Could an Independent Yorkshire Win the World Cup - Finding British Player's Birthplaces

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 6 separate parts and is pretty hefty but contains pretty much everything you need to clone the article. Last time, we found the position abilities of each player using LASSO regression. This time, we'll geolocate the birthplace of the British players in our dataset to find which county teaam they'd be eligible for.

```r
library(dplyr)
library(magrittr)
library(rvest)
library(data.table)
library(ggplot2)
#use pediarr to query wikipedia to find the birthplace of players
library(pediarr)
#use googleway to geocode birthplaces
library(googleway)
#use sf to bin players into counties
library(sf)
```

Find British Players Birthplaces

To select our county teams, we need to know where each British player was born (and thus their 'county' nationality). Fortunately, wikipedia has an extremely detailed database of thousands of footballers, incluiding their birthplace (which we can assume is at least reasonably correct).

First, the data needs to be filtered to include only players with British nationalities (English, Welsh, Scottish, or Northern Irish) or Irish. It's very plausible that some players representing other countries would be born in England, and so eligible for the hypothetical county teams, but unlikely, and more trouble than it's worth.

When filtering, I also remove players who have no wikipedia page/birthplace listed. For some of these, I was able to manually locate their birthplace. Some players don't get matched very well (mostly due to Australian/American footballers) and it was easiest just to manually supply the links to their wikipedia page.

```r
#players with no wikipedia birthplace listed
players_missing_data <- c("Liam Lindsay","Greg Docherty","Mikey Devlin","Josh Dacres-Cogley",
                          "Tom Broadbent","Callum Gribbin","Sam Hughes","
                          James Cook","Daniel Jarvis","Zachary Dearnley","Ro-Shaun Williams",
                          "Jack Fitzwater","Jack Hamilton","Lewis Banks","Greg Bolger","Chris Shields",
                          "Conor Wilkinson","Barry McNamee","Keith Ward","Simon Madden","Dylan Connolly
                          "Brian Gartland","Dinny Corcoran")

#players whose birthplace was manually found
missing_players_data <- readRDS("missing_player_birthplaces.rds")
```

```r
#players whose wikipedia page is manually linked
manual_links <- readRDS("manual_links.rds")
```

The function below then iterates through every player with a nationality from the British Isles and searches for a matching wikipedia page.

It then looks for the birthplace of that player on their wikipedia page and returns a df containing the player and their birthplace.

It also tries to match the birthdate listed from FIFA18 with that on their wikipedia page as a check and throws a warning if they don't match. I haven't looked into if there are mismatches there but ~50 players overall don't match perfectly.

```r
uk_players_info <- all_players_data %>%
  #only want data to help identify players by wiki page
  select(id, name, nationality, birthdate) %>%
  #only include UK nations (+Ireland)
  filter(nationality %in% c("England", "Scotland", "Wales", "Northern Ireland", "Republic of Ireland"))
  #remove duplicated names
  #might lose some players here but they're all so far down the pecking order effect should be minimal
  filter(!duplicated(name)) %>%
  #remove players who have no wikipedia birthplace
  filter(!name %in% players_missing_data)

#function to find the wikipedia page of each player
#returns a df with the player name and birthplace scraped from wikipedia
get_info <- function(row) {
  #get player info
  name <- uk_players_info$name[row]
  birthday <- uk_players_info$birthdate[row]
  id <- uk_players_info$id[row]

  #search wikipedia using the player name
  search <- pediasearch(name, extract = TRUE, limit = 10)
  #if a troublesome search use manual link
  if(name %in% manual_links$name) {
    wiki_suffix <- manual_links$link[which(manual_links$name == as.character(name))]
  } else {
    #else find the wikipedia page suffix for the player
    if(search[1] == "" & length(search) == 1) {
      wiki_suffix <- name %>%
        gsub(" ", "_", .)
    } else {
      footballer <- grep("football", search)[1]
      wiki_suffix <- names(search)[footballer] %>%
        gsub(" ", "_", .)
    }
  }

  #read the info card from the players wikipedia page
  info_card <- read_html(paste0("https://en.wikipedia.org/wiki/", wiki_suffix)) %>%
    html_nodes(".vcard") %>%
    .[1] %>%
    html_table(fill = TRUE) %>%
    data.frame()
  names(info_card) <- paste0("X", 1:ncol(info_card))
  info_card$X1 <- tolower(info_card$X1)

  #check if the wikipedia birthdate matches the FIFA one
  birthdate <- info_card %>%
    filter(X1 == "date of birth")

  birthdate <- birthdate$X2 %>%
    as.character() %>%
    gsub(" .*", "", .) %>%
    gsub("\\(|\\)", "", .) %>%
    as.Date()

  if(birthdate != birthday){
    warning(paste(row, "birthdays do not match"))
  }

  #find the players birthplace
  birthplace <- info_card %>%
    filter(X1 == "place of birth")

  birthplace <- birthplace$X2 %>%
    gsub("\\[.*", "", .)

  #return info as a df
  df <- data.frame(id = id,
                   name = name,
                   birthdate = birthdate,
                   birthplace = birthplace)
  return(df)
}

#run the function over the first 1333 players
#after this very few players are found
british_player_birthplaces <- rbindlist(lapply(1:1329, get_info)) %>%
  #bind in the manually found data
  rbind(., missing_players_data)
```

Now that we have the birthplaces for each player, we need to convert these into coordinates via geocoding. For this I use googleway, but the geocode() function from ggmap could also be used.

The function takes a place and a key (for the API which isn't included in the knitted markdown) and finds the lat lon for that place. To save on API requests I only run it on unique birthplaces then merge this back into the dataset.

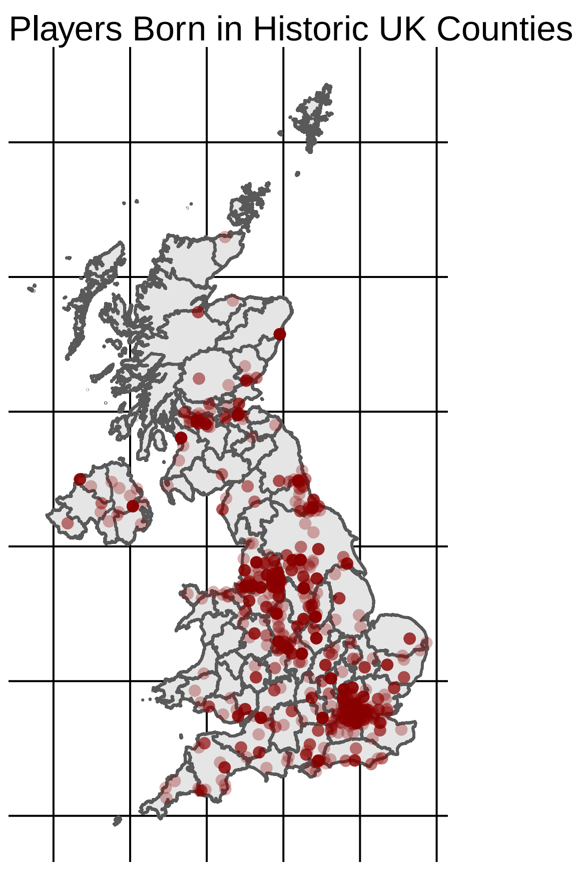
Once we have the lat/lon of each birthplace we can convert the df of players into an sf (spatial) object. If we do this, we see that a lot of players who are eligible for British nations aren't actually born on the islands (e.g. Raheem Sterling was born in Jamaica). so I only select those which are born within the grouped spatial object of all 5 countries.

```
#geocodes locations using googlemaps
#requires a google maps API key (hidden here)
googleway_geocode <- function(place, key){
  data <- google_geocode(place, key = key)
  latlon <- data$results$geometry$location[1,] %>%
```

```r
    mutate(birthplace = place)
  #returns coordinates in the form latitude/longitude
  return(latlon)
}

birthplace_coords <- rbindlist(lapply(as.character(unique(british_player_birthplaces$birthplace)),
                                      googleway_geocode, key = key))

#also melt into one spatial row for subsetting later
uk <- uk_counties %>%
  group_by("UK") %>%
  summarise()

british_player_birthplaces <- british_player_birthplaces %>%
  merge(., birthplace_coords, by = "birthplace") %>%
  #convert to an sf object
  st_as_sf(coords = c("lng", "lat"), crs = st_crs(uk_counties)) %>%
  #keep only those born within the UK proper
  .[unlist(st_contains(uk, .)),]
```

If we plot the players, we see they tend to be grouped around the large cities in London, Lancashire, and Yorkshire, with realtively few in Northern Ireland, rural Wales and the Highlands

```r
p <- ggplot(data = uk_counties) +
  geom_sf() +
  geom_sf(data = british_player_birthplaces, colour = "darkred", alpha = 0.3) +
  ggtitle("Players Born in Historic UK Counties") +
  theme_void()

plot(p)
```



To find which county each player comes from, we can take the lat/lon of their birthplace and find which county shapefile contains it. The name of that county shapefile is then returned as a new column on the df of all British players

```r
#find the historic county each player was born within
british_player_birthplaces$county <- unlist(lapply(seq(nrow(british_player_birthplaces)), function(playe
  #which county is there birthplace coordinates in
  container <- st_contains(uk_counties, british_player_birthplaces[player,])
  if(length(unlist(container)) == 1) {
    #which county name is this
    county <- as.character(uk_counties$county[as.numeric(t(container))])
    } else {
      county <- NA
    }
  return(county)
}))
```

if we table the results of the county binning, we can see that many counties contain very few players, whereas some contain many more (e.g. Lancashire has 164 available players, whereas Cambridgeshire has only 5). Later, we will only look at counties that can field at least 10 outfield players + 1 goalkeeper.

```r
#the number of players from each historic county
table(british_player_birthplaces$county)
```

```
##
##                     Aberdeen                    Anglesey
##                           12                           1
##                        Angus                    Ayrshire
##                     3                        12
##          Bedfordshire                 Berkshire
##                    10                        15
##          Berwickshire           Buckinghamshire
##                     1                        15
##             Caithness            Cambridgeshire
##                     1                         5
##         Cardiganshire           Carmarthenshire
##                     1                         2
##        Carnarvonshire                  Cheshire
##                     2                        50
##              Cornwall             County Antrim
##                     5                        13
##         County Armagh County Derry / Londonderry
##                     2                         7
##           County Down         County Fermanagh
##                     3                         2
##         County Tyrone                Cumberland
##                     3                         8
##          Denbighshire                Derbyshire
##                     4                        13
##                 Devon                    Dorset
##                    17                         3
##         Dumfriesshire            Dunbartonshire
##                     2                         5
##                Dundee                    Durham
##                     6                        26
##             Edinburgh                     Essex
##                    23                        71
##                  Fife                Flintshire
##                     5                         4
##             Glamorgan                   Glasgow
##                    12                        35
##       Gloucestershire                 Hampshire
##                    13                        28
##         Herefordshire             Hertfordshire
##                     5                        33
##       Huntingdonshire           Inverness-shire
##                     3                         3
##                  Kent               Lanarkshire
##                    50                        18
##            Lancashire            Leicestershire
##                   164                        12
##          Lincolnshire                 Middlesex
##                     8                        77
##            Midlothian              Monmouthshire
##                     5                         5
##                 Nairn                   Norfolk
##                     1                         6
##      Northamptonshire            Northumberland
##                    12                        14
##       Nottinghamshire               Oxfordshire
##                    20                         6
##            Perthshire               Renfrewshire
##                    3                    3
##         Selkirkshire            Shropshire
##                    1                   11
##             Somerset        Staffordshire
##                   12                   46
##        Stirlingshire              Suffolk
##                    4                   10
##               Surrey               Sussex
##                   63                   16
##         Warwickshire         West Lothian
##                   44                    1
##         Wigtownshire            Wiltshire
##                    1                    6
##      Worcestershire            Yorkshire
##                    6                  103
```