

Creating a World Cup Data Set

Brenna Curley
Stat 585 Final Project
April 2012

Introduction

Playing soccer is one of my greatest passions, and I have always wanted to be able to find usable soccer data to use in courses or as examples when TA-ing. However, I could never find any processed, “nice” .xls or .csv files to work with. Now with the strategies learned in this class I can start to bring together some soccer data. Specifically I will be examining World Cup data. The goal will be to bring together multiple years worth of data from past World Cups into a usable data set for future exploration and analysis.

The Data

The FIFA World Cup has been taking place every four years from 1930-2010 — excluding the two years (1942 and 1946) where there was no tournament due to WWII. The tournament lasts for around a month in the summer and has been won by eight different nations. Over the years the tournament has also evolved and grown. From 1934-1978 the tournament consisted of just 16 teams, in 1982 this expanded to 24 teams and now ever since 1998 the World Cup draws together a total of 32 teams for this popular world-wide event.

The raw data of interest may be reached from: <http://www.fifa.com/worldcup/archive/index.html> where tables on each team and for each year of the World Cup may be accessed. Not only are there separate tables (and hence urls) for each World Cup, but there are also a few separate tables each year that examine different aspects of the game such as goals scored and attempted as well as any penalties a team received. Figure 1 shows an example of these tables. We note that the table on the left — from 2010 — has slightly different information and formatting from the rest of the years — as the table on the right side of Figure 1 shows. Data from these websites were collected from all 19 World Cups and combined into one complete data set.

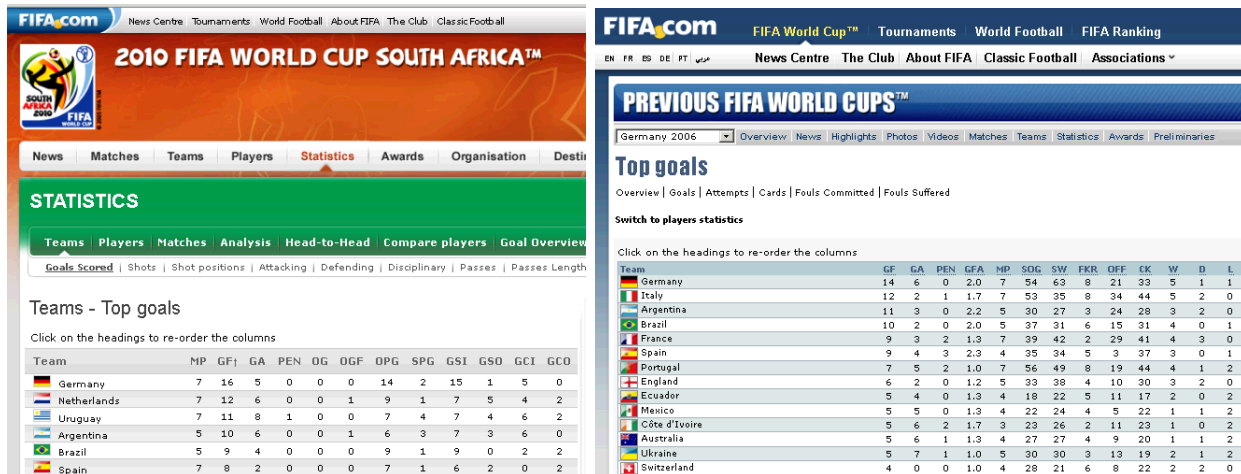


Figure 1: Examples of the data tables from 2010 (left) and all other years (right) off of the FIFA website.

Pulling data from the web

Since the data of interest comes from an online source, the `XML` package in R will be used to setup functions which will automatically pull the data from the various websites. Primary functions used from this library will include `getNodeSet` and `readHTMLTable`.

1930-2006 World Cups

Excluding the most recent 2010 World Cup, the FIFA website keeps consistent tables for each tournament. This facilitates us to create a single function in order to pull off data from these websites. Within each year, there are also two different tables of interest — one containing such information as number of goals, shots, and wins while the other contains information on penalties and number of cards given to each team. A function is created to scrape these data from the web. In order to be safe and get nicer variable names (for example to avoid formatting of variable names appearing as `GF'` that come out of the `readHTMLTable` function), the function `getNodeSet` followed by the function `sapply` are used with the `xmlValue` function in order to extract variable names. Using this cleaner version of the variable names, we can easily add them to the table that `readHTMLTable` pulls off for us. The function to pull these data tables is as follows:

```
scrape <- function(url) {
  doc <- htmlParse(url)
  tables <- getNodeSet(doc, "//table")
  tab2 <- tables[[1]]
  vars <- getNodeSet(tab2, "//acronym[@title]")
  var_names <- sapply(vars, FUN = xmlValue)

  tabs <- readHTMLTable(url)
  tab <- tabs[[1]]
  colnames(tab) <- c("Team", var_names)

  return(tab)
}
```

Now that a function to scrape the data off the web has been created, we need to figure out a way to merge together the two different tables for each year. A function called `bind` is created in which we may input a list of urls (two tables for these years), scrape the information into a list of two tables, and then bind together to output one table per year. Then we may easily use the `merge` function to combine the tables by Team.

Once merged, there is still some cleaning up of the tables that must be done. One basic cleanup dealt with the failed transcription of the accent in 'Côte d'Ivoire' from the HTML source code. A simple `gsub` statement was able to remedy this. Other variable names that caused problems in R were those that began with a numerical value such as in '2YC' which was easily fixed using : `names(dat)[15] <- "YCR"`. One of the last details changed was to fix the structure of the data table we wish to output. For example, most of the variables in the table – such as number of shots on goals, or number of yellow cards received — should be numeric. So for these pertinent variables,

this was easily fixed using `as.numeric`. By doing this we save an extra step when wishing to use these data for plots and tables in the future.

This `bind` function we create is built to run on each year of the tournament separately, however eventually we aim to merge all the tournaments together. Thus, we wish to uniquely identify each table with the year and location of that World Cup — information which may also prove useful in plots and analyses in the future. Although one could manually do this for each year once the function is run, we may also save time by automating this process within our function by using elements we can pull off the website. We note that both the year and the host nation may be found in the root of the html source code by using `xmlRoot`. Once this information is pulled off using both `getNodeSet` and `xmlGetAttr`, we are still left with a long string of characters such as: 2006 FIFA World Cup Germany,2006 World Cup,Germany 2006,2006 Football World Cup, World Cup 2006,FIFA World Cup 2006 Germany,FIFA World Cup Germany 2006, Germany World Cup,German World Cup. Using `strsplit` we can split this string up by spaces and then select which ones we would like that correspond to the year and location of the particular World Cup that website corresponds to. We must be careful with the year 2006, however, where the year is not the same numbered character in the string as all other years. A simple `if else` statement may be used to remedy this. The full process for this step can be seen below:

```
# Pull out the Year and the Host Nation
doc <- htmlParse(urls[[1]])
r <- xmlRoot(doc)

info <- getNodeSet(r, "//meta")[[3]]
info2 <- xmlGetAttr(info, "content")
info.sp <- strsplit(info2, " ")

Year <- as.numeric(info.sp[[1]][1])
if (Year == 2006) {
  Host <- info.sp[[1]][20] # all the rest seem to be in 5
} else {
  Host <- info.sp[[1]][5]
}

# Add Year and Loc to the data set
dat$Year <- Year
dat$Loc <- Host
```

2010 World Cup

Unfortunately the website for the most recent World Cup does not follow the same format as all previous years. However it also has a lot more data for each of the teams. Thus, not only will we create a table that is consistent with the other years, but we will also create a specific table just for the 2010 tournament. Perhaps if future World Cup tournaments follow this new formatting of

the website, this richer data table may be added onto.

Overall, many of the same general steps were taken with the 2010 data as in the previous section. Some extra steps needed to be added, however, in order to clean-up the data as well as make it consistent with the previous years. Since there was more data collected this year, that also meant that there were more tables to merge together. This unfortunately led to a couple of variables repeating more than once, as well as a couple different variables ending up with the same name despite the different values they represent. Those with the same name included CC (Clearances Completed and Corners Completed) and CCR (Clearances Completion Rate and Crosses Completion Rate). Using `gsub` and indexing which variables to keep, we are able to fix this issue as seen in the code below:

```
dat$CC2.y <- dat$CC2.x
dat$CCR2.y <- dat$CCR2.x
tst1 <- gsub("CC.y", "CC2.x", names(dat))
tst2 <- gsub("CCR.y", "CCR2.x", tst1)
colnames(dat) <- tst2
tst <- gsub(".y", "", tst2) # get rid of suffix '.y'
idx <- which(colnames(dat) == tst) # get rid of those '.y' columns
yep <- dat[, idx]
vars <- gsub(".x", "", names(yep))
colnames(yep) <- vars
```

A couple of things also need to be adjusted in order for the 2010 data to merge easily with all the previous years. For example, the variables for Corner Kicks and Free-Kick Shots need to be changed from COR to CK and FKR to FKS, respectively. The previous years' data also include information on the number of wins, losses, and draws each team had during the tournament. To add this to the 2010 table, we must do a bit more work and reach out to another online source. Specifically, the Wikipedia website, http://en.wikipedia.org/wiki/2010_FIFA_World_Cup_statistics#Wins_and_losses, has this information in a nice tabular format. As seen below, the `readHTMLTable` function works to pull the table off from online. Then after changing a few variable names for consistency, we may merge these data with our main 2010 data set from the FIFA website:

```
wiki =
"http://en.wikipedia.org/wiki/2010_FIFA_World_Cup_statistics#Wins_and_losses"
wdl_list <- readHTMLTable(wiki)
wdl <- wdl_list[[11]][, c(1, 3:5)]
wdl$Team <- wdl$Nation
wdl <- wdl[-dim(wdl)[1], -1]
teams <- gsub("United States", "USA", wdl$Team)
wdl$Team <- teams
res <- merge(yep, wdl, by = "Team")
```

As briefly mentioned before, we would rather not waste all the information given for the 2010 World Cup, but would also would like to merge these data with all other years. Thus the function

`bind2010` was created in order to output a list containing two different tables — the full 2010 data set (`res_all2010`) and a condensed data set (`res_unif`) in the same format of all the previous years. An index and subsetting method was used in order to pick off and order the columns as needed for the condensed table. The full R function may be found in the Appendix.

Merging All Years

Since all the data tables for each year were created to have the same formatting, they were easily merged together using `rbind`. Once this is accomplished, there are still a few things to add and adjust slightly. Of possible interest, but not a part of the FIFA website tables, would be who won each tournament. Thus we can look to a table on Wikipedia (http://en.wikipedia.org/wiki/List_of_FIFA_World_Cup_finals) that gives the winner and runner-up for each year. As a side note, this website should be used with some care, however, as just in the past couple of weeks the website must have been updated as the desired number table to pull off from the html source code needed to be adjusted. Our R code was easily fixed to accommodate this change.

Once read in, a few variables need to be modified in order to easily merge together with the main data table. For example, a few strange characters again need to be overwritten using `gsub`. Also, entries of 'West Germany' for the teams in the Wikipedia table needed to be substituted with 'Germany FR' to be consistent with our main data set. Once these alterations are made, the table may easily be combined to the main data set using the `merge` function. A sample of the first six rows of our final product for the table including all years may be seen below (and variable descriptions may be found in the Appendix):

Year	Winners	Runnersup	Team	MP	GF	GA	PEN	SOG	SW
1930	Uruguay	Argentina	Romania	2.00	3.00	5.00	0.00	3.00	0.00
1930	Uruguay	Argentina	Uruguay	4.00	15.00	3.00	0.00	15.00	0.00
1930	Uruguay	Argentina	USA	3.00	7.00	6.00	0.00	7.00	0.00
1930	Uruguay	Argentina	Yugoslavia	3.00	7.00	7.00	0.00	7.00	0.00
1930	Uruguay	Argentina	Argentina	5.00	18.00	9.00	0.00	18.00	0.00
1930	Uruguay	Argentina	Belgium	2.00	0.00	4.00	0.00	0.00	0.00

FKR	OFF	CK	W	D	L	Y	YCR	R	FC	FS	Loc
0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	Uruguay
0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	Uruguay
0.00	0.00	0.00	2.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	Uruguay
0.00	0.00	0.00	2.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	Uruguay
0.00	0.00	0.00	4.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	Uruguay
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	Uruguay

Table 1: Sample of full data table for all World Cup tournaments.

Results

After all the work to put together these nice data sets, we can now use them for various plots or analyses in the future. As an illustration, we consider a few possibly interesting examples here.

Shots on goal by team

Clearly taking as many shots as one can will only help a team on their way to winning games. Differing strategies of each team may also lead to differences in the amount of shots they take. Thus, one question of interest may deal with looking at the shots on goal for each team. In Figure 2 we consider the shots on goal for each team that participated in the World Cup at least five times. It appears Portugal and then Brazil have the most overall shots on goal on average.

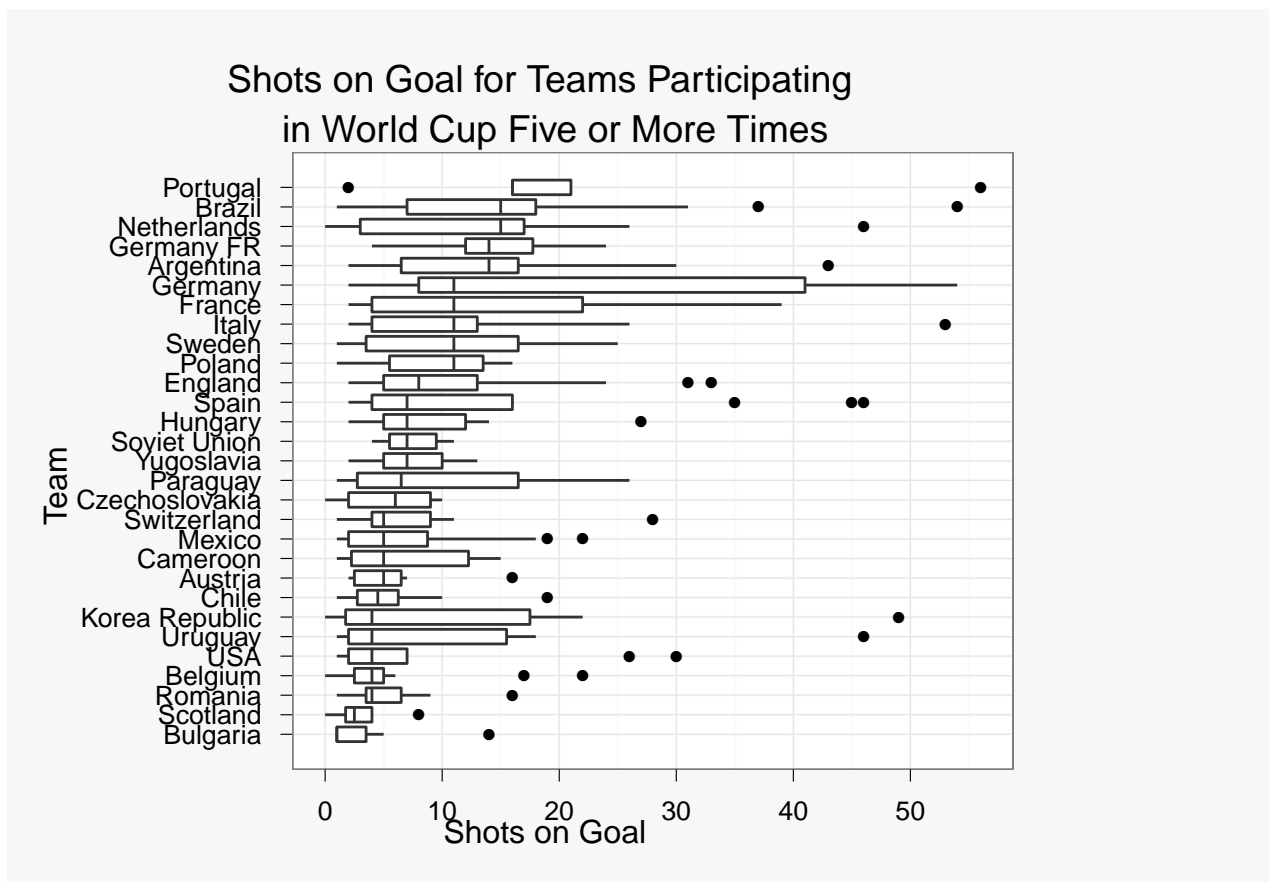


Figure 2: Shots on goal by team — considering only teams who have made over five appearances at the World Cup. We see that Portugal has the most shots on goal on average.

However, since many teams have participated in more games than others, we may also want to account for this by scaling the shots on goal for each team by the total number of matches they played. Doing this we see in Figure 3 that Portugal still has the most shots on goal per game on average.

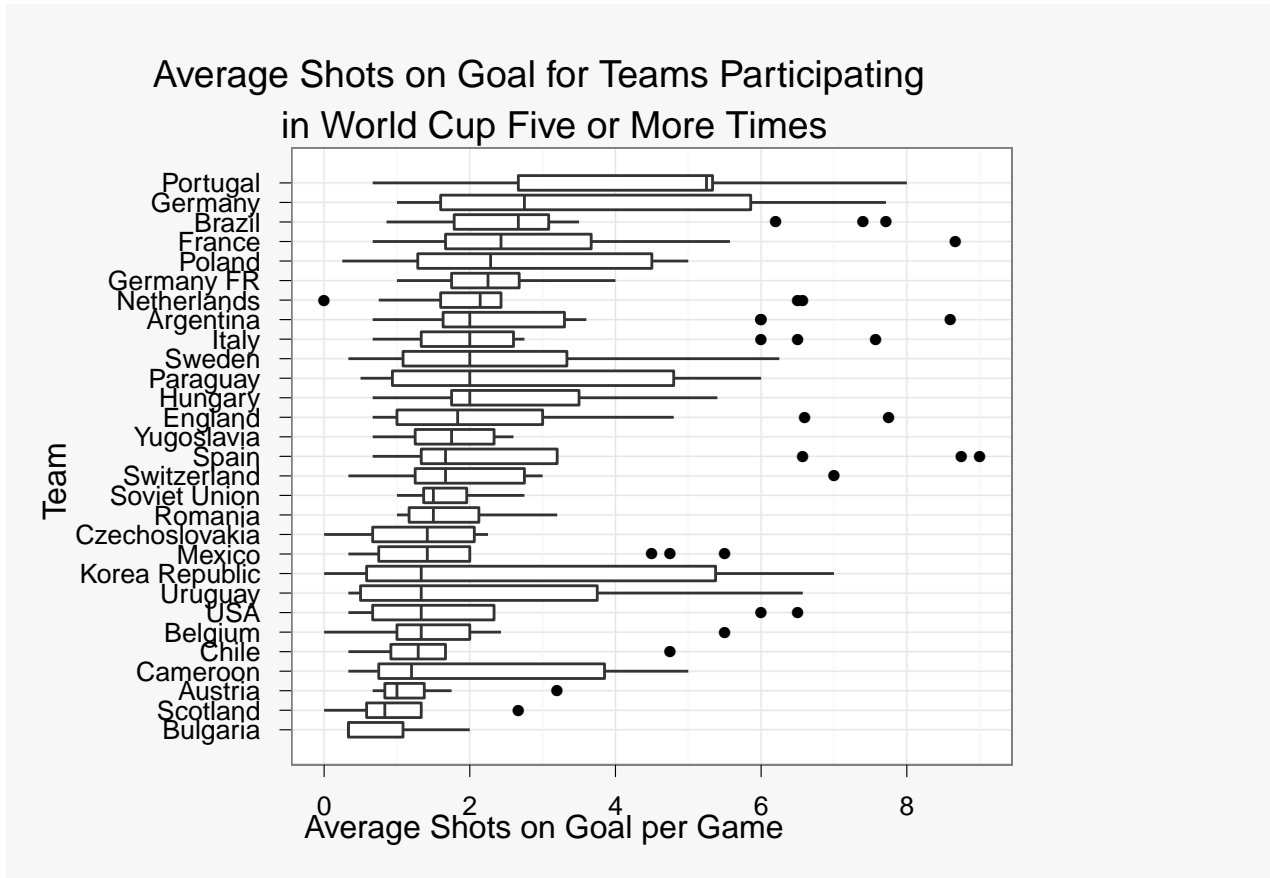


Figure 3: Average shots on goal per game by team — considering only teams who have made over five appearances at the World Cup. We see that Portugal has the most shots on goal per game on average.

Shots on goal over the years

Since our newly created data set also contains information on a team's performance by year, we may also wish to consider looking into how the number of shots on goal per game has evolved over the years. Looking at Figure 4 we note that overall there seems to be a fairly constant trend in the number of shots per game with a big jump after the 1998 World Cup.

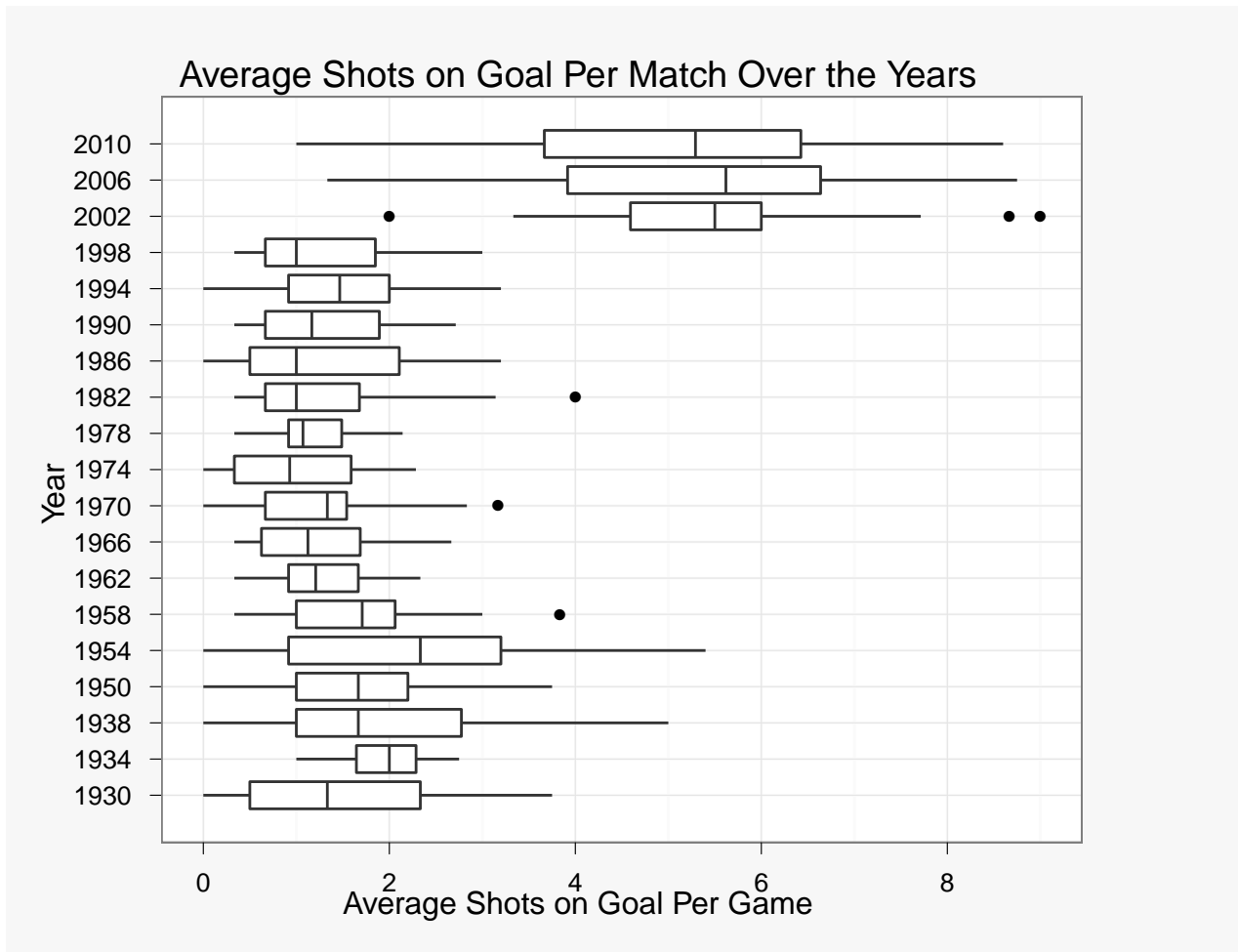


Figure 4: Average shots on goal per game over the years. We see that there is a fairly constant trend in the number of shots per game with a big jump after the 1998 World Cup.

Conclusion

Overall it was great to be able to use a variety of different packages and functions in R in order to create this World Cup data set that I look forward to using in the future. As more World Cups take place, I hope that the websites follow a similar format to either the 2010 year or the other years, so that new data may be added easily. Now just have to wait until Brazil 2014 to find out!

Appendix

2010 Variables

Variable	Description
Team	Country (Team) name
MP	Matches played
GF	Goals for
GA	Goals against
PEN	Penalty goals
OG	Own goals
OGF	Own goals for
OPG	Open play goals
SPG	Set piece goals
GSI	Goals scored in penalty area
GSO	Goals scored from outside penalty area
GCI	Goals conceded in penalty area
GCO	Goals conceded from outside penalty area
TS	Total shots
SOG	Shots on goal
SW	Shots wide
SB	Shots blocked
FKR	Free-kick shots
FKD	Free-kick shots (direct)
FKI	Free-kick shots (indirect)
SOB	Shots on bar
SOP	Shots on post
SGP	Shots on goal from penalty area
SGO	Shots on goal from outside penalty area
SWP	Shots wide from penalty area
SWO	Shots wide from outside penalty area
BSI	Blocked shots from inside the penalty area
BSO	Blocked shots from outside penalty area
SG/S	Percent shots on goal (shots on goal/shots)
AT	Attacking
AFL	Attacks from left
AFC	Attacks from center
AFR	Attacks from right

Variable	Description
TS	Tackles suffered
TSL	Tackles suffered losing possession
OFF	Offsides
ASS	Assists
SR	Solo runs
DPA	Deliveries in penalty area
CLR	Clearances
CC	Clearances completed
CCR	Clearances completion rate
TCK	Tackles
TGP	Tackles made gaining possession
SAV	Saves
Y	Yellow cards
YCR	Second yellow card and red card
R	Red cards
FC	Fouls committed
FS	Fouls suffered
HB	Handballs
PC	Passes completed
PCR	Passes completed rate
CRO	Crosses
CRC	Crosses completed
CCR2	Crosses completion rate
CK	Corners
CC2	Corners completed
COCR	Corners completion rate
THR	Throw-ins
Year	Year of the World Cup
Loc	Host nation (country where World Cup was played)

All Years' Variables

Variable	Description
Team	Country (Team) name
MP	Matches played
GF	Goals for
GA	Goals against
PEN	Penalty goal
SOG	Shots on goal
SW	Shots wide
FKR	Free kicks received
OFF	Offsides
CK	Corner kicks
W	Wins
D	Draws
L	Losses
Y	Yellow cards
YCR	Second yellow card and red card
R	Red cards
FC	Fouls committed
FS	Fouls suffered
Year	Year of the World Cup
Loc	Host nation (country where World Cup was played)

R Functions

1930-2006 World Cup Functions

```
scrape <- function(url) {
  doc <- htmlParse(url)
  tables <- getNodeSet(doc, "//table")
  tab2 <- tables[[1]]
  vars <- getNodeSet(tab2, "//acronym[@title]")
  var_names <- sapply(vars, FUN = xmlValue)

  tabs <- readHTMLTable(url)
  tab <- tabs[[1]]
  colnames(tab) <- c("Team", var_names)

  return(tab)
}

bind <- function(urls) {
  try <- as.list(NA)
  for (i in 1:length(urls)) {
    try[[i]] <- scrape(urls[[i]])
  }

  # Merge Tables #
  dat <- NA
  for (i in 1:(length(try) - 1)) {
    dat <- merge(try[[i]], try[[i + 1]], by = c("Team",
      "MP"))
  }

  # Change Cote D'Ivoire to look nicer #
  dat$Team <- gsub("Cte d'Ivoire", "Cote d'Ivoire", dat$Team)

  # Remove Goals for Average (GFA) column
  dat <- dat[, -match("GFA", names(dat))]

  # R also doesn't like the '2' in the '2YC' variable so
  # must change
  names(dat)[15] <- "YCR"

  # Pull out the Year and the Host Nation
  doc <- htmlParse(urls[[1]])
  r <- xmlRoot(doc)
```

```

info <- getNodeSet(r, "//meta")[[3]]
info2 <- xmlGetAttr(info, "content")
info.sp <- strsplit(info2, " ")

Year <- as.numeric(info.sp[[1]][1])
if (Year == 2006) {
  Host <- info.sp[[1]][20] # all the rest seem to be in 5
} else {
  Host <- info.sp[[1]][5]
}

# Add Year and Loc to the data set
dat$Year <- Year
dat$Loc <- Host

# Make reasonable variables numeric
for (i in 2:18) {
  dat[, i] <- as.numeric(as.character(dat[, i]))
}

return(dat)
}

```

2010 World Cup Functions

```

scrape2010 <- function(url) {
  doc <- htmlParse(url)
  tables <- getNodeSet(doc, "//table")
  tab2 <- tables[[2]]
  vars <- getNodeSet(tab2, "//acronym[@title]")
  var_names <- sapply(vars, FUN = xmlValue)

  tabs <- readHTMLTable(url)
  tab <- tabs[[2]]
  colnames(tab) <- c("Team", var_names)

  return(tab)
}

bind2010 <- function(urls) {
  # urls is a list of the url for all tables want to
  # merge together

```

```

try <- as.list(NA)
for (i in 1:length(urls)) {
  try[[i]] <- scrape2010(urls[[i]])
}

## Merge Tables ##
dat <- as.list(NA)
for (i in 1:(length(try) - 1)) {
  dat[[1]] <- try[[1]]
  dat[[i + 1]] <- merge(dat[[i]], try[[i + 1]], by = c("Team",
    "MP"))
}
dat <- dat[[length(try)]]
dat$Year <- "2010"
dat$Loc <- "South Africa"

# Get rid of repetitive columns and careful re repeats
# of
# CC and CCR which ARE different variables and hence
# not repetitive info
dat$CC2.y <- dat$CC2.x
dat$CCR2.y <- dat$CCR2.x
tst1 <- gsub("CC.y", "CC2.x", names(dat))
tst2 <- gsub("CCR.y", "CCR2.x", tst1)
colnames(dat) <- tst2
tst <- gsub(".y", "", tst2) # get rid of suffix '.y'
idx <- which(colnames(dat) == tst) # get rid of those '.y' columns
yep <- dat[, idx]
vars <- gsub(".x", "", names(yep))
colnames(yep) <- vars

# add wins, losses, and draws
wiki =
"http://en.wikipedia.org/wiki/2010\_FIFA\_World\_Cup\_statistics#Wins\_and\_losses"
wdl_list <- readHTMLTable(wiki)
wdl <- wdl_list[[11]][, c(1, 3:5)]
wdl$Team <- wdl$Nation
wdl <- wdl[-dim(wdl)[1], -1]
teams <- gsub("United States", "USA", wdl$Team)
wdl$Team <- teams
res <- merge(yep, wdl, by = "Team")

# Change Cote D'Ivoire to look nicer
res$Team <- gsub("Cte d'Ivoire", "Cote d'Ivoire", yep$Team)

```



```

# Change variable names in 2010 so that match up with
#   the rest of the years
# R also doesn't like the '2' in the '2YC' variable so
#   must change
names(res)[c(18, 57, 47)] <- c("FKR", "CK", "YCR")

# Make certain variables numeric
for (i in 2:60) {
  res[, i] <- as.numeric(as.character(res[, i]))
}
for (i in 63:65) {
  res[, i] <- as.numeric(as.character(res[, i]))
}

# Also create a minimized wc2010 with only variables
#   can merge together
# perhaps have our bind2010 return both small and full
#   dataset
idx <- c(1:5, 15:16, 18, 36, 57, 63:65, 46:50, 61:62)
res2 <- res[, idx]

res_all <- list(res_all2010 = res, res_unif = res2)
return(res_all)
}

wc2010 <- bind2010(maybe)

```

Code to pull off winner and runner-up each year

```

winner <- "http://en.wikipedia.org/wiki/List_of_FIFA_World_Cup_finals"
win_tab <- readHTMLTable(winner)
win_tab <- win_tab[[4]] # wiki page changes, so may need to change table number
win <- win_tab[, c(1, 2, 4)]
wins <- gsub(" ", "", win$Winners)
colnames(win) <- c("Year", "Winners", "Runnersup")
loss <- gsub(" ", "", win$Runnersup)
win$Winners <- wins
win$Runnersup <- loss
# Change West Germany to Germany FR for consistency
win$Winners <- gsub("West Germany", "Germany FR",
  win$Winners)
win$Runnersup <- gsub("West Germany", "Germany FR",

```

```
win$Runnersup)  
# Add in the winners and runnerups to our main table  
wc_all <- merge(win, wc, by = "Year")
```