

Analysis of Tennis Rankings

William Cheung

December 9, 2015

Use Rankings to determine depth of countries and players

```
library("readr")

#getwd()

#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)
}

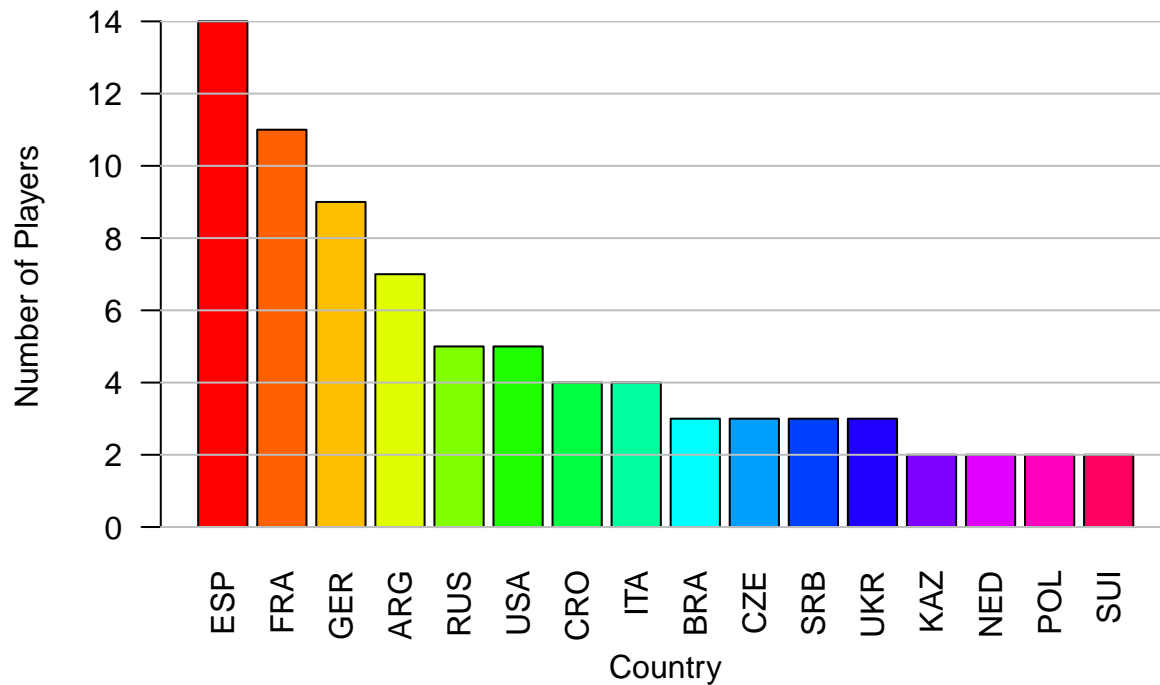
list_of_freqs <- prepare(depth_freqs)
countries <- names(list_of_freqs)

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

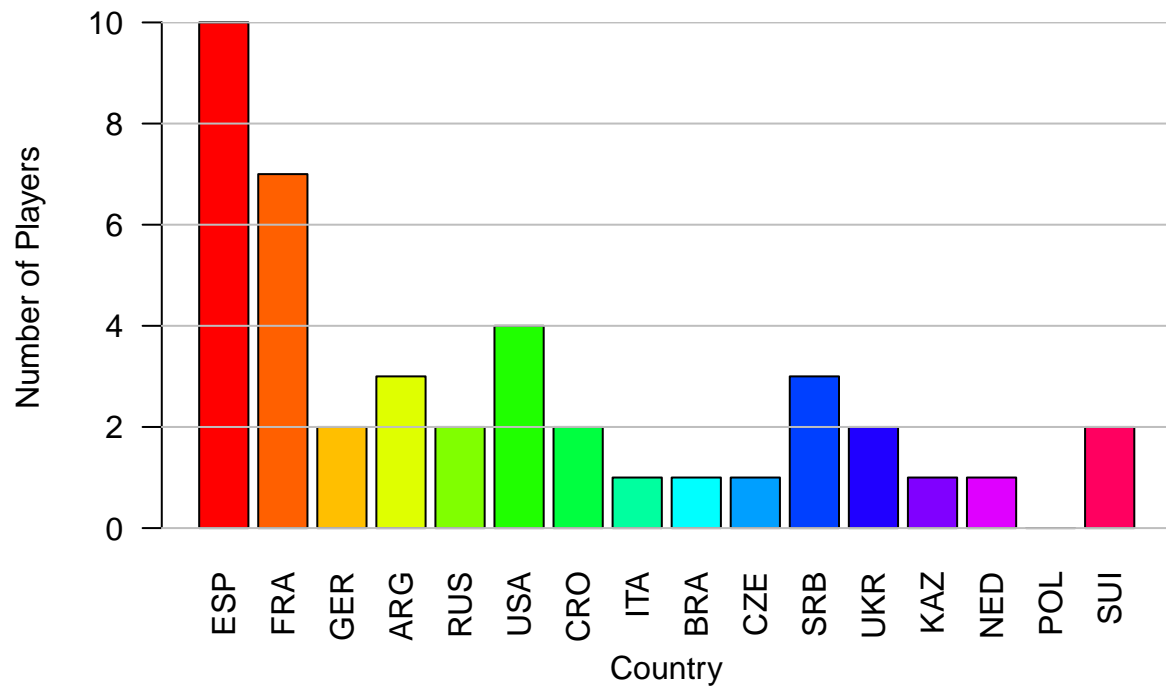
rainbow_16 <- rainbow(n=16, s = 1, v = 1, start = 0, end = max(1, 16 - 1)/16, alpha = 1)
top100_bar <- barplot(final$top100, names.arg = rownames(final), ylab = "Number of Players",
  xlab = "Country", las = 2, col = rainbow_16, main = "Top 100 for 2010 by Country")
abline(h = seq(from = 0, to = 14,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

Top 100 for 2010 by Country



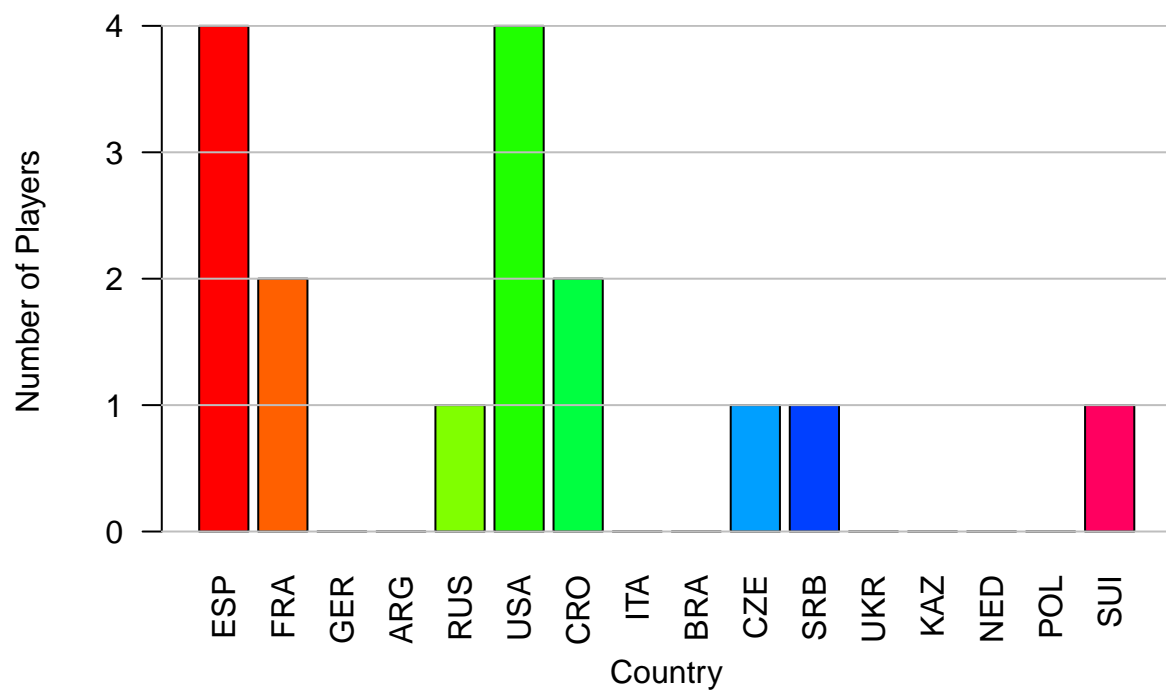
```
top50_bar <- barplot(final$top50, names.arg = rownames(final), ylab = "Number of Players",
  xlab = "Country", las = 2, col = rainbow_16, main = "Top 50 for 2010 by Country")
abline(h = seq(from = 0, to = 14, by = 2), col = 'gray')
```

Top 50 for 2010 by Country



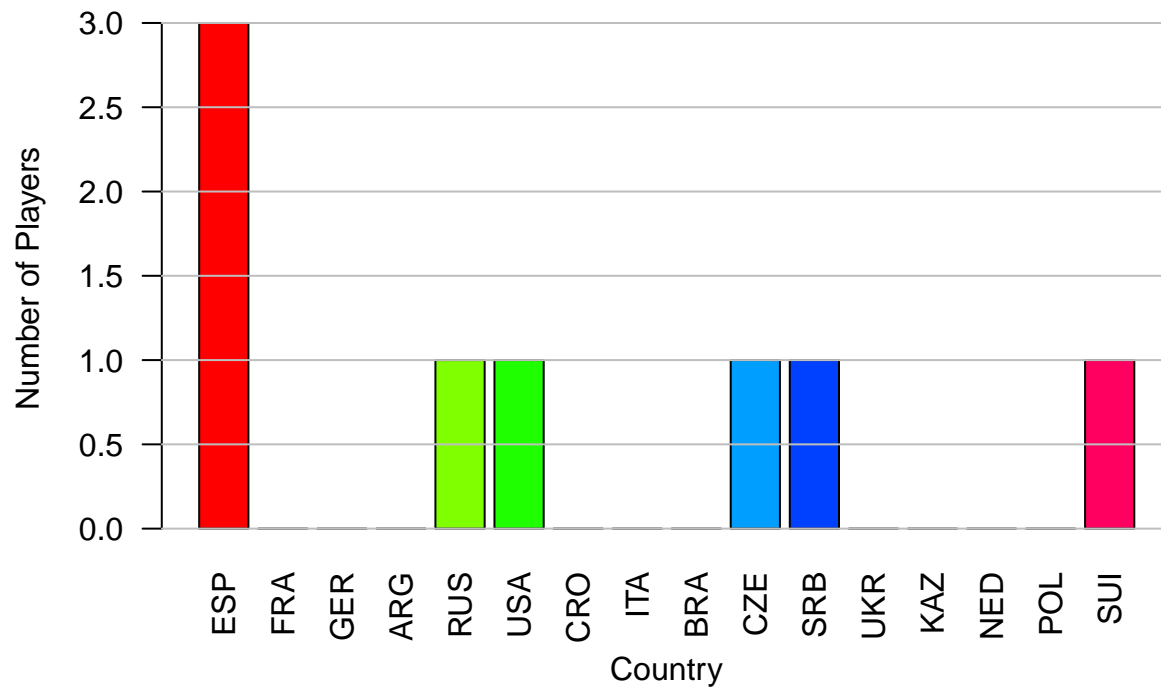
```
top20_bar <- barplot(final$top20, names.arg = rownames(final), ylab = "Number of Players",
  xlab = "Country", las = 2, col = rainbow_16, main = "Top 20 for 2010 by Country")
abline(h = seq(from = 0, to = 14, by = 1), col = 'gray')
```

Top 20 for 2010 by Country



```
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')
```

Top 10 for 2010 by Country



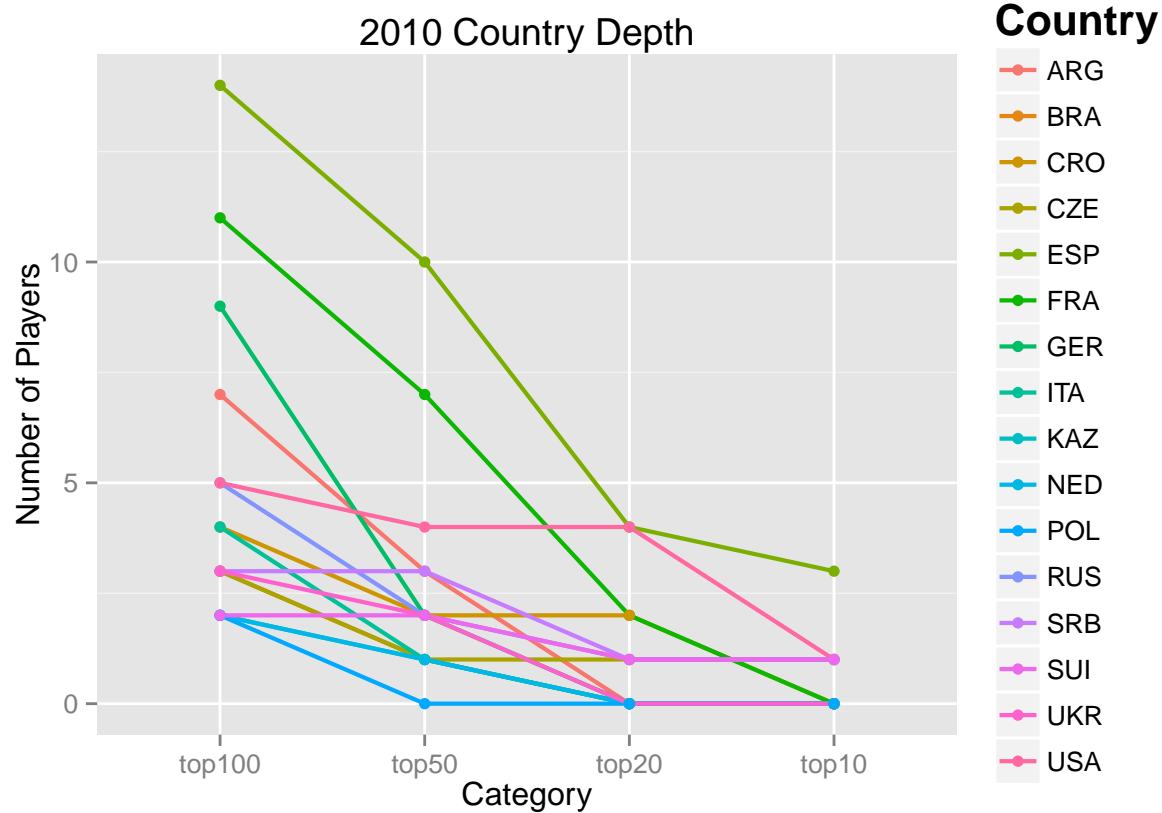
```
final_country_names <- rownames(final)
final_before_reshape <- final

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=.7)
```



```
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)
}
percentage_100
```

```
## [1] 0.17721519 0.13924051 0.11392405 0.08860759 0.06329114 0.06329114
## [7] 0.05063291 0.05063291 0.03797468 0.03797468 0.03797468 0.03797468
## [13] 0.02531646 0.02531646 0.02531646 0.02531646
```

```
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)
}
percentage_50
```

```
## [1] 0.23809524 0.16666667 0.04761905 0.07142857 0.04761905 0.09523810
## [7] 0.04761905 0.02380952 0.02380952 0.02380952 0.07142857 0.04761905
## [13] 0.02380952 0.02380952 0.00000000 0.04761905
```

```
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)
}
percentage_20
```

```
## [1] 0.2500 0.1250 0.0000 0.0000 0.0625 0.2500 0.1250 0.0000 0.0000 0.0625
## [11] 0.0625 0.0000 0.0000 0.0000 0.0000 0.0625
```

```
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_10
```

```
## [1] 0.375 0.000 0.000 0.000 0.125 0.125 0.000 0.000 0.000 0.125 0.125
## [12] 0.000 0.000 0.000 0.000 0.125
```

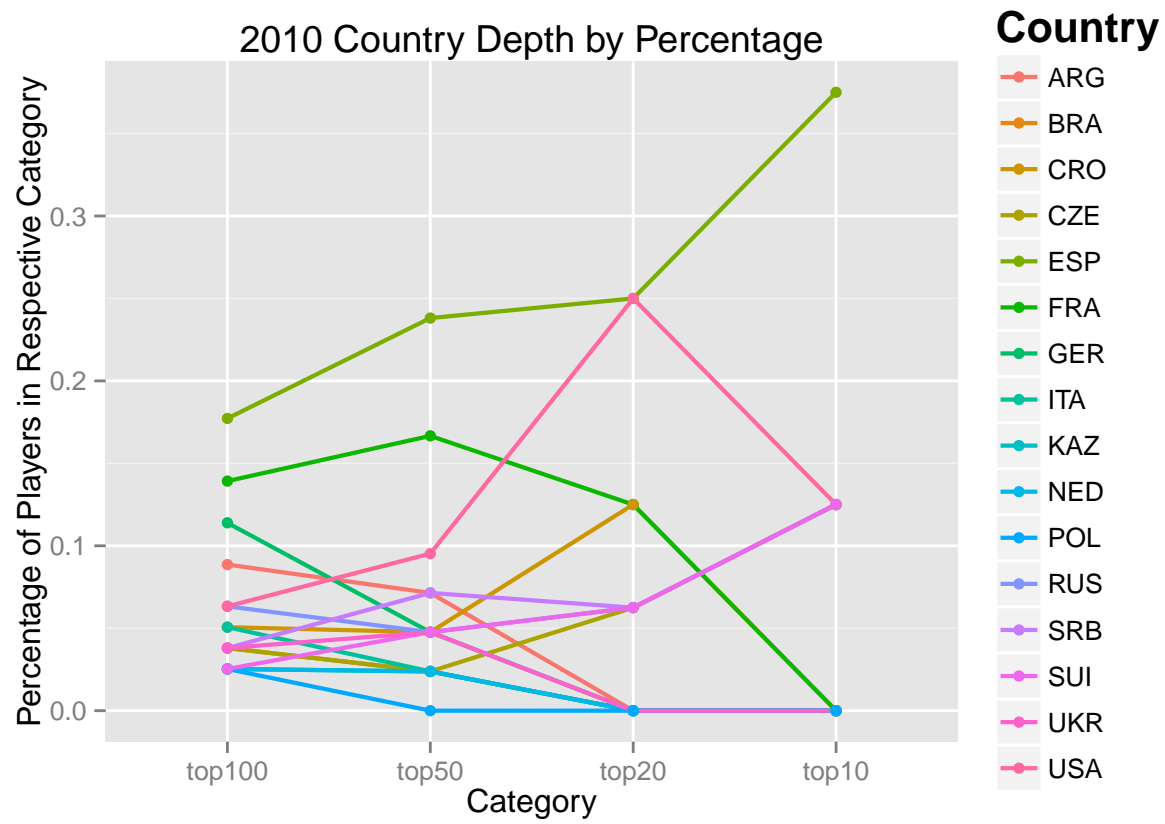
```
percentage_final <- final_before_reshape
percentage_final$top100 <- percentage_100
percentage_final$top50 <- percentage_50
percentage_final$top20 <- percentage_20
percentage_final$top10 <- percentage_10
```

```
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_l
```



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
all_merged
```

```
##      2010 2011 2012 2013 2014 2015
## ARG     7    6    7    6    5   NA
## ESP    14   13   13   14   12   10
## FRA    11   10   10   11    9    7
## GER     9    8    6    7    7    8
## RUS     5    5   NA    5   NA    5
## USA     5    9    7    7    5    6
```

```
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)
```

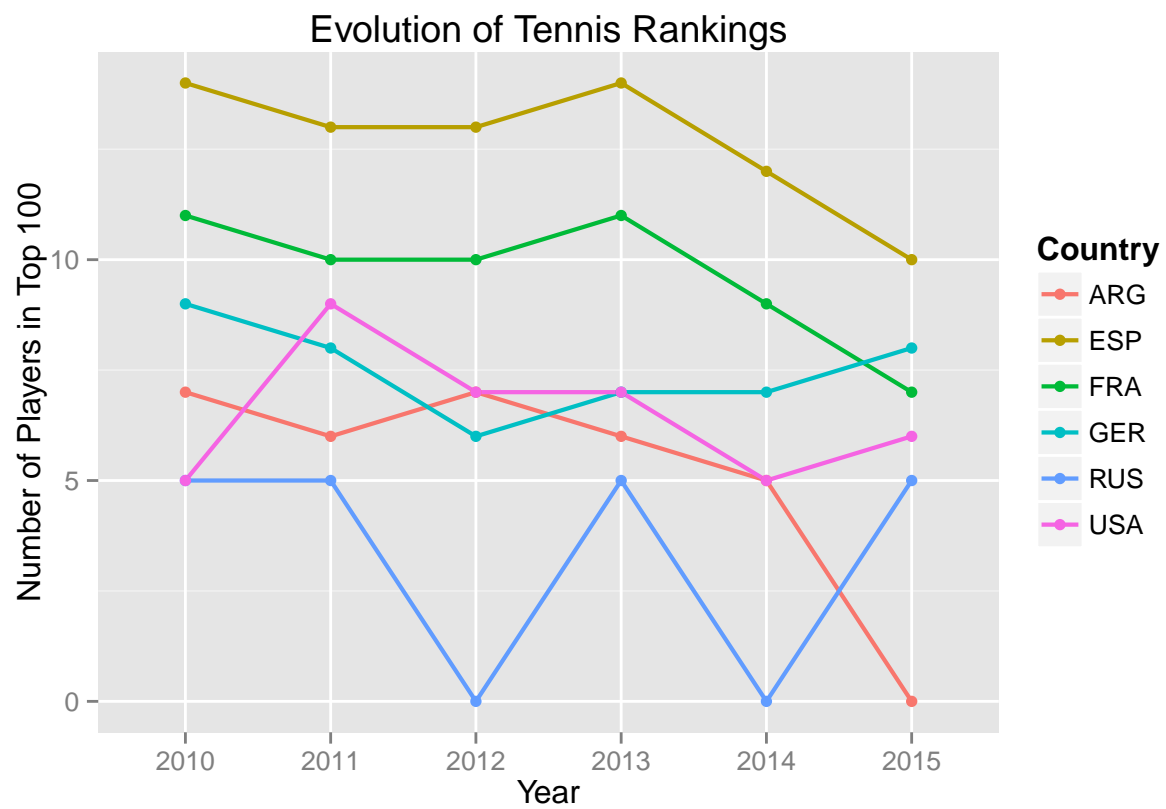
```
## No id variables; using all as measure variables
```

```
## 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(sil
```



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

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

Extraneous

```
# rankings_10$rank[which(rankings_10$country == rownames(depth_freqs)[1])]
# # get average rank of those 14. avg = 37.07143
# sum(each_best_rank) / sum(best)
#
#
# rownames(depth_freqs)[2]
# # For FRA, 11 players in top100, avg rank of those 11: 48.63636
# second <- rankings_10$country == rownames(depth_freqs)[2]
# second
# each_second_rank <- rankings_10$rank[which(second)]
# each_second_rank
# sum(each_second_rank) / sum(second)
#
#
# rownames(depth_freqs)[3]
#
# # For GER, 11 players in top100, avg rank of those 11: 62.6667
# third <- rankings_10$country == rownames(depth_freqs)[3]
# third
# each_third_rank <- rankings_10$rank[which(third)]
# each_third_rank
# sum(each_third_rank) / sum(third)
#
#
# rownames(depth_freqs)[4]
# # For ARG, 7 players in top100, avg rank of those 7: 58.57143
# fourth <- rankings_10$country == rownames(depth_freqs)[4]
# fourth
# each_fourth_rank <- rankings_10$rank[which(fourth)]
# each_fourth_rank
# sum(each_fourth_rank) / sum(fourth)
#
# rownames(depth_freqs)[5]
# # For ARG, 5 players in top100, avg rank of those 5: 58.2
# fifth <- rankings_10$country == rownames(depth_freqs)[5]
# fifth
# each_fifth_rank <- rankings_10$rank[which(fifth)]
# each_fifth_rank
# sum(each_fifth_rank) / sum(fifth)
#
# rownames(depth_freqs)[6]
# # For USA, 5 players in top100, avg rank of those 5: 32
# sixth <- rankings_10$country == rownames(depth_freqs)[6]
# sixth
# each_sixth_rank <- rankings_10$rank[which(sixth)]
# each_sixth_rank
# sum(each_sixth_rank) / sum(sixth)
#
#
#
#
```