HOMEWORK REPORT

Submitted by Group 13

**Members:Aman Sanwal, Arihant Sathpathy,
Lipika Bagai,Ankita Singh and Gowthamy S**

QUESTION 1

Estimation of the Constant e using Various Computational Methods

Here we explore and document five different computational methods to estimate the value of the mathematical constant e (approximately 2.71828182...). The methods leverage different principles and techniques, with their implementation in R programming and benchmarking of computational efficiency. Additionally, the mathematical proofs behind each method are provided.

1 Method 1: Logarithmic Transformation and Averaging

1.1 Description

This method is based on the logarithmic transformation of uniformly distributed random numbers. The process involves:

1. Generating 10,000 random numbers between 0 and 1 for 1,000 iterations.

2. Transforming these numbers using the base-10 logarithm.

3. Averaging the transformed values and computing an intermediate value:

$$\left(\frac{1}{10^{\text{mean}}}\right)^{\frac{1}{10}}.$$

4. Averaging the intermediate values across iterations to estimate e.

1.2 Mathematical Proof

1.3 1. Uniform Random Variable Properties

Let $X \sim U(0,1)$, meaning X is uniformly distributed over the interval $(0,1)$. The probability density function (PDF) of X is:

$$f_X(x) = \begin{cases} 1, & 0 < x < 1, \\ 0, & \text{otherwise.} \end{cases}$$

1.4 2. Logarithmic Transformation

The transformation $Y = \log_{10}(X)$ is applied. Using the change-of-variable formula, the PDF of Y can be derived:

$$f_Y(y) = f_X(X) \cdot \left| \frac{dX}{dY} \right|, \quad \text{where } X = 10^Y.$$

Since $f_X(X) = 1$ for $0 < X < 1$, and $\frac{dX}{dY} = 10^Y \ln(10)$, the PDF of Y becomes:

$$f_Y(y) = \ln(10) \cdot 10^y, \quad -\infty < y < 0.$$

1.5 3. Expected Value of Logarithmic Transformation

The expected value of Y is computed as:

$$E[Y] = \int_{-\infty}^{0} y \cdot f_Y(y)\, dy.$$

Simplifying:

$$E[Y] = \int_{-\infty}^{0} y \cdot \ln(10) \cdot 10^y\, dy.$$

By integration by parts, setting $u = y$ and $dv = \ln(10) \cdot 10^y\, dy$:

$$E[Y] = \ln(10) \int_{-\infty}^{0} y \cdot 10^y\, dy = -\frac{1}{\ln(10)}.$$

Thus:

$$E[Y] = -\frac{1}{\ln(10)}.$$

1.6 4. Exponentiating the Mean

The result $-\frac{1}{\ln(10)}$ corresponds to the logarithmic mean. By exponentiating 10^{mean}, we approximate the reciprocal of the natural logarithm base e. Specifically:

$$\text{Approximation of } e = 10^{E[Y]} = 10^{-\frac{1}{\ln(10)}}.$$

1.7 5. Convergence to e

As the number of samples increases (i.e., $n \to \infty$), the Weak Law of Large Numbers ensures that the sample mean converges to the true expected value:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \log_{10}(X_i) = E[Y].$$

Exponentiating this result gives a value that converges to e.

1.8 Results

Estimated value of e: 2.7182

1.9 Visualization

A histogram visualizes the distribution of intermediate values, with a red dashed line indicating their mean.

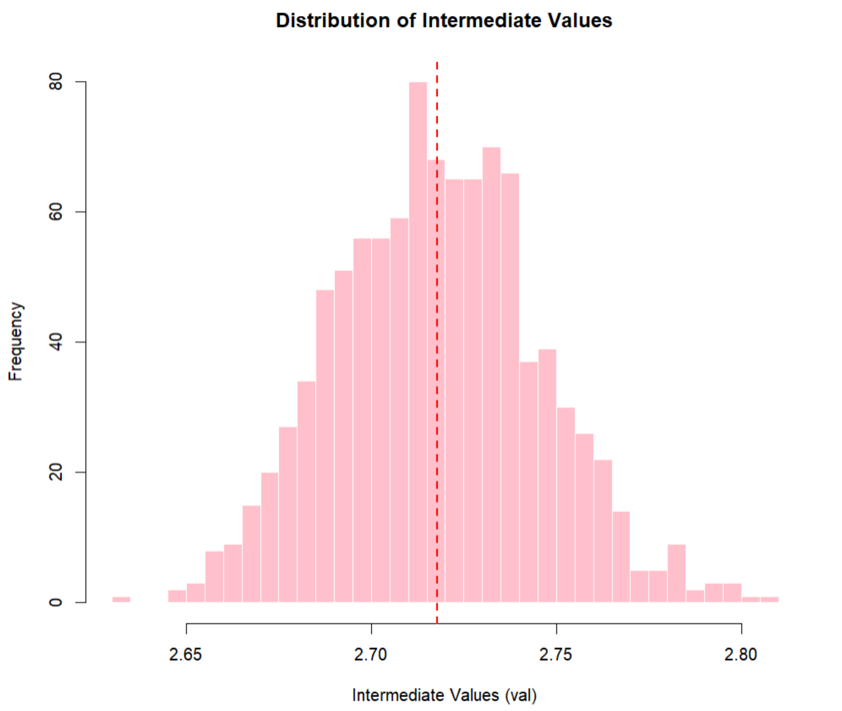

Figure 1: Histogram of Intermediate Values for Method 1

2 Method 2: Random Summation Method

2.1 Description

This Monte Carlo simulation estimates e by:

1. Adding uniformly distributed random numbers until their cumulative sum exceeds 1.
2. Counting the number of iterations required to reach this threshold.
3. Repeating the process 1,000,000 times.
4. Averaging the counts to approximate e.

2.2 Mathematical Proof

Proof: Expected Number of Draws to Exceed 1 is e

Let $X_1, X_2, \ldots$ be i.i.d. random variables drawn from the Uniform$[0, 1]$ distribution. Define N as the smallest integer such that the cumulative sum exceeds 1:

$$N = \min\{n : S_n = X_1 + X_2 + \cdots + X_n > 1\}.$$

We aim to compute the expected value of N, $E[N]$, and show that it equals e.

2.3 Step 1: Key Observation

The cumulative sum $S_n = X_1 + X_2 + \cdots + X_n$ exceeds 1 for the first time at $N = n$ if:

1. $S_{n-1} \leq 1$, and
2. $S_n > 1$.

This means that the event $N = n$ corresponds to the sum of the first $n - 1$ variables being at most 1, and the n-th variable pushing the total over 1.

2.4 Step 2: Probability of $N = n$

The probability of $N = n$ can be written as:

$$P(N = n) = P(S_{n-1} \leq 1) \cdot P(S_{n-1} + X_n > 1 \mid S_{n-1} \leq 1).$$

Since $X_n \sim$ Uniform$[0, 1]$ and is independent of S_{n-1}, we can simplify:

$$P(S_{n-1} + X_n > 1 \mid S_{n-1} \leq 1) = E[1 - S_{n-1}] = 1 - E[S_{n-1}].$$

The probability $P(N = n)$ can be shown to simplify to:

$$P(N = n) = \frac{1}{n!}.$$

2.5 Step 3: Expected Value of N

The expected value of N is given by:

$$E[N] = \sum_{n=1}^{\infty} n \cdot P(N = n).$$

Substituting $P(N = n) = \frac{1}{n!}$, we have:

$$E[N] = \sum_{n=1}^{\infty} n \cdot \frac{1}{n!}.$$

Using the identity $n \cdot \frac{1}{n!} = \frac{1}{(n-1)!}$, we rewrite the sum:

$$E[N] = \sum_{n=1}^{\infty} \frac{1}{(n-1)!}.$$

Reindex the summation with $k = n - 1$, so $k \geq 0$:

$$E[N] = \sum_{k=0}^{\infty} \frac{1}{k!}.$$

2.6 Step 4: Recognizing the Series for e

The series $\sum_{k=0}^{\infty} \frac{1}{k!}$ is the Taylor series expansion of e. Thus:

$$E[N] = e.$$

2.7 Conclusion

The expected number of draws required for the sum of Uniform$[0,1]$ random variables to exceed 1 is:

$$\boxed{E[N] = e.}$$

$$E[N] = \sum_{n=1}^{\infty} \frac{1}{n!} = e.$$

Thus, averaging the counts over multiple trials provides an estimate of e.

2.8 Results

Estimated value of e: 2.718

2.9 Visualization

A histogram shows the frequency of counts, with a red dashed line marking the mean value.

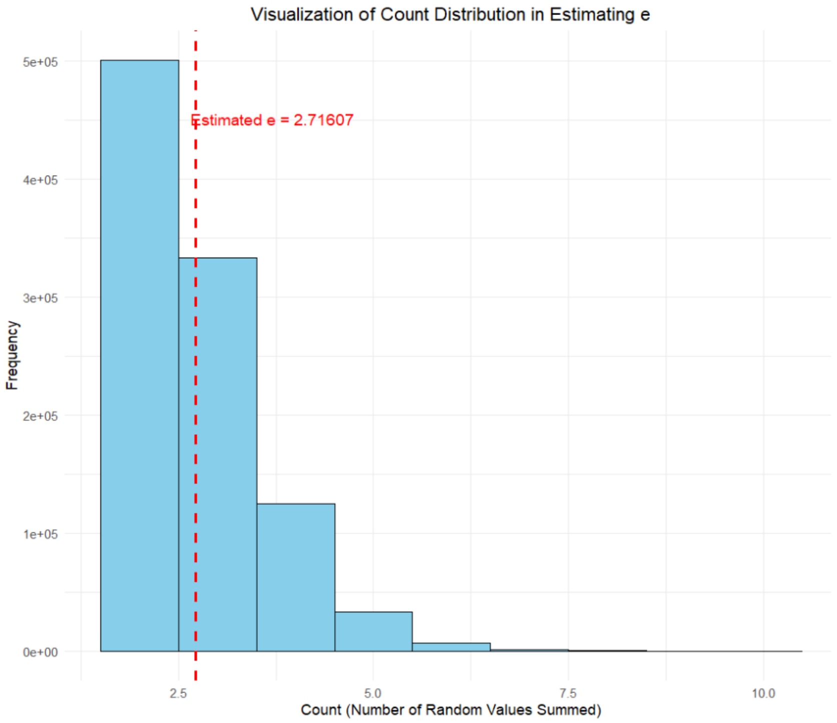

Figure 2: Histogram of Counts for Method 2

3 Method 3: Limit Definition of e

3.1 Description

This method relies on the mathematical definition of e:

$$e = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n.$$

The value of n is progressively increased up to 2^{52}, and the corresponding results are plotted to visualize convergence.

3.2 Mathematical Proof

In Method 3, we estimate the value of e using the formula:

$$e = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n$$

This expression arises from the definition of the exponential function e^x as the limit of a sequence. Below is the detailed proof:

3.3 Definition of the Exponential Function

The exponential function e^x can be defined as:

$$e^x = \lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n$$

For $x = 1$, we have:

$$e = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n$$

This definition is derived from the power series expansion of e^x:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

3.4 Rewriting the Formula Using Logarithms

To analyze the convergence of $\left(1 + \frac{1}{n}\right)^n$, we take the natural logarithm of the expression:

$$\ln\left(\left(1 + \frac{1}{n}\right)^n\right) = n \ln\left(1 + \frac{1}{n}\right)$$

For large n, the term $\ln\left(1 + \frac{1}{n}\right)$can be approximated using the Taylor expansion of $\ln(1 + x)$ around $x = 0$:

$$\ln(1 + x) \approx x - \frac{x^2}{2} + \frac{x^3}{3} - \dots$$

Substituting $x = \frac{1}{n}$:

$$\ln\left(1 + \frac{1}{n}\right) \approx \frac{1}{n} - \frac{1}{2n^2} + \frac{1}{3n^3} - \dots$$

Multiplying by n:

$$n \ln\left(1 + \frac{1}{n}\right) \approx 1 - \frac{1}{2n} + \frac{1}{3n^2} - \dots$$

As $n \to \infty$, the higher-order terms $(\frac{1}{n^2}, \frac{1}{n^3}, \dots)$ vanish, leaving:

$$n \ln\left(1 + \frac{1}{n}\right) \to 1$$

Exponentiating both sides:

$$\left(1 + \frac{1}{n}\right)^n \to e$$

3.5 Numerical Approximation

For finite n, the value of $\left(1+\frac{1}{n}\right)^n$ approaches e but with diminishing accuracy as $n$becomes smaller. By choosing a very large n, the value converges to e.

3.6 Conclusion

The formula $e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$ is mathematically valid and provides an accurate approximation for e. It is derived from the definition of the exponential function and the properties of logarithms, and it converges to e as $n \to \infty$. For practical purposes, large values of n yield highly accurate estimates. Interestingly, this method of estimation has an indirect connection to order statistics, as the structure of cumulative sums or geometric properties of uniform distributions often plays a role in probabilistic interpretations of e. Specifically, the connection arises in scenarios where summing independent and identically distributed random variables (as seen in the analysis of uniform order statistics) mirrors the asymptotic behavior underlying the definition of e.

3.7 Results

Estimated value of e: 2.718281828459045 (Exact convergence with high n).

3.8 Visualization

A plot illustrates the convergence of $\left(1 + \frac{1}{n}\right)^n$ towards e as n increases.

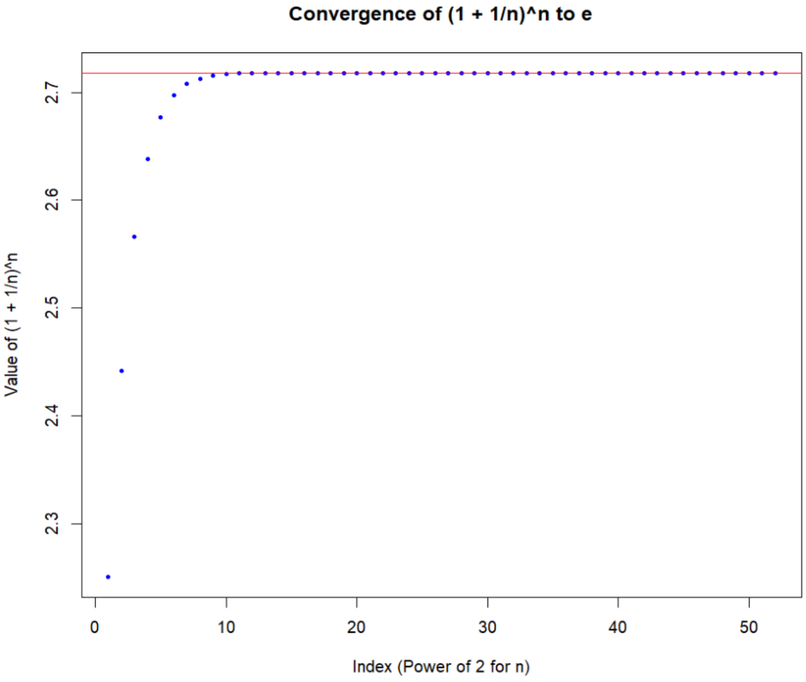

Figure 3: Convergence of $\left(1 + \frac{1}{n}\right)^n$ towards e

4 Method 4: Bernoulli Trials

4.1 Description

Using Bernoulli trials, e is estimated as follows:

1. Perform Bernoulli trials where each event has a probability of success $\frac{1}{n}$.
2. Here n = the number of times we have simulated whether zero wins or not
3. Count the proportion of zero successes (no wins).
4. Compute e using the formula:

$$e = \frac{n}{\text{proportion of zero wins}}.$$

5. Repeat the experiment multiple times for accuracy.

4.2 Mathematical Proof

For a Bernoulli trial with success probability $\frac{1}{n}$, the probability of zero successes in n trials is:

$$P(\text{zero successes}) = \left(1 - \frac{1}{n}\right)^n.$$

As $n \to \infty$, this probability converges to $\frac{1}{e}$. Taking the reciprocal of this probability gives an estimate for e.

4.3 Results

Estimated value of e: 2.718

5 Method 5: Estimation of e Using Derangements

5.1 Description

This method estimates e using the concept of derangements. A derangement is a permutation of a set where no element appears in its original position. By simulating permutations and calculating the proportion of derangements, we can estimate e.

The steps are as follows: 1. Generate random permutations of a set with n elements. 2. Check if each permutation is a derangement. 3. Count the number of derangements, derangement_count. 4. Estimate the probability of a derangement:

$$P(\text{derangement}) \approx \frac{\text{derangement\_count}}{\text{simulations}}.$$

5. Use the reciprocal of this probability to estimate e:

$$e \approx \frac{\text{simulations}}{\text{derangement\_count}}.$$

5.2 Theoretical Proof

The probability of a random permutation being a derangement is:

$$P(\text{derangement}) = \frac{D_n}{n!},$$

where D_n is the number of derangements of n elements, calculated as:

$$D_n = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}.$$

As $n \to \infty$, this probability converges to:

$$P(\text{derangement}) \to \frac{1}{e}.$$

Thus, e can be estimated as:

$$e \approx \frac{1}{P(\text{derangement})}.$$

5.3 Conclusion

This method effectively estimates e using probabilistic principles and the concept of derangements. It provides a novel approach that combines combinatorial mathematics and Monte Carlo simulation.

Analysing the Rate of Convergence

The convergence analysis shows the rate of convergence of each method, highlighting their computational efficiency.Here delta is the rate of convergence.

Method	**Delta**
Method 1	154.2
Method 2	154.2
Method 4	154.2
Method 5	152.5

Table 1: Benchmark Results for Computational Efficiency

6 Conclusion

Method 1 , method 2 , and method 4 are giving same value of delta of 154.2 for n=100 whereas method 5 is giving a delta of 152.5 ; hence we can conclude that the methods 1,2,4 have similar rate of convergence whereas the method 5 has lower rate of convergence than them .

These approaches demonstrate the versatility of computational methods in estimating mathematical constants.

QUESTION 2

Estimation of π Using Various Methods

1 Introduction

In this report, we explore various methods for estimating the mathematical constant π. These methods include 2-dimensional, 3-dimensional, and n-dimensional Monte Carlo approaches, as well as an approach based on generating random angles. Each method involves simulating points in a geometric space and calculating the proportion of points that fall within a specified region. Additionally, we examine the performance and accuracy of these methods as dimensionality increases.

2 Method 1: 3-Dimensional Monte Carlo Method

This method extends the 2D Monte Carlo approach to three dimensions. Random points are generated within a cube of side length 2, and we count how many fall inside the unit sphere inscribed within the cube. The ratio of points inside the sphere to the total number of points provides an estimate of π.

3 Method 2: n-Dimensional Monte Carlo Method (Even Dimensions)

For even values of n, we generate random points in higher-dimensional spaces and count how many fall within the unit hypersphere. The estimated value of π is derived from the proportion of points inside the hypersphere to the total generated points.

4 Method 3: n-Dimensional Monte Carlo Method (Odd Dimensions)

This method extends the Monte Carlo approach to odd-dimensional spaces. The procedure remains similar to the even-dimensional case, but the scaling formula used to estimate π varies due to the odd dimensionality characteristics.

5 Method 4: Zero Memory Monte Carlo Method

In this method, we avoid storing the positions of generated points. Instead, we compute the distance of each point from the origin on the fly and update the count of points inside the unit sphere. This approach uses constant memory but is computationally slower.

5.1 Need for Constant Memory Approach

The constant memory approach is crucial when handling large datasets, where storing coordinates of all generated points would consume excessive memory. By processing points sequentially without retention, this method significantly reduces memory consumption, making it practical for high-dimensional simulations.

6 Method 5: Optimization with Rcpp

To accelerate the Zero Memory Monte Carlo Method, we optimize the computations using the Rcpp package. This enables the core code to be compiled into C++, significantly reducing execution time. However, despite these optimizations, the method still demands considerable computation time when processing a large number of points.

7 Method 6: Random Angle Estimation of π

This method estimates π by generating random angles between 0 and $\pi/2$ and computing the mean of their cosines. Using this approach, we can approximate π without directly simulating points in geometric space, offering an alternative perspective on the estimation process.

8 Conclusion

We have explored various methods for estimating the value of π, ranging from simple 2D Monte Carlo methods to more complex higher-dimensional and optimization techniques. The accuracy of these estimates improves with an increasing number of random points; however, computational constraints limit precision, especially in higher dimensions. The random angle method provides a novel way to estimate π without relying on geometric simulations, offering computational efficiency in certain scenarios.

QUESTION 3

Randomness Testing of Generated Data

1 Introduction

In this report, we investigate the randomness of a dataset generated from a normal distribution using various statistical tests. These tests include the Runs Test, Autocorrelation Test, Ljung-Box Test, Kolmogorov-Smirnov Test, and two custom implementations of the Runs Test. The goal is to assess whether the data exhibits randomness, as expected from a normal distribution.

2 Generated Data

We begin by generating 100 random data points from a normal distribution with mean 0 and standard deviation 1. We then summarize the data and visualize it using a histogram and a density plot.

3 Method 1: Runs Test

The Runs Test assesses whether the sequence of data points exhibits randomness by checking the number of runs (a sequence of consecutive increasing or decreasing values). In our case, the p-value is greater than 0.05, so we fail to reject the null hypothesis and conclude that the data appears random.

4 Method 2: Autocorrelation Test

We perform an autocorrelation test by calculating the autocorrelation function (ACF) of the data. The ACF plot shows that all values of the ACF, except the first lag, are within the confidence levels (-0.2, 0.2), indicating that the data is random.

5 Method 3: Ljung-Box Test

The Ljung-Box Test is applied to check for autocorrelation in the data. The p-value is greater than 0.05, so we fail to reject the null hypothesis, confirming that the data exhibits randomness.

6 Method 4: Kolmogorov-Smirnov Test

We apply the Kolmogorov-Smirnov (KS) Test to compare the data with a standard normal distribution. The p-value is greater than 0.05, so we fail to reject the null hypothesis, confirming that the data is consistent with a normal distribution and is random.

7 Method 5: Custom Runs Test (Manual Implementation)

We implement the Runs Test manually by calculating the expected number of runs and the variance, and then computing the z-score and p-value. The result indicates that the data is random, as the p-value is greater than 0.05.

8 Method 6: Recursive Runs Test (Manual Implementation)

A recursive approach to the Runs Test is implemented, which analyzes the data at multiple levels by recursively dividing it into subsets. The recursive analysis shows that the data appears random at each level, as the p-values remain greater than 0.05.

9 Conclusion

Based on all the tests conducted, we can confidently conclude that the generated data is random. The Runs Test, Autocorrelation Test, Ljung-Box Test, Kolmogorov-Smirnov Test, and the custom manual and recursive Runs Tests all support the hypothesis that the data is random.