On ‘Current Season’ data – there are Two Types of data:

1. Aggregate **NFL Player and NFL Team stats, by Year.**
   1. These stats are easy to pull weekly as they can be extracted from existing sites As-Is. OR
   2. Summate the quarter stats from below to validate the data based on what is current.

I think it would be best to extract As-Is from other sites because this is easiest for consumers to cross-check and find mistakes.

1. Leveraging Play by Play data to provide output for, **Quarters and Redzone Stats.**
   1. Quarter Stats – 1st, 2nd, 3rd, 4th, Overtime (OT), 1st Half (Sum of 1st and 2nd quarters), 2nd Half (Sum of 3rd and 4th quarters)
   2. Redzone Stats – Plays that occurred inside of 20 yards from the End Zone.
   3. Redzone Stats, by Quarter
      * The only way to gather Quarter/Redzone data is to extract from the Play by Play table on [www.profootballreference.com](http://www.profootballreference.com) for the Box Scores for each game.
        + We want a process to extract these data sets for each player for each Quarter from above, AND use the data to determine the NFL Team Stats for each quarter from above.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We began to build a process, as outlined below whereby we 1. WGET the html files for the box scores, 2. Use R to transform and load data into excel 3. Clean up the columns to help filter out the data (This process is highly flawed) 4. And Use Excel VBA to place the data into the columns needed. This process needs a lot of work – If you find another process to use that is easier we are OK with starting over again.

1. **WGET from each NFL Box Score for each NFL Matchup** –

Example of Box Score- <https://www.pro-football-reference.com/boxscores/201710120car.htm>



1. **Run Script below, in R to move into Excel.** We outsourced this to someone on Upwork, I am not sure what goes into this process. This was for the game data for every game since 2007, however we will need something for each week as the games end – could be the same process.

options(stringsAsFactors = FALSE)

library(XML)

#you may change this number

max\_table35\_detail\_columns <- 84

#read file names

urlfiles <- list.files(path=".", pattern="c([0-9]){9}([[:alpha:]]){3}.htm")

for (file\_ind in 1:length(urlfiles)) {

  urlfile <- urlfiles[file\_ind]

  column1 <- substr(urlfile, 2, 5)

  column2 <- substr(urlfile, 6, 10)

  column3 <- substr(urlfile, 11, 13)

  tryCatch({

    #read file

    tables <- readHTMLTable(urlfile)

    #get necessary tables

    table23 <- tables$player\_offense

    table24 <- tables$player\_defense

    table25 <- tables$returns

    table26 <- tables$kicking

    table27 <- tables$home\_starters

    table28 <- tables$vis\_starters

    table35 <- tables$pbp

    #some preprocessing

    table23 <- table23[trimws(table23$Player) != "", ]

    table24 <- table24[trimws(table24$Player) != "", ]

    table25 <- table25[trimws(table25$Player) != "", ]

    table26 <- table26[trimws(table26$Player) != "", ]

    #creating table with all players

    Players <- rbind(table23[, c("Player", "Tm")],

                     table24[, c("Player", "Tm")],

                     table25[, c("Player", "Tm")],

                     table26[, c("Player", "Tm")])

    Players <- unique(Players)

    #processing table 35 separately

    texts <- table35[, 6]

    table35 <- table35[, -6]

    texts\_splitted <- lapply(texts, function(string) {

      if ([is.na](http://is.na/)(string)) {

        return("")

      } else {

        string <- gsub(",", "", string)

        string <- gsub("\\.", "", string)

        string <- gsub(":", "", string)

        string <- gsub("\\(", "", string)

        string <- gsub("\\)", "", string)

        words <- strsplit(string, " ")[[1]]

        new\_words <- c()

        i <- 1

        if (words[i]=="Timeout") {

          new\_words <- words

        } else {

          while (i < length(words)) {

            m <- match(paste(words[i], words[i+1], sep=" "), Players$Player)

            if (i>1 && words[i-1] %in% c("by", "to", "for", "on", "and") && substr(words[i], 1, 1) %in% LETTERS) {

              new\_words <- c(new\_words, paste(words[i], words[i+1], sep=" "), ifelse([is.na](http://is.na/)(m), "NOT FOUND", Players$Tm[m]))

              i <- i+2

            } else {

              m <- match(paste(words[i], words[i+1], sep=" "), Players$Player)

              if ([is.na](http://is.na/)(m)) {

                new\_words <- c(new\_words, words[i])

                i <- i+1

              } else {

                new\_words <- c(new\_words, paste(words[i], words[i+1], sep=" "), ifelse([is.na](http://is.na/)(m), "NOT FOUND", Players$Tm[m]))

                i <- i+2

              }

            }

          }

          if (i <= length(words)) new\_words <- c(new\_words, words[i])

        }

        return(new\_words)

      }

    })

    texts\_splitted <- lapply(texts\_splitted, function(x) {

      x <- x[1:min(length(x), max\_table35\_detail\_columns)]

      return(c(x, rep("", max\_table35\_detail\_columns-length(x))))

    })

    texts\_splitted <- do.call(rbind, texts\_splitted)

    table35 <- cbind(table35, texts\_splitted)

    #additional columns from file name

    table23 <- cbind(column1, column2, column3, table23)

    table24 <- cbind(column1, column2, column3, table24)

    table25 <- cbind(column1, column2, column3, table25)

    table26 <- cbind(column1, column2, column3, table26)

    table27 <- cbind(column1, column2, column3, table27)

    table28 <- cbind(column1, column2, column3, table28)

    table35 <- cbind(column1, column2, column3, table35)

    #saving tables

    if (file\_ind==1) { #write column names

      write.table(rbind(colnames(table23)), "table23.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

      write.table(rbind(colnames(table24)), "table24.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

      write.table(rbind(colnames(table25)), "table25.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

      write.table(rbind(colnames(table26)), "table26.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

      write.table(rbind(colnames(table27)), "table27.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

      write.table(rbind(colnames(table28)), "table28.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

      write.table(rbind(colnames(table35)), "table35.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    }

    write.table(table23, "table23.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    write.table(table24, "table24.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    write.table(table25, "table25.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    write.table(table26, "table26.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    write.table(table27, "table27.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    write.table(table28, "table28.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    write.table(table35, "table35.csv", sep=",", append=TRUE, col.names=FALSE, row.names=FALSE, quote=FALSE)

    # print(paste0("File ", urlfile, " processed (", file\_ind, " out of ", length(urlfiles), ")"))

  }, error=function(err) {

    print(paste0("Error when processing file  ", urlfile, ": ", err))

  })

}

print("Finished")



1. **Data separated into Play Type** We initially wanted to use one large Excel file, however the data was way too large to do in bulk. So we separated the data by play type to help clean – this is not something we are committed to if you have any solutions.

<https://drive.google.com/drive/folders/0ByxL_0Cqf1yNQWVkRjlUQUo3WE0?usp=sharing>

1. **Data was manually cleaned within these files to get to current state:**

\* this process takes a very long time to do manually – we have no way to clean this step through a process. This is the stickiest part of the process.

1. **Once file is cleaned up, use this file to upload** - This is the VBA Macro someone created to sort the data into the columns.



1. **File is created similar to this:** <https://drive.google.com/open?id=198ZZZEDdFujexVCi2q5lqVDzicSSohJ8ZxAwqzvgO_I>
   1. File will not have Positions – manually entered
2. **File should also create NFL Team Stats -** <https://docs.google.com/a/statroute.com/spreadsheets/d/1quFnaEn9i21mWoMMvwWR9r3KAN0xyF5t6oVT4Zf39hU/edit?usp=sharing>