```r
# Meryl Stav final project
# Goals:
# - scrape list of all players in the current NHL season
# - scrape stats from each player's page
# - put scraped information into a NoSQL database
# - build a GUI for user to get information of single player, full team, all
#   players in a certain position
library(RCurl)
library(XML)
library(scrapeR)
library(stringr)
library(mongolite)
library(openxlsx)
library(gWidgets)
library(gWidgetstcltk)

##############################################################################
# Part 1: Set up data frame with player id and team
##############################################################################

# Scrape the team names and possible positions from the first page
# Get the webpage
roster_url <-
  "http://www.nhl.com/ice/playersearch.htm?position=S&season=20152016&pg=1"
roster_webpage <- getURL(roster_url)
# Convert the page to line by line format
tc <- textConnection(roster_webpage)
roster_webpage <- readLines(tc)
close(tc)
# Parse the data
pageTree <- htmlTreeParse(roster_webpage, useInternalNodes = TRUE)
# Get table of teams
teams <- xpathApply(pageTree, "//*/div[@id='teamMenu']/a[@title]", xmlAttrs)
full_team_name <- sapply(teams, "[[", 2)
full_team_name[1] <- "All Teams"
teams <- unlist(xpathApply(pageTree,"//*/div[@id='teamMenu']/a[@title]/div",
```

```
                              xmlAttrs))
teams <- gsub("clb", "cbj", teams)
teams <- gsub("los", "LAK", teams)
teams <- data.frame(full_team_name, team_abv = as.character(teams))
rm(full_team_name)
# Get table of positions
full_position_name <-
  unlist(xpathApply(pageTree,
                    "//*/tr/td/select[@id='position']/option",
                    xmlValue))
full_position_name <- full_position_name[-1]
position_abv <- unlist(xpathApply(pageTree,
                                  "//*/tr/td/select[@id='position']/option",
                                  xmlAttrs))
position_abv <- position_abv[-c(1, 2)]
temp <- grep("Wing", full_position_name)
position_abv[temp] <- str_c(position_abv[temp], "W", sep = "")
positions <- data.frame(full_position_name, position_abv)

rm(full_position_name, position_abv, roster_webpage, temp, tc, pageTree,
   roster_url)

# Now we want to get all of the players in the current season. There are 17
# pages with current players on them.
player_ids_all <- c()
player_names_all <- c()
player_positions_all <- c()
player_teams_all <- c()
for (i in 1:17){
  # This is the base url
  roster_url <- str_c(
    "http://www.nhl.com/ice/playersearch.htm?position=S&season=20152016&pg=",
    i, sep = "")
  roster_webpage <- getURL(roster_url)
  # Convert the page to line by line format
  tc <- textConnection(roster_webpage)
  roster_webpage <- readLines(tc)
  close(tc)
```

```
  pageTree <- htmlTreeParse(roster_webpage, useInternalNodes = TRUE)
  # Get the player names and their positions. Each entry looks like:
  # Doe, John (S)    where John Doe is the name and S is the position
  player_names <- unlist(xpathApply(pageTree, "//*/tbody/tr/td[1]/a",
                                    xmlValue))
  # Get rid of weird characters
  player_names <- gsub("([\n\t])", replacement = "", player_names)
  player_names_attributes <- str_split(player_names, "[()]")
  player_names <- str_trim(sapply(player_names_attributes, "[[", 1))
  player_position <- str_trim(sapply(player_names_attributes, "[[", 2))

  # Get the player id. We will use this to get each individual player's stats
  player_id <- xpathApply(pageTree, "//*/tbody/tr/td[1]/a", xmlAttrs)
  player_page <- sapply(player_id, "[[", 2)
  player_id <- sapply(strsplit(player_page, "=", 1), "[[", 2)

  # Get each player's team
  player_teams <- unlist(xpathApply(pageTree, "//*/tbody/tr/td[2]",
                                    xmlValue))
  # Get rid of weird characters
  player_teams <- gsub("([\n\t])", replacement = "", player_teams)

  # Append to list
  player_ids_all <- c(player_ids_all, player_id)
  player_names_all <- c(player_names_all, player_names)
  player_positions_all <- c(player_positions_all, player_position)
  player_teams_all <- c(player_teams_all, player_teams)
  rm(player_id, player_page, player_names, pageTree, player_names_attributes,
     player_position, roster_webpage, tc, player_teams, roster_url)
}

# Replace abreviations with full names
for (i in 1:nrow(positions)){
  player_positions_all <- gsub(positions$position_abv[i],
                               positions$full_position_name[i],
                               player_positions_all)
}
```

```r
for (i in 1:nrow(teams)){
  player_teams_all <- gsub(str_c("^",teams$team_abv[i],"$"),
                           teams$full_team_name[i],
                           player_teams_all,
                           ignore.case = TRUE)
}

# Create the data frame
df <- data.frame(PlayerId = player_ids_all, Name = player_names_all,
                 Position = player_positions_all, Team = player_teams_all,
                 GP = NA, G = NA, A = NA, P = NA, PlusMinus = NA, PM = NA,
                 PP = NA, SH = NA, GW = NA, S = NA, SP = NA, W = NA, L = NA,
                 OT = NA, GA = NA, SA = NA, SV = NA, SVP = NA, GAA = NA,
                 SO = NA, MIN = NA)

full_name <- c("Games Played", "Goals", "Assists", "Points", "Penalty Minutes",
               "Power Play Goals", "Short Handed Goals", "Game Winning Goals",
               "Shots On Goal", "Shots On Goal Percentage", "Wins", "Loses",
               "Over Times", "Goals Against", "Shots Against", "Saves",
               "Saves Percent", "Goals Against Average", "Shut Outs", "Minutes")
abv <- c("GP", "G", "A", "P", "PM",
         "PP", "SH", "GW",
         "S", "SP", "W", "L",
         "OT", "GA", "SA", "SV",
         "SVP", "GAA", "SO", "MIN")
stats <- matrix(c(abv, full_name), nrow = 2, ncol = 20, byrow = TRUE)

rm(player_teams_all, player_positions_all, player_names_all, player_ids_all,
   abv, full_name)

lengthDF <- nrow(df)

# Fill in the rest of the stats
getPlayerStats <- function(df, rowNum){
  # Get the webpage
  player_url <- "http://www.nhl.com/ice/player.htm?id="
  # Add player id
  player_url <- str_c(player_url, df$PlayerId[rowNum], sep = "")
```

```
  player_webpage <- getURL(player_url)
  # Convert the page to line by line format
  tc <- textConnection(player_webpage)
  player_webpage <- readLines(tc)
  close(tc)
  # Parse for data
  pageTree <- htmlTreeParse(player_webpage, useInternalNodes = TRUE)
  # Get stats
  stats <- unlist(xpathApply(pageTree,
                             "//*/table[1]/tbody/tr[1]/td",
                             xmlValue))
  stats <- gsub(",", "", stats)
  # Remove row header
  stats <- stats[-1]
  # Every position except for goalie follows the same stats headers
  if (df$Position[rowNum] != "Goalie"){
    df[i,5:15] <- as.numeric(stats)
  } else {
    # Goalie stats are out of order so you have to rearrange them
    df[i, 5] <- as.numeric(stats[1])
    df[i, 16:24] <- as.numeric(stats[2:10])
    df[i, 10] <- as.numeric(stats[11])
    df[i, 25] <- as.numeric(stats[12])
  }
  rm(player_url, player_webpage, tc, pageTree, stats)
  return(df)
}

for (i in 1:lengthDF)
{
  df <- getPlayerStats(df, i)
}

rm(i, lengthDF, positions, teams)

################################################################################
# Part 2: Converting the R code to MongoDB
```

```
#########################################################################
# insert data into mongodb
mongoData <- mongo("data")
# add data to mongodb
mongoData$insert(df)
# Get lists to use in GUI:
# Get distinct positions
positions <- mongoData$distinct("Position")
positions <- sort(positions, decreasing = FALSE)
positions <- c("All Positions", positions)
# Get distinct teams
teams <- mongoData$distinct("Team")
teams <- sort(teams, decreasing = FALSE)
teams <- c("All Teams", teams)
# Get distinct player names
fullNames <- mongoData$distinct("Name")
fullNames <- sort(fullNames, decreasing = FALSE)
splitNames <- str_split(fullNames, "[,]")
lastName <- sapply(splitNames, "[[", 1)
lastName <- str_trim(lastName)
lastName <- unique(lastName)
firstName <- sapply(splitNames, "[[", 2)
firstName <- str_trim(firstName)
firstName <- unique(firstName)
fullNames <- c("All Players", fullNames)
lastName <- c("All Players", lastName)
firstName <- c("All Players", firstName)
rm(splitNames)

# Test to see if I can grab a specific player
patrick <- mongoData$find('{"Name":"Kane, Patrick"}',
                          fields = '{"_id" : 0, "PlayerId" : 0}')
# Test to see if I can search for a player
Kane <- mongoData$find('{"Name":{"$regex" : "Kane,*"}}',
                       fields = '{"_id" : 0, "PlayerId" : 0}')
# Test to see if I can grab just the Bruins players
bruins <- mongoData$find('{"Team" : {"$regex" : "Bruins"}}',
                         fields = '{"_id" : 0, "PlayerId" : 0}')
```

```
# Test to see if I can grab just goaliess
goalies <- mongoData$find('{"Position" : {"$regex" : "Goalie"}}',
                          fields = '{"_id" : 0, "PlayerId" : 0}')

rm(bruins, goalies, Kane, patrick)

##############################################################################
# Part 3: Creating a GUI
##############################################################################

## layout a collection of widgets to generate a t.test
byName <- list(type = "ggroup",
               horizontal = FALSE,
               children = list(
                 list(type = "fieldset",
                      columns = 3,
                      label = "Search By Player Name",
                      label.pos = "top",
                      children = list(
                        list(name = "x", label = "Full Name (Last, First)",
                             type = "gcombobox", items = fullNames),
                        list(name = "y", label = "First",
                             type = "gcombobox", items = firstName),
                        list(name = "z", label = "Last",
                             type = "gcombobox", items = lastName)))))

posOrTeam <- list(type = "ggroup",
                  horizontal = TRUE,
                  children = list(
                    list(type = "fieldset",
                         label = "Search By Position",
                         columns = 1,
                         children = list(
                           list(name = "x", type = "gcombobox",
                                items = positions))),
                    list(type = "fieldset",
```

```
                          label = "Search By Team",
                          columns = 1,
                          children = list(
                            list(name = "y", type = "gcombobox",
                                 items = teams)))))
showResults <- function(result, h, criteria){
  dispose(h$obj)
  w <- gwindow("Results from Search",width = 400)
  glabel("", container = w)
  glabel(str_c("Player(s) matching search criteria: ", criteria, sep = ""), container = w)
  glabel("", container = w)
  gtable(result, cont = TRUE, expand = TRUE, drop = FALSE, container = w)
  glabel("", container = w)
  glabel("Score Abreviation breakdown", container = w)
  glabel("", container = w)
  gtable(stats, cont = TRUE, expand = TRUE, drop = FALSE, container = w)
}

# Display the GUI
runTheCode <- function(){
  w <- gwindow("NHL Statistics: Search Window", width = 250)
  g <- ggroup(horizontal = FALSE, container = w)
  fl_1 <- gformlayout(byName, container = g, expand=TRUE)
  bg <- ggroup(container = g)
  addSpace(bg, 20)
  b_1 <- gbutton( "Search By Name", container = bg, handle = function(h,...) {
    out <- svalue(fl_1)
    # if full name used for search
    if (!(grepl("All Players",out$x))) {
      result <- mongoData$find(paste('{"Name":"',out$x,'"}', sep = ""),
                               fields = '{"_id" : 0, "PlayerId" : 0}')
      showResults(result, h, out$x)
    } else {
      # if first name used for search
      if (!(grepl("All Players",out$y))) {
        result <-
          mongoData$find(paste('{"Name":{"$regex":".*, ',out$y,'"}}', sep = ""),
                         fields = '{"_id" : 0, "PlayerId" : 0}')
```

```
      showResults(result, h, str_c("First name: ", out$y, sep = ""))
    } else {
      # if last name used for search
      if (!(grepl("All Players",out$z))) {
        result <-
          mongoData$find(paste('{"Name":{"$regex":"',out$z,',.*"}}', sep = ""),
                         fields = '{"_id" : 0, "PlayerId" : 0}')
        showResults(result, h, str_c("Last name: ", out$z, sep = ""))
      } else {
        # if no name specified for search, print all players
        result <- mongoData$find()
        showResults(result, h, "All players in current season")
      }
    }
  }
})
glabel("Or...", container = g)
addSpace(bg, 20)
fl_2 <- gformlayout(posOrTeam, container = g, expand=TRUE)
bg2 <- ggroup(container = g)
addSpace(bg2, 20)
b_2 <- gbutton("Search By Position and/or Team", container = bg2,
               handle = function(h,...) {
                 out <- svalue(fl_2)
                 # if only position used
                 if (!(grepl("All Positions",out$x)) && grepl("All Teams",out$y)) {
                   result <- mongoData$find(paste('{"Position":"',out$x,'"}', sep = ""),
                                            fields = '{"_id" : 0, "PlayerId" : 0}')
                   showResults(result, h, str_c(out$x, "s", sep = ""))
                 } else {
                   # if only team used
                   if (grepl("All Positions",out$x) && !(grepl("All Teams",out$y))) {
                     result <- mongoData$find(paste('{"Team":"',out$y,'"}', sep = ""),
                                              fields = '{"_id" : 0, "PlayerId" : 0}')
                     showResults(result, h, str_c(out$y, " Players", sep = ""))
                   } else {
                     # if both used
```

```
                    if (!(grepl("All Positions",out$x)) && !(grepl("All Teams",out$y))) {
                      # Test to see if I can grab a specific player
                      result <-
                        mongoData$find(paste('{"Position":"',out$x,'", "Team":"',out$y,'"}',
                                             sep = ""),
                                       fields = '{"_id" : 0, "PlayerId" : 0}')
                      showResults(result, h, str_c(out$x, "s on the ", out$y))
                    } else {
                      result <- mongoData$find()
                      showResults(result, h, "All players in current season")
                    }
                  }
                }
              })
  addSpace(bg, 20)
  glabel("Click either search with default parameters for full 2015-2016 NHL Roster",
         container = g)
  addSpace(bg, 20)
}

# Run this one line to have the first GUI pop up
runTheCode()
```

Created by Pretty R at inside-R.org