Probability In-Class Test

```r
library(ggplot2) # plotting library
library(dplyr)   # data wrangling library
```

Load libraries

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Generate a 52-card deck
suits = c('H', 'D', 'S', 'C')
cards = c(2:10, 'J', 'Q', 'K', 'A')
s = paste0(rep(cards, length(suits)), rep(suits, each = length(cards)))
```

```r
# Simulate dealing 5 cards to 8 hands (40 cards)
nsimulations = 1e5
simulatedData = replicate(nsimulations, sample(s, size = 40, prob = rep(1/length(s), length(s))))
```

1

```r
# Check event that exactly 3 aces are dealt
checkEvent1 = function(data){
  return(sum(grepl('A', data)) == 3)
}
```

2

```r
# Check event that 4 hands have exactly one ace each
checkEvent2 = function(data) {
  condition1 = (sum(grepl('A', data)) == 4)
  condition2 = (length(unique(ceiling(which(grepl('A', data)) /5))) == 4)
  return(condition1 & condition2)
}
```

3

```r
# Check event that one hand has all four aces
checkEvent3 = function(data){
  condition1 = any(rowSums(matrix(data == 'A', ncol = 5)) == 4)
  condition2 = which(rowSums(matrix(data == 'A', ncol = 5)) == 4)
  return(condition1 & condition2)
}
```

4

```
# Check event given that the ace of spades was among the cards dealt, that it is in a hand with at leas
checkEvent4 = function(data){
  hand = ceiling(which(data == 'AS') / 5)
  return('AH' %in% data[c((1+(hand-1)*5):(hand*5))] ||
         'AD' %in% data[c((1+(hand-1)*5):(hand*5))] ||
         'AC' %in% data[c((1+(hand-1)*5):(hand*5))] ||
         'AS' %in% data[c((1+(hand-1)*5):(hand*5))]
         )
  }
```

5

```
# Check event given that exactly two aces and two queens were dealt, that both aces are together in one
checkEvent5 = function(data){
  condition1 = (length(unique(ceiling(which(grepl('A', data)) /5))) == 1)
  condition2 = (length(unique(ceiling(which(grepl('Q', data)) /5))) == 1)
  condition3 = !(condition1 == condition2)
  return(condition1 & condition2 & condition3)
  }
```

6

```
# Check event given that exactly two aces and two queens were dealt with the two aces together in a han
checkEvent6 = function(data){
  return(unique(ceiling(which(grepl('A', data)) /5)) == unique(ceiling(which(grepl('Q', data)) /5)))
  }
```

p1

```
# Calculate probability that exactly 3 aces are dealt
mean(apply(simulatedData, 2, checkEvent1))
```

```
## [1] 0.43824
```

p2

```
# Calculate probability that 4 hands have exactly one ace each
mean(apply(simulatedData, 2, checkEvent2))
```

```
## [1] 0.16108
```

p3

```
# Calculate probability that one hand has all four aces
mean(apply(simulatedData, 2, checkEvent3))
```

```
## [1] NaN
```

p4

```
# Calculate probability given that the ace of spades was among the cards dealt, that it is in a hand wi
simulatedData_reduced = simulatedData[, apply(simulatedData, 2, function(data){return('AS' %in% data)})
mean(apply(simulatedData_reduced, 2, checkEvent4))
```

```
## [1] 1
```

p5

```
# Calculate probability given that exactly two aces and two queens were dealt, find the probability tha
simulatedData_reduced = simulatedData[, apply(simulatedData, 2, function(data){return('AS' %in% data)})
mean((apply(simulatedData_reduced, 2, checkEvent5)))
```

```
## [1] 0
```

p6

```
# Calculate probability given that exactly two aces and two queens were dealt, find the probability tha
simulatedData_reduced = simulatedData[, apply(simulatedData, 2, function(data) {
  return((sum(grepl('A', data)) == 2) &
         (length(unique(ceiling(which(grepl('A', data)) / 5))) == 1) &
         (length(unique(ceiling(which(grepl('Q', data)) / 5))) == 1))
  ))
})]

mean(apply(simulatedData_reduced, 2, checkEvent5))
```

```
## [1] 0
```

p7

```
# Calculate probability given that exactly two aces and two queens were dealt with the two aces togethe
simulatedData_reduced = simulatedData[, apply(simulatedData, 2, function(data) {
  return((sum(grepl('A', data)) == 2)
          & sum(grepl('Q', data)) == 2
          & length(unique(ceiling(which(grepl('A', data)) / 5))) == 1)
          & length(unique(ceiling(which(grepl('Q', data)) / 5))) == 1)
  )
})]

mean(apply(simulatedData_reduced, 2, checkEvent6))
```

```
## Warning in mean.default(apply(simulatedData_reduced, 2, checkEvent6)): argument
## is not numeric or logical: returning NA
```

```
## [1] NA
```

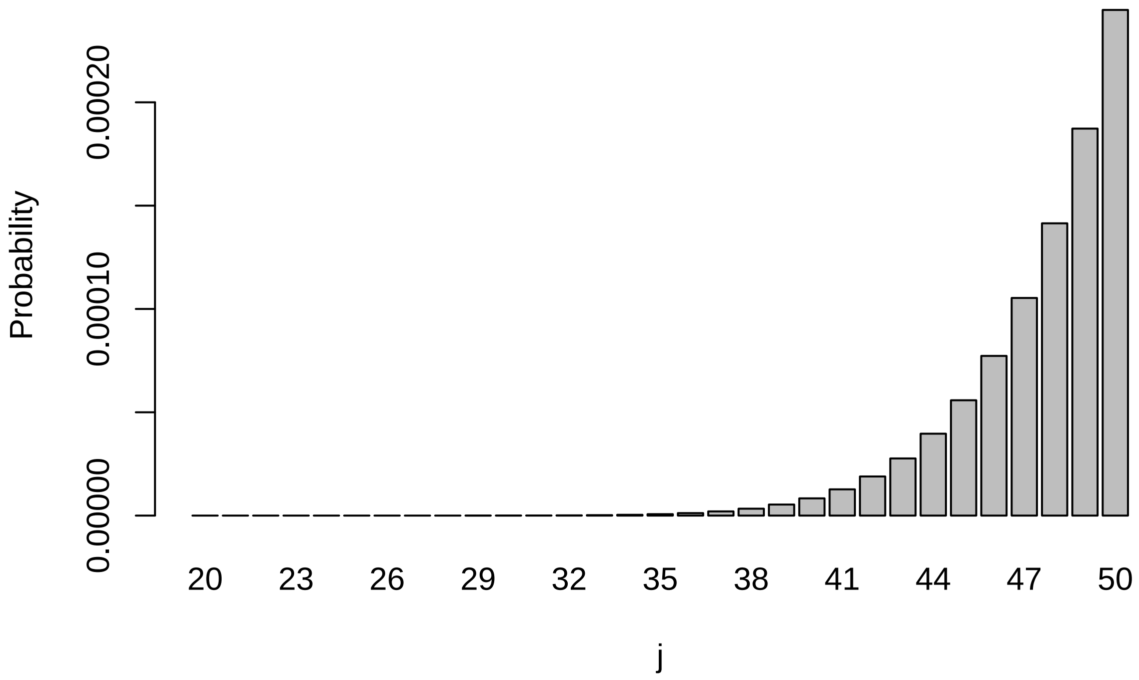
```
r=20
p=0.2
j=200
dnbinom(j-r,r,p)
```

```
## [1] 6.08962e-06
```

```
j_values <- 20:50

data <- dnbinom(j_values - r, r, p)

barplot(data, names = j_values, xlab = "j", ylab = "Probability")
```


Probability
0.00000
0.00010
0.00020
20
23
26
29
32
35
38
41
44
47
50
j