Project 6

Alisha

5/6/2020

Question 1,2

States = s , o , t
s = self score; o = opponent's score, t = current time value Pn is probability of rolling n

Action = Roll or Hold

Bellman equation:

V(s,100,t) = 0; V(100,o,t) = 1;

V(s,p,t) = max[p1*(1-V(o,s+1,0))* + *p2*V(s,o,t+2) + p3*V(s,o,t+3)* + *p4*V(s,o,t+4) + p5*V(s,o,t+5)* + *p6*V(s,o,t+6) , (1- V(o,s+t,0))]

Question 3,4

```
gameStrategy <- function(GoalP){
  GoalP = 100
  score = GoalP+1
  # Generate all combinations
  combinations = expand.grid(seq(score-2),seq(score-1))
  combinations$sum = combinations[[1]] + combinations[[2]]
  # Finding max sum
  combinations = combinations[order(combinations$sum, decreasing = TRUE),]

  U = array(NA, dim=c(score+5,score+5,score+5))
  V = array(NA, dim=c(score+5,score+5,score+5))


  # boundary conditions
  # you loose if opponent reaches goal first
  V[seq(1,score-1),score:(score+5),] = 0

  # you win if you have 100+ points then
  V[score:(score+5),,] = 1


  # Fill UV matrix based on game rules
  for(r in 1:length(combinations$sum)){
      for(t in (score):1){
```

```
        o = combinations[r,2]
        s = combinations[r,1]

        U[s,o,t] = which.max( c( ( (1/6)*(1-V[min(o,score),s+1,1]) + (1/6)*V[min(s,score),o,min(t+2,sco
                          + (1/6)*V[min(s,score),o,min(t+4,score)] + (1/6)*V[min(s,score),o,min(t+5,sco
                          + (1/6)*V[min(s,score),o,min(t+6,score)] ), 1-V[min(o,score),min(s+max(1,t-1)

        V[s,o,t] = max( ( (1/6)*(1-V[min(o,score),s+1,1]) + (1/6)*V[min(s,score),o,min(t+2,score)] + (1/
                          + (1/6)*V[min(s,score),o,min(t+4,score)] + (1/6)*V[min(s,score),o,min(t+5,sco
                          + (1/6)*V[min(s,score),o,min(t+6,score)] ), 1-V[min(o,score),min(s+max(t-1,1)


      }
  }
  save(list = c('V','U'),file = 'VUfile.Rdata')

}
```

```
#gameStrategy(100)
```