title: "UEFA Championship Win Probabilities"
author: "Soren Johansen"
date: "27 3 2020"
output: html_document

Understand the problem

The problem I want to model is a about European soccer. The hypothesis I want to test is if there is a home advantage during European UEFA Championship and also to see if decompose the teams into which is more attach versus more defense playing strategy. The data I have used is from Kaggle. The model I have used is called the Poisson random effects model.

We indicate the number of points scored by the home and the away team in the g-th game of the season (175 games) as y_{g1} and y_{g2} respectively.

The vector of observed counts y=(y_{g1}, y_{g2}) is modelled as independent Poisson: $y_{gj}|\theta_{gj}\ Possion(\theta_{gj})$ playing at home (j=1) and away (j=2), respectively. We model these parameters according to a formulation that has been used widely in the statistical literature, assuming a log-linear random effect model:

$$log\theta_{g1} = home + att_{h(g)} + def_{a(g)}$$

$$log\theta_{g2} = att_{a(g)} + def_{h(g)}$$

A more detailed study of the model can be found here (https://discovery.ucl.ac.uk/id/eprint/16040/1/16040.pdf) with Italian football(Serie A championship 1991-1992) and an implementation in pymc3 is here (https://docs.pymc.io/notebooks/rugby_analytics.html) on Rugby data. During the European Championship every team is assigned to specific football stadium and will play all their matches at the same stadium in the first round of the tournament. The notion of home advantage does make sense.

Plan and properly collect relevant data

The dataset includes 41,586 results of international football matches starting from the very first official match in 1972 up to 2019. The matches range from FIFA World Cup to FIFI World Cup to regular friendly matches. The data can be found here (https://www.kaggle.com/martj42/international-football-results-from-1872-to-2017). I will only analyze the UEFA European Championship from year 2000:

```
Soccer <- read.csv("~/Downloads/results.csv")
euro <- Soccer %>%
    filter(tournament=="UEFA Euro") %>%
    filter(year(date)>=2000)
```

This gives me:

- 175 matches (nrow(euro))
- 32 different European countries over
- UEFA Championships in 2000(Belgium/Netherlands), 2004(Portugal), 2008(Austria/Switzerland), 2012(Poland/Ukraine) and 2016(France)

Explore data

Some descriptive statistics:

```
summary(euro$home_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   0.000   1.000   1.314   2.000   6.000
```

```
summary(euro$away_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   0.000   1.000   1.103   2.000   4.000
```

and standard deviation

```
sd(euro$home_score)
```

```
## [1] 1.231097
```

```
sd(euro$away_score)
```

```
## [1] 0.9830418
```

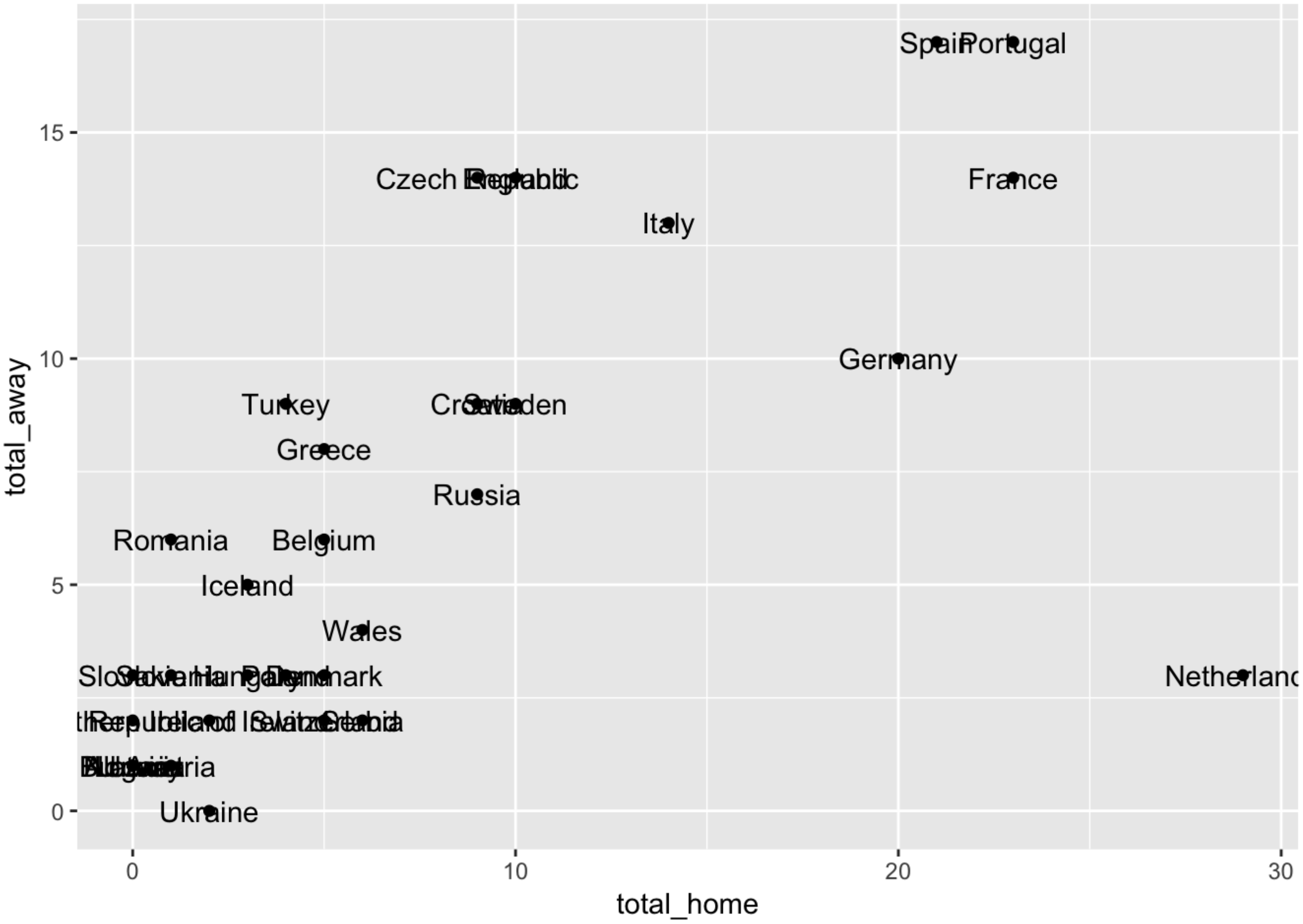
I see that on average the home team scores more than the away team but that the standard deviation for home teams is higher than the away teams. The sum of all home scores and away scores can be visualized as this:

Total home and away scores

```
home_sum_scores<-euro %>%
  group_by(home_team) %>% summarise(total_home = sum(home_score))

away_sum_scores<-euro %>%
  group_by(away_team) %>% summarise(total_away = sum(away_score))
home_away_sum<-cbind(home_sum_scores,away_sum_scores)
```

```
f <- ggplot(home_away_sum, aes(total_home,total_away))
f + geom_text(aes(label=home_team))+geom_point()
```



Interesting to see the cluster of teams in upper right corner e.g. Spain, Portugal France, Germany.

Postulate a model

The model I use, has already been presented above. Here is the jags code:

```
mod_string = " model {
for (i in 1:length(date)) {
  home_score[i] ~ dpois(theta_home[i])
  away_score[i] ~ dpois(theta_away[i])
  theta_home[i] = exp(home+att[home_ind[i]]+def[away_ind[i]])
  theta_away[i] = exp(att[away_ind[i]]+def[home_ind[i]])
  }

for (i in 1:max(home_ind)) {
  att_star[i] ~ dnorm(mu_att, tau_att)
  def_star[i] ~ dnorm(mu_def, tau_def)
  att[i]=att_star[i]-mean(att_star)
  def[i]=def_star[i]-mean(def_star)
}

mu_att ~ dnorm(0.0,1000)
tau_att ~ dgamma(0.01,0.01)
mu_def ~ dnorm(0.0,1000)
tau_def ~ dgamma(0.01, 0.01)
home ~ dnorm(0.0,1)
} "
```

You can find the full script here (https://github.com/sojohan/mcmc_example/blob/master/project.R)

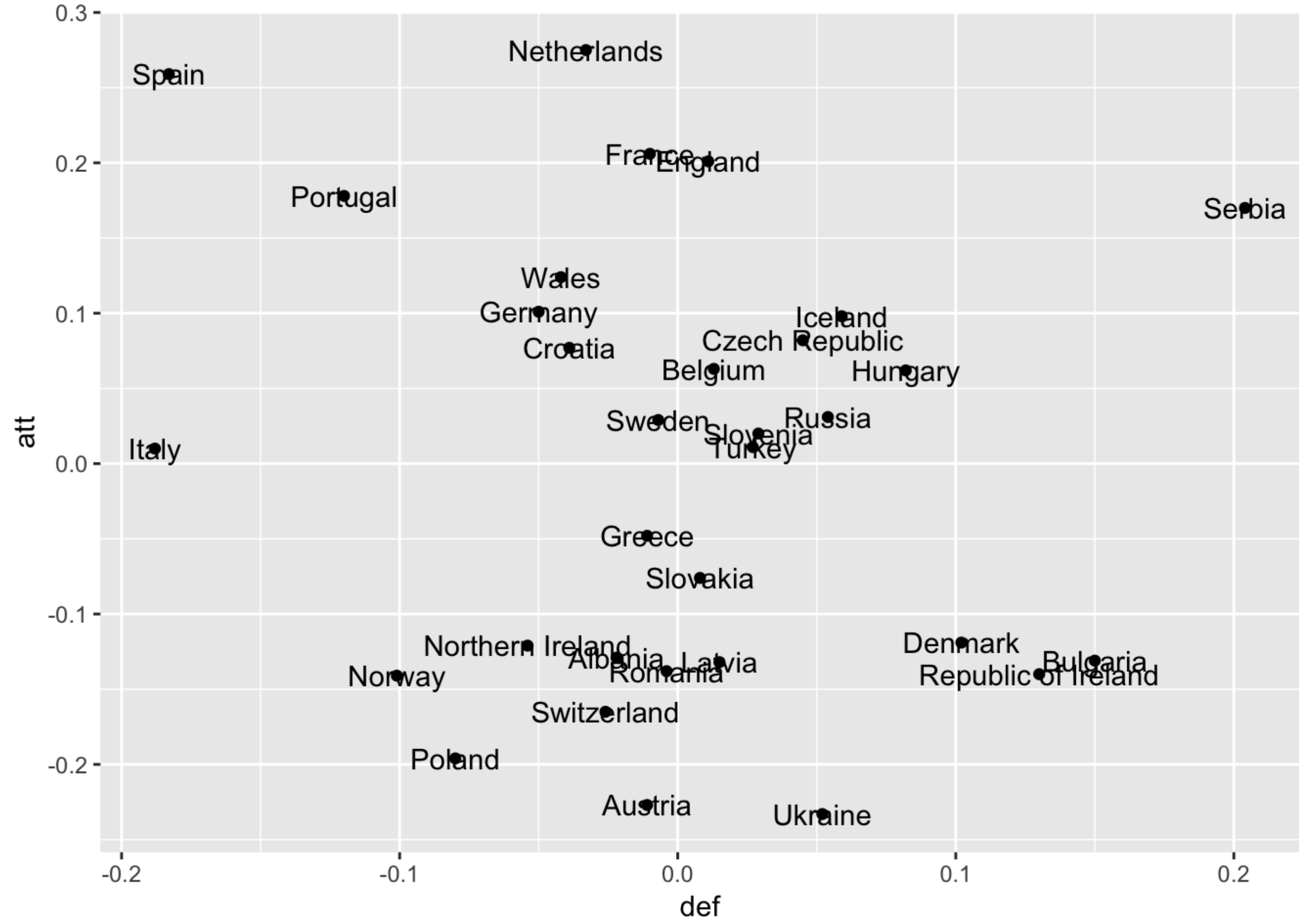
Fit the model

Empirical mean and standard deviation for the first ten variables att, def and home, mu and tau variables. Observe that the variable home is positive. Unfortunately, my model has many parameters that needs to be estimated and because this paper only can be max four pages long, I have made a spreadsheet with the full set of diagnostics here (you can download it).

Fit Summary

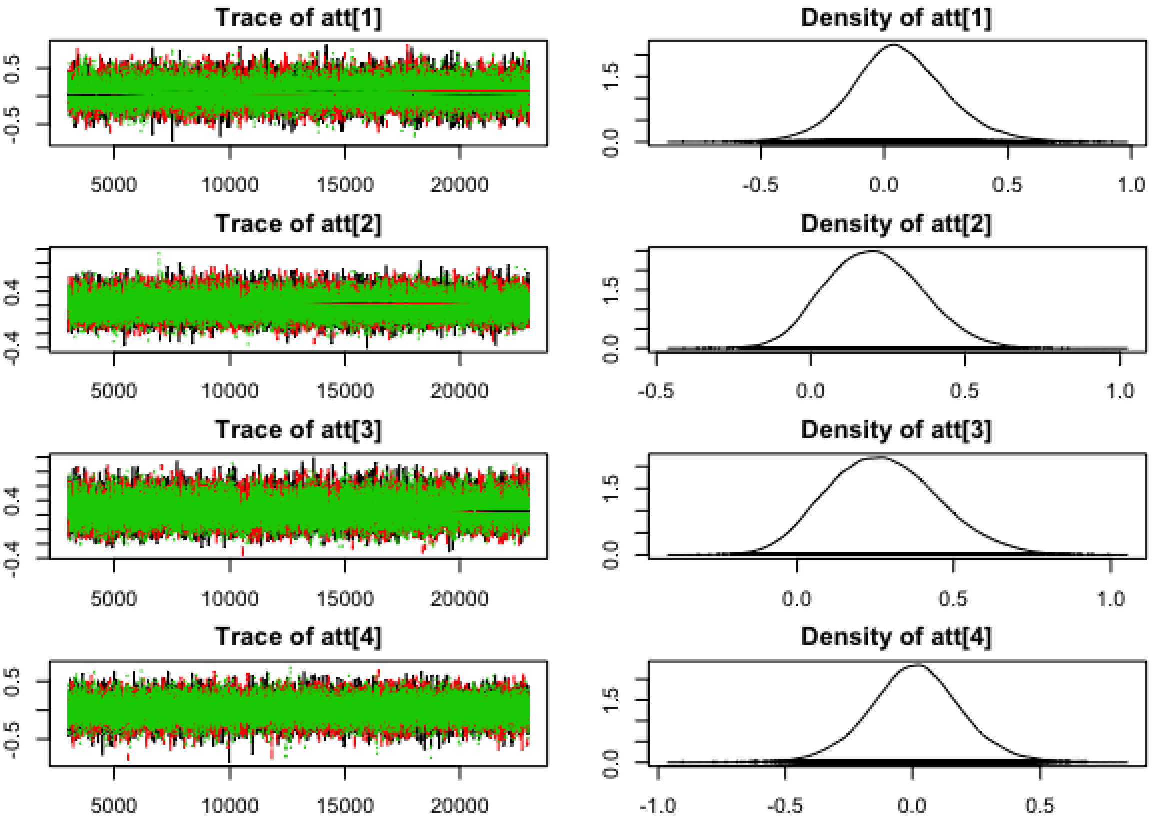
	mean	sd	median	X2.5.	X10.	X90.	X97.5.	n.iter	n.eff
att[1]	0.063	0.194	0.056	-0.310	-0.174	0.310	0.469	60000	27301.212
att[2]	0.206	0.155	0.202	-0.079	0.011	0.409	0.519	60000	9278.830
att[3]	0.275	0.175	0.269	-0.043	0.054	0.505	0.636	60000	6857.261
att[4]	0.011	0.180	0.011	-0.351	-0.215	0.236	0.370	60000	31617.324
att[5]	0.101	0.152	0.096	-0.187	-0.087	0.299	0.408	60000	18425.294
att[6]	0.178	0.145	0.174	-0.094	-0.005	0.368	0.472	60000	10665.777
att[7]	0.170	0.228	0.151	-0.233	-0.099	0.472	0.677	60000	11830.400
att[8]	0.259	0.159	0.255	-0.037	0.057	0.466	0.578	60000	8377.248
att[9]	0.029	0.165	0.028	-0.295	-0.178	0.240	0.359	60000	31077.104
att[10]	0.082	0.162	0.078	-0.231	-0.119	0.292	0.414	60000	23257.857
def[1]	0.013	0.154	0.012	-0.298	-0.176	0.203	0.329	60000	30681.418
def[2]	-0.010	0.123	-0.008	-0.263	-0.163	0.143	0.231	60000	36134.761
def[3]	-0.033	0.137	-0.027	-0.326	-0.207	0.133	0.227	60000	30377.519
def[4]	0.027	0.141	0.025	-0.253	-0.146	0.204	0.313	60000	31707.059
def[5]	-0.050	0.128	-0.044	-0.322	-0.215	0.105	0.191	60000	25106.198
def[6]	-0.120	0.135	-0.108	-0.417	-0.299	0.040	0.116	60000	11195.683
def[7]	0.204	0.197	0.174	-0.103	-0.018	0.472	0.665	60000	5574.174
def[8]	-0.183	0.154	-0.166	-0.527	-0.388	-0.001	0.072	60000	7020.389
def[9]	-0.007	0.137	-0.005	-0.291	-0.178	0.161	0.261	60000	33061.597
def[10]	0.045	0.131	0.042	-0.214	-0.116	0.212	0.315	60000	31157.153

I can plot the pairs (def,att) in a graph. Interesting to see that the upper left quadrant is all the aggressive teams like Spain, France, England, Germany and Netherlands.



Check the model

Here is the first couple of traces:



Traces for att

Here is the Gelman scale factor is 1.00 except tau_att which is 1.02.

```
##    variable Point est. Upper C.I.
## 1    att[1]       1.00       1.00
## 2    att[2]       1.00       1.00
## 3    att[3]       1.00       1.00
## 4    att[4]       1.00       1.00
## 5    att[5]       1.00       1.00
## 6    att[6]       1.00       1.00
## 7    att[7]       1.00       1.00
## 8    att[8]       1.00       1.00
## 9    att[9]       1.00       1.00
## 10  att[10]       1.00       1.00
## 11     home       1.00       1.00
## 12   mu_att       1.00       1.00
## 13  tau_att       1.02       1.02
## 14  tau_def       1.00       1.00
```

To view the full diagnostic data see this link (https://github.com/sojohan/mcmc_example/blob/master/project_mcmc.xlsx) for more. I conclude that the traces have converged. No autocorrelation was detected and here is subset of the correlation diagonal.

```
##                home       mu_att       mu_def    tau_att    tau_def
## Lag 0   1.000000000  1.000000000  1.000000000 1.00000000 1.00000000
## Lag 1   0.477421216  0.386951529  0.546749185 0.89463747 0.87521898
## Lag 5   0.168404567  0.090893106  0.176854367 0.66689008 0.63336734
## Lag 10  0.106557209  0.010626300  0.044100092 0.46883586 0.43682901
## Lag 50 -0.003242774 -0.004098353 -0.009329736 0.01573797 0.00949689
```

Note there is some autocorrelation in parameter tau_att.

Use the model

I can simulate a tournament with our model and calculate the probability that Denmark will win. Let me take Denmark against (Spain, Germany, Sweden, Netherlands). Here is code Denmark against Spain:

```
theta_home= exp(mod_csim[,"home"]+mod_csim[,"att[16]"]+mod_csim[,"def[8]"])
theta_away=exp(mod_csim[,"att[8]"]+mod_csim[,"def[16]"])
n_sim=length(theta_home)
den_spain=rpois(n=n_sim,theta_home)
spain_den=rpois(n=n_sim,theta_away)
mean(den_spain>spain_den)
```

Here are the probabilities that Denmark will win over Spain, Germany, Sweden, Netherlands:

Win Probabilities in%

Team	Denmark
Spain	24.91
Sweden	34.59
Germany	31.51
Netherlands	28.36