

1. Player Matching

'''

Consider a set of players in a game, where each player has a name and a skill level. A player's skill level is a number representing how skilled they are at the game, where higher numbers are better. More specifically, when two players play against each other, the difference in skill level between the two players tells us information about what the outcome may be. The further apart the two players' skill levels are, the more likely it is that the higher skill player will win.

'''

class Player:

```
def __init__(self, name, skill):
    self.name = name
    self.skill = skill
```

'''

We would like to predict the outcome of a match between two players, given their respective skill levels. Specifically, how do we estimate the probability that player 1 will win?

You have access to a sample dataset you can use for testing: dataset.csv, a csv file of matches that previously occurred. It contains the following five columns:

- * player1_id: an integer ID that uniquely identifies player 1
- * player2_id: an integer ID that uniquely identifies player 2
- * player1_skill: player 1's skill level
- * player2_skill: player 2's skill level
- * winner_id: the ID of the winning player

* Part 1 *

Submit a jupyter notebook exploring the data, outlining the logic of approach, and evaluating and interpreting results.

* Part 2 *

Write a function that takes the following inputs:

- * Player 1 Player Object
- * Player 2 Player Object

The function should return an estimate of the probability that player 1 will win.

You are allowed to import commonly used statistics and data analysis libraries. You may also use any online resources you want.

'''

```
def predict_outcome(player1, player2):
    # fill in the body of this function
```

example usage:

```
arpad = Player('Arpad', 365)
mark = Player('Mark', 341)
result = predict_outcome(arpad, mark)
print('The probability that Arpad will beat Mark is:', result)
```