CSE 270 Sports Analytics

Homework 2

Name: Anna Martirosyan

19.10.2022

Poisson Regression

Problem 1(20 points)

Predicting football games using Poisson regression.

1. Make predictions for the same game from the same league assigned to you in homework 1, but now use the poisson regression model instead of distributions. **(8 points)**

```
data("f_data_sm")
df = f_data_sm %>% mutate(FTAG = ifelse(FTAG>=4, "4+", FTAG),
                          FTHG = ifelse(FTHG >=4, "4+", FTHG))
table(Away=df$FTAG, Home = df$FTHG)
```

```
##     Home
## Away    0    1    2    3   4+
##   0  7794 10250  8035  4395  3101
##   1  6878 11512  8530  4440  2888
##   2  4367  6029  4816  2332  1417
##   3  1925  2646  1693   916   511
##   4+ 1161  1288   828   342   164
```

```
df %>% filter(SEASON==2022) %>% group_by(COUNTRY) %>%
  summarise(pvalue=chisq.test(table(FTAG, FTHG),
                              simulate.p.value=T)$p.value)
```

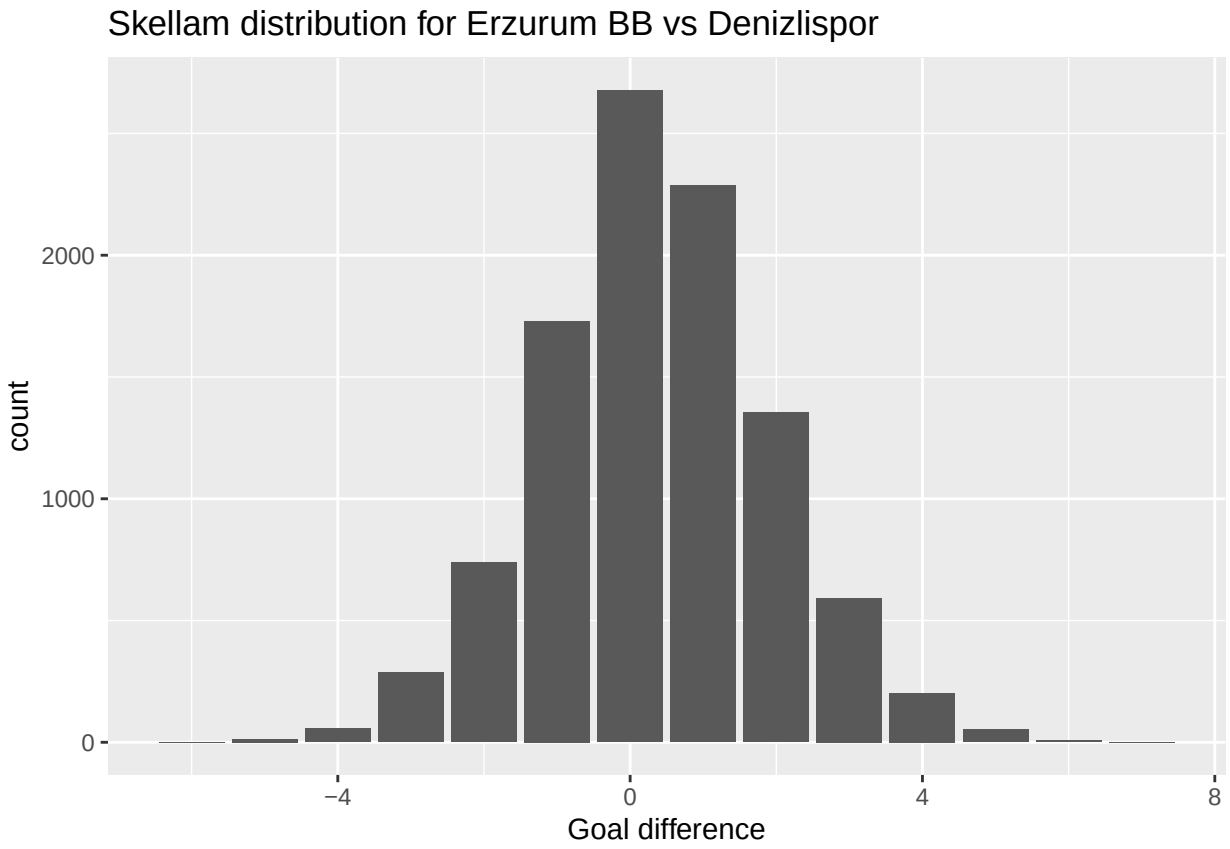
```
## # A tibble: 11 x 2
##    COUNTRY     pvalue
##    <chr>        <dbl>
##  1 Belgium     0.114
##  2 England     0.0590
##  3 France      0.355
##  4 Germany     0.0240
##  5 Greece      0.198
##  6 Italy       0.656
##  7 Netherlands 0.114
##  8 Portugal    0.389
```

```
##  9 Scotland    0.750
## 10 Spain       0.557
## 11 Turkey      0.685
```

```r
turkey = f_data_sm %>%
            filter(COUNTRY=="Turkey") %>%
            select(HOMETEAM, AWAYTEAM, FTHG, FTAG)

turk1=data.frame(turkey[,c("HOMETEAM", "AWAYTEAM", "FTHG")], Home=1)
turk2=data.frame(turkey[,c("AWAYTEAM", "HOMETEAM", "FTAG")], Home=0)
colnames(turk1)=c("Team", "Opponent", "Goal", "Home")
colnames(turk2)=c("Team", "Opponent","Goal", "Home")
turkey2=rbind(turk1, turk2)

model=glm(Goal~Team+Opponent+Home, data=turkey2, family=poisson(link=log))

options(scipen=1, digits=4)
erzurum_average <- predict(model, data.frame(Home=1, Team="Erzurum BB",
                                             Opponent="Denizlispor"),
                           type="response")
options(scipen=1, digits=4)
denizlispor_average <- predict(model, data.frame(Home=0, Team="Denizlispor",
                                           Opponent="Erzurum BB"),
                               type="response")

# After calculating the expected (average) number of goals for both teams
# we can see that expected number of goals of Erzurum BB (1.431) is higher than
# the expected (average) number of goals of Denizlispor (1.084). Therefore,
# chances of Erzurum BB winning are higher.

options(scipen=1, digits=4)
intercept <- coefficients(model)["(Intercept)"]

team_erzurum <- coefficients(model)["TeamErzurum BB"]
oponnent_denizlispor <- coefficients(model)["OpponentDenizlispor"]

home_coeff <- coefficients(model)["Home"]

erzurum_h=exp(intercept + team_erzurum + oponnent_denizlispor + home_coeff)
denizlispor_a=exp(intercept + team_erzurum + oponnent_denizlispor)
erzurum_h/denizlispor_a
```

```
## (Intercept)
##       1.347
```

```r
###### As we have lambda for both teams, we can use skellam distribution to   #######
######               come up with the probabilities.                          #######

set.seed(1)
k = rskellam(10000,mu1 = erzurum_average, mu2 = denizlispor_average)

ggplot()+geom_bar(aes(x=k))+labs(x="Goal difference",
     title="Skellam distribution for Erzurum BB vs Denizlispor")
```




```
(denizlispor_win = sum(dskellam(c(-100:-1),erzurum_average, denizlispor_average)))
```

```
## [1] 0.286
```

```
(draw = sum(dskellam(0 ,erzurum_average, denizlispor_average)))
```

```
## [1] 0.2642
```

```
(erzurum_win = sum(dskellam(c(1:100),erzurum_average, denizlispor_average)))
```

```
## [1] 0.4498
```

2. Interpret the result and compare with the betting odds from homework 1.
   Is the same team expected to win ? **(2 points)**

```
# The results that I got from Poisson Regression model and distribution approach
# are quite different. Using distribution approach I got that the expected
# winner is Denizlispor with the following probabilities:

# Erzurum BB wins - 0.3245
# Denizlispor wins - 0.3706
# Draw - 0.3049
```

```
# But while using the Poisson regression approach I got the opposite result. In
# this case instead of Denizlispor the expected winner is Erzurum BB with
# following probabilities:

# Erzurum BB wins - 0.4498
# Denizlispor wins - 0.286
# Draw - 0.2642
```

3. Define a metric to calculate the efficiency of the predictions. **(5 points)**

```
nfl_games_train =f_data_sm %>% filter(SEASON != "2022")
nfl_games_test <- f_data_sm %>% filter(SEASON == "2022")

get_poisson_form <- function(data, country ){
  nfl_home <- data %>%
    filter(COUNTRY == country) %>%
    group_by(HOMETEAM) %>%
    mutate(TEAM = as.factor(HOMETEAM),
           OPPONENT = as.factor(AWAYTEAM),
           GOAL = FTHG,
           HOME = 1) %>%
    ungroup() %>%
    select(TEAM, OPPONENT, GOAL, HOME)
  nfl_away <- data %>%
    filter(COUNTRY == country) %>%
    group_by(AWAYTEAM) %>%
    mutate(TEAM = as.factor(AWAYTEAM),
           OPPONENT = as.factor(HOMETEAM),
           GOAL = FTAG,
           HOME = 0) %>%
    ungroup() %>%
    select(TEAM, OPPONENT, GOAL, HOME)
  return(rbind(nfl_home, nfl_away)) }

nfl_train_poisson <- get_poisson_form(nfl_games_train, "Turkey")
nfl_test_poisson <- get_poisson_form(nfl_games_test, "Turkey")

nfl_model_poisson <- glm(GOAL~TEAM+OPPONENT+HOME, data=nfl_train_poisson,
                         family=poisson(link=log))

nfl_model_poisson$xlevels[["TEAM"]] <-
  union(nfl_model_poisson$xlevels[["TEAM"]], levels(nfl_test_poisson$TEAM))
nfl_model_poisson$xlevels[["OPPONENT"]] <-
  union(nfl_model_poisson$xlevels[["OPPONENT"]], levels(nfl_test_poisson$TEAM))

nfl_test_poisson$P.G <- predict(nfl_model_poisson,
                                nfl_test_poisson,
                                type = "response")
predict(nfl_model_poisson, data.frame(TEAM = "Erzurum BB",
                                      OPPONENT = "Denizlispor",
                                      HOME = 1,
                                      type = "response"))
```

```
##     1
## 0.114
```

```
rmse = sqrt(mean((nfl_test_poisson$GOAL - nfl_test_poisson$P.G)^2))
rmse
```

```
## [1] 1.261
```

```
# Lower values of RMSE indicate better fit
# RMSE for my model is 1.261
```

4. Find the under estimated and over estimated teams based on Poisson model. **(5 points)**

```
underestimated_nfl_teams <- nfl_test_poisson %>%
  filter(GOAL < P.G)
underestimated_nfl_teams
```

```
## # A tibble: 396 x 5
##    TEAM             OPPONENT       GOAL  HOME   P.G
##    <fct>            <fct>         <dbl> <dbl> <dbl>
##  1 Hatayspor        Kasimpasa         1     1 1.33
##  2 Buyuksehyr       Alanyaspor        0     1 1.32
##  3 Ad. Demirspor    Fenerbahce        0     1 0.423
##  4 Antalyaspor      Goztep            1     1 1.24
##  5 Sivasspor        Konyaspor         0     1 1.23
##  6 Yeni Malatyaspor Trabzonspor       1     1 1.38
##  7 Giresunspor      Galatasaray       0     1 0.856
##  8 Kayserispor      Ad. Demirspor     1     1 1.50
##  9 Rizespor         Karagumruk        0     1 1.84
## 10 Alanyaspor       Altay             1     1 1.39
## # ... with 386 more rows
```

```
overestimated_nfl_teams <-nfl_test_poisson %>%
  filter(GOAL > P.G)
overestimated_nfl_teams
```

```
## # A tibble: 364 x 5
##    TEAM          OPPONENT     GOAL  HOME   P.G
##    <fct>         <fct>       <dbl> <dbl> <dbl>
##  1 Besiktas      Rizespor        3     1 1.96
##  2 Karagumruk    Gaziantep       3     1 1.31
##  3 Altay         Kayserispor     3     1 0.914
##  4 Kasimpasa     Giresunspor     2     1 1.68
##  5 Konyaspor     Buyuksehyr      2     1 1.17
##  6 Fenerbahce    Antalyaspor     2     1 1.38
##  7 Trabzonspor   Sivasspor       2     1 1.33
##  8 Galatasaray   Hatayspor       2     1 1.51
##  9 Antalyaspor   Rizespor        3     1 1.31
## 10 Ad. Demirspor Konyaspor       1     1 0.572
## # ... with 354 more rows
```

Bradley terry model

Problem 2(30 points)

Construct Bradley-Terry model for NBA regular season games.

1. Load the dataset **nba2009_2021** from the package **SportsAnalytics270** and remove the game Boston Celtics vs Indiana Pacers by filtering the **home.PTS** and **away.PTS** variables to be equal to 0.**(3 points)**

```
data(nba2009_2021)
df_2 = nba2009_2021 %>%
  filter(home.PTS !=0 & away.PTS!=0)
head(df_2, n=5)
```

```
##   SEASON_ID  GAME_DATE home.TEAM_ABBREVIATION         home.TEAM_NAME home.PTS
## 1      2009 2009-10-27                    CLE    Cleveland Cavaliers       89
## 2      2009 2009-10-27                    DAL       Dallas Mavericks       91
## 3      2009 2009-10-27                    POR Portland Trail Blazers       96
## 4      2009 2009-10-27                    LAL     Los Angeles Lakers       99
## 5      2009 2009-10-28                    ATL          Atlanta Hawks      120
##   away.TEAM_ABBREVIATION     away.TEAM_NAME away.PTS home.WL
## 1                    BOS     Boston Celtics       95       L
## 2                    WAS Washington Wizards      102       L
## 3                    HOU    Houston Rockets       87       W
## 4                    LAC        LA Clippers       92       W
## 5                    IND     Indiana Pacers      109       W
```

2. Prepare the dataset for fitting a Bradley-Terry model by adding 2 variables, **ht_w** and **at_w**.Assign the values of the variables to (**ht_w** = 1, **at_w** = 0) if the home team won the game and (**ht_w** = 0, **at_w** = 1) if the away team won the game. **(5 points)**

```
df_3 = df_2 %>%
  select(home.TEAM_NAME, away.TEAM_NAME, home.WL) %>%
  mutate(ht_w = ifelse(home.WL=="W",1,0),
         at_w = ifelse(home.WL=="L",1,0))
head(df_3, n=5)
```

```
##           home.TEAM_NAME     away.TEAM_NAME home.WL ht_w at_w
## 1    Cleveland Cavaliers     Boston Celtics       L    0    1
## 2       Dallas Mavericks Washington Wizards       L    0    1
## 3 Portland Trail Blazers    Houston Rockets       W    1    0
## 4     Los Angeles Lakers        LA Clippers       W    1    0
## 5          Atlanta Hawks     Indiana Pacers       W    1    0
```

3. Convert the variables representing the home team and the away team into a factor. **(2 points)**

```
df_3$home.TEAM_NAME <- as.factor(df_3$home.TEAM_NAME)
df_3$away.TEAM_NAME <- as.factor(df_3$away.TEAM_NAME)
str(df_3)
```

```
## 'data.frame':    15428 obs. of  5 variables:
##  $ home.TEAM_NAME: Factor w/ 30 levels "Atlanta Hawks",..: 6 7 25 14 1 22 28 2 16 15 ...
##  $ away.TEAM_NAME: Factor w/ 30 levels "Atlanta Hawks",..: 2 30 11 13 12 23 6 4 20 9 ...
##  $ home.WL       : chr  "L" "L" "W" "W" ...
##  $ ht_w          : num  0 0 1 1 1 1 1 1 1 0 ...
##  $ at_w          : num  1 1 0 0 0 0 0 0 0 1 ...
```

4. Fit a Bradley-Terry model into the dataset to get the abilities of the teams to win. **(5 points)**

```
df_4 = df_3 %>%
    group_by(home.TEAM_NAME, away.TEAM_NAME) %>%
    summarise(ht=sum(ht_w), at=sum(at_w))
head(df_4, n=5)
```

```
## # A tibble: 5 x 4
## # Groups:   home.TEAM_NAME [1]
##   home.TEAM_NAME away.TEAM_NAME         ht    at
##   <fct>          <fct>               <dbl> <dbl>
## 1 Atlanta Hawks  Boston Celtics         13    10
## 2 Atlanta Hawks  Brooklyn Nets          14     9
## 3 Atlanta Hawks  Charlotte Hornets      16     8
## 4 Atlanta Hawks  Chicago Bulls          13    11
## 5 Atlanta Hawks  Cleveland Cavaliers    14     7
```

5. Plot the abilities of the teams. **(5 points)**

```
model = BTm(cbind(ht, at),
            home.TEAM_NAME,
            away.TEAM_NAME,
            data=df_4,
            id="team_")
coef=model$coefficients
sort(coef, decreasing = T)
```

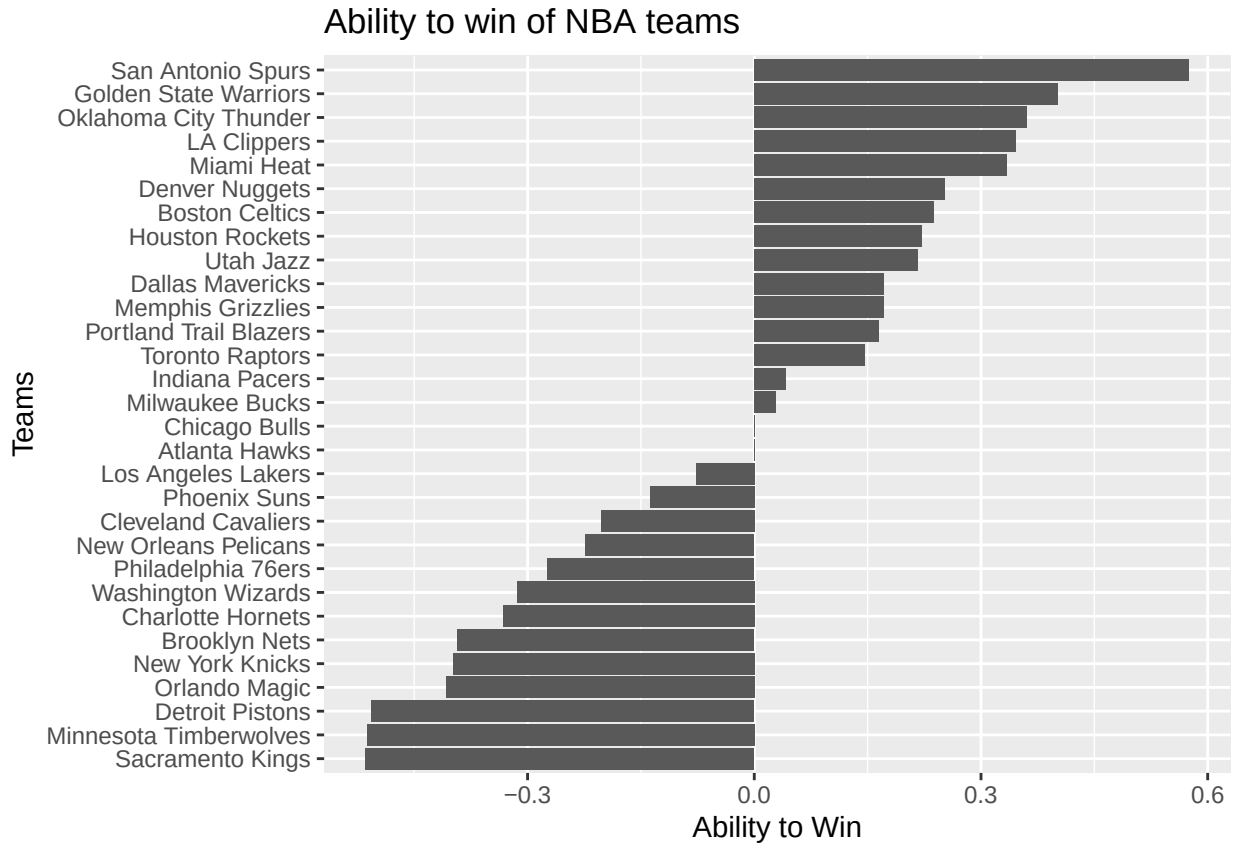
```
##      team_San Antonio Spurs  team_Golden State Warriors
##                    0.575520                    0.401244
##  team_Oklahoma City Thunder           team_LA Clippers
##                    0.360129                    0.346273
##             team_Miami Heat         team_Denver Nuggets
##                    0.333831                    0.251954
##         team_Boston Celtics        team_Houston Rockets
##                    0.237426                    0.222035
##              team_Utah Jazz       team_Dallas Mavericks
##                    0.215961                    0.171967
##      team_Memphis Grizzlies team_Portland Trail Blazers
##                    0.170909                    0.165074
##        team_Toronto Raptors         team_Indiana Pacers
##                    0.145921                    0.041565
##        team_Milwaukee Bucks          team_Chicago Bulls
##                    0.028437                    0.001287
##     team_Los Angeles Lakers           team_Phoenix Suns
##                   -0.076901                   -0.138274
```

```
##    team_Cleveland Cavaliers   team_New Orleans Pelicans
##                   -0.202993                   -0.223482
##     team_Philadelphia 76ers     team_Washington Wizards
##                   -0.273479                   -0.314223
##      team_Charlotte Hornets          team_Brooklyn Nets
##                   -0.332614                   -0.392916
##        team_New York Knicks          team_Orlando Magic
##                   -0.398761                   -0.408092
##        team_Detroit Pistons team_Minnesota Timberwolves
##                   -0.506545                   -0.512332
##       team_Sacramento Kings
##                   -0.514718
```

```
BTabilities(model)
```

```
##                          ability    s.e.
## Atlanta Hawks           0.000000 0.00000
## Boston Celtics          0.237426 0.08778
## Brooklyn Nets          -0.392916 0.08795
## Charlotte Hornets      -0.332614 0.08775
## Chicago Bulls           0.001287 0.08745
## Cleveland Cavaliers    -0.202993 0.08765
## Dallas Mavericks        0.171967 0.08868
## Denver Nuggets          0.251954 0.08891
## Detroit Pistons        -0.506545 0.08858
## Golden State Warriors   0.401244 0.08966
## Houston Rockets         0.222035 0.08886
## Indiana Pacers          0.041565 0.08734
## LA Clippers             0.346273 0.08920
## Los Angeles Lakers     -0.076901 0.08867
## Memphis Grizzlies       0.170909 0.08873
## Miami Heat              0.333831 0.08790
## Milwaukee Bucks         0.028437 0.08739
## Minnesota Timberwolves -0.512332 0.09030
## New Orleans Pelicans   -0.223482 0.08900
## New York Knicks        -0.398761 0.08809
## Oklahoma City Thunder   0.360129 0.08933
## Orlando Magic          -0.408092 0.08783
## Philadelphia 76ers     -0.273479 0.08764
## Phoenix Suns           -0.138274 0.08867
## Portland Trail Blazers  0.165074 0.08865
## Sacramento Kings       -0.514718 0.09018
## San Antonio Spurs       0.575520 0.09040
## Toronto Raptors         0.145921 0.08751
## Utah Jazz               0.215961 0.08888
## Washington Wizards     -0.314223 0.08756
```

```
abilities = as.data.frame(BTabilities(model))
abilities$team = rownames(abilities)
abilities = abilities[order(abilities$ability, decreasing = T),]

ggplot(data=abilities, aes(x = reorder(team, ability), y = ability)) +
  geom_bar(stat="identity") + coord_flip() +
  labs(x="Teams", y="Ability to Win", title="Ability to win of NBA teams")
```



6. Interpret the results of the plot, which team is the best and which team is the worst ? Why is the ability to win for team **Atlanta Hawks** equal to 0 ? **(5 points)**

```
# San Antonio Spurs: the strongest team
# Sacramento Kings : the weakest team
# The ability to win for team Atlanta Hawks is equal to 0, because the results
# were calculated with respect to that team.
```

7. Make a prediction for the upcoming 3 games.
(Schedule can be found here: https://www.nba.com/schedule) **(5 points)**

```
a_hawks=data.frame(home.TEAM_NAME=rep("Atlanta Hawks", 3),
                   away.TEAM_NAME=c("Cleveland Cavaliers",
                                    "New Orleans Pelicans",
                                    "Detroit Pistons"))

a_hawks$home.TEAM_NAME=factor(a_hawks$home.TEAM_NAME,
                              levels(df_4$away.TEAM_NAME))
a_hawks$away.TEAM_NAME=factor(a_hawks$away.TEAM_NAME,
                              levels(df_4$away.TEAM_NAME))

a_hawks_prob = predict(model, newdata=a_hawks, level=2, type="response")

(a_hawks_df = data.frame(a_hawks,
```

```
                          ht_w=a_hawks_prob,
                          at_w=1-a_hawks_prob))
```

```
##   home.TEAM_NAME       away.TEAM_NAME   ht_w   at_w
## 1  Atlanta Hawks  Cleveland Cavaliers 0.5506 0.4494
## 2  Atlanta Hawks New Orleans Pelicans 0.5556 0.4444
## 3  Atlanta Hawks      Detroit Pistons 0.6240 0.3760
```

Winning Percentage

Pythagorean wins for european football

Problem 3 (30 points)

1. Library **SportsAnalytics270** has a function final_table. It creates final league standing for the season. Use this function to create a dataframe with final standings of all seasons of your league. Combine seasons into 1 dataframe. You need to get something like nba_east. **(10 points)**

```
final_df = data.frame()

for(i in unique(f_data_sm[f_data_sm$COUNTRY == "Turkey",]$SEASON)) {
  output = final_table(f_data_sm, "Turkey", i)
  output$Season = i
  final_df = rbind(final_df, output)
}
head(final_df, n=5)
```

```
##             TEAM  M  W D L GF GA DIFF POINTS POSITION Season
## 1       Besiktas 34 24 7 3 80 26   54     79        1   1995
## 2    Trabzonspor 34 23 7 4 80 28   52     76        2   1995
## 3    Galatasaray 34 21 6 7 76 38   38     69        3   1995
## 4     Fenerbahce 34 20 7 7 78 35   43     67        4   1995
## 5 Genclerbirligi 34 17 8 9 60 45   15     59        5   1995
```

2. Create a variable for winning percentage and goal difference. Note, as there are draws in football, we are going to take draw as a half win. **(2 points)**

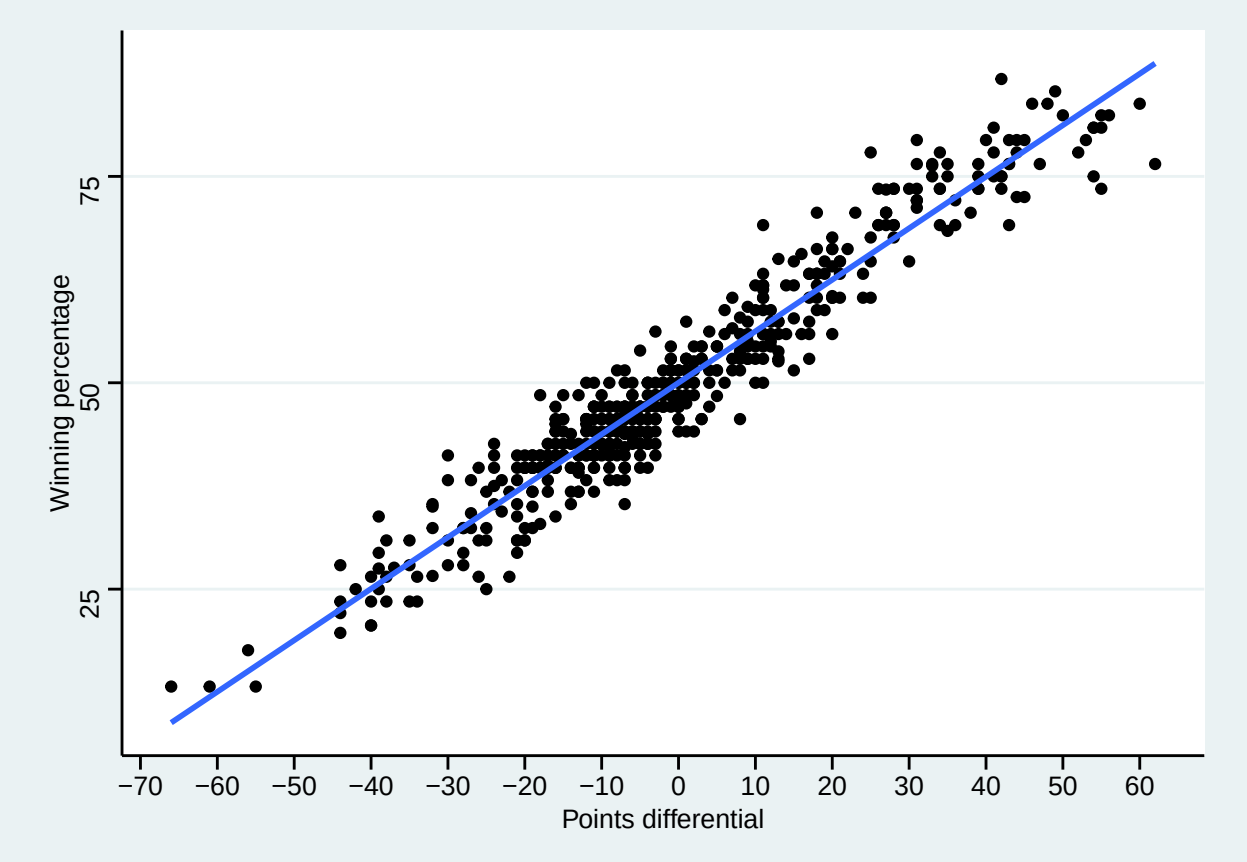
```
final_df$WP =
  as.numeric(format(round(((final_df$W + final_df$D/2)/(final_df$W+final_df$D+
                                                        final_df$L))*100,1), nsmall = 1))
head(final_df, n=5)
```

```
##             TEAM  M  W D L GF GA DIFF POINTS POSITION Season   WP
## 1       Besiktas 34 24 7 3 80 26   54     79        1   1995 80.9
## 2    Trabzonspor 34 23 7 4 80 28   52     76        2   1995 77.9
## 3    Galatasaray 34 21 6 7 76 38   38     69        3   1995 70.6
## 4     Fenerbahce 34 20 7 7 78 35   43     67        4   1995 69.1
## 5 Genclerbirligi 34 17 8 9 60 45   15     59        5   1995 61.8
```

```
# The column DIFF is already the difference of GF and GA
```

3. Plot Goal differential against Winning Percentage with a regression line.
   Make sure to have a title and appropriate axis labels for your plot. **(5 points)**

```
library(ggthemes)
ggplot(final_df, aes(x=DIFF, y=WP))+geom_point()+
  geom_smooth(method = "lm", se=F)+
  labs(x="Points differential",y="Winning percentage")+
  scale_x_continuous(breaks = seq(-70, 70, 10))+
  theme_stata()
```



4. Interpret the plot, do you think there is a strong correlation between the variables ? **(3 points)**

```
# Yes, it is nearly linear correlation between Goal differential and
# Winning Percentage. If the points difference is near 0 the winning percentage
# is near 50% which is quite expected, because if the difference is 0, means
# the game ended up with home and away team scoring the same number of goals.
```

5. Calculate Pearson correlation coefficient for Goal differential and Winning Percentage. **(2 points)**

```
(cor(final_df$WP, final_df$DIFF,method = "pearson"))
```

```
## [1] 0.9624
```

6. Build a regression model to estimate the value for k from the Pythagorean formula **(3 points)**

```
final_df_new = final_df %>%
  filter(W!=0, L!=0, GA!=0) %>%
  mutate(W = W + 0.5* T, L=W + 0.5* T, RATIO=GF/GA)

model2 <- lm(log(W/L)~0+log(RATIO), data=final_df_new)
summary(model2)
```

```
##
## Call:
## lm(formula = log(W/L) ~ 0 + log(RATIO), data = final_df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.1618 -0.0470 -0.0401 -0.0353 -0.0227
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## log(RATIO)  0.03539    0.00434    8.16  2.7e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0446 on 507 degrees of freedom
## Multiple R-squared:  0.116,  Adjusted R-squared:  0.114
## F-statistic: 66.5 on 1 and 507 DF,  p-value: 2.73e-15
```

```
## The estimated value of K is 0.03539 approximately 0.04
```

5. Using the estimated value for k, calculate Pythagorean Winning percentage and Pythagorean wins for each team. **(3 points)**

```
k = model2$coefficients[1]
final_df = final_df %>%
       mutate(PWPCT = (GF ^ k / (GF^k + GA ^ k)), PW = (W + L + T) * PWPCT)
head(final_df, 5)
```

```
##             TEAM  M  W D L GF GA DIFF POINTS POSITION Season   WP  PWPCT    PW
## 1       Besiktas 34 24 7 3 80 26   54     79        1   1995 80.9 0.5099 14.28
## 2    Trabzonspor 34 23 7 4 80 28   52     76        2   1995 77.9 0.5093 14.26
## 3    Galatasaray 34 21 6 7 76 38   38     69        3   1995 70.6 0.5061 14.68
## 4     Fenerbahce 34 20 7 7 78 35   43     67        4   1995 69.1 0.5071 14.20
## 5 Genclerbirligi 34 17 8 9 60 45   15     59        5   1995 61.8 0.5025 13.57
```

6. Find over performing and under performing teams. **(2 points)**

```
#over_performing <- final_df %>%
#  filter(GOAL < P.G)
#under_performing <- final_df %>%
#  filter(GOAL > P.G)
```

Elo Ratings

Problem 4 (20 points)

Building Elo rating model for NBA games.

1. Calculate the relative Elo ratings for all the games in the nba dataset.
   Make sure to adjust the ratings of the teams at the beginning of each season.
   This link https://fivethirtyeight.com/features/how-we-calculate-nba-elo-ratings/ shows the steps for calculating elo ratings in nba. **(10 points)**

```
data(nba2009_2021)
home_court_advantage <- 100
k_nba <- 20
starting_elos <- 1505

calculate_elos <- function(data, home_factor, k, starting_elos){
  relative_elos <- data.frame()
  final_elos <- data.frame()
  for (season in unique(nba2009_2021$SEASON_ID)){
  seasonal_data <- data %>%
    filter(SEASON_ID == season)
  season_elos <- elo.run(score(home.PTS, away.PTS) ~
          adjust(home.TEAM_NAME, home_court_advantage) + away.TEAM_NAME,
          data = seasonal_data, k = k,
          initial.elos = starting_elos
        )
  starting_elos <-0.75 * final.elos(season_elos) + (0.25 * 1505)
  relative_elos <- rbind(relative_elos,
                         data.frame(season_elos, SEASON_ID = season))
  final_elos_df <- data.frame(season = season,
                              team_abr = names(starting_elos),
                              elo = starting_elos)
  final_elos <- rbind(final_elos, final_elos_df)

  }
  return (list(final_elos=final_elos, relative_elos=relative_elos))
}

nba_relative_elos <- calculate_elos(nba2009_2021, home_court_advantage, k_nba,
                                    starting_elos)$relative_elos
nba_final <- calculate_elos(nba2009_2021, home_court_advantage, k_nba,
                            starting_elos)$final_elos
nba_relative_elos$Predicted <- ifelse(nba_relative_elos$p.A > 0.5, 1,0)
head(nba_relative_elos, n=5)
```

```
##                   team.A             team.B    p.A wins.A update.A update.B
## 1    Cleveland Cavaliers     Boston Celtics 0.6401      0  -12.801   12.801
## 2       Dallas Mavericks Washington Wizards 0.6401      0  -12.801   12.801
## 3 Portland Trail Blazers    Houston Rockets 0.6401      1    7.199   -7.199
## 4     Los Angeles Lakers        LA Clippers 0.6401      1    7.199   -7.199
## 5          Atlanta Hawks     Indiana Pacers 0.6401      1    7.199   -7.199
##   elo.A elo.B SEASON_ID Predicted
## 1  1492  1518      2009         1
## 2  1492  1518      2009         1
## 3  1512  1498      2009         1
## 4  1512  1498      2009         1
## 5  1512  1498      2009         1
```

2. Interpret the predictive power of your model using either confusion matrix or brier score. **(5 points)**

```
library(caret)
calculate_brier <- function(data){
  diff <- data$p.A - data$wins.A
  accuracy <- sum(diff^2) / nrow(data)
  return(accuracy)
}
(caret::confusionMatrix(data = factor(nba_relative_elos$Predicted),
                        reference = factor(nba_relative_elos$wins.A)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0  0.5    1
##        0   2545    0 1391
##        0.5    0    0    0
##        1   3933    1 7559
##
## Overall Statistics
##
##                Accuracy : 0.655
##                  95% CI : (0.647, 0.662)
##     No Information Rate : 0.58
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.251
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 0.5 Class: 1
## Sensitivity             0.393  0.0000000    0.845
## Specificity             0.845  1.0000000    0.393
## Pos Pred Value          0.647        NaN    0.658
## Neg Pred Value          0.658  0.9999352    0.647
## Prevalence              0.420  0.0000648    0.580
## Detection Rate          0.165  0.0000000    0.490
## Detection Prevalence    0.255  0.0000000    0.745
## Balanced Accuracy       0.619  0.5000000    0.619
```

3. Make predictions for the same games you chose in **Problem 2.7**, but now use Elo rating model. **(3 points)**

```
teams = nba_final %>%
  filter(team_abr == "Atlanta Hawks" & season=="2021")
teams
```

```
##                  season      team_abr  elo
## Atlanta Hawks12   2021 Atlanta Hawks 1538
```

4. Compare the results **(2 points)**

5. Try to play with the home advantage parameter, K value or anything else, either on football data or in nba and check if that will help to improve your models' predictive power. **(Bonus question 10 points)**