stats-lab2

2026-01-22

Setup

Consider a "loaded" die that has a 10% chance of getting a 1-5 and a 50% chance of getting a 6.

(This is from the dishonest casino in the Durbin et al. book; this example is used to define algorithms for Hidden Markov Models in the first few chapters of that text book. We will see a lot more of this die in future labs and lectures).

```r
# Define the faces of the die
die_faces <- 1:6

# Define the probabilities (0.1 for 1-5, and 0.5 for 6)
probabilities <- c(0.1, 0.1, 0.1, 0.1, 0.1, 0.5)

# Safety check: ensure they sum to 1
sum(probabilities)
```

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

Question 1

What is the mean and variance for the loaded dice?

```r
# mean
x_bar <- sum(die_faces * probabilities)

# the analog way as in plug n' chug
#analog_x_bar <- ((0.1 * 1) + (0.1 * 2) + (0.1 * 3) + (0.1 * 4) + (0.1 * 5) + (0.5 * 6))
#analog_x_bar

# variance
var_x <- sum(probabilities * (die_faces^2)) - x_bar^2

# the analog way as in plug n' chug
#analog_var_x <- ((0.1 * (1-x_bar)^2) + (0.1 * (2-x_bar)^2) + (0.1 * (3-x_bar)^2) +
#(0.1 * (4-x_bar)^2) + (0.1 * (5-x_bar)^2) + (0.5 * (6-x_bar)^2))
#analog_var_x

# printing results
as.data.frame(x_bar)
```

```
##   x_bar
## 1   4.5
```

```r
as.data.frame(var_x)
```

```
##   var_x
## 1  3.25
```

Question 2

Make a function in R that "rolls" this dice; return a vector containing the rolls.

```
    So if I call:  myRolls <- rollLoadedDie(10000)

 I would get a vector of size 10,000 that contains the rolls of my loaded die.
```

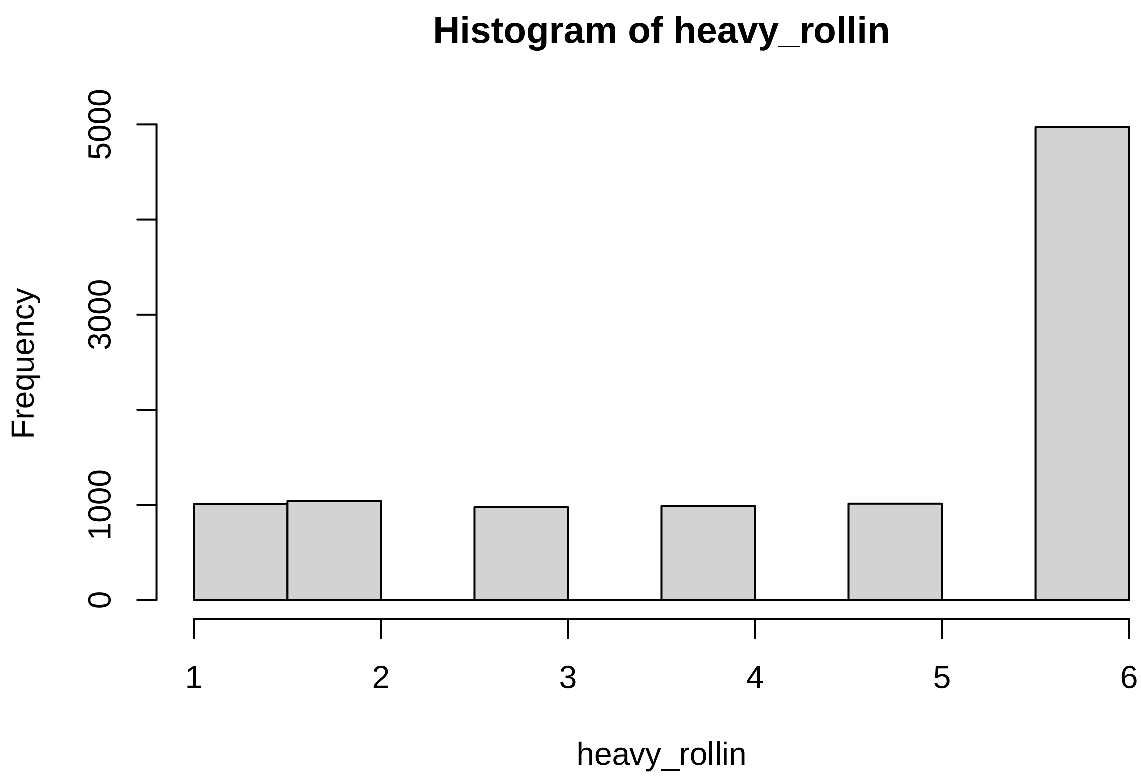
```r
# The function takes 'n' as the number of rolls desired
roll_loaded_die <- function(n) {
  rolls <- sample(
    x = die_faces,       # The faces (1-6)
    size = n,            # How many times to roll
    replace = TRUE,      # We need TRUE because we can roll the same number twice
    prob = probabilities # The specific weights for our loaded die
  )

  return(rolls)
}

# Example: Roll the die 10 times
my_rolls <- roll_loaded_die(10)
print(my_rolls)
```

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

Question 3

Make a histogram of some large number of rolls. Do the rolls of the loaded die approximate a uniform distribution?

```r
heavy_rollin <- roll_loaded_die(10000)
hist(heavy_rollin)
```



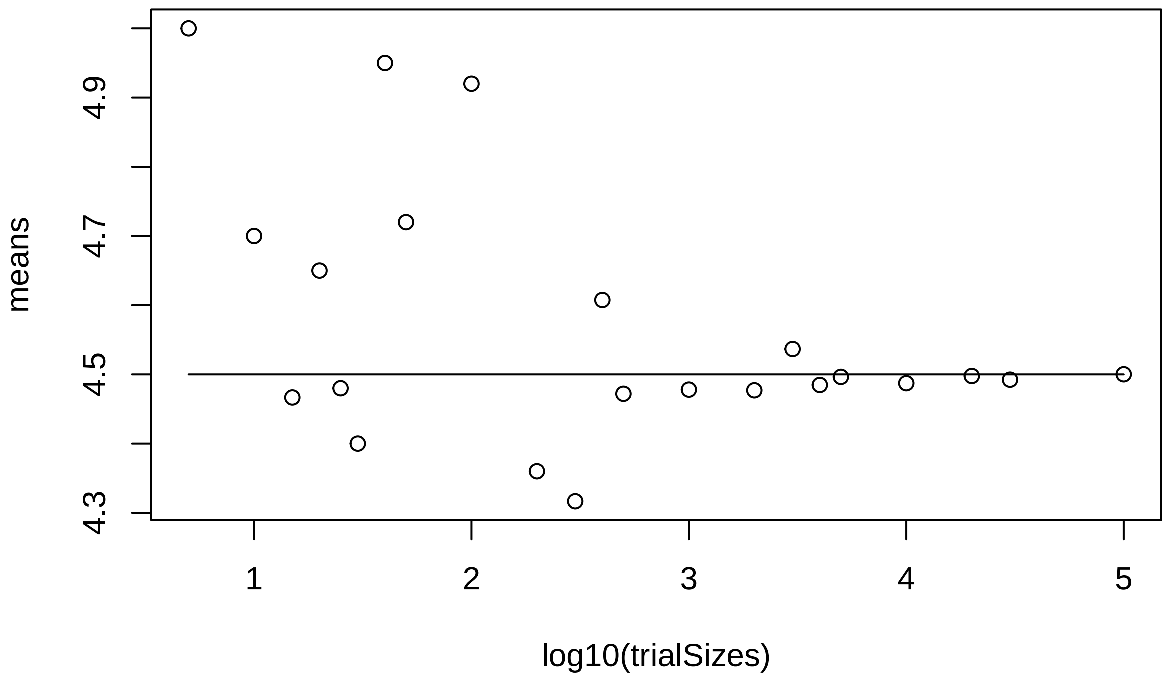
The rolls of the loaded die do NOT approximate a uniform distribution. The frequencies of rolling 1-5 are pretty uniform, but the frequency for rolling a 6 is noticeably higher than any other number.

Question 4

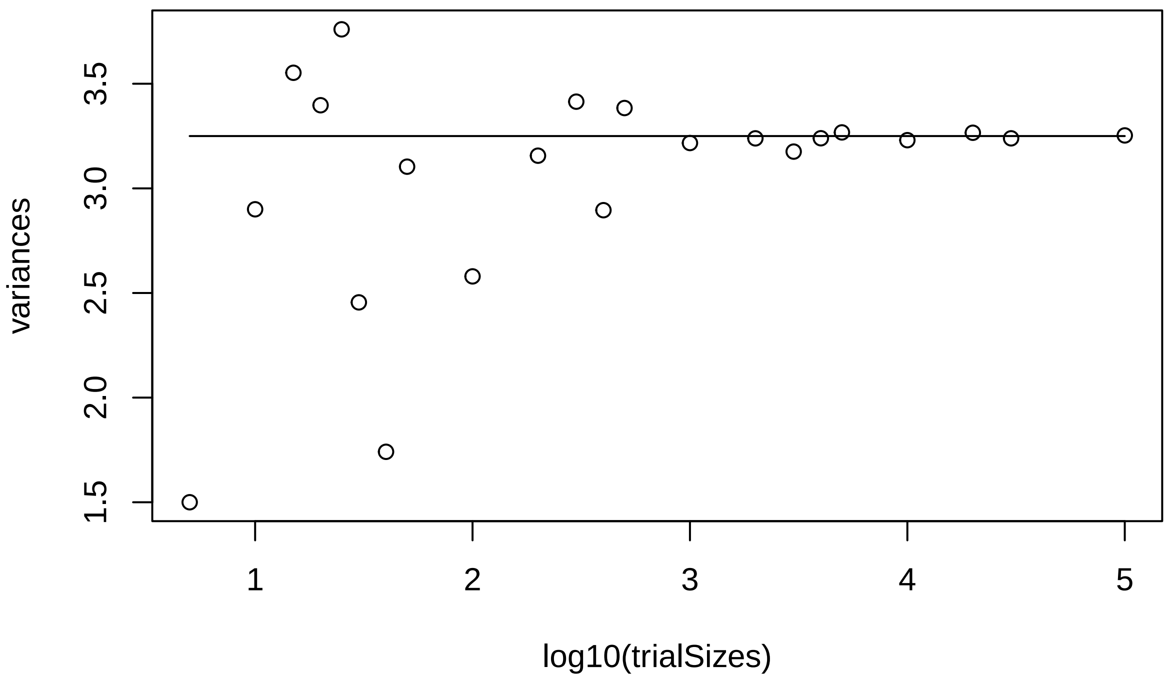
Modify the code on Slide #57 of lecture #2 so that the means vs. trial size plots are from the loaded die. Generate these plots a few times. How many rolls appear to be necessary to get convergence on the expected values for the mean and variance?

```
trialSizes <- c(5, 10, 15, 20, 25, 30, 40, 50, 100, 200, 300, 400, 500, 1000, 2000, 3000,
                4000, 5000, 10000, 20000, 30000, 100000)
means <- vector(mode = "double", length = length(trialSizes))
variances <- vector(mode = "double", length = length(trialSizes))

for( i in 1:length(trialSizes) )
{
  rolls <- vector(length = trialSizes[i], mode = "double")
    rolls <- sample(
    x = die_faces,        # The faces (1-6)
    size = trialSizes[i],# How many times to roll
    replace = TRUE,       # We need TRUE because we can roll the same number twice
    prob = probabilities # The specific weights for our loaded die
  )

  means[i] <- mean(rolls);
  # we use (N-1) in the denominator here bc we are estimating the mean from the data
  variances[i] <- var(rolls)
}
```

```r
plot(log10(trialSizes), means)
lines(log10(trialSizes), rep(4.5, length(trialSizes)))
```




```r
plot(log10(trialSizes), variances)
lines(log10(trialSizes), rep(3.25, length(trialSizes)))
```



It appears that around 1000 rolls are necessary to get convergence on the expected values for the mean and variance.

Disclaimer: AI was used to help generate some code