Avengers ELO Analysis

Emanuele Lena - 142411

Avengers ELO Rating

Figure 1: https://www.flickr.com/photos/tales2astonish/6976086962

Inspirations

Dragon Ball Power Levels

TODO...

The ELO Alghorithm

$$\begin{aligned} r_i &\leftarrow r_i + \kappa(s_{i,j} - \mu_{i,j}) \\ r_j &\leftarrow r_j + \kappa(s_{j,i} - \mu_{j,i}) \end{aligned}$$

```
# library to read the excell file
library(readxl)

# library to make the dataset acceptable
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
# library to have tibbles and manipulate thes easier
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## 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
```

```
# library to manipulate strings, in particular I need it
# for regular expressions
library(stringr)

# library for graphs
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
# read all the sheets
# https://stackoverflow.com/a/12945838
read_excel_allsheets <- function(filename, tibble = TRUE) {

    sheets <- excel_sheets(filename)

    x <- lapply(sheets, function(X) read_excel(filename, sheet = X))

    if(!tibble) x <- lapply(x, as.data.frame)

    names(x) <- sheets

    return(x)
}
```

Building my Dataframe

My sample

Movies:

```
# read excell dataset
# (now I have a list of tibbles)
raw_avengers_list_of_tibble <- read_excel_allsheets("Avengers.xlsx")

names(raw_avengers_list_of_tibble)
```

```
## [1] "Captain America"    "Iron Man"           "L'incredibile Hulk"
## [4] "Thor"               "Iron Man 2"         "Avengers"
```

Dataset Structure:

```
# turn the list of tibble in one single tibble
raw_avengers_tibble <-
  bind_rows(raw_avengers_list_of_tibble, .id="movie") %>%
  # add ordered  id's
  mutate(id = row_number()) %>%
  select(id, everything())

head(raw_avengers_tibble, n=10)
```

```
## # A tibble: 10 x 7
##       id movie    winner     loser   comment        terrain_winner terrain_loser
##    <int> <chr>    <chr>      <chr>   <chr>          <chr>          <chr>
##  1     1 Captain~ 1-civilian Steve ~ scena bullo c~ land           land
##  2     2 Captain~ Bucky      1-civi~ scena bullo c~ land           land
##  3     3 Captain~ Red Skull  1-civi~ scena scopert~ land           land
##  4     4 Captain~ 3-Hydra s~ 3-US A~ scena creazio~ land           land
##  5     5 Captain~ 3-Hydra s~ Abrahm~ scena creazio~ land           land
##  6     6 Captain~ 1-US Army~ 1-Hydr~ scena creazio~ land           land
##  7     7 Captain~ Peggy Car~ 1-Hydr~ scena creazio~ land           land
##  8     8 Captain~ Captain A~ 1-Hydr~ scena creazio~ land           land
##  9     9 Captain~ Red Skull  3-nazi~ scena visita ~ land           land
## 10    10 Captain~ Captain A~ 5-Hydr~ scena liberaz~ land           land
```

The concept of fight

```
# (select only id winner and loser to print some clear samples)
rat_s <- raw_avengers_tibble %>% select(id, winner, loser)
```

I have different variants of fights:

- 1 vs 1

```
rat_s %>% filter(id==13)
```

```
## # A tibble: 1 x 3
##      id winner          loser
##   <int> <chr>           <chr>
## 1    13 Captain America Red Skull
```

- 1 vs many

```r
rat_s %>% filter(id==49)
```

```
## # A tibble: 1 x 3
##      id winner   loser
##   <int> <chr>    <chr>
## 1    49 Iron Man 11-terrorist
```

- many vs 1 (or many vs many)

```r
rat_s %>% filter(id==34 | id==30)
```

```
## # A tibble: 2 x 3
##      id winner                 loser
##   <int> <chr>                  <chr>
## 1    30 50-Hydra soldier       Captain America
## 2    34 50-US Army Soldier WW2 31-Hydra Soldier
```

- more than one winner

```r
rat_s %>% filter(id==23)
```

```
## # A tibble: 1 x 3
##      id winner                loser
##   <int> <chr>                 <chr>
## 1    23 Captain America,Bucky 1-Hydra soldier
```

- partial damange

```r
rat_s %>% filter(id==93)
```

```
## # A tibble: 1 x 3
##      id winner            loser
##   <int> <chr>             <chr>
## 1    93 5-Us Army Soldier 0.1-Hulk
```

Data Manipulations

Tools

Libraries:

```r
library(readxl) # lib to read the excell file
library(tidyr)  # lib to make to extract the (intentionally) dirty dataset
library(dplyr)  # library to have tibbles and manipulate thes easier

library(stringr)  # library to manipulate strings, in particular I need it
                  # for some regular expressions

library(ggplot2)  # library for graphs
```

Data Tiding

- Separate rows with "charater 1,charater 2,. . . " into more partial victories (w 1/n "winning rate")
- Turn values with singular charaters into "1-charater"
- separate pairs "n-charaters" in 2 columns

```r
# split rows were winner is a list of charater
raw_avengers_tibble_2 <- raw_avengers_tibble %>%
  mutate(
    divide_loser_n_by = str_count(winner, ",")+1,
  ) %>%
  separate_rows(winner, sep=",")

# replace "charater" with "1-charater" in df$winner and df$looser
raw_avengers_tibble_3 <- raw_avengers_tibble_2 %>%

  # mutate in winner
  mutate(
    winner = ifelse(
      str_detect(winner, "-"),
      winner,
      paste("1", winner, sep="-")
      )
  ) %>%

  # mutate in loser
  mutate(
    loser = ifelse(
      str_detect(loser, "-"),
      loser,
      paste("1", loser, sep="-")
      )
  )


# separate pairs "number-charater name" in 2 columns
# (in df$winner and df$looser)
fights_tidy <- raw_avengers_tibble_3 %>%

  # separate winner
  separate(winner, into=c("winner_n", "winner_charater"), sep="-") %>%
  # separate loser
  separate(loser, into=c("loser_n", "loser_charater"), sep="-") %>%

  # convert cols winner_n and loser_n in numbers
  mutate(
    winner_n = as.double(winner_n),
    loser_n = as.double(loser_n)/divide_loser_n_by
    ) %>%

  # order columns
  select(everything(), -one_of(c("divide_loser_n_by")))
```

```
## Warning: si è prodotto un NA per coercizione

## Warning: si è prodotto un NA per coercizione
```

```r
fights_tidy
```

```
## # A tibble: 409 x 9
##        id movie winner_n winner_charater loser_n loser_charater comment
##    <int> <chr>    <dbl> <chr>              <dbl> <chr>          <chr>
##  1     1 Capt~        1 civilian               1 Steve Rogers   scena ~
##  2     2 Capt~        1 Bucky                  1 civilian       scena ~
##  3     3 Capt~        1 Red Skull              1 civilian       scena ~
##  4     4 Capt~        3 Hydra soldier          3 US Army Soldi~ scena ~
##  5     5 Capt~        3 Hydra soldier          1 Abrahm Eskine  scena ~
##  6     6 Capt~        1 US Army Soldie~        1 Hydra soldier  scena ~
##  7     7 Capt~        1 Peggy Carter           1 Hydra soldier  scena ~
##  8     8 Capt~        1 Captain America        1 Hydra soldier  scena ~
##  9     9 Capt~        1 Red Skull              3 nazi soldier   scena ~
## 10    10 Capt~        1 Captain America        5 Hydra soldier  scena ~
## # ... with 399 more rows, and 2 more variables: terrain_winner <chr>,
## #   terrain_loser <chr>
```

Charater list + general statistics

```r
all_charaters <-

  # all winners
  fights_tidy %>%
  select(winner_charater) %>%
  rename(charater=winner_charater) %>%

  bind_rows(
    # all losers
    fights_tidy %>%
      select(loser_charater) %>%
      rename(charater=loser_charater)
  ) %>%
  distinct() %>%
  mutate(id=row_number()) %>%
  select(id, everything())

# save the dataset
write.csv(all_charaters, "charaters.csv")
```

```r
# statistics for each charater

# add wins statistics foreach charater
all_charaters_statistics <- fights_tidy %>%
  group_by(winner_charater) %>%
  summarise(
    n_win = n(),
  ) %>%
  rename(charater=winner_charater) %>%

  # add loss statistics foreach charater
  full_join(

    fights_tidy %>%
      group_by(loser_charater) %>%
      summarise(
        n_lose = n()
      ) %>%
      rename(charater=loser_charater),

    by= "charater"
  ) %>%

  # replace NA values with 0
  replace_na(list(n_win=0, n_lose=0)) %>%

  # calculate total fights
  mutate(n_fights = n_win + n_lose) %>%

  # arrange by fights number
  arrange(-n_fights) %>%

  # add charater type label
  left_join(all_charaters, by="charater") %>%

  # order columns
  select(charater,n_fights, n_win, n_lose)

all_charaters_statistics
```

```
## # A tibble: 83 x 4
##    charater         n_fights n_win n_lose
##    <chr>               <dbl> <dbl>  <dbl>
##  1 Iron Man               57    42     15
##  2 Hulk                   56    37     19
##  3 Loki                   52    29     23
##  4 Thor                   52    37     15
##  5 Captain America        40    31      9
##  6 Chitauri Soldier       37     7     30
##  7 Hydra soldier          37     7     30
##  8 Ice Giant              34    10     24
##  9 Abominio               26    22      4
## 10 civilian               23     4     19
## # ... with 73 more rows
```

Prepare the dataframe for ELO

Expected Format

```
fights_tidy
```

```
## # A tibble: 409 x 9
##       id movie winner_n winner_charater loser_n loser_charater comment
##    <int> <chr>    <dbl> <chr>             <dbl> <chr>          <chr>
##  1     1 Capt~        1 civilian              1 Steve Rogers   scena ~
##  2     2 Capt~        1 Bucky                 1 civilian       scena ~
##  3     3 Capt~        1 Red Skull             1 civilian       scena ~
##  4     4 Capt~        3 Hydra soldier         3 US Army Soldi~ scena ~
##  5     5 Capt~        3 Hydra soldier         1 Abrahm Eskine  scena ~
##  6     6 Capt~        1 US Army Soldie~       1 Hydra soldier  scena ~
##  7     7 Capt~        1 Peggy Carter          1 Hydra soldier  scena ~
##  8     8 Capt~        1 Captain America       1 Hydra soldier  scena ~
##  9     9 Capt~         1 Red Skull              3 nazi soldier   scena ~
## 10    10 Capt~         1 Captain America        5 Hydra soldier  scena ~
## # ... with 399 more rows, and 2 more variables: terrain_winner <chr>,
## #   terrain_loser <chr>
```

Actual Dataset Format

```
fights_tidy
```

```
## # A tibble: 409 x 9
##       id movie winner_n winner_charater loser_n loser_charater comment
##    <int> <chr>    <dbl> <chr>             <dbl> <chr>          <chr>
##  1     1 Capt~        1 civilian              1 Steve Rogers   scena ~
##  2     2 Capt~        1 Bucky                 1 civilian       scena ~
##  3     3 Capt~        1 Red Skull             1 civilian       scena ~
##  4     4 Capt~        3 Hydra soldier         3 US Army Soldi~ scena ~
##  5     5 Capt~        3 Hydra soldier         1 Abrahm Eskine  scena ~
##  6     6 Capt~        1 US Army Soldie~       1 Hydra soldier  scena ~
##  7     7 Capt~        1 Peggy Carter          1 Hydra soldier  scena ~
##  8     8 Capt~        1 Captain America       1 Hydra soldier  scena ~
##  9     9 Capt~        1 Red Skull             3 nazi soldier   scena ~
## 10    10 Capt~        1 Captain America       5 Hydra soldier  scena ~
## # ... with 399 more rows, and 2 more variables: terrain_winner <chr>,
## #   terrain_loser <chr>
```

I decided to apply this algorithm:

- Winner goes always in White, Loser goes always in Black => score will be always be a numbers in (0,1]
- given n winners and m losers, I will have…
  - m/n lines where White=winner, Black=loser, Score=1
  - (eventually) one line with White=winner, Black=loser and

$$Score = \frac{rest(\frac{m}{n})}{n} * 0.5 + 0.5$$

Dataset in ELO Format

```
# remove fights where the number of winners/losers is unknow
fights_tidy_clear <- fights_tidy %>%
  filter(!is.na(winner_n) & !is.na(loser_n))

# calculate the number of "integer" rows + extra foreach fight
fights_tidy_clear <- fights_tidy_clear %>%
  mutate(
    number_integer_rows = loser_n %/% winner_n,
    extra = (loser_n %% winner_n) / winner_n
  )


fights_elo_format_readable <-

  # fights with integer score
  fights_tidy_clear %>%
  filter(number_integer_rows > 0) %>%
  group_by(id) %>%
  expand(count = seq(1:number_integer_rows), winner=winner_charater, loser=loser_charater, score=1,
         movie=movie, comment=comment) %>%

  bind_rows(

    # fights with decimal scores
    fights_tidy_clear %>%
      filter(extra>0) %>%
      rename(
        winner = winner_charater,
        loser = loser_charater,
        score = extra,
      ) %>%
      mutate(count = number_integer_rows + 1) %>%
      select(id, count, winner, loser, score, movie, comment)
  ) %>%

  arrange(id, count)
```

```
## Warning in 1:number_integer_rows: numerical expression has 2 elements: only the
## first used

## Warning in 1:number_integer_rows: numerical expression has 2 elements: only the
## first used

## Warning in 1:number_integer_rows: numerical expression has 2 elements: only the
## first used
```

```
# make the score from (0,1] -> (0.5,1]
# (they are all victories)
fights_elo_format_readable <- fights_elo_format_readable %>%
  mutate(
    score=0.5 + score/2
  )

# insert charater ids
fights_elo_format_readable <- fights_elo_format_readable %>%

  # set the id of winner
  left_join(
    all_charaters %>% rename(winner_id = id),
    by=c("winner"="charater")
  ) %>%

  # # set the id of winner
  left_join(
    all_charaters %>% rename(loser_id = id),
    by=c("loser"="charater")
  )

fights_elo_format <- fights_elo_format_readable %>%
  # set correct var names
  rename(White=winner_id, Black=loser_id, Score=score)
```

```r
# save the dataset
write.csv(fights_elo_format, "fights_elo_format.csv")

fights_elo_format
```

```
## # A tibble: 828 x 9
## # Groups:   id [374]
##       id count winner     loser     Score movie    comment            White Black
##    <int> <dbl> <chr>      <chr>     <dbl> <chr>    <chr>              <int> <int>
##  1     1     1 civilian   Steve Ro~ 1     Captain~ scena bullo cine~      1    61
##  2     2     1 Bucky      civilian  1     Captain~ scena bullo cine~      2     1
##  3     3     1 Red Skull  civilian  1     Captain~ scena scoperta c~      3     1
##  4     4     1 Hydra sol~ US Army ~ 1     Captain~ scena creazione ~      4     8
##  5     5     1 Hydra sol~ Abrahm E~ 0.667 Captain~ scena creazione ~      4    62
##  6     6     1 US Army S~ Hydra so~ 1     Captain~ scena creazione ~      5     4
##  7     7     1 Peggy Car~ Hydra so~ 1     Captain~ scena creazione ~      6     4
##  8     8     1 Captain A~ Hydra so~ 1     Captain~ scena creazione ~      7     4
##  9     9     1 Red Skull  nazi sol~ 1     Captain~ scena visita uff~      3    63
## 10     9     2 Red Skull  nazi sol~ 1     Captain~ scena visita uff~      3    63
## # ... with 818 more rows
```

ELO Classification

Results and comments

Results

Problems

Correlation Score-Fights

Correlation Score-Screen Time

Shiny App

The ELO Algorithm

```r
##  Elo rating system
# INPUT
# games: a game *matrix* with columns White, Black and Score
#        Players are integer numbers starting at 1
#        The matrix is sorted in chronological order
# zeta: logistic parameter
# k: update factor
# OUTPUT
# r: rating vector
elo = function(games, z = 400, k = 25) {

  # number of players
  # (players are integer numbers starting at 1)
  n = max(c(games[, "White"], games[, "Black"]))

  # number of games
  m = nrow(games)

  # rating vector
  r = rep(0, n)

  # iterate through games
  for (i in 1:m) {
    score = games[i, "Score"]
    white = games[i, "White"]
    black = games[i, "Black"]

    # compute update
    spread = r[white] - r[black]
    mu = 1 / (1 + 10^(-spread / z))
    update = k * (score - mu)

    # update ratings
    r[white] = r[white] + update
    r[black] = r[black] - update

  }
  return(r)
}
```

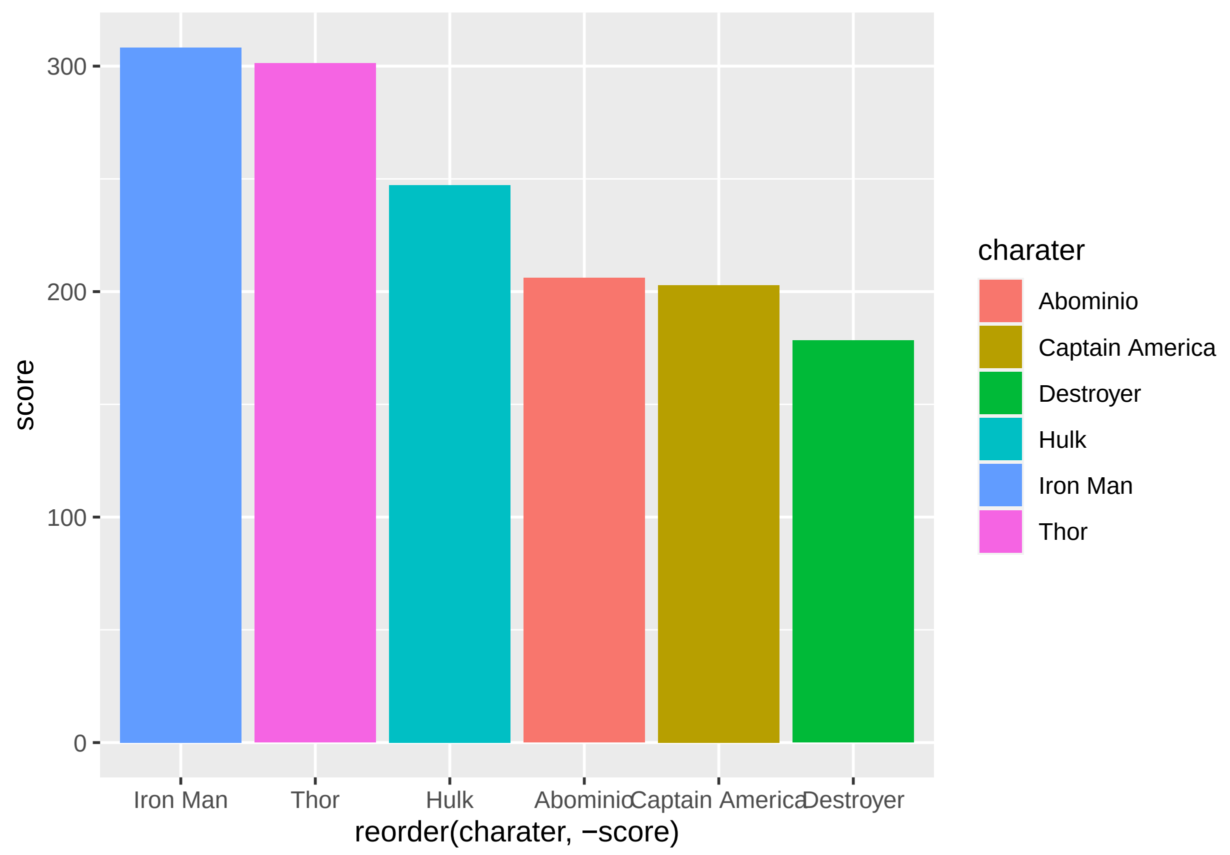
First test

```
scores_list <- elo(as.data.frame(fights_elo_format))

scores <-
  tibble(score = scores_list) %>%
  mutate(id = row_number()) %>%
  select(id, everything()) %>%
  left_join(all_charaters, by="id") %>%
  arrange(-score)



ggplot(data=scores %>% head(), mapping = aes(x=reorder(charater, -score), y=score, fill=charater)) +
  geom_bar(stat="identity")
```



```r
print(scores)
```

```
## # A tibble: 83 x 3
##       id score charater
##    <int> <dbl> <chr>
##  1    11  308. Iron Man
##  2    32  301. Thor
##  3    17  247. Hulk
##  4    25  206. Abominio
##  5     7  203. Captain America
##  6    31  178. Destroyer
##  7    49  148. Black Widow
##  8    52  145. Hawkeye
##  9    36  139. Loki
## 10    50  108. War Machine
## # ... with 73 more rows
```