Analysis of Tennis Rankings

William Cheung

December 9, 2015

DATA ANALYSIS BY RANKINGS

Introduction

We will use the Tennis ATP Rankings from 2015 to determine the competitive depth of a country's tennis representation. First we will read in the data from clean_data directory. We want the top 100 players from each year, and their corresponding country they represent. We have three parts to this analysis.
In Part I we will analyze at the rankings for 2015, looking for a country's depth of players. In Part II we will analyze the evolution of rankings over the years from 2010 to 2015. In Part III we will analyze the average amount of weight a player countributes to his country from 2015 stats.

```r
library("readr")
setwd("../code")

# read in the files from clean_data
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)

# Sorted frequencies of countries with > 1 player in the top 100 for 2010
country_freqs <- sort(table(rankings_15$country), decreasing = TRUE)
depth_freqs <- as.data.frame(country_freqs)
depth_freqs <- as.data.frame(depth_freqs[!(depth_freqs$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)
```

Part I.

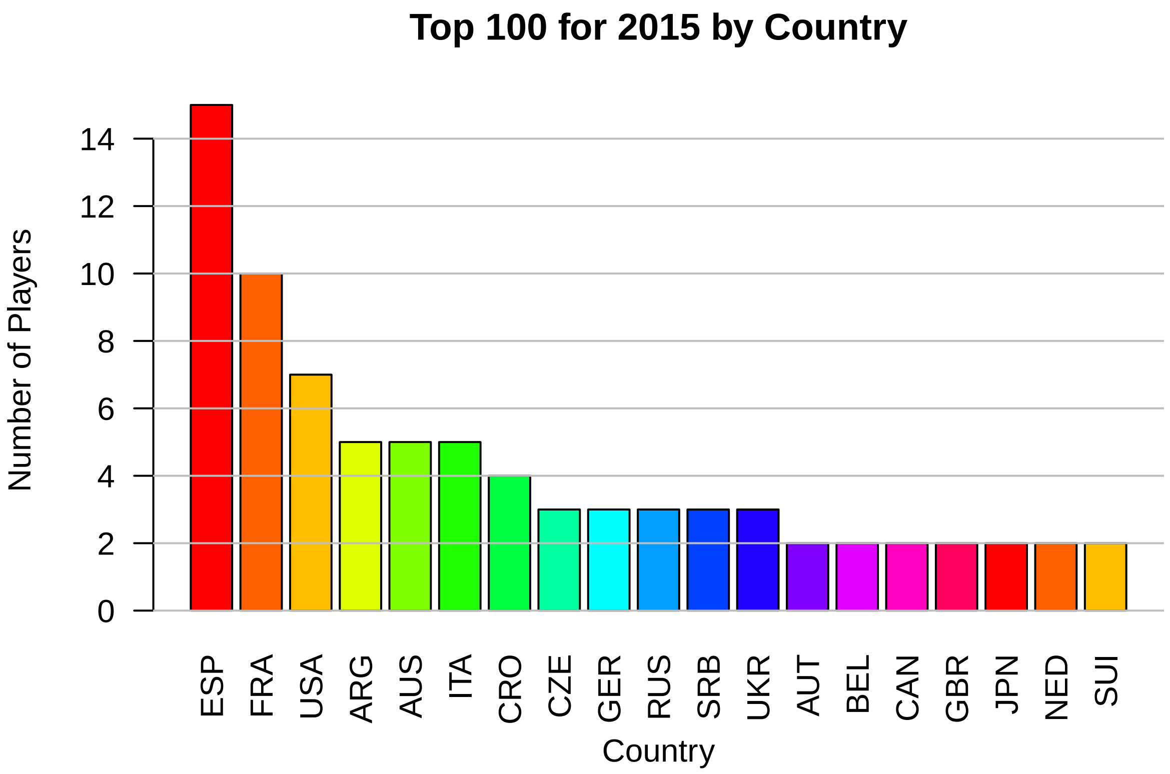
We will now break down the countries with most players in the top 100, then categorize them by counting the number of players in the top 50, top 20, and top 10. We define depth in this case as a country having more than 1 player in the top 100. Then we will dig deeper to see if these countries are still represented in the top 50, top 20, and top 10.

```r
library(reshape2)
library(ggplot2)

# Create a list, map the country with index of player ranking from 2010.
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_15$country == rownames(depth_freqs)[i])
  temp <- c(rankings_15$rank[index])
  list_top100[[i]] <- temp
}

# Create a list and map the country with ranking index in the top 50.
list_top50 <- list()
for (i in 1:nrow(depth_freqs)) {
  list_top50[[i]] <- list_top100[[i]][top50(list_top100[[i]])]
}

# Create a list and map the country with ranking index in the top 20.
list_top20 <- list()
for (i in 1:nrow(depth_freqs)) {
  list_top20[[i]] <- list_top100[[i]][top20(list_top100[[i]])]
}

# Create a list and map the country with ranking index in the top 10.
list_top10 <- list()
for (i in 1:nrow(depth_freqs)) {
  list_top10[[i]] <- list_top100[[i]][top10(list_top100[[i]])]
}

# Helper function to map names of countries to respective numbers.
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)
}
# Create list of countries and player freqeuncy in top 100, 50, 20, 10
list_of_freqs <- prepare(depth_freqs)
countries <- names(list_of_freqs)
```

```r
# Create vectors of each country's frequency of top 100, 50, 20, 10
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])
}

# Create data frame of countries with respective number
# of players in the top 100, 50, 20 and 10.
final <- data.frame(top100 = all_100,
                    top50 = all_50,
                    top20 = all_20,
                    top10 = all_10)
rownames(final) <- countries
```

First we will create basic barplots to visualize the number of players each country has in each category for 2015.

```
# Create barplots of the top 100, 50, 20, and 10 for each country.
rainbow_16 <-rainbow(n=16, s = 1, v = 1,
                     start = 0, end = max(1, 16 - 1)/16,
                     alpha = 1)


pdf("../images/top_100_2015.pdf")
a<-dev.cur()
png("../images/top_100_2015.png")
dev.control("enable")
top100_bar <- barplot(final$top100, names.arg = rownames(final),
                      ylab = "Number of Players",
                      xlab = "Country", las = 2,
                      col = rainbow_16,
                      main = "Top 100 for 2015 by Country")
abline(h = seq(from = 0, to = 14, by = 2), col = ' gray')
```



```
dev.copy(which=a)
```

```
## pdf
##   3
```

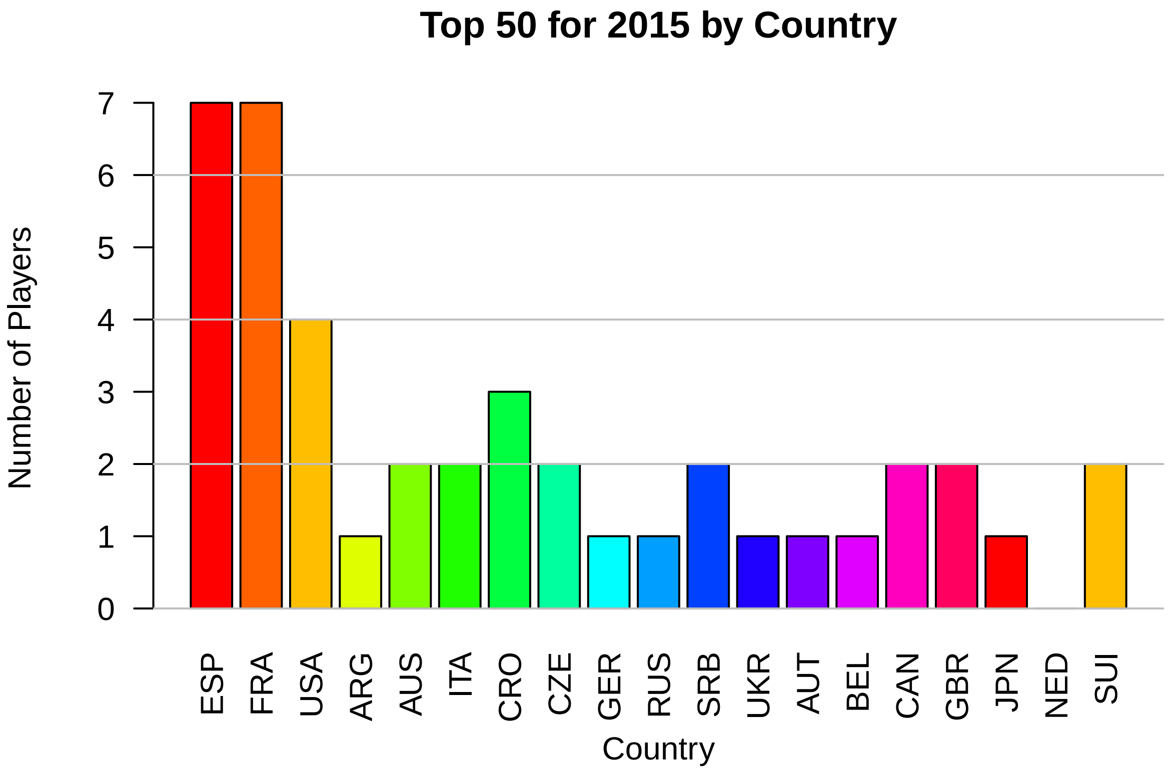
```
dev.off()
```

```
## quartz_off_screen
##                 4
```

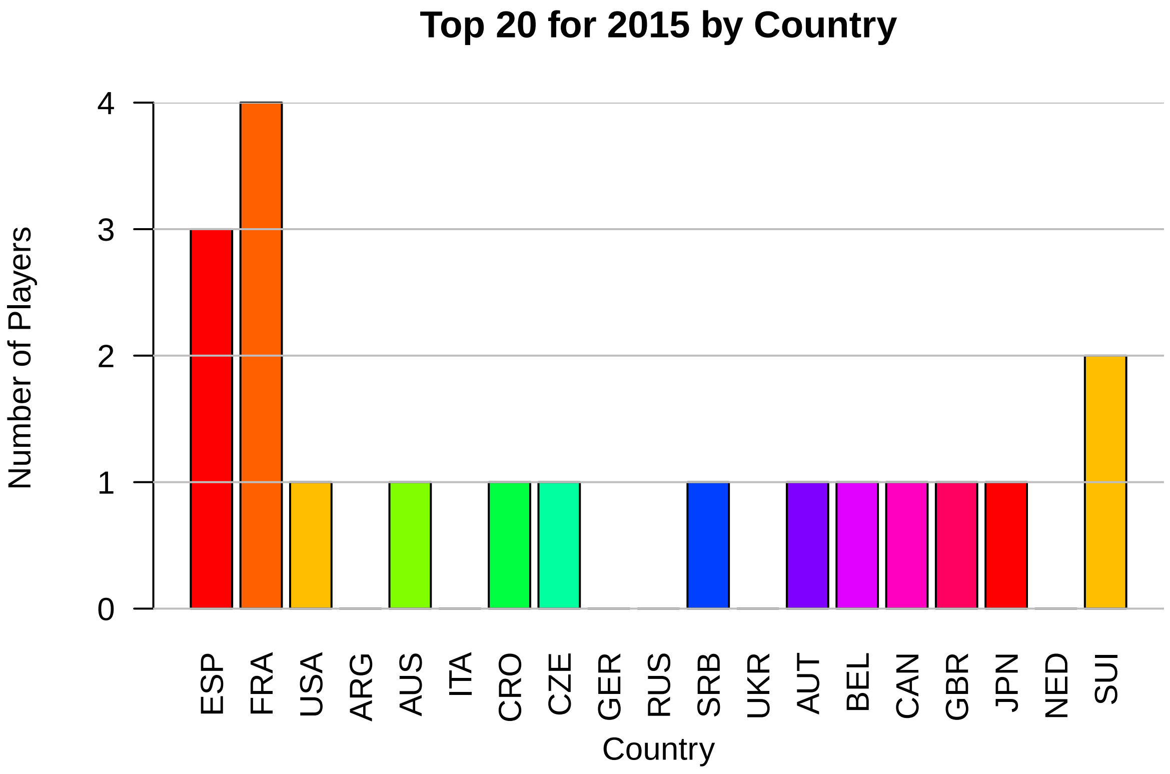
```
dev.off()
```

```
## pdf
##   2
```

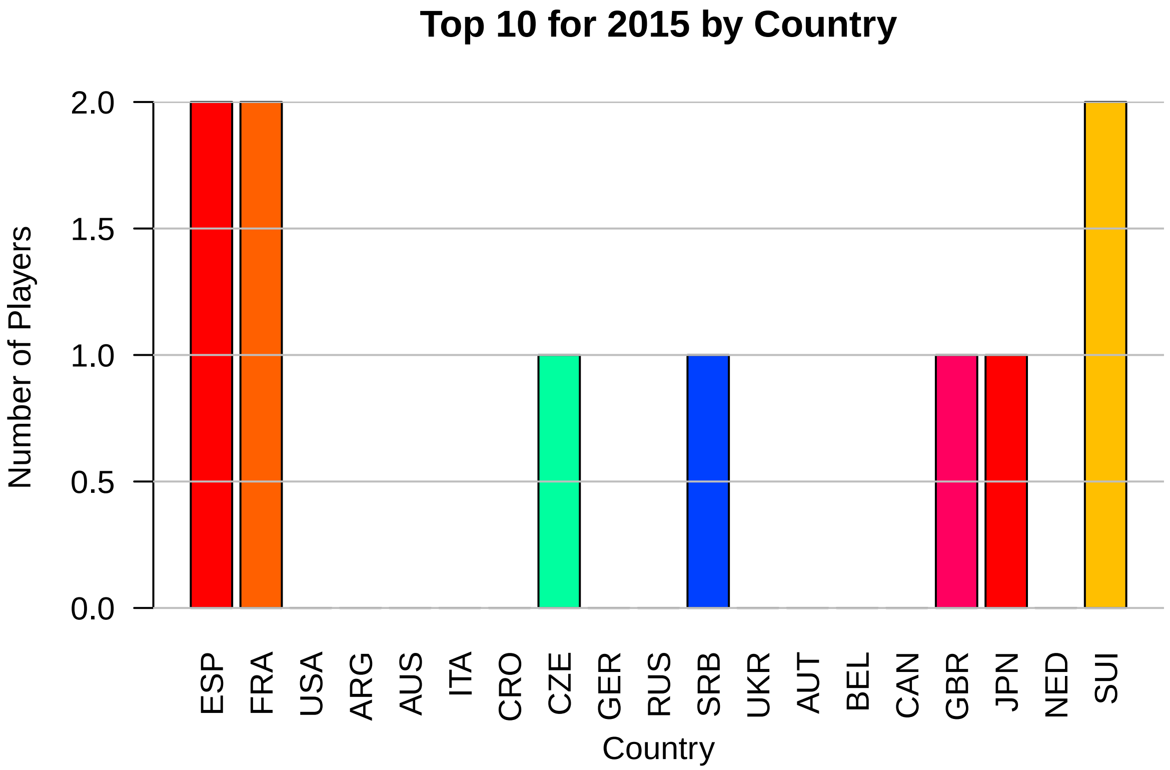
```
pdf("../images/top_50_2015.pdf")
a<-dev.cur()
png("../images/top_50_2015.png")
dev.control("enable")
top50_bar <- barplot(final$top50,
                     names.arg = rownames(final),
                     ylab = "Number of Players",
                     xlab = "Country",
                     las = 2,
                     col = rainbow_16,
                     main = "Top 50 for 2015 by Country")
abline(h = seq(from = 0, to = 14, by = 2), col = ' gray')
```



```
invisible(dev.copy(which=a))
invisible(dev.off())
invisible(dev.off())

pdf("../images/top_20_2015.pdf")
a<-dev.cur()
png("../images/top_20_2015.png")
dev.control("enable")
top20_bar <- barplot(final$top20,
                     names.arg = rownames(final),
                     ylab = "Number of Players",
                     xlab = "Country",
                     las = 2,
                     col = rainbow_16,
                     main = "Top 20 for 2015 by Country")
abline(h = seq(from = 0, to = 14, by = 1), col = ' gray')
```



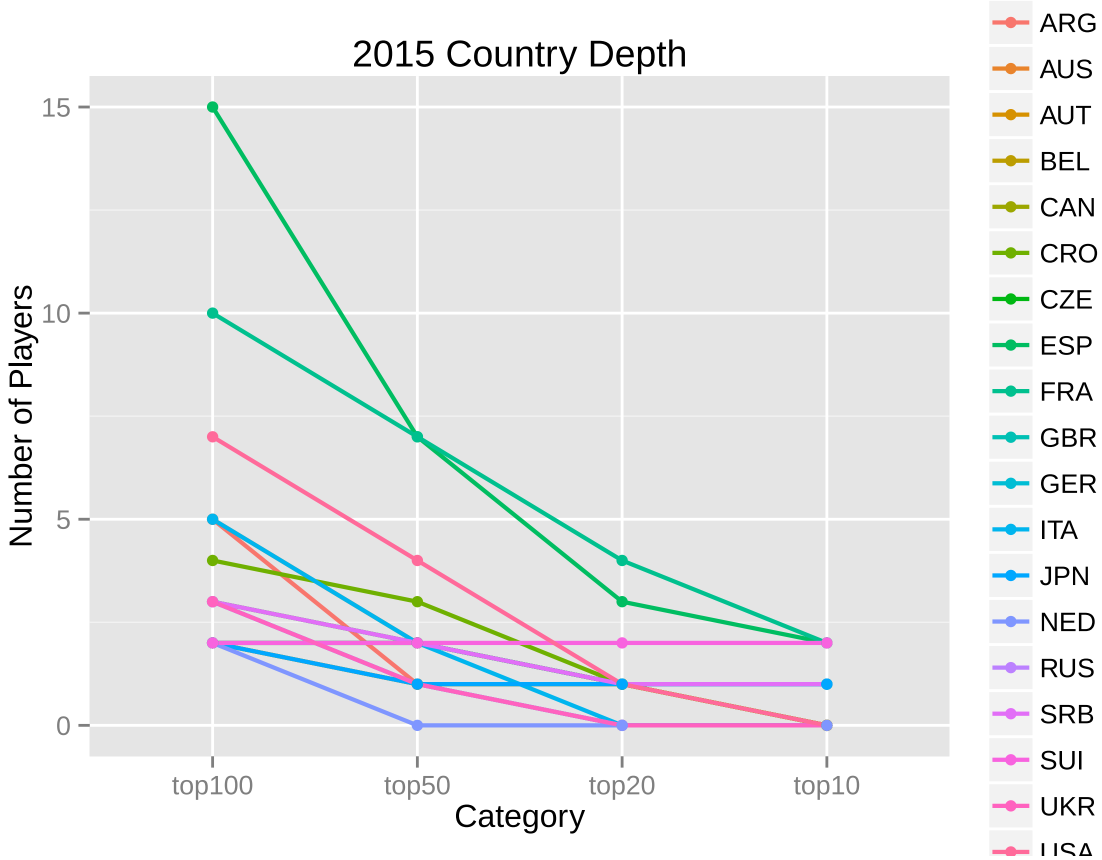
```
invisible(dev.copy(which=a))
invisible(dev.off())
invisible(dev.off())

pdf("../images/top_10_2015.pdf")
a<-dev.cur()
png("../images/top_10_2015.png")
dev.control("enable")
top10_bar <- barplot(final$top10,
                     names.arg = rownames(final),
                     ylab = "Number of Players",
                     xlab = "Country",
                     las = 2, col = rainbow_16,
                     main = "Top 10 for 2015 by Country")
abline(h = seq(from = 0, to = 14, by = 0.5), col = ' gray')
```



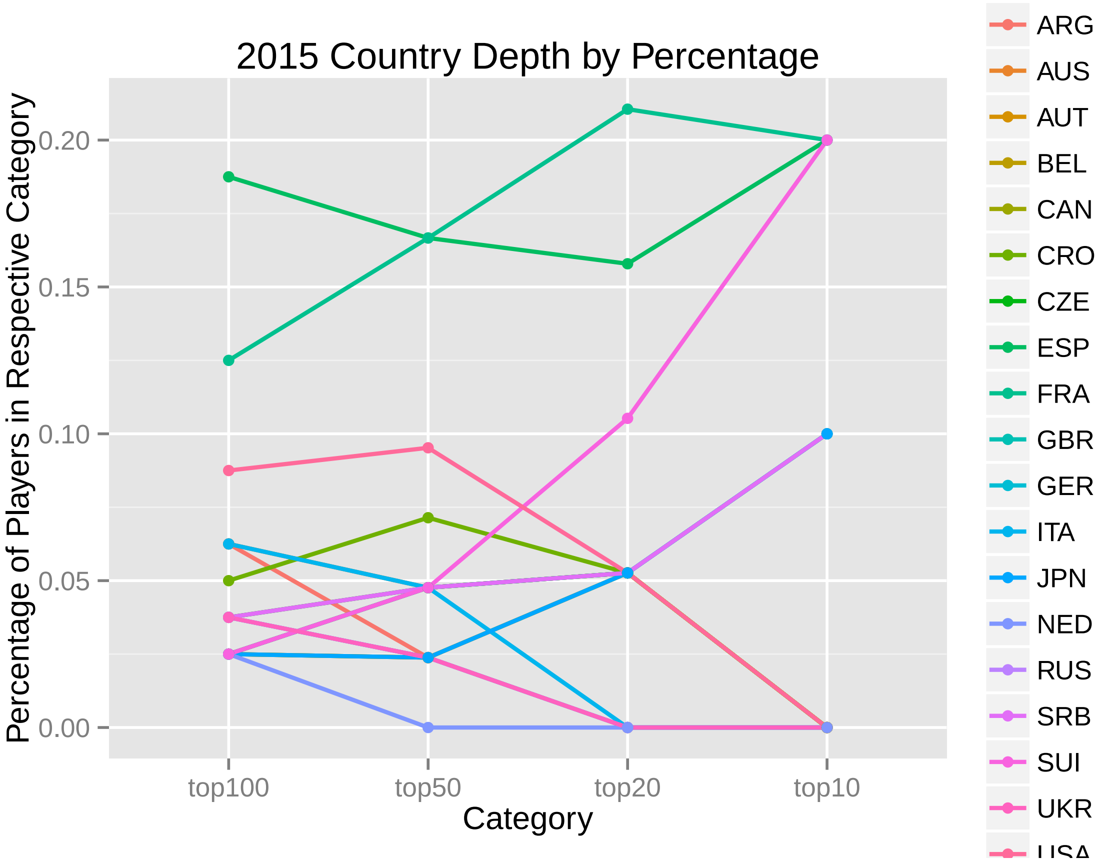
```
invisible(dev.copy(which=a))
invisible(dev.off())
invisible(dev.off())
```

After viewing the top 100, 50, 20, and 10 in 2010 by country on their own, we will now see if there is a difference in depth by tracking how many players were in the top 100, 50, 20, and 10 per each country together. We will do this by raw number of players, then percentage of players in that respective category.

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

# Reshape the data frame in preparation for ggplot
final <- melt(final)
final$country <- final_country_names

# Plot Country's Depth for 2015
pdf("../images/country_depth_2015.pdf")
a<-dev.cur()
png("../images/country_depth_2015.png")
dev.control("enable")
ggplot(final, aes(variable, value, group=factor(country),
                  color = factor(country))) +
        geom_line(size=.75) + geom_point() +
        ylab("Number of Players") + xlab("Category") +
        ggtitle("2015 Country Depth") +
        theme(legend.title = element_text(size=16, face="bold")) +
        scale_color_discrete(name="Country")
```



```
invisible(dev.copy(which=a))
invisible(dev.off())
invisible(dev.off())


# Now Plot the percentage of players in each category.
sum_top100 <- sum(final_before_reshape$top100)
percentage_100 = c()
for (i in 1:length(rownames(final_before_reshape))) {
  percentage_100 = c(percentage_100,
                     final_before_reshape[,1][i] / sum_top100)
}

# Calculate percentages
sum_top50 <- sum(final_before_reshape$top50)
percentage_50 = c()
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()
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()
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)
percentage_final$country <- final_country_names

# Plot the Country's Depth in 2015 by Percentage
pdf("../images/country_percentage_2015.pdf")
a<-dev.cur()
png("../images/country_percentage_2015.png")
dev.control("enable")
ggplot(percentage_final, aes(variable, value, group=factor(country),
                             color = factor(country))) +
    geom_line(size=.75) + geom_point() +
    ylab("Percentage of Players in Respective Category") +
    xlab("Category") +
    ggtitle("2015 Country Depth by Percentage") +
    theme(legend.title = element_text(size=16, face="bold")) +
    scale_color_discrete(name="Country")
```



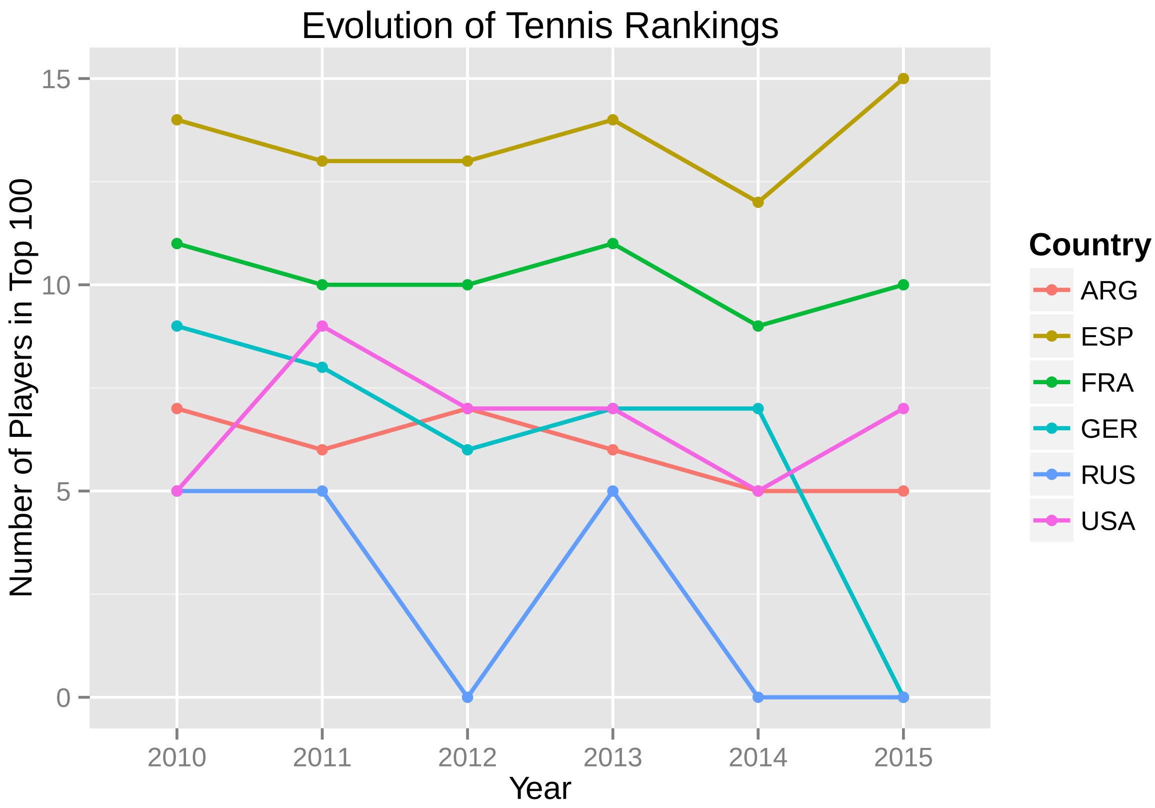
```
invisible(dev.copy(which=a))
invisible(dev.off())
invisible(dev.off())
```

Part II.

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
```

```r
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)
library(reshape2)
all_merged <- melt(all_merged)
all_merged[is.na(all_merged)] <- 0
all_merged$country <- country_names

# Plot the top players in countries by year
pdf("../images/evolution_rankings_2015.pdf")
a<-dev.cur()
png("../images/evolution_rankings_2015.png")
dev.control("enable")
ggplot(all_merged, aes(variable, value, group=factor(country),
                       color = factor(country))) + geom_line(size=.75) +
  geom_point() +
  ylab("Number of Players in Top 100") +
  xlab("Year") +
  ggtitle("Evolution of Tennis Rankings") +
  theme(legend.title = element_text(size=12)) +
  scale_color_discrete(name="Country")
```



```
invisible(dev.copy(which=a))
invisible(dev.off())
invisible(dev.off())
```

Part III.

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. We will use data from 2015. Average number of points per any country: this is the sum of all points / number of unique countries = averae points for a country. Our metric for determining how good a player is will be comparing the number of points that player has versus the numbr of points the average player has. Therefore we have: number of points of the country / # of players in that country = how good that player is.

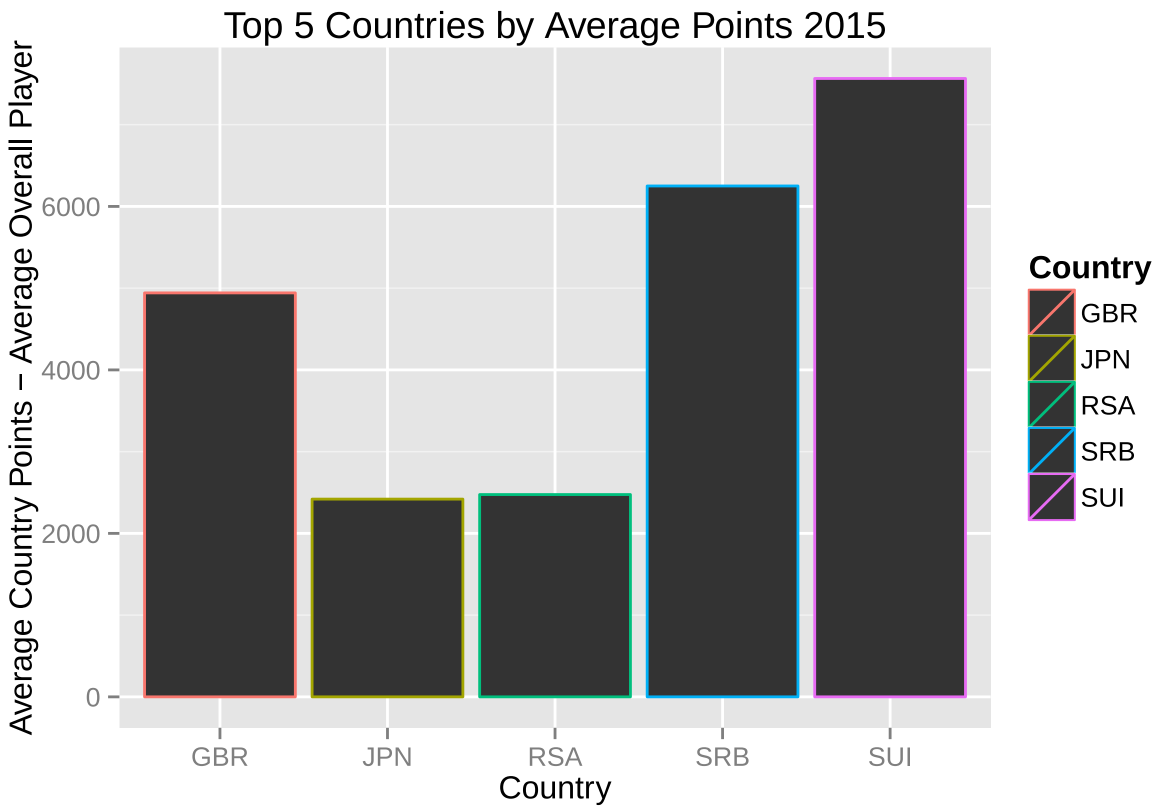
```r
# get sum of all the points
sum_2015_points <- sum(rankings_15$points)
distinct_countries <- unique(rankings_15$country)
avg_points_per_country <- sum_2015_points / length(distinct_countries)

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

# 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)

# 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 all average player points 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"

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)

top_5_countries_by_avg_points
```

```
##     difference
## SUI       7565
## SRB       6250
## GBR       4942
## RSA       2475
## JPN       2419
```

```
bottom_5_countries_by_avg_points
```

```
##     difference
## LAT        655
## BIH        649
## LTU        641
## IND        610
## ISR        578
```

```
pdf("../images/country_points_2015.pdf")
a<-dev.cur()
png("../images/country_points_2015.png")
dev.control("enable")
t <- ggplot(top_5_countries_by_avg_points,
            aes(rownames(top_5_countries_by_avg_points),
                difference, group=factor(country),
                color = factor(rownames(top_5_countries_by_avg_points))))
t + geom_bar(stat = "identity") + xlab("Country") +
  ylab("Average Country Points - Average Overall Player") +
  ggtitle("Top 5 Countries by Average Points 2015") +
  theme(legend.title = element_text(size=12)) +
  scale_color_discrete(name="Country")
```



```
invisible(dev.copy(which=a))
invisible(dev.off())
invisible(dev.off())
```