Code

Lab 7 | Jon Lee

Lab 7 | BINF 6310 | Spring 2020 | Jon Lee

1. Consider a tournament with the following prize structure:

Number of Wins Payoff ($) 0 1.45 1 1.72 2 2.24 3 2.76 4 3.55 5 4.60 6 5.65 7 6.75

You play in the tournament until you get three losses or 7 wins.

Make a graph of the expected value of the tournament (on the y-axis) vs. the probability of winning each game (on the x-axis). If the tournament costs $3.75, at what win percentage does the expected value exceed the cost of the tournament? (Plot 3.75 as a horizontal line ranging from 0 to 1).

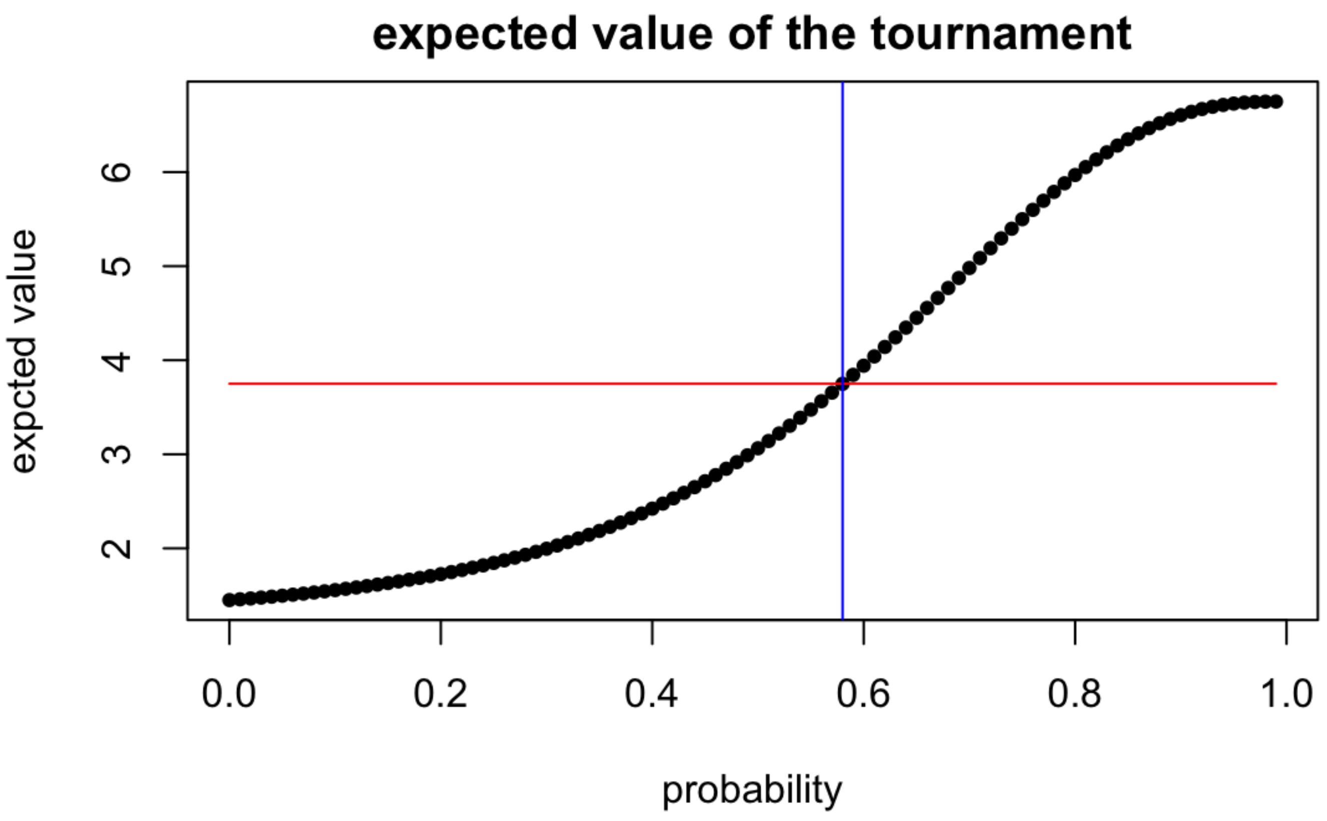
(hint: Calculate the probability of 7 wins a 1 – sum(prob of all other outcomes)).

Hide

```
#graph of expected values for tourney
#assuming the probability of a win is 0.5
entryPrice <- 3.75
numberOfWins <- 0:7
payoffPerWin <- c(1.45, 1.72, 2.24, 2.76, 3.55, 4.60, 5.65, 6.75)
probOfWin <- seq(0, 0.99, 0.01)
#did not include 1 becuase the dnbinom function's probability is not defined at a probab
ility of 0

expectVals <- vector()

for(i in 1:length(probOfWin))
{
  probs <- vector()
  expect <- vector()

  for(j in 1:(length(payoffPerWin)-1))
  {
    probs[j] <- dnbinom((j-1), 3, (1-probOfWin[i]))
    expect[j] <- probs[j]*payoffPerWin[j]
  }

  probs[8] <- 1-sum(probs[1:7])
  expect[8] <- probs[8]*payoffPerWin[8]

  expectVals[i] <- sum(expect)
}

plot(probOfWin, expectVals, xlab = "probability", ylab = "expcted value", main = "expect
ed value of the tournament", pch = 20)
lines(probOfWin, rep(3.75, length(probOfWin)), col = "red")
```

Hide

```
abline(v = 0.58, col= "blue")
```



Answer: At a win percentage (rate) of about 0.58 or 58% do we see the epected value of the tournament exceeding the cost.

Collabroated with David B. and Conor F. on Question 1

2. Generate a simulated dataset in which the variance is a function of the mean but the null hypothesis of differential expression is always true. The following code will accomplish this:

Hide

```
numRows = 3000
numCols = 20
for( i in 1:numCols)

myFrame <- data.frame(1:numRows)

#initiate the data.frame with the correct # of rows to suppress error messages.
#likely, there are much better ways to do this!
names(myFrame)[1] <- "tempColumn"

for( i in 1: numCols)
{
        vals <- vector(length=numRows)

        for( j in 1:numRows)
        {
            aMean = j /10
            aMean = max( aMean,5)
            aVar = aMean+ 5* aMean
            aVal = round( max( rnorm(1,mean=aMean,sd=sqrt(aVar)), 1))
            vals[j] = aVal
        }

        colName <- paste( "sample" , i ,sep="")

        myFrame[[colName]] = vals
}

myFrame["tempColumn"] <- NULL
row.names(myFrame) <- paste("Gene_",1:numRows,sep="")
```

For each row in the spreadsheet, we will consider the first 10 columns to be “case” and the last 10 to be “control”. For each row in the spreadsheet, run a t-test and generate a p-value for the null hypothesis that the case and control samples follow the same distribution (see slide 5 of lecture 11 for code to do this). Show your code for all answers.

Hide

```
allPValsTTest <- vector()

for(i in 1:nrow(myFrame))
{
  vals1 <- as.numeric(myFrame[i,1:10])
  vals2 <- as.numeric(myFrame[i,11:20])
  allPValsTTest[i] <- t.test(vals1, vals2)$p.value
}
```

Hide

```
hist(allPValsTTest, xlab = "t-test p values", main = "Histogram of t-test")
```

A. Use a simple threshold of $p < 0.05$. How many significant hits would you expect to find if the null hypothesis is always true? How many hits did you actually find in your run?

Hide

```
#how many expected hits
totalNumOfPVals <- length(allPValsTTest)
expectNumoOfPVals <- 0.05*totalNumOfPVals
print(expectNumoOfPVals)

#how many actual hits
countPVals <- 0
for(i in 1:length(allPValsTTest))
{
  if(allPValsTTest[i] < 0.05)
  {
    countPVals =  countPVals + 1
  }
}
print(countPVals)
```

Answer: If the null hypothesis is true, we would expect to see 5% of the p-values to be significant, becuase the resulting p-values should be uniform. From the data we saw 156 p-values were significant, which is approximately 50, which confirms what we expected.

B. What is the Bonferroni adjusted p-value threshold? What percentage of the time would you expect to see a significant gene under this threshold? How many genes in fact do you see significant at this threshold?

Hide

```
#Bonferroni adjusted p-value threshold
adjustedPVal <- 0.05/totalNumOfPVals
print(adjustedPVal)

#how many expected hits
adjustedExpectedPVals <- adjustedPVal*totalNumOfPVals
print(adjustedExpectedPVals)

#how many actual hits
countAdjustPVal <- 0
for(i in 1:length(allPValsTTest))
{
  if(allPValsTTest[i] < adjustedPVal)
  {
    countAdjustPVal =  countAdjustPVal + 1
  }
}
print(countAdjustPVal)
```

Answer: Our new adjusted p-value threshold via Bonferroni is 1.67e-05, and we would expect to see none of our p-values to be below this threshold, and when looking at the data we see that this is true.

C. Next use a BH FDR corrected threshold of $p < 0.05$ (which R can do for you with p.adjust(pvals,method="BH") if pvals is a vector that holds the unadjusted p-values). How many hits do you find with the BH adjusted p-values.

Hide

```
#adjust p-values via BH method
adjustedPVals <- p.adjust(allPValsTTest, method = "BH")

#how many actual hits
countAdjustPVals <- 0
for(i in 1:length(adjustedPVals))
{
  if(adjustedPVals[i] < 0.05)
  {
    countAdjustPVals =  countAdjustPVals + 1
  }
}
print(countAdjustPVals)
```

```
[1] 0
```

Answer: After adjusting our p-values using the BH method, we find that none of our p-value lie beyondthe 0.05 threshold and therefore are significant.