Elo and the English Premier League

The Elo rating system is a dynamically updated rating system originally created for chess by Arpad Elo that thrives in ranking head to head interactions with many iterations.

```
library(EloRating)
```

```
## Warning: package 'EloRating' was built under R version 4.0.2

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sna

## Warning: package 'sna' was built under R version 4.0.2

## Loading required package: statnet.common

## Warning: package 'statnet.common' was built under R version 4.0.2

##
## Attaching package: 'statnet.common'

## The following object is masked from 'package:base':
##
##     order

## Loading required package: network

## Warning: package 'network' was built under R version 4.0.2

## network: Classes for Relational Data
## Version 1.16.0 created on 2019-11-30.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                     Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Martina Morris, University of Washington
##                     Skye Bender-deMoll, University of Washington
##  For citation information, type citation("network").
##  Type help("network-package") to get started.
```

```
## sna: Tools for Social Network Analysis
## Version 2.5 created on 2019-12-09.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("sna").
##  Type help(package="sna") to get started.
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
epl <- read.csv("epl.csv")
head(epl[,1:5])
```

```
##   Div     Date       HomeTeam   AwayTeam FTHG
## 1  E0 13/08/16        Burnley    Swansea    0
## 2  E0 13/08/16 Crystal Palace  West Brom    0
## 3  E0 13/08/16        Everton  Tottenham    1
## 4  E0 13/08/16           Hull  Leicester    2
## 5  E0 13/08/16       Man City Sunderland    2
## 6  E0 13/08/16  Middlesbrough      Stoke    1
```

For the most basic application of Elo, we need to know what the result was (win, lose, draw), and who was playing.

```
epl$winner = case_when(epl$FTR == 'H' ~ as.character(epl$HomeTeam),
                           epl$FTR == 'A' ~ as.character(epl$AwayTeam),
                           epl$FTR == 'D' ~ as.character(epl$HomeTeam))
epl$loser = case_when(epl$FTR == 'A' ~ as.character(epl$HomeTeam),
                           epl$FTR == 'H' ~ as.character(epl$AwayTeam),
                           epl$FTR == 'D' ~ as.character(epl$AwayTeam))

epl$Draw = case_when(epl$FTR == "D" ~ TRUE,
                         epl$FTR != "D" ~ FALSE)
head(epl[,c('winner', 'loser', 'Draw')])
```

```
##          winner          loser  Draw
## 1       Swansea        Burnley FALSE
## 2     West Brom Crystal Palace FALSE
## 3       Everton      Tottenham  TRUE
## 4          Hull      Leicester FALSE
## 5      Man City     Sunderland FALSE
## 6 Middlesbrough          Stoke  TRUE
```

For the package we'll be using to calculate elo (EloRating), we need a winner, loser, and a Boolean column for a Draw in the next column. Also, if the Draw column is TRUE, it doesn't matter who is in the winner column vs the loser so I just put the home team in the winning column and the away team in the losing column.

Now let's filter for the columns we need.

```
epl_elo <- epl[,c('Date', 'winner', 'loser', 'Draw')]
```

Currently the Date column is in the wrong format, and is a factor.

```
epl_elo$Date <- as.Date(epl_elo$Date,"%d/%m/%Y")
epl_elo$Date <- as.character(epl_elo$Date)
substr(epl_elo$Date, 1, 2) <- "20"
epl_elo$Date <- as.Date(epl_elo$Date)
head(epl_elo[,1:4])
```

```
##         Date        winner          loser  Draw
## 1 2016-08-13       Swansea        Burnley FALSE
## 2 2016-08-13     West Brom Crystal Palace FALSE
## 3 2016-08-13       Everton      Tottenham  TRUE
## 4 2016-08-13          Hull      Leicester FALSE
## 5 2016-08-13      Man City     Sunderland FALSE
## 6 2016-08-13 Middlesbrough          Stoke  TRUE
```

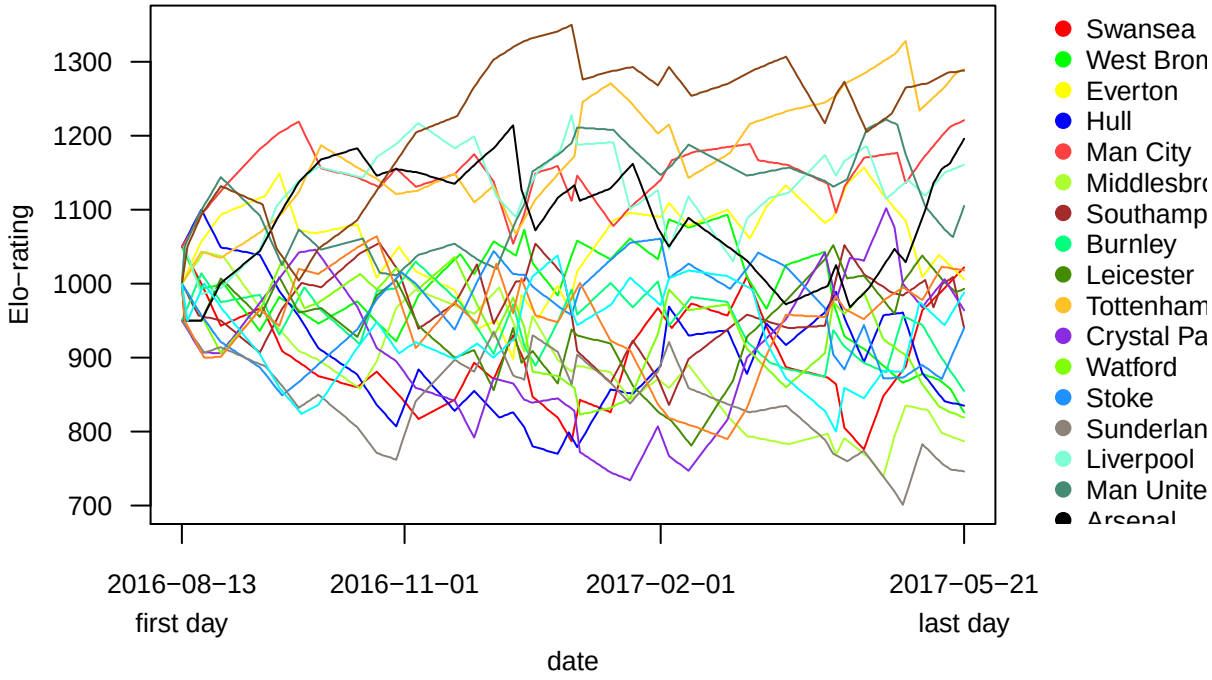
Now we have all the data in the right format. The function elo.seq returns an object with the calculated elo scores, with each team starting at 1000 points.

```
res_elo <- elo.seq(winner = epl_elo$winner, loser = epl_elo$loser, Date = epl_elo$Date, runcheck = TRUE
summary(res_elo)
```

```
## Elo ratings from 20 individuals
## total (mean/median) number of interactions: 380 (38/38)
## range of interactions: 38 - 38
## date range: 2016-08-13 - 2017-05-21
## startvalue: 1000
## uppon arrival treatment: average
## k: 100
## proportion of draws in the data set: 0.22
```

We know that 22% percent of matches last year were draws, and the date range is correct. We can use those fields to make sure the function did what we wanted. We can use the eloplot() function to look at a time series calculation for each team.

```
eloplot(res_elo)
```



This isn't the best visualization for our use case. We can do so much better. The res_elo$mat matrix has everything we'll need.

```
elo_totals <- res_elo$mat
elo_totals <- as.data.frame(elo_totals)
head(elo_totals[,1:5])
```

```
##   Swansea West Brom Everton Hull Man City
## 1    1050       1050    1000 1050     1050
## 2      NA         NA      NA   NA       NA
## 3      NA         NA      NA   NA       NA
## 4      NA         NA      NA   NA       NA
## 5      NA         NA      NA   NA       NA
## 6      NA         NA      NA   NA       NA
```

This data frame has each team's Elo score by index where the index is related to the different game days in the Premier League. Note that not every team plays on the same day, so let's add the dates to make visualization easier.

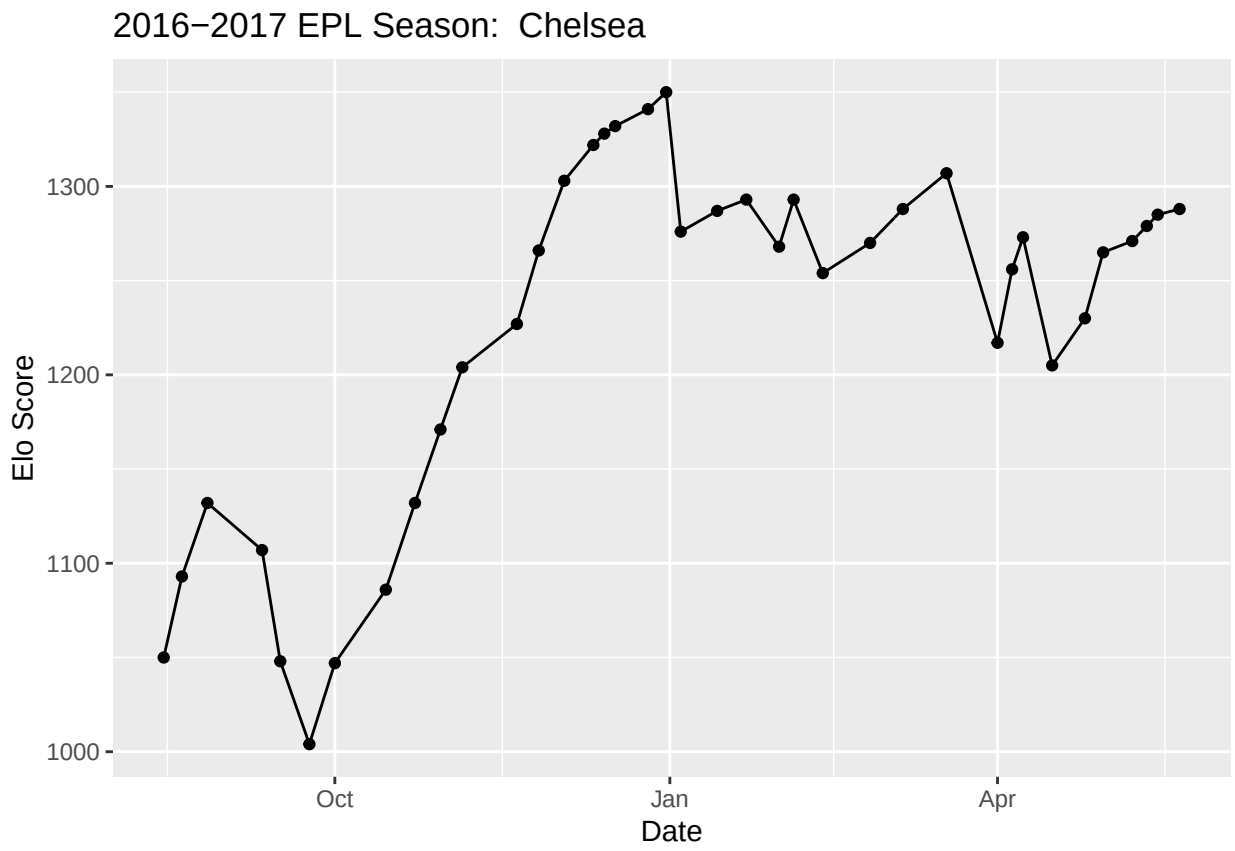
```
dates <- res_elo$truedates
elo_totals$Dates <- dates
```

Creating a function for graphing each team's performance throughout the year.

```
plotting_elo <- function(team_name){
  filtered_data <- elo_totals[,c(team_name, "Dates")]
  filtered_data <- filtered_data[!is.na(filtered_data[,team_name]),]

  x <- ggplot(data = filtered_data, aes(x = Dates, y = filtered_data[,1])) +
    geom_line() +
    ggtitle((paste("2016-2017 EPL Season: ", team_name))) +
    labs(y = "Elo Score", x = "Date") +
    geom_point()
  return(x)
}
```

Let's test it out with the winner of the 16/17 season, Chelsea.

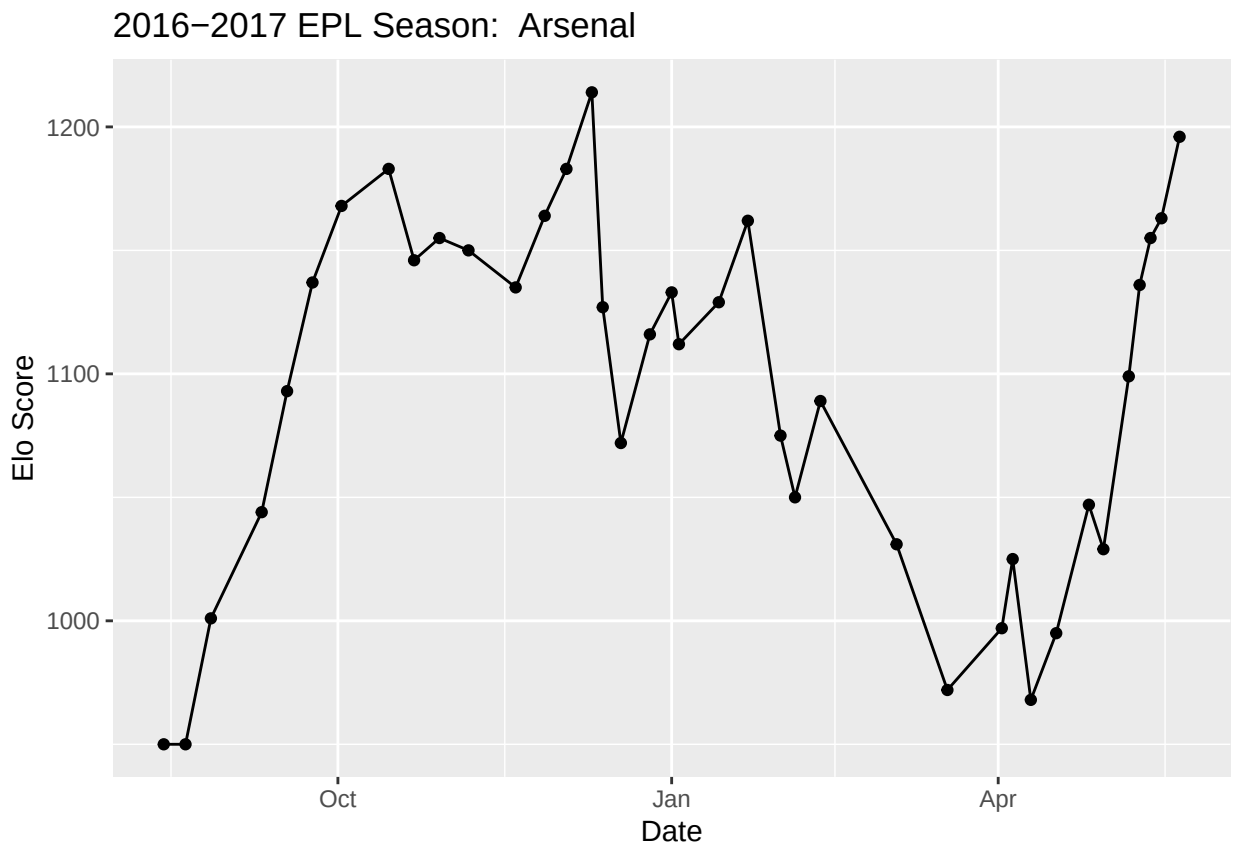
```
Chelsea_elo <- plotting_elo("Chelsea")
Chelsea_elo
```



Chelsea had a couple key losses to top talent in September to Arsenal and Liverpool, and tied a worse team (Swansea). The drop between December and January is explained by Chelsea's 2-0 loss to Tottenham.

Now let's check out the most continuously disappointing team in the league, Arsenal.

```
Arsenal_elo <- plotting_elo("Arsenal")
Arsenal_elo
```



Arsenal managed to get almost to the 1200 Elo score with their late push for the Champion's League spot but still ended far below the league champions, finishing in 5th.

How does the final Elo score compare to the final league ranking? Let's extract the elo ranking from the result of our model and compare it with the actual result.

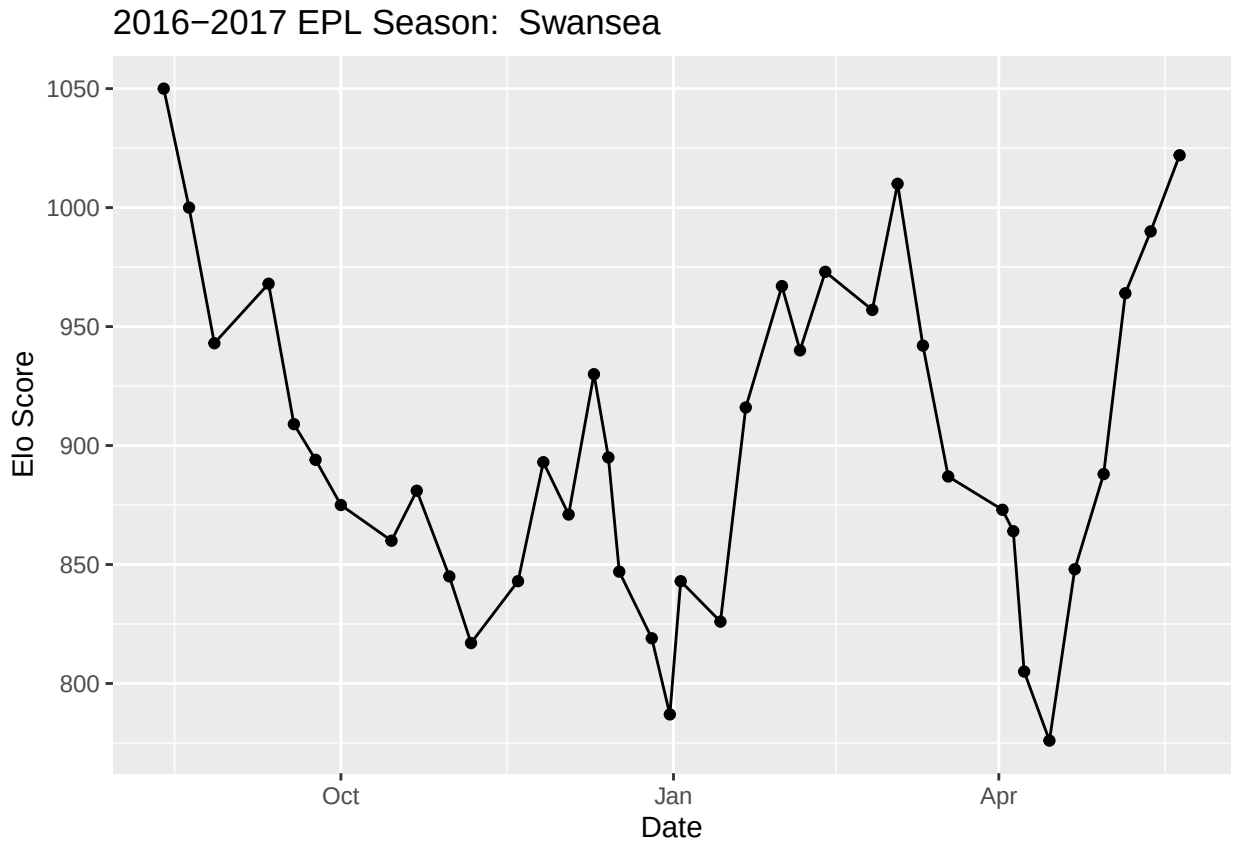
```
final_elo <- as.data.frame(extract_elo(res_elo))
teams <- rownames(final_elo)
final_elo$Team <- teams
rownames(final_elo) <- NULL
ActualFinal <- c("Chelsea", "Tottenham", "Man City", "Liverpool", "Arsenal", "Man United", "Everton", "
final_elo$ActualResult <- ActualFinal

colnames(final_elo) <- c("Elo Score", "Elo Rank", "Actual Final")
head(final_elo, 20)
```

```
##    Elo Score       Elo Rank   Actual Final
## 1       1290      Tottenham        Chelsea
## 2       1288        Chelsea      Tottenham
## 3       1221       Man City       Man City
## 4       1196        Arsenal      Liverpool
## 5       1161      Liverpool        Arsenal
## 6       1105     Man United     Man United
## 7       1022        Swansea        Everton
## 8       1018    Bournemouth    Southampton
## 9       1006        Everton    Bournemouth
```

```
## 10       993      Leicester      West Brom
## 11       988       West Ham       West Ham
## 12       964 Crystal Palace Leicester City
## 13       940    Southampton     Stoke City
## 14       940          Stoke Crystal Palace
## 15       855        Burnley   Swansea City
## 16       835           Hull        Burnley
## 17       826      West Brom        Watford
## 18       819        Watford      Hull City
## 19       787  Middlesbrough  Middlesbrough
## 20       746     Sunderland     Sunderland
```

The Elo score seems to compare fairly well to the final rankings. Note that the goal was not to predict who would win the league, but to measure the skill of each team in comparison so we should not be worried with small errors like Arsenal and Liverpool being swapped. The largest error is clearly Swansea, who is ranked highly by Elo but finished near the bottom of the league. Why would that be?

```
Swansea_elo <- plotting_elo("Swansea")
Swansea_elo
```



By early April, Swansea was ranked at 775, one of the lowest scores. However, they went on a streak, beating Stoke, Everton, Sunderland, and West Brom while tying Man United, all at the end of the season. This illustrates some of the fundamental flaws of Elo, mainly that depending on the k value we specify (we used the default value) it can shift scores in a disproportionate way compared to how much games at the end of the season matter (games at the end of the season matter more for those who have the potential to win the league, get a spot in the Champion's League, or who can get relegated). Elo is therefore overly simplistic, but can provide insight regardless.