Report on Lab 1

Author: Kevin Deshayes

Email: kede23@student.bth.se

Course Name: Mathematical Statistics

Course Code: MS1403

Contents

0.1 R code

```r
#Assignment 1 MS1403
#Kevin Deshayes
#kede23@student.bth.se
# Task 1: Implement three estimators and run experiments for different scenarios
n_values = c(10,20,50,90,140) #Global vector
# --- Define Estimators ---
# Estimator N1: Returns the maximum element in the sample
firstEstimator = function(n_values){
  #Will return the biggest element of the sample values
  return(max(n_values));
}
# Estimator N2: (n + 1) / n * Max(Sample)
secondEstimator = function(n_values){
  #do the (n+1/n)*firstEstimator(n_values)
    n=length(n_values)
    x_n = max(n_values)
    N_2 = ( (n+1) / n) * x_n
    return (N_2)
}
#For estimator N3
thirdEstimator = function(n_values){
    X_mean = mean(n_values)
    N_3 = 2*X_mean-1
    return(N_3)
  }
# --- Main Experiment Function ---
Experiment = function(N, n_values, repetitions = 5){
  ResultMean = data.frame(
    sample_size = n_values,
    Mean_Estimation1 = numeric(length(n_values)),  # Use lists to store multiple repetitions
    Mean_Estimation2 = numeric(length(n_values)),
    Mean_Estimation3 = numeric(length(n_values))
  )
  ResultVariance = data.frame(
    sample_size = n_values,
    Variance_Estimation1 = numeric(length(n_values)),
    Variance_Estimation2 = numeric(length(n_values)),
    Variance_Estimation3 = numeric(length(n_values))
  )
  #Iterate for 5 times
```

```r
  for (n in n_values){
    #Store the estimations
    N1_reps = numeric(repetitions)
    N2_reps = numeric(repetitions)
    N3_reps = numeric(repetitions)
    for (i in 1:repetitions){
      #generate the result of the draws, size nrOF draws, results range from 1 to N
      #And stores them in the vector drawResult
      drawResult = sample(1:N, size = n, replace = TRUE, prob = NULL)
      #Runs estimations using the drawResult and stores them in respective N_x container
      N1_reps[i] = firstEstimator(drawResult)
      N2_reps[i] = secondEstimator(drawResult)
      N3_reps[i] = thirdEstimator(drawResult)
    }
    # Find the row index that matches the current sample size
    row_index = which(ResultMean$sample_size == n)
    # Adds the mean estimation and variance for this n to results and rounds them to 5 digits, places it in the column for the current n
    ResultMean$Mean_Estimation1[row_index] = round(mean(N1_reps), digits = 5)
    ResultVariance$Variance_Estimation1[row_index] = round(var(N1_reps), digits = 5)
    ResultMean$Mean_Estimation2[row_index] = round(mean(N2_reps), digits = 5)
    ResultVariance$Variance_Estimation2[row_index] = round(var(N2_reps), digits = 5)
    ResultMean$Mean_Estimation3[row_index] = round(mean(N3_reps), digits = 5)
    ResultVariance$Variance_Estimation3[row_index] = round(var(N3_reps), digits = 5)
  }
  #Combines the two data frames into a list of two data frames and returns that.
    return(list(Means = ResultMean, Variances = ResultVariance))
}
# call the experiment function twice once for each scenario (Ns)
smallScenario = Experiment(30,n_values)
largeScenario = Experiment(150,n_values)
cat("For small scenario N=30", "\n")
```

```r
cat("------------------------------", "\n")
print(smallScenario$Means)
print(smallScenario$Variances)
cat("\n\n")
cat("For large scenario N=150", "\n")
cat("------------------------------", "\n")
print(largeScenario$Means)
print(largeScenario$Variances)
```

0.2 Results

```
> source("Assigment 1 [Lab].R")
For small scenario N=30
------------------------------
  sample_size Mean_Estimation1 Mean_Estimation2 Mean_Estimation3
1          10             29.2         32.12000         33.32000
2          20             29.0         30.45000         31.92000
3          50             29.6         30.19200         30.47200
4          90             30.0         30.33333         30.96889
5         140             30.0         30.21429         28.84286
  sample_size Variance_Estimation1 Variance_Estimation2 Variance_Estimation3
1          10                  0.7              0.84700             21.41200
2          20                  1.0              1.10250             20.55700
3          50                  0.3              0.31212             10.32272
4          90                  0.0              0.00000              1.64360
5         140                  0.0              0.00000              1.87990


For large scenario N=150
------------------------------
  sample_size Mean_Estimation1 Mean_Estimation2 Mean_Estimation3
1          10            130.8         143.8800         144.6000
2          20            143.8         150.9900         135.8000
3          50            148.4         151.3680         157.1360
4          90            148.4         150.0489         161.1867
5         140            149.4         150.4671         155.0886
  sample_size Variance_Estimation1 Variance_Estimation2 Variance_Estimation3
1          10                145.2            175.69200            193.70000
2          20                 15.7             17.30925            723.83500
3          50                  1.3              1.35252             37.29168
4          90                  4.8              4.90726             44.32360
5         140                  0.8              0.81147            100.03708
>
```

Figure 1: CLI output after running the code

0.3 Analysis

0.3.1 Discussion

In Figure 1, it is clear that all all estimators generally follow the Law of Large Numbers, where larger sample sizes tend to produce smaller variances, making the estimators more consistent and stable. However, there are still apparent spikes and deviations from this behavior. For example, in the large scenario, $\hat{N}_3$ variance jumps from 193 to 723 before eventually decreasing.

This can be explained by the sensitivity of the estimator to outliers or inherent instability when working with random sampling. The high variance of 723 demonstrates how unstable $\hat{N}_3$ can be when dealing with extreme values in the sample. This result is not an anomaly but highlights the nature of $\hat{N}_3$. The reason is that $\hat{N}_3$ is based on the sample mean, which is highly susceptible to outliers or extreme values, especially for small sample sizes.

To address this, increasing the sample size and the number of repetitions would help stabilize the variance for all estimators, but especially for $\hat{N}_3$.

0.3.2 Observations from the small scenario

$\hat{N}_2$ is most accurate overall with inaccuracy in smaller sample sizes. $\hat{N}_1$ is most accurate for smaller sample sizes and demonstrates higher consistency, with a marginally lower variance.$\hat{N}_3$ is relatively inaccurate with a high variance which leads to low consistency.

0.3.3 Observations from the large scenario

$\hat{N}_2$ is most accurate overall with inaccuracy in smaller sample sizes.$\hat{N}_1$ generally exhibits the lowest variance but has high inaccuracies for smaller sample sizes.$\hat{N}_3$ is relatively inaccurate with a high variance which leads to low consistency.

0.3.4 Conclusions

- $\hat{N}_1$: Best suited for smaller N values. Consistent with low variance.
- $\hat{N}_2$: The most accurate estimator in both small and large scenarios.
- $\hat{N}_3$: Highly unstable, particularly with small sample sizes, due to its reliance on the sample mean, which is prone to large variances from extreme values.
- Law of Large Numbers: Observed in general trends, but temporary spikes can still occur, especially in $\hat{N}_3$ due to sensitivity to random fluctuations.