MaxCs2

Max Moro

9/9/2020

# Functions

getLink = function(year,division,section,page=1){  
 #this function just create the link with the query parameters   
 paste0( 'http://www.cballtimeresults.org/performances'  
 ,'?utf8=%E2%9C%93&section=',section  
 ,'&year=',year,'&division=',division,'&page=', page)  
}  
  
getTable = function(year,division,section,page){  
 #this function reads the page and convert to a dataframe  
   
 #get the link to the apge  
 link=getLink(year,division,section,page=page)  
   
 #read the page, and grab to 'table' tag  
 t=read\_html(link) %>%   
 html\_nodes("table") %>%   
 html\_table(fill=TRUE)   
   
 #get the table and add metadat for the query paramters  
 out = t[[1]] %>%  
 mutate(year=year, divisionTitle=division, section=section, page=page, link=link)  
}  
  
getAllTables = function(years,division,section,useCachedWebData = FALSE){  
 library(progress)  
 library(doParallel)  
 library(doSNOW)  
   
 if(!useCachedWebData) {  
 #init parallel process with max number of cores  
 cl <- makeCluster(detectCores())  
 doSNOW::registerDoSNOW(cl)  
   
 #progress bar for the parallel loop  
 pb <- progress::progress\_bar$new(total = length(years),format='[:bar] :percent :eta')  
 progress <- function(n) pb$tick()  
   
 #start a parellel loop per each year  
 dataRaw=NULL  
 dataRaw = foreach(y=years  
 ,.combine=rbind,.export=c('getTable','getLink')  
 ,.options.snow = list(progress=progress)) %dopar%  
 {  
 library(foreach)  
 library(rvest)  
 library(dplyr)  
 isCompleted=FALSE  
   
 #loop to parse all pages  
 dataRaw=foreach(p=c(1:1000),.combine=rbind) %do%   
 if(!isCompleted) {  
 message('getting year:',y, ' page:',p,appendLF = F)  
 #get the table of the current page  
 table = getTable(year=y  
 ,division=division  
 ,section=section  
 ,page=p)  
 message(' rows:',nrow(table))  
 isCompleted = nrow(table)==0 #if there is record, we are at the last page, no need to read further  
 return(table)  
 }  
 return(dataRaw)  
 }  
 stopCluster(cl)  
 #save the raw data  
 saveRDS(dataRaw,file='../../data/dataRaw.rds')  
 } else {  
 #read teh raw data  
 dataRaw= readRDS(file='../../data/dataRaw.rds')  
 }  
 return(dataRaw)  
}  
  
transformAllTables =function(dataRaw){  
 #fix columns and content  
 data = dataRaw %>%  
 separate(col='Name',c('Name','Gender'),sep='[\\(\\)]'  
 ,extra='drop',remove=TRUE) %>% #split name cols to name and gender  
 separate(col='PiS/TiS',c('PiS','TiS'),sep='\\/'  
 ,extra='drop',remove=TRUE) %>% #split PiS/TiS cols  
 separate(col='PiD/TiD',c('PiD','TiD'),sep='\\/'  
 ,extra='drop',remove=TRUE) %>% #split PiD/TiD cols  
 separate(col='Hometown',c('Hometown','HomeState')  
 ,sep=',',extra='merge',remove=TRUE,fill='right') %>% #split Hometown and state  
 mutate(Hometown=trimws(Hometown) #remove trim space  
 ,Hometown = ifelse(Hometown %in% c('NR',''),NA, Hometown) #convert NR and '' to NA  
 ,HomeState = trimws(HomeState) #remove trim space  
 ,HomeState = ifelse(HomeState == '',NA, HomeState) #convert '' to NA  
 ,HomeState = trimws(toupper(ifelse(is.na(HomeState),Hometown,HomeState))) #if state is null, it is saved in the hometown  
 ,HomeCountry = ifelse(HomeState %in% toupper(c('DC',state.abb)), 'USA',HomeState) #check if is a USA state  
 ,Hometown = ifelse(toupper(Hometown) == HomeState,NA,Hometown) #set town to NA if is the same of state  
 ,HomeState = ifelse(HomeState == HomeCountry,NA,HomeState) #set State to NA if is = to coutnry  
 ,Race = trimws(substr(Race,5,100)) #gram race type race cols  
 ,Time = parse\_date\_time(gsub('\\\*\\\*','',Time)  
 , orders = c("HMS", "MS")) #convert time col to time format  
 ,TimeMins = second(Time)/60 + minute(Time) + hour(Time)\*60 #convert time to minus  
 ,Pace = parse\_date\_time(gsub('\\\*\\\*','',Pace)  
 , orders = c("HMS", "MS")) #convert pace col to time format  
 ,PaceMins = second(Pace)/60 + minute(Pace) + hour(Pace)\*60 #convert pace time to mins  
 ,Age = as.numeric(Age) #convert age to numeric  
 ,PiS = as.numeric(PiS) #convert PiS to numeric  
 ,TiS = as.numeric(TiS) #convert TiS to numeric  
 ,PiD = as.numeric(PiD) #convert PiD to numeric  
 ,TiD = as.numeric(TiD) #convert TiD to numeric  
 )   
   
 #fixing missing home state (the state is save as town)  
 return(data )  
}

# Run ETL

dataRaw=getAllTables(years=years,division = division,section=section,useCachedWebData = TRUE)

## Loading required package: parallel

## Loading required package: snow

##   
## Attaching package: 'snow'

## The following objects are masked from 'package:parallel':  
##   
## clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,  
## clusterExport, clusterMap, clusterSplit, makeCluster, parApply,  
## parCapply, parLapply, parRapply, parSapply, splitIndices,  
## stopCluster

data=transformAllTables(dataRaw=dataRaw)

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 20 rows [3, 8,  
## 17, 2176, 7135, 7766, 8777, 9680, 10831, 18391, 18399, 18981, 19694, 20735,  
## 21480, 22189, 28388, 29223, 38455, 47506].

## Warning: Problem with `mutate()` input `Age`.  
## i NAs introduced by coercion  
## i Input `Age` is `as.numeric(Age)`.

## Warning in mask$eval\_all\_mutate(dots[[i]]): NAs introduced by coercion

## Warning: Problem with `mutate()` input `PiD`.  
## i NAs introduced by coercion  
## i Input `PiD` is `as.numeric(PiD)`.

## Warning in mask$eval\_all\_mutate(dots[[i]]): NAs introduced by coercion

if(T) saveRDS(data,file='../../data/data.rds')  
head(data)

## Race Name Gender Age Time Pace  
## 1 10M Jane Omoro W 26 0000-01-01 00:53:37 0000-01-01 00:05:22  
## 2 10M Jane Ngotho W 29 0000-01-01 00:53:38 0000-01-01 00:05:22  
## 3 10M Lidiya Grigoryeva W NA 0000-01-01 00:53:40 0000-01-01 00:05:22  
## 4 10M Eunice Sagero W 20 0000-01-01 00:53:55 0000-01-01 00:05:24  
## 5 10M Alla Zhilyayeva W 29 0000-01-01 00:54:08 0000-01-01 00:05:25  
## 6 10M Teresa Wanjiku W 24 0000-01-01 00:54:10 0000-01-01 00:05:25  
## PiS TiS Division PiD TiD Hometown HomeState year divisionTitle section page  
## 1 1 2358 W2529 1 559 <NA> <NA> 1999 Overall+Women 10M 1  
## 2 2 2358 W2529 2 559 <NA> <NA> 1999 Overall+Women 10M 1  
## 3 3 2358 NR NA NA <NA> <NA> 1999 Overall+Women 10M 1  
## 4 4 2358 W2024 1 196 <NA> <NA> 1999 Overall+Women 10M 1  
## 5 5 2358 W2529 3 559 <NA> <NA> 1999 Overall+Women 10M 1  
## 6 6 2358 W2024 2 196 <NA> <NA> 1999 Overall+Women 10M 1  
## link  
## 1 http://www.cballtimeresults.org/performances?utf8=%E2%9C%93&section=10M&year=1999&division=Overall+Women&page=1  
## 2 http://www.cballtimeresults.org/performances?utf8=%E2%9C%93&section=10M&year=1999&division=Overall+Women&page=1  
## 3 http://www.cballtimeresults.org/performances?utf8=%E2%9C%93&section=10M&year=1999&division=Overall+Women&page=1  
## 4 http://www.cballtimeresults.org/performances?utf8=%E2%9C%93&section=10M&year=1999&division=Overall+Women&page=1  
## 5 http://www.cballtimeresults.org/performances?utf8=%E2%9C%93&section=10M&year=1999&division=Overall+Women&page=1  
## 6 http://www.cballtimeresults.org/performances?utf8=%E2%9C%93&section=10M&year=1999&division=Overall+Women&page=1  
## HomeCountry TimeMins PaceMins  
## 1 KENYA 53.61667 5.366667  
## 2 KENYA 53.63333 5.366667  
## 3 RUSSIA 53.66667 5.366667  
## 4 KENYA 53.91667 5.400000  
## 5 RUSSIA 54.13333 5.416667  
## 6 KENYA 54.16667 5.416667

data %>%  
 group\_by(year) %>%  
 summarize(rows = n())

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 14 x 2  
## year rows  
## <int> <int>  
## 1 1999 2358  
## 2 2000 2166  
## 3 2001 2972  
## 4 2002 3333  
## 5 2003 3526  
## 6 2004 3885  
## 7 2005 4331  
## 8 2006 5435  
## 9 2007 5532  
## 10 2008 6395  
## 11 2009 8323  
## 12 2010 8853  
## 13 2011 9030  
## 14 2012 9727

# Check Data Structurs

summary(data)

## Race Name Gender Age   
## Length:75866 Length:75866 Length:75866 Min. : 7.00   
## Class :character Class :character Class :character 1st Qu.:27.00   
## Mode :character Mode :character Mode :character Median :32.00   
## Mean :33.85   
## 3rd Qu.:39.00   
## Max. :87.00   
## NA's :20   
## Time Pace PiS   
## Min. :0000-01-01 00:51:44 Min. :0000-01-01 00:05:10 Min. : 1   
## 1st Qu.:0000-01-01 01:28:39 1st Qu.:0000-01-01 00:08:52 1st Qu.:1356   
## Median :0000-01-01 01:37:29 Median :0000-01-01 00:09:45 Median :2786   
## Mean :0000-01-01 01:38:13 Mean :0000-01-01 00:09:50 Mean :3305   
## 3rd Qu.:0000-01-01 01:46:58 3rd Qu.:0000-01-01 00:10:42 3rd Qu.:4905   
## Max. :0000-01-01 02:57:31 Max. :0000-01-01 00:17:45 Max. :9729   
##   
## TiS Division PiD TiD   
## Min. :2166 Length:75866 Min. : 1.0 Min. : 1   
## 1st Qu.:4333 Class :character 1st Qu.: 165.0 1st Qu.: 559   
## Median :6395 Mode :character Median : 404.0 Median : 953   
## Mean :6609 Mean : 595.6 Mean :1190   
## 3rd Qu.:8853 3rd Qu.: 816.0 3rd Qu.:1678   
## Max. :9729 Max. :5302.0 Max. :2803   
## NA's :20 NA's :20   
## Hometown HomeState year divisionTitle   
## Length:75866 Length:75866 Min. :1999 Length:75866   
## Class :character Class :character 1st Qu.:2005 Class :character   
## Mode :character Mode :character Median :2008 Mode :character   
## Mean :2007   
## 3rd Qu.:2010   
## Max. :2012   
##   
## section page link HomeCountry   
## Length:75866 Min. : 1.0 Length:75866 Length:75866   
## Class :character 1st Qu.: 68.0 Class :character Class :character   
## Mode :character Median :140.0 Mode :character Mode :character   
## Mean :165.7   
## 3rd Qu.:246.0   
## Max. :487.0   
##   
## TimeMins PaceMins   
## Min. : 51.73 Min. : 5.167   
## 1st Qu.: 88.65 1st Qu.: 8.867   
## Median : 97.48 Median : 9.750   
## Mean : 98.22 Mean : 9.823   
## 3rd Qu.:106.97 3rd Qu.:10.700   
## Max. :177.52 Max. :17.750   
##