AHMET ÇÖVÜT

214962709

240 PROJECT

SECTION1:

What is relation between player stats (points and assist) and accepted all-star?

What is the relation club salary and championship?

What is the relation player's average points and their body (height-weight) index?

Is player's race affect their success rate?

First, we can look points, assist and game play of players and we can create new column like successful of player. After, merge allstar and players stats of data frames and we can compare, it which is success rate enough for allstar.

Secondly, it is easy question. We should look salary of clubs and which club be champion year by year.

I select third question I wonder about body index effect performance of players and firstly I calculate body index: weight(kg)/height(m)^2. and we calculate average scores calculate with points an game played of players.

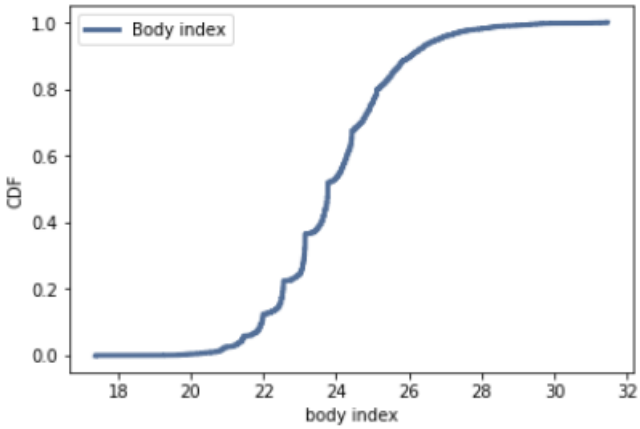
Therefore, my hypothesis is low body index, increase points of players.

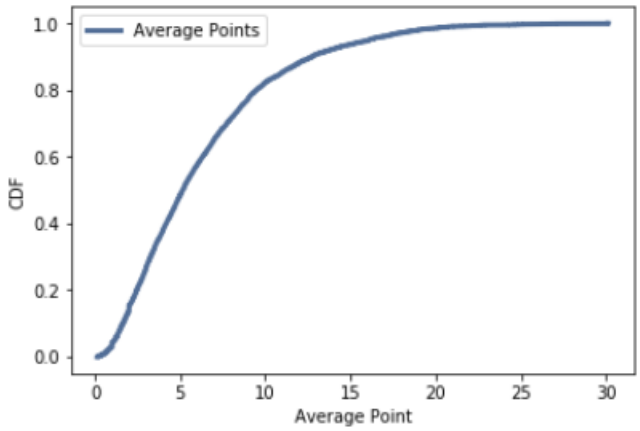
Null hypothesis: low body index does not affect point of player.

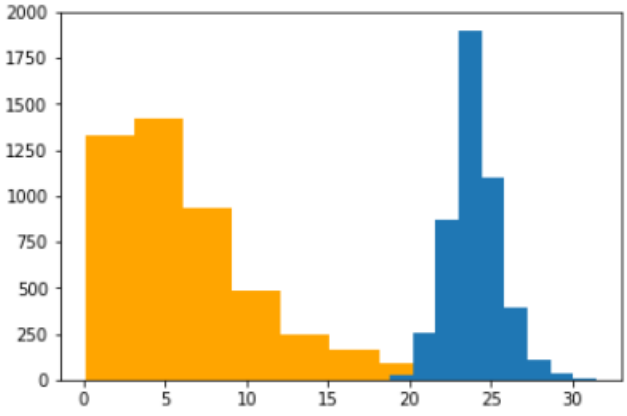
SECTION2:

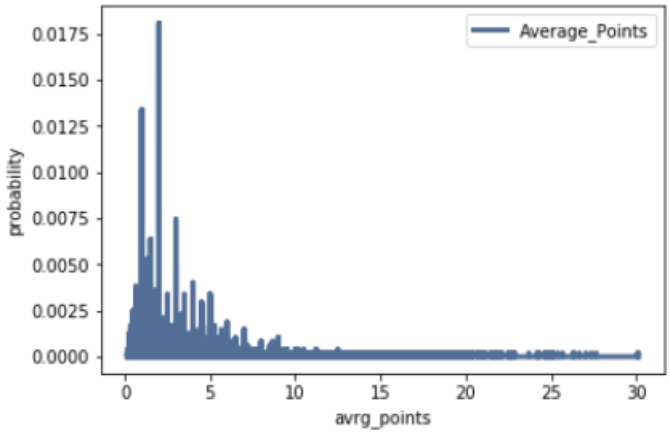
I will use basketball_players.csv and basketball_master.csv. I take playerID, points, GP column from basketball_players.csv and I take bioID, weight height from basketball_master.csv. I calculate average points of player like as points/GP and I should convert height (inch) and weight (lbs). After, I convert they like meter and kg. I will merge these two data frame bioID and playerID include same variable and I grouped these column and I will union these data frames. I select only playerID, average points and body index of player.

SECTION3:



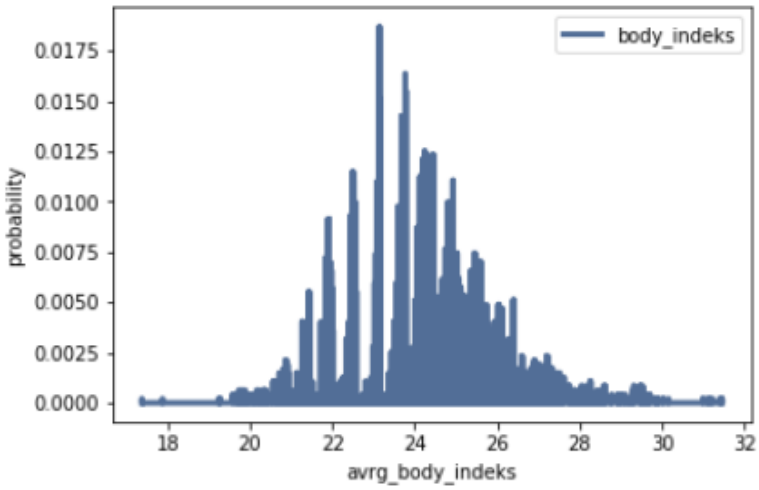







```
print (mergelast.avrg_points.mode())
print (mergelast.body_indeks.mode())

print (mergelast.avrg_points.mean())
print (mergelast.body_indeks.mean())

print (mergelast.avrg_points.max())
print (mergelast.body_indeks.max())

print (mergelast.avrg_points.std())
print (mergelast.body_indeks.std())


print (mergelast.avrg_points.var())
print (mergelast.body_indeks.var())
```

```
0    0.0
dtype: float64
0    23.143957
dtype: float64
6.127174409326469
23.785249308792327
30.12313432835821
31.461755476856865
4.650716983058464
2.6513558725276214
21.629168456508424
7.0296879627867055
```



Cdf graph has going to increase because they are cumulative and add probability of body index and points these cdf graphs

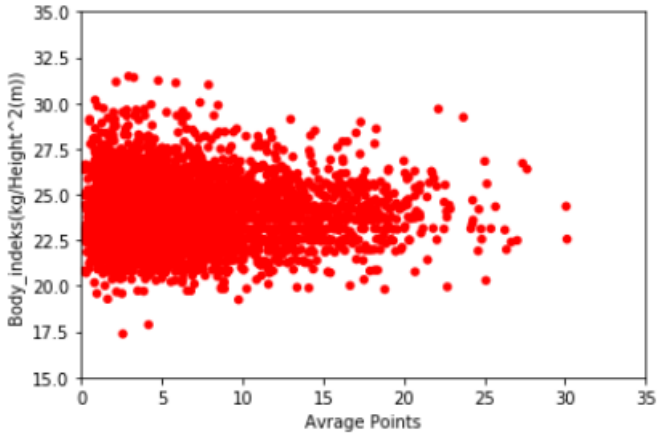
Histogram show how many average body index and average points of player

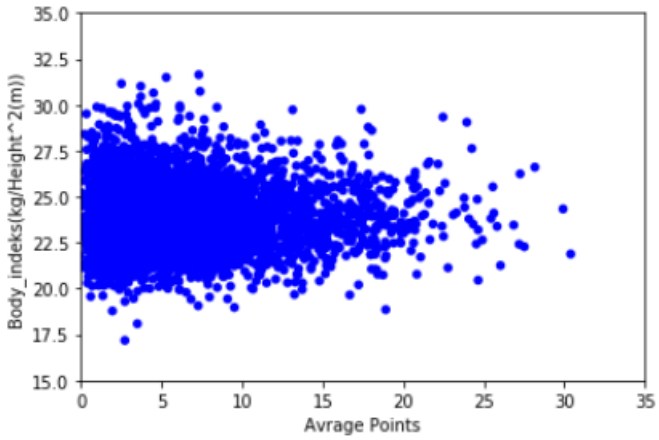
Pdf graph show probability of average body index and average points of player.

Section4:

I delete only has 0 about points and body index. Points may be possible but body index cannot be 0 of human

Section5: Scatter of my datas and I used random jitters but it has not big changes.





The second graph with jitter.

```
Corr(mergelast.avrg_points, mergelast.body_indeks)
```

```
-0.03847371739315529
```

Result has negative correlation. It is near the zero so it has not much effect on points of player. But little opposite relation between body index and points.

```
Cov(mergelast.avrg_points,mergelast.body_indeks)
```

```
-0.29059028849501134
```

Covariance of the data.

```
SpearmanCorr(mergelast.avrg_points, mergelast.body_indeks)
```

```
-0.04545486002283289
```

Spearman correlation is give similar result like as first correlation.

SECTION6:

```
ht.actual, ht.MaxTestStat()
```

```
(0.03847371739315529, 0.05256339426190857)
```

Null hypothesis is rejected because result 0.038< 0.05. Low body index is better than high. Over 25 body index mean is overweight and it can affect players.

SECTION 7:

Briefly, I observe data files, manually and I select data files which are may be using. I should create relation between these columns. In health, body index is important and data files include data for calculation of body index and I try make connection with performance of players. Thus, I calculate average points of players. I clean unnecessary data from data files and merge these files. I use statistic for these data columns and my hypothesis is correct or not. In the results my hypothesis is little bit correct.

References

Lecture slides and codes