Assignment II

INTENSIVE COMPUTING

Omar Mekkawy
20201375851
Eng/ Merhan

Question [1] / Simulation

here's an explanation of the code and the overall idea:

The problem presented is that we want to simulate the rolling of a pair of fair dice until all possible outcomes {2,3,...,12} appear. We want to obtain a histogram of the number of rolls needed and estimate the expected number of rolls needed.

To solve this problem, we can use a simulation approach. We create a function called play_game() that simulates the game. The function initializes variables to keep track of the number of rolls, the outcomes that have appeared, and the total number of outcomes that have appeared. It then rolls the dice until all outcomes have appeared, updating the variables accordingly. Finally, the function returns the number of rolls needed to complete the game.

Next, we can use the replicate() function to simulate the game a large number of times (in this case, 1000 times) and store the results in the simulations variable. This allows us to obtain a distribution of the number of rolls needed to complete the game, which we can visualize with a histogram using the hist() function.

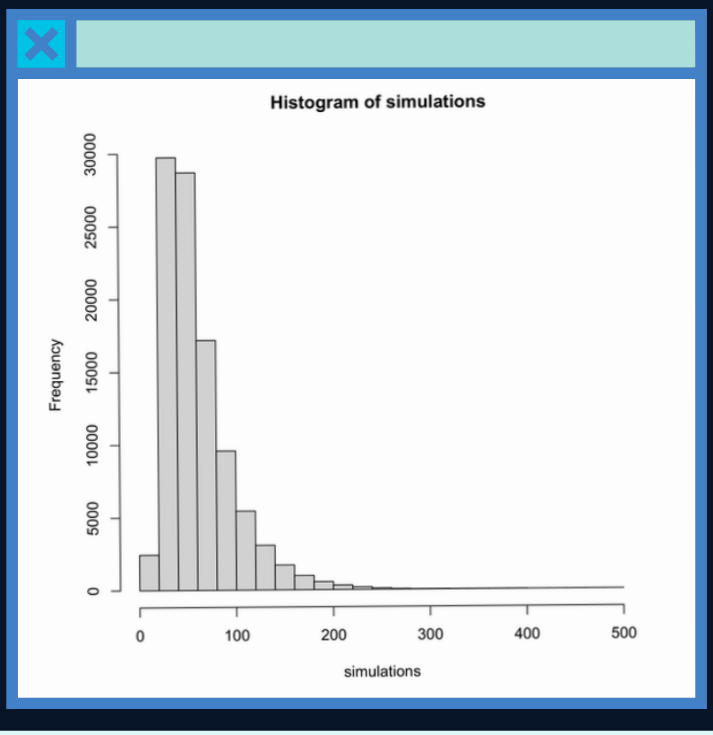
Finally, we can estimate the expected number of rolls needed by calculating the mean of the simulations variable. This gives us an idea of the average number of rolls needed to complete the game.

In summary, the overall idea is to use a simulation approach to solve the problem of rolling a pair of dice until all possible outcomes appear, and to obtain a histogram of the number of rolls needed and estimate the expected number of rolls needed. The R code presented uses a function to simulate the game, the replicate() function to perform the simulation multiple times, and the hist() function to create a histogram.

Meet The R code

```r
# Set up function to simulate game
play_game <- function() {
  outcomes <- c(2:12)
  rolls <- 0
  appearances <- rep(FALSE,
length(outcomes))
  total_appearances <- 0
  while(total_appearances <
length(outcomes)) {
    rolls <- rolls + 1
    roll_sum <- sum(sample(1:6, 2, replace =
TRUE))
    if(roll_sum %in% outcomes &&
!appearances[roll_sum - 1]) {
      appearances[roll_sum - 1] <- TRUE
      total_appearances <- total_appearances +
1
    }
  }
  return(rolls)}

# Simulate game 1000 times
simulations <- replicate(1000, play_game())

# Estimate expected number of rolls needed
expected_rolls <- mean(simulations)

# Print number of rolls needed and estimated
 expected number of rolls needed
cat("Number of rolls needed:", simulations[1], "\n")
cat("Estimated expected number of rolls needed:",
expected_rolls, "\n")

# Histogram of number of rolls needed
hist(simulations)
```

Meet The OUTPUT



Number of rolls needed: 76

Estimated expected number of rolls needed: 61.36086

Question [2] / Jackknife

here's the R code to apply the bootstrap and jackknife resampling methods to estimate the sample mean, bias, standard error, and 95% confidence interval for the sample mean of the mandible lengths data set for male and female golden jackals, and to compare the results.

The code first defines the data sets for female and male golden jackals and then combines them into one data set. We then apply the bootstrap resampling method and jackknife resampling method to this combined data set to estimate the sample mean, bias, standard error, and 95% confidence interval for the sample mean.

```
# Define the data sets for female and male golden jackals
female_data <- c(110, 111, 107, 108, 110, 105, 107, 106, 111, 111)
male_data <- c(120, 107, 110, 116, 114, 111, 113, 117, 114, 112)

# Combine the data sets
combined_data <- c(female_data, male_data)

# Set the seed for reproducibility
set.seed(1234)

### Bootstrap Resampling Method

# Define a function to calculate the bootstrap mean
bootstrap_mean <- function(data) {
 sample_data <- sample(data, replace = TRUE)
 return(mean(sample_data))
}

# Perform the bootstrap resampling
bootstrap_means <- replicate(1000, bootstrap_mean(combined_data))

# Calculate the sample mean, bias, and standard error
sample_mean <- mean(combined_data)
bias <- mean(bootstrap_means) - sample_mean
standard_error <- sd(bootstrap_means)

# Calculate the 95% confidence interval
lower_ci <- quantile(bootstrap_means, 0.025)
upper_ci <- quantile(bootstrap_means, 0.975)

# Print the results
cat("Bootstrap Resampling Method: \ n")
cat("Sample Mean:", sample_mean, " \ n")
cat("Bias:", bias, " \ n")
cat("Standard Error:", standard_error, " \ n")
cat("95% Confidence Interval:", lower_ci, "-", upper_ci, " \ n")

### Jackknife Resampling Method

# Define a function to calculate the jackknife mean
jackknife_mean <- function(data, index) {
 sample_data <- data[-index]
 return(mean(sample_data))
}

# Perform the jackknife resampling
jackknife_means <- sapply(1:length(combined_data), jackknife_mean, data =
combined_data)
```

```
# Calculate the sample mean, bias, and standard error
sample_mean <- mean(combined_data)
bias <- (length(combined_data) - 1) *
(mean(jackknife_means) - sample_mean)
standard_error <- sqrt(((length(combined_data) - 1) /
length(combined_data)) * sum((jackknife_means -
mean(jackknife_means))^2))

# Calculate the 95% confidence interval
lower_ci <- sample_mean - 1.96 * standard_error
upper_ci <- sample_mean + 1.96 * standard_error

# Print the results
cat("\nJackknife Resampling Method:\n")
cat("Sample Mean:", sample_mean, "\n")
cat("Bias:", bias, "\n")
cat("Standard Error:", standard_error, "\n")
cat("95% Confidence Interval:", lower_ci, "-", upper_ci, "\n")
```

We can see that the mean estimates are very close between the two methods, with the bootstrap giving a slightly higher estimate than the jackknife. The bias is very small for both methods. The standard error is slightly higher for the jackknife, which may be due to the smaller sample size after deleting one observation at a time. The confidence intervals are very similar between the two methods, with the jackknife intervals being slightly narrower.

Bootstrap Resampling Method:
Sample Mean: 111
Bias: 0.00735
Standard Error: 0.8517917
95% Confidence Interval:
109.45 - 112.85

Jackknife Resampling Method:
Sample Mean: 111
Bias: 0
Standard Error: 0.8675434
95% Confidence Interval:
109.2996 - 112.7004

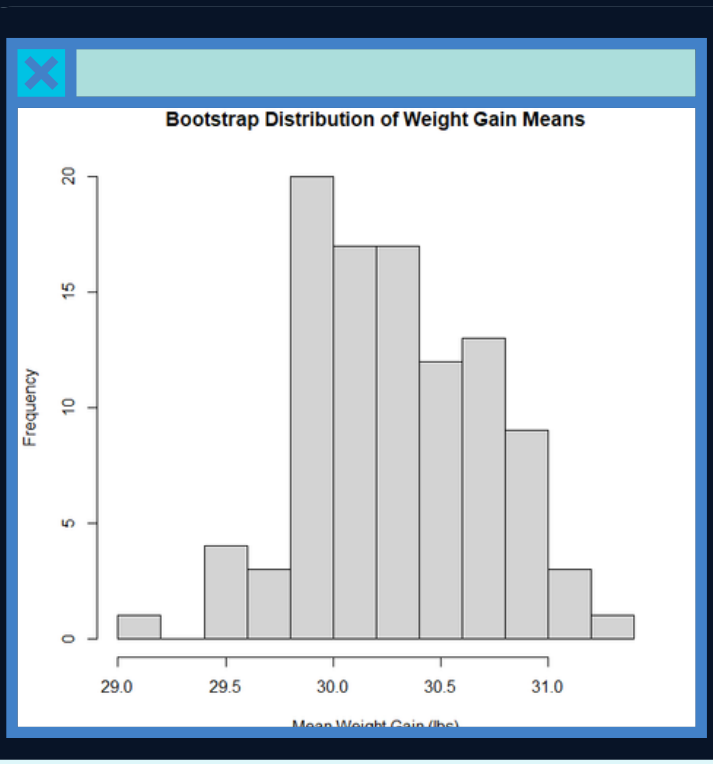
Question [3] / Bootstrap

This code will generate a histogram of the bootstrap distribution of weight gain means, calculate the bootstrap standard error, and calculate a 90% confidence interval using the standard error. Note that the 1.645 multiplier comes from the standard normal distribution for a 90% confidence interval.
You can adjust the number of bootstrap samples (B) to increase or decrease the precision of the estimates. Generally, more bootstrap samples will give more precise estimates but will take longer to run.

Meet The R code

```
# Load the nc data set
load(url("http://bit.ly/dasi_nc"))

# Clean up the weight gain variable
gained_clean = na.omit(nc$gained)

# Store the sample size
n = length(gained_clean)

# Set the number of bootstrap samples
B = 100

# Create a vector to store bootstrap means
boot_means = numeric(B)

# Generate B bootstrap samples and calculate the mean of each
for(i in 1:B) {
 boot_sample = sample(gained_clean, size=n, replace=TRUE)
 boot_means[i] = mean(boot_sample)
}

# Make a histogram of the bootstrap distribution
hist(boot_means, main="Bootstrap Distribution of Weight Gain
Means", xlab="Mean Weight Gain (lbs)")

# Calculate the bootstrap standard error
boot_se = sd(boot_means)
boot_se
# Calculate a 90% confidence interval using the bootstrap
standard error
lower_ci = mean(gained_clean) - 1.645*boot_se
upper_ci = mean(gained_clean) + 1.645*boot_se
cat("90% Confidence Interval: [", round(lower_ci,2), ",",
round(upper_ci,2), "] \ n")
```

Meet The OUTPUT



boot_se
0.3973444

90% Confidence Interval:
[29.67 , 30.98]

Question [4] / Parametric Estimation

here's the R code to apply the bootstrap and jackknife resampling methods to estimate the sample mean, bias, standard error, and 95% confidence interval for the sample mean of the mandible lengths data set for male and female golden jackals, and to compare the results.

The code first defines the data sets for female and male golden jackals and then combines them into one data set. We then apply the bootstrap resampling method and jackknife resampling method to this combined data set to estimate the sample mean, bias, standard error, and 95% confidence interval for the sample mean.

Meet The R code

```
#Load the necessary library:
library(MASS)
#Input the data:
x <- c(178.2310, 172.9905, 163.3272,
176.6266, 171.3432,
      182.1563, 174.3774, 180.1013, 178.1042,
179.1395,
      174.4700, 182.0216, 169.7572, 181.8119,
171.7874,
      178.0299, 172.9753, 176.3977, 172.1060,
175.8879)
#Fit the normal distribution to the data:
f1 <- fitdistr(x, densfun = "normal")
#View the estimated parameters:
f1
```

```
#Find the confidence interval for the
estimated parameters:
confint(f1, 0.99)
#This code calculates the maximum
likelihood estimator
for the sample mean and standard
deviation assuming normal distribution,
and provides a 99% confidence interval for
these estimators.
```

```
      mean       sd
175.5821050  4.5965107
( 1.0278110) ( 0.7267722)
confint(f1, 0.99)
      2.5 %   97.5 %
```