Bios 6301: Assignment 6

Haoyang Yi

10/17/2022

Due Tuesday, 25 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 orderd 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
path = '.'
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
  k = read.csv(paste0(path,'/proj_k21.csv'))
  qb = read.csv(paste0(path,'/proj_qb21.csv'))
  rb = read.csv(paste0(path,'/proj_rb21.csv'))
  te = read.csv(paste0(path,'/proj_te21.csv'))
  wr = read.csv(paste0(path,'/proj_wr21.csv'))
cols <- unique(c(names(k), names(qb), names(rb), names(te), names(wr)))
k[,'pos'] <- 'k'
qb[,'pos'] <- 'qb'
```

```r
rb[,'pos'] <- 'rb'
te[,'pos'] <- 'te'
wr[,'pos'] <- 'wr'

# append 'pos' to unique column list
cols <- c(cols, 'pos')

# create common columns in each data.frame
# initialize values to zero
k[,setdiff(cols, names(k))] <- 0
qb[,setdiff(cols, names(qb))] <- 0
rb[,setdiff(cols, names(rb))] <- 0
te[,setdiff(cols, names(te))] <- 0
wr[,setdiff(cols, names(wr))] <- 0

# combine data.frames by row, using consistent column order
x <- rbind(k[,cols], qb[,cols], rb[,cols], te[,cols], wr[,cols])
  ## calculate dollar values
x[,'p_fg'] <- x[,'fg']*points[1]
x[,'p_xpt'] <- x[,'xpt']*points[2]
x[,'p_pass_yds'] <- x[,'pass_yds']*points[3]
x[,'p_pass_tds'] <- x[,'pass_tds']*points[4]
x[,'p_pass_ints'] <- x[,'pass_ints']*points[5]
x[,'p_rush_yds'] <- x[,'rush_yds']*points[6]
x[,'p_rush_tds'] <- x[,'rush_tds']*points[7]
x[,'p_fumbles'] <- x[,'fumbles']*points[8]
x[,'p_rec_yds'] <- x[,'rec_yds']*points[9]
x[,'p_rec_tds'] <- x[,'rec_tds']*points[10]

# sum selected column values for every row
# this is total fantasy points for each player
x[,'points'] <- rowSums(x[,grep("^p_", names(x))])
  ## save dollar values as CSV file
  ## return data.frame with dollar values
x2 <- x[order(x[,'points'], decreasing=TRUE),]
# determine the row indeces for each position
k.ix <- which(x2[,'pos']=='k')
qb.ix <- which(x2[,'pos']=='qb')
rb.ix <- which(x2[,'pos']=='rb')
te.ix <- which(x2[,'pos']=='te')
wr.ix <- which(x2[,'pos']=='wr')

kreq = posReq[5]*nTeams
qbreq = posReq[1]*nTeams
rbreq = posReq[2]*nTeams
tereq = posReq[3]*nTeams
wrreq = posReq[4]*nTeams
# calculate marginal points by subtracting "baseline" player's points
if(kreq > 0) {x2[k.ix, 'marg'] <- x2[k.ix,'points'] - x2[k.ix[kreq],'points']}
x2[qb.ix, 'marg'] <- x2[qb.ix,'points'] - x2[qb.ix[qbreq],'points']
x2[rb.ix, 'marg'] <- x2[rb.ix,'points'] - x2[rb.ix[rbreq],'points']
x2[te.ix, 'marg'] <- x2[te.ix,'points'] - x2[te.ix[tereq],'points']
x2[wr.ix, 'marg'] <- x2[wr.ix,'points'] - x2[wr.ix[wrreq],'points']
```

```
# create a new data.frame subset by non-negative marginal points
x3 <- x2[x2[,'marg'] >= 0,]
x3 = x3[is.na(x3$marg)==F,]
# re-order by marginal points
x3 <- x3[order(x3[,'marg'], decreasing=TRUE),]

# reset the row names
rownames(x3) <- NULL

# calculation for player value
x3[,'value'] <- (nTeams*cap-nrow(x3)) * x3[,'marg'] / sum(x3[,'marg']) + 1

# create a data.frame with more interesting columns
x4 <- x3[,c('PlayerName','pos','points','marg','value')]
write.table(x4,file=file)
return(x4)
}
```

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('.')
length(which(x1$value>20))# 44 players are worth more than $20
```

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

```
x1$PlayerName[x1$pos=='rb'][15] # David Montgomery
```

```
## [1] "David Montgomery"
```

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)
length(which(x2$value>20))# 48 players are worth more than $20
```

```
## [1] 48
```

```
length(which(x2[1:40,]$pos=='wr')) # 2 receivers are in the top 40
```

```
## [1] 2
```

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))
length(which(x3$value>20))# 42 players are worth more than $20
```

```
## [1] 42
```

```
length(which(x3[1:30,]$pos=='qb')) # 11 quarterbacks are in the top 30
```

```
## [1] 11
```

```
1.  How many players are worth more than $20? (1 point)
```

```
1.  How many quarterbacks (qb) are in the top 30? (1 point)
```

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('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$date.death = as.POSIXct(haart$date.death,format = '%m/%d/%y')
haart$init.year = substr(haart$init.date,1,4)
table(haart$init.year)
```

```
##
## 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?

```
haart$time_to_death = haart$date.death-haart$init.date
haart$deathin_1year = 0
haart$deathin_1year[haart$time_to_death<=365] = 1
table(haart$deathin_1year) # 92 observations died in year 1
```

```
##
##   0   1
## 908  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.

```
haart$last.visit = as.POSIXct(haart$last.visit,format = '%m/%d/%y')
haart$followup = difftime(pmin(haart$last.visit,haart$date.death,na.rm = T),haart$init.date,units = 'da
haart$followup[haart$followup>365] = 365
quantile(haart$followup,na.rm = T)
```

```
## Time differences in days
##       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?

```
haart$loss_followup = 0
haart$loss_followup[haart$death==0 & haart$followup<365]=1
table(haart$loss_followup) # 173 patients are lost-to-followup
```

```
##
```

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

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"
```

```
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_merged <- cbind(haart, reg_drugs)
data.frame(drug_name = all_drugs, times_over_100 = colSums(reg_drugs)>100,row.names = 1)
```

```
##     times_over_100
## 3TC           TRUE
## AZT           TRUE
## EFV           TRUE
## NVP           TRUE
## D4T           TRUE
## ABC          FALSE
## DDI          FALSE
## IDV          FALSE
## LPV          FALSE
## RTV          FALSE
## SQV          FALSE
## FTC          FALSE
## TDF          FALSE
## DDC          FALSE
## NFV          FALSE
## T20          FALSE
## ATV          FALSE
## FPV          FALSE
```

```
# 3TC AZT EFV NVP D4T are found over 100 times.
```

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('haart2.csv')
haart2$init.date = as.POSIXct(haart2$init.date,format = '%m/%d/%y')
haart2$date.death = as.POSIXct(haart2$date.death,format = '%m/%d/%y')
haart2$init.year = substr(haart2$init.date,1,4)
haart2$last.visit = as.POSIXct(haart2$last.visit,format = '%m/%d/%y')
```

```r
haart_comp = rbind(haart[,1:13],haart2)
head(haart_comp,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.year
## 1 2007-02-26     0       <NA>      2003
## 2 2008-02-22     0       <NA>      2004
## 3 2005-11-21     1 2006-01-11      2003
## 4 2006-05-05     1 2006-05-07      2006
## 5 2007-11-13     0       <NA>      2004
```

```r
tail(haart_comp,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.year
## 1000 2003-07-03 2008-02-29     0       <NA>      2003
## 1001 2003-12-01 2004-01-05     0       <NA>      2003
## 1002 2002-09-26 2004-03-29     0       <NA>      2002
## 1003 2007-01-31 2007-04-16     0       <NA>      2007
## 1004 2003-12-03 2007-10-11     0       <NA>      2003
```