Bios 6301: Assignment 6

Catherine Greene

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 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

# original values, nTeams=12, cap=200

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('proj_k21.csv', header=TRUE, stringsAsFactors=FALSE)
  qb <- read.csv('proj_qb21.csv', stringsAsFactors=FALSE)
  rb <- read.csv('proj_rb21.csv')
  te <- read.csv('proj_te21.csv')
  wr <- read.csv('proj_wr21.csv')

  # what columns are present in these files?
  cols <- unique(c(names(k), names(qb), names(rb), names(te), names(wr)))
```

```
# add a column specifying the position (aka the source file)
k[,'pos'] <- 'k'
qb[,'pos'] <- 'qb'
rb[,'pos'] <- 'rb'
te[,'pos'] <- 'te'
wr[,'pos'] <- 'wr'
cols <- c(cols, 'pos')

# adds addl columns not found in each file, with values set to 0
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
df <- rbind(k[,cols], qb[,cols], rb[,cols], te[,cols], wr[,cols])

## calculate dollar values

# first calculate fantasy points, point values specified in the function arguments
df[,'p_fg'] <- df[,'fg']*points['fg']
df[,'p_xpt'] <- df[,'xpt']*points['xpt']
df[,'p_pass_yds'] <- df[,'pass_yds']*points['pass_yds']
df[,'p_pass_tds'] <- df[,'pass_tds']*points['pass_tds']
df[,'p_pass_ints'] <- df[,'pass_ints']*points['pass_ints']
df[,'p_rush_yds'] <- df[,'rush_yds']*points['rush_yds']
df[,'p_rush_tds'] <- df[,'rush_tds']*points['rush_tds']
df[,'p_fumbles'] <- df[,'fumbles']*points['fumbles']
df[,'p_rec_yds'] <- df[,'rec_yds']*points['rec_yds']
df[,'p_rec_tds'] <- df[,'rec_tds']*points['rec_tds']

# total fantasy points for each player, sum of point columns
df[,'points'] <- rowSums(df[,grep("^p_", names(df))])

# now can convert to dollar values

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

k.ix <- which(df2[,'pos']=='k')
qb.ix <- which(df2[,'pos']=='qb')
rb.ix <- which(df2[,'pos']=='rb')
te.ix <- which(df2[,'pos']=='te')
wr.ix <- which(df2[,'pos']=='wr')

# calculate marginal points by subtracting "baseline" player's points
# every 12th-best player's marginal points = 0
# posReq=c(qb=1, rb=2, wr=3, te=1, k=1)

if(posReq['k']==0) {
df2[k.ix, 'marg'] <- -1 # just the kickers for now
df2[qb.ix, 'marg'] <- df2[qb.ix,'points'] - df2[qb.ix[nTeams*posReq['qb']],'points']
df2[rb.ix, 'marg'] <- df2[rb.ix,'points'] - df2[rb.ix[nTeams*posReq['rb']],'points']
```

```
  df2[te.ix, 'marg'] <- df2[te.ix,'points'] - df2[te.ix[nTeams*posReq['te']],'points']
  df2[wr.ix, 'marg'] <- df2[wr.ix,'points'] - df2[wr.ix[nTeams*posReq['wr']],'points']
} else {
  df2[k.ix, 'marg'] <- df2[k.ix,'points'] - df2[k.ix[nTeams*posReq['k']],'points']
  df2[qb.ix, 'marg'] <- df2[qb.ix,'points'] - df2[qb.ix[nTeams*posReq['qb']],'points']
  df2[rb.ix, 'marg'] <- df2[rb.ix,'points'] - df2[rb.ix[nTeams*posReq['rb']],'points']
  df2[te.ix, 'marg'] <- df2[te.ix,'points'] - df2[te.ix[nTeams*posReq['te']],'points']
  df2[wr.ix, 'marg'] <- df2[wr.ix,'points'] - df2[wr.ix[nTeams*posReq['wr']],'points']
}

  # create a new data.frame subset by non-negative marginal points
  df3 <- df2[df2[,'marg'] >= 0,]

  # re-order by marginal points
  df3 <- df3[order(df3[,'marg'], decreasing=TRUE),]

  rownames(df3) <- NULL

  # calculation for player value
  df3[,'value'] <- (nTeams*cap-nrow(df3)) * df3[,'marg'] / sum(df3[,'marg']) + 1

  # create a data.frame with more interesting columns
  df4 <- df3[,c('PlayerName','pos','points','value')]

  ## save dollar values as CSV file
  # columns for new file = 'PlayerName', 'pos', 'points', 'value'
  write.csv(df4, file=file)

  ## return data.frame with dollar values
  df4
  #nrow(sum(posReq)*nTeams)
}
```

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

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

# nrow(x1) # = 96
# sum(posReq)*nTeams) = 8 * 12 = 96
# row number matches what it's supposed to be
```

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

```
length(x1['PlayerName'][which(x1['value'] > 20),]) # 41 players
```

```
## [1] 41
```

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

```
x1['PlayerName'][which(x1[,'pos']=='rb')[15],] # David Montgomery
```

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

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

```
x2 <- ffvalues(getwd(), '16team.csv', nTeams=16, cap=150)
#sum(posReq)*nTeams) = 8 * 16 = 128 rows
```

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

```
length(x2['PlayerName'][which(x2['value'] > 20),]) # 46 players
```

```
## [1] 46
```

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

```
nrow(x2[which(x2[1:40,]['pos']=='wr'),]) # 8 wide receivers
```

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

3. 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))
```

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

```
length(x3['PlayerName'][which(x3['value'] > 20),]) # 43 players
```

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

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

```
nrow(x3[which(x3[1:30,]['pos']=='qb'),]) # 13 quarterbacks
```

```
## [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)

```
haart <- read.table("/Users/KatietheWise/Desktop/2022_Fall/StatComp/Homework/haart.csv", header=TRUE, s
```

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

# make a new column with the year
haart['init_year'] <- strftime(haart[,'init.date'], "%Y")
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?

```
#death_indic <-
# time difference, either 1 year or 12 months

death_indic <-c()
for (i in 1:(nrow(haart))){
```

```
  if (!is.na(haart[,'date.death'][i])){ # there's a death date
    if (difftime(haart['date.death'][i,],haart['init.date'][i,], units='days') <= 365.25){
      death_indic[i] <- 1 # died within a year of initial visit
    } else {
      death_indic[i] <- 0 # died, but more than a year after initial visit
      }
  } else if (is.na(haart[,'date.death'][i])){ # there's no death date
      death_indic[i] <- 0
  }
}

sum(death_indic) # 92 deaths occurred within the first year following the initial visit
```

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

```
# append to the dataframe
haart[,'death_indic'] <- death_indic
```

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

follow_up <- c()
for (i in 1:(nrow(haart))){
  if (!is.na(haart['date.death'][i,] > haart['last.visit'][i,])){ # both values present, death date is
    difference <- difftime(haart['date.death'][i,],haart['init.date'][i,], units='days')
    if (difference > 365) {
      difference <- 365
    }
    follow_up[i] <- difference
  } else if ((is.na(haart['date.death'][i,]) && (!is.na(haart['last.visit'][i,])))){ # no death date, o
    difference <- difftime(haart['last.visit'][i,],haart['init.date'][i,], units='days')
    if (difference > 365) {
      difference <- 365
    }
    follow_up[i] <- difference
  } else if ((is.na(haart['last.visit'][i,]) && (!is.na(haart['date.death'][i,])))){ # no last visit, o
    difference <- difftime(haart['date.death'][i,],haart['init.date'][i,], units='days')
    if (difference > 365) {
      difference <- 365
    } else if ((is.na(haart['last.visit'][i,]) && (is.na(haart['date.death'][i,])))) { # both NA
      difference <- NA
    }
    follow_up[i] <- difference
  }
}

#follow_up

quantile(follow_up) # usually it's more than a year between initial visit and last visit / death
```

```
##    0%   25%   50%   75%  100%
```

```
##   0.0 329.5 365.0 365.0 365.0
```

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 <- c()
for (i in 1:(nrow(haart))){
  if ((is.na(haart['date.death'][i,]) && (!is.na(haart['last.visit'][i,])))){ # no death date, only las
    difference <- difftime(haart['last.visit'][i,],haart['init.date'][i,], units='days')
    if (difference < 365) {
      lost[i] <- 0 # followed up within the first year but not after, lost to follow-up
    } else if (difference > 365) {
      lost[i] <- 1 # followed up after the first year
    }
  } else {
    lost[i] <- 1 # they did follow-up at some point
  }
}

lost # 0 = lost to follow-up
```

```
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1
##   [38] 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
##   [75] 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 0 1
##  [112] 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1
##  [149] 0 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1
##  [186] 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
##  [223] 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1 0
##  [260] 0 1 0 1 0 0 1 1 1 0 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [297] 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
##  [334] 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1
##  [371] 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1
##  [408] 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1
##  [445] 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1
##  [482] 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1
##  [519] 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0
##  [556] 1 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
##  [593] 0 1 1 1 0 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [630] 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1
##  [667] 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
##  [704] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1
##  [741] 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
##  [778] 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1
##  [815] 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1
##  [852] 1 1 1 1 1 1 1 0 1 0 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
##  [889] 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
##  [926] 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1
##  [963] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1
## [1000] 1
```

```
length(which(lost == 0)) # 173
```

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

```
# append to dataframe

haart[,'loss_to_followup'] <- lost
```

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?

```
unlist(haart$init.reg_list)[seq(50)]
```

```
##  [1] "3TC" "AZT" "EFV" "3TC" "AZT" "EFV" "3TC" "AZT" "EFV" "3TC" "AZT" "NVP"
## [13] "3TC" "D4T" "EFV" "3TC" "AZT" "NVP" "3TC" "AZT" "NVP" "3TC" "AZT" "EFV"
## [25] "3TC" "ABC" "AZT" "3TC" "DDI" "NVP" "3TC" "AZT" "NVP" "3TC" "AZT" "IDV"
## [37] "3TC" "AZT" "NVP" "3TC" "AZT" "EFV" "3TC" "AZT" "EFV" "3TC" "D4T" "NVP"
## [49] "3TC" "AZT"
```

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

```
##  [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

# append to the haart database as new columns
haart_merged <- cbind(haart, reg_drugs)

# which combination(s) are found over 100 times?

haart[,'init.reg'] <- as.factor(haart[,'init.reg'])
#str(haart[,'init.reg']) #47 different combinations
#table(haart[,'init.reg'])
which(table(haart[,'init.reg']) > 100) # 3TC,AZT,EFV & 3TC,AZT,NVP
```

```
## 3TC,AZT,EFV 3TC,AZT,NVP
##          10          17
```

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.table("/Users/KatietheWise/Desktop/2022_Fall/StatComp/Homework/haart2.csv", header=TRUE,

# convert to usable date formats
haart2[,'last.visit'] <- as.POSIXct(haart2[,'last.visit'], format="%m/%d/%y")
haart2[,'init.date'] <- as.POSIXct(haart2[,'init.date'], format="%m/%d/%y")
haart2[,'date.death'] <- as.POSIXct(haart2[,'date.death'], format="%m/%d/%y")

# make a new column with the year
haart2['init_year'] <- strftime(haart2[,'init.date'], "%Y")

# death indicator
```

```
death_indic2 <-c()
for (i in 1:(nrow(haart2))){
  if (!is.na(haart2[,'date.death'][i])){ # there's a death date
    if (difftime(haart2['date.death'][i,],haart2['init.date'][i,], units='days') <= 365.25){
      death_indic2[i] <- 1
    } else {
      death_indic2[i] <- 0
      }
  } else if (is.na(haart2[,'date.death'][i])){
      death_indic2[i] <- 0
  }
}

# append to the dataframe
haart2[,'death_indic'] <- death_indic2

# loss to followup
lost2 <- c()
for (i in 1:(nrow(haart2))){
  if ((is.na(haart2['date.death'][i,]) && (!is.na(haart2['last.visit'][i,])))){ # no death date, only l
    difference <- difftime(haart2['last.visit'][i,],haart2['init.date'][i,], units='days')
    if (difference < 365) {
      lost2[i] <- 0 # followed up within the first year but not after, lost to follow-up
    } else if (difference > 365) {
      lost2[i] <- 1 # followed up after the first year
    }
  } else {
    lost2[i] <- 1 # they did follow-up at some point
  }
}

# append to dataframe
haart2[,'loss_to_followup'] <- lost2

# drug combos
init.reg2 <- as.character(haart2[,'init.reg'])
(haart2[['init.reg_list']] <- strsplit(init.reg2, ","))
```

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

```
unlist(haart2$init.reg_list)[seq(50)]
```

```
##  [1] "3TC" "AZT" "NVP" "3TC" "AZT" "NVP" "3TC" "DDI" "EFV" "3TC" "D4T" "NVP"
## [13] NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
```

```
## [25] NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## [37] NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## [49] NA    NA
```

```
(all_drugs2 <- unique(unlist(haart2$init.reg_list))) # all unique drugs
```

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

```
reg_drugs2 <- matrix(FALSE, nrow=nrow(haart2), ncol=length(all_drugs2))
for(i in seq_along(all_drugs2)) {
  reg_drugs2[,i] <- sapply(haart2$init.reg_list, function(x) all_drugs2[i] %in% x)
}
reg_drugs2 <- data.frame(reg_drugs2)
names(reg_drugs2) <- all_drugs2
head(reg_drugs2)
```

```
##    3TC   AZT   NVP   DDI   EFV   D4T
## 1 TRUE  TRUE  TRUE FALSE FALSE FALSE
## 2 TRUE  TRUE  TRUE FALSE FALSE FALSE
## 3 TRUE FALSE FALSE  TRUE  TRUE FALSE
## 4 TRUE FALSE  TRUE FALSE FALSE  TRUE
```

```
# append to the haart database as new columns
#haart2_merged <- cbind(haart2, reg_drugs2)

# noticed an issue here with empty headers
orig_names <- c(names(reg_drugs))
add_names <- c(names(reg_drugs2))
to_add <- c(setdiff(orig_names, add_names))

# this part is overly complicated but I ran out of ideas
df1 <- data.frame('ABC' = c(FALSE, FALSE, FALSE, FALSE),
                  'IDV' = c(FALSE, FALSE, FALSE, FALSE),
                  'LPV' = c(FALSE, FALSE, FALSE, FALSE),
                  'RTV' = c(FALSE, FALSE, FALSE, FALSE),
                  'SQV' = c(FALSE, FALSE, FALSE, FALSE),
                  'FTC' = c(FALSE, FALSE, FALSE, FALSE),
                  'TDF' = c(FALSE, FALSE, FALSE, FALSE),
                  'DDC' = c(FALSE, FALSE, FALSE, FALSE),
                  'NFV' = c(FALSE, FALSE, FALSE, FALSE),
                  'T2O' = c(FALSE, FALSE, FALSE, FALSE),
                  'ATV' = c(FALSE, FALSE, FALSE, FALSE),
                  'FPV' = c(FALSE, FALSE, FALSE, FALSE)
)

new_df <- cbind(reg_drugs2, df1)
haart2_merged <- cbind(haart2, new_df)

# not in 100% the right order, but rbind should take care of that
# now merge haart_merged and haart2_merged
haart_final <- rbind(haart_merged, haart2_merged)

# first five records
haart_final[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 init_year death_indic loss_to_followup
## 1 2007-02-26     0       <NA>      2003           0                1
## 2 2008-02-22     0       <NA>      2004           0                1
## 3 2005-11-21     1 2006-01-11      2003           0                1
## 4 2006-05-05     1 2006-05-07      2006           1                1
## 5 2007-11-13     0       <NA>      2004           0                1
##   init.reg_list  3TC   AZT   EFV   NVP   D4T   ABC   DDI   IDV   LPV   RTV
## 1 3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3 3TC, AZT, EFV TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4 3TC, AZT, NVP TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 5 3TC, D4T, EFV TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##     SQV   FTC   TDF   DDC   NFV   T20   ATV   FPV
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 5 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
# last five records
haart_final[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 init_year death_indic
## 1000 2003-07-03 2008-02-29     0       <NA>      2003           0
## 1001 2003-12-01 2004-01-05     0       <NA>      2003           0
## 1002 2002-09-26 2004-03-29     0       <NA>      2002           0
## 1003 2007-01-31 2007-04-16     0       <NA>      2007           0
## 1004 2003-12-03 2007-10-11     0       <NA>      2003           0
##      loss_to_followup init.reg_list  3TC   AZT   EFV   NVP   D4T   ABC   DDI
## 1000                1 3TC, D4T, NVP TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
## 1001                0 3TC, AZT, NVP TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
## 1002                1 3TC, AZT, NVP TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
## 1003                0 3TC, DDI, EFV TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
## 1004                1 3TC, D4T, NVP TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
##        IDV   LPV   RTV   SQV   FTC   TDF   DDC   NFV   T20   ATV   FPV
## 1000 FALSE 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 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1004 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```