IST 772 Week 2 Breakout 2

Your name here

How many heads?

Instructions:

Pascal's Triangle was named after 17th century French mathematician Blaise Pascal, but the structure appears to have emerged independently from India, China, and Persia as early as the second century BCE. Each element is the number of different ways of choosing k elements from a set with n elements. Think of n as the row number (starting with 0) and the number of coins we are tossing. Then k is the position within a row (starting with zero) and is also the number of heads. The third row below would read like this: Tossing two coins, there's one way to get zero heads, two ways to get one head, and one way to get two heads. Because each of these combinations is an independent event, with two coins, there are four possible outcomes (HH, HT, TH, and TT). However, if we are just counting heads, then two of the outcomes give the same number of heads (1 for HT and TH).

```
      1
    1   1
 1    2    1
```

The same logic applies as we add more coins: if we flip n 2-sided coins there are 2^n possible outcomes. Few of these combinations yield extreme values for the count of the number of heads–there's only one way to get all (i.e., n) heads or all tails (i.e., 0 heads)–while there are more combinations for intermediate values (e.g., lots of ways to get half heads and half tails). If we assume that each combination of heads and tails is equally likely (i.e., we have a fair coin), and focus on the count of heads, we can see that the Pascal triangle tells us the likelihood of particular count of heads.

1. Create a Pascal's Triangle with this R code:

```r
lapply(0:7, function(i) choose(i, 0:i))
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 1 1
##
## [[3]]
## [1] 1 2 1
##
## [[4]]
## [1] 1 3 3 1
##
```

```
## [[5]]
## [1] 1 4 6 4 1
##
## [[6]]
## [1]  1  5 10 10  5  1
##
## [[7]]
## [1]  1  6 15 20 15  6  1
##
## [[8]]
## [1]  1  7 21 35 35 21  7  1
```

If you were running trials for the final row of the triangle empirically with actual coin tosses, how many coins would you need to throw to conduct each trial? Answer: 128

2. If all is well, the sum of the entries on your final row should be 128 (by the way, that is also two raised to the seventh power). If we considered these entries to be the number of heads across a total of 128 trials, for what number of trials would we have observed three heads? Four heads?

Convert the lowest layer of your triangle to probabilities using this R command:

```r
pascal <- choose(7, 0:7)/128
pascal
```

```
## [1] 0.0078125 0.0546875 0.1640625 0.2734375 0.2734375 0.1640625 0.0546875
## [8] 0.0078125
```

3. Use rbinom() to create a list of ten trials, where each trial consists of seven events (e.g., coin tosses). Set prob=0.50 to use a fair coin.

```r
set.seed(0)
event <- rbinom(n=10, size= 7, prob= 0.5)
```

Did your list of ten include a zero or a seven? Why?

Because the odds are very low a 0 & 7 means all tosses were tails or heads the odds if that happening are very low the odds are (0.5)^7

4. Modify your rbinom() code to run a large empirical simulation of a coin toss with seven events. Run 100,000 trials and save the results in a variable. Create a table showing the number of events for each of the eight outcomes. Hint: in addition to rbinom() you will need the table() command.

```r
set.seed(0)

coins <- rbinom(n=100000, size= 7, prob= 0.5)

table(coins)
```

```
## coins
##     0     1     2     3     4     5     6     7
##   789  5522 16486 27173 27349 16369  5505   807
```

5. Using the capabilities of R, convert the event counts in the table from question 3 into probabilities. Use barplot() to make a graph of the results.

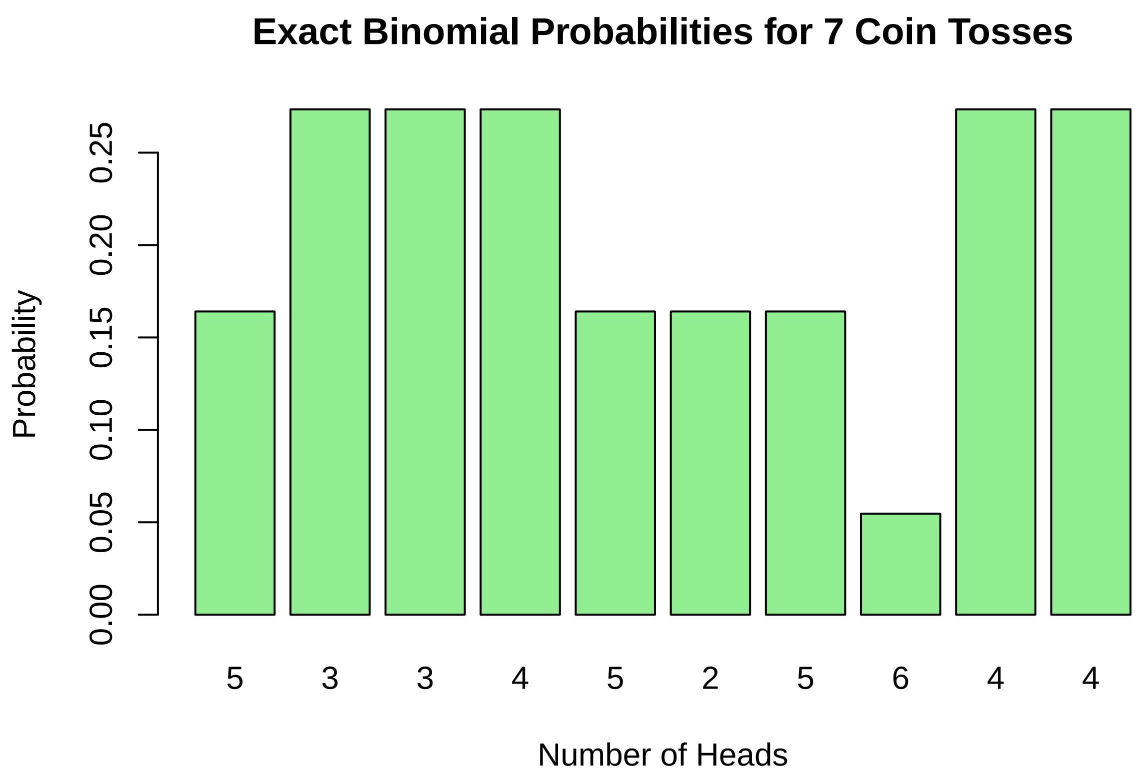
```r
print(event) #outcomes
```

```
##  [1] 5 3 3 4 5 2 5 6 4 4
```

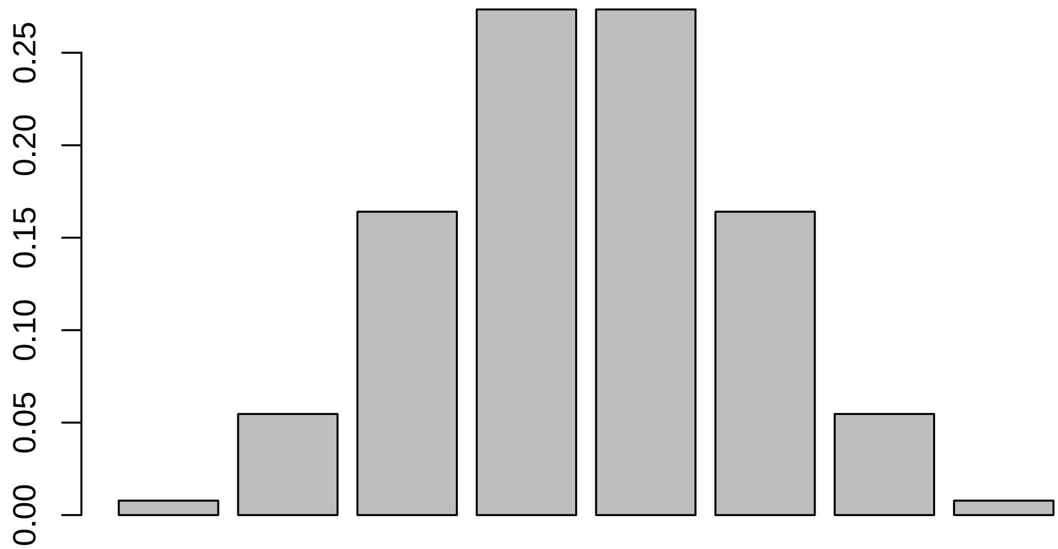
```r
event_prob <- choose(7, event) / 128
print(event_prob)
```

```
##  [1] 0.1640625 0.2734375 0.2734375 0.2734375 0.1640625 0.1640625 0.1640625
##  [8] 0.0546875 0.2734375 0.2734375
```

```r
barplot(event_prob,
      main="Exact Binomial Probabilities for 7 Coin Tosses",
      xlab="Number of Heads",
      ylab="Probability",
      col="lightgreen",
      names.arg = event)
```




```r
barplot(pascal)
```



6. Bonus question: Use the following line of code to produce the theoretical probabilities for a binomial distribution with seven events:

```
dbinom(x=0:7, size=7, prob=0.5)
```

```
## [1] 0.0078125 0.0546875 0.1640625 0.2734375 0.2734375 0.1640625 0.0546875
## [8] 0.0078125
```

```
num_rows <- 7
triangle <- list()

for (n in 0:num_rows) {
  triangle[[n + 1]] <- choose(n, 0:n)
}


probabilities <- lapply(triangle, function(row) {
  row / sum(row)
})
print(probabilities)
```

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

```
## [[2]]
## [1] 0.5 0.5
##
## [[3]]
## [1] 0.25 0.50 0.25
##
## [[4]]
## [1] 0.125 0.375 0.375 0.125
##
## [[5]]
## [1] 0.0625 0.2500 0.3750 0.2500 0.0625
##
## [[6]]
## [1] 0.03125 0.15625 0.31250 0.31250 0.15625 0.03125
##
## [[7]]
## [1] 0.015625 0.093750 0.234375 0.312500 0.234375 0.093750 0.015625
##
## [[8]]
## [1] 0.0078125 0.0546875 0.1640625 0.2734375 0.2734375 0.1640625 0.0546875
## [8] 0.0078125
```

Why are these theoretical probabilities slightly different from your results for question #5?

Also try: http://www.randomservices.org/random/apps/BinomialCoinExperiment.html

Make sure to post your code to the code share page: https://codeshare.io/aJDyRX