Bios 6301: Assignment 6

Haley Yaremych

Due Tuesday, 26 October, 1:00 PM

$5^{n=day}$ points taken off for each day late.

40 points total.

Submit a single knitr file (named `homework6.rmd`), along with a valid PDF output file. Inside the file, clearly indicate which parts of your responses go with which problems (you may use the original homework document as a template). Add your name as `author` to the file's metadata section. Raw R code/output or word processor files are not acceptable.

Failure to name file `homework6.rmd` or include author name may result in 5 points taken off.

Question 1

16 points

Obtain a copy of the football-values lecture. Save the five 2021 CSV files in your working directory.

Modify the code to create a function. This function will create dollar values given information (as arguments) about a league setup. It will return a data.frame and write this data.frame to a CSV file. The final data.frame should contain the columns 'PlayerName', 'pos', 'points', 'value' and be ordered by value descendingly. Do not round dollar values.

Note that the returned data.frame should have `sum(posReq)*nTeams` rows.

Define the function as such (10 points):

```
# path: directory path to input files
# file: name of the output file; it should be written to path
# nTeams: number of teams in league
# cap: money available to each team
# posReq: number of starters for each position
# points: point allocation for each category
ffvalues <- function(path, file='outfile.csv',
                     nTeams=12, cap=200,
                     posReq=c(qb=1, rb=2, wr=3, te=1, k=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)) {
  ## read in CSV files
  setwd(path)
  files = list.files()
  csvs = files[grep("21.csv", files)]
  dfs = lapply(csvs, read.csv)

  dfs[[grep("k21", csvs)]]$pos = "k"
  dfs[[grep("qb21", csvs)]]$pos = "qb"
```

```
dfs[[grep("rb21", csvs)]]$pos = "rb"
dfs[[grep("te21", csvs)]]$pos = "te"
dfs[[grep("wr21", csvs)]]$pos = "wr"

df = dplyr::bind_rows(dfs)

## calculate dollar values
# points
df$points = NA

for (i in 1:nrow(df)){
  sum = 0

  for (p in 1:length(points)){
    if (!is.na(df[i, names(points)[p]]*points[p])) {
      sum = sum + df[i, names(points)[p]]*points[p]
    }
  }

  df[i, 'points'] = sum

}

df2 = df[order(df[,'points'], decreasing=TRUE),]

# marginal points
df2$marg = NA

k.ix <- which(df2[,'pos']=='k')
if (posReq['k'] != 0){
  df2[k.ix, 'marg'] <- df2[k.ix,'points'] - df2[k.ix[nTeams*posReq['k']],'points']
} else {df2 = df2[-k.ix,]}

qb.ix <- which(df2[,'pos']=='qb')
if (posReq['qb'] != 0) {
  df2[qb.ix, 'marg'] <- df2[qb.ix,'points'] - df2[qb.ix[nTeams*posReq['qb']],'points']
} else {df2 = df2[-qb.ix,]}

rb.ix <- which(df2[,'pos']=='rb')
if (posReq['rb'] != 0) {
  df2[rb.ix, 'marg'] <- df2[rb.ix,'points'] - df2[rb.ix[nTeams*posReq['rb']],'points']
} else {df2 = df2[-rb.ix,]}

te.ix <- which(df2[,'pos']=='te')
if (posReq['te'] != 0) {
  df2[te.ix, 'marg'] <- df2[te.ix,'points'] - df2[te.ix[nTeams*posReq['te']],'points']
} else {df2 = df2[-te.ix,]}

 wr.ix <- which(df2[,'pos']=='wr')
 if (posReq['wr'] != 0) {
  df2[wr.ix, 'marg'] <- df2[wr.ix,'points'] - df2[wr.ix[nTeams*posReq['wr']],'points']
} else {df2 = df2[-wr.ix,]}
```

```
  positive <- df2[df2[,'marg'] >= 0,]

  # dollar values
  positive$value = NA
  positive$value <- (nTeams*cap-nrow(positive)) * positive[,'marg'] / sum(positive[,'marg']) + 1

  # to return
  final = positive[,c("PlayerName", "pos", "points", "value")]
  final = final[order(final[,'value'], decreasing=TRUE),]

  ## save dollar values as CSV file
  write.csv(final, file.path(path, file), row.names = FALSE)

  ## return data.frame with dollar values
  return(final)
}
```

1. Call `x1 <- ffvalues('.')`
    1. How many players are worth more than $20? (1 point)
    2. Who is 15th most valuable running back (rb)? (1 point)

```
x1 = ffvalues('.')

nrow(x1[which(x1$value > 20),])
```

```
## [1] 44
```

```
rbs = x1[which(x1$pos=='rb'),]
rbs[15, 'PlayerName']
```

```
## [1] "Chris Carson"
```

1. Call `x2 <- ffvalues(getwd(), '16team.csv', nTeams=16, cap=150)`
    1. How many players are worth more than $20? (1 point)
    2. How many wide receivers (wr) are in the top 40? (1 point)

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

nrow(x2[which(x2$value > 20),])
```

```
## [1] 44
```

```
top40 = x2[c(1:40),]
nrow(top40[which(top40$pos=='wr'),])
```

```
## [1] 8
```

1. Call:

```
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))
```

    1. How many players are worth more than $20? (1 point)
    2. How many quarterbacks (qb) are in the top 30? (1 point)

```
nrow(x3[which(x3$value > 20),])
```

```
## [1] 47
```

```
top30 = x3[c(1:30),]
nrow(top30[which(top30$pos=='qb'),])
```

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

Question 2

24 points

Import the HAART dataset (`haart.csv`) from the GitHub repository into R, and perform the following manipulations: (4 points each)

```
haart = read.csv("~/Documents/Fall 2021/Statistical Computing/datasets/haart.csv")
```

1. Convert date columns into a usable (for analysis) format. Use the `table` command to display the counts of the year from `init.date`.

```
haart$init.date = as.POSIXct(haart$init.date, format = "%m/%d/%y")
haart$last.visit = as.POSIXct(haart$last.visit, format = "%m/%d/%y")
haart$date.death = as.POSIXct(haart$date.death, format = "%m/%d/%y")

table(format(haart$init.date, format="%Y"))
```

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

2. Create an indicator variable (one which takes the values 0 or 1 only) to represent death within 1 year of the initial visit. How many observations died in year 1?

```
diffs = difftime(haart$date.death, haart$init.date, units="weeks")
ind = as.numeric(diffs <= 52)

length(which(ind==1))
```

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

3. Use the `init.date`, `last.visit` and `death.date` columns to calculate a followup time (in days), which is the difference between the first and either the last visit or a death event (whichever comes first). If these times are longer than 1 year, censor them (this means if the value is above 365, set followup to 365). Print the quantile for this new variable.

```
diff.visits = difftime(haart$last.visit, haart$init.date, units="days")
diff.death = difftime(haart$date.death, haart$init.date, units="days")

followup = numeric(length(diff.visits))

for (i in 1:length(followup)){

  if (is.na(diff.death[i])){
    followup[i] = diff.visits[i]

  } else if (is.na(diff.visits[i])){
    followup[i] = diff.death[i]
```

```
  } else if (diff.visits[i] < diff.death[i]){
    followup[i] = diff.visits[i]

  } else {followup[i] = diff.death[i]}
}

for (i in 1:length(followup)) {
  if (followup[i] > 365) {followup[i] = 365}
}

quantile(followup)
```

```
##       0%      25%      50%      75%     100%
##   0.0000 320.7188 365.0000 365.0000 365.0000
```

4. Create another indicator variable representing loss to followup; this means the observation is not known to be dead but does not have any followup visits after the first year. How many records are lost-to-followup?

```
lost = numeric(length(followup))

for (i in 1:length(lost)){
  if (is.na(haart$date.death[i]) & diff.visits[i]<365){
    lost[i] = 1
  } else lost[i] = 0
}

table(lost)
```

```
## lost
##   0   1
## 827 173
```

```
# 173 records were lost to followup.
```

5. Recall our work in class, which separated the `init.reg` field into a set of indicator variables, one for each unique drug. Create these fields and append them to the database as new columns. Which drug regimen are found over 100 times?

```
init.reg <- as.character(haart[,'init.reg'])
haart[['init.reg_list']] <- strsplit(init.reg, ",")

(all_drugs <- unique(unlist(haart$init.reg_list)))
```

```
##  [1] "3TC" "AZT" "EFV" "NVP" "D4T" "ABC" "DDI" "IDV" "LPV" "RTV" "SQV" "FTC"
## [13] "TDF" "DDC" "NFV" "T20" "ATV" "FPV"
```

```
length(all_drugs)
```

```
## [1] 18
```

```
reg_drugs <- matrix(FALSE, nrow=nrow(haart), ncol=length(all_drugs))
for(i in seq_along(all_drugs)) {
  reg_drugs[,i] <- sapply(haart$init.reg_list, function(x) all_drugs[i] %in% x)
}
reg_drugs <- data.frame(reg_drugs)
names(reg_drugs) <- all_drugs
```

```
haart <- cbind(haart, reg_drugs)

regs = unique(haart$init.reg)
for (i in 1:length(regs)){
  if (length(which(haart$init.reg == regs[i])) > 100){print(regs[i])}
}
```

```
## [1] "3TC,AZT,EFV"
## [1] "3TC,AZT,NVP"
```

6. The dataset `haart2.csv` contains a few additional observations for the same study. Import these and append them to your master dataset (if you were smart about how you coded the previous steps, cleaning the additional observations should be easy!). Show the first five records and the last five records of the complete (and clean) data set.

```
haart2 = read.csv("~/Documents/Fall 2021/Statistical Computing/datasets/haart2.csv")

haart2$init.date = as.POSIXct(haart2$init.date, format = "%m/%d/%y")
haart2$last.visit = as.POSIXct(haart2$last.visit, format = "%m/%d/%y")
haart2$date.death = as.POSIXct(haart2$date.death, format = "%m/%d/%y")

init.reg <- as.character(haart2[,'init.reg'])
haart2[['init.reg_list']] <- strsplit(init.reg, ",")

reg_drugs <- matrix(FALSE, nrow=nrow(haart2), ncol=length(all_drugs))
for(i in seq_along(all_drugs)) {
  reg_drugs[,i] <- sapply(haart2$init.reg_list, function(x) all_drugs[i] %in% x)
}
reg_drugs <- data.frame(reg_drugs)
names(reg_drugs) <- all_drugs

haart2 <- cbind(haart2, reg_drugs)

master = rbind(haart,haart2)
head(master, 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 init.reg_list  3TC   AZT   EFV   NVP   D4T   ABC
## 1 2007-02-26     0       <NA> 3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE
## 2 2008-02-22     0       <NA> 3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE
## 3 2005-11-21     1 2006-01-11 3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE
## 4 2006-05-05     1 2006-05-07 3TC, AZT, NVP TRUE  TRUE FALSE  TRUE FALSE FALSE
## 5 2007-11-13     0       <NA> 3TC, D4T, EFV TRUE FALSE  TRUE FALSE  TRUE FALSE
##     DDI   IDV   LPV   RTV   SQV   FTC   TDF   DDC   NFV   T20   ATV   FPV
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 5 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
tail(master, 5)
```

```
##      male      age aids cd4baseline    logvl  weight hemoglobin    init.reg
## 1000    0 40.00000    1         131       NA 46.2672          8 3TC,D4T,NVP
## 1001    0 27.00000    0         232       NA      NA         NA 3TC,AZT,NVP
## 1002    1 38.72142    0         170       NA 84.0000         NA 3TC,AZT,NVP
## 1003    1 23.00000   NA         154 3.995635 65.5000         14 3TC,DDI,EFV
## 1004    0 31.00000    0         236       NA 45.8136         NA 3TC,D4T,NVP
##       init.date last.visit death date.death init.reg_list  3TC   AZT   EFV
## 1000 2003-07-03 2008-02-29     0       <NA> 3TC, D4T, NVP TRUE FALSE FALSE
## 1001 2003-12-01 2004-01-05     0       <NA> 3TC, AZT, NVP TRUE  TRUE FALSE
## 1002 2002-09-26 2004-03-29     0       <NA> 3TC, AZT, NVP TRUE  TRUE FALSE
## 1003 2007-01-31 2007-04-16     0       <NA> 3TC, DDI, EFV TRUE FALSE  TRUE
## 1004 2003-12-03 2007-10-11     0       <NA> 3TC, D4T, NVP TRUE FALSE FALSE
##        NVP   D4T   ABC   DDI   IDV   LPV   RTV   SQV   FTC   TDF   DDC   NFV
## 1000  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1001  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1002  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1003 FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1004  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##        T20   ATV   FPV
## 1000 FALSE FALSE FALSE
## 1001 FALSE FALSE FALSE
## 1002 FALSE FALSE FALSE
## 1003 FALSE FALSE FALSE
## 1004 FALSE FALSE FALSE
```