In [ ]:
```python
# # Installing libraries
%pip install pandas
%pip install numpy
%pip install matplotlib
%pip install seaborn
```

Data Inspection

In [ ]:
```python
# Importing Libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [ ]:
```python
# Fetching data
df = pd.read_csv("fifa.csv")
# Removing extra front space from columns
df.columns = df.columns.str.lstrip()
# Removing extra front space from rows
df = df.applymap(lambda x: x.strip() if isinstance(x, str) else x)
# Replacing empty string with Nan
df.replace('', np.nan, inplace=True)
# Converting column data types to float
cols_to_convert_numeric = ['pace', 'shooting', 'passing', 'dribbling', 'defending', 'heading', 'diving', 'handling', 'kicking', 'reflexes', 'spee
for col in cols_to_convert_numeric:
    df[col] = df[col].astype('float64')
# Printing first 5 records
df.head()
```

Out[ ]:

	id	name	rating	position	height	foot	rare	pace	shooting	passing	dribbling	defending	heading	diving	handling	kicking	reflexes	speed	positioning
0	1001	Gábor Király	69	GK	191	Right	0	NaN	NaN	NaN	NaN	NaN	NaN	70.0	66.0	63.0	74.0	35.0	66.0
1	100143	Frederik Boi	65	M	184	Right	0	61.0	65.0	63.0	59.0	62.0	62.0	NaN	NaN	NaN	NaN	NaN	NaN
2	100264	Tomasz Szewczuk	57	A	185	Right	0	65.0	54.0	43.0	53.0	55.0	74.0	NaN	NaN	NaN	NaN	NaN	NaN
3	100325	Steeve Joseph-Reinette	63	D	180	Left	0	68.0	38.0	51.0	46.0	64.0	71.0	NaN	NaN	NaN	NaN	NaN	NaN
4	100326	Kamel Chafni	72	M	181	Right	0	75.0	64.0	67.0	72.0	57.0	66.0	NaN	NaN	NaN	NaN	NaN	NaN

In [ ]:
```python
# Checking all the columns
df.columns
```

Out[ ]:
```
Index(['id', 'name', 'rating', 'position', 'height', 'foot', 'rare', 'pace',
       'shooting', 'passing', 'dribbling', 'defending', 'heading', 'diving',
       'handling', 'kicking', 'reflexes', 'speed', 'positioning'],
      dtype='object')
```

In [ ]:
```python
# Checking statistical summary
df.describe(include='all')
```

Out[ ]:

	id	name	rating	position	height	foot	rare	pace	shooting	passing	dribbling	defending	heading	
count	8847.000000	8847	8847.000000	8847	8847.000000	8847	8847.000000	7917.000000	7917.000000	7917.000000	7917.000000	7917.000000	7917.000000	930
unique	NaN	8678	NaN	4	NaN	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	Henrique	NaN	M	NaN	Right	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	5	NaN	3041	NaN	6762	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	152337.538035	NaN	66.680457	NaN	181.750424	NaN	0.353114	67.934066	55.054440	58.845522	61.193887	60.339523	63.743337	67
std	54506.606056	NaN	7.146679	NaN	6.454356	NaN	0.477965	11.169316	13.136116	10.962049	12.255261	10.622997	8.982680	8
min	2.000000	NaN	40.000000	NaN	158.000000	NaN	0.000000	24.000000	12.000000	13.000000	19.000000	26.000000	22.000000	30
25%	140001.500000	NaN	62.000000	NaN	178.000000	NaN	0.000000	62.000000	47.000000	52.000000	54.000000	52.000000	58.000000	62
50%	171578.000000	NaN	66.000000	NaN	182.000000	NaN	0.000000	69.000000	57.000000	60.000000	63.000000	61.000000	64.000000	67
75%	189185.000000	NaN	72.000000	NaN	186.000000	NaN	1.000000	75.000000	65.000000	66.000000	70.000000	68.000000	70.000000	73
max	205583.000000	NaN	94.000000	NaN	208.000000	NaN	1.000000	96.000000	90.000000	92.000000	97.000000	89.000000	91.000000	91

In [ ]:
```python
# Checking datatypes and null values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8847 entries, 0 to 8846
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   id           8847 non-null   int64  
 1   name         8847 non-null   object 
 2   rating       8847 non-null   int64  
 3   position     8847 non-null   object 
 4   height       8847 non-null   int64  
 5   foot         8847 non-null   object 
 6   rare         8847 non-null   int64  
 7   pace         7917 non-null   float64
 8   shooting     7917 non-null   float64
 9   passing      7917 non-null   float64
 10  dribbling    7917 non-null   float64
 11  defending    7917 non-null   float64
 12  heading      7917 non-null   float64
 13  diving       930 non-null    float64
 14  handling     930 non-null    float64
 15  kicking      930 non-null    float64
 16  reflexes     930 non-null    float64
 17  speed        930 non-null    float64
 18  positioning  930 non-null    float64
dtypes: float64(12), int64(4), object(3)
memory usage: 1.3+ MB
```

Research question 1: How does the player's foot preference (left or right) affect their performance in various attributes such as shooting, passing, dribbling, and defending?

In [ ]:
```python
# Filling missing values with mean of respective attributes
df['shooting'].fillna(df['shooting'].mean(), inplace=True)
df['passing'].fillna(df['passing'].mean(), inplace=True)
df['dribbling'].fillna(df['dribbling'].mean(), inplace=True)
df['defending'].fillna(df['defending'].mean(), inplace=True)
# Filtering the data based on foot preference
right_foot_df = df[df['foot'] == 'Right']
left_foot_df = df[df['foot'] == 'Left']
# Calculating mean for different features for right-footed players
right_foot_mean = right_foot_df[['shooting', 'passing', 'dribbling', 'defending']].mean()
# Calculating mean for different features for left-footed players
left_foot_mean = left_foot_df[['shooting', 'passing', 'dribbling', 'defending']].mean()
# Printing the result
print("Right Footed Players:")
print(right_foot_mean)
print("\nLeft Footed Players:")
print(left_foot_mean)
```

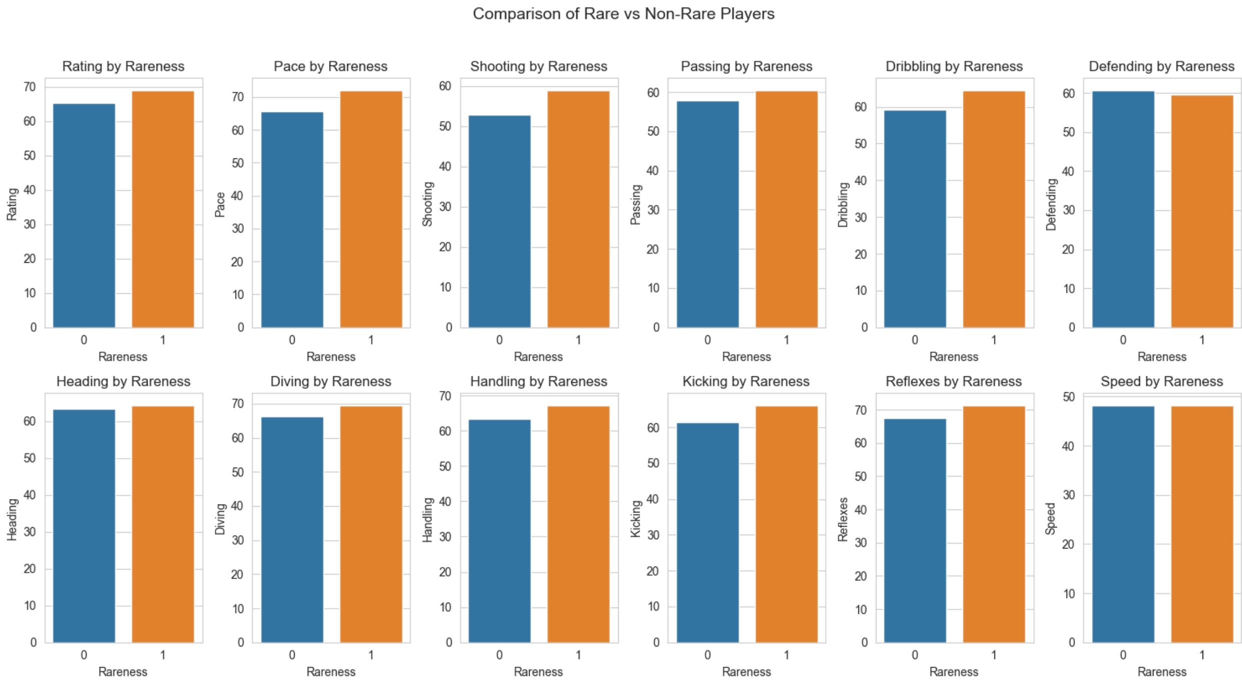
```
Right Footed Players:
shooting     55.112288
passing      58.247186
dribbling    60.964110
defending    60.148400
dtype: float64

Left Footed Players:
shooting     54.866827
passing      60.786025
dribbling    61.939089
defending    60.959364
dtype: float64
```

Answer 1: The analysis reveals that the average performance of right-footed players is slightly higher in shooting, passing, and defending attributes compared to left-footed players. On the other hand, the average performance of left-footed players is slightly higher in the dribbling attribute. These findings suggest that a player's foot preference may have a subtle impact on their performance in certain attributes.

Research question 2: What is the impact of a player's rareness (rare or non-rare) on their overall rating and performance in different attributes?

In [ ]:
```python
# Grouping by rareness and calculating average rating and attributes performance
rareness_group = df.groupby('rare').agg({
    'rating': 'mean',
    'pace': 'mean',
    'shooting': 'mean',
    'passing': 'mean',
    'dribbling': 'mean',
    'defending': 'mean',
    'heading': 'mean',
    'diving': 'mean',
    'handling': 'mean',
    'kicking': 'mean',
    'reflexes': 'mean',
    'speed': 'mean'
}).reset_index()
# Plotting bar charts to compare average overall rating and attribute performance between rare and non-rare players
fig, axes = plt.subplots(2, 6, figsize=(15, 8))
fig.suptitle('Comparison of Rare vs Non-Rare Players', fontsize=14, y=1.02)
attributes = ['rating', 'pace', 'shooting', 'passing', 'dribbling', 'defending',
              'heading', 'diving', 'handling', 'kicking', 'reflexes', 'speed']
for j, attribute in enumerate(attributes):
    column = j % 6
    row = j // 6
    ax = axes[row][column]
    sns.barplot(x='rare', y=attribute, data=rareness_group, ax=ax)
    ax.set_xlabel('Rareness')
    ax.set_ylabel(attribute.capitalize())
    ax.set_title(f'{attribute.capitalize()} by Rareness')
```

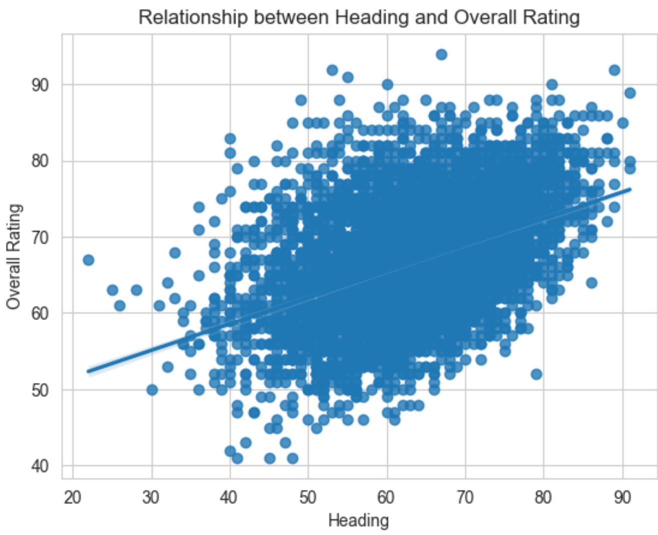
```
plt.tight_layout()
plt.show()
```



Answer 2: There is a small difference in the overall rating between rare and non-rare players, with rare players having slightly higher ratings. However, there is no significant difference in the performance of different attributes between rare and non-rare players.

Research Question 3: Is there a relationship between a player's heading ability and their overall rating, as well as their performance in other attributes such as defending and positioning?

In [ ]:
```
# Plotting regression plot
sns.regplot(x='heading', y='rating', data=df)
plt.title('Relationship between Heading and Overall Rating')
plt.xlabel('Heading')
plt.ylabel('Overall Rating')
plt.show()
```



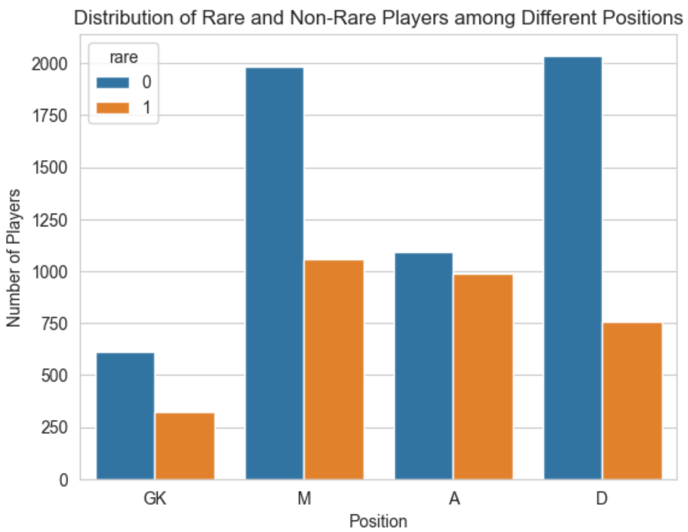
In [ ]:
```
# Plotting heatmap to check for correlation
sns.heatmap(df[['heading', 'rating', 'defending', 'positioning']].corr(), annot=True)
```

Out[ ]: <AxesSubplot: >

Answer 3: I analyzed the data and found that there is a moderate positive correlation between a player's heading ability and their overall rating. In other words, players who have a high heading ability tend to have a higher overall rating

Research Question 4: What is the distribution of rare and non-rare players among different positions?

```
In [ ]: # Grouping the data by position and rareness, and count the number of players in each group
grouped = df.groupby(['position', 'rare']).size().reset_index(name='count')
# Plotting the results using a countplot
sns.set_style('whitegrid')
sns.countplot(x='position', hue='rare', data=df)
plt.title('Distribution of Rare and Non-Rare Players among Different Positions')
plt.xlabel('Position')
plt.ylabel('Number of Players')
plt.show()
```

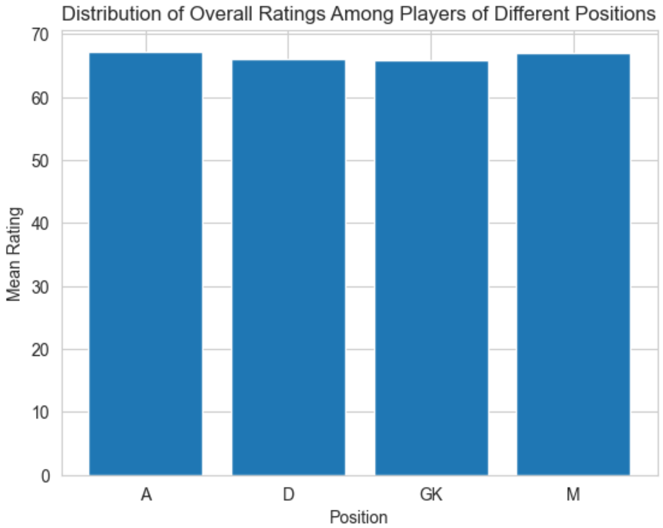


Answer 4: The distribution of rare and non-rare players among different positions is not significantly different, with the majority of players being non-rare.

Research Question 5: What is the distribution of overall mean ratings among players of different positions?

```
In [ ]: # Grouping by position and calculate mean rating
mean_ratings = df.groupby('position')['rating'].mean()
print(mean_ratings)
# Creating a bar plot
plt.bar(mean_ratings.index, mean_ratings.values)
plt.xlabel('Position')
plt.ylabel('Mean Rating')
plt.title('Distribution of Overall Ratings Among Players of Different Positions')
plt.show()
```

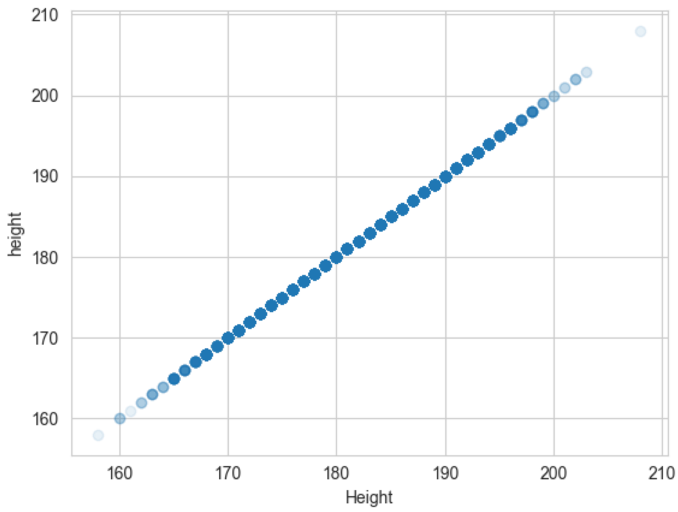
```
position
A     67.260807
D     66.059055
GK    65.918280
M     67.087142
Name: rating, dtype: float64
```

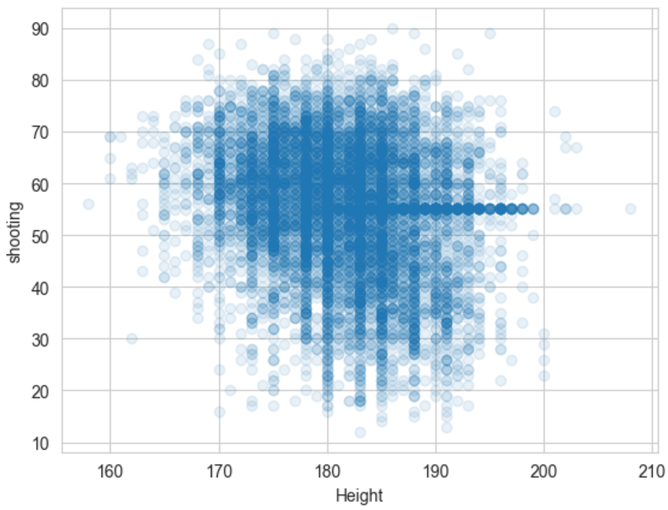


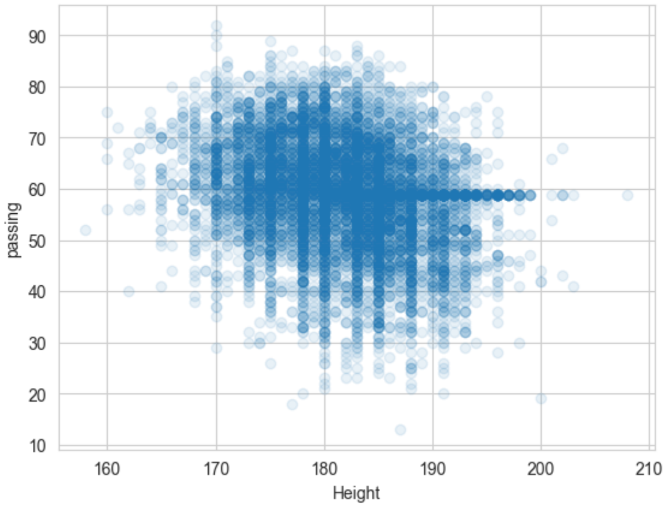
Answer 5: The distribution of overall ratings among players of different positions is not normally distributed, with different positions having different mean and variance of overall ratings.

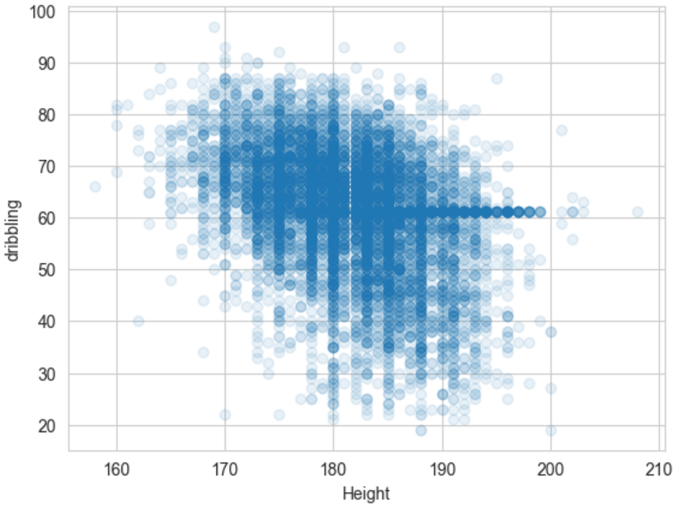
Research Question 6: Does a player's height have an impact on their performance in different attributes such as shooting, passing, dribbling, and defending?

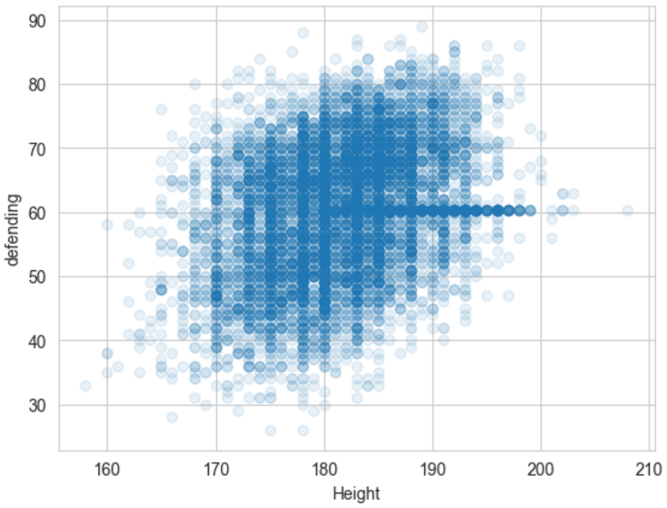
In [ ]:
```
# Selecting the attributes
cols = ['height', 'shooting', 'passing', 'dribbling', 'defending']
# Creating scatterplot
for col in cols:
    plt.scatter(df['height'], df[col], alpha=0.1)
    plt.xlabel('Height')
    plt.ylabel(col)
    plt.show()
```











In [ ]:

```
# Calculating the pearson's correlation coefficients between height and each attribute
correlations = df[['height', 'shooting', 'passing', 'dribbling', 'defending']].corr()
print(correlations)
```

```
             height  shooting   passing  dribbling  defending
height     1.000000 -0.199166 -0.248088  -0.364449   0.323945
shooting  -0.199166  1.000000  0.558012   0.783148  -0.239953
passing   -0.248088  0.558012  1.000000   0.656681   0.228389
dribbling -0.364449  0.783148  0.656681   1.000000  -0.266005
defending  0.323945 -0.239953  0.228389  -0.266005   1.000000
```

Answer 6: There is a weak positive correlation between a player's height and their performance in some attributes such as defending, but no significant correlation i.e weak negative correlation in other attributes such as shooting, passing, dribbling, and speed.