Predicting with models and distributions

Narek Sahakyan

September 25, 2019

You were right Mr. Madoyan, I started on wednesday.

I am assigned to analyze the Dutch league(Eredivisie). Before starting the analysis I know that Eredivisie is considered one of the most attacking leageues in the world, so let's see if this statement is expressed by their statistics.

```
library(SportsAnalytics270)
data(f_data_sm)
```

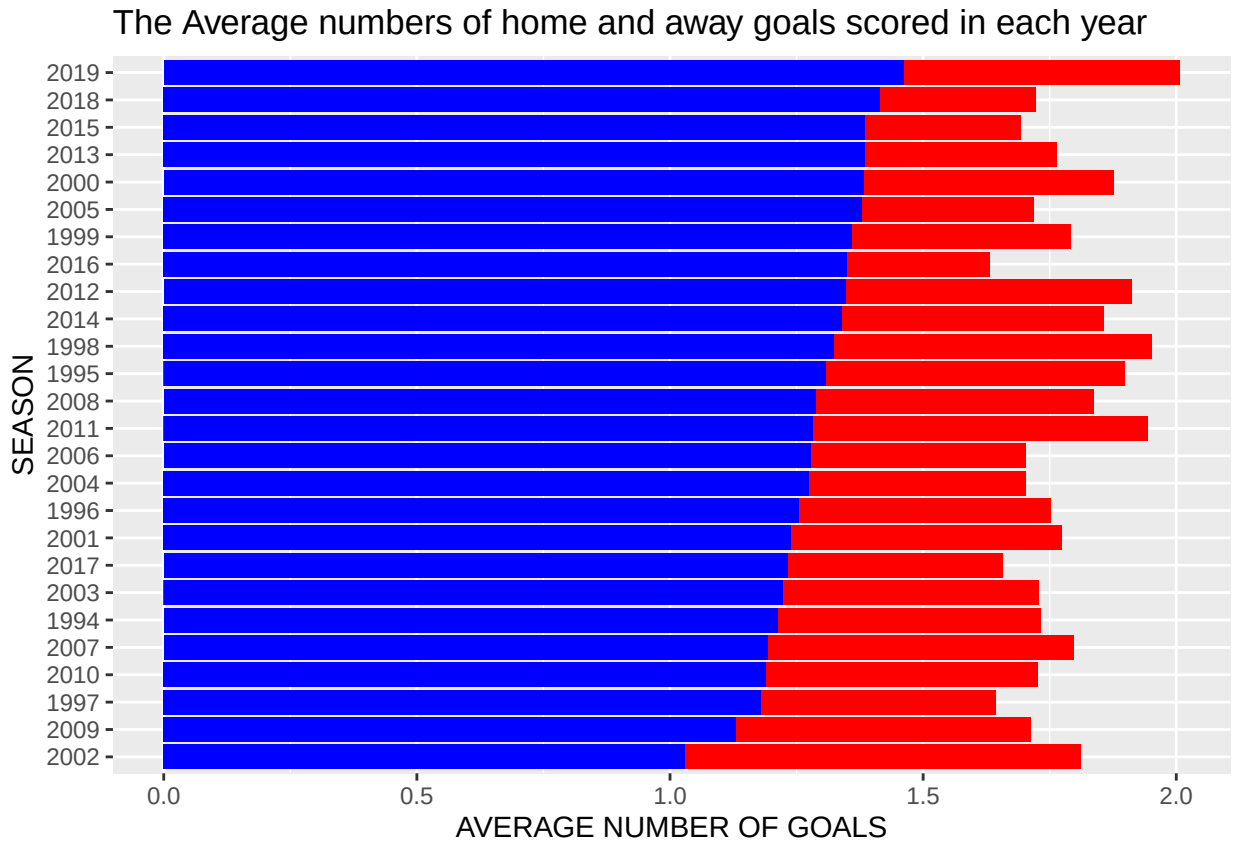
1. Calculate average number of goals for home and away teams per SEASON for that league, use dplyr. (4p)

```
eredivisie <- f_data_sm %>%
  filter(LEAGUE == "Eredivisie") %>%
    select(-c(COUNTRY, LEAGUE))

home_goals <- eredivisie %>%
  group_by(SEASON, HOMETEAM) %>%
  summarise(HOME_MEAN = mean(FTHG))

away_goals <- eredivisie %>%
  group_by(SEASON, HOMETEAM) %>%
  summarise(AWAY_MEAN = mean(FTAG))

#binding the results into a data frame for later usage
mean_goals <- home_goals
mean_goals <- rename(mean_goals, TEAM = HOMETEAM)
mean_goals$AWAY_MEAN <- away_goals$AWAY_MEAN
```

2. Construct a plot using ggplot to show how this number is changing over time. Comment Note: you need to have SEASON on x-axis. Show average Home goals and Away goals on the same plot. Be sure that your plot has appropriate axis names and title.(4p)

```
#TODO MUST BE IMPROVED !!!!!

all_mean <- eredivisie %>%
group_by(SEASON) %>%
summarise(HOME_MEAN = mean(FTHG),
          AWAY_MEAN = mean(FTAG))

all_mean %>%
  ggplot(aes(x = reorder(factor(SEASON), AWAY_MEAN))) +
  geom_bar(aes(y = HOME_MEAN),
           stat = "identity",
           fill = "red") +
  geom_bar(aes(y = AWAY_MEAN),
           stat = "identity",
           fill = "blue") +
  coord_flip() +
```

```
  labs(x = "SEASON", y = "AVERAGE NUMBER OF GOALS") +
  ggtitle("The Average numbers of home and away goals scored in each year")
```



3. Interpret the plot (4p)

As it is clearly visible from the plot, the avarege number of goals scored by the home teams was higher in every year of the provided region. 2018/2019 was the most productive season, as we can see that the highest number of average home goals scored by home and away teams occured in that period. Whereas the lowest numbers in these categories were recorded un 1997. The highest difference in average home goals and average away goals was in 2002, while the lowest ones were in 2015, 2016, 2018

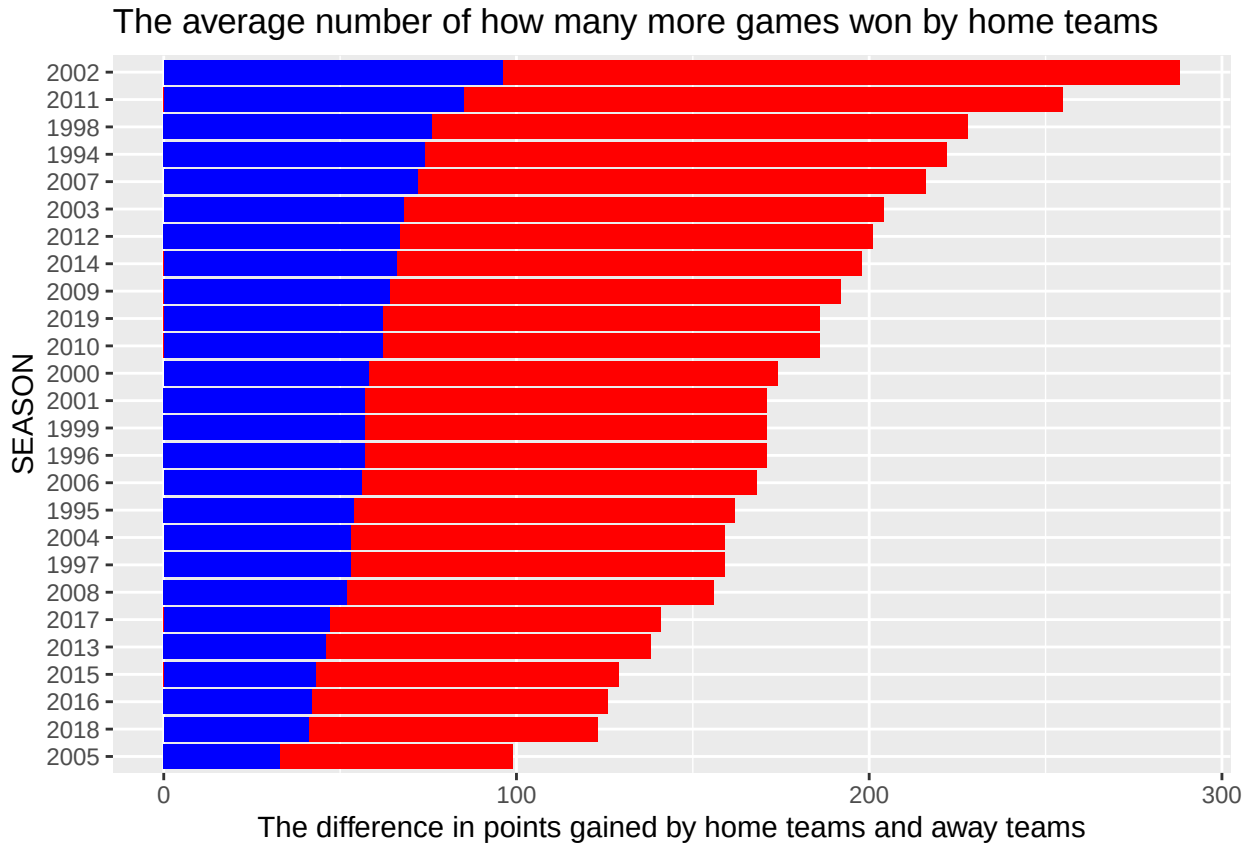
4. Think of your own approach on how will you measure home team advantage given the data you have. Calculate Home team advantage for your league and for all other leagues over time (4p).

```
#Let's add points to our data set
#Divide the points into two sections
#Points got from home games and
#Points got from away games
#and then take the differences
pick_leageue <- function(data = f_data_sm,league) {
  league_df <- f_data_sm %>%
    filter(LEAGUE == league)
}

home_advantage <- function(data){

  data_home_wins <- data %>%
```

```
    group_by(SEASON) %>%
    filter(FTR == "H") %>%
    summarise(POINTS = 3*n())

  data_home_draws <- data %>%
    group_by(SEASON) %>%
    filter(FTR == "D") %>%
    summarise(POINTS = n())

  data_home_points <- data.frame(SEASON = data_home_wins$SEASON,
                                 HOME_POINTS = data_home_wins$POINTS + data_home_draws$POINTS)

  data_away_wins <- data %>%
    group_by(SEASON) %>%
    filter(FTR == "A") %>%
    summarise(POINTS = 3*n())

  #The draw points for home and away teams are the same

  data_away_points <- data.frame(SEASON = data_away_wins$SEASON,
                                 AWAY_POINTS = data_away_wins$POINTS + data_home_draws$POINTS)

  data_points <- data_home_points
  data_points$AWAY_POINTS <- data_away_points$AWAY_POINTS

  data_points <- data_points %>%
    group_by(SEASON) %>%
    mutate(DIFF = (HOME_POINTS - AWAY_POINTS),
           MATCHES = (HOME_POINTS - AWAY_POINTS)/3 ) %>%
    arrange(desc(DIFF))
}
```

5. Plot the results of the exercise 4 here using ggplot2 and interpret it (4p).

```
ered_points <- home_advantage(eredivisie)

ered_points %>%
  ggplot(aes(x = reorder(factor(SEASON), DIFF))) +
  geom_bar(aes(y = DIFF),
           stat = "identity",
           fill = "red") +
  geom_bar(aes(y = MATCHES),
         stat = "identity",
         fill = "blue") +
  coord_flip() +
  labs(x = "SEASON", y = "The difference in points gained by home teams and away teams") +
  ggtitle("The average number of how many more games won by home teams")
```




```
#As we can se the biggest difference in the amount of
#points gained by teams at home and away was in 2002
#and the smallest difference was in 2005
#If we divide the largest difference in points gained in 2002(288)
#by 3, we get that on average home team won the game in 96 more matches
#and if we go further and divide 96/18 = 5.33333 we get that in that season
#a team won on average 5 more games at home
```

Predictions

1. With your chosen league, predict the probabilities of the first week of the season. Use distribution approach. (4p)

```
#Let's use the data from the previous season
#2018/2019 for the relevant conditions of the teams.
#However for the three teams that were promoted to the
#league we will use the last season when the teams were
#in Eredivisie

relegated <- c( "NAC Breda",
                "Excelsior",
                "Graafschap" )

promoted <- c( "Twente",
               "Waalwijk",
               "Sparta Rotterdam" )
```

```r
ered_latest <- eredivisie %>%
  filter(SEASON == 2019,
         !(HOMETEAM %in% relegated | AWAYTEAM %in% relegated))

latest_stats <- function(data, team){
  team_df <- data %>%
    filter(HOMETEAM == team | AWAYTEAM == team) %>%
    filter(SEASON == max(SEASON))

  return (as.data.frame(team_df))
}

twente <- latest_stats(eredivisie, "Twente")
waalwijk <- latest_stats(eredivisie, "Waalwijk")
sparta <- latest_stats(eredivisie, "Sparta Rotterdam")

promoted_teams <- rbind(twente,waalwijk,sparta)
ered_latest <- rbind(ered_latest, promoted_teams)

home_means_promoted <- ered_latest %>%
  filter((SEASON != "2019") & HOMETEAM %in% promoted) %>%
  group_by(HOMETEAM) %>%
  summarise(mean = mean(FTHG))

away_means_promoted <- ered_latest %>%
  filter((SEASON != "2019") & AWAYTEAM %in% promoted) %>%
  group_by(AWAYTEAM) %>%
  summarise(mean = mean(FTAG))

home_means <- ered_latest %>%
  filter(SEASON == "2019") %>%
  group_by(HOMETEAM) %>%
  summarise(mean = mean(FTHG))

away_means <- ered_latest %>%
  filter(SEASON == "2019") %>%
  group_by(AWAYTEAM) %>%
  summarise(mean = mean(FTAG))

home_means_latest <- rbind(home_means, home_means_promoted) %>%
  arrange(desc(mean))
away_means_latest <- rbind(away_means, away_means_promoted) %>%
  arrange(desc(mean))



get_probs <- function(home_means = home_means_latest,
                      away_means = away_means_latest,
                      home_team, away_team) {
  home_team_mean <- (home_means %>%
    filter(HOMETEAM == home_team))$mean
  away_team_mean <- (away_means %>%
    filter(AWAYTEAM == away_team))$mean
```

```
  options(scipen = 999)
  goal_probs_home <- dpois(c(0:9),
                           lambda = home_team_mean)
  goal_probs_away <- dpois(c(0:9),
                           lambda = away_team_mean)
  game_matrix <- goal_probs_home %*% t(goal_probs_away)
  draw_prob <- sum(diag(game_matrix))
  home_prob <- sum(game_matrix[lower.tri(game_matrix)])
  away_prob <- sum(game_matrix[upper.tri(game_matrix)])

  probs <- list(Home_Win = home_prob,
                Draw = draw_prob,
                Away_Win = away_prob)
  print(probs)

  return (probs)
}



# PEC Zwolle    -   Willem II
# Vitesse   -   AFC Ajax
# FC Emmen  -   FC Groningen
# VVV-Venlo -   RKC Waalwijk
# FC Twente -   PSV Eindhoven
# Heracles Almelo   -   sc Heerenveen
# Feyenoord -   Sparta Rotterdam
# ADO Den Haag  -   FC Utrecht
# AZ Alkmaar    -   Fortuna Sittard
```

Probabilities

```
zwolle_willem <- get_probs(home_team = "Zwolle",
                           away_team = "Willem II")
```

```
## $Home_Win
## [1] 0.4111489
##
## $Draw
## [1] 0.2049024
##
## $Away_Win
## [1] 0.3838403
```

```
vitesse_ajax <- get_probs(home_team = "Vitesse",
                          away_team = "Ajax")
```

```
## $Home_Win
## [1] 0.3683587
##
## $Draw
## [1] 0.1870864
##
## $Away_Win
## [1] 0.4441406
```

```r
emmen_groningen <- get_probs(home_team = "FC Emmen",
                             away_team = "Groningen")
```

```
## $Home_Win
## [1] 0.5615702
##
## $Draw
## [1] 0.2819127
##
## $Away_Win
## [1] 0.1565154
```

```r
venlo_rkc <- get_probs(home_team = "VVV Venlo",
                       away_team = "Waalwijk")
```

```
## $Home_Win
## [1] 0.396266
##
## $Draw
## [1] 0.2845669
##
## $Away_Win
## [1] 0.3191663
```

```r
twente_psv <- get_probs(home_team = "Twente",
                        away_team = "PSV Eindhoven")
```

```
## $Home_Win
## [1] 0.2222125
##
## $Draw
## [1] 0.1970416
##
## $Away_Win
## [1] 0.5806071
```

```r
hearcles_heerenven <- get_probs(home_team = "Heracles",
                                away_team = "Heerenveen")
```

```
## $Home_Win
## [1] 0.4815936
##
## $Draw
## [1] 0.1956763
##
## $Away_Win
## [1] 0.3225207
```

```r
feyenord_spart <- get_probs(home_team = "Feyenoord",
                            away_team = "Sparta Rotterdam")
```

```
## $Home_Win
## [1] 0.7138305
##
## $Draw
## [1] 0.1748708
##
```

```
## $Away_Win
## [1] 0.1111617
```

```
ado_utrecht <- get_probs(home_team = "Den Haag",
                         away_team = "Utrecht")
```

```
## $Home_Win
## [1] 0.4939507
##
## $Draw
## [1] 0.2116401
##
## $Away_Win
## [1] 0.2943411
```

```
az_fortuna <- get_probs(home_team = "AZ Alkmaar",
                        away_team = "For Sittard")
```

```
## $Home_Win
## [1] 0.5733045
##
## $Draw
## [1] 0.2197227
##
## $Away_Win
## [1] 0.2069382
```

2. Now use Poisson regression for the same task as above. Create the model and predict the probabilities. Interpret the home team coefficient. (6p)

Data preperation

```
#Let's use the same data frame, constructed in the previous example,
#that contains the most relevant data for all of the teams


home_games <- ered_latest %>%
  filter(SEASON == "2019") %>%
  select(c("HOMETEAM", "AWAYTEAM", "FTHG")) %>%
  mutate(Home = 1, New = 0)

away_games <- ered_latest %>%
  filter(SEASON == "2019") %>%
  select(c("AWAYTEAM", "HOMETEAM", "FTAG")) %>%
  mutate(Home = 0, New = 0)


home_games_promoted <- ered_latest %>%
  filter(SEASON != "2019" & HOMETEAM %in% promoted) %>%
  select(c("HOMETEAM", "AWAYTEAM", "FTHG")) %>%
  mutate(Home = 1, New = 1)

away_games_promoted <- ered_latest %>%
  filter(SEASON != "2019" & AWAYTEAM %in% promoted) %>%
  select(c("AWAYTEAM", "HOMETEAM", "FTAG")) %>%
```

```
  mutate(Home = 0, New = 1)

colnames(home_games) <- c("Team", "Opponent", "Goal", "Home", "Newcomer")
colnames(away_games) <- c("Team", "Opponent", "Goal", "Home", "Newcomer")
colnames(home_games_promoted) <- c("Team", "Opponent", "Goal", "Home", "Newcomer")
colnames(away_games_promoted) <- c("Team", "Opponent", "Goal", "Home", "Newcomer")

ered_double_games <- rbind(home_games,
                           away_games,
                           home_games_promoted,
                           away_games_promoted)
```

Model

```
poisson_model <- glm(Goal~Team+Opponent+Home,
                     data = ered_double_games,
                     family = poisson(link = log))

poisson_model
```

```
##
## Call:  glm(formula = Goal ~ Team + Opponent + Home, family = poisson(link = log),
##     data = ered_double_games)
##
## Coefficients:
##              (Intercept)            TeamAZ Alkmaar
##                  0.47274                -0.53806
##             TeamDen Haag              TeamFC Emmen
##                 -0.54580                -0.85177
##            TeamFeyenoord           TeamFor Sittard
##                 -0.37198                -0.81757
##            TeamGroningen            TeamHeerenveen
##                 -1.15195                -0.48484
##             TeamHeracles         TeamPSV Eindhoven
##                 -0.58337                -0.07656
##     TeamSparta Rotterdam                TeamTwente
##                 -1.08237                -0.99255
##              TeamUtrecht               TeamVitesse
##                 -0.61840                -0.45068
##            TeamVVV Venlo              TeamWaalwijk
##                 -0.83361                -0.87244
##            TeamWillem II               TeamZwolle
##                 -0.55750                -0.78815
##       OpponentAZ Alkmaar           OpponentCambuur
##                  0.09152                 0.91866
##         OpponentDen Haag         OpponentExcelsior
##                  0.60186                 0.38952
##         OpponentFC Emmen         OpponentFeyenoord
##                  0.71228                 0.10290
##      OpponentFor Sittard   OpponentGo Ahead Eagles
##                  0.77378                 0.63098
##        OpponentGroningen        OpponentHeerenveen
```

```
##                 0.13728                 0.60577
##         OpponentHeracles         OpponentNAC Breda
##                 0.65929                 0.93795
##         OpponentNijmegen     OpponentPSV Eindhoven
##                 0.63098                -0.10062
##            OpponentRoda  OpponentSparta Rotterdam
##                 0.73728                -0.34752
##           OpponentTwente           OpponentUtrecht
##                 0.17814                 0.50067
##          OpponentVitesse         OpponentVVV Venlo
##                 0.28384                 0.60033
##         OpponentWaalwijk         OpponentWillem II
##                -0.34752                 0.70669
##           OpponentZwolle                    Home
##                 0.49172                 0.32249
##
## Degrees of Freedom: 527 Total (i.e. Null);  486 Residual
## Null Deviance:       737.1
## Residual Deviance: 555   AIC: 1661
```

```
coefficients(poisson_model)
```

```
##              (Intercept)          TeamAZ Alkmaar             TeamDen Haag
##               0.47274346             -0.53805909             -0.54579570
##             TeamFC Emmen           TeamFeyenoord         TeamFor Sittard
##             -0.85177472             -0.37198488             -0.81756514
##            TeamGroningen           TeamHeerenveen            TeamHeracles
##             -1.15195375             -0.48484101             -0.58336822
##        TeamPSV Eindhoven     TeamSparta Rotterdam             TeamTwente
##             -0.07656277             -1.08237241             -0.99255299
##             TeamUtrecht             TeamVitesse           TeamVVV Venlo
##             -0.61840105             -0.45067937             -0.83360631
##            TeamWaalwijk           TeamWillem II             TeamZwolle
##             -0.87244179             -0.55750454             -0.78814894
##       OpponentAZ Alkmaar          OpponentCambuur         OpponentDen Haag
##                0.09151657              0.91865880              0.60185940
##        OpponentExcelsior         OpponentFC Emmen       OpponentFeyenoord
##                0.38952443              0.71228108              0.10290139
##      OpponentFor Sittard  OpponentGo Ahead Eagles        OpponentGroningen
##                0.77378055              0.63097673              0.13728383
##       OpponentHeerenveen         OpponentHeracles         OpponentNAC Breda
##                0.60577306              0.65928954              0.93794827
##         OpponentNijmegen    OpponentPSV Eindhoven             OpponentRoda
##                0.63097673             -0.10061880              0.73727758
## OpponentSparta Rotterdam           OpponentTwente          OpponentUtrecht
##               -0.34752436              0.17813924              0.50066670
##          OpponentVitesse        OpponentVVV Venlo         OpponentWaalwijk
##                0.28384053              0.60033077             -0.34752436
##        OpponentWillem II           OpponentZwolle                    Home
##                0.70668680              0.49172280              0.32248599
```

Probalities

```
get_probs_poisson <- function(model = poisson_model,
                              home_team,
                              away_team){
  options(scipen = 1, digits = 4)
  home_average <- predict(model,data.frame(Home = 1,
          Team = home_team,
          Opponent = away_team),
          type = "response")

  away_average <- predict(model,data.frame(Home = 0,
        Team = away_team,
        Opponent = home_team),
        type = "response")

  simulated_games <- rskellam(10000,
                              mu1 = home_average,
                              mu2 = away_average)
  home_win <- sum(dskellam(c(1:100),
                           home_average,
                           away_average))
  away_win <- sum(dskellam(c(-100:-1),
                         home_average,
                         away_average))
  draw <- sum(dskellam(0,
                       home_average,
                       away_average))

  probs <- list(Home_Win = home_win,
                Away_Win = away_win,
                Draw = draw)
  print(probs)
  return(probs)
}
```

```
zwolle_willem_poiss <- get_probs_poisson(home_team = "Zwolle",
                           away_team = "Willem II")
```

```
## $Home_Win
## [1] 0.502
##
## $Away_Win
## [1] 0.2843
##
## $Draw
## [1] 0.2137
```

```
vitesse_ajax_poiss <- get_probs_poisson(home_team = "Vitesse",
                           away_team = "Ajax")
```

```
## $Home_Win
```

```
## [1] 0.2518
##
## $Away_Win
## [1] 0.5402
##
## $Draw
## [1] 0.208
```

```
emmen_groningen_poiss <- get_probs_poisson(home_team = "FC Emmen",
                            away_team = "Groningen")
```

```
## $Home_Win
## [1] 0.3639
##
## $Away_Win
## [1] 0.3381
##
## $Draw
## [1] 0.2979
```

```
venlo_rkc_poiss <- get_probs_poisson(home_team = "VVV Venlo",
                        away_team = "Waalwijk")
```

```
## $Home_Win
## [1] 0.2064
##
## $Away_Win
## [1] 0.4921
##
## $Draw
## [1] 0.3016
```

```
twente_psv_poiss <- get_probs_poisson(home_team = "Twente",
                        away_team = "PSV Eindhoven")
```

```
## $Home_Win
## [1] 0.1518
##
## $Away_Win
## [1] 0.6208
##
## $Draw
## [1] 0.2275
```

```
hearcles_heerenven_poiss <- get_probs_poisson(home_team = "Heracles",
                                away_team = "Heerenveen")
```

```
## $Home_Win
## [1] 0.4671
##
## $Away_Win
## [1] 0.3334
##
## $Draw
## [1] 0.1996
```

```
feyenord_spart_poiss <- get_probs_poisson(home_team = "Feyenoord",
                            away_team = "Sparta Rotterdam")
```

```
## $Home_Win
## [1] 0.468
##
## $Away_Win
## [1] 0.2038
##
## $Draw
## [1] 0.3283
```

```
ado_utrecht_poiss <- get_probs_poisson(home_team = "Den Haag",
                          away_team = "Utrecht")
```

```
## $Home_Win
## [1] 0.5023
##
## $Away_Win
## [1] 0.2887
##
## $Draw
## [1] 0.2091
```

```
az_fortuna_poiss <- get_probs_poisson(home_team = "AZ Alkmaar",
                         away_team = "For Sittard")
```

```
## $Home_Win
## [1] 0.793
##
## $Away_Win
## [1] 0.07626
##
## $Draw
## [1] 0.1307
```

3. Compare the probabilities to each other and to the actual results. How good do you think you were? Can you think of a metric that can help us to understand how good are our predictions over the week (2p)

```
# PEC Zwolle    -   Willem II   1:3 (1:1)
# Vitesse   -   AFC Ajax    2:2 (1:1)
# FC Emmen  -   FC Groningen    0:1 (0:0)
# VVV-Venlo -   RKC Waalwijk    3:1 (0:1)
# FC Twente -   PSV Eindhoven   1:1 (1:0)
# Heracles Almelo   -   sc Heerenveen   0:4 (0:3)
# Feyenoord -   Sparta Rotterdam    2:2 (0:0)
# ADO Den Haag  -   FC Utrecht  2:4 (2:2)
# AZ Alkmaar    -   Fortuna Sittard 4:0 (0:0)


zwolle_willem #38% A
```

```
## $Home_Win
## [1] 0.4111
##
```

```
## $Draw
## [1] 0.2049
##
## $Away_Win
## [1] 0.3838
```

```
#dist did better
zwolle_willem_poiss #28% A
```

```
## $Home_Win
## [1] 0.502
##
## $Away_Win
## [1] 0.2843
##
## $Draw
## [1] 0.2137
```

```
vitesse_ajax #
```

```
## $Home_Win
## [1] 0.3684
##
## $Draw
## [1] 0.1871
##
## $Away_Win
## [1] 0.4441
```

```
#poisson again did better for predicting the draw
vitesse_ajax_poiss #54%-A 20%-D
```

```
## $Home_Win
## [1] 0.2518
##
## $Away_Win
## [1] 0.5402
##
## $Draw
## [1] 0.208
```

```
emmen_groningen #15%-A
```

```
## $Home_Win
## [1] 0.5616
##
## $Draw
## [1] 0.2819
##
## $Away_Win
## [1] 0.1565
```

```
#poisson
emmen_groningen_poiss #34%-A
```

```
## $Home_Win
## [1] 0.3639
##
```

```
## $Away_Win
## [1] 0.3381
##
## $Draw
## [1] 0.2979
```

```
venlo_rkc #40% -H
```

```
## $Home_Win
## [1] 0.3963
##
## $Draw
## [1] 0.2846
##
## $Away_Win
## [1] 0.3192
```

```
#dist
venlo_rkc_poiss#49% - A
```

```
## $Home_Win
## [1] 0.2064
##
## $Away_Win
## [1] 0.4921
##
## $Draw
## [1] 0.3016
```

```
twente_psv #30% -D
```

```
## $Home_Win
## [1] 0.2222
##
## $Draw
## [1] 0.197
##
## $Away_Win
## [1] 0.5806
```

```
#dist
twente_psv_poiss #20% -D
```

```
## $Home_Win
## [1] 0.1518
##
## $Away_Win
## [1] 0.6208
##
## $Draw
## [1] 0.2275
```

```
hearcles_heerenven
```

```
## $Home_Win
## [1] 0.4816
##
## $Draw
```

```
## [1] 0.1957
##
## $Away_Win
## [1] 0.3225
```

```
#both did very bad
hearcles_heerenven_poiss
```

```
## $Home_Win
## [1] 0.4671
##
## $Away_Win
## [1] 0.3334
##
## $Draw
## [1] 0.1996
```

```
feyenord_spart
```

```
## $Home_Win
## [1] 0.7138
##
## $Draw
## [1] 0.1749
##
## $Away_Win
## [1] 0.1112
```

```
#
feyenord_spart_poiss
```

```
## $Home_Win
## [1] 0.468
##
## $Away_Win
## [1] 0.2038
##
## $Draw
## [1] 0.3283
```

```
ado_utrecht
```

```
## $Home_Win
## [1] 0.494
##
## $Draw
## [1] 0.2116
##
## $Away_Win
## [1] 0.2943
```

```
#vs
ado_utrecht_poiss
```

```
## $Home_Win
## [1] 0.5023
##
## $Away_Win
```

```
## [1] 0.2887
##
## $Draw
## [1] 0.2091
```

```
az_fortuna
```

```
## $Home_Win
## [1] 0.5733
##
## $Draw
## [1] 0.2197
##
## $Away_Win
## [1] 0.2069
```

```
#vs
az_fortuna_poiss
```

```
## $Home_Win
## [1] 0.793
##
## $Away_Win
## [1] 0.07626
##
## $Draw
## [1] 0.1307
```

```
#It was 23:50 So I did not finish some parts,
#but from what I have analyzed it seems
#like predicitons with dist are more precise in predictng draws
```

Bradley terry model

You need to construct Bradley-Terry model for NBA regular season games. Dataset is nba2009_2018 from SportsAnalytics270 package

Remove the game Boston Celtics vs Indiana Pacers from the dataframe. This is a game that was cancelled due to Boston Marathon bombing

1. Do the Data Prep here (4p)

```
data("nba2009_2018")

nba_clean <- nba2009_2018 %>%
  filter(!(GAME_DATE == as.Date("2013-04-16") &
             home.TEAM_NAME == "Boston Celtics")) %>%
  mutate(result = ifelse(home.WL == "W","H","A")) %>%
  select(-c(home.WL, home.TEAM_ABBREVIATION, away.TEAM_ABBREVIATION))

nba_clean <- nba_clean %>%
  mutate(ht_w = ifelse(result == "H",1,0),
         at_w = ifelse(result == "A",1,0))

nba_filtered <- nba_clean %>%
  mutate(home.TEAM_NAME = as.factor(home.TEAM_NAME),
         away.TEAM_NAME = as.factor(away.TEAM_NAME))
```

```r
nba_win_count <- nba_filtered %>%
    group_by(home.TEAM_NAME, away.TEAM_NAME) %>%
      summarise(htw_count = sum(ht_w),
                atw_count = sum(at_w))

home_levels <- levels(nba_filtered$home.TEAM_NAME)
away_levels <- levels(nba_filtered$away.TEAM_NAME)
```

2. Model here (4p)

```r
model_1 <- BTm(cbind(htw_count,atw_count),
               home.TEAM_NAME,
               away.TEAM_NAME,
               data = nba_win_count,
               id = "team_")

summary(model_1)
```

```
##
## Call:
## BTm(outcome = cbind(htw_count, atw_count), player1 = home.TEAM_NAME,
##     player2 = away.TEAM_NAME, id = "team_", data = nba_win_count)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.471   0.135   0.697   1.287   3.423
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## team_Boston Celtics            0.1644     0.0996    1.65  0.09873 .
## team_Brooklyn Nets            -0.6122     0.1008   -6.07  1.3e-09 ***
## team_Charlotte Hornets        -0.4056     0.0995   -4.08  4.5e-05 ***
## team_Chicago Bulls             0.0248     0.0992    0.25  0.80280
## team_Cleveland Cavaliers      -0.1722     0.0992   -1.74  0.08261 .
## team_Dallas Mavericks          0.0790     0.1009    0.78  0.43337
## team_Denver Nuggets            0.1574     0.1010    1.56  0.11900
## team_Detroit Pistons          -0.4431     0.0998   -4.44  9.0e-06 ***
## team_Golden State Warriors     0.4957     0.1025    4.84  1.3e-06 ***
## team_Houston Rockets           0.4159     0.1020    4.08  4.5e-05 ***
## team_Indiana Pacers            0.0709     0.0993    0.71  0.47499
## team_LA Clippers               0.2861     0.1013    2.82  0.00476 **
## team_Los Angeles Lakers       -0.2122     0.1010   -2.10  0.03569 *
## team_Memphis Grizzlies         0.1168     0.1009    1.16  0.24713
## team_Miami Heat                0.2832     0.0998    2.84  0.00454 **
## team_Milwaukee Bucks          -0.1867     0.0992   -1.88  0.05983 .
## team_Minnesota Timberwolves   -0.5913     0.1028   -5.75  8.8e-09 ***
## team_New Orleans Pelicans     -0.2468     0.1011   -2.44  0.01465 *
## team_New York Knicks          -0.4908     0.1001   -4.90  9.4e-07 ***
## team_Oklahoma City Thunder     0.5530     0.1029    5.38  7.6e-08 ***
## team_Orlando Magic            -0.3720     0.0994   -3.74  0.00018 ***
## team_Philadelphia 76ers       -0.5364     0.1004   -5.34  9.1e-08 ***
## team_Phoenix Suns             -0.3963     0.1016   -3.90  9.5e-05 ***
## team_Portland Trail Blazers    0.2271     0.1011    2.25  0.02470 *
## team_Sacramento Kings         -0.5991     0.1028   -5.83  5.7e-09 ***
```

```
## team_San Antonio Spurs        0.7956     0.1050    7.58  3.6e-14 ***
## team_Toronto Raptors          0.0814     0.0993    0.82  0.41234
## team_Utah Jazz                0.0860     0.1009    0.85  0.39393
## team_Washington Wizards      -0.3414     0.0993   -3.44  0.00059 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1953.1  on 870  degrees of freedom
## Residual deviance: 1190.6  on 841  degrees of freedom
## AIC: 3727
##
## Number of Fisher Scoring iterations: 4
```

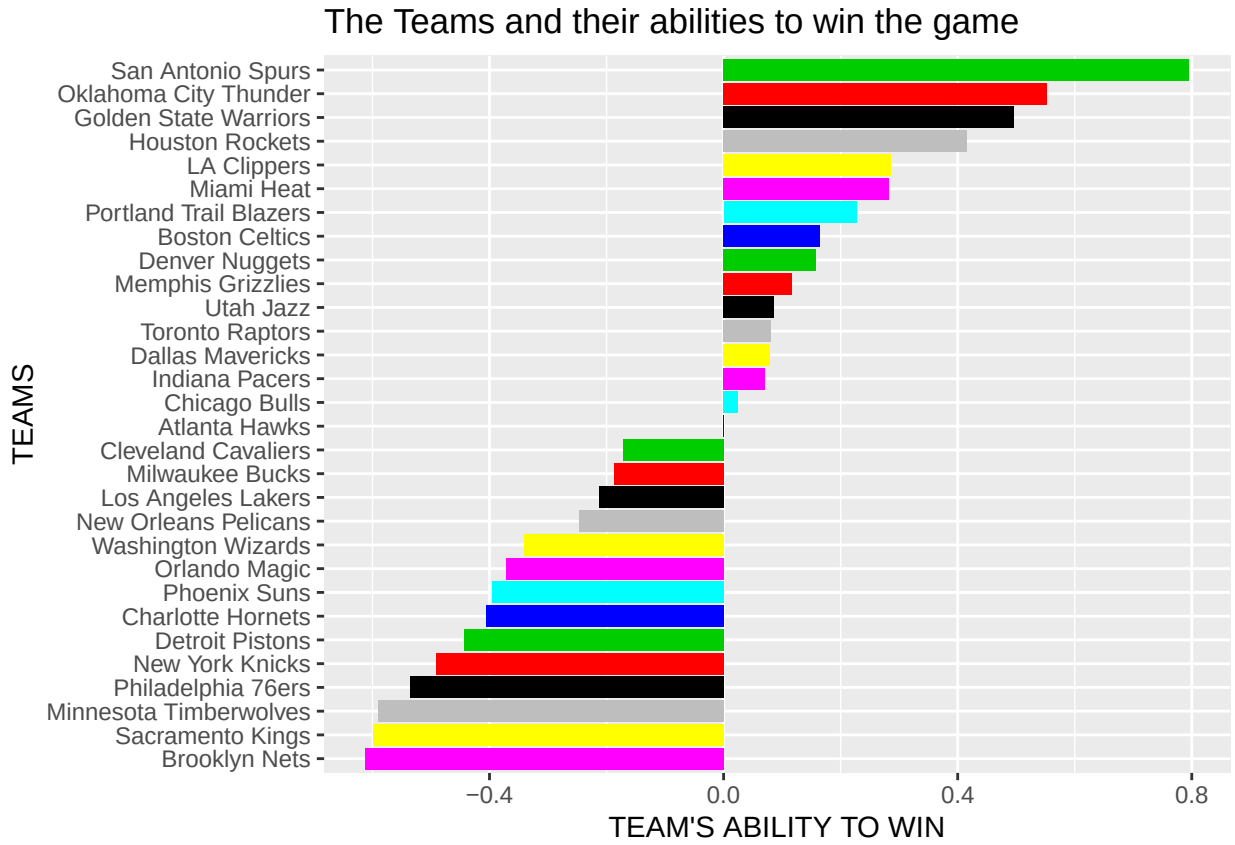
```
abilities <- as.data.frame(BTabilities(model_1))
abilities$team <- rownames(abilities)
abilities <- abilities %>%
  arrange(desc(ability))

worst_team <- (abilities %>%
  filter(ability == min(ability)))

# relevel(nba_win_count$home.TEAM_NAME, worst_team)
# relevel(nba_win_count$away.TEAM_NAME, worst_team)
#
nba_leveled <- nba_clean %>%
  mutate(home.TEAM_NAME = as.factor(home.TEAM_NAME),
         away.TEAM_NAME = as.factor(away.TEAM_NAME))

nba_leveled$home.TEAM_NAME <- relevel(nba_leveled$home.TEAM_NAME, ref = worst_team$team)
nba_leveled$away.TEAM_NAME <- relevel(nba_leveled$away.TEAM_NAME, ref = worst_team$team)


nba_win_leveled <- nba_leveled %>%
  group_by( home.TEAM_NAME, away.TEAM_NAME ) %>%
  summarise( htw_count = sum(ht_w),
             atw_count = sum(at_w))

abilities_leveled <- abilities
abilities_leveled$ability <- abilities_leveled$ability - worst_team$ability

model_2 <- BTm(cbind(htw_count,atw_count),
               home.TEAM_NAME,
               away.TEAM_NAME,
               data = nba_win_leveled,
               id = "team_")
BTabilities(model_2)
```

```
##                       ability   s.e.
## Brooklyn Nets         0.00000 0.0000
## Atlanta Hawks         0.61221 0.1008
## Boston Celtics        0.77662 0.1011
## Charlotte Hornets     0.20661 0.1010
## Chicago Bulls         0.63698 0.1009
```

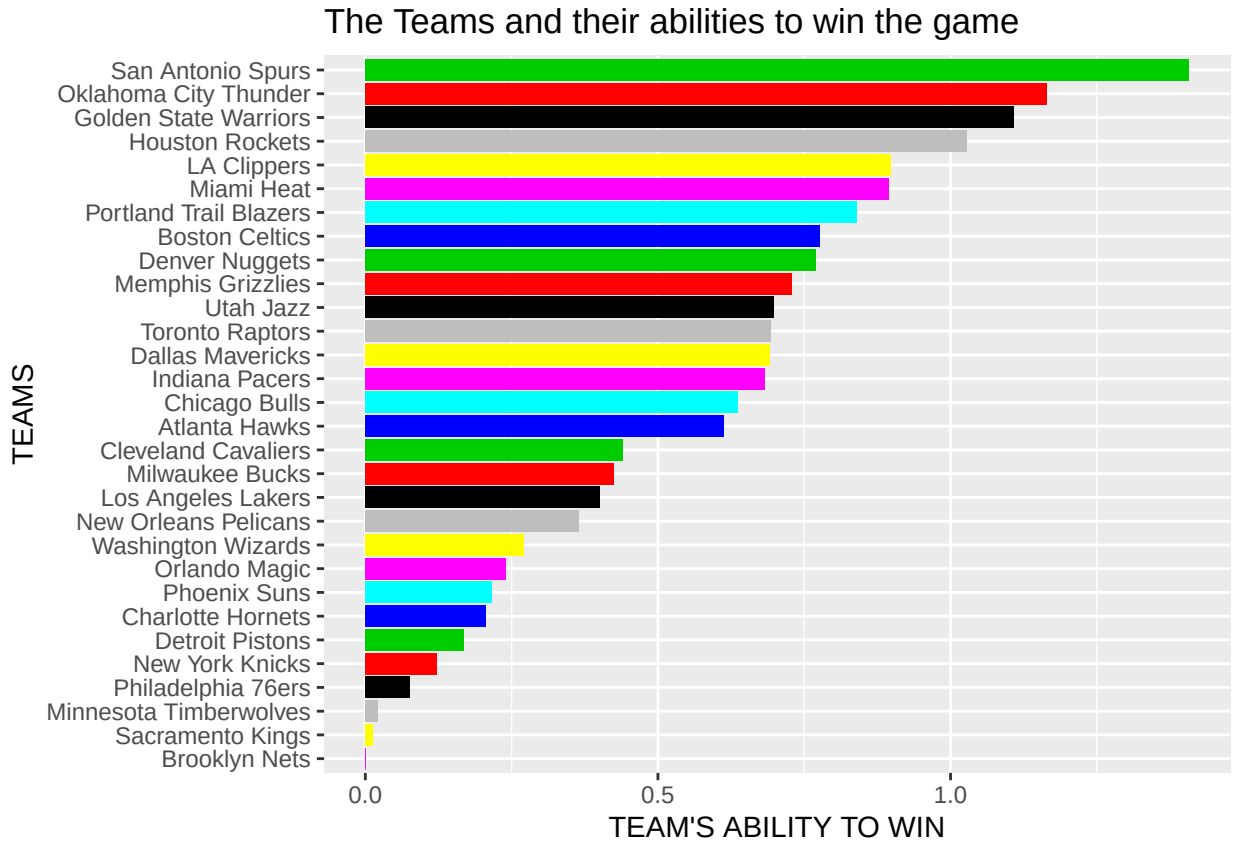
```
## Cleveland Cavaliers    0.44001 0.1005
## Dallas Mavericks       0.69122 0.1025
## Denver Nuggets         0.76966 0.1027
## Detroit Pistons        0.16916 0.1012
## Golden State Warriors  1.10788 0.1042
## Houston Rockets        1.02809 0.1037
## Indiana Pacers         0.68316 0.1010
## LA Clippers            0.89827 0.1030
## Los Angeles Lakers     0.40006 0.1025
## Memphis Grizzlies      0.72900 0.1026
## Miami Heat             0.89537 0.1018
## Milwaukee Bucks        0.42548 0.1007
## Minnesota Timberwolves 0.02095 0.1041
## New Orleans Pelicans   0.36542 0.1026
## New York Knicks        0.12140 0.1012
## Oklahoma City Thunder  1.16522 0.1046
## Orlando Magic          0.24022 0.1010
## Philadelphia 76ers     0.07581 0.1013
## Phoenix Suns           0.21590 0.1031
## Portland Trail Blazers 0.83931 0.1029
## Sacramento Kings       0.01310 0.1042
## San Antonio Spurs      1.40780 0.1068
## Toronto Raptors        0.69363 0.1008
## Utah Jazz              0.69820 0.1025
## Washington Wizards     0.27084 0.1009
```

3. Plot the abilities: Which team is the beast, which the worst ? (4p)

```r
# abilities_leveled <- abilities %>%
#   mutate(ho)

abilities %>%
  ggplot(aes(x = reorder(team,ability))) +
  geom_bar(aes(y = ability),
           stat = "identity",
           fill = 30:59) +
  coord_flip() +
  labs(x = "TEAMS", y = "TEAM'S ABILITY TO WIN") +
  ggtitle("The Teams and their abilities to win the game")
```




```
abilities_leveled <- abilities
abilities_leveled$ability <- abilities_leveled$ability - worst_team$ability

abilities_leveled %>%
  ggplot(aes(x = reorder(team, ability))) +
  geom_bar(aes(y = ability),
           stat = "identity",
           fill = 30:59) +
  coord_flip() +
  labs(x = "TEAMS", y = "TEAM'S ABILITY TO WIN") +
  ggtitle("The Teams and their abilities to win the game")
```



4. Make prediciton for the first 3 games of the new season (Schedule can be found here: https://tinyurl.com/y4hpbl8f) (6p)

```
# Wed, Oct 23, 2019 7:00p   Cleveland Cavaliers     Orlando Magic
# Wed, Oct 23, 2019 7:00p   Detroit Pistons      Indiana Pacers
# Wed, Oct 23, 2019 7:00p   Chicago Bulls        Charlotte Hornets


games <- data.frame(home.TEAM_NAME = c("Cleveland Cavaliers",
                                       "Detroit Pistons",
                                       "Chicago Bulls"),
                    away.TEAM_NAME = c("Orlando Magic",
                                       "Indiana Pacers",
                                       "Charlotte Hornets"))

home_relevel <- levels(nba_win_leveled$home.TEAM_NAME)

away_relevel <- levels(nba_win_leveled$away.TEAM_NAME)

games$home.TEAM_NAME <- factor(games$home.TEAM_NAME,
                               home_relevel)
games$away.TEAM_NAME <- factor(games$away.TEAM_NAME,
                               away_relevel)

games_prob <- predict(model_2,
                      newdata = games,
                      level = 2,
```

```
                        type = "response")

games_df <- data.frame(games,
                       ht_w = games_prob,
                       at_w = 1 - games_prob)
```