Basic Data Formatting

In this notebook we will prepare an arbitrary data file containing at least a list of matches, and the winner and loser in "winner_name" and "loser_name" columns.

The resulting csv file will contain a list of pairwise player comparisons containing the number of times each player beat the other. This format is ready for use by the standard Bradley-Terry model.

First we import the required libraries.

```
library(utils)
library(BradleyTerry2)
library(ggplot2)
library(tidyr)
```

Below, we state the names of the data files we read from, and read them into a dataframe.

```
# Define the file path
file_paths <- c("data/atp_matches_2023.csv", "data/atp_matches_2022.csv",
                "data/atp_matches_2021.csv", "data/atp_matches_2020.csv",
                "data/atp_matches_2019.csv")

df <- read.csv(file_paths[1])

# compile

for (i in 2:(length(file_paths)-1)) {
  df <- rbind(df, read.csv(file_paths[i]))
}
```

We then extract the relevant columns from the dataframe, and rename them to df2.

```
# make a dataframe with the data we need
df2 <- df[,c("winner_name", "loser_name")]


winners = df2$winner_name
losers = df2$loser_name
players = intersect(winners, losers)
```

The next chunk calculates the number of matches each player has played.

```
player_freqs <- setNames(rep(0, length(players)), players)
for (j in 1:length(players)) {
  for (k in 1:length(winners)) {
    if (winners[k] == players[j]) {
      player_freqs[j] <- player_freqs[j] + 1
    } else if (losers[k] == players[j]) {
      player_freqs[j] <- player_freqs[j] + 1
    }
  }
}
```

Now we can filter out players who have not played a sufficient number of matches. Here we remove any player that has not played at least 10 matches.

```
# filter the players to only have players with more than 10 matches
players <- names(player_freqs[player_freqs >= 10])

# Filter df2
df2 <- df2[df2$winner_name %in% players & df2$loser_name %in% players, ]

winners = df2$winner_name
losers = df2$loser_name
```

We now want to set up a dataframe with the number of times each player has beaten each other player. First, we create a matrix encoding this information.

```
# make a dataframe of 0s with the column and row names as the player names
zero_matrix <- matrix(0, nrow = length(players), ncol = length(players))

df3 <- as.data.frame(zero_matrix)

# Set the row and column names
rownames(df3) <- players
colnames(df3) <- players

# populate the matrix
for (j in 1:nrow(df2)) {
  winner <- df2$winner_name[j]
  loser <- df2$loser_name[j]
  df3[winner, loser] <- df3[winner, loser] + 1
}
```

We now want to convert this matrix into a dataframe. The rows will have unique player pairs, followed by the number of times player 1 has won, and the number of times player 2 has won.

```
# find the number of zeros in df3
count_data <- data.frame(matrix(nrow=0, ncol=4))
colnames(count_data) <- c("players", "win1", "win2")

for (i in 1:ncol(df3)) {
  for (j in i:nrow(df3)) {
    if (df3[i,j] != 0 || df3[j,i] != 0) {
      count_data <- rbind(count_data, c(players[i], players[j], df3[i,j],
      df3[j,i]))
    }
  }
}

colnames(count_data) <- c("player1", "player2", "wins1", "wins2")
```

Here, an issue arises. For the Bradley-Terry model to work, we need each of the player rows to be factors of the same level. This means that every player must appear as both player 1 and player 2 at least once in the dataframe. The following code ensures this.

```r
# switch pairs so that each column has the same players in

matches <- data.frame(count_data)

# find last occurence of the first player1
last <- max(which(matches$player1 == matches$player1[1]))

# swap player1 with player 2 and wins1 with wins2 at index last
matches[last, c("player1", "player2", "wins1", "wins2")] <- matches[last, c("player2", "player1", "wins1

missing_players = setdiff(matches$player2, matches$player1)

# swap the first occurrence of each player in missing players as above
for (i in 1:length(missing_players)) {
  last <- min(which(matches$player2 == missing_players[i]))
  matches[last, c("player1", "player2", "wins1", "wins2")] <- matches[last, c("player2", "player1", "wi
}
```

Finally, we write the dataframe to a csv file.

```r
write.csv(matches, "data/matches.csv", row.names = FALSE)
```