Lesson 04: Probability

References

- Black, Chapter 4 Probability (pp. 89-115)
- Davies, Chapter 10 Conditions and Loops (pp. 179-208)
- Library Reserves: Teetor, Chapter 8 Probability (pp. 180-185)

Data sets: shoppers.csv

Description: shoppers.csv contains the dollar amounts spent in a store by individual shoppers during one day.

Exercises:

```
# Read the comma-delimited text file creating a data frame object in R,
# then examine its structure:

shoppers <- read.csv("shoppers.csv")
str(shoppers)
```

```
## 'data.frame':    50 obs. of  1 variable:
##  $ Spending: num  2.32 6.61 6.9 8.04 9.45 ...
```

1) Assume the fifty shoppers exit the store individually in random order.

   a) If one shopper is picked at random, what is the probability of picking a shopper who spent $40 or more dollars? What is the probability of picking a shopper who spent less than $10?

```
# First, the number of shoppers spending $40 or more (TRUE = second column in table)

table(shoppers$Spending >= 40)
```

```
##
## FALSE  TRUE
##    42     8
```

```
# Second, the probability:
sum(shoppers$Spending >= 40)/nrow(shoppers)
```

```
## [1] 0.16
```

```
# Similarly, the probability of picking a shopper who spent < $10:
sum(shoppers$Spending < 10)/nrow(shoppers)
```

```
## [1] 0.1
```

   b) If two shoppers are picked at random, what is the probability the pair will include a shopper who spent $40 or more dollars and one who spent less than $10?

```
# This is the intersection of two events. However, since we are sampling without
# replacement we must take this into account. The easiest way to solve this problem is
# to compute the total possible pairs that might be chosen, and the total number of
# pairs that satisfy the condition.  The ratio gives the probability of picking the
# two shoppers as described. The total number of pairs is n(n-1)/2 and the number that
# satisfies the condition is (number of shoppers purchasing < $10)*(number of shoppers
# purchasing > $40)
```

```r
n <- nrow(shoppers)
n1 <- sum(shoppers$Spending < 10)
n2 <- sum(shoppers$Spending >= 40)

n1*n2/(n*(n-1)/2)
```

```
## [1] 0.03265306
```

Hint : For parts c) and d) it will be necessary to assume sampling without replacement.

c) If two shoppers are picked at random, what is the probability the pair will include two shoppers who spent no less than $10 and no more than $40?

```r
# This situation is different from 1)b) since we are sampling without replacement
# those shoppers that satisfy this condition.  This is important. if there is only one
# shopper that meets the condition, it is impossible to sample a second shopper, so
# the probability would be zero.

m <- sum(((shoppers$Spending >= 10) & (shoppers$Spending <= 40))) # "&" == AND

m*(m-1)/(n*(n-1))
```

```
## [1] 0.5436735
```

d) If four shoppers are picked at random, what is the probability one shopper will have spent less than $10, one shopper will have spent $40 or more dollars and two shoppers will have spent no less than $10 and no more than $40?

```r
successful_combinations <- n1*n2*m*(m-1)/2
total_combinations <- n*(n-1)*(n-2)*(n-3)/(4*3*2*1)

successful_combinations/total_combinations
```

```
## [1] 0.1156752
```

e) If we know a randomly picked shopper has spent more than $30, what is the probability that shopper has spent more than $40?

```r
# Let's try first reducing the set of shoppers to those spending more than $30:
shoppers_more_than_30 <- subset(shoppers, subset = Spending > 30)

# next compute the probability in this smaller set of shoppers
sum((shoppers_more_than_30$Spending > 40) == TRUE) /
  nrow(shoppers_more_than_30)
```

```
## [1] 0.4705882
```

2) Use R to answer the following questions.

a) Draw 100 samples with replacement of size 22 from the 365 integers (i.e. 1,2,..,365). Count the number of samples in which one or more of the numbers sampled is duplicated. Divide by 100 to estimate the probability of such duplication occurring. (If 22 people are selected at random, what is the probability of two or more matching birthdays?)

```r
set.seed(1234)  # set random number seed for reproducibility
count_duplicates <- 0  # initialize count
for (i in 1:100) {
  this_sample <- sample(1:365, size = 22, replace = TRUE)
  if(length(this_sample) != length(unique(this_sample)))
```

```
    count_duplicates <- count_duplicates + 1
}
prob_any_duplicates <- count_duplicates/100
prob_any_duplicates
```

```
## [1] 0.53
```

```
# (If 22 people are selected at random, what is the probability
# of two or more matching birthdays?) This is known as the birthday problem,
# a classic problem in probability. We solved the problem using the method
# above with a for-loop. It can also be solved with one line of R code
# following our setting of the random number seed for reproducibility,
# so we can show that the two results are identical.

set.seed(1234)  # set random number seed for reproducibility
mean(replicate(100,any(duplicated(sample(1:365, 22, replace=TRUE)))))
```

```
## [1] 0.53
```

```
# So, as it turns out, there is about a 50/50 chance of two people having
# the same birthday in a class of 22 students. Let's get a more precise estimate by
# increasing the number of iterations/replications
set.seed(1234)  # set random number seed for reproducibility
mean(replicate(10000,any(duplicated(sample(1:365, 22, replace=TRUE)))))
```

```
## [1] 0.4817
```

b) Suppose that 60% of marbles in a bag are black and 40% are white. Generate a random sample of size 20 with replacement using uniform random numbers. For the numbers in each sample, if a random number is 0.6 or less, code it as a 1. If it is not 0.6 or less code it a zero. Add the twenty coded numbers. Do this 50 times and calculate the proportion of times the sum is 11 or greater. What have you estimated? Expand the number of trials to 10,000. The exact binomial estimated probability is 0.755 and the expectation is 12.

```
# Define a function that codes outcomes as TRUE if <= p and FALSE otherwise.
# The function will also sum up the number of times TRUE occurs and put in count.
count <- function(N,p){
     x <- runif(n = N)
     count <- x <= p
     m <- sum(count)}

set.seed(1234)
result = NULL
for (i in 1:50)
   +{result <- c(result, count(20,0.6))}

# Score results and convert to a probability.
result <- result >= 11
sum(result)/50
```

```
## [1] 0.66
```

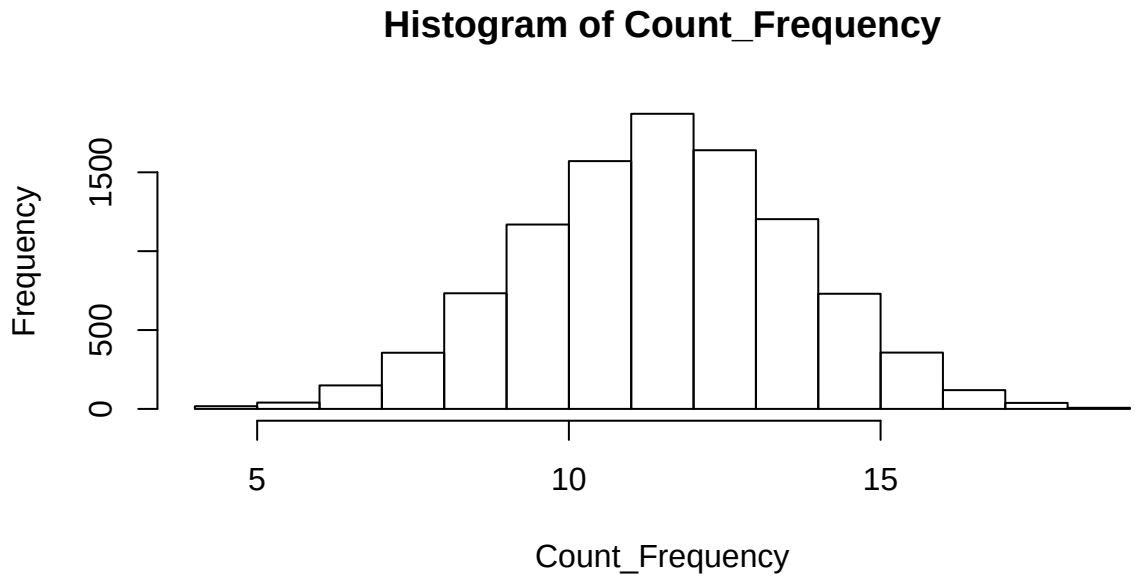
```
# Expand the number of trials to 10000

# Plot a histogram and score results converting to a probability.
set.seed(1234)
result = NULL
```

```
for (i in 1:10000)
  +{result <- c(result, count(20,0.6))}

# Save outcome for later plotting
outcome <- table(result)
outcome <- as.data.frame(outcome)

Count_Frequency <- result
hist(Count_Frequency)
```




```
sum(result)/10000  # This will give the expected value for the distribution.
```

```
## [1] 11.9897
```

```
summary(result)
```

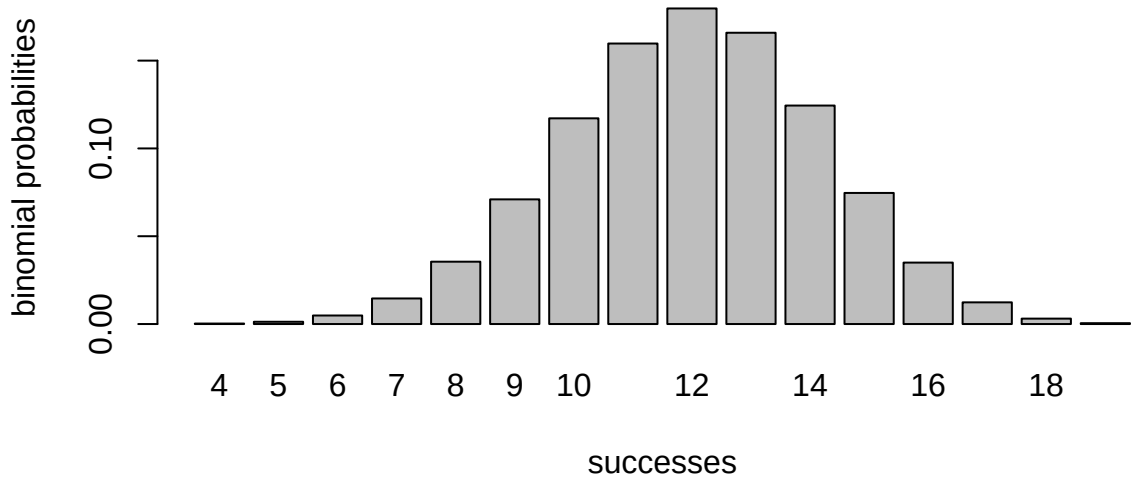
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.00   11.00   12.00   11.99   13.00   19.00
```

```
result <- result >= 11
sum(result)/10000  # This will give the probability of the event >= 11
```

```
## [1] 0.7536
```

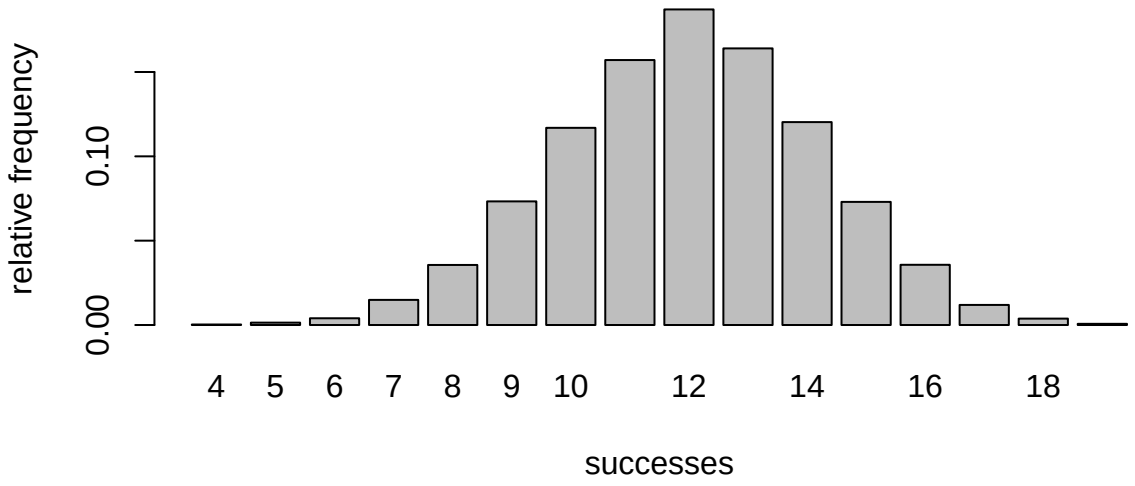
```
# note that R provides binomial probabilties directly
# dbinom(x, size, prob, log = FALSE)
# pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)
# qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)
# rbinom(n, size, prob)
# We use packages and functions in R whenever possible.
# We build upon the foundation that R provides.
# The R environment includes more than five thousand packages,
# many written by the leading experts in statistics and data science.
```

```r
# calculate the exact binomial probabilities
# set trials <- c(0:20) and calculate dbinom(q,size=20,prob=0.6)
trials <- c(0:20)
probabilities <- dbinom(trials,size=20,prob=0.6)
successes <- trials[5:20]
binomial_probabilities <- probabilities[5:20]
successes <- factor(successes)
barplot(binomial_probabilities, names.arg = successes, xlab = "successes",
    ylab = "binomial probabilities")
```



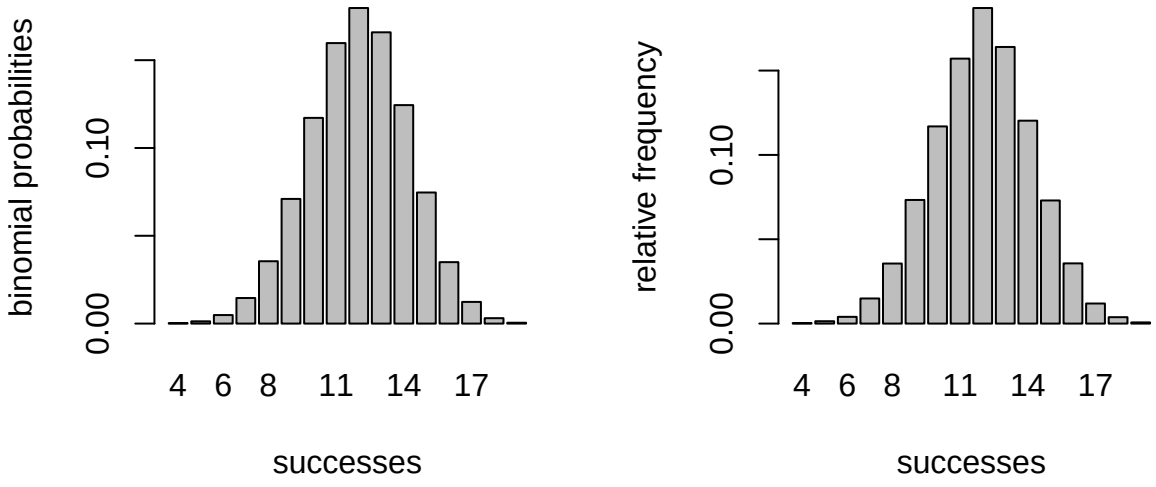

```r
# Return to the simulation results with outcome
# Convert outcome to a dataframe and then compute relative frequencies

relative_frequency <- outcome[,2]/sum(outcome[,2])
successes <- outcome[,1]
successes <- factor(successes)
barplot(relative_frequency, names.arg = successes, xlab = "successes",
    ylab = "relative frequency")
```




```
# Compare the two plots.

par(mfrow=c(1,2))
barplot(binomial_probabilities, names.arg = successes, xlab = "successes",
    ylab = "binomial probabilities")
barplot(relative_frequency, names.arg = successes, xlab = "successes",
    ylab = "relative frequency")
```




```
par(mfrow=c(1,1))
```