Assignment 3

Nicholas Colonna

3/2/2018

#PROBLEM 1  
#Create Vectors  
v1 <- c(1,3,5,7,9,11,13,15,17,19)  
v1

## [1] 1 3 5 7 9 11 13 15 17 19

v2 <- seq(0,20,by=2)  
v2

## [1] 0 2 4 6 8 10 12 14 16 18 20

#Explicit Conversion  
v2 <- as.character(v2)  
v2

## [1] "0" "2" "4" "6" "8" "10" "12" "14" "16" "18" "20"

v3 <- as.logical(v1)  
v3

## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

#Matrix and List  
matrix1 <- matrix(v1, nrow=2, ncol=5, byrow=TRUE)  
matrix1

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 3 5 7 9  
## [2,] 11 13 15 17 19

myFirstList <- list(v1,v2,v3)  
names(myFirstList) <- c("Number","Character","Logical")  
myFirstList

## $Number  
## [1] 1 3 5 7 9 11 13 15 17 19  
##   
## $Character  
## [1] "0" "2" "4" "6" "8" "10" "12" "14" "16" "18" "20"  
##   
## $Logical  
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

#PROBLEM 2  
smallest\_number <- 2520  
testDiv<-vector()  
for(i in 1:20){  
 testDiv[i] <- smallest\_number %% i  
}  
while(sum(testDiv) != 0){  
 smallest\_number <- smallest\_number + 2520  
 for(i in 1:20){  
 testDiv[i] <- smallest\_number %% i  
 }  
}  
smallest\_number

## [1] 232792560

#PROBLEM 3  
JPM2018 <- read.csv("JPM.csv", header=TRUE)  
subJPM <- JPM2018[2:5]  
subJPM

## Open High Low Close  
## 1 109.26 109.55 107.78 108.34  
## 2 108.15 108.68 107.70 108.50  
## 3 108.72 109.63 108.49 109.05  
## 4 109.47 110.70 109.39 110.25  
## 5 110.67 110.93 110.05 110.84  
## 6 111.65 112.85 110.84 112.67  
## 7 111.51 113.43 111.07 112.27  
## 8 111.89 113.30 111.31 112.99  
## 9 112.76 113.72 112.27 113.26  
## 10 113.94 114.34 112.80 113.01  
## 11 112.66 114.39 112.50 114.33  
## 12 113.67 114.64 113.35 114.21  
## 13 114.86 116.00 114.66 115.67  
## 14 116.04 116.17 115.08 115.70  
## 15 115.70 116.32 114.96 116.32

colMeans <- sapply(subJPM, mean)  
colMeans

## Open High Low Close   
## 112.0633 112.9767 111.4833 112.4940

rowMeans <- matrix(apply(subJPM, 1, mean), nrow=3, ncol=5, byrow=TRUE)  
rowMeans

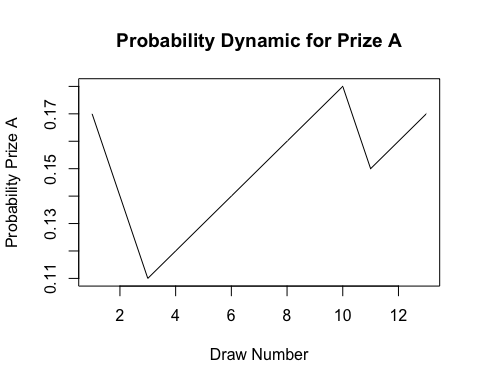
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 108.7325 108.2575 108.9725 109.9525 110.6225  
## [2,] 112.0025 112.0700 112.3725 113.0025 113.5225  
## [3,] 113.4700 113.9675 115.2975 115.7475 115.8250

#PROBLEM 4  
#1 How many draws on average to get one level A prize  
numDraws <- rgeom(1000, 0.2)  
mean(numDraws)

## [1] 4.041

#2 Probability of A within 5 draws  
theoreticalDraws <- .2\*(.8^0) + .2\*(.8^1) + .2\*(.8^2) + .2\*(.8^3) + .2\*(.8^4)  
theoreticalDraws

## [1] 0.67232

#3 Probability Dynamic  
numWins <- 0  
pA <- 0.2  
drawNum <- 0  
pVector <- vector()  
while(numWins < 5){  
 if(rbinom(1,1,pA) == 1){  
 numWins <- numWins + 1  
 pVector[drawNum] <- pA  
 pA <- pA - 0.03  
 }else{  
 pVector[drawNum] <- pA  
 pA <- pA + 0.01  
 }  
 drawNum<-drawNum+1  
}  
  
plot(pVector, type="l", main="Probability Dynamic for Prize A", xlab="Draw Number", ylab="Probability Prize A")

#4 Probability Dynamic Function  
probabilityDynamic <- function(winGoal){  
 numWins <- 0  
 pA <- 0.2  
 drawNum <- 0  
 pVector <- vector()  
 while(numWins < winGoal){  
 if(rbinom(1,1,pA) == 1){  
 numWins <- numWins + 1  
 pVector[drawNum] <- pA  
 pA <- pA - 0.03  
 }else{  
 pVector[drawNum] <- pA  
 pA <- pA + 0.01  
 }  
 drawNum<-drawNum+1  
}  
  
plot(pVector, type="l", main="Probability Dynamic for Prize A", xlab="Draw Number", ylab="Probability Prize A")  
}  
  
probabilityDynamic(10)

