**1. Average versus median**

You now know how to use numpy functions to get a better feeling for your data. It basically comes down to importing numpy and then calling several simple functions on the numpy arrays:

import numpy as np

x = [1, 4, 8, 10, 12]

np.mean(x)

np.median(x)

The baseball data is available as a 2D numpy array with 3 columns (height, weight, age) and 1015 rows. The name of this numpy array is np\_baseball. After restructuring the data, however, you notice that some height values are abnormally high. Follow the instructions and discover which summary statistic is best suited if you're dealing with so-called *outliers*.

* Create numpy array np\_height that is equal to first column of np\_baseball.
* Print out the mean of np\_height.
* Print out the median of np\_height.

# 2. Explore the baseball data

Because the mean and median are so far apart, you decide to complain to the MLB. They find the error and send the corrected data over to you. It's again available as a 2D Numpy array np\_baseball, with three columns.

The Python script on the right already includes code to print out informative messages with the different summary statistics. Can you finish the job?

* The code to print out the mean height is already included. Complete the code for the median height. Replace None with the correct code.
* Use **[np.std()](http://docs.scipy.org/doc/numpy-1.10.0/reference/generated/numpy.std.html" \t "_blank)** on the first column of np\_baseball to calculate stddev. Replace None with the correct code.
* Do big players tend to be heavier? Use **[np.corrcoef()](http://docs.scipy.org/doc/numpy-1.10.0/reference/generated/numpy.corrcoef.html" \t "_blank)** to store the correlation between the first and second column of np\_baseball in corr. Replace None with the correct code.

# 3. Blend it all together

In the last few exercises you've learned everything there is to know about heights and weights of baseball players. Now it's time to dive into another sport: soccer.

You've contacted FIFA for some data and they handed you two lists. The lists are the following:

positions = ['GK', 'M', 'A', 'D', ...]

heights = [191, 184, 185, 180, ...]

Each element in the lists corresponds to a player. The first list, positions, contains strings representing each player's position. The possible positions are: 'GK' (goalkeeper), 'M' (midfield), 'A' (attack) and 'D' (defense). The second list, heights, contains integers representing the height of the player in cm. The first player in the lists is a goalkeeper and is pretty tall (191 cm).

You're fairly confident that the median height of goalkeepers is higher than that of other players on the soccer field. Some of your friends don't believe you, so you are determined to show them using the data you received from FIFA and your newly acquired Python skills.

* Convert heights and positions, which are regular lists, to numpy arrays. Call them np\_heights and np\_positions.
* Extract all the heights of the goalkeepers. You can use a little trick here: use np\_positions == 'GK' as an index for np\_heights. Assign the result to gk\_heights.
* Extract all the heights of all the other players. This time use np\_positions != 'GK' as an index for np\_heights. Assign the result to other\_heights.
* Print out the median height of the goalkeepers using **[np.median()](http://docs.scipy.org/doc/numpy-1.10.0/reference/generated/numpy.median.html" \t "_blank)**. Replace None with the correct code.
* Do the same for the other players. Print out their median height. Replace None with the correct code.