Convergence and simulations

TASK 1

1.) Out of the two following sequences, which one is more likely to be generated from a random fair coin toss (H or T with equal probability) and which one is more likely to be fabricated? Explain your reasoning.

Following the logic that probability of H and T should be uniformly distributed with P=0.5 on each trial, let's identify probability of outcomes in each sequence. If probability signigicantly deviates from 0.5, it will mean that sequence is fabricated.

```
sequence1 <- c("T", "T", "H", "T", "H",
sequence2 <- c("H", "H", "H", "T", "H", "T",
count_H_sequence1 <- sum(sequence1 == "H")</pre>
count_T_sequence1 <- sum(sequence1 == "T")</pre>
count_H_sequence2 <- sum(sequence2 == "H")</pre>
count_T_sequence2 <- sum(sequence2 == "T")</pre>
prob_H_sequence1 <- count_H_sequence1 / length(sequence1)</pre>
prob_T_sequence1 <- count_T_sequence1 / length(sequence1)</pre>
prob_H_sequence2 <- count_H_sequence2 / length(sequence2)</pre>
prob_T_sequence2 <- count_T_sequence2 / length(sequence2)</pre>
cat("Probability of H in Sequence 1:", prob_H_sequence1)
## Probability of H in Sequence 1: 0.55
cat("Probability of T in Sequence 1:", prob_T_sequence1)
## Probability of T in Sequence 1: 0.45
```

```
cat("Probability of H in Sequence 2:", prob_H_sequence2)
```

```
## Probability of H in Sequence 2: 0.49
```

```
cat("Probability of T in Sequence 2:", prob_T_sequence2)
## Probability of T in Sequence 2: 0.51
```

In Sequence 2, the probabilities are closer to 0.5 for both H and T, which is in line with what is expected in a fair coin toss. Hence, Sequence 2 less likely to be fabricated based on the calculations.

In Sequence 1, the higher probability of H and lower probability of T result in deviation from the expected 0.5 probability for a fair coin toss. This

TASK 2

2.) Generate 1,000 numbers from continuous uniform distribution with bounds 0 and 1. Choose a value of parameter π from the interval (0; 1) (use at least three digits). From the 1,000 generated random numbers create Bernoulli distribution with value 1 if number is lesser than π and 0 if it is

1.0

0.8

0.7

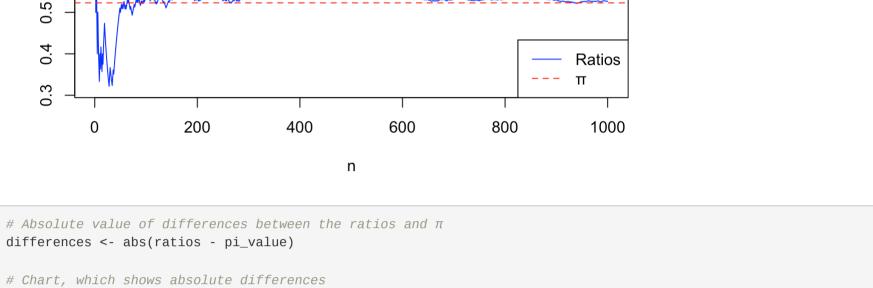
9 0

might be an indication that the sequence is more likely fabricated.

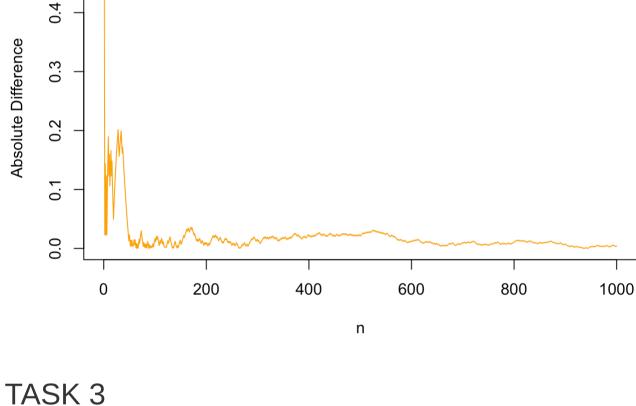
greater. Next calculate ratio of 1s among 1, 2, 3... 1 000 random numbers (so you have 1 000 such a ratios). Create chart showing convergence of the ratios to π with rising n. Next calculate absolute value of differences between the ratios and π and again create chart. set.seed(123)

```
# 1,000 numbers from continuous uniform distribution
uniform_numbers <- runif(n, min = 0, max = 1)
# Bernoulli distribution
pi_value <- 0.523
bernoulli_distribution <- as.integer(uniform_numbers < pi_value)</pre>
# Ratios of 1s with rising n
ratios <- cumsum(bernoulli_distribution) / seq_along(bernoulli_distribution)</pre>
# Chart, which shows convergence of the ratios to \pi with rising n
plot(1:n, ratios, type = "l", col = "blue", xlab = "n", ylab = "Ratio", main = "Convergence of Ratios to \pi")
abline(h = pi_value, col = "red", lty = 2)
legend("bottomright", legend = c("Ratios", "\pi"), col = c("blue", "red"), lty = 1:2)
```

Convergence of Ratios to π



```
plot(1:n, differences, type = "l", col = "orange", xlab = "n", ylab = "Absolute Difference", main = "Absolute Dif
ferences from \pi")
                           Absolute Differences from \pi
```



sequence of means, which should show convergence of them to the selected expected value. Next calculate absolute value of differences between these means and selected expected value and again create a chart.

variance <- 0.1

0.15

0.10

0.05

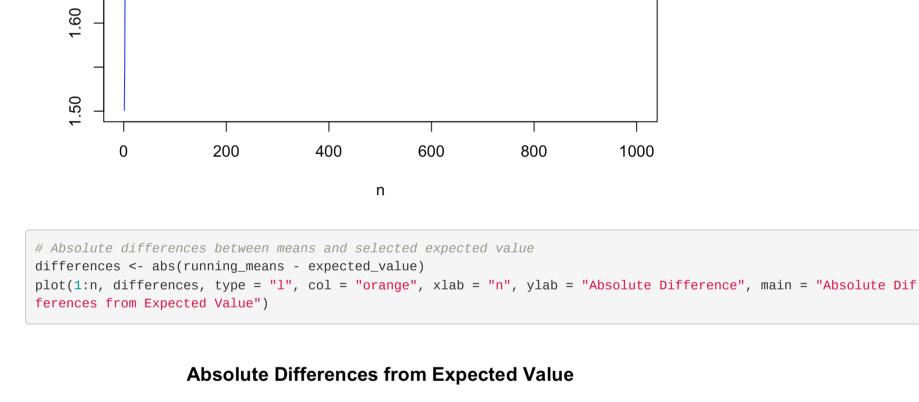
Absolute Difference

set.seed(123) # 1,000 numbers from normal distribution n <- 1000 expected_value <- 1.678</pre>

calculate mean from first, first two, first three..., thousand of these random numbers (so you have 1,000 such means). Create a chart showing the

3.) Generate 1,000 numbers from normal distribution (select the expected value by your own, use 3 digits; use variance equal to 0.1). Then

```
normal_numbers <- rnorm(n, mean = expected_value, sd = sqrt(variance))</pre>
# Running means
running_means <- cumsum(normal_numbers) / seq_along(normal_numbers)</pre>
# Sequence of means
plot(1:n, running_means, type = "l", col = "blue", xlab = "n", ylab = "Mean", main = "Sequence of Means")
                                 Sequence of Means
```





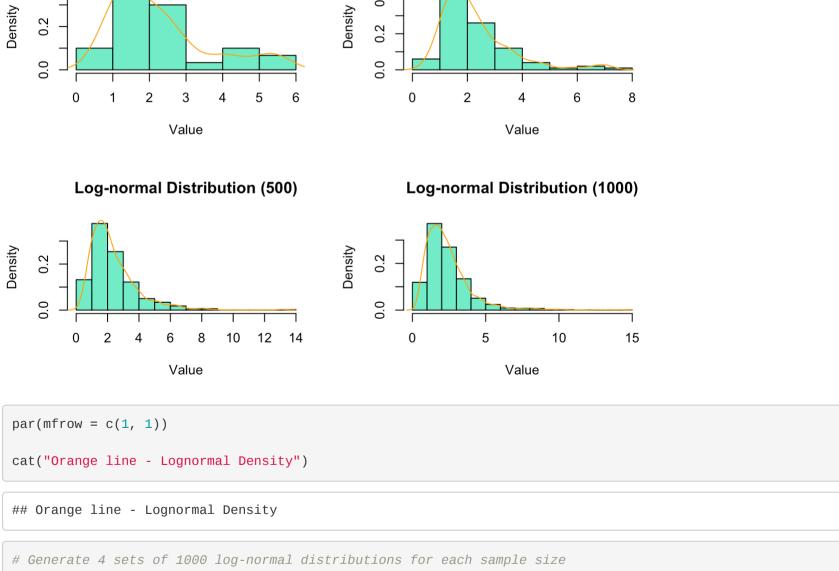
0.00 0 200 400 600 800 1000 n TASK 4 4.) Generate subsequently 30, 100, 500 and 1,000 numbers from log-normal distribution (choose your own value of μ with at least three digits and use σ = 0.1). Plot four histograms of these generated distributions together with originating density. Then generate 4x1,000 of these distributions and for each of them calculate sample mean (hence you will have 1,000 sample means from distributions of sample size 30, another 1,000 for sample size 100 and so on). Plot histogram of these sample means (four plots – for four different sample sizes). What do you observe? set.seed(123) # Expected value and variance pi_value <- 0.678 var <- 0.34 # Log-normal distributions

log_normal_dist_30_nums <- rlnorm(30, meanlog = pi_value, sdlog = sqrt(var))</pre> log_normal_dist_100_nums <- rlnorm(100, meanlog = pi_value, sdlog = sqrt(var))</pre> log_normal_dist_500_nums <- rlnorm(500, meanlog = pi_value, sdlog = sqrt(var))</pre> log_normal_dist_1000_nums <- rlnorm(1000, meanlog = pi_value, sdlog = sqrt(var))</pre>

Plots, which show histograms and originating density

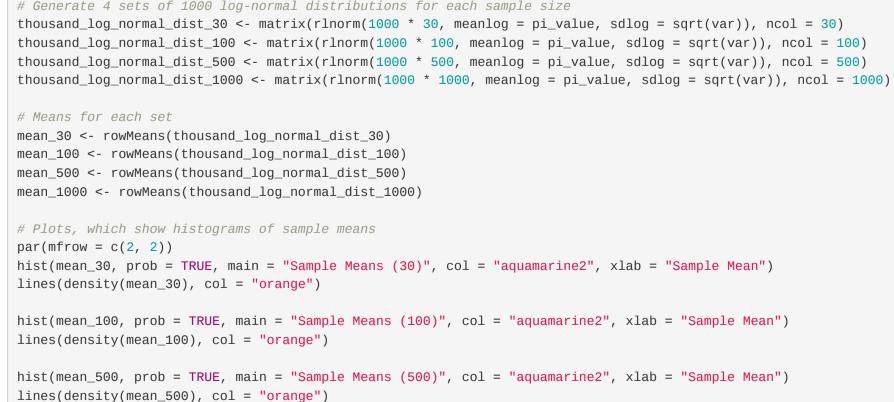
par(mfrow = c(2, 2))

```
hist(log_normal_dist_30_nums, prob = TRUE, main = "Log-normal Distribution (30)", col = "aquamarine2", xlab = "Va
lines(density(log_normal_dist_30_nums), col = "orange")
hist(log_normal_dist_100_nums, prob = TRUE, main = "Log-normal Distribution (100)", col = "aquamarine2", xlab =
lines(density(log_normal_dist_100_nums), col = "orange")
hist(log_normal_dist_500_nums, prob = TRUE, main = "Log-normal Distribution (500)", col = "aquamarine2", xlab =
lines(density(log_normal_dist_500_nums), col = "orange")
hist(log_normal_dist_1000_nums, prob = TRUE, main = "Log-normal Distribution (1000)", col = "aquamarine2", xlab =
"Value")
lines(density(log_normal_dist_1000_nums), col = "orange")
        Log-normal Distribution (30)
                                                   Log-normal Distribution (100)
```

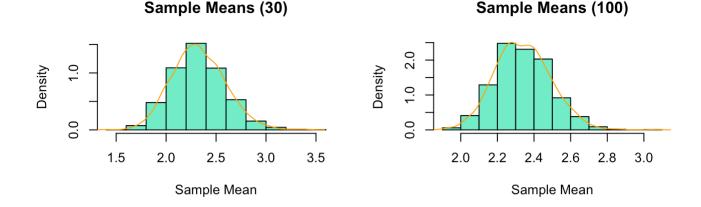


0.4

0.2



hist(mean_1000, prob = TRUE, main = "Sample Means (1000)", col = "aquamarine2", xlab = "Sample Mean")



lines(density(mean_1000), col = "orange")

Sample Means (500)

Density

In a given example we could observe outcome of Central Limit Theorem, which states that regardless of origination of the distribution, sampling distribution of the mean will be normal.

Sample Means (1000)