

N-Play Variability

davegoblue

March 19, 2016

Overview and Synopsis

This routine analyzes variability of N-play video poker results based on simulation data made available by the Wizard of Odds <http://wizardofodds.com/games/video-poker/appendix/2/> (<http://wizardofodds.com/games/video-poker/appendix/2/>). The game simulated was 9/6 Jacks or Better, and this routine looks at the impact of playing 5,000 hands with 1/3/5/10/50/100 lines per hand. The code broadly copies routines from Exercise002_v003.R, but with the intent of storing figures and summaries.

Data Processing

Coding Routine

The Wizard of Odds file was pre-processed in Excel and saved as a CSV once per each of the desired N-play outcomes. The file is read in, parsed to eliminate any unwanted values, and then assessed based on random draws from $\text{uniform}(0,1)$. Results are stored for the final outcome and the minimum cumulative outcomes.

Below is the routine for creating the data. It has been converted to a function so it can be re-run multiple times.

First, a function is created to read and process files for mapping probabilities to outcomes.

```

getBaseOutcomes <- function(myFileName="BaseOutcomes.csv", myDelete=NULL, force
EQ=FALSE) {

  if (file.exists(myFileName)) {
    baseOutcomes <- read.csv(myFileName,stringsAsFactors = FALSE)
    if (ncol(baseOutcomes) != 2) { stop("Error in CSV file, should have exa
ctly 2 columns") }
    colnames(baseOutcomes) <- c("probs","outcomes")
  } else {
    baseOutcomes <- data.frame(probs=c(0.01,0.02,0.05,0.18,0.24,0.50),outco
mes=c(10,5,2,1,0,-1))
  }

  baseOutcomes <- baseOutcomes[baseOutcomes$probs != 0,] ## Can have zeroes a
s inputs -- ignore those

  if ( forceEQ ) {
    pDelta <- sum(baseOutcomes$probs) - 1
    if ( abs(pDelta) < 0.0000001 &
        abs(pDelta) / baseOutcomes[nrow(baseOutcomes),]$probs < 0.1
        )
    {
      print(paste0("Modifying probablities ",paste0(baseOutcomes[nrow(bas
eOutcomes),],collapse=" ")))
      baseOutcomes[nrow(baseOutcomes),]$probs <- baseOutcomes[nrow(baseOu
tcomes),]$probs - pDelta
      print(paste0("New probablities ",paste0(baseOutcomes[nrow(baseOutco
mes),],collapse=" ")))
    }
  }

  if (sum(baseOutcomes$probs)!=1 | min(baseOutcomes$probs) < 0 |
      sum(is.na(baseOutcomes$probs)) > 0 | sum(is.na(baseOutcomes$outcomes))
> 0) {
    stop("Please resolve the issue with inputs for probs and outcomes, abor
ting")
  }

  ## Store the original value read in as outcomes
  baseOutcomes$oldOutcomes <- baseOutcomes$outcomes

  ## Null the baseOutcomes$outcomes where outcomes >= X
  if (!is.null(myDelete)) {
    myCond <- parse(text=paste0("baseOutcomes$outcomes",myDelete))
    baseOutcomes$outcomes[eval(myCond)] <- 0
    print(paste0("Converted all cases where ",myCond," to baseOutcomes$outc
omes = 0"))
  }
}

```

```
baseMean <- sum(baseOutcomes$probs*baseOutcomes$outcomes)
baseVar <- sum(baseOutcomes$probs*(baseOutcomes$outcomes-baseMean)^2)

print(paste0("Probabilities sum to 1. Outcomes has mean ",format(baseMean,
digits=3),
            " and variance ",format(baseVar,digits=3)))

return(baseOutcomes)
}
```

Second, a function is created to draw the random variables and calculate the outcomes database.

```

calcOutcomes <- function(baseOutcomes=baseOutcomes,nPlay=1) {

  ## Allow nTrials, nPerTrial, and myHurdle to come from global environment
  print(paste0("Running ",nPlay,"-play with nTrials=",nTrials,
               " nPerTrial=",nPerTrial," and hurdle ",myHurdle
               )
        )

  myCDF <- numeric(nrow(baseOutcomes)+1)
  myCDF[1] <- 0

  for ( intCtr in 1:nrow(baseOutcomes) ) {
    myCDF[intCtr+1] <- myCDF[intCtr] + baseOutcomes$probs[intCtr]
  }

  mtxCumOutcomes <- matrix(baseOutcomes$outcomes[findInterval(matrix(data=run
                                                                    if(nTrials*nPerTrial,0,1),
                                                                    nrow=nPerTrial,
                                                                    ncol=nTrials
                                                                    ),
                                                                    myCDF,rightmost.closed=TRUE
                                                                    )
                                                                    ],
                                                                    nrow=nPerTrial,
                                                                    ncol=nTrials
                                                                    )

  print(paste0("Outcomes across ",nTrials*nPerTrial," draws of ",nPlay,"-play have mean: ",
               format(mean(mtxCumOutcomes),digits=3)," and variance: ",
               format(sd(mtxCumOutcomes)^2,digits=3)
               )
        )

  mtxCumOutcomes <- apply(mtxCumOutcomes,2,FUN=cumsum) ## About 2.5 seconds for 12,000 x 5,000
  maxPerTrial <- apply(mtxCumOutcomes,2,FUN=max) ## About 1.0 seconds for 12,000 x 5,000
  minPerTrial <- apply(mtxCumOutcomes,2,FUN=min) ## About 1.0 seconds for 12,000 x 5,000
  lastPerTrial <- as.numeric(mtxCumOutcomes[nrow(mtxCumOutcomes),])
  dfSummary <- data.frame(myTrial = 1:nTrials, myMax = maxPerTrial, myMin = minPerTrial,
                          myLast = lastPerTrial, myCond = FALSE, myN_Cond = NA, myVal_Cond = NA

```

```
    )

    dfSummary$myCond <- eval(parse(text=paste0("dfSummary$myMin",myHurdle)))

    foo <- function(x) {
      which(eval(parse(text=paste0("x",myHurdle))))[1]
    }

    dfSummary$myN_Cond <- apply(mtxCumOutcomes,2,FUN=foo)  ## About 2.5 second
s for 12,000 x 5,000

    for ( intCtr in 1:nTrials ) {
      dfSummary$myVal_Cond[intCtr] <- mtxCumOutcomes[dfSummary$myN_Cond[intCtr],dfSummary$myTrial[intCtr]]
    }

    return(dfSummary)
}
```

Additionally, a function is created to graph the data and store the outputs.

```

graphSummary <- function(graphData, nPlay=1) {

  graphData <- graphData[order(-graphData$myCond, graphData$myN_Cond, -graphData$myLast),]
  print(summary(graphData))

  ## Have the x and y units auto-calculated
  minX <- min(graphData$myMin)          ## Find most negative element
  maxX <- max(0, graphData$myLast)      ## Find most positive element
  t (use 0 if all are negative)
  powX <- log10(max(1, abs(minX), abs(maxX))) ## Find rough "power" of data

  unitX <- 10^(round(powX-0.5,0)-1)      ## If thousands, use hundreds; if hundreds, use tens; etc.
  minX <- unitX*(floor(minX/unitX)-1)    ## Round to similar units as unitX
  maxX <- unitX*(ceiling(maxX/unitX)+1)  ## Round to similar units as unitX

  hist(graphData$myMin,
        col=rgb(1,0,0,.25),
        main=paste0("Results: ",nTrials," Trials (",nPerTrial," ",
                     nPlay,"-play draws per trial)"
                     ),
        xlab="Units", ylab="N Trials",
        breaks=seq(minX,maxX,by=unitX),
        xlim=c(minX, maxX)
        )

  hist(graphData$myLast,col=rgb(0,0,1,.25),
        breaks=seq(minX,maxX,by=unitX),
        xlim=c(minX,maxX),
        add=TRUE
        )

  legend("topright",col=c(rgb(1,0,0,.25),rgb(0,0,1,.25),rgb(0.5,0,0.5,.5)),
        legend=c("Minimum","Final","Overlap"),pch=20,pt.cex=2
        )

}

```

Prepare the global parameters

Finally, the key global parameters are set.

```
nTrials <- 2000
nPerTrial <- 5000
myHurdle <- "<=-500"
```

Results

The simulation is repeated for each of the desired N-play outcomes. In this case, we have run the routine for 1/3/5/10/50/100 play.

Results for 1-play

```
## Run for 1-play
baseOutcomes <- getBaseOutcomes(myFileName="Play001Outcomes.csv", forceEQ=TRUE)
```

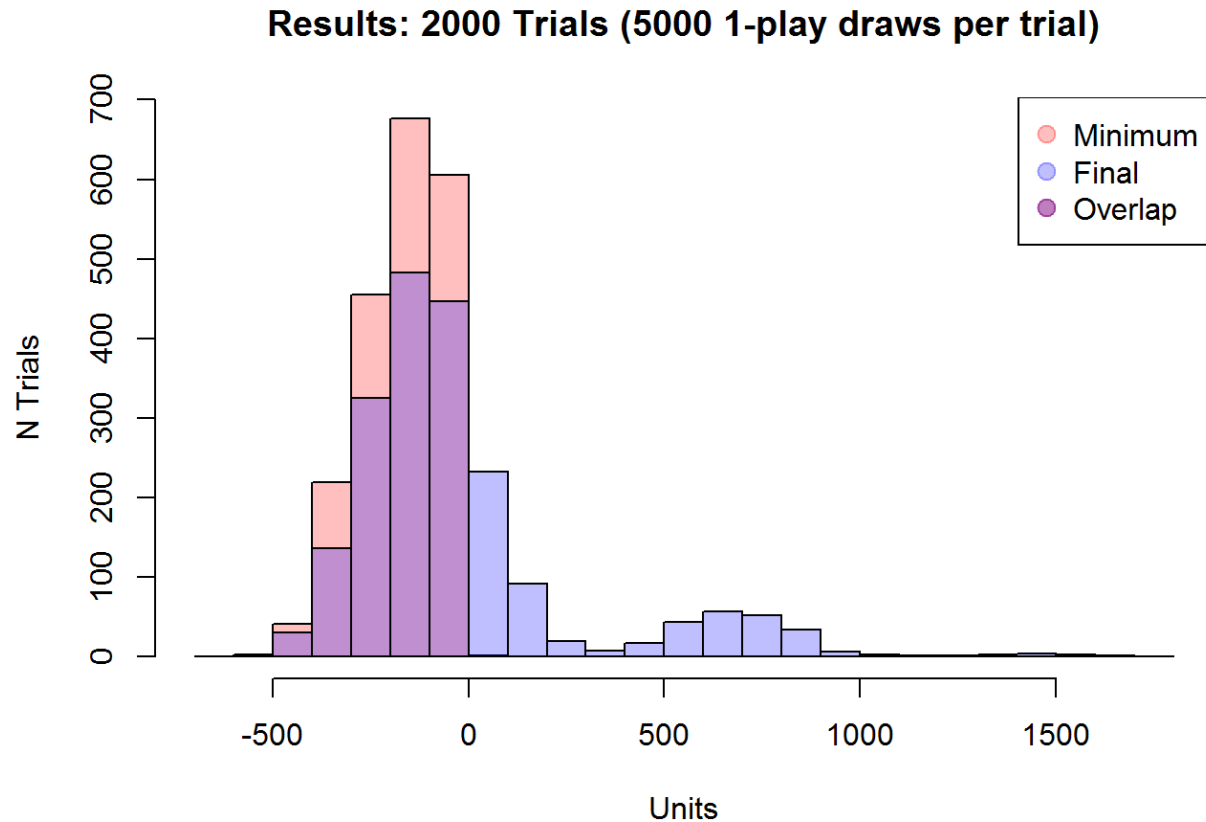
```
## [1] "Modifying probablities 0.54543467 -1"
## [1] "New probablities 0.54543466 -1"
## [1] "Probabilities sum to 1. Outcomes has mean -0.00456 and variance 19.5"
```

```
dfSummary <- calcOutcomes(baseOutcomes=baseOutcomes, nPlay=1)
```

```
## [1] "Running 1-play with nTrials=2000 nPerTrial=5000 and hurdle <=-500"
## [1] "Outcomes across 1e+07 draws of 1-play have mean: -0.0039 and variance: 19.1"
```

```
graphSummary(dfSummary, nPlay=1)
```

```
##      myTrial      myMax      myMin      myLast
## Min.   : 1.0    Min.   : -1.0    Min.   : -554.0    Min.   : -515.00
## 1st Qu.: 500.8  1st Qu.: 22.0    1st Qu.: -239.0    1st Qu.: -198.00
## Median :1000.5  Median : 56.0    Median : -157.0    Median : -94.00
## Mean   :1000.5  Mean   : 150.4    Mean   : -169.5    Mean   : -19.49
## 3rd Qu.:1500.2  3rd Qu.: 118.0    3rd Qu.: -85.0    3rd Qu.: 26.00
## Max.   :2000.0  Max.   :1743.0    Max.   : 1.0      Max.   :1617.00
##
##      myCond      myN_Conc      myVal_Conc
## Mode :logical  Min.   :4302    Min.   : -500
## FALSE:1997    1st Qu.:4597    1st Qu.: -500
## TRUE :3       Median :4892    Median : -500
## NA's :0       Mean   :4710    Mean   : -500
##              3rd Qu.:4914    3rd Qu.: -500
##              Max.   :4937    Max.   : -500
##              NA's   :1997    NA's   :1997
```



Results for 3-play

```
## Run for 3-play
baseOutcomes <- getBaseOutcomes(myFileName="Play003Outcomes.csv", forceEQ=TRUE)
```

```
## [1] "Modifying probablities 0.262602735 -3"
## [1] "New probablities 0.26260273 -3"
## [1] "Probabilities sum to 1. Outcomes has mean -0.0152 and variance 67.7"
```

```
dfSummary <- calcOutcomes(baseOutcomes=baseOutcomes, nPlay=3)
```

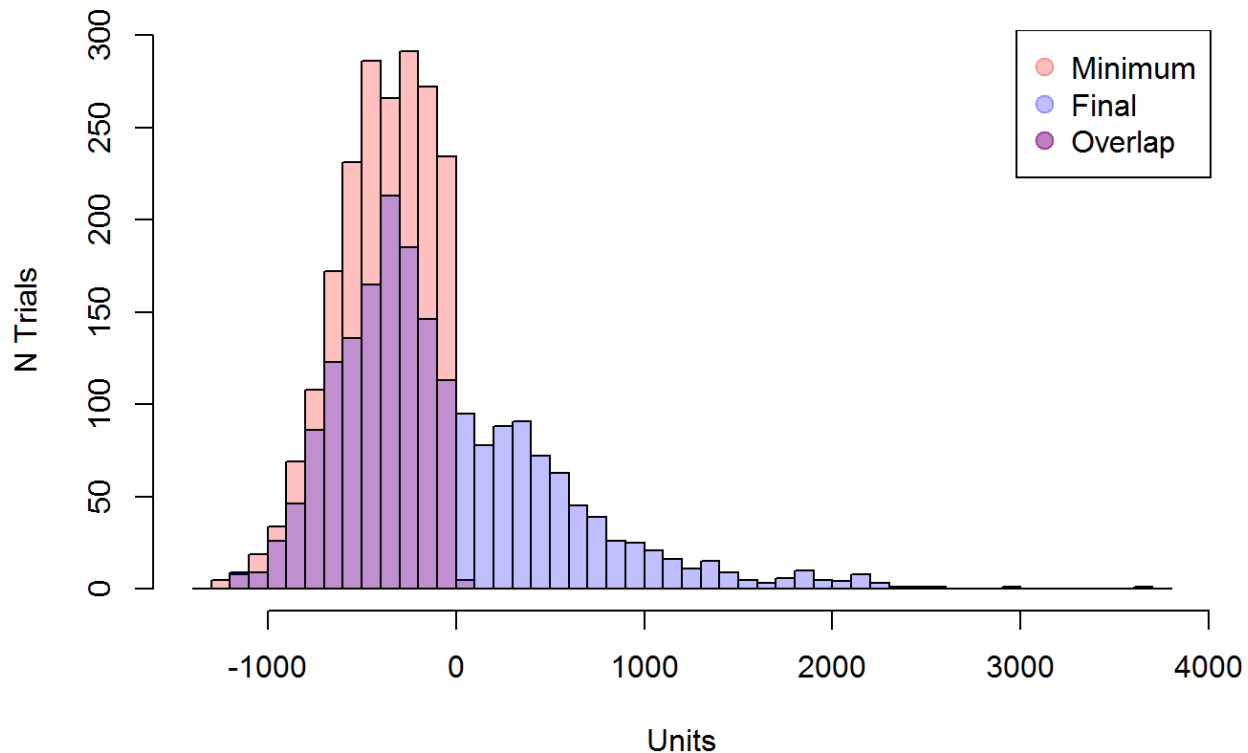
```
## [1] "Running 3-play with nTrials=2000 nPerTrial=5000 and hurdle <=-500"
## [1] "Outcomes across 1e+07 draws of 3-play have mean: -0.00986 and variance: 71"
```

```
graphSummary(dfSummary, nPlay=3)
```



```
##      myTrial      myMax      myMin      myLast
## Min.   : 1.0    Min.   : -3.0    Min.   : -1296.0    Min.   : -1199.00
## 1st Qu.: 500.8  1st Qu.: 43.0    1st Qu.: -561.2    1st Qu.: -463.00
## Median :1000.5  Median : 134.0    Median : -377.5    Median : -199.00
## Mean   :1000.5  Mean   : 349.1    Mean   : -395.4    Mean   : -49.31
## 3rd Qu.:1500.2  3rd Qu.: 561.8    3rd Qu.: -196.0    3rd Qu.: 275.00
## Max.   :2000.0  Max.   :3823.0    Max.   : 9.0      Max.   : 3668.00
##
##      myCond      myN_Cond      myVal_Cond
## Mode :logical    Min.   :1189    Min.   : -502.0
## FALSE:1354      1st Qu.:2639    1st Qu.: -501.0
## TRUE :646       Median :3352    Median : -501.0
## NA's :0         Mean   :3360    Mean   : -500.7
##                3rd Qu.:4158    3rd Qu.: -500.0
##                Max.   :4995    Max.   : -500.0
##                NA's   :1354    NA's   :1354
```

Results: 2000 Trials (5000 3-play draws per trial)



Results for 5-play

```
## Run for 5-play
baseOutcomes <- getBaseOutcomes(myFileName="Play005Outcomes.csv", forceEQ=TRUE)
```

```
## [1] "Modifying probablities 0.130120405 -5"
## [1] "New probablities 0.130120414 -5"
## [1] "Probabilities sum to 1. Outcomes has mean -0.0253 and variance 130"
```

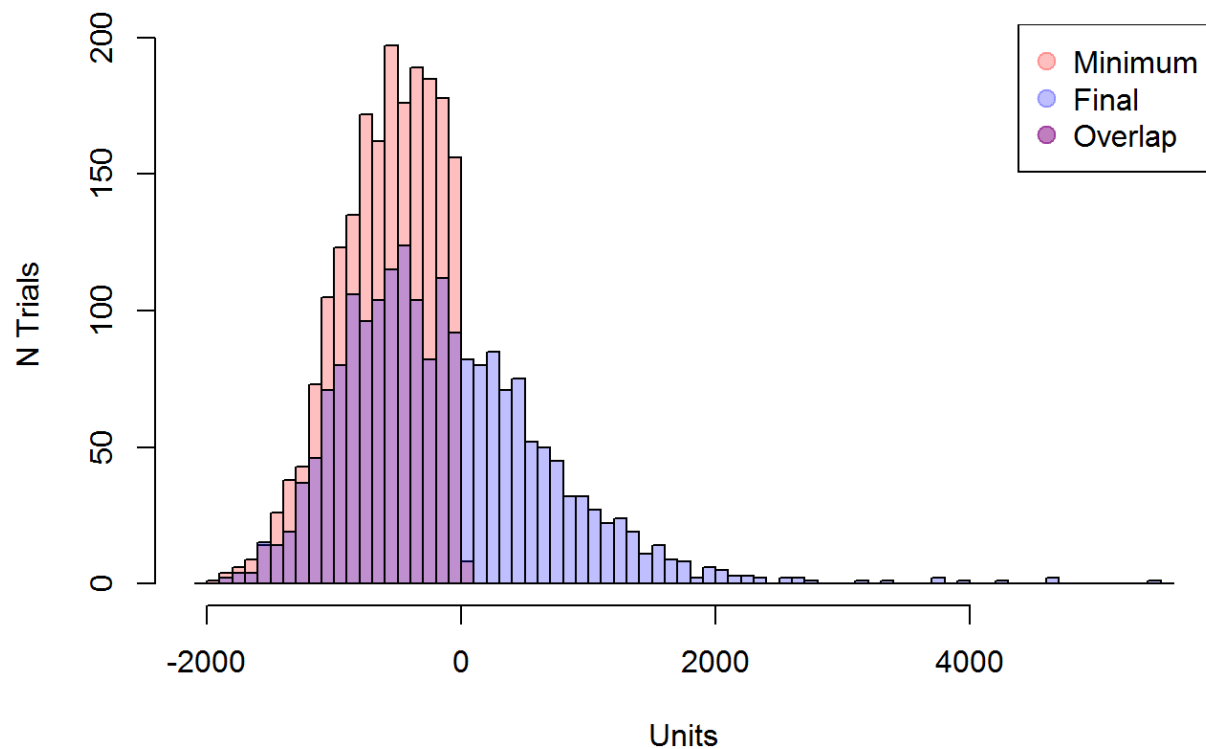
```
dfSummary <- calcOutcomes(baseOutcomes=baseOutcomes, nPlay=5)
```

```
## [1] "Running 5-play with nTrials=2000 nPerTrial=5000 and hurdle <=-500"
## [1] "Outcomes across 1e+07 draws of 5-play have mean: -0.0245 and variance:
130"
```

```
graphSummary(dfSummary, nPlay=5)
```

```
##      myTrial      myMax      myMin      myLast
## Min.   : 1.0    Min.   : -5.0    Min.   : -1922.0  Min.   : -1891.0
## 1st Qu.: 500.8  1st Qu.:  66.0    1st Qu.: -856.5  1st Qu.: -696.0
## Median :1000.5  Median : 237.5    Median : -554.5  Median : -231.5
## Mean   :1000.5  Mean   : 477.4    Mean   : -596.4  Mean   : -122.6
## 3rd Qu.:1500.2  3rd Qu.: 733.0    3rd Qu.: -284.0  3rd Qu.:  330.2
## Max.   :2000.0  Max.   :5668.0    Max.    :  21.0   Max.    : 5461.0
##
##      myCond      myN_Cond      myVal_Cond
## Mode :logical  Min.    : 533    Min.    : -504.0
## FALSE:892     1st Qu.:1589    1st Qu.: -502.0
## TRUE :1108     Median :2317    Median : -501.0
## NA's :0        Mean     :2503    Mean     : -501.3
##              3rd Qu.:3311    3rd Qu.: -500.0
##              Max.     :4988    Max.     : -500.0
##              NA's     :892     NA's     :892
```

Results: 2000 Trials (5000 5-play draws per trial)



Results for 10-play

```
## Run for 10-play
baseOutcomes <- getBaseOutcomes(myFileName="Play010Outcomes.csv", forceEQ=TRUE)
```

```
## [1] "Modifying probablities 0.025913774 -10"
## [1] "New probablities 0.025913775 -10"
## [1] "Probabilities sum to 1. Outcomes has mean -0.0506 and variance 345"
```

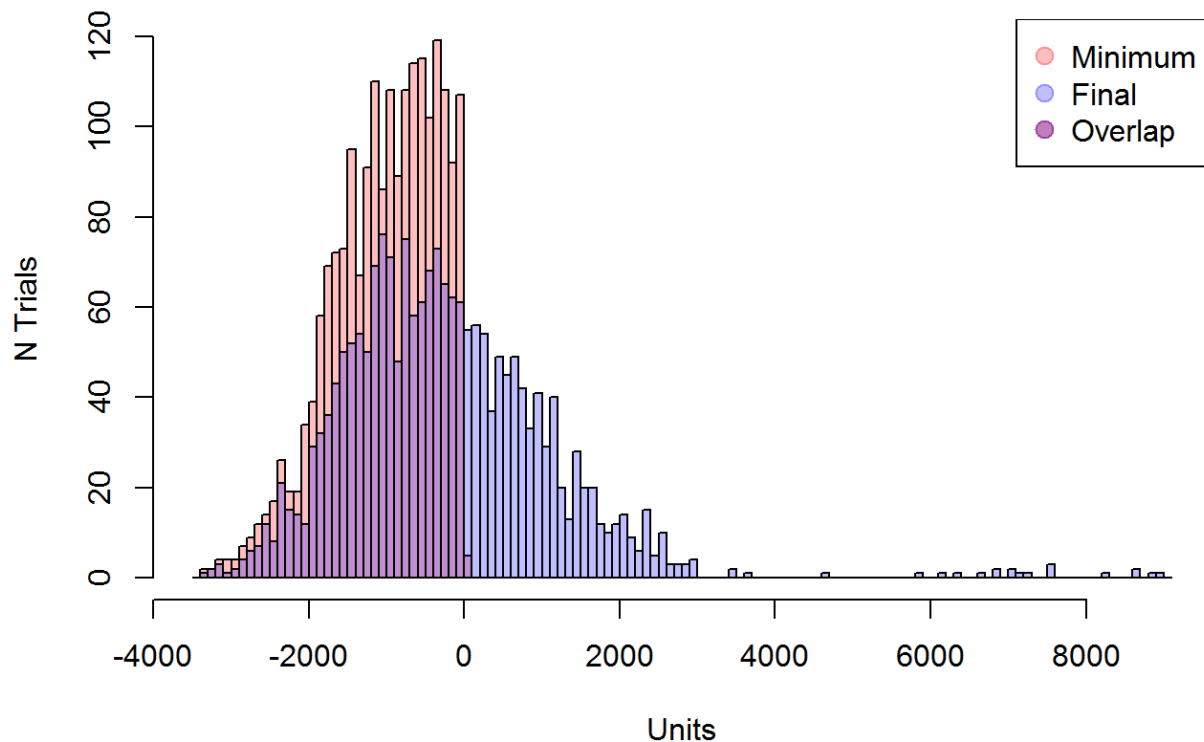
```
dfSummary <- calcOutcomes(baseOutcomes=baseOutcomes, nPlay=10)
```

```
## [1] "Running 10-play with nTrials=2000 nPerTrial=5000 and hurdle <=-500"
## [1] "Outcomes across 1e+07 draws of 10-play have mean: -0.0459 and variance: 397"
```

```
graphSummary(dfSummary, nPlay=10)
```

```
##      myTrial      myMax      myMin      myLast
## Min.   : 1.0    Min.   : -10.0   Min.   : -3372.0   Min.   : -3334.0
## 1st Qu.: 500.8  1st Qu.: 139.0   1st Qu.: -1479.0   1st Qu.: -1130.5
## Median :1000.5  Median : 498.5   Median : -935.5   Median : -362.0
## Mean   :1000.5  Mean   : 796.4   Mean   : -1020.4   Mean   : -229.6
## 3rd Qu.:1500.2  3rd Qu.:1138.8   3rd Qu.: -475.0   3rd Qu.: 521.2
## Max.   :2000.0  Max.   :9760.0   Max.   : 35.0     Max.   : 8975.0
##
##      myCond      myN_Cond      myVal_Cond
## Mode :logical   Min.   : 204    Min.   : -509.0
## FALSE:533      1st Qu.: 731    1st Qu.: -504.0
## TRUE :1467      Median :1205   Median : -502.0
## NA's :0         Mean   :1579   Mean   : -502.5
##                3rd Qu.:2114   3rd Qu.: -501.0
##                Max.   :4993   Max.   : -500.0
##                NA's   :533    NA's   :533
```

Results: 2000 Trials (5000 10-play draws per trial)



Results for 50-play

```
## Run for 50-play
baseOutcomes <- getBaseOutcomes(myFileName="Play050Outcomes.csv", forceEQ=TRUE)
```

```
## [1] "Modifying probablities 2.1776e-05 -50"
## [1] "New probablities 2.1764999999978e-05 -50"
## [1] "Probabilities sum to 1. Outcomes has mean -0.253 and variance 5139"
```

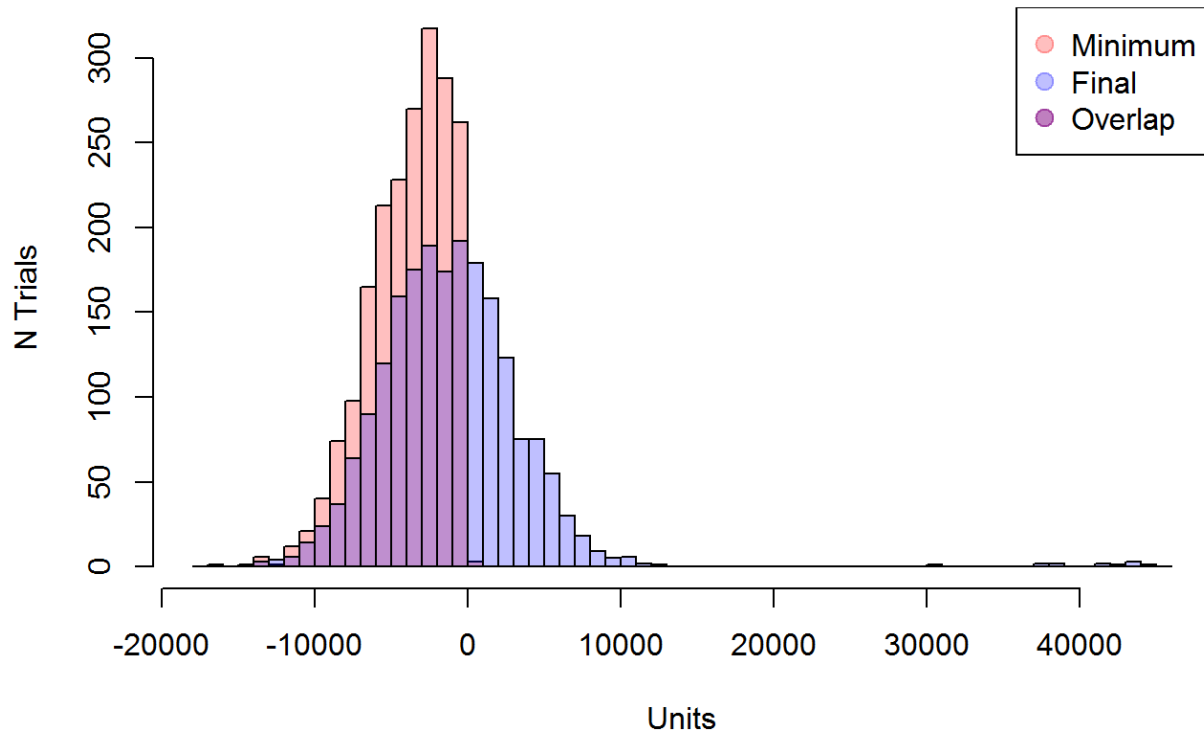
```
dfSummary <- calcOutcomes(baseOutcomes=baseOutcomes, nPlay=50)
```

```
## [1] "Running 50-play with nTrials=2000 nPerTrial=5000 and hurdle <=-500"
## [1] "Outcomes across 1e+07 draws of 50-play have mean: -0.218 and variance:
5249"
```

```
graphSummary(dfSummary, nPlay=50)
```

```
##      myTrial      myMax      myMin      myLast
## Min.   : 1.0    Min.   : -41.0  Min.   :-16435  Min.   :-14915
## 1st Qu.: 500.8  1st Qu.:  758.5  1st Qu.: -5570  1st Qu.: -4164
## Median :1000.5  Median : 1946.5  Median : -3428  Median : -1312
## Mean   :1000.5  Mean   : 2744.9  Mean   : -3889  Mean   : -1092
## 3rd Qu.:1500.2  3rd Qu.: 3733.2  3rd Qu.: -1835  3rd Qu.:  1470
## Max.   :2000.0  Max.   :48369.0  Max.    :    86  Max.   : 44346
##
##      myCond      myN_Cond      myVal_Cond
## Mode :logical  Min.    :  24  Min.    :-544.0
## FALSE:123     1st Qu.:  81  1st Qu.: -518.0
## TRUE :1877    Median : 170  Median : -510.0
## NA's :0       Mean   : 457  Mean   : -511.5
##              3rd Qu.: 442  3rd Qu.: -504.0
##              Max.   :4818  Max.   : -500.0
##              NA's   :123  NA's   :123
```

Results: 2000 Trials (5000 50-play draws per trial)



Results for 100-play

```
## Run for 100-play
baseOutcomes <- getBaseOutcomes(myFileName="Play100Outcomes.csv", forceEQ=TRUE)
```

```
## [1] "Modifying probablities 6.3e-08 -99"
## [1] "New probablities 6.48000000379104e-08 -99"
## [1] "Probabilities sum to 1. Outcomes has mean -0.506 and variance 18791"
```

```
dfSummary <- calcOutcomes(baseOutcomes=baseOutcomes, nPlay=100)
```

```
## [1] "Running 100-play with nTrials=2000 nPerTrial=5000 and hurdle <=-500"
## [1] "Outcomes across 1e+07 draws of 100-play have mean: -0.623 and variance: 14590"
```

```
graphSummary(dfSummary, nPlay=100)
```

```
##      myTrial      myMax      myMin      myLast
## Min.   : 1.0    Min.   : -73    Min.   : -28505   Min.   : -28461
## 1st Qu.: 500.8  1st Qu.: 1352   1st Qu.: -10916  1st Qu.: -8290
## Median :1000.5  Median : 3420   Median : -6768   Median : -3188
## Mean   :1000.5  Mean   : 4667   Mean   : -7698   Mean   : -3114
## 3rd Qu.:1500.2  3rd Qu.: 6585   3rd Qu.: -3608   3rd Qu.: 1512
## Max.   :2000.0  Max.   : 93758   Max.   : 704     Max.   : 93462
##
##      myCond      myN_Cond      myVal_Cond
## Mode :logical   Min.   : 9    Min.   : -573.0
## FALSE:62       1st Qu.: 30   1st Qu.: -535.0
## TRUE :1938      Median : 60   Median : -521.0
## NA's :0         Mean   : 274   Mean   : -523.1
##                3rd Qu.: 184   3rd Qu.: -509.0
##                Max.   : 4932   Max.   : -500.0
##                NA's   : 62    NA's   : 62
```

Results: 2000 Trials (5000 100-play draws per trial)

