

FIDE Chess Download

Anuj Dahiya

December 22, 2019

Introduction

The purpose of this document is to show how I downloaded all of the standard rating files from the FIDE organizations website.

Import libraries

I don't think any packages are necessary to download any of the files. Still though, I use `knitr` and `dplyr` to display a table of urls that we will call on to download from later on.

```
library(knitr)
library(dplyr)
```

Set up working directory

Before proceeding, we need to set up a working directory to store all of the downloaded files. Make sure to have an empty folder you can dump the files into. I've created a `Downloaded files` subfolder within the folder of this `.Rmd` document.

IMPORTANT: Please adjust path to the path of the folder you want the data to be in.

If you're not sure how to set your directory, you can choose `path` interactively using `choose.dir()`. If you choose to do so, make sure to comment out the first line and uncomment out the 2nd line.

```
path = "~/GitHub/FIDE/Chess Scripts/Step 1 - Download/Downloaded files"
#path = choose.dir()

opts_knit$set(root.dir = path)
```

FIDE file format

R's default is to adjust all numbers, beginning with 0, and truncatte the leading zeros.

Below we setup many useful vectors of strings that will be useful for creating urls that we will download the `.zip` files from.

```
year_vector <- as.character(1:as.numeric(substr(Sys.Date(), 3, 4)))
month_vector <- tolower(substr(month.name, 1, 3))
url_vector = url_vect_destfile = rep(0, length(month_vector)*length(year_vector))

for(i in 1:length(year_vector)){
  if(nchar(year_vector[i]) == 1){
```

```

    year_vector[i] = paste("0", year_vector[i], sep = "")
  }
}

latest <- format(Sys.Date(), format="%b%Y")
latest <- paste(tolower(substr(latest, 1, 1)), substr(latest, 2, 3), substr(latest, nchar(latest)-1, nch

latest

## [1] "dec19frl.zip"

```

`latest` tells us the most recent file FIDE should have hosted on their website.

Below, we put it all together using a somewhat ugly for loop, but it accomplishes our job nicely to create the urls we desire. Note that in August 2014 (which explains the `else if` statement at the 140th iteration), FIDE added an extra word, “standard_” to the URLs to be downloaded from.

```

for(i in seq_along(year_vector)){
  for(j in seq_along(month_vector)){
    if(12*(i-1)+j <= 140){
      url_vector[12*(i-1)+j] = paste("http://ratings.fide.com/download/", month_vector[j], year_vector[i])
    }
    else if(12*(i-1)+j > 140){
      url_vector[12*(i-1)+j] = paste("http://ratings.fide.com/download/standard_", month_vector[j], year_vector[i])
    }
    url_vect_destfile[12*(i-1)+j] <- substr(url_vector[12*(i-1)+j], nchar(url_vector[12*(i-1)+j])- 11 , 11)
  }
}

url_vect_destfile <- url_vect_destfile[1:which(url_vect_destfile == latest)]
url_vector <- url_vector[1:which(url_vect_destfile == latest)]

rm(list=setdiff(ls(), c("url_vector", "url_vect_destfile")))

```

URLs and their files

URL	File
http://ratings.fide.com/download/jan01frl.zip	jan01frl.zip
http://ratings.fide.com/download/feb01frl.zip	feb01frl.zip
http://ratings.fide.com/download/mar01frl.zip	mar01frl.zip
http://ratings.fide.com/download/apr01frl.zip	apr01frl.zip
http://ratings.fide.com/download/may01frl.zip	may01frl.zip
http://ratings.fide.com/download/standard_aug19frl.zip	aug19frl.zip
http://ratings.fide.com/download/standard_sep19frl.zip	sep19frl.zip
http://ratings.fide.com/download/standard_oct19frl.zip	oct19frl.zip
http://ratings.fide.com/download/standard_nov19frl.zip	nov19frl.zip
http://ratings.fide.com/download/standard_dec19frl.zip	dec19frl.zip

Above is a trimmed table of URLs and their corallary files. This is what will be inputted in the `download_all()` function down below.

Download all chess files silently

I'm fairly sure you can remove the two lines involving `old`, but I'd keep them in case you start to get spammed with messages.

`download_all()` quietly checks each URL we will visit and downloads the file present at each URL. If a file doesn't exist at a URL (many of the early year URLs don't), then the function skips over the error.

Note: Download times vary!

```
download_all <- function(link, dest){  
  if (!file.exists(dest)) {  
    tryCatch({  
      download.file(link, dest, method="auto", quiet = TRUE)  
    }, error=function(e){})  
  }  
}  
  
old <-getOption("warn"); options(warn = -1)  
invisible(mapply(download_all, url_vector, url_vect_destfile))  
options(warn = old)
```

Unzipping and cleaning

Some brief housecleaning is taken care of below. After the step below, you should only have text files in the directory you set at the beginning. All that's done below is unzipping the `.zip` files and deleting the `.zip` files.

```
invisible(sapply(list.files(pattern = "*.zip"), function(x) unzip(x, exdir = getwd())))  
  
unlink(list.files(pattern = "*.zip"))
```

Lastly, we can verify what is in our directory.

```
list.files(pattern = "*txt")%>%  
head()  
  
## [1] "apr08frl.txt" "apr09frl.txt" "aug12frl.txt" "jan03frl.txt"  
## [5] "jan09frl.txt" "jan10frl.txt"
```

As we can see, we can see the downloaded and unzipped text files: Success!