Bios 6301: Assignment 6

Jeongwon Choi

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 (https://github.com/couthcommander/football-values). 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):

```
## ----- load package & set wd
# assumed that the current working directory (and the location that this function will be calle
d) is in the 2021 year folder of football-values-main

library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.8     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.1
## ✓ readr   2.1.2     ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
## ----- define the function

# 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)) {

  ## ----- 1. read in CSV files
  year <- 2021

  positions <- c('k','qb','rb','te','wr')
  files <- paste0(path,'/','proj_',positions,substr(year,3,4),'.csv')

  names(files) <- positions

  # read csv from each position
  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'])

  # make a column name for integrated df
  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])

  ## 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']

  # calculate the point
  x[,'points'] <- rowSums(x[,grep("^p_", names(x))])
  x2 <- x[order(x[,'points'], decreasing=TRUE),]
  rownames(x2) <-NULL

  # get an index for each position
  k.idx <- which(x2[,'pos']=='k')
  qb.idx <- which(x2[,'pos']=='qb')
  rb.idx <- which(x2[,'pos']=='rb')
  te.idx <- which(x2[,'pos']=='te')
  wr.idx <- which(x2[,'pos']=='wr')

  # calculate marginal values
  n_req.k <- posReq['k']*nTeams
  n_req.qb <- posReq['qb']*nTeams
  n_req.rb <- posReq['rb']*nTeams
  n_req.te <- posReq['te']*nTeams
  n_req.wr <- posReq['wr']*nTeams

  # If the number of required position is zero, make marginal values -Inf, so that it will not b
e
  # included in the final data frame
  if (n_req.k==0) {
    x2[k.idx, 'marg'] <- -Inf
  } else {
    x2[k.idx, 'marg'] <- x2[k.idx,'points'] - x2[k.idx[n_req.k],'points']
  }

  if (n_req.qb==0){
    x2[qb.idx, 'marg'] <- -Inf
  } else {
    x2[qb.idx, 'marg'] <- x2[qb.idx,'points'] - x2[qb.idx[n_req.qb],'points']
  }

  if (n_req.rb==0){
    x2[rb.idx, 'marg'] <- -Inf
  } else {
    x2[rb.idx, 'marg'] <- x2[rb.idx,'points'] - x2[rb.idx[n_req.rb],'points']
  }

  if (n_req.te==0){
    x2[te.idx, 'marg'] <- -Inf
  } else {
    x2[te.idx, 'marg'] <- x2[te.idx,'points'] - x2[te.idx[n_req.te],'points']
  }
```

```
  if (n_req.wr==0){
    x2[wr.idx, 'marg'] <- -Inf
  } else {
    x2[wr.idx, 'marg'] <- x2[wr.idx,'points'] - x2[wr.idx[n_req.wr],'points']
  }

  # create data.frame by removing players with negative marginal values
  x3 <- x2[x2[,'marg'] >= 0,]
  x3 <- x3[order(x3[,'marg'], decreasing=TRUE),]
  rownames(x3) <- NULL

  # calculate a dollar value!
  x3[,'value'] <- (nTeams*cap-nrow(x3)) * ( x3[,'marg'] / sum(x3[,'marg']) ) + 1

  # genearte a final data frame
  x4 <- x3[,c('PlayerName','pos','points','value')]

  ## save dollar values as CSV file
  write.csv(x=x4, file=paste0(path,'/',file))

  ## return data.frame with dollar values
  return(x4)
}
```

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

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

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

  41 players are worth more than $20.

```r
sum(sum(x1$value > 20))
```

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

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

  David Montgomery

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

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

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

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

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


```r
sum(x2$value > 20)
```

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

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


```r
sum(x2[1:40,'pos']=='wr')
```

```
## [1] 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))
```

    1. How many players are worth more than $20? (1 point) 43 players

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

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

    1. How many quarterbacks (qb) are in the top 30? (1 point) 13 quarterbacks

```
sum(x3[1:30,'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)

```
# read haart data and load packages.
haart <- read_csv("haart.csv",show_col_types=FALSE)
haart <- data.frame(haart)
library(lubridate)
```

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

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
library(tidyverse)
```

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$init.date <- mdy(haart$init.date)
haart$last.visit <- mdy(haart$last.visit)
haart$date.death <- mdy(haart$date.death)
head(haart)
```

	m…	…	aids	cd4baseline	logvl	weight	hemoglobin	init.reg	init.date
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<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
6	0	34	0	157	*NA*	54.8856	*NA*	3TC,AZT,NVP	2003-12-02

6 rows | 1-10 of 13 columns

```
haart_t <- haart %>%
  mutate(ydiff_last.visit=year(last.visit)-year(init.date)) %>%
  mutate(ydiff_date.death=year(date.death)-year(init.date))

print('last.visit ( a count of years from initial date )')
```

```
## [1] "last.visit ( a count of years from initial date )"
```

```
table(haart_t$ydiff_last.visit)
```

```
##
##   0   1   2   3   4   5   6   7   9
## 174 178 194 165 153 112   9   3   1
```

```
print('date.death ( a count of years from initial date )')
```

```
## [1] "date.death ( a count of years from initial date )"
```

```
table(haart_t$ydiff_date.death)
```

```
##
##  0  1  2  3  4
## 72 28 11  3  3
```

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?

92 observations died in year 1.

```
# create indicator variable
haart_t <- haart_t %>% mutate(ind = as.numeric((date.death-init.date)<=365))

# calculate the number of observations died in year 1 (within 365 days)
sum(haart_t$ind, na.rm=TRUE)
```

```
## [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.

```
haart[,'followup'] <- NA
haart[,'ddiff_last.visit'] <- as.numeric(haart$last.visit - haart$init.date)
haart[,'ddiff_date.death'] <- as.numeric(haart$date.death - haart$init.date)

# create a initial followup variable
for (i in 1:length(haart[,'followup'])) {

  if (haart[i,'death']==0){ # if date.death is NA
    # in this case, last.visit should be used (because date.death would be NA)
    # if last.visit is NA, it would be just NA
    haart[i,'followup'] <- haart[i,'ddiff_last.visit']
  } else if (is.na(haart[i,'ddiff_last.visit'])) { # if last.visit is NA
    haart[i,'followup'] <- haart[i,'ddiff_date.death']
  } else{
    # compare which one is earlier between date.death & last.visit
    if (haart[i,'ddiff_last.visit']>haart[i,'ddiff_date.death']){
      haart[i, 'followup'] <- haart[i, 'ddiff_date.death']
      } else {
        haart[i, 'followup'] <- haart[i, 'ddiff_date.death']
      }
  }
}

# censor followup value
cut.val <- 365

# conduct censoring
# if its value is larger than cut.val make it as cut.val.
# if not, don't change it.
haart <- haart %>% mutate(followup=if_else(followup>cut.val, cut.val, followup))

# calculate quantile of followup variable (after censoring)
quantile(haart$followup)
```

```
##    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? 173 records

```
haart[,'loss'] <- NA
for (i in 1:length(haart[,'loss'])) {
  if (haart[i, 'death'] == 0 & haart[i, 'ddiff_last.visit'] < 365) {
    haart[i, 'loss'] <- 1
  } else
    haart[i, 'loss'] <- 0
}
sum(haart$loss)
```

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

```
init.reg <- as.character(haart[,'init.reg'])
init.reg.unique <- unique(unlist(strsplit(haart[,'init.reg'],",")))

# initialize
mat_drugs <- matrix(0, nrow=nrow(haart), ncol=length(init.reg.unique))

# create new columns and fill-in
for (i in 1:nrow(mat_drugs)){
  for (j in 1:length(init.reg.unique))
    mat_drugs[i,j] <- grepl(pattern=init.reg.unique[[j]], x=init.reg[[i]])
}
mat_drugs <- data.frame(mat_drugs)
names(mat_drugs) <- init.reg.unique
haart_merged <- cbind(haart, mat_drugs)

# check which regimen are found over 100 times?
colSums(haart_merged[,init.reg.unique]) > 100
```

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

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", show_col_types=FALSE)
haart2[,setdiff(names(haart), names(haart2))] <- 0
merged_data <- rbind(haart, haart2)
merged_data
```

… <dbl>	age <dbl>	ai… <dbl>	cd4baseline <dbl>	logvl <dbl>	weight <dbl>	hemoglobin <dbl>	init.reg <chr>	init.date <date>
1	25.00000	0	*NA*	*NA*	*NA*	*NA*	3TC,AZT,EFV	2003-07-01
1	49.00000	0	143	*NA*	58.06080	11.000000	3TC,AZT,EFV	2004-11-23
1	42.00000	1	102	*NA*	48.08160	1.000000	3TC,AZT,EFV	2003-04-30
0	33.00000	0	107	*NA*	46.00000	*NA*	3TC,AZT,NVP	2006-03-25
1	27.00000	0	52	4.000000	*NA*	*NA*	3TC,D4T,EFV	2004-09-01

… <dbl>	age <dbl>	ai… <dbl>	cd4baseline <dbl>	logvl <dbl>	weight <dbl>	hemoglobin <dbl>	init.reg <chr>	init.date <date>
0	34.00000	0	157	*NA*	54.88560	*NA*	3TC,AZT,NVP	2003-12-02
0	39.00000	0	65	*NA*	55.33920	11.000000	3TC,AZT,NVP	2004-02-06
1	31.00000	0	*NA*	*NA*	*NA*	*NA*	3TC,AZT,EFV	2001-09-06
1	52.00000	0	*NA*	*NA*	*NA*	*NA*	3TC,ABC,AZT	2002-08-13
1	23.00000	1	3	5.718295	*NA*	*NA*	3TC,DDI,NVP	2005-06-21

1-10 of 1,004 rows | 1-10 of 16 columns Previous **1** 2 3 4 5 6 … 101 Next

```
head(merged_data)
```

m… <dbl>	… <dbl>	aids <dbl>	cd4baseline <dbl>	logvl <dbl>	weight <dbl>	hemoglobin <dbl>	init.reg <chr>	init.date <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
6	0	34	0	157	*NA*	54.8856	*NA*	3TC,AZT,NVP	2003-12-02

6 rows | 1-10 of 17 columns

```
tail(merged_data)
```

	… <dbl>	age <dbl>	ai… <dbl>	cd4baseline <dbl>	logvl <dbl>	weight <dbl>	hemoglobin <dbl>	init.reg <chr>	init.date <date>
999	0	31.00000	0	102	*NA*	61.6896	11	3TC,AZT,NVP	2003-05-22
1000	0	40.00000	1	131	*NA*	46.2672	8	3TC,D4T,NVP	2003-07-03
1001	0	27.00000	0	232	*NA*	*NA*	*NA*	3TC,AZT,NVP	0012-01-03
1002	1	38.72142	0	170	*NA*	84.0000	*NA*	3TC,AZT,NVP	<NA>
1003	1	23.00000	*NA*	154	3.995635	65.5000	14	3TC,DDI,EFV	<NA>
1004	0	31.00000	0	236	*NA*	45.8136	*NA*	3TC,D4T,NVP	0012-03-03

6 rows | 1-10 of 17 columns