Lab

April 3, 2017

1 COMP 10020 Introduction to Programming 2

1.1 Lab 1 - Rugby Sabremetrics

In this lab you will be tasked with exercising your key Python programming skills. The **Pick & Go Test Match Results Database** (http://www.lassen.co.nz/pickandgo.php) contains the results of every international rugby match played since 1875. The following code block reads data from a data file (*'RugbyResultsData.csv'*) scraped from Pick & Go and stores this in a list of dictionary objects, each of which contains the detaisl of a match. The details stored about each match are stored in a dictionary object with the following keys:

- **Date**: The date on which the match was played
- **Day**: The day of the week on which the match was played
- **Year**: The year in which the match was played
- **Team_1**: The home team (three letter country code, e.g. IRL = Ireland, NZL = New Zealand)
- **Team_2**: The away team (three letter country code, e.g. IRL = Ireland, NZL = New Zealand)
- **Team_1_Score**: The score achieved by the home team.

- **Team_2_Score**: The score achieved by the away team.

- **Team_1_Tries**: The number of tries scored by the home team.
- **Team_2_Tries**: The number of tries scored by the away team.
- **Neutral**: Was the match played at a neutral venue?

```
In [3]: matches = list()
        count = 0
        with open('RugbyResultsData.csv') as f:
            for line in f:
                words = line.split(',')
                match = {"Date":words[0],"Day":words[1],"Year":int(words[2]),"Team_
                matches.append(match)
                count = count + 1

        print(str(count) + " matches loaded")
```

```
3234 matches loaded
```

The details of a specific match are accessed by first selecting an element from the **matches** list and then accessing specific pieces of data about that match from the dictionary. For example, the year and teams involved in the third match in the list could be accessed as follows:

```
In [46]: matches[2]["Year"]

Out[46]: 1875

In [47]: matches[2]["Team_1"]

Out[47]: 'IRE'

In [48]: matches[2]["Team_2"]

Out[48]: 'ENG'
```

The task in this lab is to answer the question below using simple Python code. **DO NOT** use advanced packages (e.g. pandas or numpy) for these tasks, write all code yourself.

1.1.1 Question 1

What is the highest score that any team has achieved in a single match?

```
In [9]: # Write code to determine and print the answer here.

        high_score = 0
        high_score_match = None

        for match in matches:
            if(match["Team_1_Score"] > high_score):
                high_score = match["Team_1_Score"]
                high_score_match = match

            if(match["Team_2_Score"] > high_score):
                high_score = match["Team_2_Score"]
                high_score_match = match

        print("high score:" + str(high_score))
        print(high_score_match)
```

```
high score:145
{'Neutral': 'Y', 'Team_2_Score': 17, 'Team_1_Score': 145, 'Team_1': 'NZL', 'Team_2'
```

1.1.2 Question 2

What is the highest number of tries that any team has achieved in a single match?

```
In [10]: # Write code to determine and print the answer here.
         high_tries = 0
         high_tries_match = None

         for match in matches:
             if(match["Team_1_Tries"] > high_tries):
                 high_tries = match["Team_1_Tries"]
                 high_tries_match = match

             if(match["Team_2_Tries"] > high_tries):
                 high_tries = match["Team_2_Tries"]
                 high_tries_match = match

         print("high tries:" + str(high_tries))
         print(high_tries_match)
```

```
high tries:22
{'Neutral': 'N', 'Team_2_Score': 0, 'Team_1_Score': 142, 'Team_1': 'AUS', 'Team_2':
```

1.1.3 Question 3

What is the biggest points difference ever seen in a match? Who was playing?

```
In [12]: # Write code to determine and print the answer here.
         high_points_diff = 0
         high_points_diff_match = None

         for match in matches:
             points_diff = abs(match["Team_1_Score"] - match["Team_2_Score"])

             if(points_diff > high_points_diff):
                 high_points_diff = points_diff
                 high_points_diff_match = match

         print("high points diff:" + str(high_points_diff))
         print(high_points_diff_match)
```

```
high points diff:142
{'Neutral': 'N', 'Team_2_Score': 0, 'Team_1_Score': 142, 'Team_1': 'AUS', 'Team_2':
```

1.1.4 Question 4

How many matches have Ireland played since records began?

```
In [18]: # Write code to determine and print the answer here.
         ire_games = 0
         for match in matches:

             if(match["Team_1"] == "IRE" or match["Team_2"] == "IRE"):
                 ire_games = ire_games + 1

         print("Ireland games:" + str(ire_games))
```

```
Ireland games:669
```

1.1.5 Question 6

How many times have Ireland played England since records began?

```
In [19]: # Write code to determine and print the answer here.
         ire_eng_games = 0

         for match in matches:

             if((match["Team_1"] == "IRE" and match["Team_2"] == "ENG") or (match['
                 ire_eng_games = ire_eng_games + 1

         print("Ireland-England games:" + str(ire_eng_games))
```

```
Ireland-England games:131
```

1.1.6 Question 5

Which team have played the most matches since records began?

```
In [22]: # Write code to determine and print the answer here.
         games_played_map = dict()

         # Count the number of matches played by each team
         for match in matches:
             if(match["Team_1"] in games_played_map):
                 games_played_map[match["Team_1"]] += 1
             else:
                 games_played_map[match["Team_1"]] = 1

             if(match["Team_2"] in games_played_map):
                 games_played_map[match["Team_2"]] += 1
             else:
                 games_played_map[match["Team_2"]] = 1

         print(games_played_map)

         # Determine the team that has played the most matches
```

```
most_played = 0
most_played_team = None

for team in games_played_map:
    if(games_played_map[team] > most_played):
        most_played = games_played_map[team]
        most_played_team = team

print(most_played_team)
print(most_played)
```

```
{'PAR': 1, 'USA': 75, 'LIO': 108, 'CAV': 4, 'NAM': 27, 'PAC': 9, 'TON': 59, 'CAN':
FRA
723
```

1.1.7 Question 7

Which team is the "*winningest*" (i.e. has managed the most wins since records began)?

```
In [24]: # Write code to determine and print the answer here.
         games_won_map = dict()

         # Count the number of wins for each team
         for match in matches:
             if(match["Team_1_Score"] > match["Team_2_Score"]):
                 if(match["Team_1"] in games_won_map):
                     games_won_map[match["Team_1"]] += 1
                 else:
                     games_won_map[match["Team_1"]] = 1

             elif(match["Team_2_Score"] > match["Team_1_Score"]):
                 if(match["Team_2"] in games_won_map):
                     games_won_map[match["Team_2"]] += 1
                 else:
                     games_won_map[match["Team_2"]] = 1

         print(games_won_map)

         # Determine the team that has won the most matches
         most_won = 0
         most_won_team = None

         for team in games_won_map:
             if(games_won_map[team] > most_won):
                 most_won = games_won_map[team]
                 most_won_team = team
```

```
        print(most_won_team)
        print(most_won)

{'USA': 10, 'LIO': 40, 'CAV': 1, 'NAM': 2, 'PAC': 1, 'TON': 13, 'CAN': 16, 'GER': 2
NZL
442
```

1.1.8 Question 8

Is there really a home advantage – i.e. does the home team win more often than the away team? Justify your answer.

```
In [27]: # Write code to determine and print the answer here.
         home_wins = 0
         away_wins = 0

         # Count the number of wins for each team
         for match in matches:
             if(match["Neutral"] == "N"):
                 if(match["Team_1_Score"] > match["Team_2_Score"]):
                     home_wins += 1
                 elif(match["Team_2_Score"] > match["Team_1_Score"]):
                     away_wins += 1

         print(home_wins)
         print(away_wins)

         if(home_wins > away_wins):
             print("Home advantage is real")
         elif(away_wins > home_wins):
             print("Home advantage is fake news")

1711
1122
Home advantage is real


In [ ]:
```