STAT 80A Final Project

Darren Ung, Phoebe Mason, & Andrew Byi

2025-03-11

Problem 1

```
# Normal roulette normally has 38 pockets, each with probability 1/38.
# For this problem, we bias 0.028 probability for pockets 2, 4, and 21.

pockets <- c("00", 0:36)
pockets
```

```
##  [1] "00" "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13"
## [16] "14" "15" "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28"
## [31] "29" "30" "31" "32" "33" "34" "35" "36"
```

```
biased_pockets <- c(2, 4, 21)
biased_prob <- 0.028

#this is to find probability of unbiased pockets after applying the biased ones
remaining_prob <- 1 - (length(biased_pockets) * biased_prob)
remaining_prob
```

```
## [1] 0.916
```

```
unbiased_count <- length(pockets) - length(biased_pockets)
unbiased_count
```

```
## [1] 35
```

```
prob_each <- remaining_prob / unbiased_count # evenly distribute the remaining
                                             # probability over the other pockets
prob_each
```

```
## [1] 0.02617143
```

```
probs <- ifelse(pockets %in% as.character(biased_pockets), biased_prob, prob_each)
probs
```

```
##  [1] 0.02617143 0.02617143 0.02617143 0.02800000 0.02617143 0.02800000
##  [7] 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143
## [13] 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143
## [19] 0.02617143 0.02617143 0.02617143 0.02617143 0.02800000 0.02617143
## [25] 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143
## [31] 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143 0.02617143
## [37] 0.02617143 0.02617143
```

```r
sum(probs) # check that the probabilities still sum to 1
```

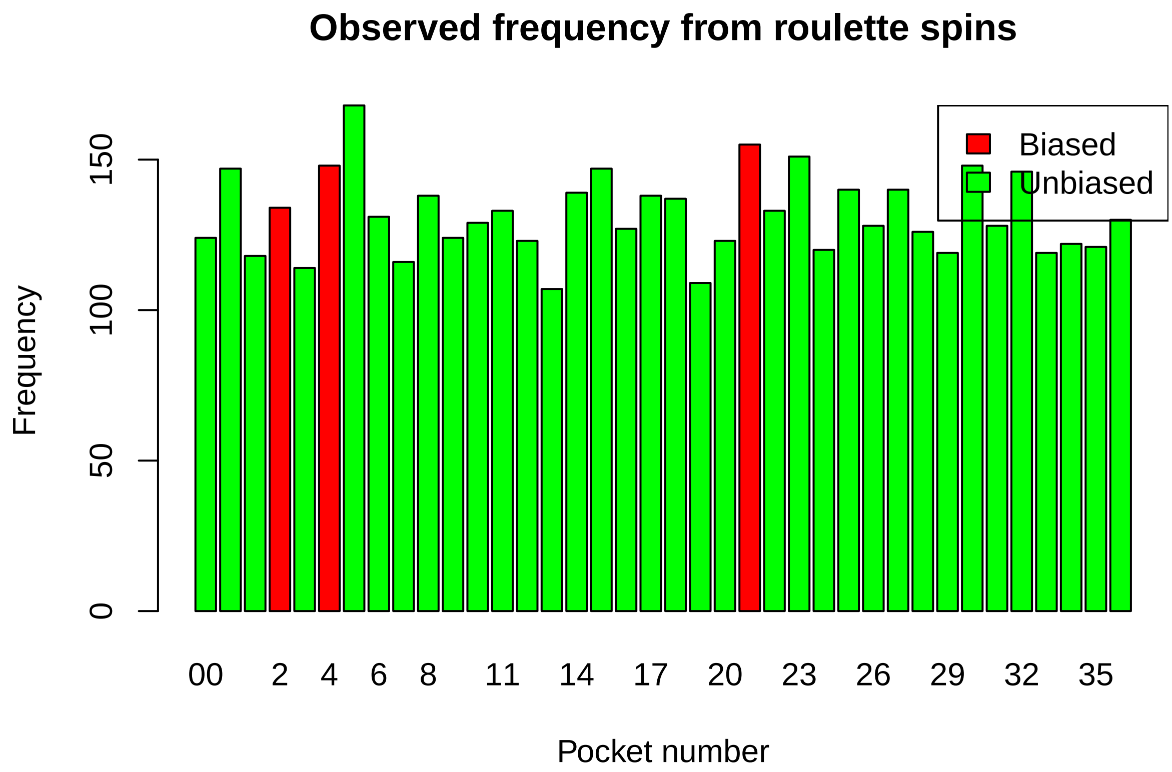
```
## [1] 1
```

```r
# you'll have to modify this when more than one pocket is biased
# ```
#
# ```{r}
# sample like in previous homeworks
num_sims <- 5000
set.seed(123) # set seed for reproducibility
spins <- sample(pockets, size = num_sims, replace = TRUE, prob = probs)
spins <- factor(spins, levels = c("00", 0:36)) # sorts the plot labels from 00 to 36

# calculate expected frequencies for Section 4
expected_freq <- num_sims * probs

# create barplot of observed frequencies
observed_freq <- table(spins)
barplot(observed_freq,
        main = "Observed frequency from roulette spins",
        xlab = "Pocket number",
        ylab = "Frequency",
        col = ifelse(names(observed_freq) %in% as.character(biased_pockets), "red", "green"))

# add a legend
legend("topright",
       legend = c("Biased", "Unbiased"),
       fill = c("red", "green"))
```




```r
# compare observed vs expected frequencies
comparison <- data.frame(
  Pocket = pockets,
  Expected = expected_freq,
  Observed = as.vector(table(factor(spins, levels = pockets)))
)

# show results for biased pockets
biased_results <- comparison[comparison$Pocket %in% as.character(biased_pockets),]
print(biased_results)
```

```
##    Pocket Expected Observed
## 4       2      140      134
## 6       4      140      148
## 23     21      140      155
```

```r
# chi-squared test to detect if there's a significant bias
chi_squared_test <- chisq.test(observed_freq, p = probs)
print(chi_squared_test)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed_freq
## X-squared = 46.495, df = 37, p-value = 0.1362
```

Problem 2

```r
# For each lottery draw, randomly select 5 numbers from the range 1 to 47 and a
# Mega number from the range 1 to 27.

# for one day
set.seed(123)
# generate winning numbers of a single draw
winning_5_numbers <- sample(1:47, 5)
winning_mega_number <- sample(1:27, 1)

# simulate 500,000 lottery draws
prizes <- character(500000)
for(i in 1:500000) {
  player_5_numbers <- sample(1:47, 5)
  player_mega_number <- sample(1:27, 1)

  # match 5 numbers between win and player lottery
  matches_5 <- sum(player_5_numbers %in% winning_5_numbers)
  match_mega <- player_mega_number == winning_mega_number

  # issue out prizes
  if(matches_5 == 5 && match_mega) {
    prizes[i] <- "first prize"
  } else if(matches_5 == 5 && !match_mega) {
    prizes[i] <- "second prize"
  } else if(matches_5 == 4 && match_mega) {
    prizes[i] <- "third prize"
  } else {
    prizes[i] <- "no prize"
  }
}

# calculate simulated prize occurrence
prize_counts <- table(prizes)
print(prize_counts)
```

```
## prizes
##    no prize third prize
##      499998           2
```

```r
# calculate simulated prize probabilities
simulated_probabilities <- prize_counts / 500000
print(simulated_probabilities)
```

```
## prizes
##    no prize third prize
##    0.999996    0.000004
```

```r
# calculate theoretical/analytic probabilities
```

```r
# total combos: C(47, 5) * 27 -- essentially the first prize already
first_prize_theoretical <- 1 / (choose(47, 5) * 27)
# second prize: C(47,5) ways to choose 5 numbers correctly, 26/27 ways to
# choose Mega incorrectly
second_prize_theoretical <- 1 / (choose(47, 5)) * (26 / 27)
# third prize: ways to match 4 out of 5: C(5,4) * C(42,1) = 5 * 42 = 210
# Probability = [C(5,4) * C(42,1)] / [C(47,5)] * (1/27)
third_prize_theoretical <- (choose(5, 4) * choose(42, 1)) / choose(47, 5) * (1 / 27)

# print theoretical probabilities
cat("Theoretical probabilities:\n")
```

```
## Theoretical probabilities:
```

```r
cat("First prize:", first_prize_theoretical, "\n")
```

```
## First prize: 2.414505e-08
```

```r
cat("Second prize:", second_prize_theoretical, "\n")
```

```
## Second prize: 6.277714e-07
```

```r
cat("Third prize:", third_prize_theoretical, "\n")
```

```
## Third prize: 5.070461e-06
```

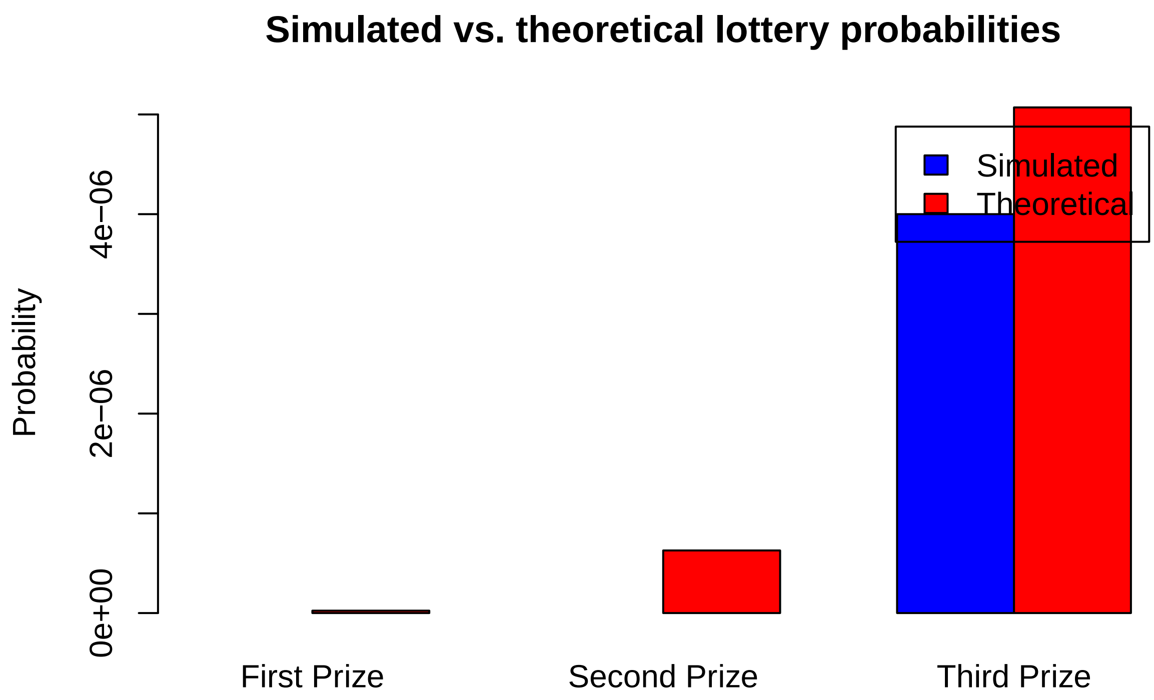
```r
# compare simulated and theoretical probabilities
comparison <- data.frame(
  Prize = c("First Prize", "Second Prize", "Third Prize"),
  Simulated = c(
    simulated_probabilities["first prize"],
    simulated_probabilities["second prize"],
    simulated_probabilities["third prize"]
  ),
  Theoretical = c(
    first_prize_theoretical,
    second_prize_theoretical,
    third_prize_theoretical
  )
)

# column for the ratio of simulated to theoretical
comparison$Ratio <- comparison$Simulated / comparison$Theoretical

# print comparison
print(comparison)
```

```
##          Prize Simulated  Theoretical     Ratio
## 1  First Prize        NA 2.414505e-08        NA
## 2 Second Prize        NA 6.277714e-07        NA
## 3  Third Prize     4e-06 5.070461e-06 0.7888829
```

```
# plot the results for visualization
barplot(
  rbind(comparison$Simulated, comparison$Theoretical),
  beside = TRUE,
  names.arg = comparison$Prize,
  col = c("blue", "red"),
  main = "Simulated vs. theoretical lottery probabilities",
  ylab = "Probability",
  legend.text = c("Simulated", "Theoretical")
)
```



Methodology

In this project, we use R to implement a mathematical model of American roulette and the California lottery to estimate their frequencies and outcomes.

American roulette, Problem 1.

In the American roulette model, we first define the vector [pockets] to represent all 38 pockets on a standard American roulette wheel: 0 to 36, and 00.

We then bias the probabilities for the three pockets [2], [4], and [21] at 0.028 each. Their combined biased probability is 0.028 x 3 = 0.084.

Knowing that the sum of all probabilities must be 1, the remaining probability for the unbiased pockets is 1 - 0.084 is 0.916. We then evenly distribute this adjusted residual probability over the 35 unbiased pockets. 0.916 / 35 is roughly 0.02617.

We create the probability vector [probs] with ifelse(), which assigns the biased and unbiased values accordingly. We verify the sum of [probs] to be 1.

We simulate 5,000 spins with sample(), with replacement. We store it in [spins], and convert it into a factor for labeling. Finally, we visualize the results in a bar plot and a table.

California lottery, Problem 2.

In the California lottery model, a player selects five distinct numbers from 1 to 47, and one Mega number from 1 to 27. We initialize this setup with variables and sampling. Winning numbers are generated with [sample()], though set.seed ensures reproducibility.

In the empiric simulation, we apply a loop that runs 500,000 times, each time generating a set of 5 random numbers from 1 to 47, and a Mega number from 1to 27. We compare these numbers against the predetermined winning numbers to determine the prize tier. In the first prize, all 5 numbers and the Mega number match. In the second prize, all 5 numbers match, except for the Mega number. In the third prize, only 4 of the 5 numbers match, but the Mega number matches. All other outcomes result in no prize. The simulated probabilities are calculated as [observed prize counts] / 500,000 rounds.

In the theoretical/analytic evaluation, we compute exact theoretical probabilities with combinations. First prize probability follows 1 / ((47 choose 5) * 27). Second prize probability follows (1 / 47 choose 5) * (26 / 27); the (26 / 27) term accounts for picking the wrong Mega number. Third prize probability follows (((5 choose 4) * (42 choose 1)) / (47 choose 5)) * (1 / 27). This accounts for the possibilities of choosing only 4 correct numbers and choosing the correct Mega number.

For visualization, we establish a data frame to compare the simulated and theoretical probabilities. We establish a ratio to ensure accuracy (though values that do not exist are N/A), and a barplot allows us to better see the difference.

Results

Please see prior pages for results.

Discussion and Reporting

American roulette, Problem 1.

The biased pockets (2, 4, and 21) do *not* have higher frequencies than the others. Numerically, the discrepancy between biased pockets and unbiased pockets is very small, even relative to the total probability of 1. Between the unbiased pocket probability, roughly 0.02632, and the biased pocket probability, roughly 0.02617, the difference is only 0.00015. Even a chi-square test does not show a statistically significant p-value ($p < 0.05$); our p-value was only 0.1362.

In the real world, statistically significant biases will be much more overt. We can employ observational and statistical methods to detect biases across roulette games. Basic observational methods include but are not limited to mass data collection, physical wheel inspection, and unusual environmental factors (temperature, humidity, table tilt). Statistical methods synergize with prior observational methods, so researchers can apply chi-square tests (as prior), make runs tests for randomness, use Monte Carlo simulations (like this quarter's R labs), and even use Bayesian inference to update prior beliefs about pocket probabilities with present observed data (even from a single, biased game). Other factors such as sample size, wheel maintenance, ball type, and casino techniques may also affect bias detection itself.

California lottery, Problem 2.

In essence, we find that winning the lottery for the first and second prizes are rare events, and that they cannot be observed in smal simulations. Even in the theoretical/analytic probability, these chances are very low. Empiric findings for the third prize match the theoretical probability more closely, though it remained below that of the theoretical bound. Third prize wins occured only 2 times out of 500,000 trials, having about a 0.7888829 ratio to the bound.

Our bar chart visually confirms this discrepancy between first/second prizes, and third prizes. Per the law of large numbers, higher simulation counts (e.g. 1e6, 1e9), may provide better accuracy, but this does not disprove the simple fact that the first/second prizes themselves are highly rare. Our findings strongly confirm winning a major lottery prize is nearly impossible, and our somewhat smaller sample size aligns well with theoretical expectations for moderately rare occurences.