# CHS 0008
# University of Pittsburgh
# Project 2

## Overview

For this project, you will be looking at a competition of your choosing (e.g., the NHL season/playoffs), and analyzing the statistics of the competitors to see what attributes are shared by successful competitors.

Consider the 2015-2016 NHL season. You could write a program that would look at (for each NHL team), the average number of points scored in a game, the average number of points scored by opponents in a game, the average number of shots taken per game, the average number of shots taken by opponents per game, and the percentage of penalties killed throughout the season. You program would then compute the average, min, and max of each of these stats across all teams in the NHL, the average, min, and max of teams that made the playoffs, and the average, min, and max of teams that did not make the playoffs. Are all of these statistics important to a winning team? Is there one that stands out as prevalent amongst winning teams? Is there one that seems to indicate that a team will not make the playoffs?

# 1 Gathering data

The first part of this project will be to pick a competition to analyze and gather data. You must pick a competition where at least 8 competitors/teams will be considered winners (e.g., make the playoffs), and at least 8 will not. Then, you must pick 5 statistics for each competitor/team (e.g., goals scored per game, goals scored against per game, shots taken per game, shots taken against per game, % of penalties killed) and gather the necessary data into a single text file. All of your statistics should be of type `float`.

Once you have selected a competition and statistics to track, you should construct a text file named "data.txt" that will store all of your data. This file should be formatted such that there is one value on each line. E.g.,

```
Name1    (string)
Stat1a   (float)
Stat1b   (float)
Stat1c   (float)
Stat1d   (float)
Stat1e   (float)
Winner?  (boolean)
Name2    (string)
Stat2a   (float)
Stat2b   (float)
Stat2c   (float)
Stat2d   (float)
Stat2e   (float)
Winner?  (boolean)
...
```

For example, a file for the 2015-2016 NHL season tracking the stats listed above could result in a data.txt that looks like this:

```
Pittsburgh Penguins
2.94
2.43
33.2
29.7
84.4
True
New Jersey Devils
2.22
2.46
24.4
28.6
83.0
False
...
```

# 2   Basic analysis

For the second part of the project, you should write a Python program that will read your "data.txt" and print out the min, max, and average for each of your chosen statistics. Note that you will need to be very careful about reading through your file to ensure that you read all of your data correctly.

An example run for the 2015-2016 NHL season would look like this:

```
          PPG     PAPG   SHOTPG  SHOTAPG      PK
   MIN    2.22    2.29    24.40    27.30   75.50
   MAX    3.23    3.13    33.20    32.80   87.20
   AVG    2.67    2.67    29.74    29.73   81.32
```

# 3    Utilizing dictionaries, lists, and functions

For the third part of this project, you will have to redo your previous solution to split it up into multiple parts. Specifically, you will be writing 2 functions (one of which should be able to be called 2 different ways) to aid in your analysis:

`readFile(filename)`

The first function, `readFile`, should be written with a default argument value so that it can be called with either one or two arguments (with or without the `winner` argument). When called with only the `filename` argument, it should be used to read a file specified by `filename` and produce a dictionary containing all of the stats for each team or competitor. Each entry in the dictionary should be a list containing the stats for a single team. Hence, the dictionary will contain $n$ sublists where $n$ is the number of teams/competitors. For the NHL example, this would be the result:

| "Pittsburgh Penguins": | 2.94 | 2.43 | 33.2 | 29.7 | 84.4 |
|---|---|---|---|---|---|
| "New Jersey Devils": | 2.22 | 2.46 | 24.4 | 28.6 | 83.0 |

...

`readFile(filename, winner)`

When called with two arguments (both `filename` and `winner`), `readFile` should operate in a very similar way, but it should only process only a portion of the file specified by `filename`. If `winner` is set to `True`, it should return a dictionary containing only the stats of teams/competitors that are considered winners (e.g., NHL teams that made the playoffs). If set to `False`, it should return a dictionary of only teams/competitors who are not considered winners (e.g., NHL teams that did not make the playoffs).

`analyze(data)`

This function should take in a dictionary of statistics and print out the min, max, and average of each sublist (i.e., of each statistic).

## Bringing the functions together

Once you have finished writing both of these functions, your program should operate as follows:

- Call `readFile(filename)` to get the statistics on all teams/competitors.

- Call `analyze(data)` on the resulting dictionary to display the min, max, and average across all teams/competitors.

- Call `readFile(filename, winner)` with winner set to `True` to get the statistics on all winners.

- Call `analyze(data)` on the resulting dictionary to display the min, max, and average across all winners.

- Call `readFile(filename, winner)` with winner set to `False` to get the statistics on all non-winners.

- Call `analyze(data)` on the resulting 2D array to display the min, max, and average across all non-winners.

# Notes and hints

- Note that this assignment will require you to carefully keep track of the ordering of your statistics (e.g., points scored per game is the first stat after the team name in "data.txt" and will appear as the first entry in the corresponding statistics list).

- Since implementing the option to call `readFile` with two arguments can be tricky, it may be helpful to write it to accept only a single argument and get that functionality working before modifying it optionally accept two arguments.