Bios 6301: Assignment 6

Yan Yan

Due Tuesday, 30 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 2018 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
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
  ## path <- "~/Desktop/football-values/2018"
  #cat(nTeams," ",cap,"\n")
  #cat("Reading csv files\n")
  position <- c("qb","rb","wr","te","k")
  files <- paste("proj_",position,"18.csv",sep="")
  filelist <- paste(path,files,sep="/")
  data <- lapply(filelist,read.csv)
  uniquenames <- unique(unlist(lapply(data,names)))

  data1 <- lapply(1:length(data),function(x){data[[x]][,'pos'] <- position[x];
                                             data[[x]][,setdiff(uniquenames,names(data[[x]]))]<-0;
                                             data[[x]][,c(uniquenames,'pos')]})

  x <- do.call("rbind", data1)
```

```
## 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=-1

## calculate points
## points=c(fg=4, xpt=1, pass_yds=1/25, pass_tds=4, pass_ints=-2,rush_yds=1/10, rush_tds=6, fumbles=-1
#cat("Calculate points\n")
x.select <- x[,match(names(points),names(x))]
x.points <- t(apply(x.select,1,FUN=function(y){y*points}))
pts <- rowSums(x.points)
output <- data.frame(PlayerName=x[,"PlayerName"],pos=x[,"pos"],points=pts)
## head(output)

# sort by points

output <- output[order(output$points,decreasing = T),]

qb.idx <-which(output$pos=="qb")
rb.idx <-which(output$pos=="rb")
wr.idx <-which(output$pos=="wr")
te.idx <-which(output$pos=="te")
k.idx <- which(output$pos=="k")
## calculate marginal
## posReq=c(qb=1, rb=2, wr=3, te=1, k=1)
## nTeams = 15
## cap =200
#cat("Calculating marginal\n")
if(posReq['qb']!=0){
  output[qb.idx,'marginal'] <- output[qb.idx,'points']-output[qb.idx[nTeams*posReq['qb']],'points']
}
if(posReq['rb']!=0){
  output[rb.idx,'marginal'] <- output[rb.idx,'points']-output[rb.idx[nTeams*posReq['rb']],'points']
}
if(posReq['wr']!=0){
  output[wr.idx,'marginal'] <- output[wr.idx,'points']-output[wr.idx[nTeams*posReq['wr']],'points']
}
if(posReq['te']!=0){
  output[te.idx,'marginal'] <- output[te.idx,'points']-output[te.idx[nTeams*posReq['te']],'points']
}
if(posReq['k']!=0){
  output[k.idx,'marginal'] <- output[k.idx,'points']-output[k.idx[nTeams*posReq['k']],'points']
}


## keep players who have a positive marginal
#cat("Keep marginal >0\n")
output1 <- output[output$marginal>=0 & !is.na(output$marginal),]
#cat("keep",dim(output1),"\n")
#print(head(output1))
## calculate dollar values

#cat("Calculate dollar values\n")
output1[,'value'] <- output1[,'marginal']*(nTeams*cap-nrow(output1))/sum(output1[,'marginal']) + 1
## drop the column of "marginal"
```

```
  #print(head(output1))
  output2 <- output1[,c(1,2,3,5)]
  output2 <- output2[order(output2$value,decreasing = T),]
  #print(head(output2))
  ## row name
  rownames(output2) <- 1:dim(output2)[1]
  ## save dollar values as CSV file
  #cat("Write csv\n")
  write.csv(output2,file=paste(path,file,sep="/"))
  ## return data.frame with dollar values
  return(output2)
}
```

1. Call `x1 <- ffvalues('.')`
2. How many players are worth more than $20? (1 point)

```
x1 <- ffvalues('.')
sum(x1$value > 20)
```

```
## [1] 43
```

1. Who is 15th most valuable running back (rb)? (1 point)

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

```
##       PlayerName pos points    value
## 30 Derrick Henry  rb 147.73 28.36969
```

1. Call `x2 <- ffvalues(getwd(), '16team.csv', nTeams=16, cap=150)`
    1. How many players are worth more than $20? (1 point)

```
x2 <- ffvalues(getwd(), '16team.csv', nTeams=16, cap=150)
sum(x2$value > 20)
```

```
## [1] 43
```

1. How many wide receivers (wr) are in the top 40? (1 point)

```
top40 <- x2[1:40,]
sum(top40$pos == "wr")
```

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

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)

```
sum(x3$value > 20)
```

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

```
#head(x3)
```

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

```
top30 <-x3[1:30,]
sum(top30$pos=="qb")
```

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

Question 2

24 points

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

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

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.4.4
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
haart <-  read.csv("haart.csv")
haart$init.date <- mdy(haart$init.dat)
haart$last.visit <- mdy(haart$last.visit)
haart$date.death <- mdy(haart$date.death)
table(year(haart$init.date))
```

```
##
## 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$death.inoneyear <- ifelse(haart$date.death <= haart$init.date + years(1),1,0)
table(haart$death.inoneyear)['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.

```
idx.death.early <- which(haart$date.death < haart$last.visit | is.na(haart$last.visit))
idx.visit.early <- which(haart$date.death >= haart$last.visit | is.na(haart$date.death))
haart[idx.death.early,'followup'] <- haart[idx.death.early,]$init.date %--% haart[idx.death.early,]$dat
haart[idx.visit.early,'followup'] <- haart[idx.visit.early,]$init.date %--% haart[idx.visit.early,]$las

haart$followupdays <- as.duration(haart$followup) / ddays(1)
haart$followupdays[which(haart$followupdays>365)] <- 365
print(quantile(haart$followupdays))
```

```
##     0%    25%    50%    75%   100%
##   0.00 320.75 365.00 365.00 365.00
```

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[,'los.fol'] <- ifelse(is.na(haart$date.death) & haart$followupdays < 365,1,0)
sum(haart$los.fol)
```

```
## [1] 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_list2']] <- strsplit(init.reg, ","))[1:4]
```

```
## [[1]]
## [1] "3TC" "AZT" "EFV"
##
## [[2]]
## [1] "3TC" "AZT" "EFV"
##
## [[3]]
## [1] "3TC" "AZT" "EFV"
##
## [[4]]
## [1] "3TC" "AZT" "NVP"
```

```
# unique drugs
all_drugs <- unique(unlist(haart$init.reg_list2))
# indicator for each drug
reg_drugs <- matrix(FALSE, nrow=nrow(haart), ncol=length(all_drugs))
for(i in seq_along(all_drugs)) {
  reg_drugs[,i] <- sapply(haart$init.reg_list, function(z) all_drugs[i] %in% z)
}
reg_drugs <- data.frame(reg_drugs)
names(reg_drugs) <- all_drugs
#merge to haart
haart_merged <- cbind(haart, reg_drugs)

sumdrug <- apply(reg_drugs, 2, sum)
#sumdrug
which(sumdrug > 100)
```

```
## 3TC AZT EFV NVP D4T
##   1   2   3   4   5
```

3TC AZT EFV NVP and 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")

# repeat all the steps
```

```
haart2$init.date <- mdy(haart2$init.dat)
haart2$last.visit <- mdy(haart2$last.visit)
haart2$date.death <- mdy(haart2$date.death)
haart2$death.inoneyear <- ifelse(haart2$date.death <= haart2$init.date + years(1),1,0)
idx.death.early.2 <- which(haart2$date.death < haart2$last.visit | is.na(haart2$last.visit))
idx.visit.early.2 <- which(haart2$date.death >= haart2$last.visit | is.na(haart2$date.death))
haart2[idx.death.early.2,'followup'] <- haart2[idx.death.early.2,]$init.date %--% haart2[idx.death.early
haart2[idx.visit.early.2,'followup'] <- haart2[idx.visit.early.2,]$init.date %--% haart2[idx.visit.early

haart2$followupdays <- as.duration(haart2$followup) / ddays(1)
# loss of follow-up
haart2[,'los.fol'] <- ifelse(is.na(haart2$date.death) & haart2$followupdays < 365,1,0)
## indicator var for drug
init.reg <- as.character(haart2[,'init.reg'])
(haart2[['init.reg_list2']] <- strsplit(init.reg, ","))[1:4]
```

```
## [[1]]
## [1] "3TC" "AZT" "NVP"
##
## [[2]]
## [1] "3TC" "AZT" "NVP"
##
## [[3]]
## [1] "3TC" "DDI" "EFV"
##
## [[4]]
## [1] "3TC" "D4T" "NVP"
```

```
# unique drugs
all_drugs <- unique(unlist(haart$init.reg_list2))
all_drugs.2 <- unique(unlist(haart2$init.reg_list2))
# check whether there are new drugs in all_drugs.2
all_drugs.2 %in% all_drugs
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

```
# indicator for each drug
reg_drugs.2 <- matrix(FALSE, nrow=nrow(haart2), ncol=length(all_drugs))
for(i in seq_along(all_drugs)) {
  reg_drugs.2[,i] <- sapply(haart2$init.reg_list, function(z) all_drugs[i] %in% z)
}
reg_drugs.2 <- data.frame(reg_drugs.2)
names(reg_drugs.2) <- all_drugs

#merge to haart2
haart2_merged <- cbind(haart2, reg_drugs.2)
#merge to haart_merged
haart3 <- rbind(haart_merged,haart2_merged)
# first 5 records
haart3[1:5,]
```

```
##   male age aids cd4baseline logvl  weight hemoglobin    init.reg
## 1    1  25    0          NA    NA      NA         NA 3TC,AZT,EFV
## 2    1  49    0         143    NA 58.0608         11 3TC,AZT,EFV
## 3    1  42    1         102    NA 48.0816          1 3TC,AZT,EFV
```

```
## 4    0  33    0         107    NA 46.0000         NA 3TC,AZT,NVP
## 5    1  27    0          52     4      NA         NA 3TC,D4T,EFV
##    init.date last.visit death date.death death.inoneyear  followup
## 1 2003-07-01 2007-02-26     0       <NA>              NA 115430400
## 2 2004-11-23 2008-02-22     0       <NA>              NA 102470400
## 3 2003-04-30 2005-11-21     1 2006-01-11               0  80870400
## 4 2006-03-25 2006-05-05     1 2006-05-07               1   3542400
## 5 2004-09-01 2007-11-13     0       <NA>              NA 100915200
##   followupdays los.fol init.reg_list2  3TC   AZT   EFV   NVP   D4T   ABC
## 1          365       0  3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE
## 2          365       0  3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE
## 3          365       0  3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE
## 4           41       0  3TC, AZT, NVP TRUE  TRUE FALSE  TRUE FALSE FALSE
## 5          365       0  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
```

```
# last 5 records
tail(haart3, n = 5)
```

```
##      male      age aids cd4baseline    logvl  weight hemoglobin
## 1000    0 40.00000    1         131       NA 46.2672          8
## 1001    0 27.00000    0         232       NA      NA         NA
## 1002    1 38.72142    0         170       NA 84.0000         NA
## 1003    1 23.00000   NA         154 3.995635 65.5000         14
## 1004    0 31.00000    0         236       NA 45.8136         NA
##         init.reg  init.date last.visit death date.death death.inoneyear
## 1000 3TC,D4T,NVP 2003-07-03 2008-02-29     0       <NA>              NA
## 1001 3TC,AZT,NVP 2003-12-01 2004-01-05     0       <NA>              NA
## 1002 3TC,AZT,NVP 2002-09-26 2004-03-29     0       <NA>              NA
## 1003 3TC,DDI,EFV 2007-01-31 2007-04-16     0       <NA>              NA
## 1004 3TC,D4T,NVP 2003-12-03 2007-10-11     0       <NA>              NA
##       followup followupdays los.fol init.reg_list2  3TC   AZT   EFV   NVP
## 1000 147052800          365       0  3TC, D4T, NVP TRUE FALSE FALSE  TRUE
## 1001   3024000           35       1  3TC, AZT, NVP TRUE  TRUE FALSE  TRUE
## 1002  47520000          550       0  3TC, AZT, NVP TRUE  TRUE FALSE  TRUE
## 1003   6480000           75       1  3TC, DDI, EFV TRUE FALSE  TRUE FALSE
## 1004 121651200         1408       0  3TC, D4T, NVP TRUE FALSE FALSE  TRUE
##        D4T   ABC   DDI   IDV   LPV   RTV   SQV   FTC   TDF   DDC   NFV
## 1000  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1001 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1002 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1003 FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1004  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
```