Bios 301: Homework 2

Question 1

20 points

Here is the function for the secant algorithm:

```
options(digits=10)
secant <- function(guess,tol=1e-8,max_count=1000) {
  x0 <- guess
  x1 <- x0+0.1
  count <- 0
  while(count < max_count){
    f0 <- cos(x0)-x0
    f1 <- cos(x1)-x1
    x2 <- x1-f1*(x1-x0)/(f1-f0)
    if(abs(x2-x1) <= tol){
      cat("Converged! x=", x2)
      return(x2)
      break}
    x0 <- x1
    x1 <- x2
    count <- count+1
  }
}
system.time(secant(1))
```

```
## Converged! x= 0.7390851332
```

```
##    user  system elapsed
##   0.000   0.000   0.001
```

And here is the function that implement Newton-Raphson algorithm:

```
newton_raphson <- function(guess,tol=1e-8,max_count=1000) {
  x <- guess
  count <- 0
  while(count < max_count){
    f <- cos(x) - x
    df <- -sin(x)-1
    xnew <- x-f/df
    if(abs(x-xnew) <= tol){
      cat("Converged! x=",xnew)
      return(xnew)
      break}
    x <- xnew
    count <- count+1
  }
}
system.time(newton_raphson(1))
```

```
## Converged! x= 0.7390851332
```

```
##    user  system elapsed
##       0       0       0
```

From the results above, I think there is no great difference of the speed between these 2 methods.

Question 2

18 points

```
haart.df <- read.csv("haart.csv")
```

1.

```
haart.df$init.date <- as.Date(haart.df$init.date,format="%m/%d/%Y")
haart.df$last.visit <- as.Date(haart.df$last.visit,format="%m/%d/%Y")
haart.df$date.death <- as.Date(haart.df$date.death,format="%m/%d/%Y")
```

2.

```
haart.df["death within a year"] <- +((haart.df$date.death-haart.df$init.date)<365)
```

3.

```
haart.df["follow up time"] <- NA
j <- 1
for (i in haart.df$init.date) {
  if (is.na(haart.df$last.visit[j])) {
    haart.df[j,14] <- haart.df$date.death[j]-i
  } else if(is.na(haart.df$date.death[j])) {
    haart.df[j,14] <- haart.df$last.visit[j]-i
  } else {
    haart.df[j,14] <- min(haart.df$last.visit[j],haart.df$date.death[j])-i
  }
"  if (haart.df[j,14]>365){
    print(haart.df[j,])
  }"
  j <- j+1
}
```

4.

```
haart.df["Unknown after 1 year"] <- +(haart.df[,14]<=365)
```

5.

```
temp<-strsplit(as.character(haart.df$init.reg),",")
nmax<-max(sapply(temp,length))
temp2 <- do.call(rbind, lapply(temp,`[`,seq_len(nmax)))
haart.df <- cbind(haart.df, temp2)
```

6.

```
addition.df <- read.csv("haart2.csv")
addition.df[, setdiff(names(haart.df),names(addition.df))] <- NA
haart.df.df <- rbind(haart.df, addition.df)
```

Question 3

12 points

```
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)) {
  k <- read.csv('proj_k14.csv', header=TRUE, stringsAsFactors=FALSE)
  qb <- read.csv('proj_qb14.csv', header=TRUE, stringsAsFactors=FALSE)
  rb <- read.csv('proj_rb14.csv', header=TRUE, stringsAsFactors=FALSE)
  te <- read.csv('proj_te14.csv', header=TRUE, stringsAsFactors=FALSE)
  wr <- read.csv('proj_wr14.csv', header=TRUE, stringsAsFactors=FALSE)
  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
  cols <- c(cols, 'pos')
  x <- rbind(k[,cols], qb[,cols], rb[,cols], te[,cols], wr[,cols])
  names(x) <- gsub('[.]', '', names(x))
  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))])
  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')
  if (posReq["k"]>0){
    x2[k.ix, 'marg'] <- x2[k.ix,'points'] - x2[k.ix[posReq["k"]*nTeams],'points']
  }
  if (posReq["qb"]>0){
    x2[qb.ix, 'marg'] <- x2[qb.ix,'points'] - x2[qb.ix[posReq["qb"]*nTeams],'points']
  }
  if (posReq["rb"]>0){
    x2[rb.ix, 'marg'] <- x2[rb.ix,'points'] - x2[rb.ix[posReq["rb"]*nTeams],'points']
  }
  if (posReq["te"]>0){
    x2[te.ix, 'marg'] <- x2[te.ix,'points'] - x2[te.ix[posReq["te"]*nTeams],'points']
  }
  if (posReq["wr"]){
    x2[wr.ix, 'marg'] <- x2[wr.ix,'points'] - x2[wr.ix[posReq["wr"]*nTeams],'points']
  }
  x3 <- x2[x2[,'marg'] >= 0 & !is.na(x2[,'marg']),]
  x3 <- x3[order(x3[,'marg'], decreasing=TRUE),]
  rownames(x3) <- NULL
  x3[,'value'] <- x3[,'marg']*(nTeams*cap-nrow(x3))/sum(x3[,'marg']) + 1
  x4 <- x3[,c('PlayerName','pos','points','value')]
  write.csv(x4,file = file)
  return(x4)
}
```

1.

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

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

```
sum((x1[,"value"]>20))
```

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

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

```
x1[x1[,'pos'] == 'rb',][15,]
```

```
##      PlayerName pos points       value
## 40 Toby Gerhart  rb 170.75 23.21051281
```

2.

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

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

```
sum((x2[,"value"]>20))
```

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

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

```
sum(x2[x2[,'pos'] == 'wr','value']> x2[41,'value'])
```

```
## [1] 12
```

3.

```
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
```

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

```
sum(x3[x3[,'pos'] == 'qb','value']> x3[31,'value'])
```

```
## [1] 16
```

Question 4

5 bonus points

```
objs <- mget(ls("package:base"), inherits = TRUE)
funs <- Filter(is.function, objs)
```

1. Which function has the most arguments? (3 points)

```
indx <- integer(length(funs))
i <- 1
while(i < length(funs)){
  indx[i] <- length(formals(funs[[i]]))
  i <- i+1
}
funs[which(indx==max(indx))]
```

```
## $scan
## function (file = "", what = double(), nmax = -1L, n = -1L, sep = "",
##     quote = if (identical(sep, "\n")) "" else "'\"", dec = ".",
##     skip = 0L, nlines = 0L, na.strings = "NA", flush = FALSE,
##     fill = FALSE, strip.white = FALSE, quiet = FALSE, blank.lines.skip = TRUE,
##     multi.line = TRUE, comment.char = "", allowEscapes = FALSE,
##     fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
## {
##     na.strings <- as.character(na.strings)
##     if (!missing(n)) {
##         if (missing(nmax))
##             nmax <- n/pmax(length(what), 1L)
##         else stop("either specify 'nmax' or 'n', but not both.")
##     }
##     if (missing(file) && !missing(text)) {
##         file <- textConnection(text, encoding = "UTF-8")
##         encoding <- "UTF-8"
##         on.exit(close(file))
##     }
##     if (is.character(file))
##         if (file == "")
##             file <- stdin()
##         else {
##             file <- if (nzchar(fileEncoding))
##                 file(file, "r", encoding = fileEncoding)
##             else file(file, "r")
##             on.exit(close(file))
##         }
##     if (!inherits(file, "connection"))
##         stop("'file' must be a character string or connection")
##     .Internal(scan(file, what, nmax, sep, dec, quote, skip, nlines,
##         na.strings, flush, fill, strip.white, quiet, blank.lines.skip,
##         multi.line, comment.char, allowEscapes, encoding, skipNul))
## }
## <bytecode: 0x1061566d8>
## <environment: namespace:base>
```

2. How many functions have no arguments? (2 points)

```
sum(indx==0)
```

```
## [1] 222
```