**Background**

Reading through the [vignette](https://cran.r-project.org/web/packages/magick/vignettes/intro.html) Its spotted the image\_morph() function. In this post , experiment with the function to build the GIF below that shows the changes in the England football first kit over time, using images from the excellent [Historical Football Kits](http://www.historicalkits.co.uk/) website.

**Scraping**

The Historical Football Kits website has a detailed section on England kits spread over six pages, starting from the first outfits used in [1872](http://www.historicalkits.co.uk/international/england/england-1872-1939.html). Each pages includes some interesting discussion – and importantly for this post – images of the kits.

We can use the read\_html() from the [xml2](https://cran.r-project.org/web/packages/xml2/) package and map() from [purrr](https://cran.r-project.org/web/packages/purrr) to read and save the source code of each page.

library(rvest)

library(tidyverse)

htmls <- c(

"http://www.historicalkits.co.uk/international/england/england-1872-1939.html",

"http://www.historicalkits.co.uk/international/england/england-1946-1960.html",

"http://www.historicalkits.co.uk/international/england/england-1960-1983.html",

"http://www.historicalkits.co.uk/international/england/england-1984-1997.html",

"http://www.historicalkits.co.uk/international/england/england-1997-2010.html",

"http://www.historicalkits.co.uk/international/england/england-2010-2019.html"

) %>%

map(read\_html)

From the source code we can then find the URLs of each kit image files using html\_nodes() and html\_attr() from [rvest](https://cran.r-project.org/web/packages/rvest/). I used purrr’s map\_dfr() to store the links in a tibble and then dropped rows that do not contain kit image links or are images of away kits, kits used in single game or links to shops to buy replicas. This filtering was based on the image label or image URL and performed with the aid of the str\_detect() function from [stringr](https://cran.r-project.org/web/packages/stringr/).

library(stringr)

scrape\_img\_url <- function(html){

html %>%

html\_nodes(".float p , .float img") %>%

html\_attr("src") %>%

tbl\_df() %>%

set\_names("img\_url") %>%

mutate(label = html %>%

html\_nodes(".float p , .float img") %>%

html\_text() %>%

c(., NA) %>%

.[-1])

}

d1 <- htmls %>%

map\_dfr(scrape\_img\_url) %>%

filter(str\_detect(string = img\_url, pattern = "/international/england"),

!str\_detect(string = label, pattern = "change|alternate|Alternate|Change"),

!str\_detect(string = label, pattern = " v |Third"),

!str\_detect(string = img\_url, pattern = "lithuania|italy|yellow|red"))

head(d1)

## img\_url label

## 1 /international/england/images/england-1872.gif 1872

## 2 /international/england/images/england-1882.gif 1879-1900

## 3 /international/england/images/england-1900.gif 1900-1914

## 4 /international/england/images/england-1920-1932.gif 1920-1930

## 5 /international/england/images/england-1921.gif 1930-1934

## 6 /international/england/images/england-1934.gif 1934

Given these URLs I then downloaded each of the images which are stored in a single R object kits

library(magick)

kits <- d1 %>%

mutate(img\_url = paste0("http://www.historicalkits.co.uk", img\_url),

img\_url = str\_replace(string =img\_url, pattern =" ", replacement = "%20")) %>%

select(img\_url) %>%

map(image\_read) %>%

set\_names("img")

Typing kits into R will display each kit in the RStudio viewer (it will quickly run through each image). The console displays summary information for each image in the kits object.

> kits

$img

format width height colorspace filesize

1 GIF 170 338 sRGB 0

2 GIF 170 338 sRGB 0

3 GIF 170 338 sRGB 0

4 GIF 170 338 sRGB 0

5 GIF 170 338 sRGB 0

6 GIF 170 338 sRGB 0

7 GIF 170 338 sRGB 0

8 GIF 170 338 sRGB 0

9 GIF 170 338 sRGB 0

10 GIF 170 338 sRGB 0

**Annotating Images**

Before creating any GIF I wanted add annotations for the year and the copyright information. To do this I first created a border using image\_border() in magick and then image\_annotate() to add the text. I wrapped these edits into an add\_text() function and then applied each to the kit images.

add\_text <- function(img, label){

img %>%

image\_border(geometry = "10x60", color = "white") %>%

image\_chop("0x45") %>%

image\_annotate(text = label, gravity = "north") %>%

image\_annotate(

text = "Animation by @guyabelguyabel", gravity = "south", location = "+0+45"

) %>%

image\_annotate(

text = "Images are Copyright of Historical\nFootball Kits and reproduced by\nkind permission.",

gravity = "south"

)

}

for(i in 1:length(kits$img)){

kits$img[i] <- add\_text(img = kits$img[i], label = d1$label[i])

}

**Creating a GIF**

The final step was to bind together the set of images in an animated GIF with smooth transition images between each frame. To do this I used the image\_morph() twice. First to repeat the same image so that the GIF would remain stable for a few frames (kits\_morph1 below). Then again to create a set of morphing images between successive kits (kits\_morph0 below). The full set of frames were stored in kits\_ani

kits\_ani <- image\_morph(c(kits$img[1], kits$img[1]), frames = 5)

for(i in 2:length(kits$img)){

kits\_morph0 <- image\_morph(c(kits$img[i-1], kits$img[i]), frames = 5)

kits\_morph1 <- image\_morph(c(kits$img[i], kits$img[i]), frames = 5)

kits\_ani <- c(kits\_ani, kits\_morph0)

kits\_ani <- c(kits\_ani, kits\_morph1)

}

To create an animation I passed the set of frames in the kits\_morph object to the image\_animate() and image\_write() functions to give the image above.

kits\_ani %>%

image\_animate(fps = 10) %>%

image\_write(path = "england.gif")

**Club Teams**

Similar code as above can be used to create images for club teams. I tried this out for the mighty Reading. As the Reading kits on [Historical Football Kits](http://www.historicalkits.co.uk/Reading/Reading.htm) are on only one page and includes only home kits, finding the image URLs was much easier…

d1 <- read\_html("http://www.historicalkits.co.uk/Reading/Reading.htm") %>%

scrape\_img\_url() %>%

filter(str\_detect(string = img\_url, pattern = "/Reading"),

!str\_detect(string = img\_url, pattern = "unknown")) %>%

mutate(

label = str\_replace\_all(string = label,

pattern = "[:alpha:]|\\s",

replacement = "")

)