Analysis of Tennis Rankings

William Cheung

December 9, 2015

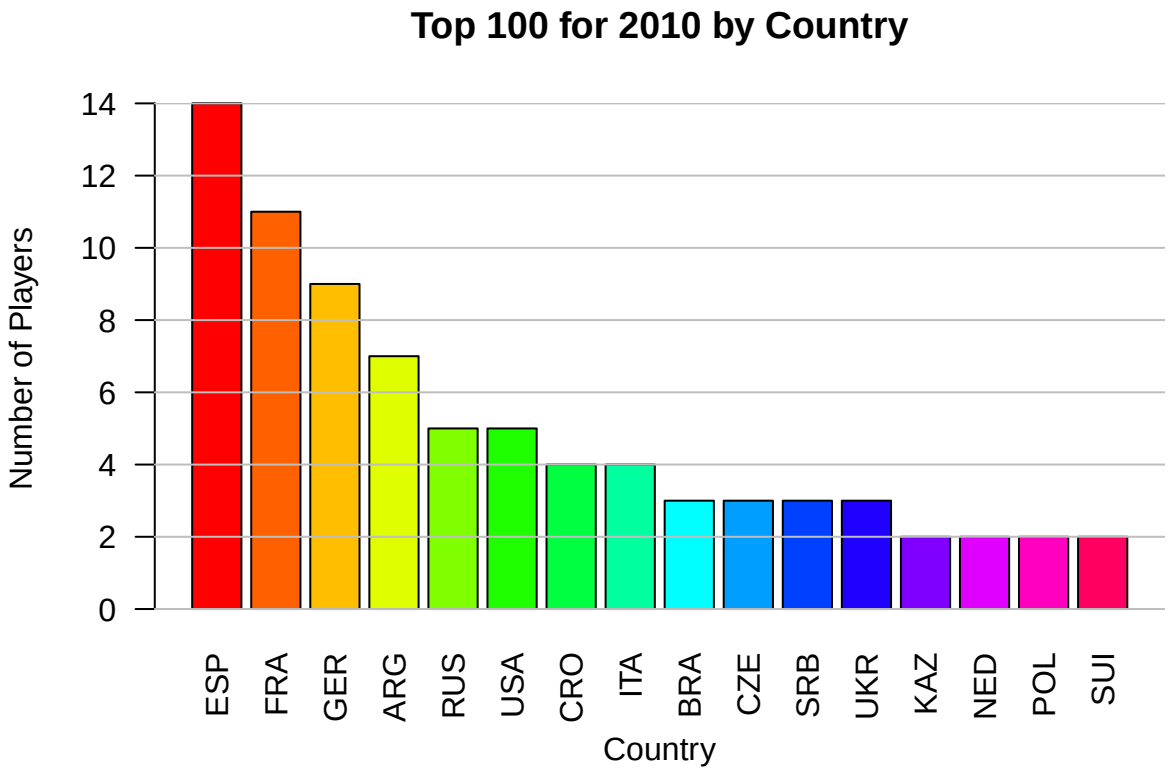
Use Rankings to determine depth of countries and players

```
library("readr")

#getwd()
setwd("/Users/williamcheung/Desktop/stats133/Tennis/")
setwd("./code")

# read in the files
rankings_10 <- read.csv("../clean_data/2010_top100.csv", header = TRUE)
rankings_11 <- read.csv("../clean_data/2011_top100.csv", header = TRUE)
rankings_12 <- read.csv("../clean_data/2012_top100.csv", header = TRUE)
rankings_13 <- read.csv("../clean_data/2013_top100.csv", header = TRUE)
rankings_14 <- read.csv("../clean_data/2014_top100.csv", header = TRUE)
rankings_15 <- read.csv("../clean_data/2015_top100.csv", header = TRUE)

# Analyze rankings for 2010, looking for country's depth of players
# define depth as country having more than 1 player in the top 100
# Get sorted frequencies of countries with more than 1 player in the top 100 for 2010
country_freqs <- sort(table(rankings_10$country), decreasing = TRUE)
depth_freqs <- as.data.frame(country_freqs)
depth_freqs <- as.data.frame(depth_freqs[!(depth_freqs$country_freqs == 1),])
colnames(depth_freqs) <- "Num_Players"

# function to find top50
top50 <- function(vect) {
  if (vect <= 50) return (TRUE) else return (FALSE)
}
top50 = Vectorize(top50)

# function to find top20
top20 <- function(vect) {
  if (vect <= 20) return (TRUE) else return (FALSE)
}
top20 = Vectorize(top20)

# function to find top10
top10 <- function(vect) {
  if (vect <= 10) return (TRUE) else return (FALSE)
}
top10 = Vectorize(top10)
```

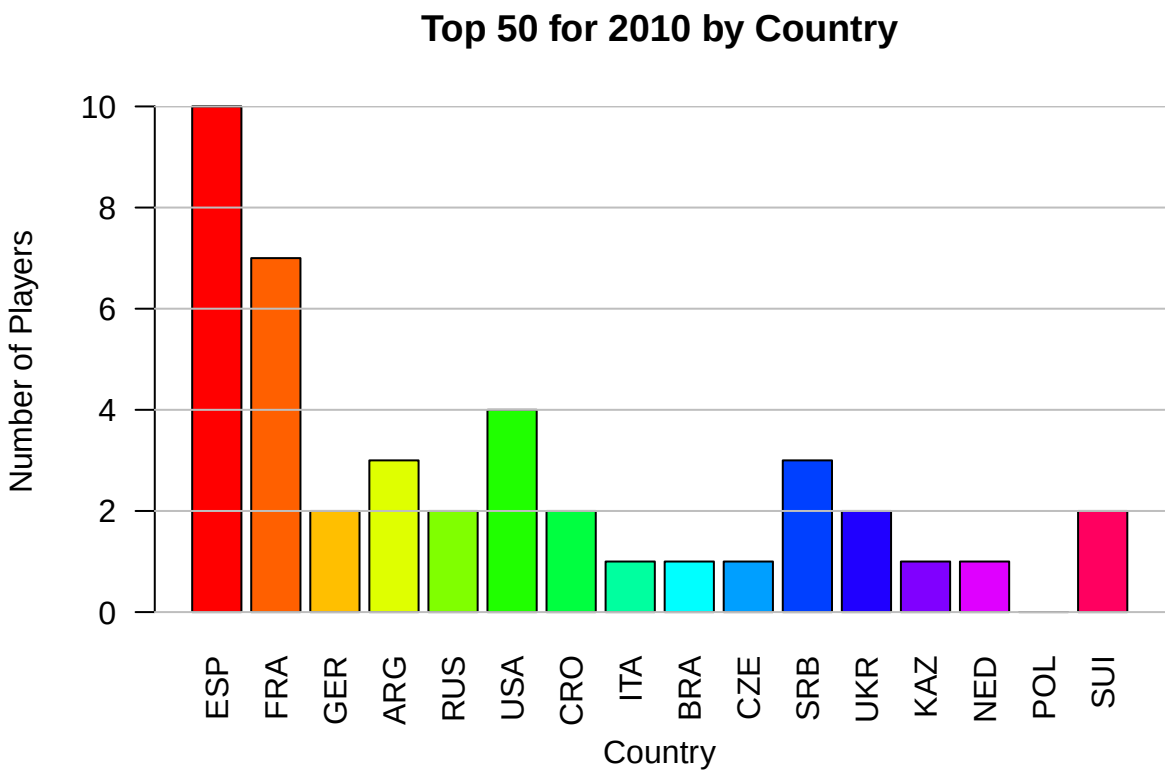
We will now break down the countries with most players in the top 100, then categorize them by counting the number of players in the top 50, top 20, and top 10.

```
library(reshape2)
library(ggplot2)

country_vec <- c()
list_top100 <- list()
for (i in 1:nrow(depth_freqs)) {
  country_vec <- c(country_vec, rownames(depth_freqs)[i])
  index <- which(rankings_10$country == rownames(depth_freqs)[i])
  temp <- c(rankings_10$rank[index])
  list_top100[[i]] <- temp
}

list_top50 <- list()
for (i in 1:nrow(depth_freqs)) {
  list_top50[[i]] <- list_top100[[i]][top50(list_top100[[i]])]
}

list_top20 <- list()
for (i in 1:nrow(depth_freqs)) {
  list_top20[[i]] <- list_top100[[i]][top20(list_top100[[i]])]
}

list_top10 <- list()
for (i in 1:nrow(depth_freqs)) {
  list_top10[[i]] <- list_top100[[i]][top10(list_top100[[i]])]
}

# create list of countries and how many of its players are in top100, 50, 20, 10
prepare <- function(input_freqs) {
  country_stats <- list()
  for (i in 1:nrow(input_freqs)) {
    country_stats[[i]] <- c(length(list_top100[[i]]), length(list_top50[[i]]),
                            length(list_top20[[i]]), length(list_top10[[i]]))
  }
  names(country_stats) = country_vec
  return (country_stats)
}

list_of_freqs <- prepare(depth_freqs)
countries <- names(list_of_freqs)

all_100 <- c()
all_50 <- c()
all_20 <- c()
all_10 <- c()
for (country in 1:length(list_of_freqs)) {
  all_100 <- c(all_100, list_of_freqs[[country]][1])
  all_50 <- c(all_50, list_of_freqs[[country]][2])
  all_20 <- c(all_20, list_of_freqs[[country]][3])
  all_10 <- c(all_10, list_of_freqs[[country]][4])
}
```

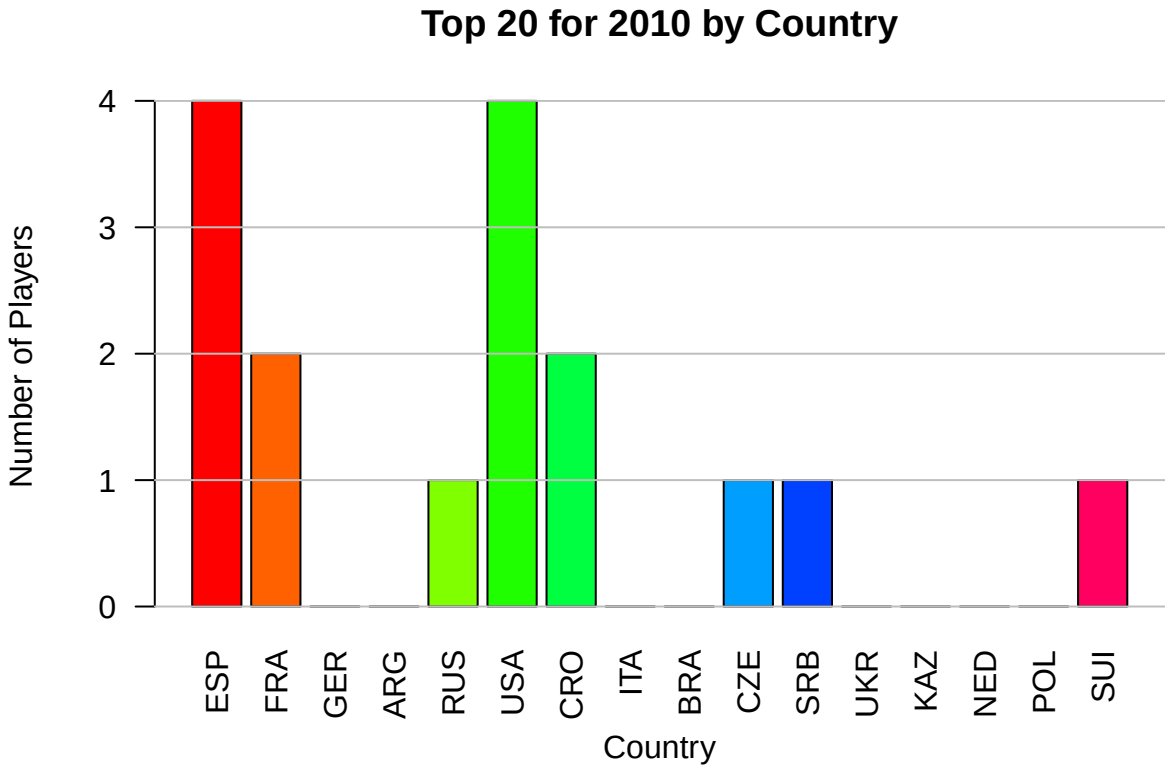
```
final <- data.frame(top100 = all_100, top50 = all_50, top20 = all_20, top10 = all_10)
rownames(final) <- countries


rainbow_16 <-rainbow(n=16, s = 1, v = 1, start = 0, end = max(1, 16 - 1)/16, alpha = 1)
top100_bar <- barplot(final$top100, names.arg = rownames(final), ylab = "Number of Players",
        xlab = "Country", las = 2, col = rainbow_16, main = "Top 100 for 2010 by Country")
abline(h = seq(from = 0, to = 14, by = 2), col = ' gray')
```



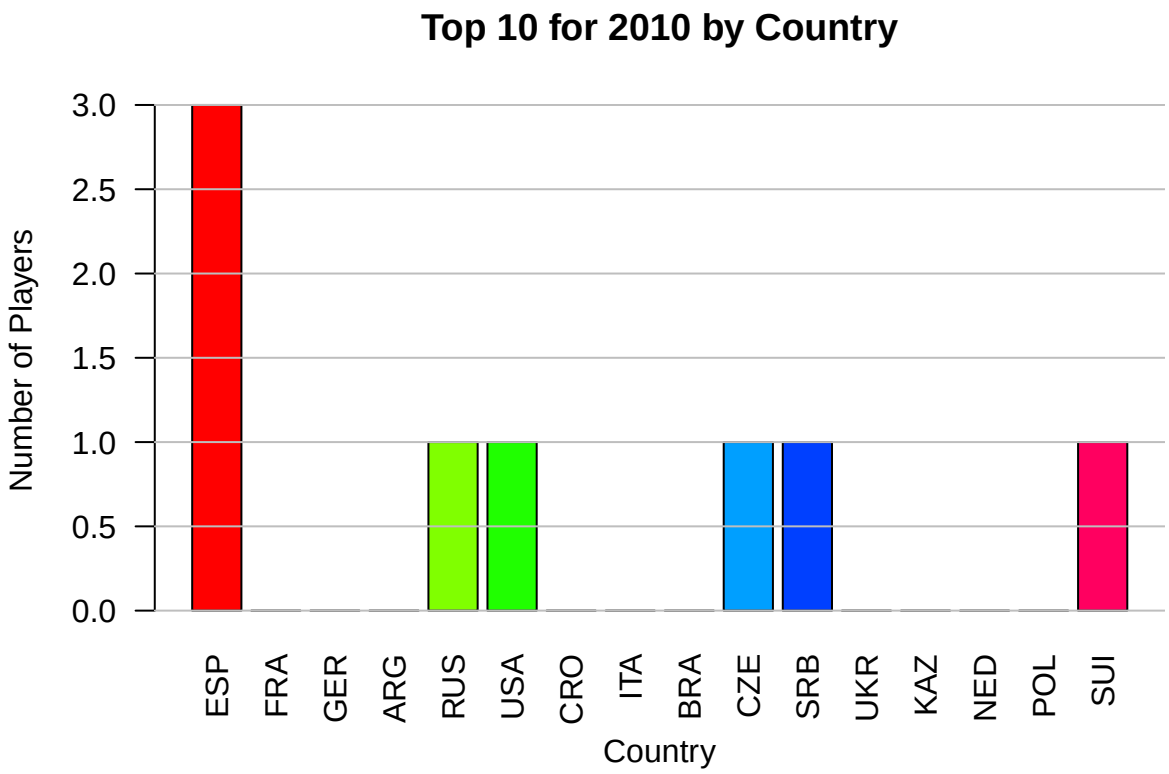

```
top50_bar <- barplot(final$top50, names.arg = rownames(final), ylab = "Number of Players",
        xlab = "Country", las = 2, col = rainbow_16, main = "Top 50 for 2010 by Country")
abline(h = seq(from = 0, to = 14, by = 2), col = ' gray')
```




```
top20_bar <- barplot(final$top20, names.arg = rownames(final), ylab = "Number of Players",
        xlab = "Country", las = 2, col = rainbow_16, main = "Top 20 for 2010 by Country")
abline(h = seq(from = 0, to = 14, by = 1), col = ' gray')
```



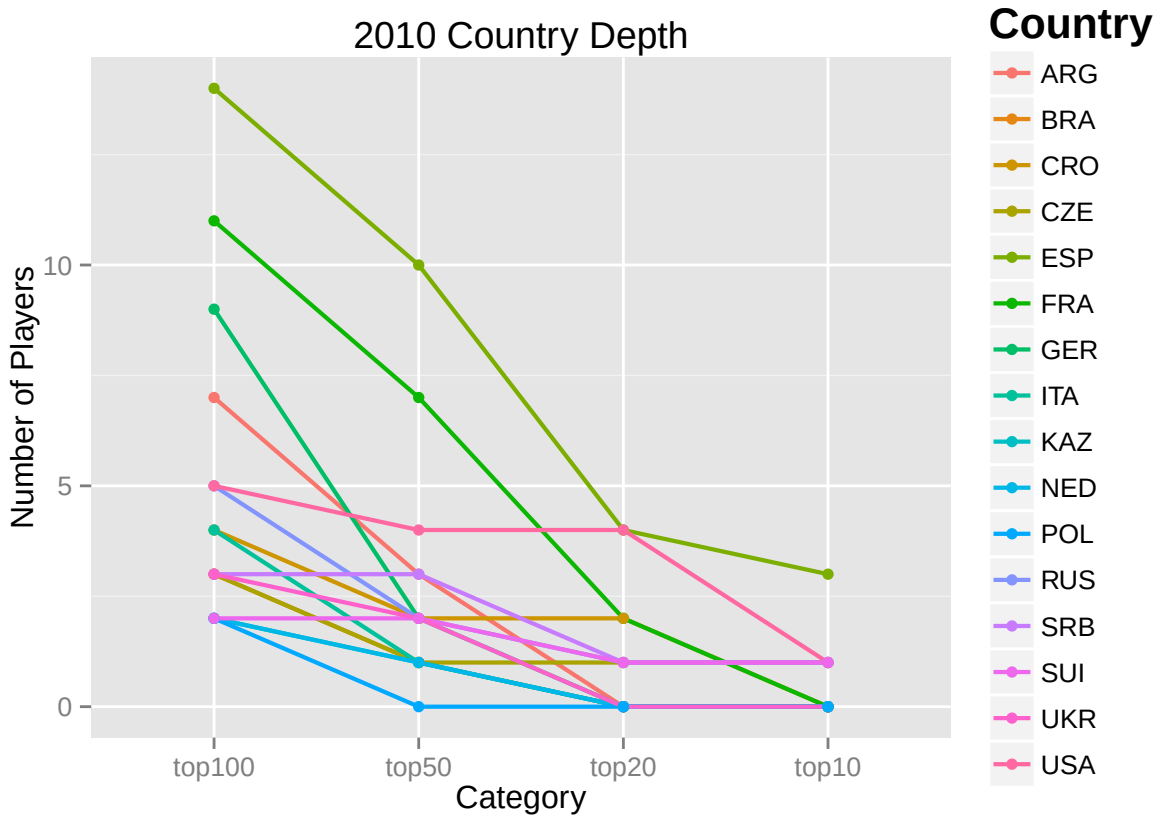
```
top10_bar <- barplot(final$top10, names.arg = rownames(final), ylab = "Number of Players",
        xlab = "Country", las = 2, col = rainbow_16, main = "Top 10 for 2010 by Country")
abline(h = seq(from = 0, to = 14, by = 0.5), col = ' gray')
```




```
final_country_names <- rownames(final)
final_before_reshape <- final

final <- melt(final)
```

```
## No id variables; using all as measure variables
```

```
final$country <- final_country_names

ggplot(final, aes(variable, value, group=factor(country), color = factor(country))) + geom_line(size=.75
```




```
sum_top100 <- sum(final_before_reshape$top100)
percentage_100 = c()
final_before_reshape[,1]
```

```
##  [1] 14 11  9  7  5  5  4  4  3  3  3  3  2  2  2  2
```

```
for (i in 1:length(rownames(final_before_reshape))) {
  percentage_100 = c(percentage_100, final_before_reshape[,1][i] / sum_top100)
}

sum_top50 <- sum(final_before_reshape$top50)
percentage_50 = c()
final_before_reshape[,2]
```

```
##  [1] 10  7  2  3  2  4  2  1  1  1  3  2  1  1  0  2
```

```
for (i in 1:length(rownames(final_before_reshape))) {
  percentage_50 = c(percentage_50, final_before_reshape[,2][i] / sum_top50)
}

sum_top20 <- sum(final_before_reshape$top20)
percentage_20 = c()
final_before_reshape[,3]
```

```
##  [1] 4 2 0 0 1 4 2 0 0 1 1 0 0 0 0 1
```

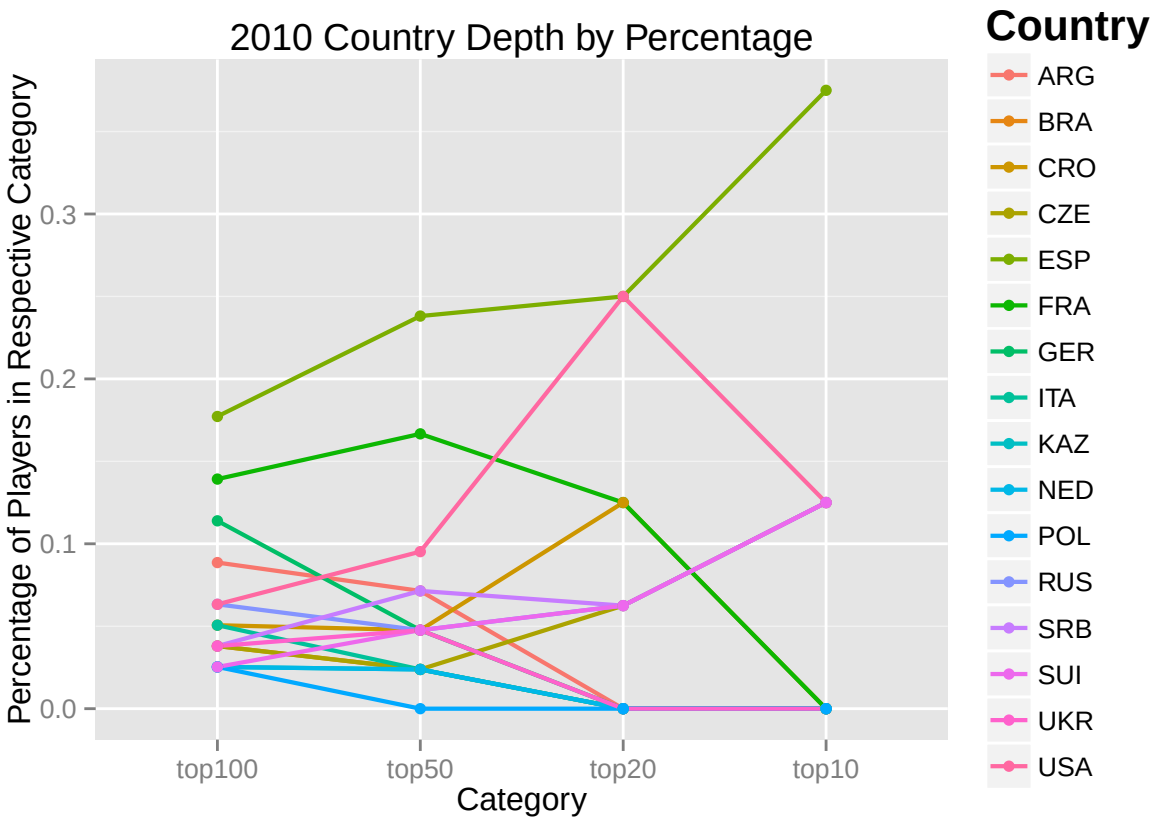
```
for (i in 1:length(rownames(final_before_reshape))) {
  percentage_20 = c(percentage_20, final_before_reshape[,3][i] / sum_top20)
}

sum_top10 <- sum(final_before_reshape$top10)
percentage_10 = c()
final_before_reshape[,4]
```

```
##  [1] 3 0 0 0 1 1 0 0 0 1 1 0 0 0 0 1
```

```
for (i in 1:length(rownames(final_before_reshape))) {
  percentage_10 = c(percentage_10, final_before_reshape[,4][i] / sum_top10)
}


percentage_final <- final_before_reshape
percentage_final$top100 <- percentage_100
percentage_final$top50 <- percentage_50
percentage_final$top20 <- percentage_20
percentage_final$top10 <- percentage_10


final_country_names <- rownames(final_before_reshape)
p_final_before_reshape <- percentage_final
percentage_final <- melt(percentage_final)
```

```
## No id variables; using all as measure variables
```

```
percentage_final$country <- final_country_names

ggplot(percentage_final, aes(variable, value, group=factor(country), color = factor(country))) + geom_li
```


2010 Country Depth by Percentage
Percentage of Players in Respective Category
0.0
0.1
0.2
0.3
top100
top50
top20
top10
Category
Country
ARG
BRA
CRO
CZE
ESP
FRA
GER
ITA
KAZ
NED
POL
RUS
SRB
SUI
UKR
USA

Evolution of rankings- a year on year change of how many players in each category: show most consistent countries, based off the starting point of year 2010.

```
#now we need at least 5 players in the top100
country_freqs_10 <- sort(table(rankings_10$country), decreasing = TRUE)
depth_freqs_10 <- as.data.frame(country_freqs_10)
depth_freqs_10 <- as.data.frame(depth_freqs_10[(depth_freqs_10$country_freqs_10 >= 5),])
colnames(depth_freqs_10) <- "2010"

country_freqs_11 <- sort(table(rankings_11$country), decreasing = TRUE)
depth_freqs_11 <- as.data.frame(country_freqs_11)
depth_freqs_11 <- as.data.frame(depth_freqs_11[(depth_freqs_11$country_freqs_11 >= 5),])
colnames(depth_freqs_11) <- "2011"

country_freqs_12 <- sort(table(rankings_12$country), decreasing = TRUE)
depth_freqs_12 <- as.data.frame(country_freqs_12)
depth_freqs_12 <- as.data.frame(depth_freqs_12[(depth_freqs_12$country_freqs_12 >= 5),])
colnames(depth_freqs_12) <- "2012"

country_freqs_13 <- sort(table(rankings_13$country), decreasing = TRUE)
depth_freqs_13 <- as.data.frame(country_freqs_13)
depth_freqs_13 <- as.data.frame(depth_freqs_13[(depth_freqs_13$country_freqs_13 >= 5),])
colnames(depth_freqs_13) <- "2013"

country_freqs_14 <- sort(table(rankings_14$country), decreasing = TRUE)
depth_freqs_14 <- as.data.frame(country_freqs_14)
depth_freqs_14 <- as.data.frame(depth_freqs_14[(depth_freqs_14$country_freqs_14 >= 5),])
colnames(depth_freqs_14) <- "2014"

country_freqs_15 <- sort(table(rankings_15$country), decreasing = TRUE)
depth_freqs_15 <- as.data.frame(country_freqs_15)
depth_freqs_15 <- as.data.frame(depth_freqs_15[(depth_freqs_15$country_freqs_15 >= 5),])
colnames(depth_freqs_15) <- "2015"

# created all_merged data frame

all_merged <- merge(depth_freqs_10, depth_freqs_11, by = 0, all.x=TRUE)
row.names(all_merged) <- all_merged$Row.names
all_merged$Row.names <- NULL

all_merged <- merge(all_merged, depth_freqs_12, by = 0, all.x = TRUE)
row.names(all_merged) <- all_merged$Row.names
all_merged$Row.names <- NULL

all_merged <- merge(all_merged, depth_freqs_13, by = 0, all.x = TRUE)
row.names(all_merged) <- all_merged$Row.names
all_merged$Row.names <- NULL

all_merged <- merge(all_merged, depth_freqs_14, by = 0, all.x = TRUE)
row.names(all_merged) <- all_merged$Row.names
all_merged$Row.names <- NULL

all_merged <- merge(all_merged, depth_freqs_15, by = 0, all.x = TRUE)
row.names(all_merged) <- all_merged$Row.names
```

```
all_merged$Row.names <- NULL

# merged_before_reshape is the data.frame before a reshape
merged_before_reshape <- all_merged

country_names <- rownames(all_merged)
country_names
```

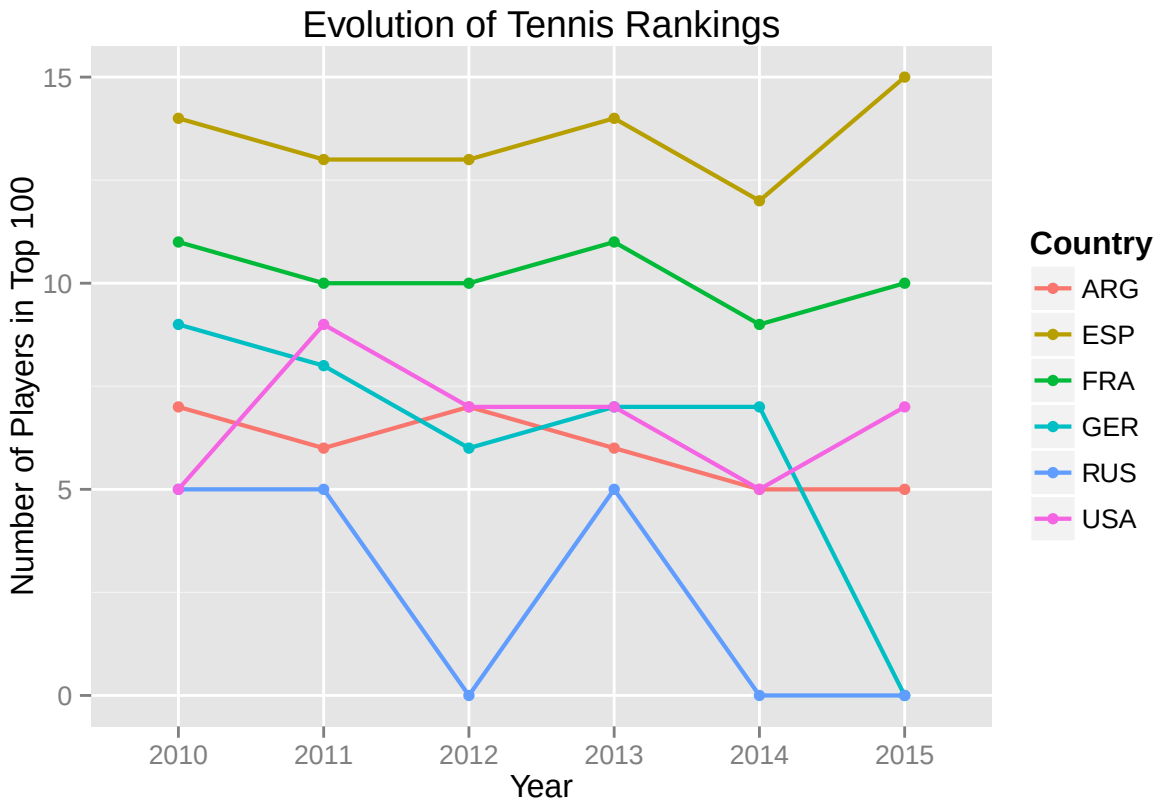
```
## [1] "ARG" "ESP" "FRA" "GER" "RUS" "USA"
```

```
library(reshape2)
all_merged <- melt(all_merged)
```

```
## No id variables; using all as measure variables
```

```
## Warning: attributes are not identical across measure variables; they will
## be dropped
```

```
all_merged[is.na(all_merged)] <- 0
all_merged$country <- country_names

# Got the graph!
ggplot(all_merged, aes(variable, value, group=factor(country), color = factor(country))) + geom_line(si
```



For every country add the points on. Average number of points per country. We want to find out how good the average player is. How much a player contributes on average

```
# Average number of points per any country
# this is the sum of all points / number of unique countries = averae points for a country
# number of points of the country / # of players in that country = how good that player is

# start from rankings_15
# get sum of all the points
sum_2015_points <- sum(rankings_15$points)
distinct_countries <- unique(rankings_15$country)
avg_points_per_country <- sum_2015_points / length(distinct_countries)

# make data frame with points and country
country_points <- data.frame(rankings_15$points, rankings_15$country)
names(country_points) <- c("points", "country")
nrow(country_points[unique(rankings_15$country),])
```

```
## [1] 39
```

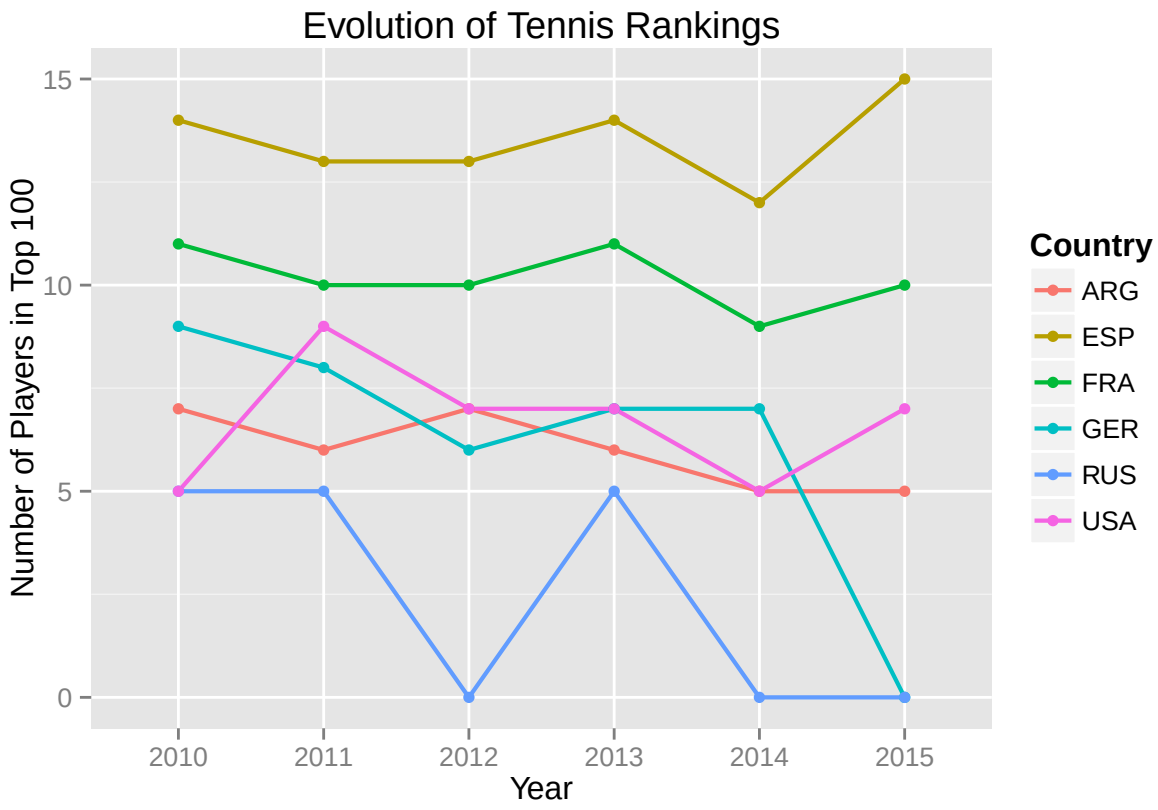
```
# create list of country and points
list_points <- c()
for (i in 1:length(distinct_countries)) {
  index1 <- which(country_points$country == distinct_countries[i])
  temp1 <- c(rankings_15$points[index1])
  list_points[[i]] <- temp1
}
names(list_points) <- distinct_countries

# get the average points for top100 players
avg_points_per_player <- sum_2015_points / nrow(rankings_15)
avg_points_per_player
```

```
## [1] 1546.68
```

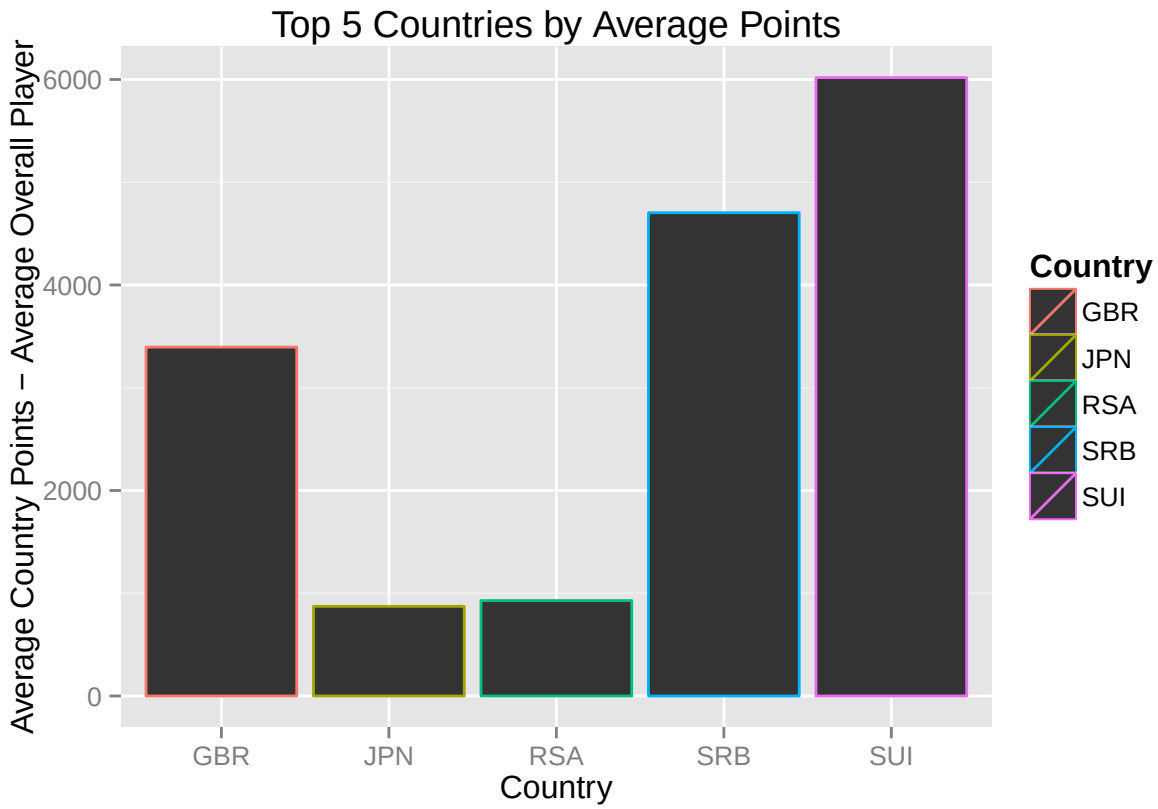
```
# get avergae points for a country's players
avg_player_for_country = list()
for (i in 1:length(list_points)) {
  avg_player_for_country[[i]] <- sum((list_points[[i]]) / length(list_points[[i]]))
}
names(avg_player_for_country) <- distinct_countries

# compare average points for all players vs the country's average player points
list_difference_country_vs_avg <- list()
for (i in 1:length(avg_player_for_country)) {
  list_difference_country_vs_avg[[i]] <- avg_player_for_country[[i]] - avg_points_per_player
}
names(list_difference_country_vs_avg) <- distinct_countries

avg_vs_country_2015 <- data.frame(matrix(unlist(list_difference_country_vs_avg),
                                         nrow=length(list_difference_country_vs_avg), byrow=T),
                                         row.names = distinct_countries)
names(avg_vs_country_2015) <- "difference"
avg_vs_country_2015
```

```
##     difference
## SRB  4703.3200
## GBR  3395.3200
## SUI  6018.3200
## ESP  -103.3467
## CZE   611.3200
## JPN   872.3200
## FRA   -54.7800
## USA  -370.3943
## RSA   928.3200
## CRO  -179.9300
## CAN    75.8200
## BEL  -287.6800
## AUS  -549.8800
## AUT  -359.1800
## ITA  -545.0800
## BUL  -186.6800
## POR  -355.6800
## GER  -739.0133
## ARG  -728.2800
## UKR  -706.6800
## BRA  -441.6800
## LUX  -441.6800
## URU  -481.6800
## SVK  -566.6800
## CYP  -613.6800
## RUS  -833.0133
## KOR  -729.6800
## DOM  -749.6800
## POL  -751.6800
## UZB  -765.6800
## KAZ  -784.6800
## NED  -883.1800
## COL  -830.6800
## TPE  -872.6800
## LAT  -891.6800
## BIH  -897.6800
## LTU  -905.6800
## IND  -936.6800
## ISR  -968.6800
```

```
country_vs_avg_2015_sorted <- avg_vs_country_2015[order(
                              avg_vs_country_2015$difference,
                                                  decreasing = TRUE),
                                    , drop = FALSE]

top_5_countries_by_avg_points <- head(country_vs_avg_2015_sorted, n = 5)
bottom_5_countries_by_avg_points <- tail(country_vs_avg_2015_sorted, n = 5)

ggplot(all_merged, aes(variable, value, group=factor(country), color = factor(country))) + geom_line(si
```




```
t <- ggplot(top_5_countries_by_avg_points, aes(rownames(top_5_countries_by_avg_points), difference, grou
t + geom_bar(stat = "identity") + xlab("Country") + ylab("Average Country Points - Average Overall Playe
```



```
bottom_5_countries_by_avg_points
```

```
##     difference
## LAT    -891.68
## BIH    -897.68
## LTU    -905.68
## IND    -936.68
## ISR    -968.68
```

```
# b <- ggplot(bottom_5_countries_by_avg_points, aes(rownames(bottom_5_countries_by_avg_points), differe
# b + geom_bar(stat = "identity", position = "identity") + xlab("Country") + ylab("Average Country Poin
```