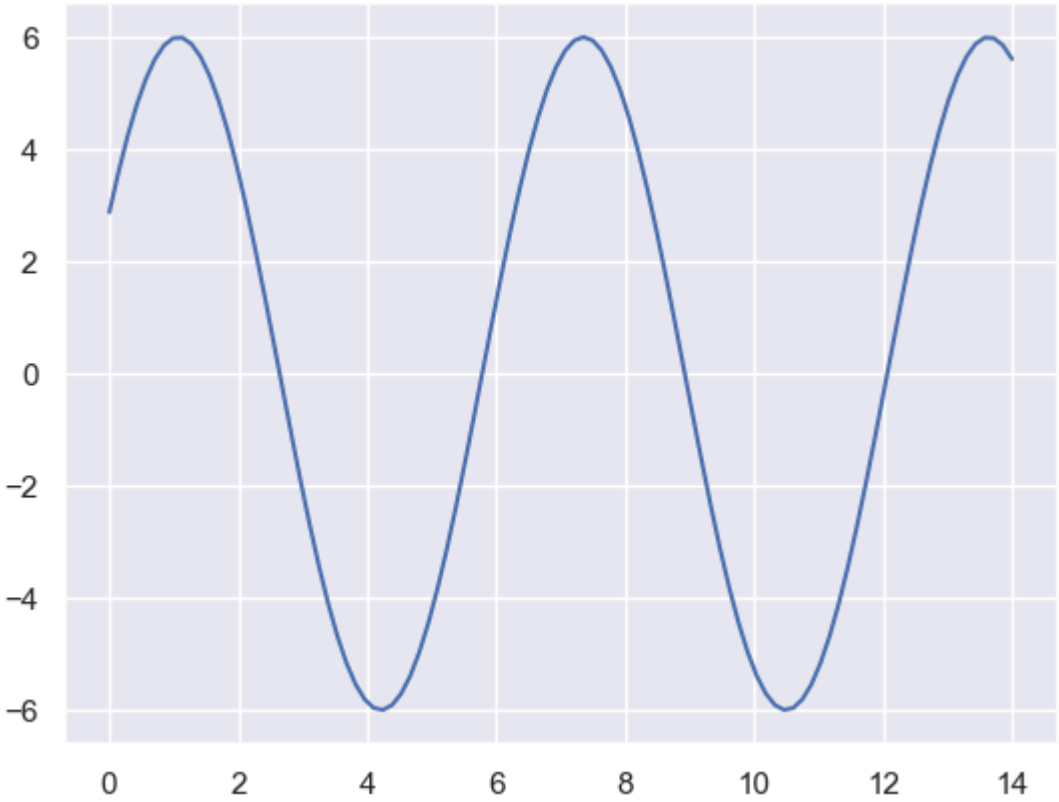

3
2
1
0
−1
−2
−3
0
2
4
6
8
10
12
14

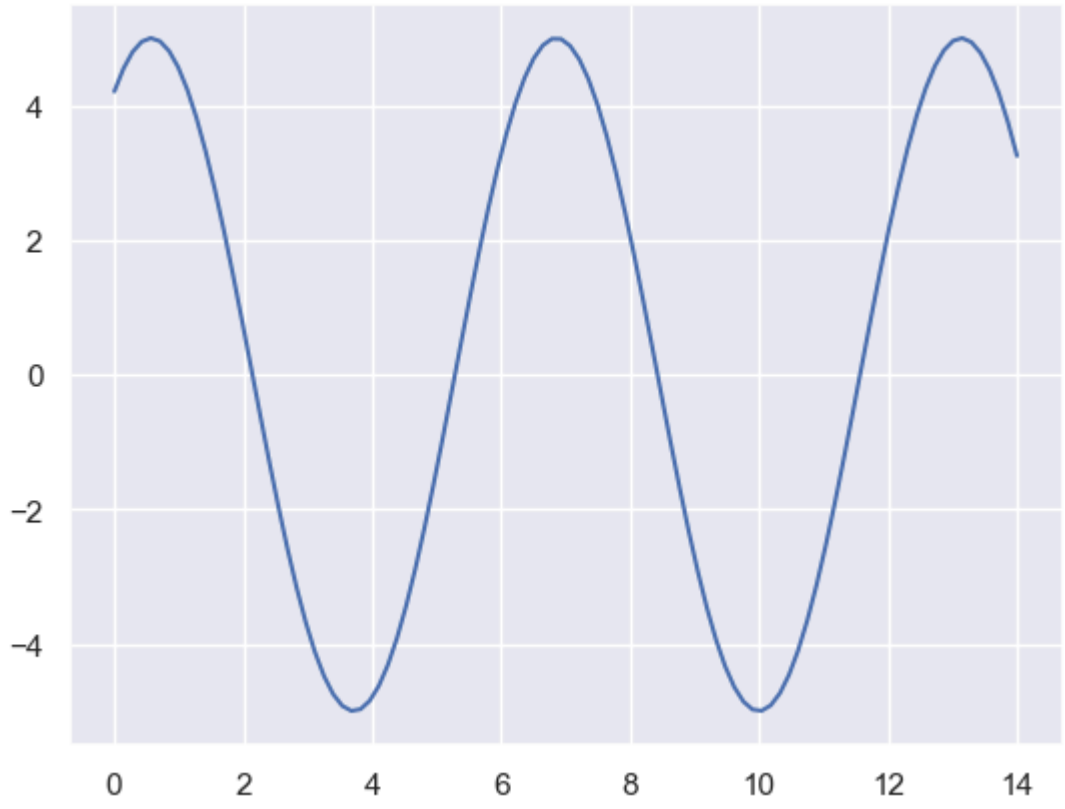

2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
2
4
6
8
10
12
14

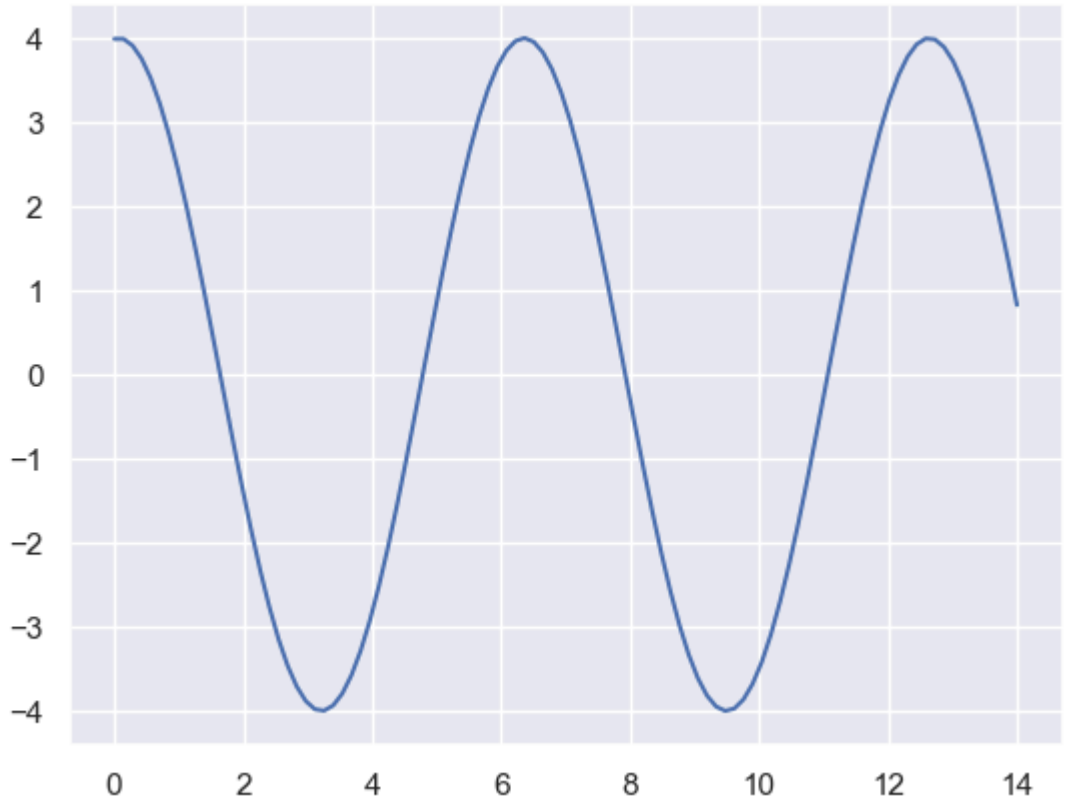


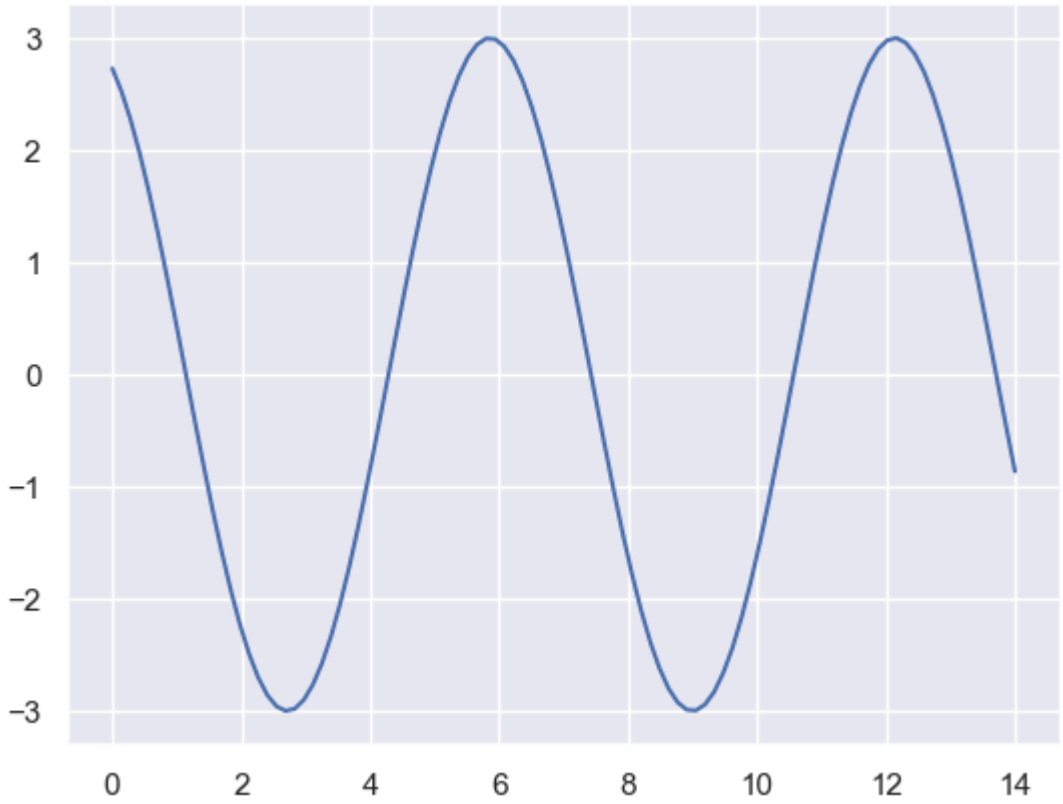
In [110…

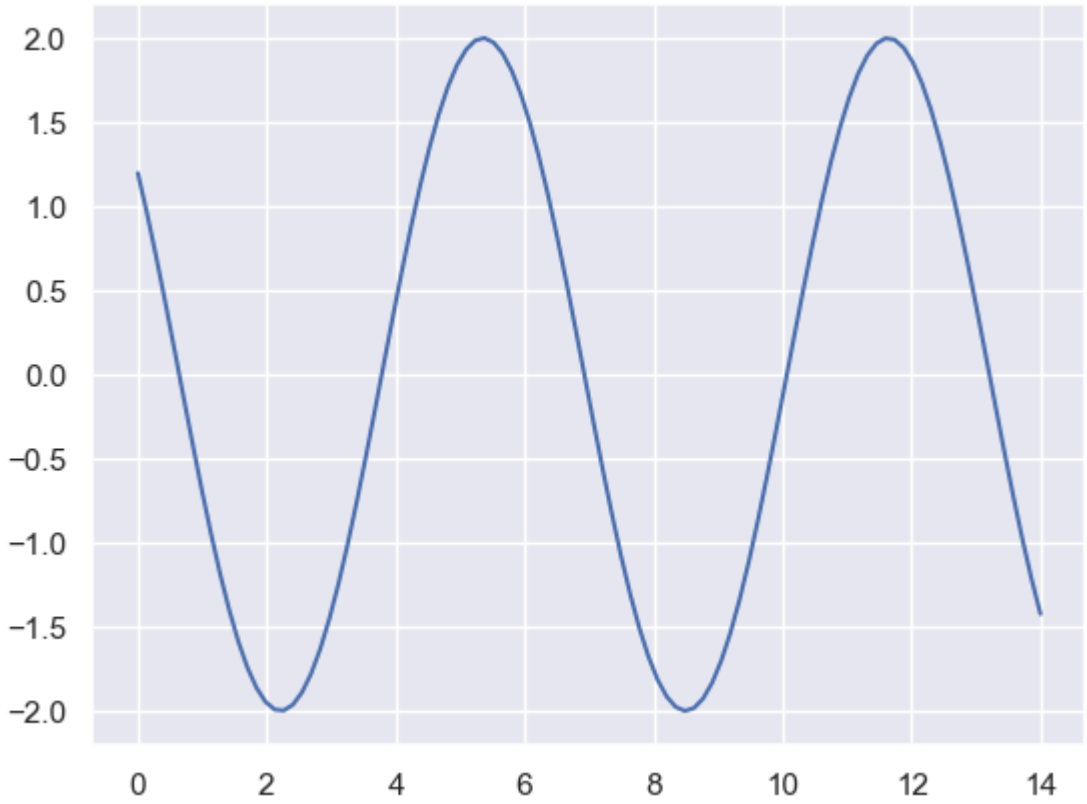
```
sns.set_style='white'
sinplot()
```

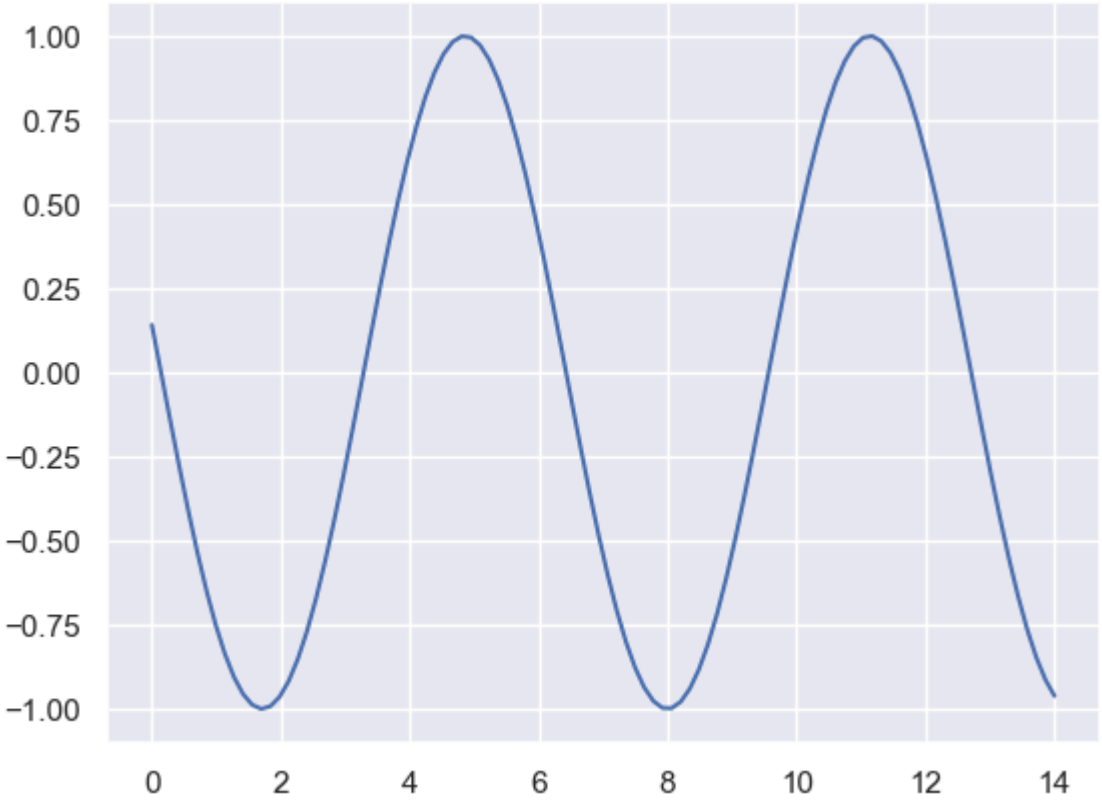



4
2
0
−2
−4
0
2
4
6
8
10
12
14


4
3
2
1
0
−1
−2
−3
−4
0
2
4
6
8
10
12
14

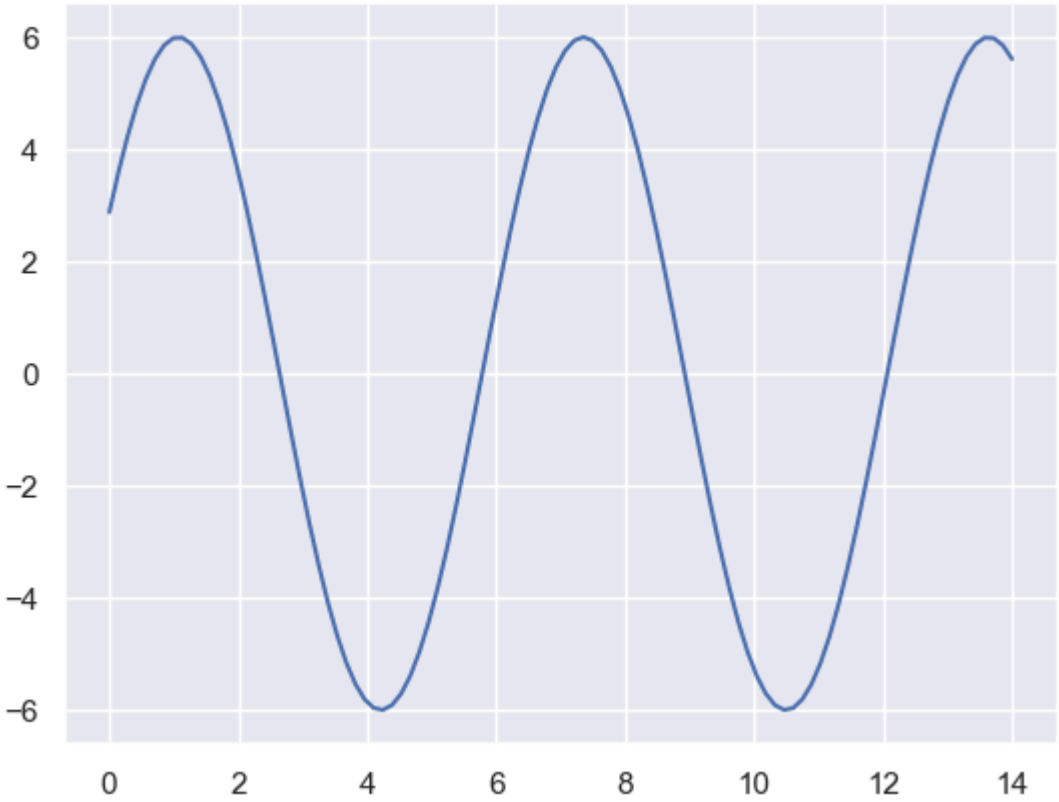

3
2
1
0
−1
−2
−3
0
2
4
6
8
10
12
14

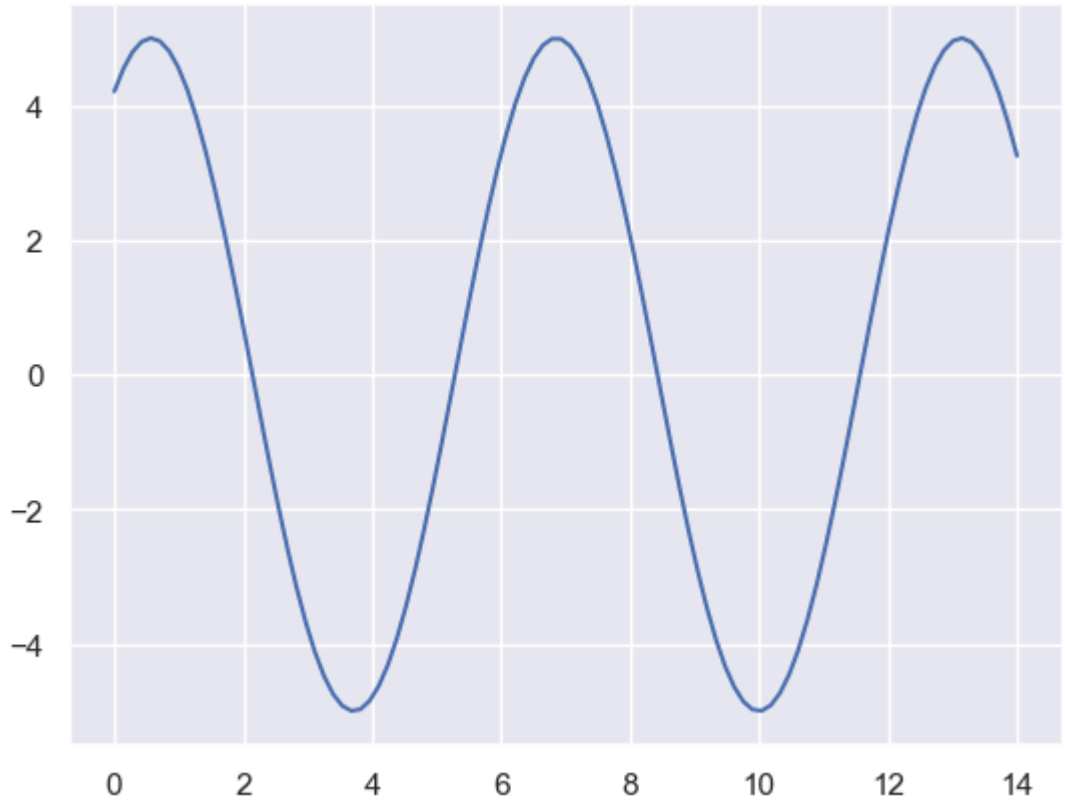

2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
2
4
6
8
10
12
14

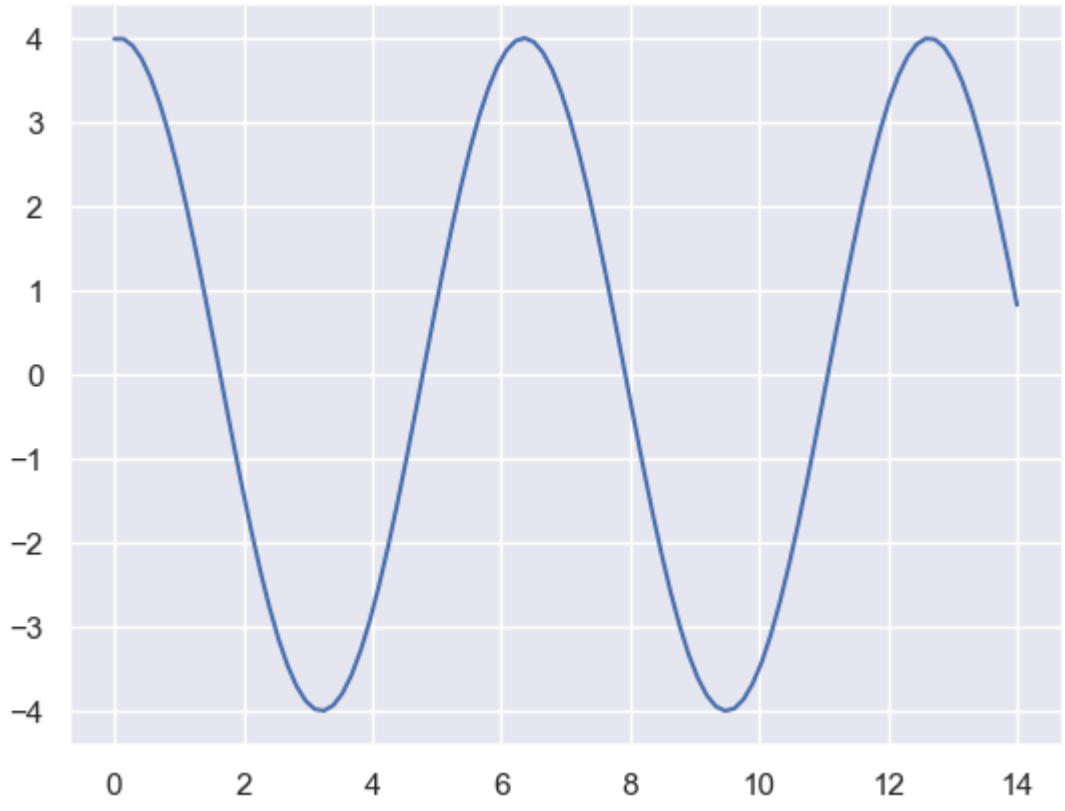


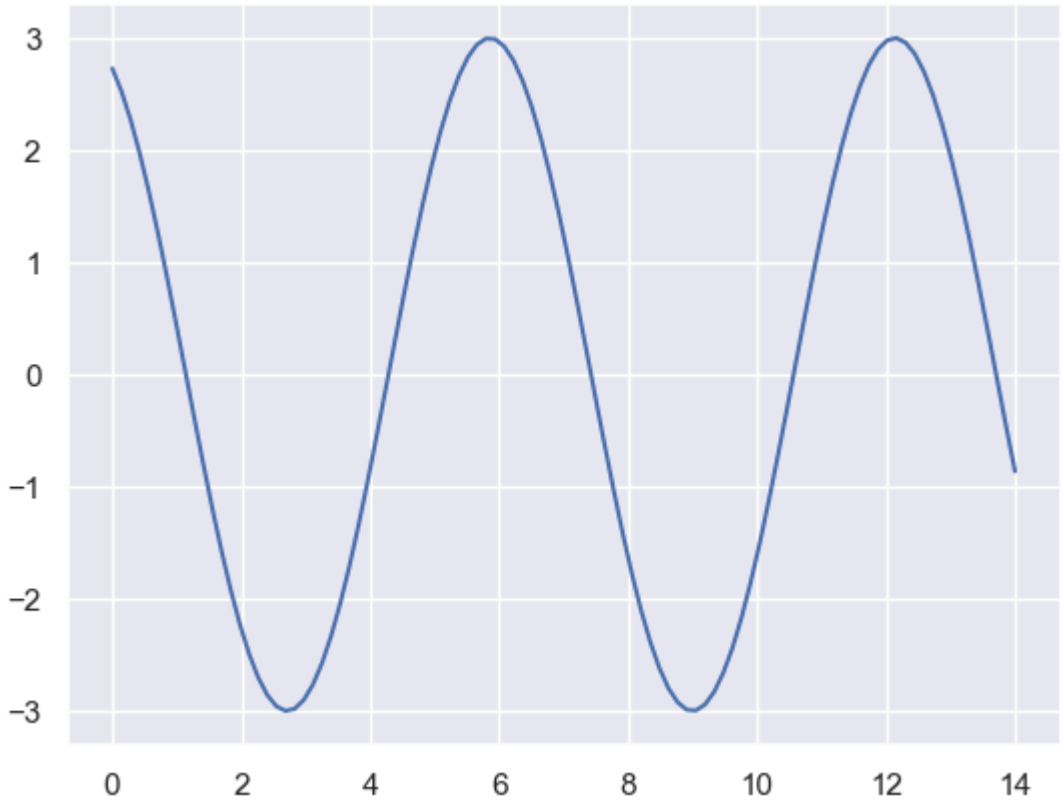
In [111…

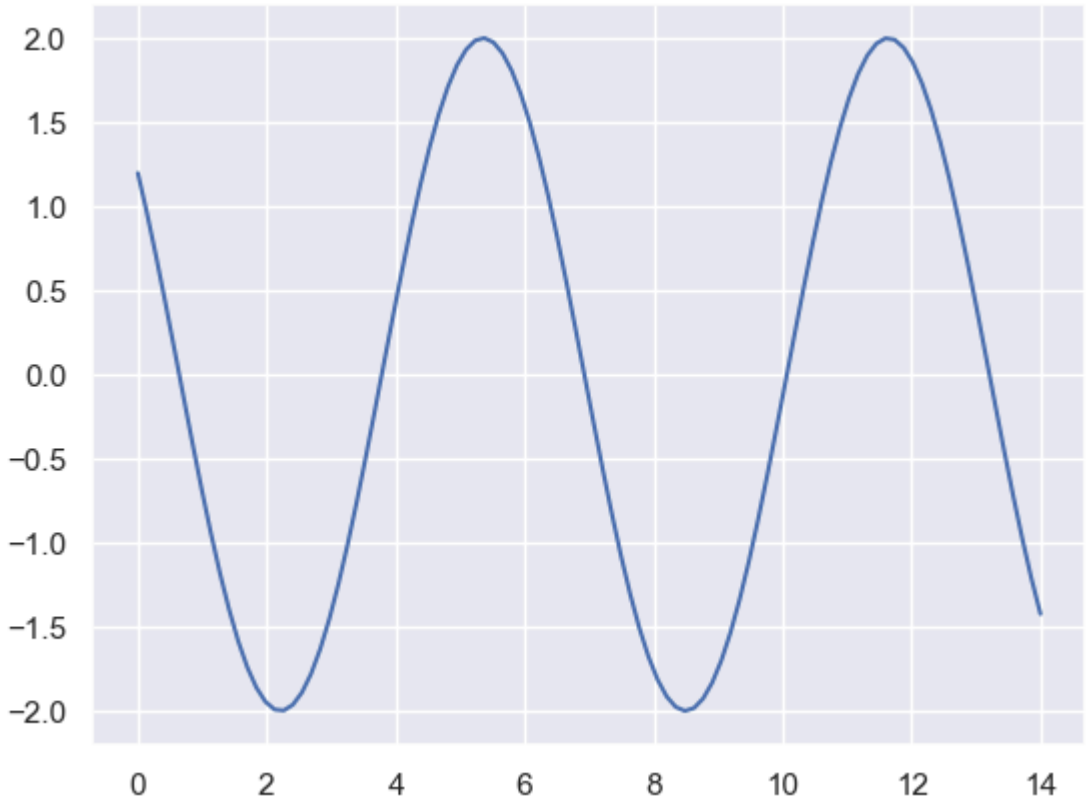
```
sns.set_style='ticks'
sinplot()
```

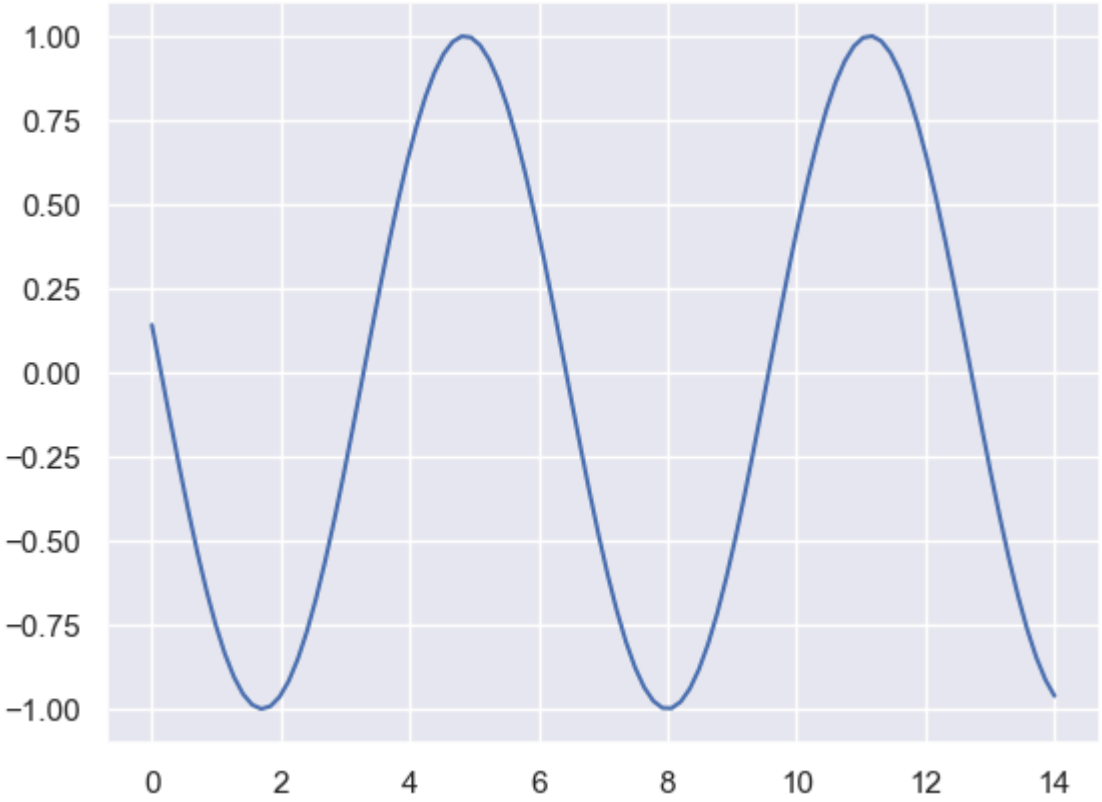



4
2
0
−2
−4
0
2
4
6
8
10
12
14


4
3
2
1
0
−1
−2
−3
−4
0
2
4
6
8
10
12
14


3
2
1
0
−1
−2
−3
0
2
4
6
8
10
12
14


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
2
4
6
8
10
12
14



we have finally completed analysis on FIFA Sports

In [ ]: