Exploratory data analysis - violence in the world and woman grand master likelihood.

ldurazo

28/11/20

In this project we are going to analyze to different set of data, and see what they tell us about the likelihood of a woman grandmaster chess appearing and how it relates to violence against women in said country, this will imply a huge amount of data cleaning and interpretation before arriving to a relevant correlation. We expect in advance and acknowledge that other variables such as economic variables per country may correlate more strongly to both violence and women grandmaster players.

We may also use the term grandmaster interchangeably with just the women top player, this is both a practicality and an intentional way to recognize women in countries were grandmasters are less likely to appear but still have top ranked players.

The next block is the setup script to load data, and setup this notebook utilities.

```
chooseCRANmirror(ind = 52)
# EDA & Kaggle auth packages
install.packages(c("summarytools", "explore", "dataMaid", "devtools", "configr", "rsconnect", "dplyr"))
```

```
##
## The downloaded binary packages are in
##  /var/folders/f3/2p9snhhj759g5l1f2ztjbwdw0000gn/T//RtmpOntCJq/downloaded_packages
```

```
devtools::install_github("ldurazo/kaggler")
```

```
## Skipping install of 'kaggler' from a github remote, the SHA1 (bfb8fb69) has not changed since last in
##   Use 'force = TRUE' to force installation
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(summarytools)
```

```
## Registered S3 method overwritten by 'pryr':
##   method      from
##   print.bytes Rcpp
```

```
## For best results, restart R session and update pander using devtools:: or remotes::install_github('ra
```

```
library(explore)
library(dataMaid)
```

```
##
## Attaching package: 'dataMaid'
```

```
## The following object is masked from 'package:dplyr':
##
##     summarize
```

```
library(configr)
library(readr)
library(rsconnect)
library(kaggler)

# files downloading
kgl_auth(creds_file = 'kaggle.json')
```

```
## <request>
## Options:
## * httpauth: 1
## * userpwd: ldurazo:48ce1a6f4b7269d4b6f8ce2d3b854199
```

```
response_violence <- kgl_datasets_download_all(owner_dataset = "andrewmvd/violence-against-women-and-gi
download.file(response_violence[["url"]], "data/violence_temp.zip", mode = "wb")
unzipResult <- unzip("data/violence_temp.zip", exdir = "data/", overwrite = TRUE)
```

```
## Warning in unzip("data/violence_temp.zip", exdir = "data/", overwrite = TRUE):
## error -1 al extraer del archivo zip
```

```
violence_data <- read_csv("data/makeovermonday-2020w10/violence_data.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   RecordID = col_double(),
##   Country = col_character(),
##   Gender = col_character(),
##   `Demographics Question` = col_character(),
##   `Demographics Response` = col_character(),
##   Question = col_character(),
##   `Survey Year` = col_character(),
##   Value = col_double()
## )
```

```
## Warning: 1 parsing failure.
##   row col  expected    actual                                          file
## 11354  -- 8 columns 6 columns 'data/makeovermonday-2020w10/violence_data.csv'
```

```
response_chessplayers <- kgl_datasets_download_all(owner_dataset = "vikasojha98/top-women-chess-players
download.file(response_chessplayers[["url"]], "data/chess_temp.zip", mode = "wb")
unzipResult <- unzip("data/chess_temp.zip", exdir = "data/", overwrite = TRUE)
```

```
## Warning in unzip("data/chess_temp.zip", exdir = "data/", overwrite = TRUE):
## error -1 al extraer del archivo zip
```

```
chess_data <- read_csv("data/top_women_chess_players_aug_2020.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   `Fide id` = col_double(),
##   Name = col_character(),
##   Federation = col_character(),
##   Gender = col_logical(),
##   Year_of_birth = col_double(),
##   Title = col_character(),
##   Standard_Rating = col_double(),
##   Rapid_rating = col_double(),
##   Blitz_rating = col_double(),
##   Inactive_flag = col_character()
## )
```

With these two files we can now see a summary of the data. Note that these two are html generated files available if you run this notebook. Alternatively, the explore package returns interesting results in a shiny app, turn the following statements on if you want to see the data.

```
#dfSummary(violence_data, file = "data/violence_data_summary.html")
#dfSummary(chess_data, file = "data/violence_data_summary.html")

#explore(chess_data)
#explore(violence_data)
```

We will need a file that maps the ISO-3166 country alpha 3 on the chess data, to the country name in violence data.

```
download.file("https://raw.githubusercontent.com/lukes/ISO-3166-Countries-with-Regional-Codes/master/al
countries_mapping <- read_csv("data/iso-3166")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   name = col_character(),
##   `alpha-2` = col_character(),
##   `alpha-3` = col_character(),
##   `country-code` = col_character(),
```

```
##   `iso_3166-2` = col_character(),
##   region = col_character(),
##   `sub-region` = col_character(),
##   `intermediate-region` = col_character(),
##   `region-code` = col_character(),
##   `sub-region-code` = col_character(),
##   `intermediate-region-code` = col_character()
## )
```

```
countries_mapping <- setNames(select(countries_mapping, "name", "alpha-3"), c("name", "code"))
```

Let's clean up our data by removing the NA values and transforming the percentage to a number.

```
violence_data <- na.omit(violence_data, "Value")
violence_data$Value <- as.numeric(sub("%", "", violence_data$Value))
```

Now I want to create an aggregate of the data of my first data set of violence against women, and generate a weighted mean out of the results between men and women answering, so that I can effectively create a "violence score" per country, in a very subjective way. There are a number of better techniques to do such a process, but only for the sake of the excercise we will use this score formula.

```
violence_data$WeightedValue <- ifelse(violence_data$Gender == "F", violence_data$Value * 0.7, violence_
violence_data_slim <- select(violence_data, "Country", "WeightedValue")
violence_data_slim_grouped <- setNames(aggregate(violence_data_slim$WeightedValue, by = list(violence_da

head(violence_data_slim_grouped)
```

```
##       Country     Score
## 1 Afghanistan 22.528810
## 2     Albania  2.452444
## 3      Angola  7.079778
## 4     Armenia  4.011310
## 5  Azerbaijan 16.704583
## 6  Bangladesh 10.303333
```

With the score per country done, we need to do similar work with the chess players data frame.

```
chess_data <- na.omit(chess_data, "Standard_Rating", "Rapid_rating", "Blitz_rating")
chess_data_slim <- select(chess_data, "Federation", "Standard_Rating", "Rapid_rating", "Blitz_rating")
chess_data_slim_grouped <- chess_data_slim %>%
  group_by(chess_data_slim$Federation) %>%
  summarise(across(ends_with("rating"), list(mean = mean, n = length, max = max, min = min)))
```

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

```
head(chess_data_slim_grouped)
```

```
## # A tibble: 6 x 13
##   `chess_data_sli~ Standard_Rating~ Standard_Rating~ Standard_Rating~
##   <chr>                       <dbl>            <int>            <dbl>
## 1 ALB                         1971                1             1971
```

```
## 2 ALG                          1900                 1              1900
## 3 ARG                          1930.                2              1940
## 4 ARM                          2285                 1              2285
## 5 AUS                          1926.                4              2062
## 6 AUT                          1975                 1              1975
## # ... with 9 more variables: Standard_Rating_min <dbl>,
## #   Rapid_rating_mean <dbl>, Rapid_rating_n <int>, Rapid_rating_max <dbl>,
## #   Rapid_rating_min <dbl>, Blitz_rating_mean <dbl>, Blitz_rating_n <int>,
## #   Blitz_rating_max <dbl>, Blitz_rating_min <dbl>
```

Now, we need to join the tables with the countries table in order to finally obtain a single dataset.

```
violence_df <- left_join(violence_data_slim_grouped, countries_mapping, by = c("Country" = "name"))
violence_df %>% arrange(!is.na(violence_df$code))
```

```
##                      Country      Score code
## 1                    Bolivia  5.3954444 <NA>
## 2  Congo Democratic Republic 22.1008333 <NA>
## 3              Cote d'Ivoire 13.1191111 <NA>
## 4            Kyrgyz Republic 10.6180247 <NA>
## 5                    Moldova  5.4929012 <NA>
## 6                   Tanzania 16.2912222 <NA>
## 7                Afghanistan 22.5288095  AFG
## 8                    Albania  2.4524444  ALB
## 9                     Angola  7.0797778  AGO
## 10                   Armenia  4.0113095  ARM
## 11                Azerbaijan 16.7045833  AZE
## 12                Bangladesh 10.3033333  BGD
## 13                     Benin  7.8707778  BEN
## 14              Burkina Faso 10.8768889  BFA
## 15                   Burundi 15.9398889  BDI
## 16                  Cambodia 11.2942222  KHM
## 17                  Cameroon 11.9049444  CMR
## 18                      Chad 23.4807778  TCD
## 19                  Colombia  1.1404444  COL
## 20                   Comoros  9.5546111  COM
## 21                     Congo 19.1671667  COG
## 22        Dominican Republic  0.7340476  DOM
## 23                     Egypt 15.1758333  EGY
## 24                   Eritrea 32.3306667  ERI
## 25                  Eswatini  5.8334444  SWZ
## 26                  Ethiopia 16.6565556  ETH
## 27                     Gabon 12.3352778  GAB
## 28                    Gambia 14.8097222  GMB
## 29                     Ghana  7.0600000  GHA
## 30                 Guatemala  2.3038333  GTM
## 31                    Guinea 21.2895000  GIN
## 32                    Guyana  4.2798889  GUY
## 33                     Haiti  3.5145000  HTI
## 34                  Honduras  3.0452778  HND
## 35                     India 12.5360556  IND
## 36                 Indonesia  7.5503571  IDN
## 37                    Jordan  5.6417778  JOR
```

```
## 38                    Kenya 11.3071111  KEN
## 39                  Lesotho  8.9674444  LSO
## 40                  Liberia 10.6376111  LBR
## 41               Madagascar  8.1401667  MDG
## 42                   Malawi  3.6356111  MWI
## 43                 Maldives  6.1069167  MDV
## 44                     Mali 22.4886667  MLI
## 45                  Morocco 31.6493333  MAR
## 46               Mozambique  3.5043333  MOZ
## 47                  Myanmar 11.9392222  MMR
## 48                  Namibia  7.3674444  NAM
## 49                    Nepal  6.4663889  NPL
## 50                Nicaragua  4.6332222  NIC
## 51                    Niger 17.0797778  NER
## 52                  Nigeria 10.1447778  NGA
## 53                 Pakistan 13.1031548  PAK
## 54                     Peru  1.2701111  PER
## 55              Philippines  3.5443333  PHL
## 56                   Rwanda  9.1943889  RWA
## 57    Sao Tome and Principe  5.3686782  STP
## 58                  Senegal 13.2816667  SEN
## 59             Sierra Leone 16.9027778  SLE
## 60             South Africa  1.6648851  ZAF
## 61               Tajikistan 31.0131111  TJK
## 62              Timor-Leste 24.9147778  TLS
## 63                     Togo  7.3982222  TGO
## 64                   Turkey  4.2592308  TUR
```

```
head(violence_df)
```

```
##       Country     Score code
## 1 Afghanistan 22.528810  AFG
## 2     Albania  2.452444  ALB
## 3      Angola  7.079778  AGO
## 4     Armenia  4.011310  ARM
## 5  Azerbaijan 16.704583  AZE
## 6  Bangladesh 10.303333  BGD
```

Notice that we have a few exemptions where the mapping did not occur correctly, in this instance we will fix them by hand. - Bolivia - Congo Democratic Republic - Cote d'Ivoire - Kyrgyz Republic - Moldova - Tanzania

```
violence_df <- within(violence_df, code[Country == "Bolivia"] <- "BOL")
violence_df <- within(violence_df, code[Country == "Congo Democratic Republic"] <- "COD")
violence_df <- within(violence_df, code[Country == "Cote d'Ivoire"] <- "CIV")
violence_df <- within(violence_df, code[Country == "Kyrgyz Republic"] <- "KGZ")
violence_df <- within(violence_df, code[Country == "Moldova"] <- "MDA")
violence_df <- within(violence_df, code[Country == "Tanzania"] <- "TZA")
violence_df %>% arrange(!is.na(violence_df$code))
```

```
##                     Country      Score code
## 1               Afghanistan 22.5288095  AFG
```

```
## 2                    Albania  2.4524444  ALB
## 3                     Angola  7.0797778  AGO
## 4                    Armenia  4.0113095  ARM
## 5                 Azerbaijan 16.7045833  AZE
## 6                 Bangladesh 10.3033333  BGD
## 7                      Benin  7.8707778  BEN
## 8                    Bolivia  5.3954444  BOL
## 9                Burkina Faso 10.8768889  BFA
## 10                    Burundi 15.9398889  BDI
## 11                   Cambodia 11.2942222  KHM
## 12                   Cameroon 11.9049444  CMR
## 13                       Chad 23.4807778  TCD
## 14                   Colombia  1.1404444  COL
## 15                    Comoros  9.5546111  COM
## 16                      Congo 19.1671667  COG
## 17 Congo Democratic Republic 22.1008333  COD
## 18              Cote d'Ivoire 13.1191111  CIV
## 19        Dominican Republic  0.7340476  DOM
## 20                      Egypt 15.1758333  EGY
## 21                    Eritrea 32.3306667  ERI
## 22                   Eswatini  5.8334444  SWZ
## 23                   Ethiopia 16.6565556  ETH
## 24                      Gabon 12.3352778  GAB
## 25                     Gambia 14.8097222  GMB
## 26                      Ghana  7.0600000  GHA
## 27                  Guatemala  2.3038333  GTM
## 28                     Guinea 21.2895000  GIN
## 29                     Guyana  4.2798889  GUY
## 30                      Haiti  3.5145000  HTI
## 31                   Honduras  3.0452778  HND
## 32                      India 12.5360556  IND
## 33                  Indonesia  7.5503571  IDN
## 34                     Jordan  5.6417778  JOR
## 35                      Kenya 11.3071111  KEN
## 36           Kyrgyz Republic 10.6180247  KGZ
## 37                   Lesotho  8.9674444  LSO
## 38                   Liberia 10.6376111  LBR
## 39                Madagascar  8.1401667  MDG
## 40                    Malawi  3.6356111  MWI
## 41                  Maldives  6.1069167  MDV
## 42                      Mali 22.4886667  MLI
## 43                   Moldova  5.4929012  MDA
## 44                   Morocco 31.6493333  MAR
## 45                Mozambique  3.5043333  MOZ
## 46                   Myanmar 11.9392222  MMR
## 47                   Namibia  7.3674444  NAM
## 48                     Nepal  6.4663889  NPL
## 49                 Nicaragua  4.6332222  NIC
## 50                     Niger 17.0797778  NER
## 51                   Nigeria 10.1447778  NGA
## 52                  Pakistan 13.1031548  PAK
## 53                      Peru  1.2701111  PER
## 54               Philippines  3.5443333  PHL
## 55                    Rwanda  9.1943889  RWA
```

```
## 56     Sao Tome and Principe  5.3686782  STP
## 57                   Senegal 13.2816667  SEN
## 58              Sierra Leone 16.9027778  SLE
## 59              South Africa  1.6648851  ZAF
## 60                Tajikistan 31.0131111  TJK
## 61                  Tanzania 16.2912222  TZA
## 62               Timor-Leste 24.9147778  TLS
## 63                      Togo  7.3982222  TGO
## 64                    Turkey  4.2592308  TUR
```

```
head(violence_df)
```

```
##       Country     Score code
## 1 Afghanistan 22.528810  AFG
## 2     Albania  2.452444  ALB
## 3      Angola  7.079778  AGO
## 4     Armenia  4.011310  ARM
## 5  Azerbaijan 16.704583  AZE
## 6  Bangladesh 10.303333  BGD
```

Now, assuming the FIDE and ISO-3166 codes are the same, let's see how the joined data looks like. Because the countries that have women chess players may not intersect with the countries visited for questionnaire in the violence dataset, I expect plenty of this missed intersections to have NA values. For this analysis we will pay closer attention to the violence score aggregation, and see which countries have top chess players rather than joining all countries in the FIDE and ignore violence score for countries that do not have chess players.

```
merged_df <- left_join(violence_df, chess_data_slim_grouped, by = c("code" = "chess_data_slim$Federatio
merged_df %>% arrange(desc(merged_df$Score))
```

```
##                       Country      Score code Standard_Rating_mean
## 1                     Eritrea 32.3306667  ERI                   NA
## 2                     Morocco 31.6493333  MAR              1853.000
## 3                  Tajikistan 31.0131111  TJK                   NA
## 4                 Timor-Leste 24.9147778  TLS                   NA
## 5                       Chad 23.4807778  TCD                   NA
## 6                 Afghanistan 22.5288095  AFG                   NA
## 7                       Mali 22.4886667  MLI                   NA
## 8  Congo Democratic Republic 22.1008333  COD                   NA
## 9                      Guinea 21.2895000  GIN                   NA
## 10                     Congo 19.1671667  COG                   NA
## 11                     Niger 17.0797778  NER                   NA
## 12               Sierra Leone 16.9027778  SLE                   NA
## 13                 Azerbaijan 16.7045833  AZE              2103.375
## 14                   Ethiopia 16.6565556  ETH                   NA
## 15                   Tanzania 16.2912222  TZA                   NA
## 16                    Burundi 15.9398889  BDI                   NA
## 17                      Egypt 15.1758333  EGY              1995.000
## 18                     Gambia 14.8097222  GMB                   NA
## 19                    Senegal 13.2816667  SEN                   NA
## 20              Cote d'Ivoire 13.1191111  CIV                   NA
## 21                   Pakistan 13.1031548  PAK                   NA
```

```
## 22                   India 12.5360556  IND             2086.143
## 23                   Gabon 12.3352778  GAB                   NA
## 24                 Myanmar 11.9392222  MMR                   NA
## 25                Cameroon 11.9049444  CMR                   NA
## 26                   Kenya 11.3071111  KEN                   NA
## 27                Cambodia 11.2942222  KHM                   NA
## 28            Burkina Faso 10.8768889  BFA                   NA
## 29                 Liberia 10.6376111  LBR                   NA
## 30          Kyrgyz Republic 10.6180247  KGZ                   NA
## 31              Bangladesh 10.3033333  BGD                   NA
## 32                 Nigeria 10.1447778  NGA                   NA
## 33                 Comoros  9.5546111  COM                   NA
## 34                  Rwanda  9.1943889  RWA                   NA
## 35                 Lesotho  8.9674444  LSO                   NA
## 36              Madagascar  8.1401667  MDG                   NA
## 37                   Benin  7.8707778  BEN                   NA
## 38               Indonesia  7.5503571  IDN                   NA
## 39                    Togo  7.3982222  TGO                   NA
## 40                 Namibia  7.3674444  NAM                   NA
## 41                  Angola  7.0797778  AGO                   NA
## 42                   Ghana  7.0600000  GHA                   NA
## 43                   Nepal  6.4663889  NPL                   NA
## 44                Maldives  6.1069167  MDV                   NA
## 45                Eswatini  5.8334444  SWZ                   NA
## 46                  Jordan  5.6417778  JOR             1964.000
## 47                 Moldova  5.4929012  MDA             2161.000
## 48                 Bolivia  5.3954444  BOL             1888.000
## 49    Sao Tome and Principe  5.3686782  STP                   NA
## 50                Nicaragua  4.6332222  NIC                   NA
## 51                   Guyana  4.2798889  GUY                   NA
## 52                   Turkey  4.2592308  TUR             1940.667
## 53                  Armenia  4.0113095  ARM             2285.000
## 54                   Malawi  3.6356111  MWI                   NA
## 55              Philippines  3.5443333  PHL                   NA
## 56                    Haiti  3.5145000  HTI                   NA
## 57                Mozambique  3.5043333  MOZ                   NA
## 58                  Honduras  3.0452778  HND                   NA
## 59                   Albania  2.4524444  ALB             1971.000
## 60                 Guatemala  2.3038333  GTM                   NA
## 61              South Africa  1.6648851  ZAF                   NA
## 62                      Peru  1.2701111  PER             2158.500
## 63                  Colombia  1.1404444  COL             2001.875
## 64        Dominican Republic  0.7340476  DOM                   NA
##    Standard_Rating_n Standard_Rating_max Standard_Rating_min Rapid_rating_mean
## 1                 NA                  NA                  NA                NA
## 2                  1                1853                1853          1788.000
## 3                 NA                  NA                  NA                NA
## 4                 NA                  NA                  NA                NA
## 5                 NA                  NA                  NA                NA
## 6                 NA                  NA                  NA                NA
## 7                 NA                  NA                  NA                NA
## 8                 NA                  NA                  NA                NA
## 9                 NA                  NA                  NA                NA
## 10                NA                  NA                  NA                NA
```

```
## 11   NA   NA   NA   NA
## 12   NA   NA   NA   NA
## 13    8 2279 1871 1950.000
## 14   NA   NA   NA   NA
## 15   NA   NA   NA   NA
## 16   NA   NA   NA   NA
## 17    4 2182 1832 1895.250
## 18   NA   NA   NA   NA
## 19   NA   NA   NA   NA
## 20   NA   NA   NA   NA
## 21   NA   NA   NA   NA
## 22    7 2202 1888 1878.714
## 23   NA   NA   NA   NA
## 24   NA   NA   NA   NA
## 25   NA   NA   NA   NA
## 26   NA   NA   NA   NA
## 27   NA   NA   NA   NA
## 28   NA   NA   NA   NA
## 29   NA   NA   NA   NA
## 30   NA   NA   NA   NA
## 31   NA   NA   NA   NA
## 32   NA   NA   NA   NA
## 33   NA   NA   NA   NA
## 34   NA   NA   NA   NA
## 35   NA   NA   NA   NA
## 36   NA   NA   NA   NA
## 37   NA   NA   NA   NA
## 38   NA   NA   NA   NA
## 39   NA   NA   NA   NA
## 40   NA   NA   NA   NA
## 41   NA   NA   NA   NA
## 42   NA   NA   NA   NA
## 43   NA   NA   NA   NA
## 44   NA   NA   NA   NA
## 45   NA   NA   NA   NA
## 46    1 1964 1964 1941.000
## 47    1 2161 2161 2121.000
## 48    1 1888 1888 1931.000
## 49   NA   NA   NA   NA
## 50   NA   NA   NA   NA
## 51   NA   NA   NA   NA
## 52    3 2033 1839 1931.333
## 53    1 2285 2285 2282.000
## 54   NA   NA   NA   NA
## 55   NA   NA   NA   NA
## 56   NA   NA   NA   NA
## 57   NA   NA   NA   NA
## 58   NA   NA   NA   NA
## 59    1 1971 1971 1788.000
## 60   NA   NA   NA   NA
## 61   NA   NA   NA   NA
## 62    2 2244 2073 2126.500
## 63    8 2257 1817 2031.000
## 64   NA   NA   NA   NA
```

```
##    Rapid_rating_n Rapid_rating_max Rapid_rating_min Blitz_rating_mean
## 1              NA               NA               NA                NA
## 2               1             1788             1788          1769.000
## 3              NA               NA               NA                NA
## 4              NA               NA               NA                NA
## 5              NA               NA               NA                NA
## 6              NA               NA               NA                NA
## 7              NA               NA               NA                NA
## 8              NA               NA               NA                NA
## 9              NA               NA               NA                NA
## 10             NA               NA               NA                NA
## 11             NA               NA               NA                NA
## 12             NA               NA               NA                NA
## 13              8             2144             1623          1953.750
## 14             NA               NA               NA                NA
## 15             NA               NA               NA                NA
## 16             NA               NA               NA                NA
## 17              4             2139             1664          1883.000
## 18             NA               NA               NA                NA
## 19             NA               NA               NA                NA
## 20             NA               NA               NA                NA
## 21             NA               NA               NA                NA
## 22              7             2075             1569          1907.286
## 23             NA               NA               NA                NA
## 24             NA               NA               NA                NA
## 25             NA               NA               NA                NA
## 26             NA               NA               NA                NA
## 27             NA               NA               NA                NA
## 28             NA               NA               NA                NA
## 29             NA               NA               NA                NA
## 30             NA               NA               NA                NA
## 31             NA               NA               NA                NA
## 32             NA               NA               NA                NA
## 33             NA               NA               NA                NA
## 34             NA               NA               NA                NA
## 35             NA               NA               NA                NA
## 36             NA               NA               NA                NA
## 37             NA               NA               NA                NA
## 38             NA               NA               NA                NA
## 39             NA               NA               NA                NA
## 40             NA               NA               NA                NA
## 41             NA               NA               NA                NA
## 42             NA               NA               NA                NA
## 43             NA               NA               NA                NA
## 44             NA               NA               NA                NA
## 45             NA               NA               NA                NA
## 46              1             1941             1941          1857.000
## 47              1             2121             2121          2123.000
## 48              1             1931             1931          2020.000
## 49             NA               NA               NA                NA
## 50             NA               NA               NA                NA
## 51             NA               NA               NA                NA
## 52              3             2016             1802          1987.333
## 53              1             2282             2282          2275.000
```

```
## 54              NA               NA               NA               NA
## 55              NA               NA               NA               NA
## 56              NA               NA               NA               NA
## 57              NA               NA               NA               NA
## 58              NA               NA               NA               NA
## 59               1             1788             1788          1886.000
## 60              NA               NA               NA               NA
## 61              NA               NA               NA               NA
## 62               2             2204             2049          2108.000
## 63               8             2312             1784          2003.250
## 64              NA               NA               NA               NA
##    Blitz_rating_n Blitz_rating_max Blitz_rating_min
## 1              NA               NA               NA
## 2               1             1769             1769
## 3              NA               NA               NA
## 4              NA               NA               NA
## 5              NA               NA               NA
## 6              NA               NA               NA
## 7              NA               NA               NA
## 8              NA               NA               NA
## 9              NA               NA               NA
## 10             NA               NA               NA
## 11             NA               NA               NA
## 12             NA               NA               NA
## 13              8             2109             1656
## 14             NA               NA               NA
## 15             NA               NA               NA
## 16             NA               NA               NA
## 17              4             2072             1698
## 18             NA               NA               NA
## 19             NA               NA               NA
## 20             NA               NA               NA
## 21             NA               NA               NA
## 22              7             2076             1619
## 23             NA               NA               NA
## 24             NA               NA               NA
## 25             NA               NA               NA
## 26             NA               NA               NA
## 27             NA               NA               NA
## 28             NA               NA               NA
## 29             NA               NA               NA
## 30             NA               NA               NA
## 31             NA               NA               NA
## 32             NA               NA               NA
## 33             NA               NA               NA
## 34             NA               NA               NA
## 35             NA               NA               NA
## 36             NA               NA               NA
## 37             NA               NA               NA
## 38             NA               NA               NA
## 39             NA               NA               NA
## 40             NA               NA               NA
## 41             NA               NA               NA
## 42             NA               NA               NA
```

```
## 43              NA                NA                NA
## 44              NA                NA                NA
## 45              NA                NA                NA
## 46               1              1857              1857
## 47               1              2123              2123
## 48               1              2020              2020
## 49              NA                NA                NA
## 50              NA                NA                NA
## 51              NA                NA                NA
## 52               3              2026              1933
## 53               1              2275              2275
## 54              NA                NA                NA
## 55              NA                NA                NA
## 56              NA                NA                NA
## 57              NA                NA                NA
## 58              NA                NA                NA
## 59               1              1886              1886
## 60              NA                NA                NA
## 61              NA                NA                NA
## 62               2              2168              2048
## 63               8              2240              1791
## 64              NA                NA                NA
```

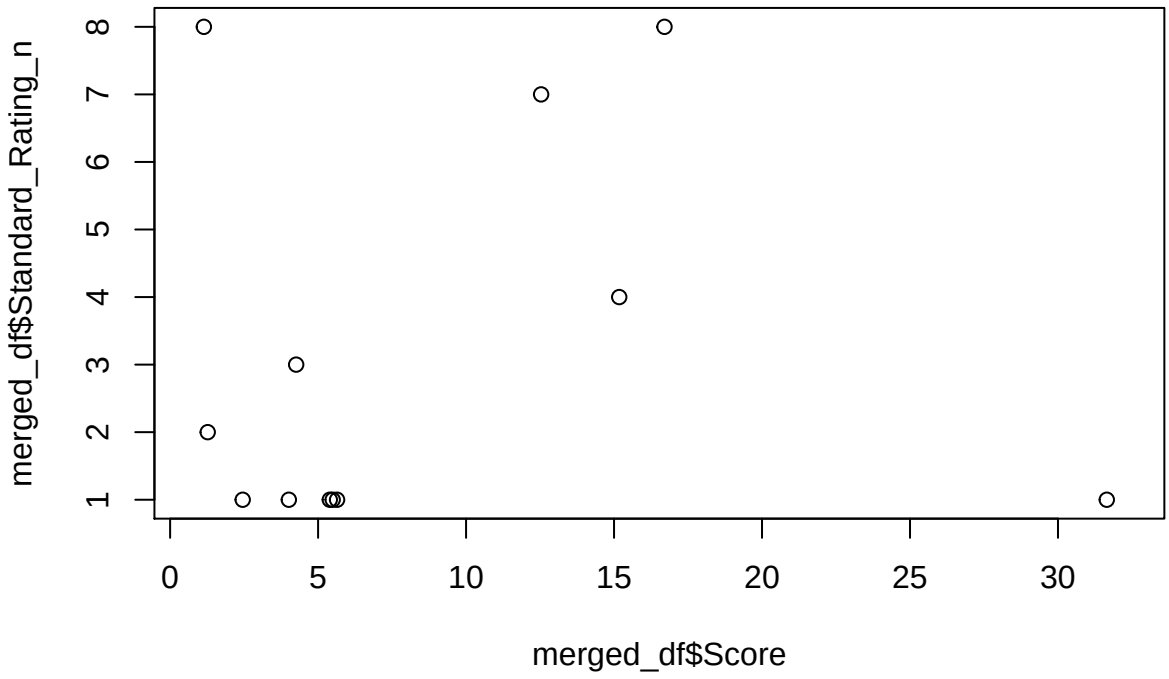
```
head(merged_df)
```

```
##       Country     Score code Standard_Rating_mean Standard_Rating_n
## 1 Afghanistan 22.528810  AFG                   NA                NA
## 2     Albania  2.452444  ALB             1971.000                 1
## 3      Angola  7.079778  AGO                   NA                NA
## 4     Armenia  4.011310  ARM             2285.000                 1
## 5  Azerbaijan 16.704583  AZE             2103.375                 8
## 6  Bangladesh 10.303333  BGD                   NA                NA
##   Standard_Rating_max Standard_Rating_min Rapid_rating_mean Rapid_rating_n
## 1                  NA                  NA                NA             NA
## 2                1971                1971              1788              1
## 3                  NA                  NA                NA             NA
## 4                2285                2285              2282              1
## 5                2279                1871              1950              8
## 6                  NA                  NA                NA             NA
##   Rapid_rating_max Rapid_rating_min Blitz_rating_mean Blitz_rating_n
## 1               NA               NA                NA             NA
## 2             1788             1788           1886.00              1
## 3               NA               NA                NA             NA
## 4             2282             2282           2275.00              1
## 5             2144             1623           1953.75              8
## 6               NA               NA                NA             NA
##   Blitz_rating_max Blitz_rating_min
## 1               NA               NA
## 2             1886             1886
## 3               NA               NA
## 4             2275             2275
## 5             2109             1656
## 6               NA               NA
```

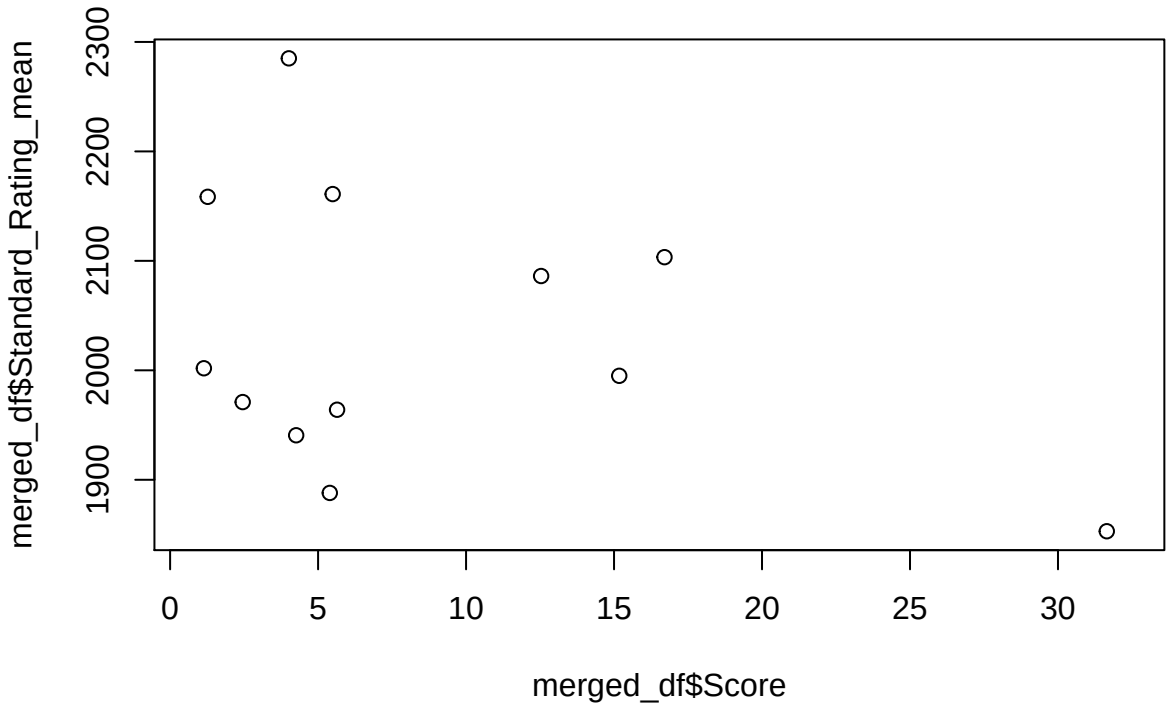
The generated output is very small due to a very small intersection between interviewed countries and top women chess players, but we will still attempt to see correlation between the violence index and the rating, and number, of players.

But first, a little data visualization.

```
plot(merged_df$Score, merged_df$Standard_Rating_n)
```




```
plot(merged_df$Score, merged_df$Standard_Rating_mean)
```



Now, let's see the correlation values.

```
merged_df_no_na <- na.omit(merged_df)
print(cor(merged_df_no_na$Score, merged_df_no_na$Standard_Rating_n))
```

```
## [1] 0.09331977
```

```
print(cor(merged_df_no_na$Score, merged_df_no_na$Standard_Rating_mean))
```

```
## [1] -0.3560483
```

```
print(cor(merged_df_no_na$Score, merged_df_no_na$Rapid_rating_mean))
```

```
## [1] -0.507026
```

```
print(cor(merged_df_no_na$Score, merged_df_no_na$Blitz_rating_mean))
```

```
## [1] -0.6013585
```

From the previous result we see two interesting observations after peaking into the aggregated data:

- 1) Because the tiny size of the sample, the correlation between the violence score and the number of players is meaningless.

- 2) there is a moderate negative correlation between the violence score and the rating of players on all three categories, that means that as the violence index increases, there is an apparent negative impact into how well the players of that country perform.

Now we will create a dataset out of our aggregated data.

```
write.csv(merged_df,"data/violence_chess_ds.csv", row.names = TRUE)
write.csv(chess_data_slim_grouped,"data/chess_aggregate_ds.csv", row.names = TRUE)
```

Before wrapping up, let's create a data report of our dataset

```
makeDataReport(merged_df,
               render = FALSE,
               file = "codebook.Rmd",
               codebook = TRUE,
               replace = TRUE,
               reportTitle = "Violence index and women chess players across the world")
```