

homework6

Ashley Spirrison

2023-10-24

#Question 1

```
setwd("C:/Users/ashle/OneDrive - Vanderbilt/BIOS 6301-Fall 2023/football23")
```

#creating path

```
path <- "C:/Users/ashle/OneDrive - Vanderbilt/BIOS 6301-Fall 2023/football23"
```

#creating function

```
ffvalues <- function(path=path, file='outfile.csv', nTeams=12, cap=200, posReq=c(qb=1, rb=2, wr=3, te=1,
                                         points=c(fg=4, xpt=1, pass_yds=1/25, pass_tds=4, pass_ints=-2,
                                                    rush_yds=1/10, rush_tds=6, fumbles=-2, rec_yds=1/20, rec_tds=6))) {
```

#loading data and creating new columns

```
proj_k23 <- read.csv(paste0(path, "/proj_k23.csv"))
proj_k23$pos <- "k"
proj_qb23 <- read.csv(paste0(path, "/proj_qb23.csv"))
proj_qb23$pos <- "qb"
proj_rb23 <- read.csv(paste0(path, "/proj_rb23.csv"))
proj_rb23$pos <- "rb"
proj_te23 <- read.csv(paste0(path, "/proj_te23.csv"))
proj_te23$pos <- "te"
proj_wr23 <- read.csv(paste0(path, "/proj_wr23.csv"))
proj_wr23$pos <- "wr"
```

Defining all expected column names

```
all_columns <- c("PlayerName", "Team", "fg", "fga", "xpt", "fpts", "pos", "pass_att", "pass_cmp", "pass_yds", "pass_tds", "rush_yds", "rush_tds", "fumbles", "rec_yds", "rec_tds")
```

#Adding missing columns with default values

```
add_missing_columns <- function(df, all_columns) {
  missing_columns <- setdiff(all_columns, colnames(df))
  for (col in missing_columns) {
    df[[col]] <- ifelse(col %in% c("fg", "fga", "xpt", "fpts", "pass_att", "pass_cmp", "pass_yds", "pass_tds", "rush_yds", "rush_tds", "fumbles", "rec_yds", "rec_tds"),
                        NA, 0)
  }
  return(df)
}
```

Adding missing columns to each data frame

```
proj_k23 <- add_missing_columns(proj_k23, all_columns)
proj_qb23 <- add_missing_columns(proj_qb23, all_columns)
```

```

proj_rb23 <- add_missing_columns(proj_rb23, all_columns)
proj_te23 <- add_missing_columns(proj_te23, all_columns)
proj_wr23 <- add_missing_columns(proj_wr23, all_columns)

#Combining data frames
df <- rbind(proj_k23, proj_qb23, proj_rb23, proj_te23, proj_wr23)

#Calculating dollar values
calculate_value <- function(df, pos, points) {
  player_points <- rowSums(df[, names(points)] * points)
  total_points <- sum(player_points)
  if (total_points != 0) {
    df$points <- player_points
    df$value <- pos * player_points
  } else {
    df$points <- 0
    df$value <- 0
  }
  return(df)
}

#Calculating value for each player in the combined data frame
df <- calculate_value(df, posReq["k"], points)
df <- calculate_value(df, posReq["qb"], points)
df <- calculate_value(df, posReq["rb"], points)
df <- calculate_value(df, posReq["te"], points)
df <- calculate_value(df, posReq["wr"], points)

#Ordering by value
df <- df[order(-df$value), ]

#Writing to CSV file
write.csv(df, file = file, row.names = FALSE)

return(df)
}

#1.Calling the ffvalues function
x1 <- ffvalues('.')

# Counting the number of players worth more than $20
players_over_20 <- sum(x1$value > 20)

# Identifying the 15th most valuable running back (rb)
rb_players <- subset(x1, pos == 'rb')
rb_players <- rb_players[order(-rb_players$value), ]
fifteenth_rb <- rb_players[15, "PlayerName"]

print(players_over_20)

```

```
## [1] 490
```

```
print(fifteenth_rb)
```

```
## [1] "AJ Dillon"
```

```
#2. calling the ffvalues function a second time
```

```
x2 <- ffvalues(getwd(), '16team.csv', nTeams=16, cap=150)
```

```
#Summing the number of players over 20
```

```
num_players_over_20 <- sum(x2$value > 20)
```

```
#Summing the top 40 wrs
```

```
num_wr_top_40 <- sum(x2$pos == 'wr')
```

```
print(num_players_over_20)
```

```
## [1] 490
```

```
print(num_wr_top_40)
```

```
## [1] 256
```

```
#3. Calling and updating the ffvalues function
```

```
x3 <- ffvalues('.', 'qbheavy.csv', posReq=c(qb=2, rb=2, wr=3, te=1, k=0),  
             points=c(fg=0, xpt=0, pass_yds=1/25, pass_tds=6, pass_ints=-2,  
                      rush_yds=1/10, rush_tds=6, fumbles=-2, rec_yds=1/20, rec_tds=6))
```

```
#Summing the number of players over 20
```

```
num_players_over_20 <- sum(x3$value > 20)
```

```
#Summing top 30 qbs
```

```
num_qb_top_30 <- sum(x3$pos == 'qb') # Assuming 'pos' is the column that specifies the position
```

```
print(num_players_over_20)
```

```
## [1] 405
```

```
print(num_qb_top_30)
```

```
## [1] 86
```

```
#Question 2
```

```
#Loading haart from github
```

```
url_to_raw_csv_file <- "https://raw.githubusercontent.com/couthcommander/Bios6301/main/datasets/haart.csv"
```

```
#Reading the lines of haart into R
```

```
lines <- readLines(url_to_raw_csv_file)
```

```
## Warning in readLines(url_to_raw_csv_file): incomplete final line found on
```

```
## 'https://raw.githubusercontent.com/couthcommander/Bios6301/main/datasets/haart.csv'
```

```

#Removing the incomplete final line if present
if (length(lines) > 0 && nchar(lines[length(lines)]) == 0) {
  lines <- lines[-length(lines)]
}

#Creating a text connection and reading the data using read.csv
haart_data <- read.csv(text = lines, header = TRUE)

#Checking the first few rows of the dataset
head(haart_data)

```

```

##   male age aids cd4baseline logvl  weight hemoglobin  init.reg init.date
## 1    1  25   0         NA      NA      NA          NA 3TC,AZT,EFV   7/1/03
## 2    1  49   0        143      NA 58.0608         11 3TC,AZT,EFV  11/23/04
## 3    1  42   1        102      NA 48.0816          1 3TC,AZT,EFV   4/30/03
## 4    0  33   0        107      NA 46.0000         NA 3TC,AZT,NVP   3/25/06
## 5    1  27   0         52      4      NA          NA 3TC,D4T,EFV   9/1/04
## 6    0  34   0        157      NA 54.8856         NA 3TC,AZT,NVP  12/2/03
##   last.visit death date.death
## 1    2/26/07    0      <NA>
## 2    2/22/08    0      <NA>
## 3   11/21/05    1   1/11/06
## 4     5/5/06    1   5/7/06
## 5   11/13/07    0      <NA>
## 6    2/28/08    0      <NA>

```

```

#1. Converting 'init.date' into a usable date format
haart_data$init.date <- as.Date(haart_data$init.date, format = "%m/%d/%y")
class(haart_data$init.date)

```

```
## [1] "Date"
```

```

# Using the table command to display the counts of the year from 'init.date'
year_counts <- table(format(haart_data$init.date, "%Y"))

```

```

# Printing the counts of the year from 'init.date'
print(year_counts)

```

```

##
## 1998 2000 2001 2002 2003 2004 2005 2006 2007
##    1    5   17   60  270  292  207  104   44

```

```

# Converting 'death.date' to date format, considering NA values
haart_data$date.death <- as.Date(haart_data$date.death, format = "%m/%d/%y", na.rm = TRUE)

```

```

# Creating an indicator variable 'death_within_1_year' (1 for death within 1 year, 0 otherwise)
haart_data$death_within_1_year <- ifelse(is.na(haart_data$date.death), NA, as.integer(difftime(haart_da

```

```

# Counting the number of observations that died within the first year
deaths_within_1_year <- sum(haart_data$death_within_1_year, na.rm = TRUE)

```

```

# Printing the number of observations that died within the first year
print(deaths_within_1_year)

```

```
## [1] 92
```

```
#Converting last.visit to usable format
haart_data$last.visit <- as.Date(haart_data$last.visit, format = "%m/%d/%y")

#3. Calculating the follow-up time in days
haart_data$followup_time <- pmin(difftime(pmax(haart_data$last.visit, haart_data$date.death, na.rm = TRUE)))

# Printing the quantile for the follow-up time
print(quantile(haart_data$followup_time))
```

```
## Time differences in days
##      0%   25%   50%   75%  100%
##      0.0 329.5 365.0 365.0 365.0
```

```
#4. Creating an indicator variable for loss to follow-up
haart_data$loss_to_followup <- ifelse(is.na(haart_data$date.death) & difftime(haart_data$last.visit, haart_data$date.death) > 365, 1, 0)

# Counting the number of records that are lost to follow-up
lost_to_followup_count <- sum(haart_data$loss_to_followup)

# Printing the number of records that are lost to follow-up
print(lost_to_followup_count)
```

```
## [1] 710
```

```
# 5. Converting 'init.reg' to a factor to create indicator variables for each unique drug
haart_data$init.reg <- as.factor(haart_data$init.reg)

# Using model.matrix to create indicator variables for each unique drug
regimen_indicator <- model.matrix(~ 0 + init.reg, data = haart_data)

# Appending the indicator variables to the database as new columns
haart_data <- cbind(haart_data, regimen_indicator)

# Identifying which drug regimen appears over 100 times
regimen_counts <- colSums(regimen_indicator)
drugs_over_100_times <- names(regimen_counts[regimen_counts > 100])

# Printing the drug regimen found over 100 times
print(drugs_over_100_times)
```

```
## [1] "init.reg3TC,AZT,EFV" "init.reg3TC,AZT,NVP"
```

```
# 6. Reading the additional dataset 'haart2.csv'
haart2_data <- read.csv("haart2.csv")

# Selecting only the relevant columns from each dataset
relevant_haart_data <- haart_data[, 1:12]
relevant_haart2_data <- haart2_data

# Appending the relevant data from haart2 to large_data
```

```
complete_data <- rbind(relevant_haart_data, relevant_haart2_data)
```

```
# Showing the first five records of the complete dataset
head(complete_data, 5)
```

```
##   male age aids cd4baseline logvl  weight hemoglobin  init.reg  init.date
## 1    1  25   0         NA    NA      NA          NA 3TC,AZT,EFV 2003-07-01
## 2    1  49   0        143    NA  58.0608         11 3TC,AZT,EFV 2004-11-23
## 3    1  42   1        102    NA  48.0816          1 3TC,AZT,EFV 2003-04-30
## 4    0  33   0        107    NA  46.0000         NA 3TC,AZT,NVP 2006-03-25
## 5    1  27   0         52     4      NA          NA 3TC,D4T,EFV 2004-09-01
##   last.visit death date.death
## 1 2007-02-26     0      <NA>
## 2 2008-02-22     0      <NA>
## 3 2005-11-21     1 2006-01-11
## 4 2006-05-05     1 2006-05-07
## 5 2007-11-13     0      <NA>
```

```
# Showing the last five records of the complete dataset
tail(complete_data, 5)
```

```
##   male      age aids cd4baseline  logvl  weight hemoglobin  init.reg
## 1000    0 40.00000    1      131     NA  46.2672          8 3TC,D4T,NVP
## 1100    0 27.00000    0      232     NA      NA          NA 3TC,AZT,NVP
## 2100    1 38.72142    0      170     NA  84.0000          NA 3TC,AZT,NVP
## 3100    1 23.00000   NA      154  3.995635  65.5000         14 3TC,DDI,EFV
## 4100    0 31.00000    0      236     NA  45.8136          NA 3TC,D4T,NVP
##   init.date last.visit death date.death
## 1000 2003-07-03 2008-02-29     0      <NA>
## 1100 0012-01-03 0001-05-04     0      <NA>
## 2100      <NA>      <NA>     0      <NA>
## 3100      <NA>      <NA>     0      <NA>
## 4100 0012-03-03 0010-11-07     0      <NA>
```