Could an Independent Yorkshire Win the World Cup - Simulate World Cups

Robert Hickman

2018-06-07

Recently, a Yorkshire national football team appeared in a league of national teams for stateless people. This got me wondering how the historic counties of the UK would do at the world cup. Could any of them compete with full international teams?

This is the complete script for an short article I wrote for CityMetric on the topic. It's split over 5 separate parts and is pretty hefty but contains pretty much everything you need to clone the article. Now that we've picked the teams for each nation and county, it's finally time to make predictions about the World Cup.

```r
library(dplyr)
library(magrittr)
library(data.table)
library(ggplot2)
```

Get County Rankings

Now that we have the teams for each county, we want to work out how well they would do at a world cup. For this, we need to know roughly what their ranking would be compared to actual nations.

Two sources of rankings of nations are the official FIFA world rankings, and also the world ELO ratings of each nation at www.eloratings.net.

I scraped both of these (accurate to mid-May) and cleaned the data to match the nation names to those in the player dataset we're using.

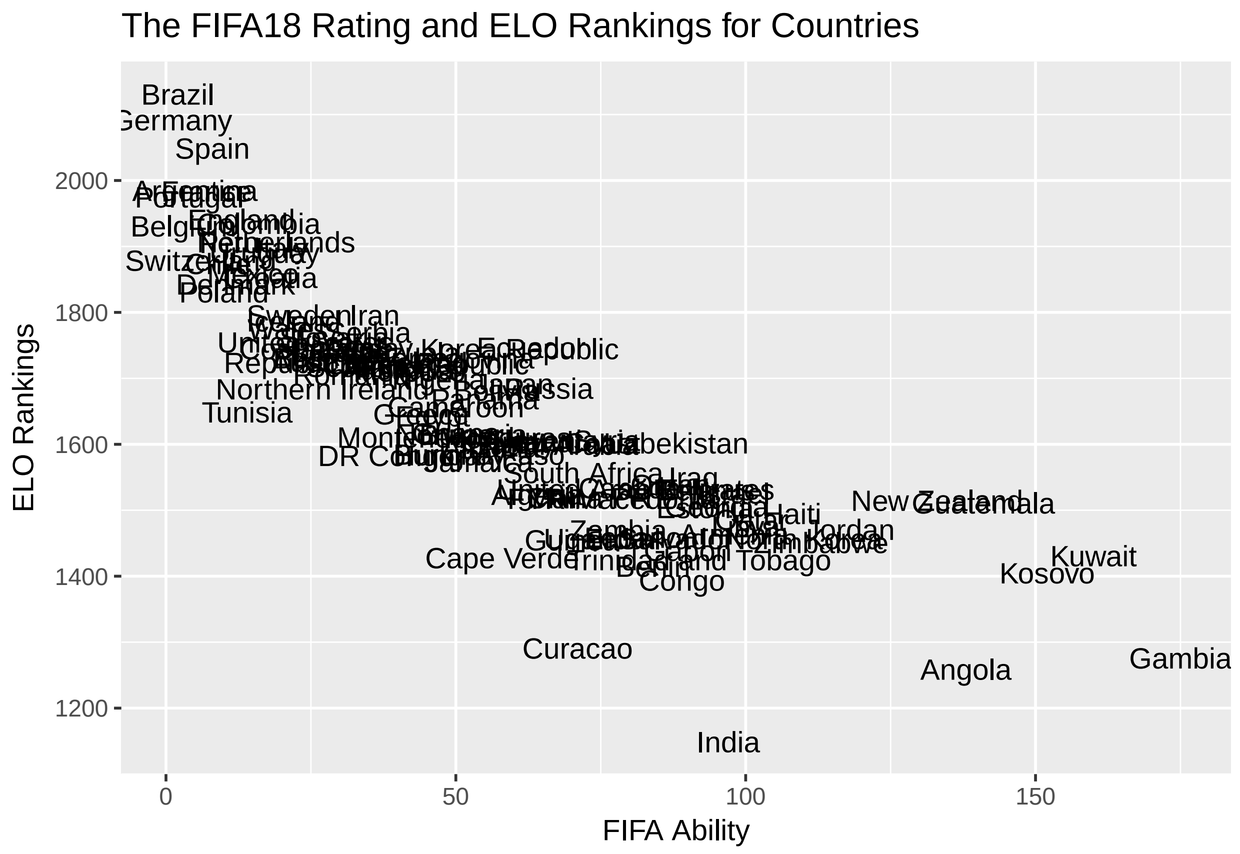
```r
#scraped world rankings from FIFA and world ELO
#http://www.fifa.com/fifa-world-ranking/ranking-table/men/index.html
#https://www.eloratings.net/
#accurate for mid-May
#have matched country names between world rankings and FIFA player data
world_rankings <- readRDS("national_rankings.rds")
```

```r
#glimpse the data
head(world_rankings)
```

```
##      nation  ELO FIFA
## 1    Brazil 2131    2
## 2   Germany 2092    1
## 3     Spain 2049    8
## 4 Argentina 1985    5
## 5    France 1984    7
## 6  Portugal 1975    4
```

ELO is a chess rating mechanism which can be used to make predictions about which team would win in a matchup. If we compare it to the FIFA rankings, we can see there's a clear negative correlation (the lower the ranking (e.g. top 10 teams in the world), the higher the ELO)

```r
#plot FIFA rankings vs. ELO
p <- ggplot(data = world_rankings, aes(x = FIFA, y = ELO)) +
  geom_text(aes(label = nation)) +
  xlab("FIFA Ability") +
  ylab("ELO Rankings") +
  ggtitle("The FIFA18 Rating and ELO Rankings for Countries")

plot(p)
```



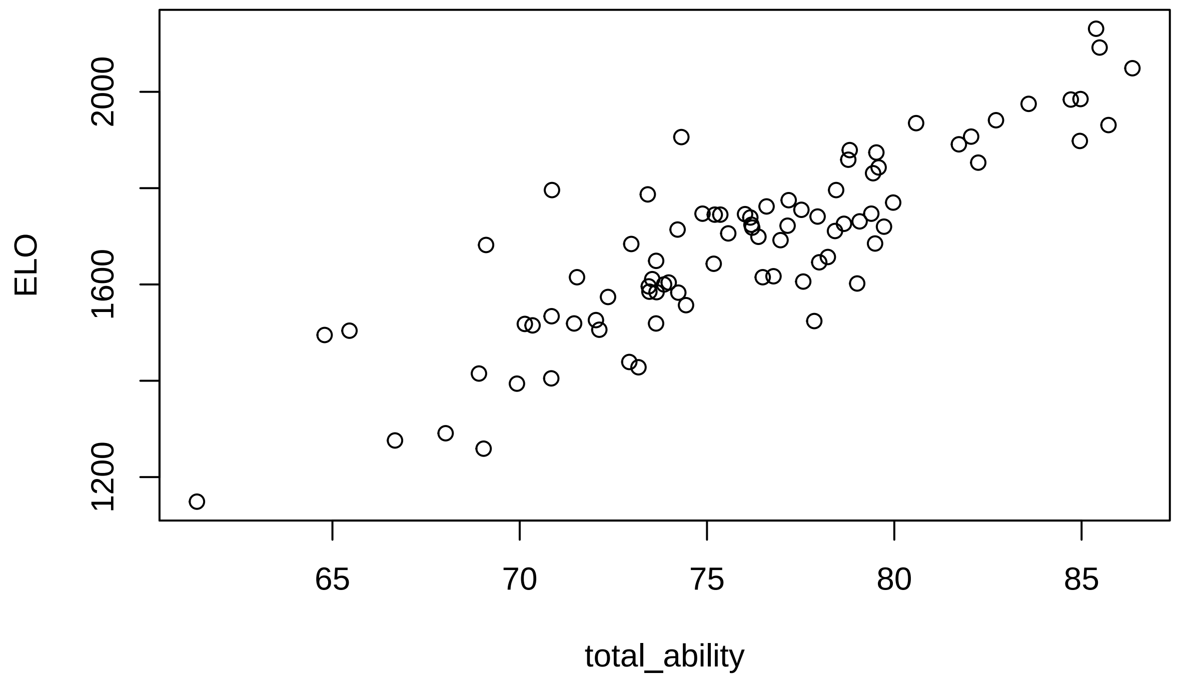
To validate our method, the total ability of each team from their players in FIFA18 should correlate with this ELO rating.

If we merge in the optimal team data and plot it against ELO we see nice linear positive correlation.

```r
#merge in the world rankings for each fieldable national team
national_teams %<>% merge(., world_rankings, by = "nation") %>%
  #merge in the optimal team total_ability for each nation
  merge(., unique(select(optimal_national_teams, nation, total_ability)), by = "nation")

head(national_teams)
```

```
##      nation players gks  ELO FIFA total_ability
## 1   Albania      36   2 1596   56      73.44459
## 2   Algeria      58   3 1524   64      77.86387
## 3    Angola      16   1 1259  138      69.03657
## 4 Argentina     875 100 1985    5      84.97171
## 5 Australia     199  33 1714   40      74.21456
## 6   Austria     226  39 1726   26      78.65603
```

```r
plot(data = national_teams, ELO ~ total_ability)
```



We can quanitfy this correlation by creating a linear model using lm() and see that the adjusted R-squared is rather high- 0.7354.

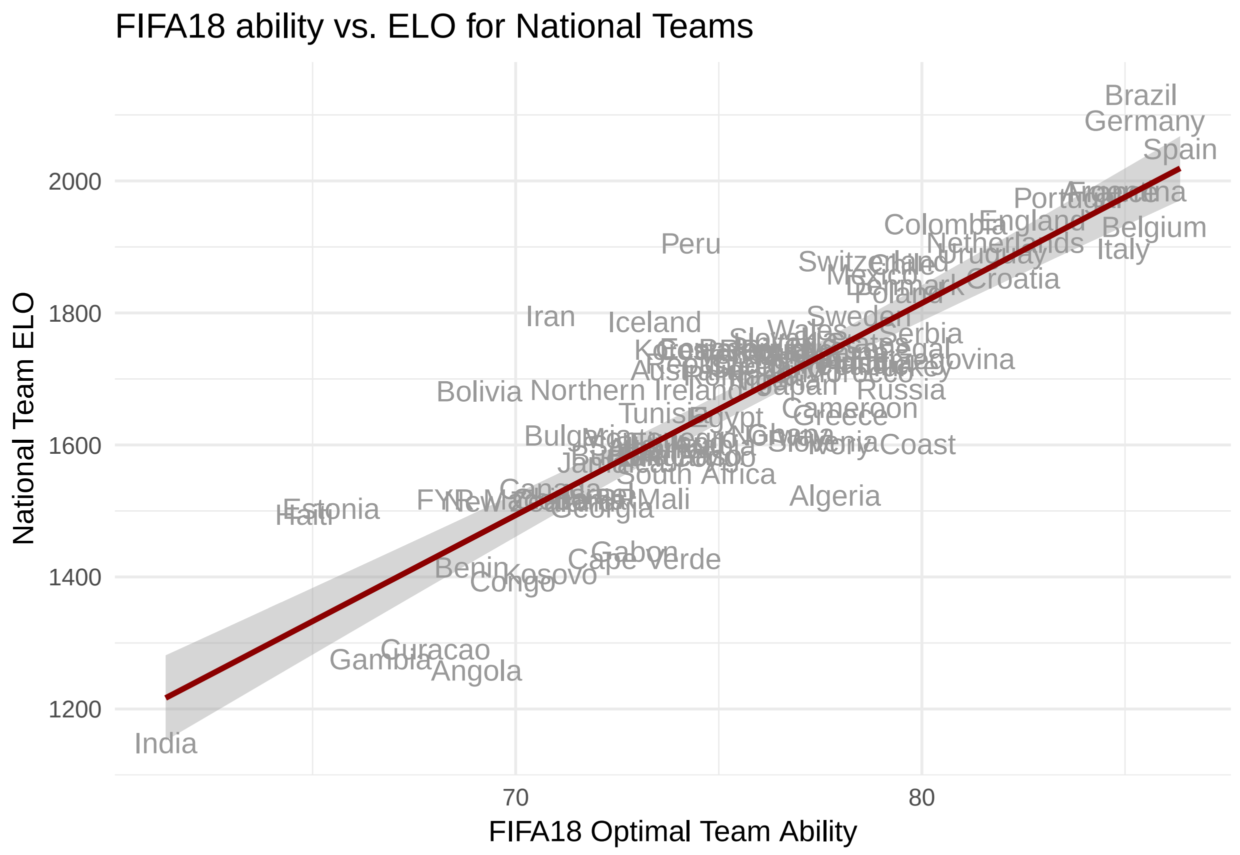
```r
#regress ELO against total_ability (as judged by selection of FIFA18 players)
ability_regression <- lm(data = national_teams, ELO ~ total_ability)

#summary
summary(ability_regression)
```

```
##
## Call:
## lm(formula = ELO ~ total_ability, data = national_teams)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -222.229  -58.773    2.228   48.415  274.785
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -755.764    160.669  -4.704 1.02e-05 ***
## total_ability   32.133      2.111  15.221  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 97.31 on 82 degrees of freedom
## Multiple R-squared:  0.7386, Adjusted R-squared:  0.7354
## F-statistic: 231.7 on 1 and 82 DF,  p-value: < 2.2e-16
```

We can also plot this regression to further convince ourselves that predicting ELO from FIFA18 ability is a fairly valid move.

```
#plot ELO vs. total_ability
p <- ggplot(data = national_teams, aes(x = total_ability, y = ELO)) +
  geom_text(aes(label = nation), colour = "grey60") +
  #add in the linear regression line + confidence intervals
  stat_smooth(method = "lm", colour = "darkred") +
  xlab("FIFA18 Optimal Team Ability") +
  ylab("National Team ELO") +
  ggtitle("FIFA18 ability vs. ELO for National Teams") +
  theme_minimal()

plot(p)
```

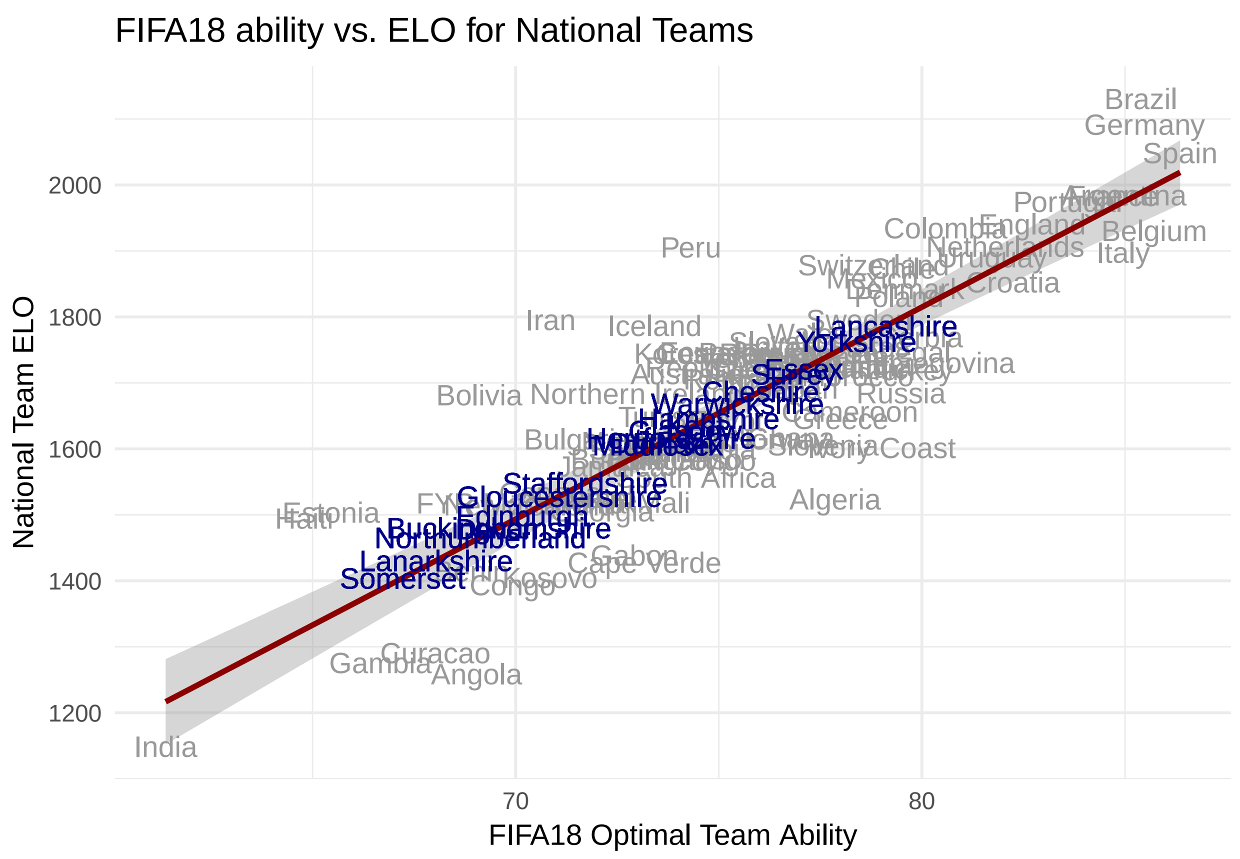


As we have a predictor for ELO based on FIFA18 ability, we can now predict the ELO of each county team. We simply feed the model back into our df of optimal county teams.

If we plot this over the previous plot we can see the counties have ELOs which fall within a range of national team abilities. The best counties (Yorkshire and Lancashire) are about as good as teams which generally qualify for world cups (e.g. Sweden and Serbia) whereas some counties (e.g.) are much less proficient and would probably struggle to qualify.

Given the teams we saw that were selected earlier, this makes sense- Yorkshire and Lancashire can field generally pretty solid teams of international/near-international level footballers and so would be expected to be competitive.

```
county_teams %<>% merge(., select(optimal_county_teams, county = nation, total_ability), by = "county")
  #predict the ELO of each county using the previous regression
  mutate(predicted_ELO = predict(ability_regression, .))

#add these to the plots of ELO ~ FIFA team ability
plot(p + geom_text(data = county_teams, aes(x = total_ability, y = predicted_ELO, label = county), colo
```



Simulate World Cups

Finally, we want to know if any of these counties have a shot at winning the World Cup.

To do this, the best method is simply to simulate lots of World Cups and see what the percentage chance for each team is. This is possible as ELO gives us a quantifiable measure of how likely a given team is to beat another.

Before we can simulate the World Cup however, we need some information about the draw.

```
wc_teams <- data.frame(nation = c("Russia", "Saudi Arabia", "Egypt", "Uruguay",
                                  "Portugal", "Spain", "Morocco", "Iran",
                                  "France", "Australia", "Peru", "Denmark",
                                  "Argentina", "Iceland", "Croatia", "Nigeria",
                                  "Brazil", "Switzerland", "Costa Rica", "Serbia",
                                  "Germany", "Mexico", "Sweden", "Korea Republic",
                                  "Belgium", "Panama", "Tunisia", "England",
                                  "Poland", "Senegal", "Colombia", "Japan"),
                           group = c(rep(letters[1:8], each = 4)),
                           draw = rep(1:4, 8))

group_matches <- data.frame(match = 1:6,
                            team1 = c(1,3,1,4,4,2),
                            team2 = c(2,4,3,2,1,3))

knockout_matches <- data.frame(round = c(rep("R16", 8), rep("QF", 4), rep("SF", 2), "F"),
                               team1 = c("a1", "c1", "e1", "g1", "b1", "d1", "f1", "h1",
                                         "m49", "m53", "m51", "m55", "m57", "m59", "m61"),
                               team2 = c("b2", "d2", "f2", "h2", "a2", "c2", "e2", "g2",
                                         "m50", "m54", "m52", "m56", "m58", "m60", "m62"),
                               match_id = c("m49", "m50", "m53", "m54", "m51", "m52", "m55", "m56",
                                            "m57", "m58", "m59", "m60", "m61", "m62", "FINAL"))
```

And then we need to write functions to do the simulation.

The first of these simply takes the ELO of the two teams and works out the win percentage for teamA (for teamB = 1 - p(teamA)).

This is used in two further functions which simulate the group stages, and then the knockout stages respectively.

For the groups, teams are drawn against each other as they will be in Russia and their ELOs compared. A random number generator is used to decided which teams wins (if p(teamA wins based on ELO) > random_number, teamA wins). I also included the chance to draw if the difference between the win_chance and the random_number is less than 0.1 in either direction.

The points each team is predicted to win in the groups is then summed and the top two teams from each group progresses to the knockout stage.

The knockout stage is easier to simulate as we don't need to worry about points/draws. The same method as above is used to predict the winning team and that team progresses, whilst we remove the other from a df. Eventually only one team is left- the winner of the tournament.

```
#uses ELO to calculate the chance of team A winning
calc_win_chance <- function(ratingA, ratingB) {
  win_chance <- 1/ (1+10^((ratingB-ratingA)/400))
}

#simulate the group stages of the tournament
simulate_groups <- function(group_letter, national_teams, group_matches) {
  group <- national_teams %>%
    filter(group == group_letter) %>%
    mutate(points = 0) %>%
    mutate(av_difference = 0) %>%
    arrange(draw)

  #six matches per group
  for(match in 1:6){
    team1 <- group$nation[group_matches$team1[match]]
    team2 <- group$nation[group_matches$team2[match]]

    #calculate winner using a random number generator and comparing to the ELO win percentages
    random_number_draw <- runif(1)
    win_chance <- calc_win_chance(group$ELO[group$nation == team1], group$ELO[group$nation == team2])
    #update ELOs and assign group stage points
    if(win_chance - random_number_draw > 0.1) {
      group$points[group$nation == team1] <- group$points[group$nation == team1] + 3
      group$points[group$nation == team2] <- group$points[group$nation == team2] + 0

      group$ELO[group$nation == team1] <- group$ELO[group$nation == team1] + 50*(1-win_chance)
      group$ELO[group$nation == team2] <- group$ELO[group$nation == team2] + 50*(0-(1-win_chance))

    } else if(win_chance - random_number_draw < -0.1) {
      group$points[group$nation == team1] <- group$points[group$nation == team1] + 0
      group$points[group$nation == team2] <- group$points[group$nation == team2] + 3

      group$ELO[group$nation == team1] <- group$ELO[group$nation == team1] + 50*(0-win_chance)
      group$ELO[group$nation == team2] <- group$ELO[group$nation == team2] + 50*(1-(1-win_chance))

    } else {
      group$points[group$nation == team1] <- group$points[group$nation == team1] + 1
      group$points[group$nation == team2] <- group$points[group$nation == team2] + 1

      group$ELO[group$nation == team1] <- group$ELO[group$nation == team1] + 50*(0.5-win_chance)
      group$ELO[group$nation == team2] <- group$ELO[group$nation == team2] + 50*(0.5-(1-win_chance))
    }

    group$av_difference[group$nation == team1] <- group$av_difference[group$nation == team1] +
      (group$ELO[group$nation == team1] - group$ELO[group$nation == team2])
    group$av_difference[group$nation == team2] <- group$av_difference[group$nation == team2] -
      (group$ELO[group$nation == team1] - group$ELO[group$nation == team2])
  }

  #arrange the groups by points per team, then by the ELO difference between a team and it's rivals
  #use ELO difference as secondary sorter as proxy for goal difference
  group <- arrange(group, -points, -av_difference) %>%
    mutate(position = 1:4)
  return(group)
}

#simulate the knockout rounds
simulate_knockout_rounds <- function(national_teams, knockout_matches) {
  for(match in seq(nrow(knockout_matches))) {
    #get the teams and the match id
    team1 <- as.character(national_teams$nation[which(national_teams$id == knockout_matches$team1[match]
    team2 <- as.character(national_teams$nation[which(national_teams$id == knockout_matches$team2[match]
    match_id <- as.character(knockout_matches$match_id[match])

    national_teams$id[which(national_teams$nation %in% c(team1, team2))] <- match_id

    #use a random number generator to decide the winner
    random_number_draw <- runif(1)

    #use ELO chances vs. the random number to work out which team wins
    win_chance <- calc_win_chance(national_teams$ELO[national_teams$nation == team1], national_teams$EL

    #update ELOs and remove losing team
    if(win_chance > random_number_draw) {
      national_teams$ELO[national_teams$nation == team1] <- national_teams$ELO[national_teams$nation ==
      national_teams <- national_teams[-which(national_teams$nation == team2),]
    } else {
      national_teams$ELO[national_teams$nation == team2] <- national_teams$ELO[national_teams$nation ==
      national_teams <- national_teams[-which(national_teams$nation == team1),]
    }
  }
  #returns the nation from the last remain row of the df == the winner of the tournament
  return(national_teams$nation)
}

#simulate the whole tournament
simulate_tournament <- function(national_teams, knockout_matches, group_matches) {
  #simulate the group stages
  knockout_rounds <- rbindlist(lapply(letters[1:8], simulate_groups,
                                      national_teams = national_teams, group_matches = group_matches))
    #filter the top two teams from each group
    filter(position < 3) %>%
    mutate(id = paste0(group, position)) %>%
    select(nation, ELO, id)

  #simulate the knockout rounds until only 1 team remains
  winner <- simulate_knockout_rounds(national_teams = knockout_rounds, knockout_matches = knockout_matc
    as.character()
  return(winner)
}
```

To simulate the world cups, first we merge the ELO data with the world cup draw information. We also have to add Panama as they were missing from the teams based on our player data.

Then here I run 10 simulations of the tournament and print the winners. Generally the clear favourites of the simulation are Brazil, then Germany, Spain and Argentina. This makes sense- they have the highest ELOs of all the nations.

```
#merge the ELOs with the world cup draw information
wc_teams %<>% merge(., select(national_teams, nation, ELO) %>%
                      rbind(., data.frame(nation = "Panama", ELO = 1669)), by = "nation")

#run 10 simulations of the world cup choosing winners via ELO
for(simulation in 1:10) {
    winner <- simulate_tournament(wc_teams, knockout_matches, group_matches)
    if(simulation == 1) {
      winners <- winner
    } else {
      winners <- append(winners, winner)
    }
}

#list the winners of these 10 simulations
winners
```

```
##  [1] "Brazil"      "Brazil"      "Germany"     "Brazil"      "Spain"
##  [6] "Brazil"      "Switzerland" "Germany"     "England"     "Germany"
```

Finally, we can substitute in each county for the English national team and run x simulations (I use 10000 as anything more would take an unrealistic amount of processing time) to see what the chance of them winning the world cup would be.

I iterate this through each county and then get a df of the chances for every nation (-England) and that county to win.

```r
simulate_counties <- function(county, simulations) {
  #replace Englands ELO with that of the county team replacing them
  wc_teams$ELO[wc_teams$nation == "England"] <- county_teams$predicted_ELO[county_teams$county == count

  #run x number of simulations
  for(simulation in 1:simulations) {
    winner <- simulate_tournament(wc_teams, knockout_matches, group_matches)
    #if 'England' wins, replace England with the county
    if(winner == "England") {
      winner <- county
    }
    if(simulation == 1) {
      winners <- winner
    } else {
      winners <- append(winners, winner)
    }
  }

  #spit out a df with each winner and the number of times they win
  simulation_df <- data.frame(table(winners))
  names(simulation_df) <- c("nation", "championships")

  #work out the percentage chane of each nation/county winning
  simulation_df$percentage <- simulation_df$championships / (simulations/100)
  simulation_df$county_test <- county
  return(simulation_df)
}

#run for many simulations
#TAKES A LOT OF TIME
simulations_results <- rbindlist(lapply(county_team_rankings$nation, simulate_counties, 10000))
```

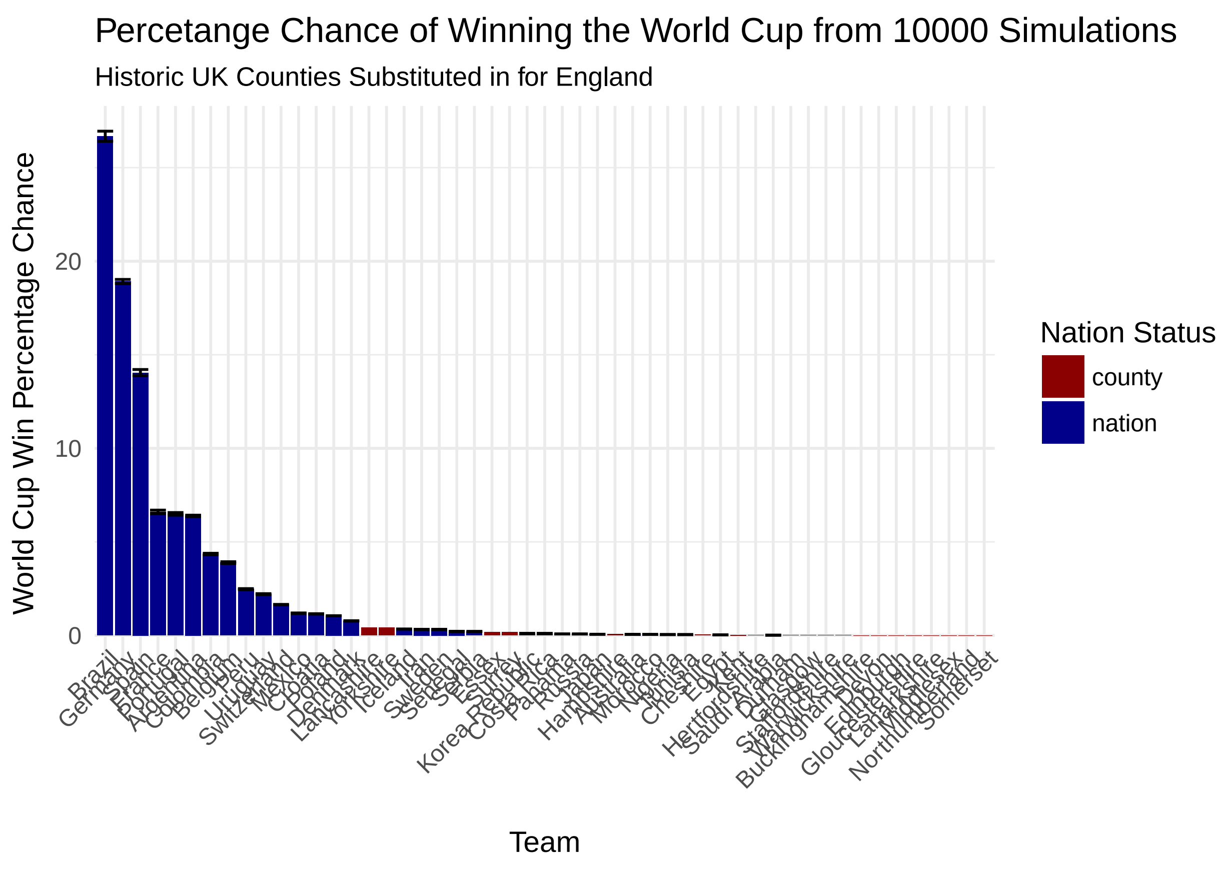
Once we have that data out, some munging is necessary to get the average chance of winning the World Cup for each nation and label the counties and nations separately.

```r
#munge the simulation_results
simulation_results %<>% setDT() %>%
  #get the average wc winning chance per nation across all simulations
  .[, perc_chance := mean(percentage), by = "nation"] %>%
  .[, perc_var := var(percentage), by = "nation"] %>%
  .[, c("nation", "perc_chance", "perc_var")] %>%
  unique(.) %>%
  #bind in the nations which never win the world cup in any simulation
  rbind(., unique(data.frame(nation = county_teams$county[which(!county_teams$county %in% .$nation)],
                      perc_chance = 0,
                      perc_var = NA))) %>%
  #is the team a nation or a county
  .[nation %in% county_teams$county, nation_status := "county"] %>%
  .[!nation %in% county_teams$county, nation_status := "nation"] %>%
  #order by percentage chance of winning the WC
  .[, nation := factor(nation, levels = nation[order(-.$perc_chance)])]
```

And can then plot the results…

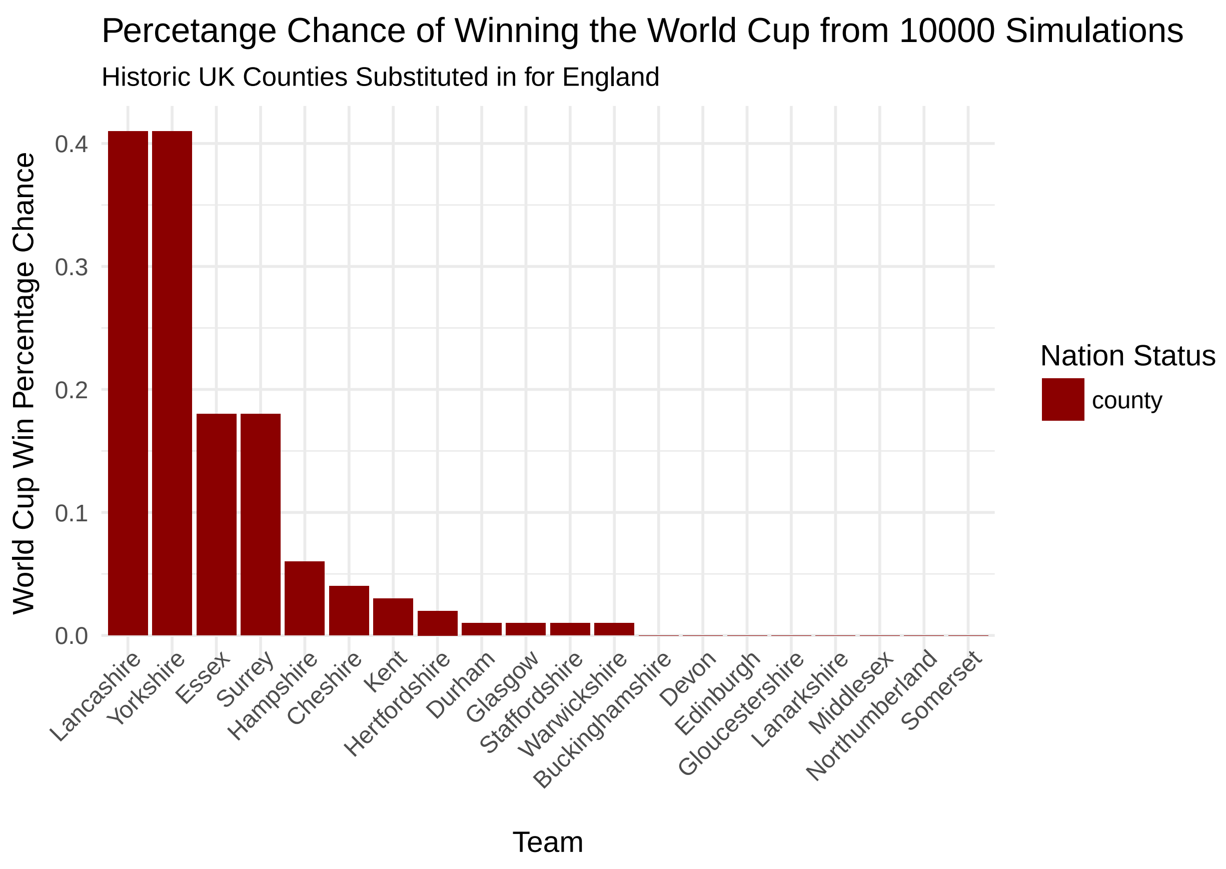
```
#plot the results
p <- ggplot(data = simulation_results, aes(x = nation, y = perc_chance)) +
  geom_bar(stat = "identity", aes(fill = nation_status)) +
  geom_errorbar(aes(ymax = perc_chance + perc_var, ymin = perc_chance - perc_var)) +
  scale_fill_manual(values = c("darkred", "darkblue"), name = "Nation Status") +
  xlab("Team") +
  ylab("World Cup Win Percentage Chance") +
  ggtitle("Percetange Chance of Winning the World Cup from 10000 Simulations",
          subtitle = "Historic UK Counties Substituted in for England") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1.2))

p2 <- ggplot(data = filter(simulation_results, nation_status == "county"), aes(x = nation, y = perc_char
  geom_bar(stat = "identity", aes(fill = nation_status)) +
  geom_errorbar(aes(ymax = perc_chance + perc_var, ymin = perc_chance - perc_var)) +
  scale_fill_manual(values = c("darkred", "darkblue"), name = "Nation Status") +
  xlab("Team") +
  ylab("World Cup Win Percentage Chance") +
  ggtitle("Percetange Chance of Winning the World Cup from 10000 Simulations",
          subtitle = "Historic UK Counties Substituted in for England") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1.2))

plot(p)
```

```
## Warning: Removed 20 rows containing missing values (geom_errorbar).
```



```r
plot(p2)
```

```
## Warning: Removed 20 rows containing missing values (geom_errorbar).
```


Percetange Chance of Winning the World Cup from 10000 Simulations
Historic UK Counties Substituted in for England
World Cup Win Percentage Chance
0.4
0.3
0.2
0.1
0.0
Lancashire
Yorkshire
Essex
Surrey
Hampshire
Cheshire
Kent
Hertfordshire
Durham
Glasgow
Staffordshire
Warwickshire
Buckinghamshire
Devon
Edinburgh
Gloucestershire
Lanarkshire
Middlesex
Northumberland
Somerset
Team
Nation Status
county