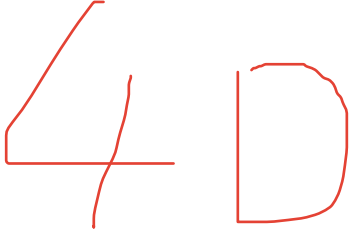

Bios 6301: Assignment 6

Lan Shi

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 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
  positions <- c('k','qb','rb','te','wr')
  csvfile <- paste('proj_', positions, '21', '.csv', sep='')
  files <- file.path(path, csvfile)
  names(files) <- positions
  k <- read.csv(files['k'], header=TRUE, stringsAsFactors=FALSE)
  qb <- read.csv(files['qb'], stringsAsFactors=FALSE)
  rb <- read.csv(files['rb'])
  te <- read.csv(files['te'])
```

```
wr <- read.csv(files['wr'])
cols <- unique(c(names(k), names(qb), names(rb), names(te), names(wr)))
k[,'pos'] <- 'k'
qb[,'pos'] <- 'qb'
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[['fg']]
x[,'p_xpt'] <- x[,'xpt']*points[['xpt']]
x[,'p_pass_yds'] <- x[,'pass_yds']*points[['pass_yds']]
x[,'p_pass_tds'] <- x[,'pass_tds']*points[['pass_tds']]
x[,'p_pass_ints'] <- x[,'pass_ints']*points[['pass_ints']]
x[,'p_rush_yds'] <- x[,'rush_yds']*points[['rush_yds']]
x[,'p_rush_tds'] <- x[,'rush_tds']*points[['rush_tds']]
x[,'p_fumbles'] <- x[,'fumbles']*points[['fumbles']]
x[,'p_rec_yds'] <- x[,'rec_yds']*points[['rec_yds']]
x[,'p_rec_tds'] <- x[,'rec_tds']*points[['rec_tds']]
# sum selected column values for every row
# this is total fantasy points for each player
x[,'points'] <- rowSums(x[,grep("^p_", names(x))])
x2 <- x[order(x[,'points'], decreasing=TRUE),]
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')

# calculate marginal points by subtracting "baseline" player's points
x2[,'marg'] = -1
if (posReq[['k']]!=0) x2[k.ix, 'marg'] <-
  x2[k.ix,'points'] - x2[k.ix[nTeams*posReq[['k']]],'points']
if (posReq[['qb']]!=0) x2[qb.ix, 'marg'] <-
  x2[qb.ix,'points'] - x2[qb.ix[nTeams*posReq[['qb']]],'points']
if (posReq[['rb']]!=0) x2[rb.ix, 'marg'] <-
  x2[rb.ix,'points'] - x2[rb.ix[nTeams*posReq[['rb']]],'points']
if (posReq[['te']]!=0) x2[te.ix, 'marg'] <-
  x2[te.ix,'points'] - x2[te.ix[nTeams*posReq[['te']]],'points']
if (posReq[['wr']]!=0) x2[wr.ix, 'marg'] <-
  x2[wr.ix,'points'] - x2[wr.ix[nTeams*posReq[['wr']]],'points']
```

```
  # create a new data.frame subset by non-negative marginal points
  x3 <- x2[x2['marg'] >= 0,]
  # 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','value')]

  ## save dollar values as CSV file
  write.csv(x4,file.path(path,file))
  ## return data.frame with dollar values
  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('.')
# number of players are worth more than $20
sum(x1$value>20)
```

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

```
# Who is 15th most valuable running back (rb)
x1[x1$pos=='rb',][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)
# number of players are worth more than $20
sum(x2$value>20)
```

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

```
# number of wide receivers (wr) in the top 40
table(x2$pos[1:40])["wr"]
```

```
## wr
##  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))
```

```r
# number of players are worth more than $20
sum(x3$value>20)
```

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

```r
# number of quarterbacks (qb)in the top 30
table(x3$pos[1:30])["qb"]
```

```
## qb
## 14
```

1. How many players are worth more than $20? (1 point)
2. 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)

```r
haart <-
  read.csv('https://raw.githubusercontent.com/couthcommander/Bios6301/main/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`.

```r
q1 = function(haart){
  date_col = c("init.date","last.visit","date.death")
  haart[,date_col] = data.frame(lapply(haart[,date_col],as.Date,format="%m/%d/%y"))
  return(haart)
}
haart = q1(haart)
str(haart)
```

```
## 'data.frame':    1000 obs. of  12 variables:
##  $ male       : int  1 1 1 0 1 0 0 1 1 1 ...
##  $ age        : num  25 49 42 33 27 34 39 31 52 23 ...
##  $ aids       : int  0 0 1 0 0 0 0 0 0 1 ...
##  $ cd4baseline: int  NA 143 102 107 52 157 65 NA NA 3 ...
##  $ logvl      : num  NA NA NA NA 4 ...
##  $ weight     : num  NA 58.1 48.1 46 NA ...
##  $ hemoglobin : num  NA 11 1 NA NA NA 11 NA NA NA ...
##  $ init.reg   : chr  "3TC,AZT,EFV" "3TC,AZT,EFV" "3TC,AZT,EFV" "3TC,AZT,NVP" ...
##  $ init.date  : Date, format: "2003-07-01" "2004-11-23" ...
##  $ last.visit : Date, format: "2007-02-26" "2008-02-22" ...
##  $ death      : int  0 0 1 1 0 0 0 0 0 1 ...
##  $ date.death : Date, format: NA NA ...
```

```r
table(lubridate::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?

```
q2 = function(haart){
  # for patients observed death, find the year difference between init.date and date.death
  deathIn1yr = rep(0,nrow(haart))
  deathIn1yr[difftime(haart$date.death, haart$init.date, units="days") <= 365] = 1
  haart = cbind(haart,deathIn1yr)
}
haart = q2(haart)
# number of observations died in year 1:
sum(haart$deathIn1yr)
```

```
## [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.

```
q3 = function(haart){
  deathdiff = difftime(haart$date.death, haart$init.date, units="days")
  deathdiff[is.na(deathdiff)] = Inf
  lastdiff = difftime(haart$last.visit, haart$init.date, units="days")
  lastdiff[is.na(lastdiff)] = Inf
  followup = apply(cbind(deathdiff,lastdiff),1,min)
  followup[followup>365] = 365
  haart = cbind(haart,followup)
}
haart = q3(haart)
# quantiles of followup
print(quantile(haart$followup,probs = .1*(1:10)))
```

```
##   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##  41.0 208.6 365.0 365.0 365.0 365.0 365.0 365.0 365.0 365.0
```

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 to followup
# 1. unknown date.death, death == 0
# 2. followup within first year.
q4 = function(haart){
  lost_to_followup = as.numeric(haart$death==0 & haart$followup<365)
  haart = cbind(haart,lost_to_followup)
}
haart = q4(haart)
# number of records are lost-to-followup.
sum(haart$lost_to_followup)
```

```
## [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?

```
# codes from class:
q5 = function(haart){
  init.reg <- haart$init.reg
```

```
  haart[['init.reg_list']] <- strsplit(init.reg, ",")
  all_drugs <- unique(unlist(haart$init.reg_list))
  reg_drugs <- matrix(FALSE, nrow=nrow(haart), ncol=length(all_drugs))
  for(i in seq_along(all_drugs)) {
    reg_drugs[,i] <- sapply(haart$init.reg_list, function(x) all_drugs[i] %in% x)
  }
  reg_drugs <- data.frame(reg_drugs)
  names(reg_drugs) <- all_drugs
  return(reg_drugs )
}
reg_drugs = q5(haart)
haart = cbind(haart,reg_drugs)
haart[1:3,]
```

```
##   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
##   last.visit death date.death deathIn1yr followup lost_to_followup  3TC  AZT
## 1 2007-02-26     0       <NA>          0      365                0 TRUE TRUE
## 2 2008-02-22     0       <NA>          0      365                0 TRUE TRUE
## 3 2005-11-21     1 2006-01-11          0      365                0 TRUE TRUE
##    EFV   NVP   D4T   ABC   DDI   IDV   LPV   RTV   SQV   FTC   TDF   DDC   NFV
## 1 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3 TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     T20   ATV   FPV
## 1 FALSE FALSE FALSE
## 2 FALSE FALSE FALSE
## 3 FALSE FALSE FALSE
```

```
# Which drug regimen are found over 100 times?
drugcount = apply(reg_drugs,2,sum)
drugcount[drugcount > 100]
```

```
## 3TC AZT EFV NVP D4T
## 973 794 516 358 146
```

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.

```
library(magrittr)
haart2 <-
  read.csv('https://raw.githubusercontent.com/couthcommander/Bios6301/main/datasets/haart2.csv')
haart2 = haart2 %>% q1 %>% q2 %>% q3 %>% q4
haart2
```

```
##   male      age aids cd4baseline    logvl  weight hemoglobin    init.reg
## 1    0 27.00000    0         232       NA      NA         NA 3TC,AZT,NVP
## 2    1 38.72142    0         170       NA 84.0000         NA 3TC,AZT,NVP
## 3    1 23.00000   NA         154 3.995635 65.5000         14 3TC,DDI,EFV
## 4    0 31.00000    0         236       NA 45.8136         NA 3TC,D4T,NVP
##    init.date last.visit death date.death deathIn1yr followup lost_to_followup
## 1 2003-12-01 2004-01-05     0       <NA>          0       35                1
```

```
## 2 2002-09-26 2004-03-29     0       <NA>          0      365                0
## 3 2007-01-31 2007-04-16     0       <NA>          0       75                1
## 4 2003-12-03 2007-10-11     0       <NA>          0      365                0
```

```
reg_drugs2 = q5(haart2)
reg_drugs2_full = data.frame(matrix(FALSE,nrow=nrow(reg_drugs2),
                                    ncol=ncol(reg_drugs)))
colnames(reg_drugs2_full) = colnames(reg_drugs)
reg_drugs2_full[,colnames(reg_drugs2)] = reg_drugs2
haart2 = cbind(haart2,reg_drugs2_full)

complete_dt = rbind(haart,haart2)

complete_dt[c(1:5,((nrow(complete_dt)-4):(nrow(complete_dt)))),]
```

```
##      male      age aids cd4baseline    logvl  weight hemoglobin    init.reg
## 1       1 25.00000    0          NA       NA      NA         NA 3TC,AZT,EFV
## 2       1 49.00000    0         143       NA 58.0608         11 3TC,AZT,EFV
## 3       1 42.00000    1         102       NA 48.0816          1 3TC,AZT,EFV
## 4       0 33.00000    0         107       NA 46.0000         NA 3TC,AZT,NVP
## 5       1 27.00000    0          52 4.000000      NA         NA 3TC,D4T,EFV
## 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 deathIn1yr followup
## 1    2003-07-01 2007-02-26     0       <NA>          0      365
## 2    2004-11-23 2008-02-22     0       <NA>          0      365
## 3    2003-04-30 2005-11-21     1 2006-01-11          0      365
## 4    2006-03-25 2006-05-05     1 2006-05-07          1       41
## 5    2004-09-01 2007-11-13     0       <NA>          0      365
## 1000 2003-07-03 2008-02-29     0       <NA>          0      365
## 1001 2003-12-01 2004-01-05     0       <NA>          0       35
## 1002 2002-09-26 2004-03-29     0       <NA>          0      365
## 1003 2007-01-31 2007-04-16     0       <NA>          0       75
## 1004 2003-12-03 2007-10-11     0       <NA>          0      365
##      lost_to_followup  3TC   AZT   EFV   NVP   D4T   ABC   DDI   IDV   LPV
## 1                   0 TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 2                   0 TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 3                   0 TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 4                   0 TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 5                   0 TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## 1000                0 TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
## 1001                1 TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 1002                0 TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 1003                1 TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## 1004                0 TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
##        RTV   SQV   FTC   TDF   DDC   NFV   T20   ATV   FPV
## 1    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 5    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1000 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## 1001 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1002 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1003 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1004 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```