

In [24]:

```
import numpy as np
import pandas as pd
import sqlite3
import matplotlib.pyplot as plt

database = 'data.sqlite'

conn = sqlite3.connect(database)
```

In [25]:

```
tables = pd.read_sql("""SELECT *
                        FROM sqlite_master
                        WHERE type='table';""", conn)

tables.head(5)
```

Out[25]:

	type	name	tbl_name	rootpage	sql
0	table	sqlite_sequence	sqlite_sequence	4	CREATE TABLE sqlite_sequence(name,seq)
1	table	Player_Attributes	Player_Attributes	11	CREATE TABLE "Player_Attributes" (\n\t'id'\tIN...
2	table	Player	Player	14	CREATE TABLE `Player` (\n\t'id`\tINTEGER PRIMA...
3	table	Match	Match	18	CREATE TABLE `Match` (\n\t'id`\tINTEGER PRIMAR...
4	table	League	League	24	CREATE TABLE `League` (\n\t'id`\tINTEGER PRIMA...

In [26]:

```
team = pd.read_sql('''select * from Team;''', conn)
team.head(5)
```

Out[26]:

	id	team_api_id	team_fifa_api_id	team_long_name	team_short_name
0	1	9987	673.0	KRC Genk	GEN
1	2	9993	675.0	Beerschot AC	BAC
2	3	10000	15005.0	SV Zulte-Waregem	ZUL
3	4	9994	2007.0	Sporting Lokeren	LOK
4	5	9984	1750.0	KSV Cercle Brugge	CEB

In [27]:

```
TAri = pd.read_sql(''select * from Team_Attributes;'', conn)
TAri.head(5)
```

Out[27]:

	id	team_fifa_api_id	team_api_id	date	buildUpPlaySpeed	buildUpPlaySpeedClass	buil
0	1	434	9930	2010-02-22 00:00:00	60	Balanced	
1	2	434	9930	2014-09-19 00:00:00	52	Balanced	
2	3	434	9930	2015-09-10 00:00:00	47	Balanced	
3	4	77	8485	2010-02-22 00:00:00	70	Fast	
4	5	77	8485	2011-02-22 00:00:00	47	Balanced	

5 rows × 25 columns

In [28]:

```
team_api = pd.read_sql(''select Team_Attributes.defenceAggressionClass , Team_Attributes
                        from Team_Attributes left join Team on Team_Attributes.te
team_api.head(5)
```

Out[28]:

	defenceAggressionClass	defenceTeamWidth	date	team_api_id	team_fifa_api_id	team_
0	Press	45	2010-02-22 00:00:00	9930	434	
1	Press	54	2014-09-19 00:00:00	9930	434	
2	Press	54	2015-09-10 00:00:00	9930	434	
3	Double	70	2010-02-22 00:00:00	8485	77	
4	Press	52	2011-02-22 00:00:00	8485	77	

In [29]:

```
player = pd.read_sql('''select * from Player order by height desc;''', conn)
player.head(5)
```

Out[29]:

	id	player_api_id	player_name	player_fifa_api_id	birthday	height	weight
0	5908	148325	Kristof van Hout	185306	1987-02-09 00:00:00	208.28	243
1	1301	150209	Bogdan Milic	189967	1987-11-24 00:00:00	203.20	216
2	1926	103428	Costel Pantilimon	192613	1987-02-01 00:00:00	203.20	212
3	3274	601304	Fejsal Mulic	226114	1994-10-03 00:00:00	203.20	185
4	5568	26585	Jurgen Wevers	148875	1979-01-12 00:00:00	203.20	212

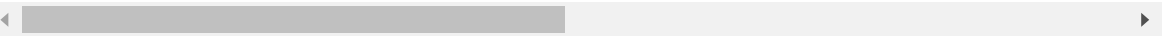
In [30]:

```
Match = pd.read_sql('''select * from Match;''', conn)
Match.head(5)
```

Out[30]:

	id	country_id	league_id	season	stage	date	match_api_id	home_team_api_id	av
0	1	1	1	2008/2009	1	2008-08-17 00:00:00	492473		9987
1	2	1	1	2008/2009	1	2008-08-16 00:00:00	492474		10000
2	3	1	1	2008/2009	1	2008-08-16 00:00:00	492475		9984
3	4	1	1	2008/2009	1	2008-08-17 00:00:00	492476		9991
4	5	1	1	2008/2009	1	2008-08-16 00:00:00	492477		7947

5 rows × 115 columns



In [31]:

```
detail_player = pd.read_sql('''select  case
                                when round(height)<165 then 165
                                when round(height)>195 then 195
                                else round(height)
                                end as calc_height,
                                count(height),
                                avg(PAG.avg_overall_rating),
                                avg(weight)
                                from Player left join (
                                    select player_api_id,
                                    avg(overall_rating) as avg_overall_rating
                                    from Player_Attributes
                                    group by player_api_id) as PAG
                                on Player.player_api_id = PAG.player_api_id
                                group by calc_height
                                ;''', conn)
detail_player.head(5)
```

Out[31]:

	calc_height	count(height)	avg(PAG.avg_overall_rating)	avg(weight)
0	165.0	74	67.365543	139.459459
1	168.0	118	67.500518	144.127119
2	170.0	403	67.726903	147.799007
3	173.0	530	66.980272	152.824528
4	175.0	1188	66.805204	156.111953

In [32]:

```
detail_player.plot(x='calc_height', y='avg(PAG.avg_overall_rating)')
```

Out[32]:

<AxesSubplot:xlabel='calc\_height'>

