Dataset: Soccer database *Report prepared by:* **Mahmoud Albiali**

Assessing the contribution of players' heights, and their preferred foot of play on defensive prowess and attacking effeciency in professional soccer.

Table of Contents

Introduction

In this report we are exploring on two general conceptions. First is that players' height may affect their defensive abilities, where taller players might have an aerial advantage with headers defending volley balls, and also have a longer reach span which might cause them to perform sliding tackles and complete interceptions more successfully.

The second is that left-footed players are perceived to be more talented when it comes to ball control,dribbling and attacking efficiency.

In this brief analysis, we have extracted data from two tables in the soccer dataset.sqlite, "player" and "player_attributes", and formed one clean dataframe to perform the analysis and test the previously mentioned popular perceptions. we have used the following parameters as surrogates to gauge both of defensive prowess and attacking efficiency.

Defensive prowess surrogates (ratings on 0-100 scale):

- Heading accuracy.
- Interceptions.
- Standing tackles.
- Sliding tackles.

Attacking efficiency surrogates (ratings on 0-100 scale):

- Finishing.
- Dribbling.
- Ball control.

Data Wrangling

General Properties - loading relevant tables from soccer.sqlite dataset into dataframes and examining them.

In [40]:

```
# Importing relevant libraries

import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [41]:

```
#Loading player data table, saving it as CSV file, then loading it into a dataframe
# with player 'id' column as index.

con = sqlite3.connect("database.sqlite")
sql= """ SELECT * FROM Player; """
df_player = pd.read_sql_query(sql,con)
df_player.to_csv('player_data.csv')
df_p = pd.read_csv('player_data.csv', index_col= ['id'])
df_p.head()
```

Out[41]:

id	Unnamed: 0	player_api_id	player_name	player_fifa_api_id	birthday	height	weight
1	0	505942	Aaron Appindangoye	218353	1992-02-29 00:00:00	182.88	187
2	1	155782	Aaron Cresswell	189615	1989-12-15 00:00:00	170.18	146
3	2	162549	Aaron Doran	186170	1991-05-13 00:00:00	170.18	163
4	3	30572	Aaron Galindo	140161	1982-05-08 00:00:00	182.88	198
5	4	23780	Aaron Hughes	17725	1979-11-08 00:00:00	182.88	154

In the following four cells, player data dataframe (df_p) is being assessed, checking for proper loading, missing values or duplicates, also data types were checked to be proper for following wrangling and analysis.

In [42]:

```
df_p.shape
```

Out[42]:

```
(11060, 7)
```

In [43]:

```
df_p.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11060 entries, 1 to 11075
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          11060 non-null  int64  
 1   player_api_id       11060 non-null  int64  
 2   player_name         11060 non-null  object 
 3   player_fifa_api_id  11060 non-null  int64  
 4   birthday            11060 non-null  object 
 5   height              11060 non-null  float64
 6   weight              11060 non-null  int64  
dtypes: float64(1), int64(4), object(2)
memory usage: 691.2+ KB
```

In [44]:

```
df_p.duplicated().sum()
```

Out[44]:

```
0
```

In [45]:

```
df_p.isnull().sum()
```

Out[45]:

```
Unnamed: 0            0
player_api_id         0
player_name           0
player_fifa_api_id    0
birthday              0
height                0
weight                0
dtype: int64
```

In [46]:

```
# Loading player attributes data table, saving it as CSV file,
# then loading it into a dataframe 'df_pa' with player 'id' column as index.

con = sqlite3.connect("database.sqlite")
sql= """ SELECT * FROM Player_Attributes; """
df_player_attributes = pd.read_sql_query(sql,con)
df_player_attributes.to_csv('player_data_attributes.csv')
df_pa = pd.read_csv('player_data_attributes.csv', index_col= ['id'])
df_pa.head()
```

Out[46]:

id	Unnamed: 0	player_fifa_api_id	player_api_id	date	overall_rating	potential	preferred_foot
1	0	218353	505942	2016-02-18 00:00:00	67.0	71.0	right
2	1	218353	505942	2015-11-19 00:00:00	67.0	71.0	right
3	2	218353	505942	2015-09-21 00:00:00	62.0	66.0	right
4	3	218353	505942	2015-03-20 00:00:00	61.0	65.0	right
5	4	218353	505942	2007-02-22 00:00:00	61.0	65.0	right

5 rows × 42 columns

In the following four cells, player attributes data dataframe (df_pa) is being assessed, checking for proper loading, missing values or duplicates, also data types were checked to be proper for following wrangling and analysis.

In [47]:

```
df_pa.shape
```

Out[47]:

```
(183978, 42)
```

In [48]:

```
df_pa.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 183978 entries, 1 to 183978
Data columns (total 42 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Unnamed: 0            183978 non-null  int64  
 1   player_fifa_api_id    183978 non-null  int64  
 2   player_api_id         183978 non-null  int64  
 3   date                  183978 non-null  object 
 4   overall_rating        183142 non-null  float64
 5   potential             183142 non-null  float64
 6   preferred_foot        183142 non-null  object 
 7   attacking_work_rate   180748 non-null  object 
 8   defensive_work_rate   183142 non-null  object 
 9   crossing              183142 non-null  float64
 10  finishing             183142 non-null  float64
 11  heading_accuracy      183142 non-null  float64
 12  short_passing         183142 non-null  float64
 13  volleys               181265 non-null  float64
 14  dribbling             183142 non-null  float64
 15  curve                 181265 non-null  float64
 16  free_kick_accuracy    183142 non-null  float64
 17  long_passing          183142 non-null  float64
 18  ball_control          183142 non-null  float64
 19  acceleration          183142 non-null  float64
 20  sprint_speed          183142 non-null  float64
 21  agility               181265 non-null  float64
 22  reactions             183142 non-null  float64
 23  balance               181265 non-null  float64
 24  shot_power            183142 non-null  float64
 25  jumping               181265 non-null  float64
 26  stamina               183142 non-null  float64
 27  strength              183142 non-null  float64
 28  long_shots            183142 non-null  float64
 29  aggression            183142 non-null  float64
 30  interceptions         183142 non-null  float64
 31  positioning           183142 non-null  float64
 32  vision                181265 non-null  float64
 33  penalties             183142 non-null  float64
 34  marking               183142 non-null  float64
 35  standing_tackle       183142 non-null  float64
 36  sliding_tackle        181265 non-null  float64
 37  gk_diving             183142 non-null  float64
 38  gk_handling           183142 non-null  float64
 39  gk_kicking            183142 non-null  float64
 40  gk_positioning        183142 non-null  float64
 41  gk_reflexes           183142 non-null  float64
dtypes: float64(35), int64(3), object(4)
memory usage: 60.4+ MB
```

In [49]:

```
df_pa.duplicated().sum()
```

Out[49]:

```
0
```

In [50]:

```
df_pa.isnull().sum()
```

Out[50]:

```
Unnamed: 0                0
player_fifa_api_id        0
player_api_id             0
date                      0
overall_rating          836
potential               836
preferred_foot          836
attacking_work_rate    3230
defensive_work_rate     836
crossing                836
finishing               836
heading_accuracy        836
short_passing           836
volleys                2713
dribbling               836
curve                  2713
free_kick_accuracy      836
long_passing            836
ball_control            836
acceleration            836
sprint_speed            836
agility                2713
reactions               836
balance                2713
shot_power              836
jumping                2713
stamina                 836
strength                836
long_shots              836
aggression              836
interceptions           836
positioning             836
vision                 2713
penalties               836
marking                 836
standing_tackle         836
sliding_tackle         2713
gk_diving               836
gk_handling             836
gk_kicking              836
gk_positioning          836
gk_reflexes             836
dtype: int64
```

Player data dataframe cleaning

In [51]:

```
# dropping the columns that will not be used in the analyis

df_p.drop(['player_fifa_api_id', 'Unnamed: 0', 'player_api_id', 'birthday', 'weight'
, axis=1, inplace=True)
df_p.head()
```

Out[51]:

id	height
1	182.88
2	170.18
3	170.18
4	182.88
5	182.88

In [52]:

```
# creating a list from the height column in order to be appended to
# player attributes dataframe.

df_p_list= list(df_p['height'])
```

Player attributes data dataframe cleaning

In [53]:

```
# Viewing columns titles and their indices to select relevant columns
# into a modified dataframe

for index, value in enumerate(df_pa.columns):
    print(index, value)
```

```
0 Unnamed: 0
1 player_fifa_api_id
2 player_api_id
3 date
4 overall_rating
5 potential
6 preferred_foot
7 attacking_work_rate
8 defensive_work_rate
9 crossing
10 finishing
11 heading_accuracy
12 short_passing
13 volleys
14 dribbling
15 curve
16 free_kick_accuracy
17 long_passing
18 ball_control
19 acceleration
20 sprint_speed
21 agility
22 reactions
23 balance
24 shot_power
25 jumping
26 stamina
27 strength
28 long_shots
29 aggression
30 interceptions
31 positioning
32 vision
33 penalties
34 marking
35 standing_tackle
36 sliding_tackle
37 gk_diving
38 gk_handling
39 gk_kicking
40 gk_positioning
41 gk_reflexes
```

In [54]:

```
# creating a new dataframe for only the relevant player attribute ratings

df_pa_selected = df_pa.iloc[:, [4,6,10,14,18,11,30,35,36]]
df_pa_selected.head()
```

Out[54]:

id	overall_rating	preferred_foot	finishing	dribbling	ball_control	heading_accuracy	interception
1	67.0	right	44.0	51.0	49.0	71.0	70.
2	67.0	right	44.0	51.0	49.0	71.0	70.
3	62.0	right	44.0	51.0	49.0	71.0	41.
4	61.0	right	43.0	50.0	48.0	70.0	40.
5	61.0	right	43.0	50.0	48.0	70.0	40.

In [55]:

```
# cutting the sample size of the new dataframe into 11060 entries to match
# the player height list number of entries in order to be
# merged successfully to each other into one combined dataframe

df_pa_matching= df_pa_selected.iloc[:11060, :]
df_pa_matching.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11060 entries, 1 to 11060
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   overall_rating    11021 non-null  float64
 1   preferred_foot    11021 non-null  object 
 2   finishing         11021 non-null  float64
 3   dribbling         11021 non-null  float64
 4   ball_control      11021 non-null  float64
 5   heading_accuracy  11021 non-null  float64
 6   interceptions     11021 non-null  float64
 7   standing_tackle   11021 non-null  float64
 8   sliding_tackle    10935 non-null  float64
dtypes: float64(8), object(1)
memory usage: 864.1+ KB
```

Forming a combined dataframe for both players height and player attributes, and dropping missing data.

In [56]:

```
df_combined = df_pa_matching.assign(player_height = df_p_list)
df_combined.head()
```

Out[56]:

id	overall_rating	preferred_foot	finishing	dribbling	ball_control	heading_accuracy	interception
1	67.0	right	44.0	51.0	49.0	71.0	70.
2	67.0	right	44.0	51.0	49.0	71.0	70.
3	62.0	right	44.0	51.0	49.0	71.0	41.
4	61.0	right	43.0	50.0	48.0	70.0	40.
5	61.0	right	43.0	50.0	48.0	70.0	40.

In [57]:

```
df_combined.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11060 entries, 1 to 11060
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   overall_rating    11021 non-null  float64
 1   preferred_foot    11021 non-null  object 
 2   finishing         11021 non-null  float64
 3   dribbling         11021 non-null  float64
 4   ball_control      11021 non-null  float64
 5   heading_accuracy  11021 non-null  float64
 6   interceptions     11021 non-null  float64
 7   standing_tackle   11021 non-null  float64
 8   sliding_tackle    10935 non-null  float64
 9   player_height     11060 non-null  float64
dtypes: float64(9), object(1)
memory usage: 950.5+ KB
```

In [58]:

```
df_combined.dropna(inplace= True)
```

In [59]:

```
df_combined.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10935 entries, 1 to 11060
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   overall_rating    10935 non-null  float64
 1   preferred_foot    10935 non-null  object 
 2   finishing         10935 non-null  float64
 3   dribbling         10935 non-null  float64
 4   ball_control      10935 non-null  float64
 5   heading_accuracy  10935 non-null  float64
 6   interceptions     10935 non-null  float64
 7   standing_tackle   10935 non-null  float64
 8   sliding_tackle    10935 non-null  float64
 9   player_height     10935 non-null  float64
dtypes: float64(9), object(1)
memory usage: 939.7+ KB
```

In [60]:

```
df_combined.head()
```

Out[60]:

id	overall_rating	preferred_foot	finishing	dribbling	ball_control	heading_accuracy	interception
1	67.0	right	44.0	51.0	49.0	71.0	70.
2	67.0	right	44.0	51.0	49.0	71.0	70.
3	62.0	right	44.0	51.0	49.0	71.0	41.
4	61.0	right	43.0	50.0	48.0	70.0	40.
5	61.0	right	43.0	50.0	48.0	70.0	40.

Creating a new column in the combined dataframe for player height levels

> player heights segmented into four categories with quartile increments from the mean player height

In [61]:

```
df_combined.player_height.describe()
```

Out[61]:

```
count    10935.000000
mean       181.865626
std          6.368719
min        157.480000
25%        177.800000
50%        182.880000
75%        185.420000
max        208.280000
Name: player_height, dtype: float64
```

In [62]:

```
# Creating players height categories names and limits

bin_edges = [157.48, 177.8 , 182.88, 185.42, 208.28]
bin_names = ["short" ,'below_average_height' ,'above_average_height', 'tall']
```

In [63]:

```
# creating the height_levels column and adding it to the final dataframe for EDA.

df_combined['height_levels'] = pd.cut(df_combined['player_height'],
bin_edges, labels=bin_names)
df_combined.head()
```

Out[63]:

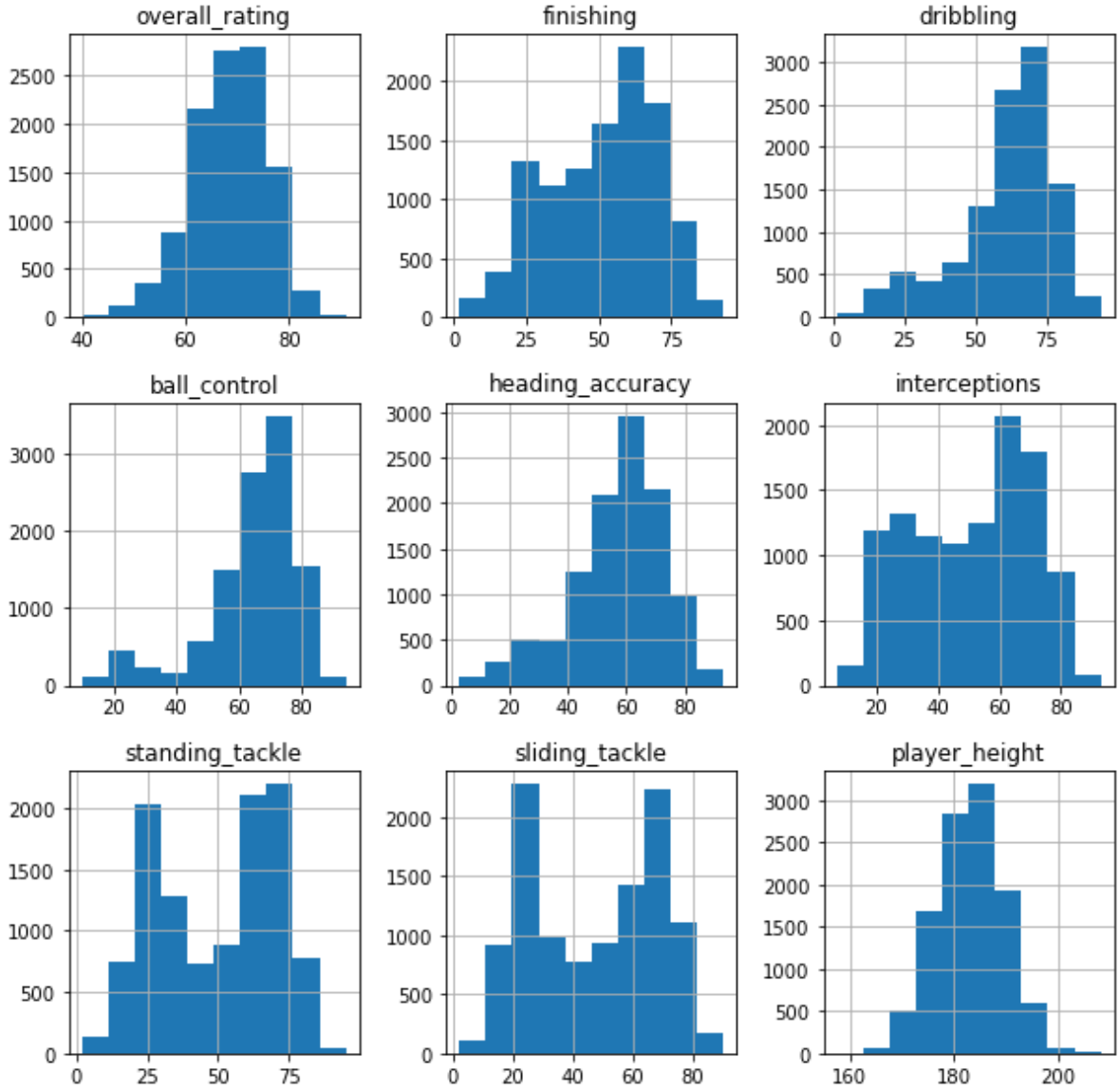
id	overall_rating	preferred_foot	finishing	dribbling	ball_control	heading_accuracy	interception
1	67.0	right	44.0	51.0	49.0	71.0	70.
2	67.0	right	44.0	51.0	49.0	71.0	70.
3	62.0	right	44.0	51.0	49.0	71.0	41.
4	61.0	right	43.0	50.0	48.0	70.0	40.
5	61.0	right	43.0	50.0	48.0	70.0	40.

Exploratory Data Analysis

General exploration

In [86]:

```
df_combined.hist(figsize= (10,10));
```



Reasoning section From the following histograms of the nine numerical player attribute ratings, we notice the distribution is skewed, that's why median was chosen to be the measure of central tendency instead of mean. moreover bimodal histograms were detected which better requires separation of data for these specific metrics for better analysis however; for the scope of this study it was used as is.

Research Question 1 - Does players' height positively affect their aerial and ground defensive abilities?

> Heading accuracy, interceptions, standing tackles, and sliding tackles.

In [65]:

```
df_combined.groupby('height_levels')['heading_accuracy'].median()
```

Out[65]:

```
height_levels
short                   59.0
below_average_height    59.0
above_average_height    59.0
tall                    59.0
Name: heading_accuracy, dtype: float64
```

In [66]:

```
df_combined.groupby('height_levels')['interceptions'].median()
```

Out[66]:

```
height_levels
short                   55.0
below_average_height    54.0
above_average_height    54.0
tall                    53.0
Name: interceptions, dtype: float64
```

In [67]:

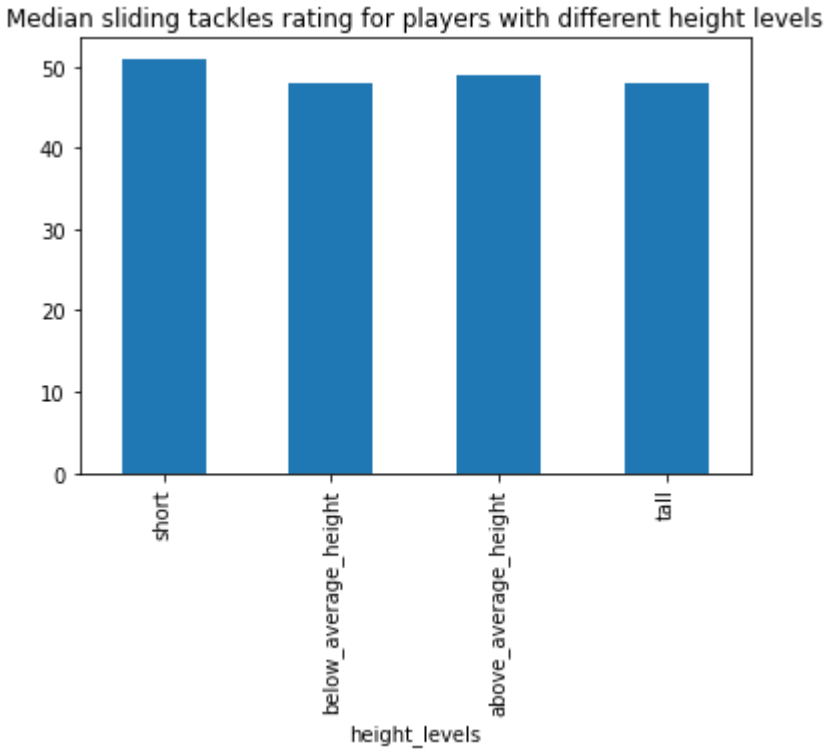
```
df_combined.groupby('height_levels')['standing_tackle'].median()
```

Out[67]:

```
height_levels
short                   56.0
below_average_height    53.0
above_average_height    55.0
tall                    54.0
Name: standing_tackle, dtype: float64
```

In [68]:

```
df_combined.groupby('height_levels')['sliding_tackle'].median().plot(kind= "bar",
title= 'Median sliding tackles rating for players with different height levels');
```



Reasonining section the previous bar chart depicts the median sliding tackle rating for player belonging to differen height categories, where we notice short player category recording the highest median.

Research Question 2 - Do left-footed players' possess better attacking pedigree compared to right-footed ones?

> Dribbling, finishing and ball control.

In [69]:

```
# Filtering right-footed and left-footed players data

right_footed_players= df_combined.query('preferred_foot == "right"')
left_footed_players= df_combined.query('preferred_foot == "left"')
```

In [70]:

```
right_footed_dribbling_median= right_footed_players['dribbling'].median()
left_footed_dribbling_median= left_footed_players['dribbling'].median()

right_footed_dribbling_median, left_footed_dribbling_median
```

Out[70]:

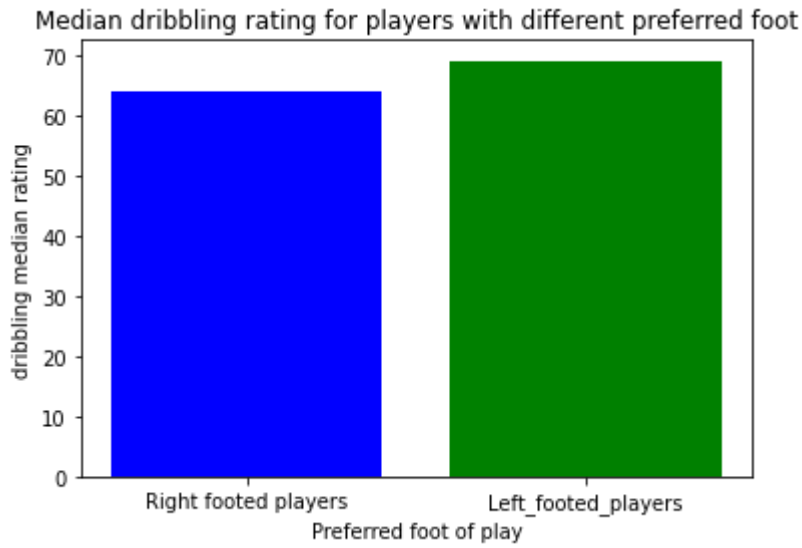
```
(64.0, 69.0)
```

In [71]:

```
# Creating a bar chart visualizing the comaprison between
# the median dribbling ratings for both player groups

locations = [1,2]
heights = [right_footed_dribbling_median, left_footed_dribbling_median]
colors = ["blue", "green"]
labels = ['Right footed players', 'Left_footed_players']

plt.bar(locations, heights, tick_label= labels, color= colors);
plt.title('Median dribbling rating for players with different preferred foot')
plt.xlabel('Preferred foot of play')
plt.ylabel('dribbling median rating')
```

Out[71]:

```
Text(0, 0.5, 'dribbling median rating')
```



Reasoning section the bar chart compares the median dribbling rating for both groups of left footed and right footed players, where it shows a higher median dribbling rating for the left footed players group.

In [72]:

```
right_footed_finishing_median= right_footed_players['finishing'].median()
left_footed_finishing_median= left_footed_players['finishing'].median()

right_footed_finishing_median, left_footed_finishing_median
```

Out[72]:

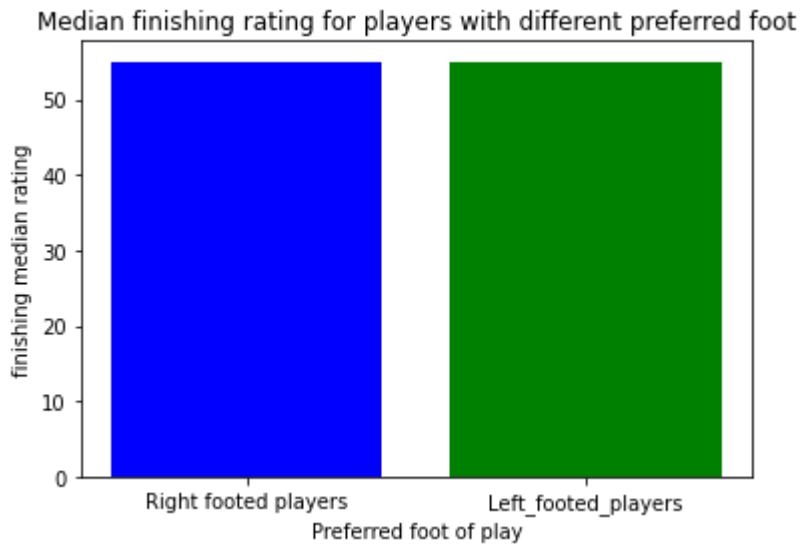
```
(55.0, 55.0)
```

In [73]:

```
# Creating a bar chart visualizing the comaprison between
# the median finishing ratings for both player groups

locations = [1,2]
heights = [right_footed_finishing_median, left_footed_finishing_median]
colors = ["blue", "green"]
labels = ['Right footed players', 'Left_footed_players']

plt.bar(locations, heights, tick_label= labels, color= colors);
plt.title('Median finishing rating for players with different preferred foot')
plt.xlabel('Preferred foot of play')
plt.ylabel('finishing median rating')
```

Out[73]:

```
Text(0, 0.5, 'finishing median rating')
```



Reasoning section the bar chart compares the median finishing rating for both groups of left footed and right footed players, where it shows an equal median finishing rating for the left footed players group.

In [74]:

```
right_footed_ballcontrol_median= right_footed_players['ball_control'].median()
left_footed_ballcontrol_median= left_footed_players['ball_control'].median()

right_footed_ballcontrol_median, left_footed_ballcontrol_median
```

Out[74]:

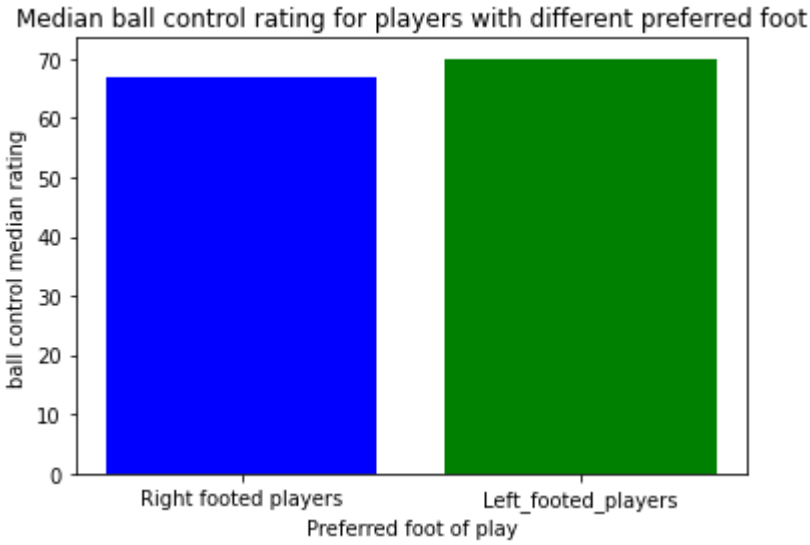
```
(67.0, 70.0)
```

In [75]:

```
# Creating a bar chart visualizing the comaprison between
# the median ball control ratings for both player groups

locations = [1,2]
heights = [right_footed_ballcontrol_median, left_footed_ballcontrol_median]
colors = ["blue", "green"]
labels = ['Right footed players', 'Left_footed_players']

plt.bar(locations, heights, tick_label= labels, color= colors);
plt.title('Median ball control rating for players with different preferred foot')
plt.xlabel('Preferred foot of play')
plt.ylabel('ball control median rating')
```

Out[75]:

```
Text(0, 0.5, 'ball control median rating')
```



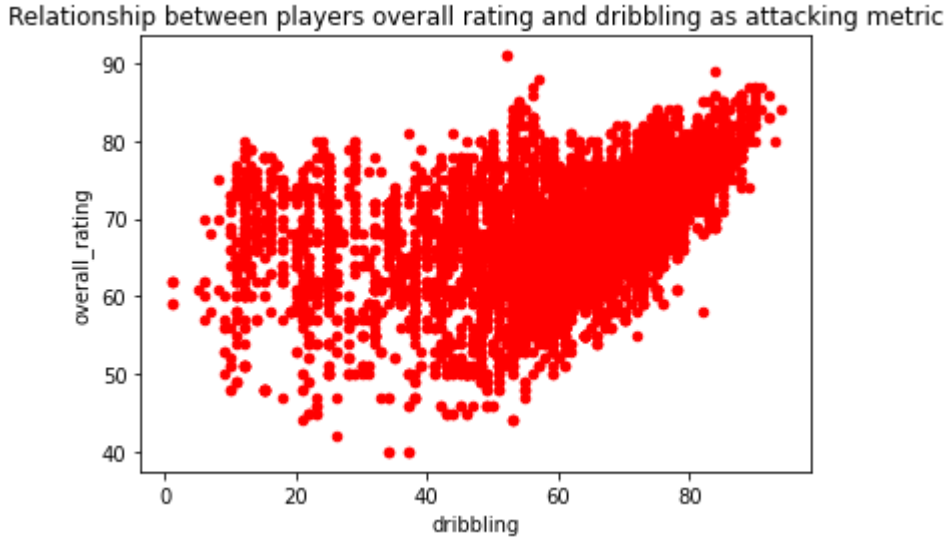
Reasoning section the bar chart compares the median ball control rating for both groups of left footed and right footed players, where it shows a higher median ball control rating for the left footed players group.

Research Question 3 - what is the relationship between overall ratings of players and their attacking and defensive ratings

> Dribbling used as attacking metric surrogate, standing tackle used as defensive metric surrogate.

In [79]:

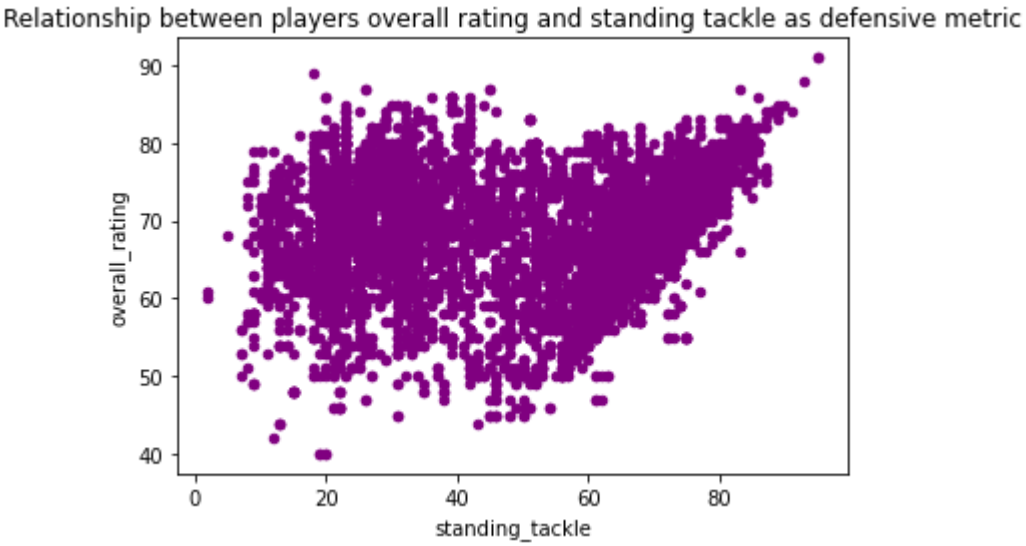
```
def scatter_plot(arg1, arg2, arg3):
    df_combined.plot(x= arg1, y= 'overall_rating', kind= 'scatter', color= arg2, tit

scatter_plot('dribbling', 'red', 'Relationship between players overall rating and dr
```



Reasoning section This scatter chart highlights the positive relationship between players dribbling rating and their overall rating.

In [82]:

```
scatter_plot('standing_tackle','purple',
'Relationship between players overall rating and standing tackle as defensive metric
```



Reasoning section This scatter chart highlights the positive relationship between players standing tackle rating and their overall rating.

Conclusions

> **Disclaimer** The conclusions drawn are based on observations from descriptive study and no inferential statistics were undertaken to test significance of the outcomes. the conclusion is also limited by sample size, and the variety of defensive and attacking performance metrics chosen for assessment.

- Players' height looks to have no infleunce on their heading accuracy ratings.
- There is little difference in interceptions ratings of players' belonging to different height levels, with players on the shorter end recording a relatively higher median ratings.
- Players in the lowest height level has the highest standing and sliding tackle median ratings compared to other height groups.
- Relative superior performance of shorter players in the chosen defensive metrics might be attributed to their higher agility and speed, however this claim requires further investigation.

The preliminary analysis suggests against the general perception that taller players perform better defensively.

- Left-footed players recorded higher median dribbling and ball control rating compared to right-footed players.
- Median finishing rating comparison showed no difference between the left and right footed player groups.

The analysis supports the claim that left-footed players have superior ball control and dribbling abilities in attack build up, however not necessarily in goal scoring.

- The positive relationship between players overall ratings and their dribbling (attacking ability surrogate), and standing tackle (defensice ability surrogate) kicks off stronger at higher dribbling and standing tackle ratings (around the 40 points rating) for each

The analysis suggests a positive relationship between players' overall ratings and both of their attacking, and defensive rating in an undifferentiated fashion.