Fantasy Score Calculator

Shane Hylton

11/22/2021

Important Note

These functions take a vector of values that represent the stats of an individual player and they return the player's score for the game. The functions are designed to convert the data to a data frame, then perform the manipulation on the dataframe. They can accept data frames as inputs as well. The data frame provided must contain variable names in columns and players must be represented by rows.

The individual position functions are great for quick calculations. the dataframe function at the end is great for computing an entire team's score. It is a more involved process, loading the data manually. It would be wise to first put the data in a .csv and load it into R.

Running Backs

Note that this function does not compute QB stats for RBs. If a RB throws a pass, the RB function will need to be combined with the QB function for a full statistic. The scoring settings are custom to my Coaches' Corner fantasy league.

```
rb <- function(x){
  df <- data.frame(rbind(x))
  colnames(df) <- c('rush_yd', 'rush_tds', 'rec', 'rec_yd', 'rec_tds', 'fum', 'fum_l', 'two_pt', 'ret_y

  score <- df$rush_yd/10 + df$rush_tds*6 + df$rec*1 + df$rec_yd/10 + df$rec_tds*6 - df$fum*1 - df$fum_l
  return(score)
}

saquon <- c(25, 0, 6, 31, 0, 0, 0, 0, 0, 0)
rb(saquon)
```

```
## [1] 11.6
```

Quarterbacks

```
qb <- function(x){
  df <- data.frame(rbind(x))
  colnames(df) <- c('p_yd', 'p_td', 'int', 'rush_yd', 'rush_tds', 'fum', 'fum_l', 'two_pt')

  score <- df$p_yd/25 + df$p_td*4 - df$int*2 + df$rush_yd/10 + df$rush_tds*6 - df$fum*1 - df$fum_l*2 + d
  return(score)
```

```
}

cam <- c(189, 2, 0, 46, 1, 0, 0, 0)

qb(cam)
```

```
## [1] 26.16
```

Wide Receivers

```
wr <- function(x){
  df <- data.frame(rbind(x))
  colnames(df) <- c('rec', 'rec_yd', 'rec_tds', 'rush_yd', 'rush_tds', 'fum', 'fum_l', 'two_pt', 'ret_y

  score <- df$rush_yd/10 + df$rush_tds*6 + df$rec*1 + df$rec_yd/10 + df$rec_tds*6 - df$fum*1 - df$fum_l
  return(score)
}

tyreek <- c(9, 77, 0, 33, 0, 0, 0, 0, 0, 0)

wr(tyreek)
```

```
## [1] 20
```

Tight Ends

This is the same code as the wide receiver function. Use them interchangeably.

```
te <- function(x){
  df <- data.frame(rbind(x))
  colnames(df) <- c('rec', 'rec_yd', 'rec_tds', 'rush_yd', 'rush_tds', 'fum', 'fum_l', 'two_pt', 'ret_y

  score <- df$rush_yd/10 + df$rush_tds*6 + df$rec*1 + df$rec_yd/10 + df$rec_tds*6 - df$fum*1 - df$fum_l
  return(score)
}

dawson <- c(6, 80, 0, 0, 0, 0, 0, 0, 0, 0)

te(dawson)
```

```
## numeric(0)
```

Kickers

```
k <- function(x){
  df <- data.frame(rbind(x))
  colnames(df) <- c('made_39', 'made_49', 'made_50', 'made_pat', 'miss', 'miss_pat')
```

```
  score <- df$made_39*3 + df$made_49*4 + df$made_50*5 + df$made_pat*1 - df$miss*1 - df$miss_pat*1
  return(score)
}

robbie <- c(1, 2, 0, 3, 0, 0)

k(robbie)
```

```
## [1] 14
```

Defense

If yards are not factored into the points calculation, use def()

If yards are part of the points calculation, use def_yds()

Include defensive and special teams touchdowns in the 'td' column.

```
def <- function(x){
  df <- data.frame(rbind(x))
  colnames(df) <- c('pts_allow', 'sacks', 'int', 'fum_f', 'fum_r', 'td', 'blk_kick', 'safety', 'two_pt_
  score <- 0
  if (df$pts_allow == 0){
    score <- score + 10
  }
  if (df$pts_allow >= 1 & df$pts_allow <= 6){
    score <- score + 7
  }
  if (df$pts_allow >= 7 & df$pts_allow <= 13){
    score <- score + 4
  }
  if (df$pts_allow >= 14 & df$pts_allow <= 20){
    score <- score + 1
  }
  if (df$pts_allow >= 21 & df$pts_allow <= 27){
    score <- score + 0
  }
  if (df$pts_allow >= 28 & df$pts_allow <= 34){
    score <- score -1
  }
  if (df$pts_allow >= 35){
    score <- score -4
  }

  stats <- df$sacks*1 + df$int*2 + df$fum_f*1 + df$fum_r*2 + df$td*6 + df$blk_kick*2 + df$safety*2 + df
  final <- score + stats
  return(final)
}


def_yds <- function(x){
  df <- data.frame(rbind(x))
  colnames(df) <- c('pts_allow', 'yds_allow', 'sacks', 'int', 'fum_f', 'fum_r', 'td', 'blk_kick', 'safe
```

```
  score <- 0
  if (df$pts_allow == 0){
    score <- score + 10
  }
  if (df$pts_allow >= 1 & df$pts_allow <= 6){
    score <- score + 7
  }
  if (df$pts_allow >= 7 & df$pts_allow <= 13){
    score <- score + 4
  }
  if (df$pts_allow >= 14 & df$pts_allow <= 20){
    score <- score + 1
  }
  if (df$pts_allow >= 21 & df$pts_allow <= 27){
    score <- score + 0
  }
  if (df$pts_allow >= 28 & df$pts_allow <= 34){
    score <- score -1
  }
  if (df$pts_allow >= 35){
    score <- score -4
  }
  yds <- 0
  if (df$yds_allow < 100){
    yds <- yds + 10
  }
  if (df$yds_allow >= 100 & df$yds_allow <= 199){
    yds <- yds + 5
  }
  if (df$yds_allow >= 200 & df$yds_allow <= 299){
    yds <- yds + 0
  }
  if (df$yds_allow >= 200 & df$yds_allow <= 399){
    yds <- yds + 0
  }
  if (df$yds_allow >=  400 & df$yds_allow <= 449){
    yds <- yds -2
  }
  if (df$yds_allow >= 450 & df$yds_allow <= 499){
    yds <- yds - 5
  }
  if (df$yds_allow > 500){
    yds <- yds - 5
  }

  stats <- df$sacks*1 + df$int*2 + df$fum_f*1 + df$fum_r*2 + df$td*6 + df$blk_kick*2 + df$safety*2 + df
  final <- score + stats + yds
  return(final)
}

bucs <- c(10, 2, 2, 1, 1, 0, 0, 0, 0)
def(bucs)
```

```
## [1] 13
```

Team Score Calculation

There are two ways to accomplish this: Enter the stats as vectors, use the position function to evaluate, then plug in to team().

OR

Use the team_full() function to evaluate an existing dataframe for the score calculation.

```
team <- function(x){
  df <- data.frame(rbind(x))
  if (ncol(df) != 10){
    return("NA")
  }
  if(ncol(df) == 10){

  colnames(df) <- c('qb', 'rb', 'rb', 'wr', 'wr', 'te', 'wrt', 'wrt', 'k', 'def')
  score <- rowSums(df)
  return(score)
  }
}

cam <- c(189, 2, 0, 46, 1, 0, 0, 0)
saquon <- c(25, 0, 6, 31, 0, 0, 0, 0, 0, 0)
moss <- c(5,0,0,0,0,0,0,0,0,0)
tyreek <- c(9, 77, 0, 33, 0, 0, 0, 0, 0, 0)
dj <- c(5,50,1,0,0,0,0,0,0,0)
dawson <- c(6, 80, 0, 0, 0, 0, 0, 0, 0, 0)
darrel <- c(15, 0, 1, 11, 0,0,0,0,0,0)
godwin <- c(6,65,1,7,0,0,0,0,0,0)
robbie <- c(1, 2, 0, 3, 0, 0)
bucs <- c(10, 2, 2, 1, 1, 0, 0, 0, 0)

anton <- c(qb(cam), rb(saquon), rb(moss), wr(tyreek), wr(dj), te(dawson), rb(darrel), wr(godwin), k(robb

team(anton)
```

```
## [1] "NA"
```

Full Team Calculation using full dataframe

This will be a very wide dataframe with many empty entries. I will load a .csv document instead of manually entering the values in R. For this method, it is not sensible to manually enter the values.

I left a comment containing the skeleton for the dataframe if the values were to be entered manually.

It helps to add a column for position. Not mandatory, but useful.

```
#jake <- data.frame(matrix(ncol = 34, nrow = 0))

#colnames(jake) <- c('rush_yd', 'rush_tds', 'rec', 'rec_yd', 'rec_tds', 'fum', 'fum_l', 'two_pt', 'ret_
```

```r
link <- url('https://raw.githubusercontent.com/st3vejobs/fantasy-score-calculator/main/jake_wk11.csv')
raw <- read.csv(link)

raw[is.na(raw)] <- 0

jake <- raw

colnames(jake) <- c('name', 'rush_yd', 'rush_tds', 'rec', 'rec_yd', 'rec_tds', 'fum', 'fum_l', 'two_pt'

pos <- c('qb', 'rb', 'rb', 'wr','wr','te','wrt','wrt','k','def')

jake <- jake %>% add_column(pos)

jake <- jake %>% relocate(pos, .after = name)



team_full <- function(x){
  df <- x
  stats <- 0
  score <- 0
  for (row in 1:nrow(df)){

  stats <- stats + df[row, ]$rush_yd/10 + df[row, ]$rush_tds*6 + df[row, ]$rec*1 + df[row, ]$rec_yd/10 +

}
    ##defense

  row <- which(df$pos == 'def')
  if (df[row, ]$pts_allow == 0){
    score <- score + 10
  }
  if (df[row, ]$pts_allow >= 1 & df[row, ]$pts_allow <= 6){
    score <- score + 7
  }
  if (df[row, ]$pts_allow >= 7 & df[row, ]$pts_allow <= 13){
    score <- score + 4
  }
  if (df[row, ]$pts_allow >= 14 & df[row, ]$pts_allow <= 20){
    score <- score + 1
  }
  if (df[row, ]$pts_allow >= 21 & df[row, ]$pts_allow <= 27){
    score <- score + 0
  }
  if (df[row, ]$pts_allow >= 28 & df[row, ]$pts_allow <= 34){
    score <- score -1
  }
  if (df[row, ]$pts_allow >= 35){
    score <- score -4
  }

  score <- score + df[row, ]$sacks*1 + df[row, ]$def_int*2 + df[row, ]$fum_f*1 + df[row, ]$fum_r*2 + df
```

```
  final <- score + stats
  return(final)
}


team_full(jake)
```

```
## [1] 139.1
```

```
for (row in 1:nrow(jake)){
if(jake[row, ]$pos == 'def'){
  paste('jake$pos')
}
}
```