Bios 6301: Assignment 6

Megan Taylor

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
  # To structure the data I used almost the exact same method as
  # presented in the first lecture
  k <- read.csv(paste(path, "\\proj_k21.csv", sep=""))
  qb <- read.csv(paste(path, "\\proj_qb21.csv", sep=""))
  rb <- read.csv(paste(path, "\\proj_rb21.csv", sep=""))
  wr <- read.csv(paste(path, "\\proj_wr21.csv", sep=""))
  te <- read.csv(paste(path, "\\proj_te21.csv", sep=""))

  # code from first lecture
  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'
cols <- c(cols, 'pos')
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
x <- rbind(k[,cols], qb[,cols], rb[,cols], te[,cols], wr[,cols])

# use point values specified by function in case default is not used
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"]

x['points'] <- rowSums(x[,grep("^p_", names(x))])


## calculate dollar values
# modified code from first lecture
x2 <- x[order(x[,'fpts'], decreasing = TRUE),]
k.i <- which(x2[,'pos']=='k')
qb.i <- which(x2[,'pos']=='qb')
rb.i <- which(x2[,'pos']=='rb')
te.i <- which(x2[,'pos']=='te')
wr.i <- which(x2[,'pos']=='wr')

# formula for dollar value (taken from first lecture): (cap*nTeams -
# nTeams*sum(posReq))*margi/(sum i=1 to nTeams*sum(posReq) of margi) +1

# marginal values (checking for 0 required of each position and applying
# code from first lecture)
if (posReq["qb"] != 0)
  x2[qb.i, 'marginal'] <- x2[qb.i, 'points']-x2[qb.i[nTeams*posReq["qb"]], 'points']
if (posReq["rb"] != 0)
  x2[rb.i, 'marginal'] <- x2[rb.i, 'points']-x2[rb.i[nTeams*posReq["rb"]], 'points']
if (posReq["wr"] != 0)
  x2[wr.i, 'marginal'] <- x2[wr.i, 'points']-x2[wr.i[nTeams*posReq["wr"]], 'points']
if (posReq["te"] != 0)
  x2[te.i, 'marginal'] <- x2[te.i, 'points']-x2[te.i[nTeams*posReq["te"]], 'points']
if (posReq["k"] != 0)
  x2[k.i, 'marginal'] <- x2[k.i, 'points']-x2[k.i[nTeams*posReq["k"]], 'points']

# more code mostly from first lecture
```

```
  x3 <- x2[x2[,'marginal'] >= 0,]
  x3 <- x3[order(x3[,'marginal'], decreasing = TRUE),]
  rownames(x3) <- NULL

  x3[,'value'] <- (cap*nTeams - nTeams*sum(posReq))*x3[,'marginal']/
    sum(x3[,'marginal'], na.rm = TRUE) + 1

  ## save dollar values as CSV file
  dol.vals = data.frame('PlayerName' = x3[,"PlayerName"], 'pos' = x3[,'pos'],
                        'points' = x3[,'points'], 'value' = x3[,'value'])
  write.table(dol.vals, file = file, sep = ",", row.names = FALSE)

  ## return data.frame with dollar values
  dol.vals
}
```

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

       45

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

       Chris Carson

```
x1 <- ffvalues('.')

# How many players are worth more than $20?
length(which(x1[,'value']>20))
```

```
## [1] 45
```

```
# Who is 15th most valuable running back (rb)?
rb <- which(x1[,'pos']=='rb')
rb[15] # equals 29
```

```
## [1] 29
```

```
x1[29,]
```

```
##      PlayerName pos points    value
## 29 Chris Carson  rb 176.24 32.30756
```

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

       44

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

       8

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

# How many players are worth more than $20?
length(which(x2[,'value']>20))
```

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

```r
# How many wide receivers (wr) are in the top 40?
val <- x2[41, 'value']
length(which(x2[,'pos']=='wr' & x2[,'value']>val))
```

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

1. Call:

```r
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)

      48

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

      14

```r
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))

# How many players are worth more than $20?
length(which(x3[,'value']>20))
```

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

```r
# How many quarterbacks (qb) are in the top 30?
val <- x3[31, 'value']
length(which(x3[,'pos']=='qb' & x3[,'value']>val))
```

```
## [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)

```r
haart = read.csv("C:\\Users\\megan\\OneDrive\\Documents\\Fall 2021\\Statistical Computing\\Bios6301-mai

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

dates <- function(df){
  df[,"init.date"] = as.POSIXct(df[,'init.date'], format = "%m/%d/%y")
  df[,"last.visit"] = as.POSIXct(df[,'last.visit'], format = "%m/%d/%y")
  df[,"date.death"] = as.POSIXct(df[,'date.death'], format = "%m/%d/%y")

  years = c()
  for (i in 1:nrow(df)){
    if (is.na(df[i,'date.death']))
      df[i,"weeks"] = difftime(df[i,"last.visit"],df[i,"init.date"], units = "weeks")
    else
      df[i,"weeks"] = difftime(df[i,"date.death"],df[i,"init.date"], units = "weeks")
    df[i,"years"]= as.double(df[i,"weeks"])/52
```

```
    years = c(years, as.double(df[i,"years"]))
  }
  df[,"years"] <- years
  df[,"round.years"] <- round(years)
  df
}

haart <- dates(haart)
table(haart[,'round.years'])
```

```
##
##   0   1   2   3   4   5   6   7  10
## 188 176 214 193 130  85  11   2   1
```

```
# 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?

death1 = function(df){
  for (i in 1:nrow(df)){
    if (df[i,"years"]<1 & is.na(df[i,'date.death'])==FALSE)
      df[i,"d1"] = 1
    else
      df[i,"d1"] = 0
  }
  df
}

haart = death1(haart)
sum(haart[,"d1"])
```

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

```
# 92 observations died in year 1


# 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.

followup = function(df){
  for (i in 1:nrow(df)){
    if (is.na(df[i,'date.death']))
      df[i,"follow.up.days"] = difftime(df[i,"last.visit"],df[i,"init.date"],
                                        units = "days")
    else
      df[i,"follow.up.days"] = difftime(df[i,"date.death"],df[i,"init.date"],
                                        units = "days")
    if (as.double(df[i, "follow.up.days"])>365)
      df[i,"follow.up.days"] = 365
  }
  df
}
```

```r
haart <- followup(haart)
quantile(haart[,"follow.up.days"])
```

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

```r
# 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?

loss = function(df){
  for (i in 1:nrow(df)){
    if (is.na(df[i,"date.death"]) & df[i,"follow.up.days"]<365)
      df[i, "lost"] = 1
    else
      df[i, "lost"] = 0
  }
  df
}

haart = loss(haart)
sum(haart[,"lost"])
```

```
## [1] 173
```

```r
# 173 observations 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?

reg = function(df){
  for (r in unique(df[,"init.reg"])){
    df[,r] <-0
    for (i in 1:nrow(df)){
      if (df[i, "init.reg"] == r)
        df[i,r] = 1
    }
  }
  df
}

haart.reg = reg(haart)

reg.names = unique(haart[,"init.reg"])
r.sums <- colSums(haart.reg[,reg.names])
which(r.sums>100)
```

```
## 3TC,AZT,EFV 3TC,AZT,NVP
##           1           2
```

```r
#The regimens 3TC,AZT,EFV and 3TC,AZT,NVP are each 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.

# Used my functions from the previous questions. I decided to leave out the 'reg'
# function from question 5 because this question is not interested in sums of the
# regimen and it makes the data hard to read if a column for each regimen is printed
haart2 = read.csv("C:\\Users\\megan\\OneDrive\\Documents\\Fall 2021\\Statistical Computing\\Bios6301-ma
haart2 <- dates(haart2)
haart2 <- death1(haart2)
haart2 <- followup(haart2)
haart2 <- loss(haart2)

# use rbind to append
haart_all = rbind(haart, haart2)

# print requested records
haart_all[1:5,]
```

```
##   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            weeks     years round.years d1
## 1 2007-02-26     0       <NA> 190.863095 weeks 3.6704441           4  0
## 2 2008-02-22     0       <NA> 169.428571 weeks 3.2582418           3  0
## 3 2005-11-21     1 2006-01-11 141.005952 weeks 2.7116529           3  0
## 4 2006-05-05     1 2006-05-07   6.136905 weeks 0.1180174           0  1
## 5 2007-11-13     0       <NA> 166.863095 weeks 3.2089057           3  0
##   follow.up.days lost
## 1 365.00000 days    0
## 2 365.00000 days    0
## 3 365.00000 days    0
## 4  42.95833 days    0
## 5 365.00000 days    0
```

```
haart_all[1000:1004,]
```

```
##      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           weeks      years
## 1000 2003-07-03 2008-02-29     0       <NA> 243.14881 weeks 4.67593864
## 1001 2003-12-01 2004-01-05     0       <NA>   5.00000 weeks 0.09615385
## 1002 2002-09-26 2004-03-29     0       <NA>  78.57738 weeks 1.51110348
## 1003 2007-01-31 2007-04-16     0       <NA>  10.70833 weeks 0.20592949
## 1004 2003-12-03 2007-10-11     0       <NA> 201.13690 weeks 3.86801740
```

```
##      round.years d1 follow.up.days lost
## 1000           5  0 365.00000 days    0
## 1001           0  0  35.00000 days    1
## 1002           2  0 365.00000 days    0
## 1003           0  0  74.95833 days    1
## 1004           4  0 365.00000 days    0
```