# Bios 6301: Assignment 4

*Wooyeol Lee*

*October 27, 2015*

### Question 1

**Write a function that implements the secant algorithm.** Validate your program by finding the root of the function $f(x) = \cos(x) - x$. Compare its performance with the Newton-Raphson method – which is faster, and by how much? For this example $f'(x) = -\sin(x) - 1$.

```
##scant method
x1<- 1
x2<- 10000
iter<- 1
while(abs(cos(x2)-x2)>1e-6 && iter <10) {
f1<- cos(x1)-x1
f2<- cos(x2)-x2
newx1 <- x2
newx2 <- x2 - f2*(x2-x1)/(f2-f1)
x1<- newx1
x2<- newx2
iter <- iter +1
}
x1
```

```
## [1] 0.7389518
```

```
x2
```

```
## [1] 0.7390853
```

```
f1
```

```
## [1] 0.009049969
```

```
f2
```

```
## [1] 0.0002230807
```

```
iter
```

```
## [1] 6
```

```
### Newton-Raphson method
guess <- 10000
iter <-1
while(abs(cos(guess)-guess)>1e-6 && iter <1000) {
  f <- cos(guess)-guess
  fp <- -sin(guess)-1
  guess <- guess -f/fp
  iter <- iter +1
}
guess
```

```
## [1] 0.7390851
```

```
f
```

```
## [1] -1.076609e-05
```

```
iter
```

```
## [1] 819
```

The scant is faster than NR.

### Question 2

1. The instructor should be able to easily import and run your program (function), and obtain output that clearly shows how the game progressed. Set the RNG seed with `set.seed(100)` and show the output of three games. (lucky 13 points)

```
crap <- function(rng) {
  set.seed(rng)
  for(i in 1:3) {
    gnum<- paste("game",i, sep="")
    print(gnum)
  x <- sum(ceiling(6*runif(2)))
  if(x==7 | x==11) {
    print("You won.")
    print(x)}
  else {
      y <- sum(ceiling(6*runif(2)))
          if(y==7 | y==11) {
            print("You won.")}
      else {
        print ("You lost.")}
      print(x)
      print(y)}
}
}
crap(100)
```

```
## [1] "game1"
## [1] "You lost."
## [1] 4
## [1] 5
## [1] "game2"
## [1] "You lost."
## [1] 6
## [1] 8
## [1] "game3"
## [1] "You lost."
## [1] 6
## [1] 10
```

1. Find a seed that will win ten straight games. Consider adding an argument to your function that disables output. Show the output of the ten games. (5 points)

```
crap <- function(rng, output=T) {
  set.seed(rng)
  nwin<-0
  iter<-0
  max_iter<-100000
     ###loop
    while((sum(nwin)<10) && (rng<max_iter)) {
      ##record results
      rec.x<-0
      rec.y<-0
      ###10 trials
    for (i in 1:10) {
    x <- sum(ceiling(6*runif(2)))
    rec.x[i]<-x
    if(x==7 | x==11) {
      nwin[i]<-1}
    else {
      y <- sum(ceiling(6*runif(2)))
      rec.y[i]<-y
      if(y==7 | y==11) {
      nwin[i]<-1}
      else {
      nwin[i]<-0}
     }
    iter<- iter+1
  }###
    rng<- rng+1
    }
  print(rng)
  if(output==T) {
  print(rec.x)
  print(rec.y)}
}

crap(1,T)
```

```
## [1] 31018
##  [1]  9 11  8 11  7  7  9  7  8  7
## [1] 11 NA  7 NA NA NA 11 NA  7
```

### Question 3

Define the function as such (6 points):

```r
path<- paste("C:/Users/Wooyeol/Dropbox/me/coursework/fall2015/statistical computing/homework/")

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)) {
  path <- setwd(path)
  ## read in CSV files
  k <- read.csv('proj_k15.csv', header=TRUE, stringsAsFactors=FALSE)
  qb <- read.csv('proj_qb15.csv', header=TRUE, stringsAsFactors=FALSE)
  rb <- read.csv('proj_rb15.csv', header=TRUE, stringsAsFactors=FALSE)
  te <- read.csv('proj_te15.csv', header=TRUE, stringsAsFactors=FALSE)
  wr <- read.csv('proj_wr15.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
  x <- rbind(k[,cols], qb[,cols], rb[,cols], te[,cols], wr[,cols])
  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')
  x2[k.ix, 'marg'] <- x2[k.ix,'points'] - x2[k.ix[nTeams*posReq['k']],'points']
  x2[qb.ix, 'marg'] <- x2[qb.ix,'points'] - x2[qb.ix[nTeams*posReq['qb']],'points']
  x2[rb.ix, 'marg'] <- x2[rb.ix,'points'] - x2[rb.ix[nTeams*posReq['rb']],'points']
  x2[te.ix, 'marg'] <- x2[te.ix,'points'] - x2[te.ix[nTeams*posReq['te']],'points']
  x2[wr.ix, 'marg'] <- x2[wr.ix,'points'] - x2[wr.ix[nTeams*posReq['wr']],'points']
  x3 <- x2[x2[,'marg'] >= 0,]
  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(file=file,x4)
  return(x4)
}
```

1. Call ‘x1 <- ffvalues(‘.’)

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

```r
x1 <- ffvalues('.')
sum(x1$value>20)
```

```
## [1] 40
```

```
40 plyaers are more than $20.

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

```r
x1[which(x1$pos=='rb'),][15,1]
```

```
## [1] "Melvin Gordon"
```

```
Melvin Gordon
```

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

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

```r
x2 <- ffvalues(getwd(), '16team.csv', nTeams=16, cap=150)
sum(x2$value>20)
```

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

```
41 plyaers are worth more than $20.

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

```r
top40<-x2[1:40,]
length(top40[which(top40$pos=='wr'),1])
```

```
## [1] 13
```

~~~
13 wide receivers are in the top40.

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)

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

### Question 4

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

n.args <- length(as.list(args(names(funs)[1])))-1
leng<-0
for(i in 1:1196) {
leng[i]<-length(as.list(args(names(funs[i]))))-1
}
n.args<- cbind(names(funs),leng)
```

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

```
max(n.args[,2])
```

```
## [1] "9"
```

```
which(n.args[,2]==9)
```

```
## [1]  645  795 1084
```

```
n.args[645,]
```

```
##                 leng
## "library"      "9"
```

```
n.args[795,]
```

```
##                             leng
## "print.default"             "9"
```

```
n.args[1084,]
```

```
##               leng
## "system"      "9"
```

library, print.defalult, system have 9 arguments.

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

```
length(n.args[which(n.args[,2] <1),])
```

```
## [1] 146
```

146 functions have no arguments.