Standing Ovation Replication and Modification

Jon Green

July 10, 2017

This script models a standing ovation, based off of Miller and Page (2004).

The model first sets an audience of size x by x, which initializes when the performance ends. Each agent in the audience perceives the performance to have a certain quality with mean y and variance yvar. Each agent has a quality threshold for standing with mean z and variance zvar. Each agent has a social threshold for standing with mean s and variance svar. Each agent has a vision rule for which other audience members they will look at when considering whether to stand.

When the performance ends, audience members stand if quality > quality threshold. In each round, audience members who are not already standing look around based on a vision rule (the default randomly selects between the five neighbors to their immediate left/right/front, a "cone" of audience members they can see in front of them, or the entire audience, but specific rules can be applied). If the percentage of agents they see standing, combined with their initial quality estimate, exceeds their social threshold, they stand.

The first function sets audience size and each agent's perceived quality, quality threshold, social threshold, and vision rule. In an important deviation from Miller and Page's model, I use truncated normal distributions to set agents' quality perception, quality threshold, and social threshold. Additionally, rather than specifying a specific vision rule for the whole population in each run as Miller and Page's model does, the default setting for this function gives each agent a vision rule sampled from three available vision rules (one or two of the rules can be manually specified).

```
do.aud <- function(dim = 20, qual = .7, qualvar = .2,
                   qualt = .75, qualtvar = .1,
                   soct = .5, soctvar = .25,
                   viz.rule = c("five","cone","global")){
  require(msm)
  require(data.table)

  # Matrices for quality, quality threshold and social threshold
  quality <- matrix(data=rtnorm(dim^2,
                                mean = qual, sd = qualvar,
                                lower = 0, upper = 1),
                    nrow=dim, ncol=dim)
  quality.threshold <- matrix(data=rtnorm(dim^2,
                                          mean = qualt, sd = qualtvar,
                                          lower = .7, upper = 1),
                              nrow=dim, ncol=dim)
  social.threshold <- matrix(data=rtnorm(dim^2,
                                         mean = soct, sd = soctvar,
                                         lower = .05, upper = .8),
                             nrow=dim, ncol=dim)

  # Data table giving each audience member an id and x/y coordinates,
  # To be filled with characteristics
  aud <<- data.table(id = 1:prod(dim*dim),
                     x = rep(1:dim, dim),
                     y = rep(1:dim, each = dim),
                     q = rep(NA, prod(dim*dim)),
                     qt = rep(NA, prod(dim*dim)),
                     st = rep(NA, prod(dim*dim)),
                     vr = sample(viz.rule, prod(dim*dim), replace = TRUE),
                     stand = rep(NA, prod(dim*dim)))

  # Fill data table with characteristics
  for(i in 1:dim){
    for(j in 1:dim){
      aud$q[aud$x == i & aud$y == j] <<- quality[i,j]
      aud$qt[aud$x == i & aud$y == j] <<- quality.threshold[i,j]
      aud$st[aud$x == i & aud$y == j] <<- social.threshold[i,j]
      #Audience member is standing at end of performance if quality > quality threshold
      aud$stand[aud$x == i & aud$y == j] <<- quality[i,j] > quality.threshold[i,j]
    }
  }
}
```

The second function plots the theater.

```
plotStand <- function(title){
  require(data.table)
  require(ggplot2)
  require(RColorBrewer)

  # find the dimensions of the grid to get the best dot size
  dims <- c(max(aud$x), max(aud$y))

  # plot each agent's status: standing or nah
  p <- ggplot(data = aud,
              aes(x = x, y = y, color = stand)) +
    # resize dots to grid
    geom_point(size = 100/sqrt(prod(dims))) +
    # theme: mostly blank
    theme_bw() +
    theme(axis.line = element_blank(),
          axis.text.x = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks = element_blank(),
          axis.title.x = element_blank(),
          axis.title.y = element_blank(),
          panel.background = element_blank(),
          panel.border = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          plot.background = element_blank(),
          plot.title = element_text(lineheight=3,
                                    face="bold",
                                    color="black",
                                    size=14)) +
    # fix axes to avoid distortion in the picture
    coord_fixed(ratio = 1) +
    # add the title
    ggtitle(title)

  return(p)
}
```

The next function determines whether a given agent will stand. In an important deviation from Miller and Page's model, an agent will stand based on a combination of the number of agents they see standing and their initial quality perception. In this model, the two are given equal weight; in subsequent models, this could vary by iteration such that initial quality perception matters less and percent of agents standing matters more the longer the applause lasts.

```
# Takes agent's row number, column number and audience dimension
do.stand <- function(row, col, dim = 20){
  if(aud[x == col & y == row, stand] == TRUE){
    return(TRUE) # If agent is standing, they stay standing
  }
  else{
    # Return percent of standing agents in field of vision, depending on selected rule
    if(aud[x == col & y == row, vr] == "five"){
      x_vals <- c(col+1, col, col-1)
      x_vals <- unique(sapply(x_vals, function(x) {
        if(x < 1){x <- 1}
        if(x > dim){x <- dim}
        else{x}
      }))

      y_vals <- c(row, row+1)
      y_vals <- unique(sapply(y_vals, function(x) {
        if(x > dim){x <- dim}
        else{x}
      }))

      neighbor.ids <- aud[x %in% x_vals & y %in% y_vals, id]
      neighbor.ids <- neighbor.ids[! neighbor.ids %in% aud[x == col & y == row, id]]
      pct.stand <- sum(aud[id %in% neighbor.ids, stand])/length(unique(neighbor.ids))
      return((pct.stand/2 + aud[x == col & y == row, q]/2) > aud[x == col & y == row, st])
    }
    if(aud[x == col & y == row, vr] == "cone"){
      coords <- rbind(c(col, row+1), c(col, row-1))
      dis <- 1
      for(i in (row+1):dim){
        y <- i
        x_vals <- c((col-dis):(col+dis))
        for(j in 1:length(x_vals)){
          coords <- rbind(coords, c(x_vals[j], y))
        }
        dis <- dis+1
      }
      coords[,1] <- sapply(coords[,1], function(x) {
        ifelse(x < 1, 1, ifelse(x > dim, dim, x))
      })
      coords <- unique(coords)

      see.ids <- aud[x %in% coords[,1] & y %in% coords[,2], id]
      pct.stand <- sum(aud[id %in% see.ids, stand])/length(unique(see.ids))
      return((pct.stand/2 + aud[x == col & y == row, q]/2) > aud[x == col & y == row, st])
    }
    if(aud[x == col & y == row, vr] == "global"){
      return((mean(aud$stand)/2 +  aud[x == col & y == row, q]/2) > aud[x == col & y == row, st])
    }
  }
}
```

The last function iterates through a specified number of rounds and creates vectors of audience characteristics by round, including the percent of the audience that is standing, the mean social threshold of the audience members who are sitting, and the mean quality perception of the audience members who are sitting.

```
iterate <- function(iterations){

  #Calculate initial percent standing
  pctstanding <<- mean(aud$stand)

  #Calculate mean social threshold of sitting audience members
  mst <<- mean(aud$st[aud$stand == FALSE])

  #Calculate mean quality perception of sitting audience members
  mq <<- mean(aud$q[aud$stand == FALSE])

  for(i in 1:iterations){
    # Update standing status for each audience member in each iteration
    aud[,stand := do.stand(row = y, col = x, dim = 20),
        by = 1:nrow(aud)]

    # Update vector of percent standing in each iteration
    pctstanding <<- c(pctstanding, mean(aud$stand))

    # Update vector of mean social threshold of sitting members in each iteration
    mst <<- c(mst, mean(aud$st[aud$stand == FALSE]))

    # Update vector of mean quality perception of sitting members in each iteration
    mq <<- c(mq, mean(aud$q[aud$stand == FALSE]))
  }

  #Plot the updated theater after i iterations
  plotStand(paste("Audience after round ", i, sep = ""))
}
```

Now we can try a couple of different runs, starting with a run based on the default settings. In this run we observe that roughly one third of the audience is standing at the end of the performance and an additional two thirds of the audience stands over the course of the first four runs, after which the remaining audience members who are sitting have lower quality perceptions and higher social thresholds relative to those who stood.

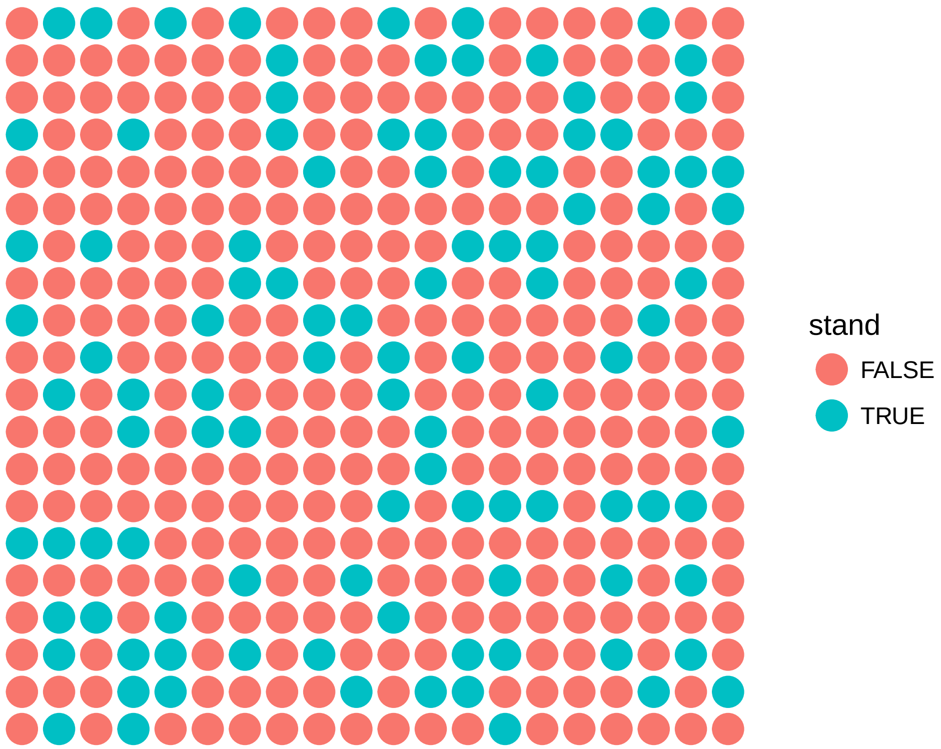
```
do.aud()
```

```
## Loading required package: msm

## Loading required package: data.table
```

```
plotStand(title = "Audience Immediately After Performance Ends")
```

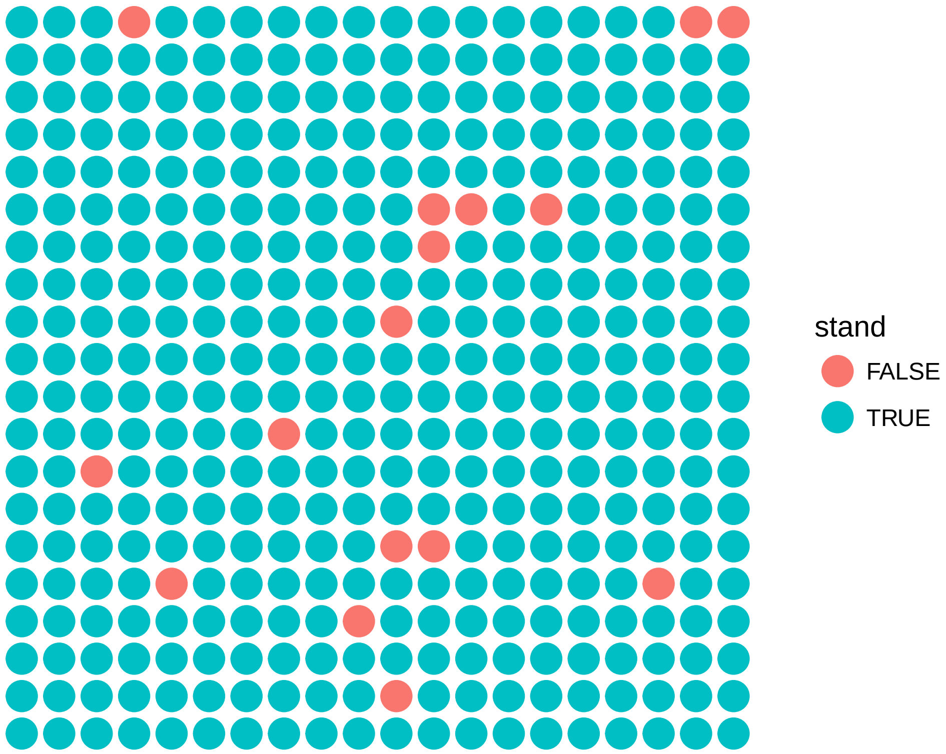
```
## Loading required package: ggplot2

## Loading required package: RColorBrewer
```

Audience Immediately After Performance Ends

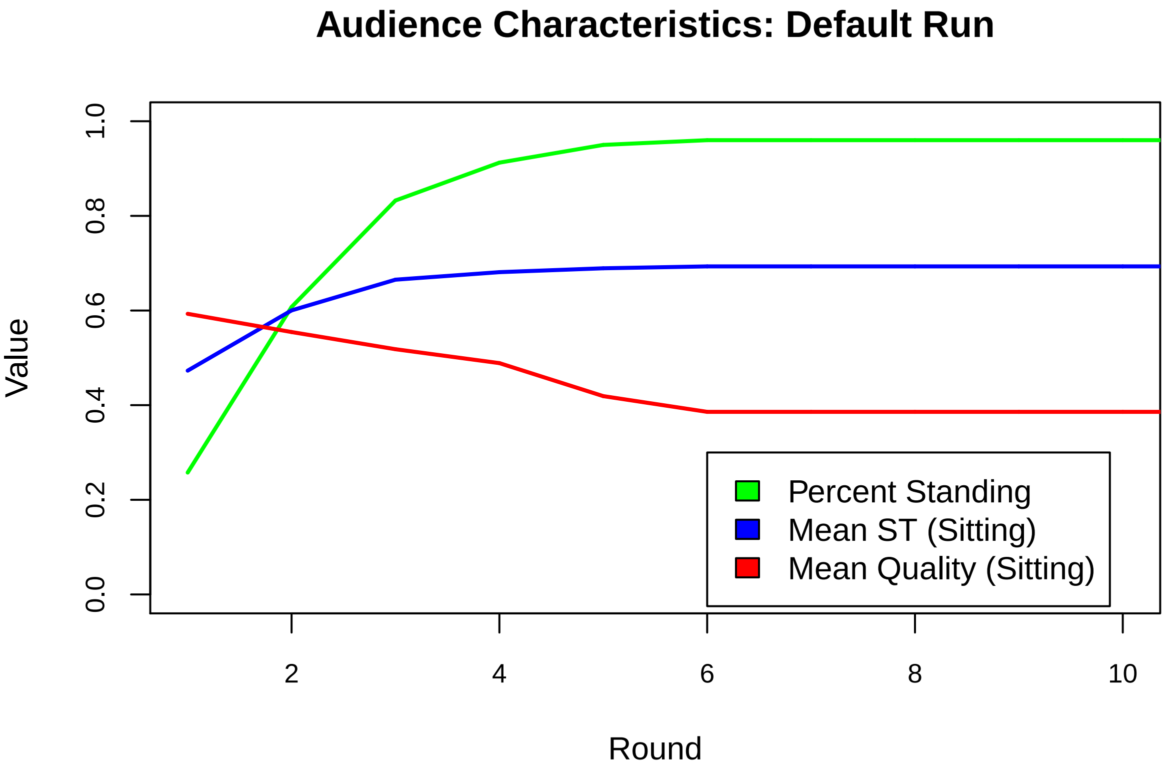


```
iterate(iterations = 10)
```

Audience after round 10



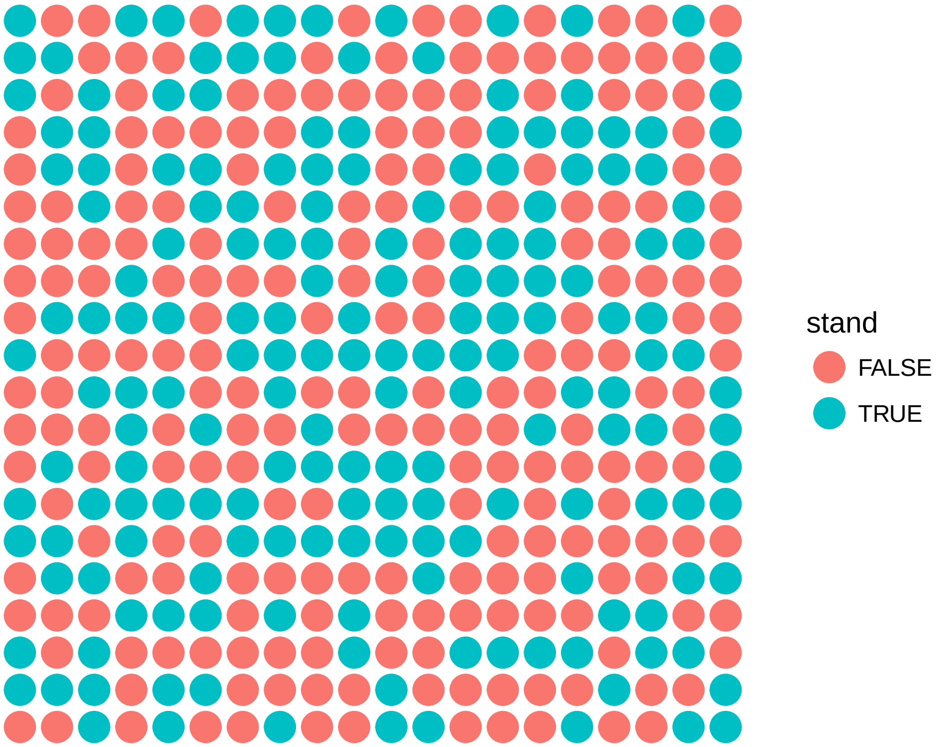
```
# Plot audience characteristics by round
for(i in 1:length(pctstanding)){
  if(i == 1){
    plot(-100, -100, xlim=c(1,10), ylim=c(0,1),
         ylab="Value", xlab="Round", type="n", cex.axis=0.8,
         main = "Audience Characteristics: Default Run")
  }else{
    segments(i-1, pctstanding[i-1], i, pctstanding[i], col = "green", lwd=2)
    segments(i-1, mst[i-1], i, mst[i], col = "blue", lwd=2)
    segments(i-1, mq[i-1], i, mq[i], col = "red", lwd=2)
  }
}
legend(x = 6, y = .3,
       legend = c("Percent Standing","Mean ST (Sitting)", "Mean Quality (Sitting)"),
       fill = c("green","blue","red"))
```



This run couples a high quality performance with universal cone rule for audience vision, along with a few other minor tweaks to the default settings. In this run we again find that nearly all audience members are standing after four rounds, though more were initially standing at the end of the performance. We also observe a similar decline in mean quality perception and increase in mean social threshold among those who remain seated over time.

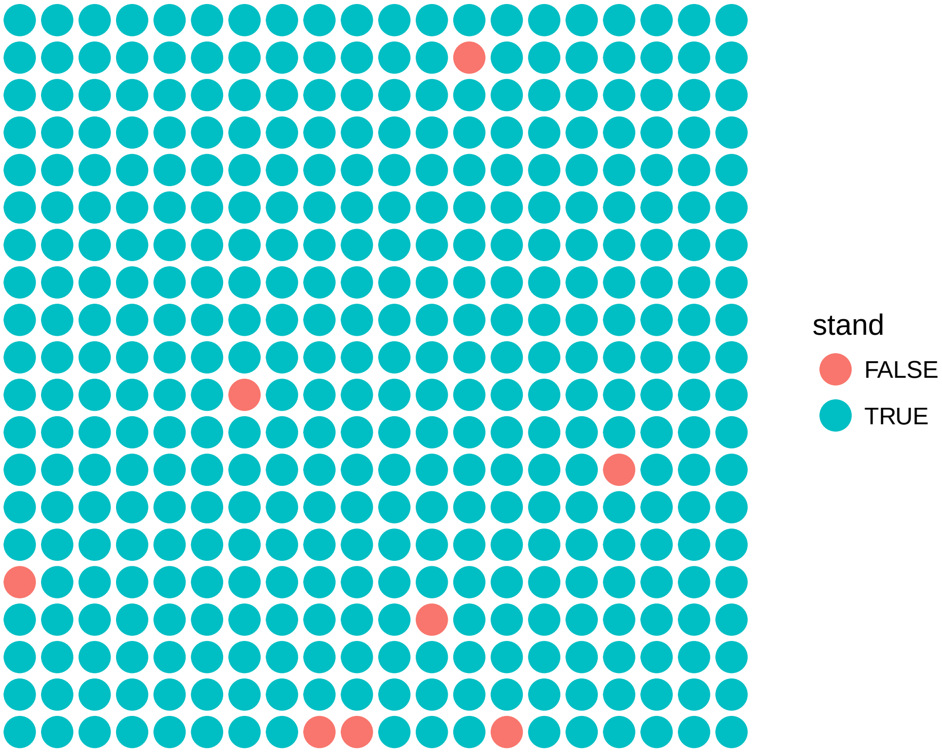
```
do.aud(qual = .8, qualvar = .2, qualt = .7, soctvar = .2, viz.rule = "cone")
plotStand(title = "Audience Immediately After Performance Ends")
```

Audience Immediately After Performance Ends

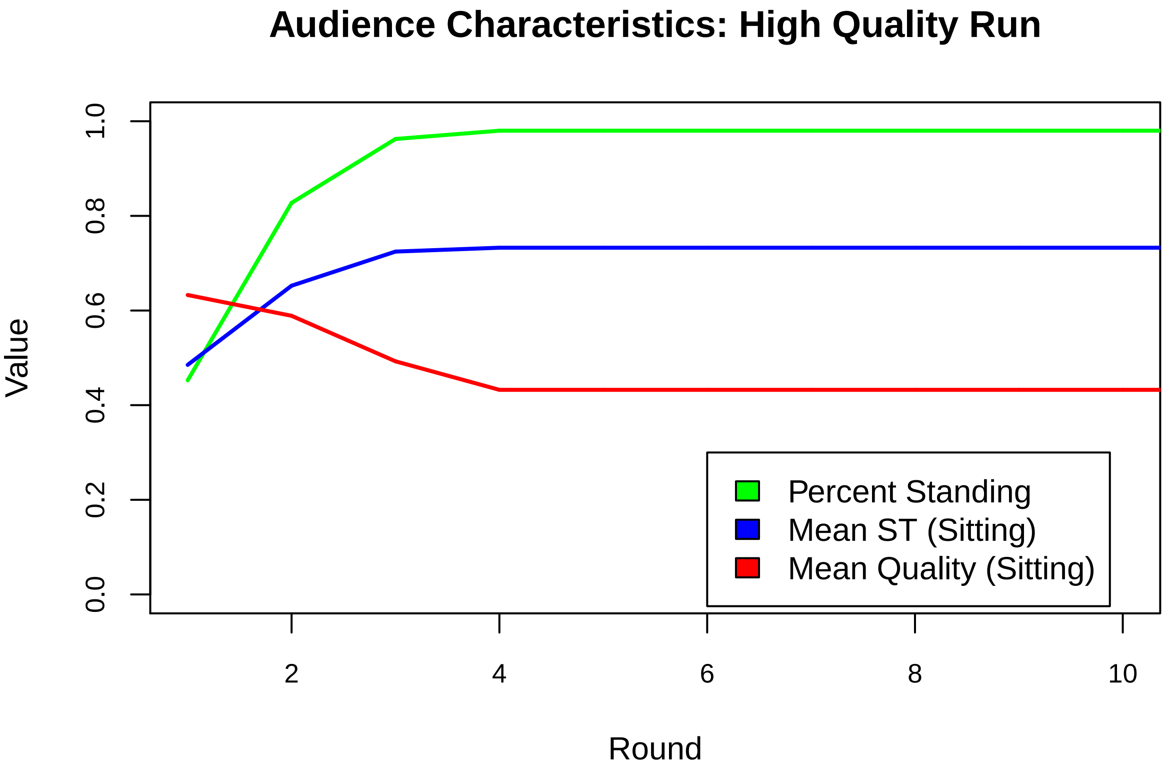


```
iterate(iterations = 10)
```

Audience after round 10

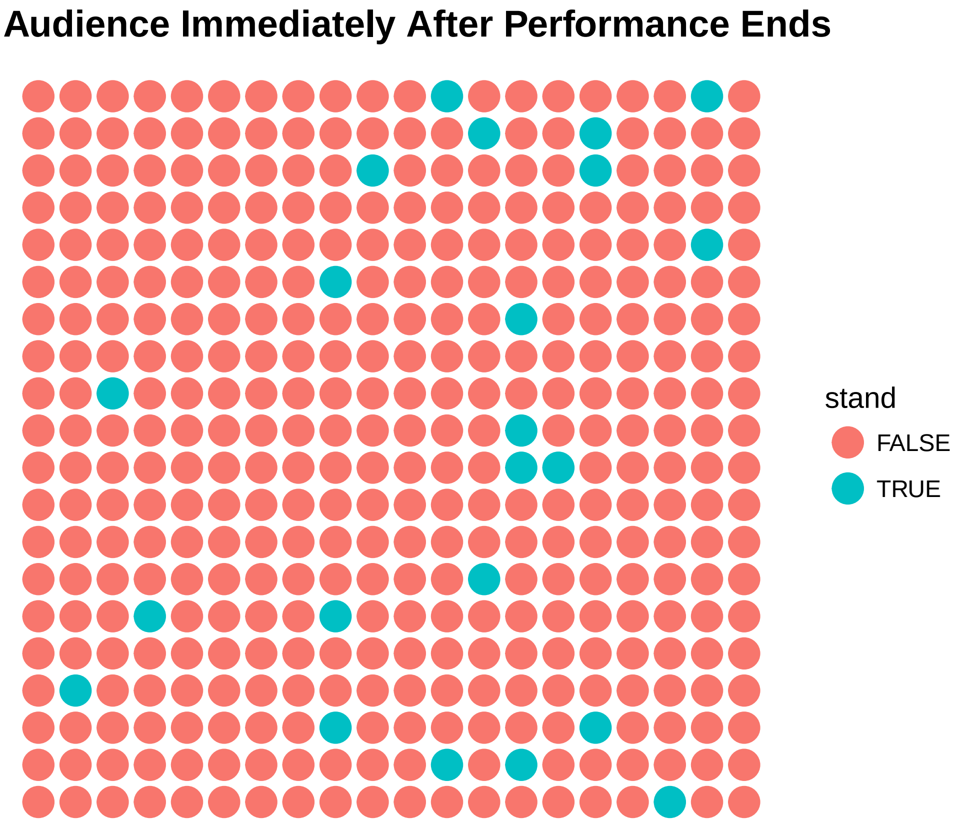


```
# Plot audience characteristics by round
for(i in 1:length(pctstanding)){
  if(i == 1){
    plot(-100, -100, xlim=c(1,10), ylim=c(0,1),
         ylab="Value", xlab="Round", type="n", cex.axis=0.8,
         main = "Audience Characteristics: High Quality Run")
  }else{
    segments(i-1, pctstanding[i-1], i, pctstanding[i], col = "green", lwd=2)
    segments(i-1, mst[i-1], i, mst[i], col = "blue", lwd=2)
    segments(i-1, mq[i-1], i, mq[i], col = "red", lwd=2)
  }
}
legend(x = 6, y = .3,
       legend = c("Percent Standing","Mean ST (Sitting)", "Mean Quality (Sitting)"),
       fill = c("green","blue","red"))
```

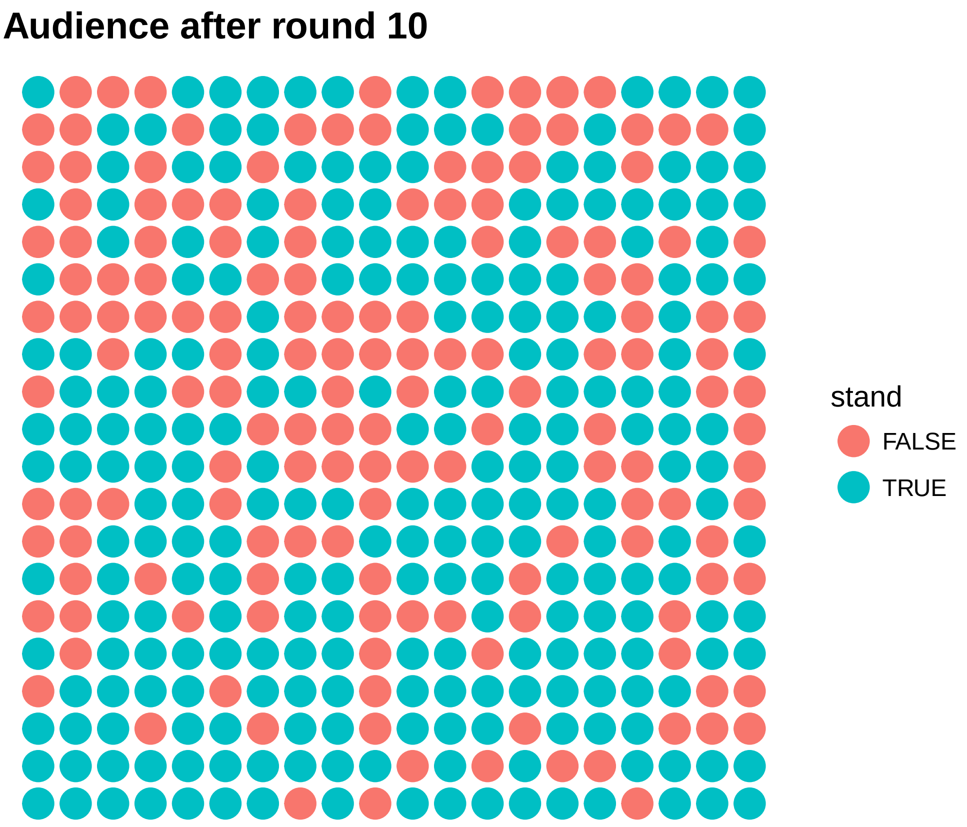


This run specifies a low quality performance, with no further tweaks to the default settings. In this run, very few audience members are standing at first, and after four rounds an equilibrium is reached with nearly half of the audience standing. Mean social threshold and mean quality perception for sitting audience members both start at .5, as specified in the model, and diverge (with social threshold increasing and quality perception decreasing) as members with lower social thresholds and higher quality perceptions stand.

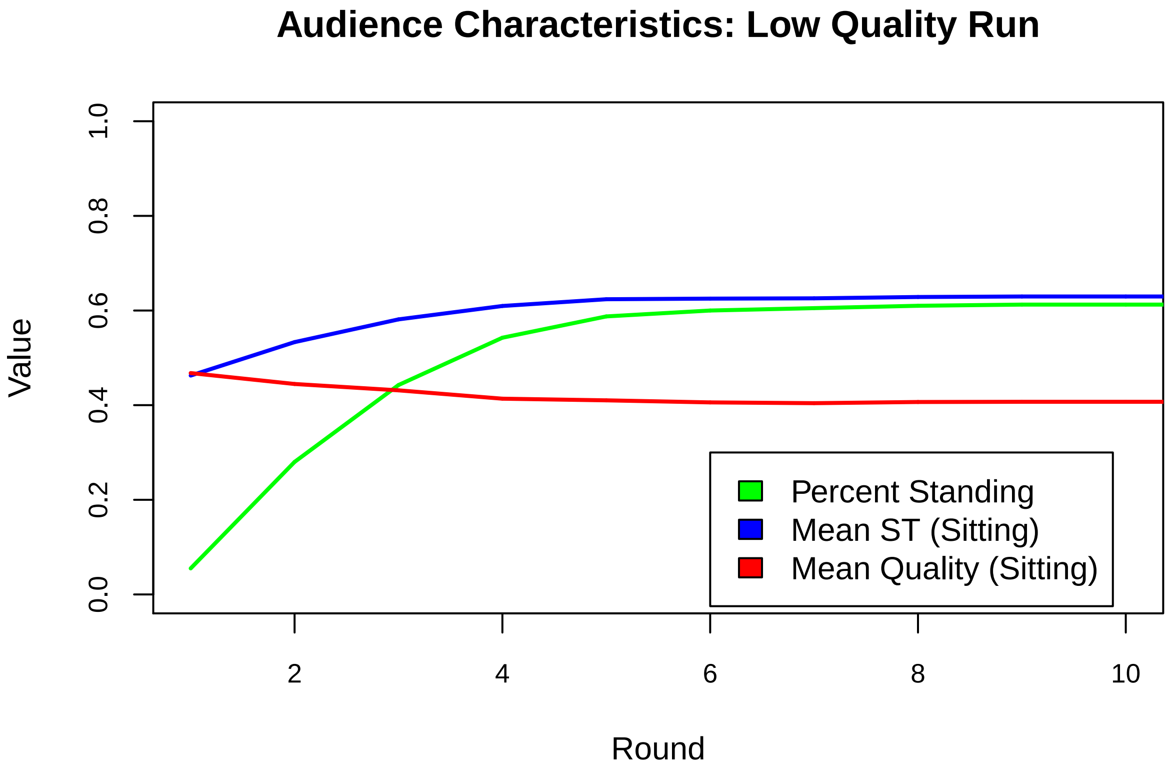
```
do.aud(qual = .5)
plotStand(title = "Audience Immediately After Performance Ends")
```



```
iterate(iterations = 10)
```



```
# Plot audience characteristics by round
for(i in 1:length(pctstanding)){
  if(i == 1){
    plot(-100, -100, xlim=c(1,10), ylim=c(0,1),
         ylab="Value", xlab="Round", type="n", cex.axis=0.8,
         main = "Audience Characteristics: Low Quality Run")
  }else{
    segments(i-1, pctstanding[i-1], i, pctstanding[i], col = "green", lwd=2)
    segments(i-1, mst[i-1], i, mst[i], col = "blue", lwd=2)
    segments(i-1, mq[i-1], i, mq[i], col = "red", lwd=2)
  }
}
legend(x = 6, y = .3,
       legend = c("Percent Standing","Mean ST (Sitting)", "Mean Quality (Sitting)"),
       fill = c("green","blue","red"))
```


Audience Characteristics: Low Quality Run
Value
Round
🟩 Percent Standing
🟦 Mean ST (Sitting)
🟥 Mean Quality (Sitting)