Analysis

11/18/2019

The purpose of this document is to make use of the data stored within the `FIDE` data folder.

Before beginning, we need to import several packages that help run the code below.

```r
library(tidyverse)
library(data.table)
library(xts)
library(dygraphs)
library(foreach)
library(doParallel)
```

`tidyverse` imports dplyr, tidyr and several other useful packages for data wrangling and manipulation.

`data.table` is used purely for speeding up imports.

Access the data

The data is stored in a path we have to manipulate to get to.

```r
path = "~/GitHub/Chess_data/FIDE Data/Data NOV19/"
temp = list.files(path = path,
                  pattern = "\\.csv?")

head(temp)
```

```
## [1] "APR01.csv" "APR02.csv" "APR03.csv" "APR04.csv" "APR05.csv" "APR06.csv"
```

Create functions to rename datasets

Below, I define a few functions that help us rename the datasets eventually

```r
num <- function(x) match(tolower(x), tolower(month.abb))
month <- function(x){return(substr(x, 1, 3))}
add_zero <- function(x){if (x <= 9){x = paste("0", x, sep = "")}; return(x)}
```

Get the month number of all of the files in the dataset

Below, I rename create the variables names when they are imported in memory.

```r
temp%>%
sapply(month)%>%
sapply(num)%>%
sapply(add_zero)%>%
paste("20", substr(temp, 4, 5), "-", ., "-", "01", sep = "")-> month_num

head(month_num)
```

```
## [1] "2001-04-01" "2002-04-01" "2003-04-01" "2004-04-01" "2005-04-01"
## [6] "2006-04-01"
```

We can see that the datasets correspond to dates on which the is recorded.

Import a sample of datasets

Due to the data being quite large, your RAM may be used heavily. Because of this, I have chosen to randomly sample an `n` number of datasets. Please adjust n to `length(month_num)` if you want to wait for a long time for all of the data to process.

```
proper_temp <- paste(path, temp, sep = "")
n = length(month_num)
dataset_random = sample.int(length(month_num), n)

# cl<-makeCluster(detectCores())
# registerDoParallel(cl)
# temp_short = proper_temp[1:5]
# FIDE<-foreach(i=proper_temp, .export = "fread") %dopar% {
# t <- fread(i, sep = "*", data.table = FALSE, strip.white = TRUE, blank.lines.skip = TRUE)
#                                                                  }
# stopCluster(cl)


for(i in 1:length(proper_temp)) {
assign(month_num[i], fread(proper_temp[i], sep = "*", data.table = FALSE, strip.white = TRUE, blank.lin
}
```

```
#Put datasets in list
FIDE <- mget(ls(pattern = "[0-9][0-9]-[0-9][0-9]"))
rm(list=setdiff(ls(), c("FIDE", "ptm")))

#Help rename columns in the data
vector_months <- c(month.abb, tolower(month.abb),toupper(month.abb))
string = ""
for (i in 1:length(vector_months)){
if (i == 1){string = vector_months[i]}
else if (i > 1) {string = paste(string, vector_months[i], sep = "|")}
}
string = paste(string, "RATING", sep = "|")

# names(FIDE) <- month_num

#Insert month column
for(i in 1:length(FIDE)){
  colnames(FIDE[[i]])[grepl("Name|NAME|name", colnames(FIDE[[i]]))] <- "Name"
  colnames(FIDE[[i]])[grepl("NUMBER", colnames(FIDE[[i]]))] <- "ID_Number"
  colnames(FIDE[[i]])[grepl("Fed|FED|COUNTRY", colnames(FIDE[[i]]))] <- "Country"
  colnames(FIDE[[i]])[grepl("Gms|GAMES|GM|Game|GAME", colnames(FIDE[[i]]))] <- "Games"
  colnames(FIDE[[i]])[grepl("K", colnames(FIDE[[i]]))] <- "K_factor"
  colnames(FIDE[[i]])[grepl("FLAG|Flag|flag", colnames(FIDE[[i]]))] <- "Activity"
  colnames(FIDE[[i]])[colnames(FIDE[[i]]) %in% c("Wtit","wtit","WTIT", "WTit")] <- "Womens_Title"
  colnames(FIDE[[i]])[colnames(FIDE[[i]]) %in% c("TITLE","Title","title","Tit")] <- "Title"
  colnames(FIDE[[i]])[grepl(string, colnames(FIDE[[i]]))] <- "Rating"
  colnames(FIDE[[i]])[grepl("Born|Age|age|BIRTHDAY|B-day|Bday", colnames(FIDE[[i]]))] <- "Age_Birthday"
  colnames(FIDE[[i]])[grepl("SEX", colnames(FIDE[[i]]))] <- "Sex"
  colnames(FIDE[[i]])[grepl("FOA", colnames(FIDE[[i]]))] <- "FIDE_Online_Arena"
  colnames(FIDE[[i]])[grepl("OTit", colnames(FIDE[[i]]))] <- "Other_Titles"



  FIDE[[i]] <-  FIDE[[i]] %>%
              mutate(Date = as.Date(names(FIDE)[i]),
                     Rating = as.numeric(Rating))


}
```

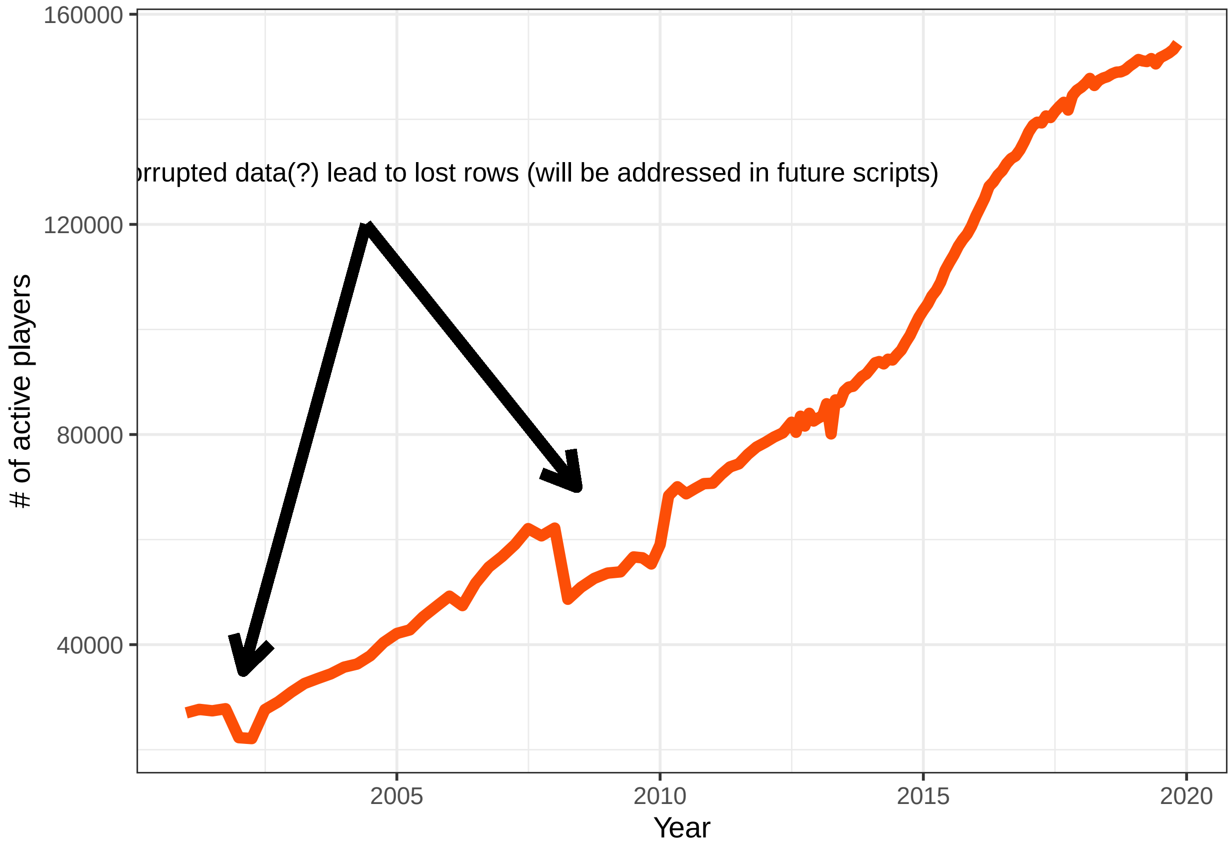
```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```
FIDE <- rbindlist(FIDE, fill = TRUE)%>%
        select(-V1)





# fwrite(FIDE, "FIDE.csv")
# FIDE <- fread("FIDE.csv", data.table = FALSE)
```

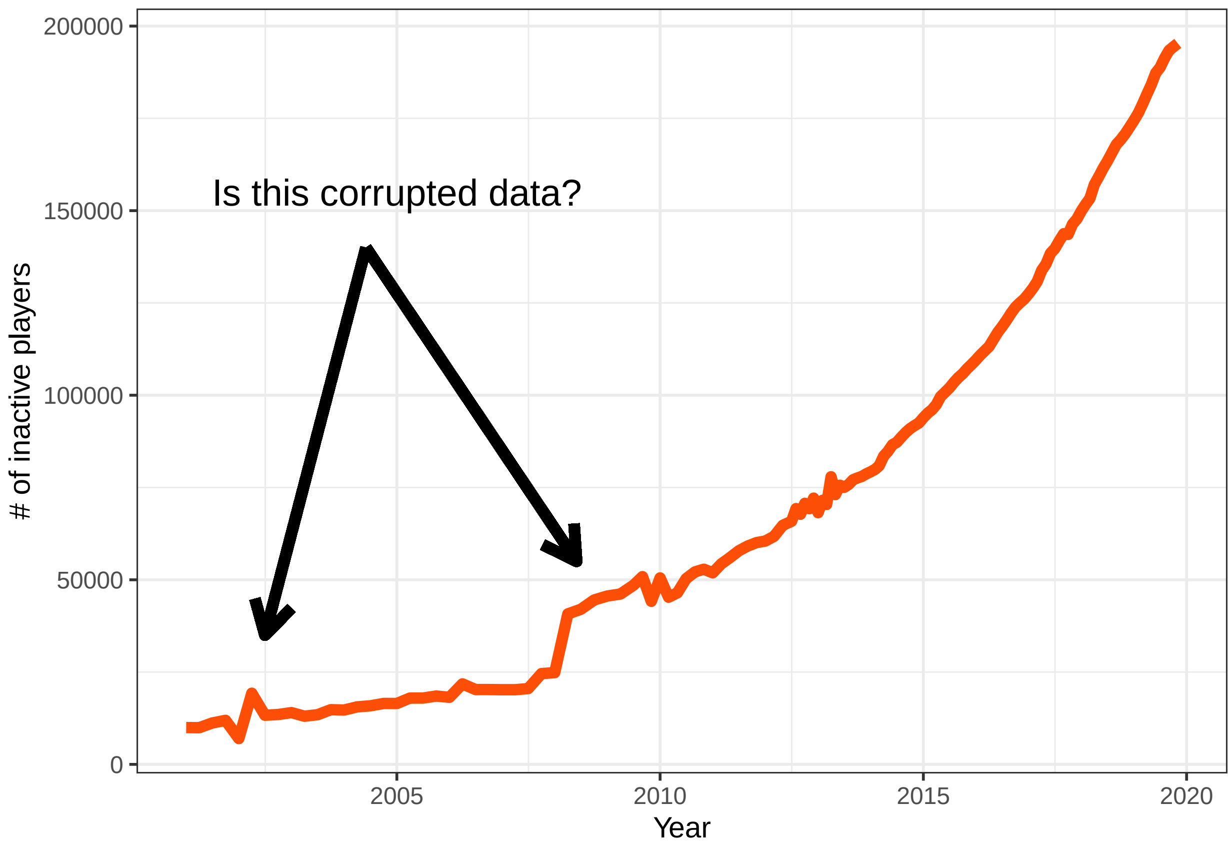
Now that we have the data in a more clean, usable format, it’s time we analyzed it.

```
# tabled <- table(FIDE$Activity)%>%
#           data.frame()%>%
#           mutate(Var1 = as.character(Var1))
#
#
# summary <- FIDE%>%
#            filter(Activity %in% tabled$Var1[1:4])%>%
#            group_by(Country, Date)%>%
#            summarise(Rating = mean(Rating, na.rm = T),
#                      Population = n())

old = c("c", "wc", "WC", "wg", "WF", "wf", "g", "m", "f", "wm", "gm" )
new = c("CM", "WCM", "WCM", "WGM", "WFM", "WFM", "GM", "IM", "FM", "WIM", "GM")
FIDE <- FIDE%>%
        mutate(Title= c(new, Title)[match(Title, c(old, Title))])

Strange <- FIDE%>%
           filter(!Title %in% c(new, ""))


Active_player <- FIDE%>%
                 filter(Activity == "")%>%
                 group_by(Date)%>%
                 summarise(total_count = n(),
                           avg_rating = mean(Rating, na.rm = T),
                           sd_rating = sd(Rating, na.rm = T))


Inactive_player <- FIDE%>%
                   filter(Activity != "")%>%
                   group_by(Date)%>%
                   summarise(total_count = n(),
                             avg_rating = mean(Rating, na.rm = T),
                             sd_rating = sd(Rating, na.rm = T))
```

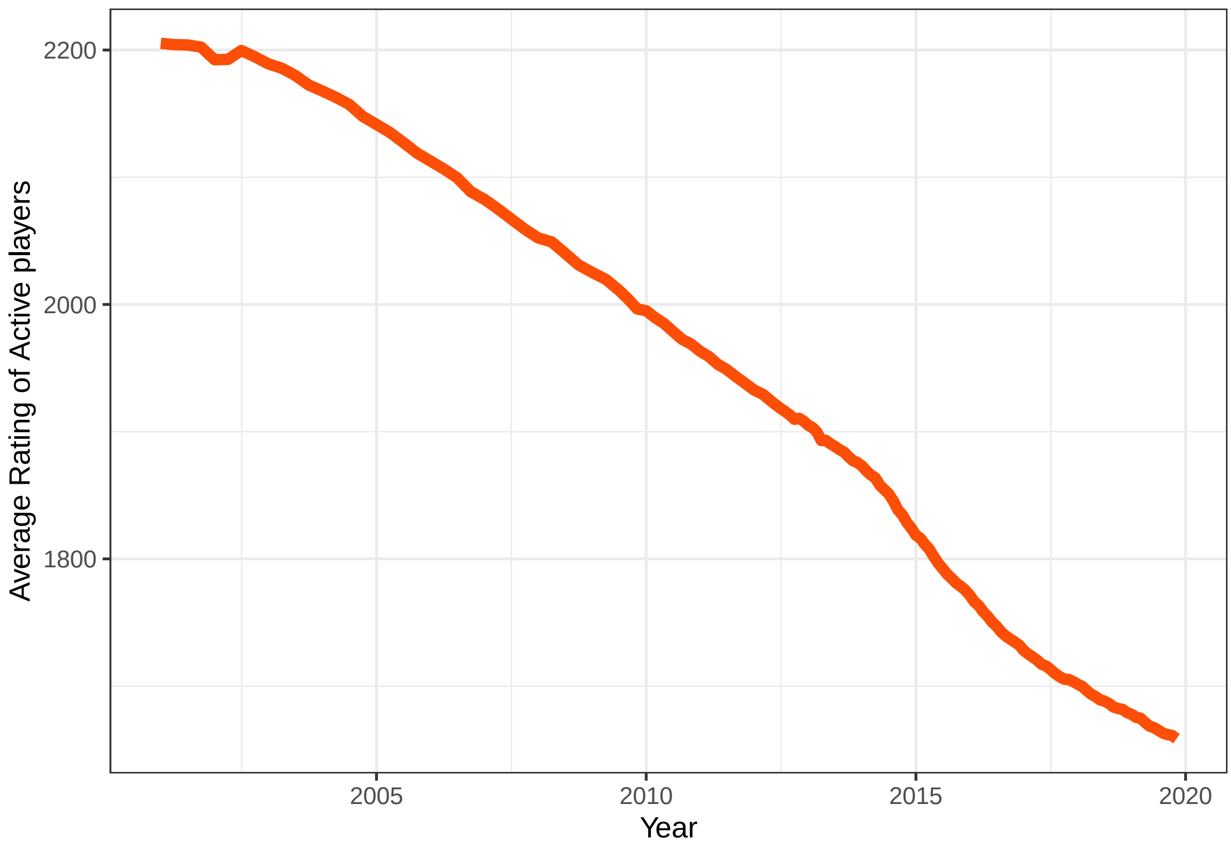
```
ggplot(data = Active_player, aes(x = Date, y = total_count)) +
  geom_line(color = "#FC4E07", size = 2)+
  xlab("Year")+
  ylab("# of active players")+
  geom_segment(aes(x=as.Date("2004-06-01"), xend=as.Date("2008-06-01"), y=120000, yend=70000),
               arrow = arrow(length = unit(.5, "cm")), size = 2)+
  geom_segment(aes(x=as.Date("2004-06-01"), xend=as.Date("2002-02-01"), y=120000, yend=35000),
               arrow = arrow(length = unit(.5, "cm")), size = 2)+
  annotate("text", x = as.Date("2007-06-01"), y = 130000,
           label = "Corrupted data(?) lead to lost rows (will be addressed in future scripts)", size =
  theme_bw()
```



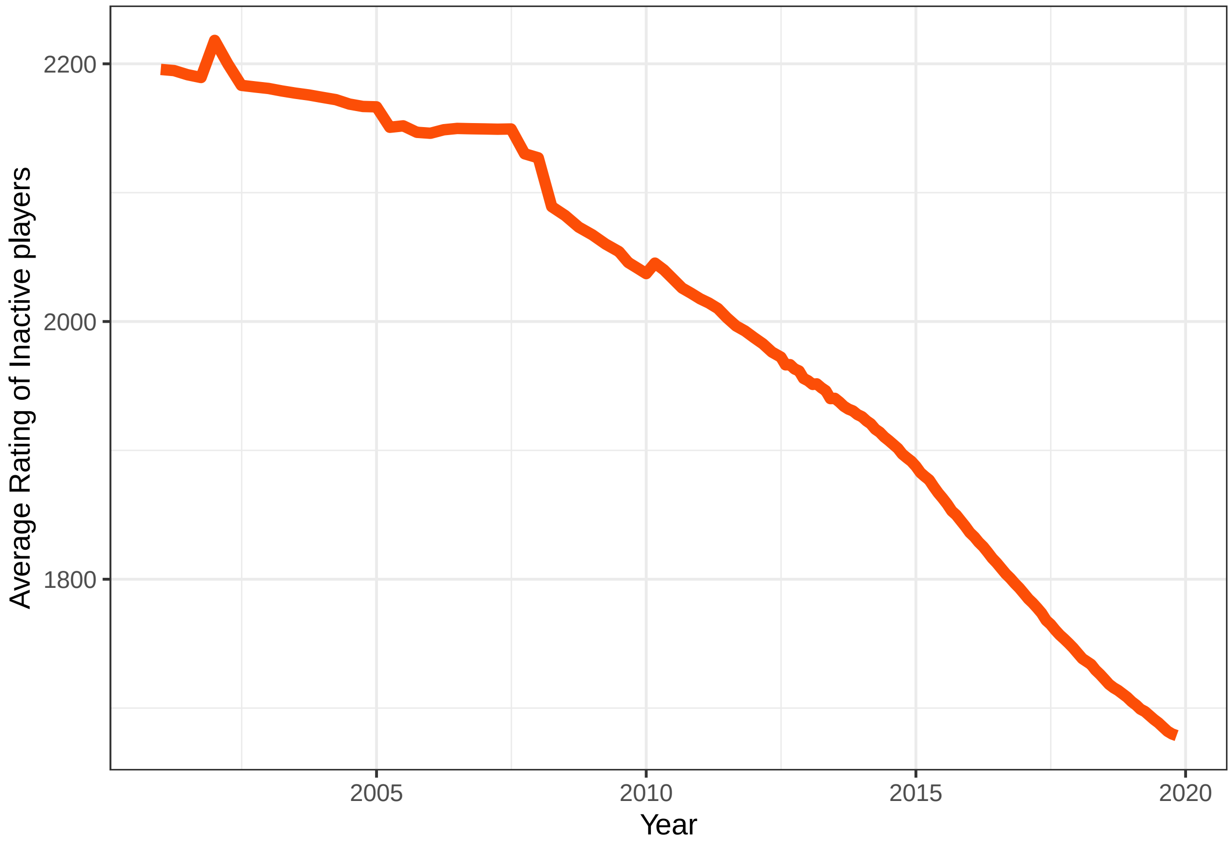
```r
ggplot(data = Inactive_player, aes(x = Date, y = total_count)) +
  geom_line(color = "#FC4E07", size = 2)+
  xlab("Year")+
  ylab("# of inactive players")+
  geom_segment(aes(x=as.Date("2004-06-01"), xend=as.Date("2008-06-01"), y=140000, yend=55000),
               arrow = arrow(length = unit(.5, "cm")), size = 2)+
  geom_segment(aes(x=as.Date("2004-06-01"), xend=as.Date("2002-07-01"), y=140000, yend=35000),
               arrow = arrow(length = unit(.5, "cm")), size = 2)+
  annotate("text", x = as.Date("2005-01-01"), y = 155000,
           label = "Is this corrupted data?", size = 5)+
  theme_bw()
```



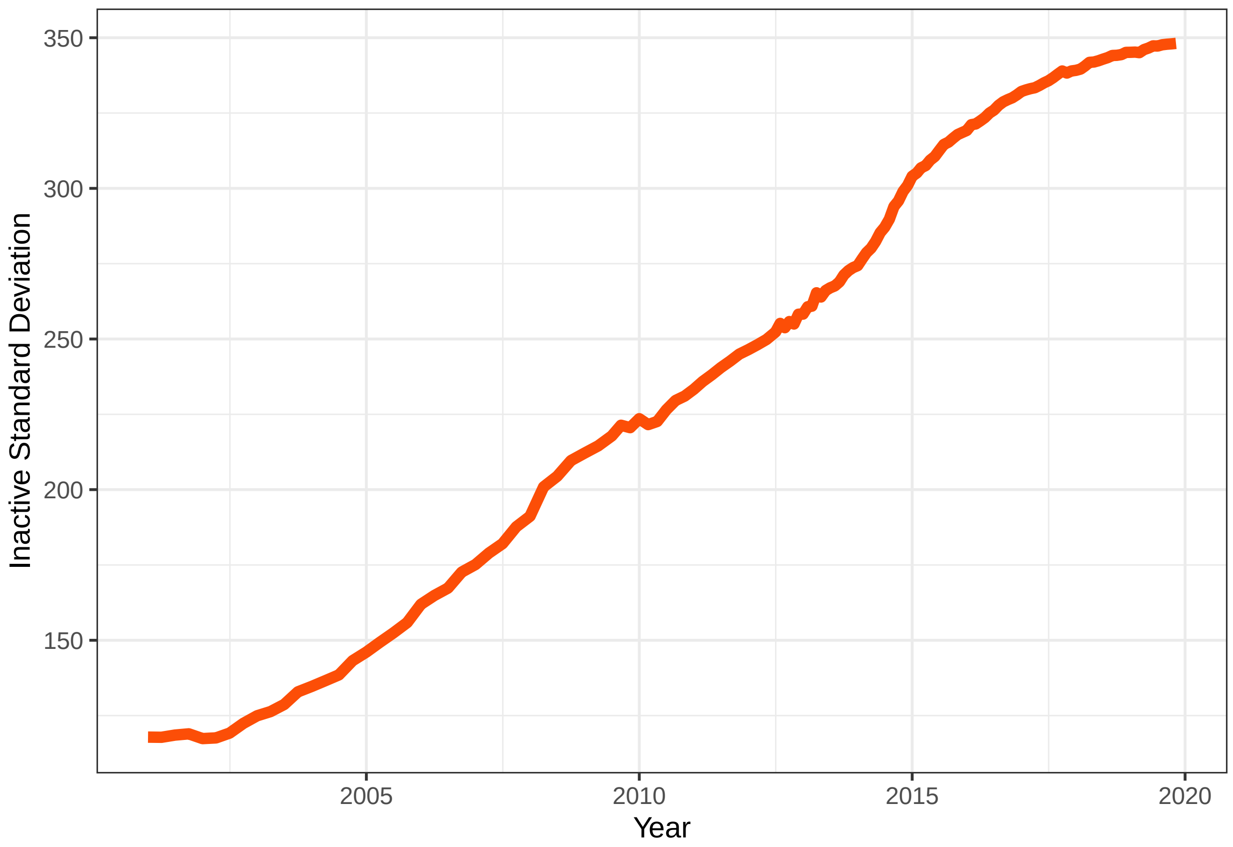
```
ggplot(data = Active_player, aes(x = Date, y = avg_rating)) +
  geom_line(color = "#FC4E07", size = 2)+
  xlab("Year")+
  ylab("Average Rating of Active players")+
  theme_bw()
```



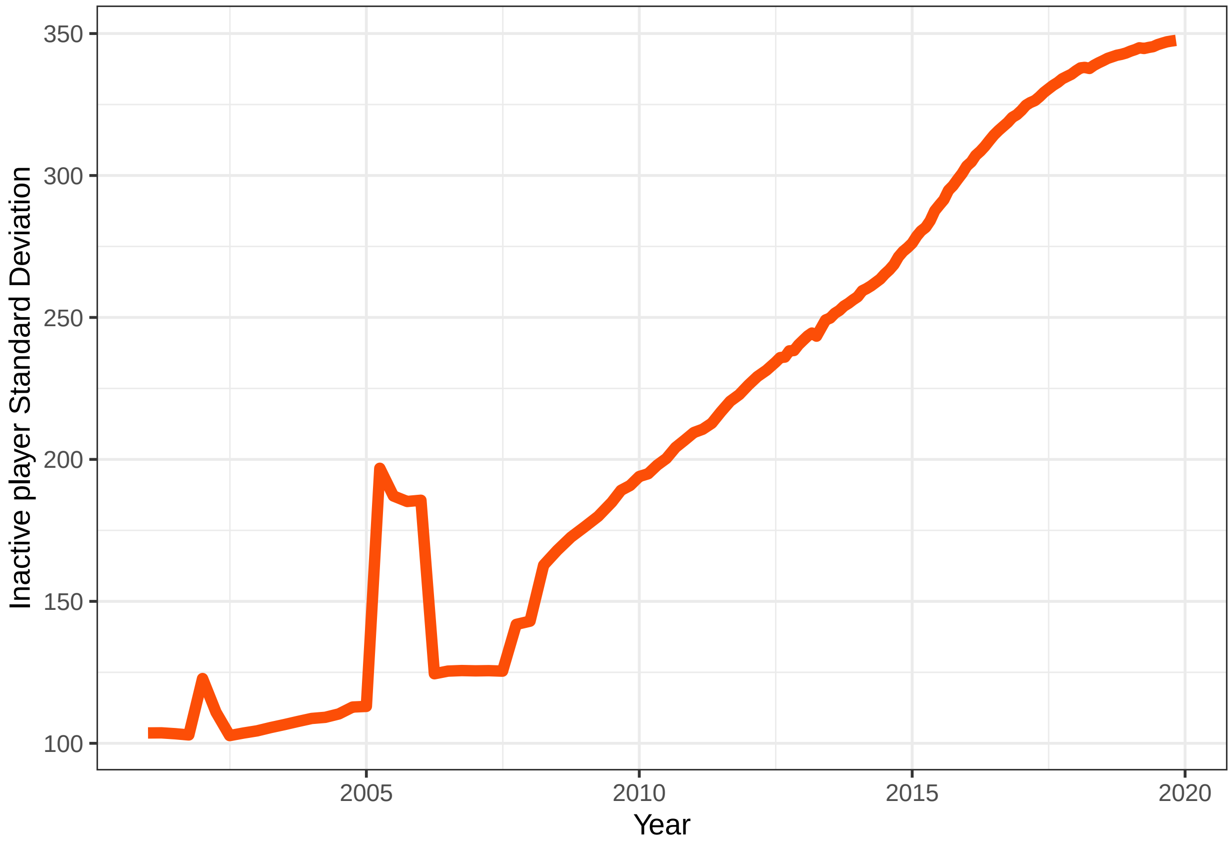
```
ggplot(data = Inactive_player, aes(x = Date, y = avg_rating)) +
  geom_line(color = "#FC4E07", size = 2)+
  xlab("Year")+
  ylab("Average Rating of Inactive players")+
  theme_bw()
```



```r
ggplot(data = Active_player, aes(x = Date, y = sd_rating)) +
  geom_line(color = "#FC4E07", size = 2)+
  xlab("Year")+
  ylab("Inactive Standard Deviation")+
  theme_bw()
```



```
ggplot(data = Inactive_player, aes(x = Date, y = sd_rating)) +
  geom_line(color = "#FC4E07", size = 2)+
  xlab("Year")+
  ylab("Inactive player Standard Deviation")+
  theme_bw()
```



```
# FIDE%>%
# filter(Rating > 2830)%>%
# .$Name%>%
# unique() -> top_player
#
# #Visualize the data
# WC_caliber_players <- FIDE%>%
#   filter(Name %in% top_player, Activity == "")%>%
#   select(Name, Rating, Date)%>%
#   arrange(Name, Date)
#
#
# dygraphed <- WC_caliber_players %>%
#               spread(key = Name, value = Rating)%>%
#               xts(.[,which(colnames(.)!= "Date")],order.by =  .$Date)
#
# dygraph(dygraphed, main = "Player's rating over Time") %>%
#   dyOptions(drawPoints = TRUE, pointSize = 2, axisLineWidth = 4) %>%
#   dyAxis("y", label = "Rating", valueRange = c(1900, 3200))%>%
#   dyRangeSelector()%>%
#   dyLegend(width = 400)
```

```
# data <- fread("SEP19.csv", sep = "*", data.table = FALSE)
#
```

```
# library(ggplot2)
# data%>%
# mutate(Rating = as.numeric(NOV19))%>%
# filter(Flag == "wi")%>%
# na.omit()%>%
# ggplot(aes_string(x="Rating"))+
# geom_density(size=2, alpha=.4)+
# geom_histogram(aes(y = ..density..), bins = 50, col= "red")+
# labs(title="Histogram overlayed with Density curve") +
# labs(x="Rating", y="Percentage of players in population")
```