Líkindareikningur og Tölfræði STÆ203G

Tölvuverkefni 4

Egill Ian Guðmundsson, 260693-2639

1 Verkefni

Gerið ráð fyrir að slembiúrtak $X_1, ..., X_n$ af stærð n sé dregið úr normaldreifingu með meðalgildi $\mu = 96$ og $\sigma = 13$. Stikann σ er hægt að meta með eftirfarandi metlum. Annars vegar með:

$$\hat{\sigma_1} = s = \sqrt{\frac{1}{n-1}\sum_{n}^{i=1}(X_i - \bar{X})^2}$$

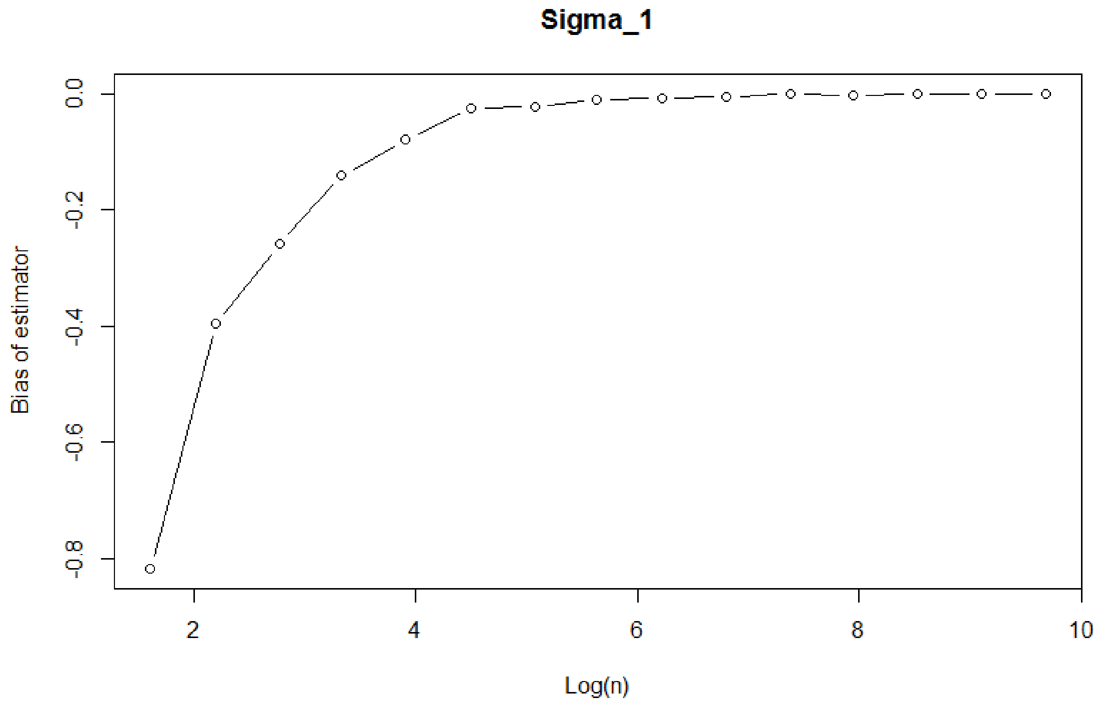
og hins vegar með:

$$\hat{\sigma_2} = \frac{IQR}{1.349}$$

Þar sem IQR er fjórðungsbil úrtaks. Þannig er $\hat{\sigma_1}$ staðalfrávik úrtaks og $\hat{\sigma_2}$ byggir á fjórðungsbili úrtaks.

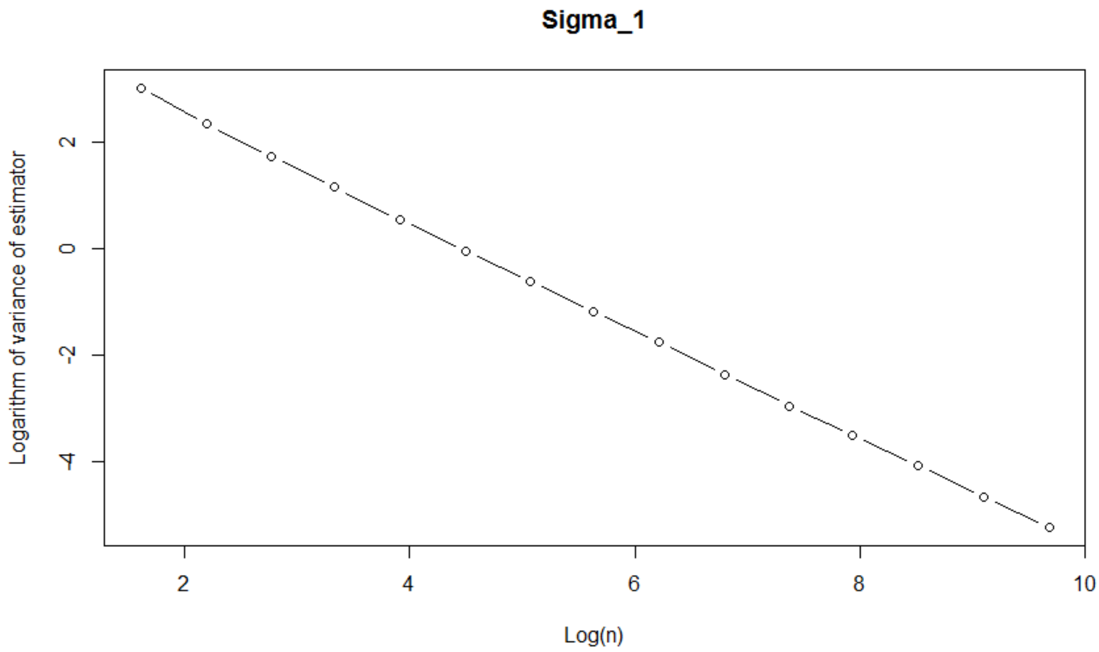
1. Teiknið bjaga metils $\hat{\sigma_1}$ sem fall af n. Notið lograskala fyrir n og notið úrtaksstærðir $n = \{5, 9, 16, 28, 50, 90, 160, 280, 500, 900, 1600, 2800, 5000, 9000, 16000\}$. Fyrir hverja úrtaksstærð n, notið 10000 úrtök til að reikna gildið á $\hat{\sigma_1}$ og reiknið meðaltalið af þessum gildum til að meta væntigildi metilsins.

   Svar: Notum fyrra forritið í viðauka með skipun `estimatorAssignment(nVector, 96, 13, 10000, 1, 1)` til að fá:



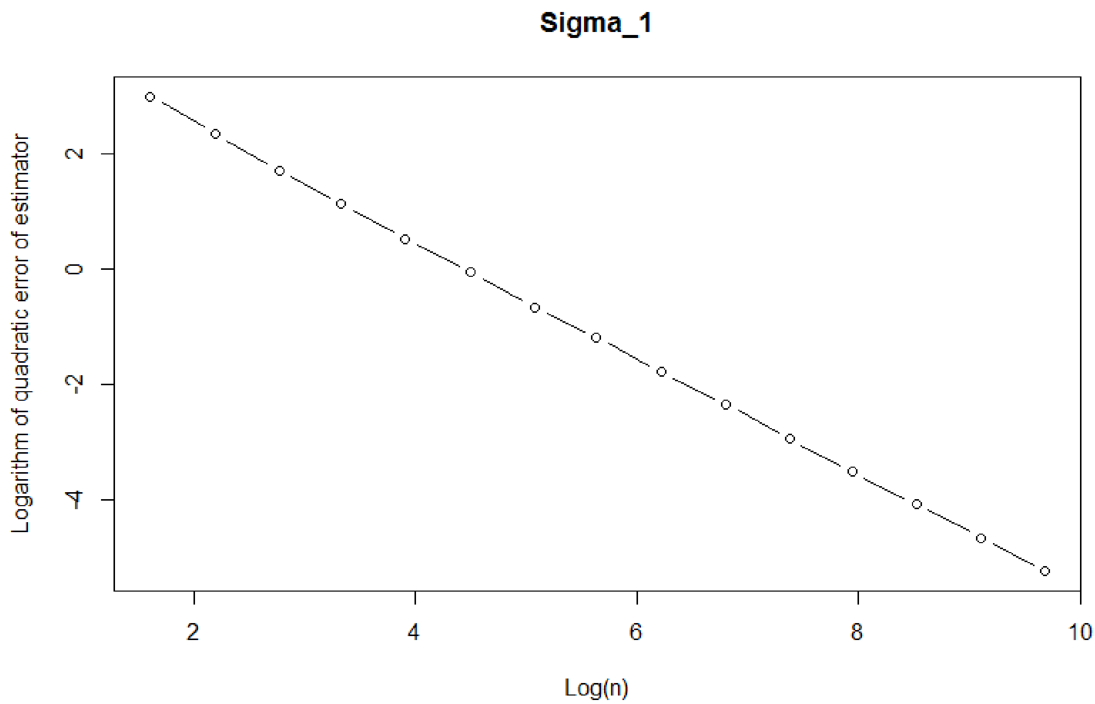
2. Teiknið dreifni metilsins $\hat{\sigma_1}$ sem fall af n. Notið lograskala fyrir bæði n og dreifni metilsins. Notið sömu skilyrði og að ofan og reiknið dreifnina fyrir hvert úrtak til að meta dreifni metilsins.

   Svar: Notum fyrra forrit með `estimatorAssignment(nVector, 96, 13, 10000, 2, 1)` til að fá:



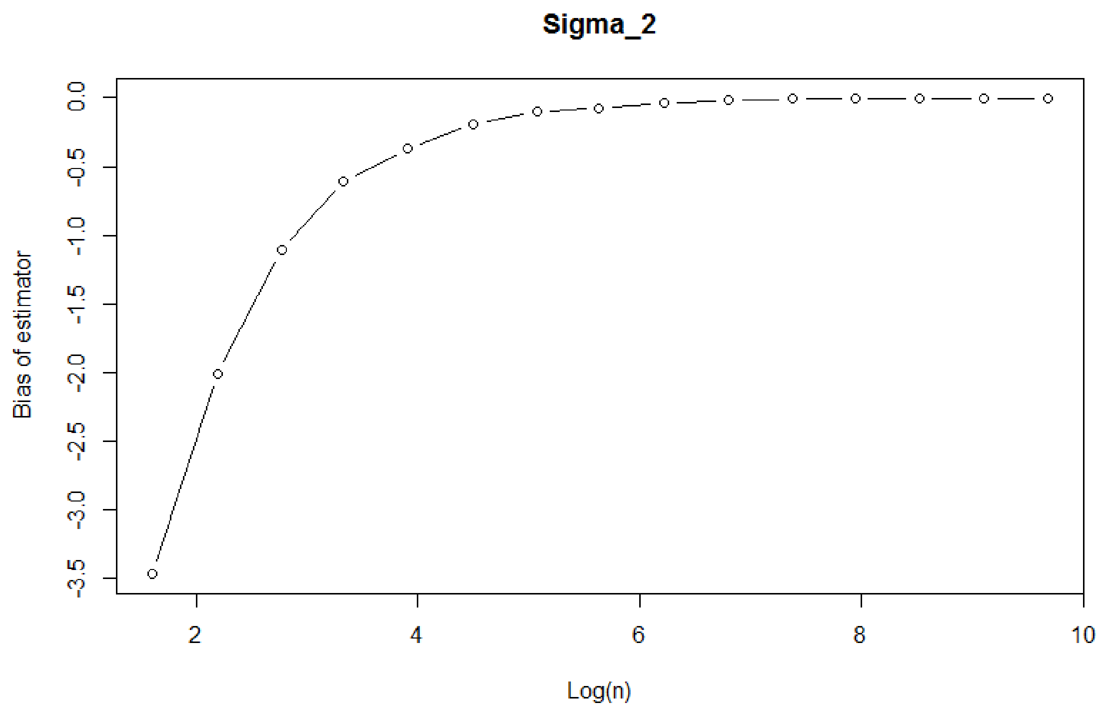
3. Teiknið meðalferskekkju metils $\hat{\sigma_1}$ sem fall af n. Notið lograskala fyrir bæði n og dreifni metilsins. Notið sömu skilyrði og að ofan og reiknið meðalferskekkjuna fyrir hvert úrtak út frá bjaga og dreifni metilsins.

   Svar: Notum fyrra forrit með skipun `estimatorAssignment(nVector, 96, 13, 10000, 3, 1)` til að fá mynd:



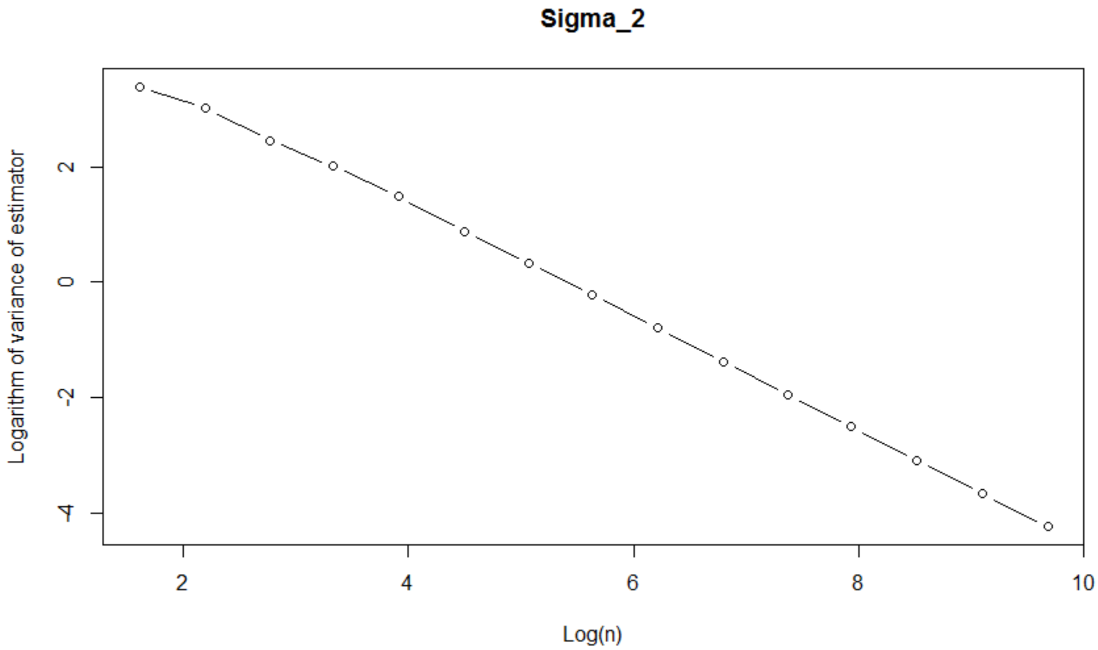
4. Teiknið bjaga metils $\hat{\sigma_2}$ sem fall af n. Notið lograskala fyrir n og notið sömu skilyrði og að ofan.

   Svar: Keyrum fyrra forrit með
   `estimatorAssignment(nVector, 96, 13, 10000, 1, 2)` og fáum:



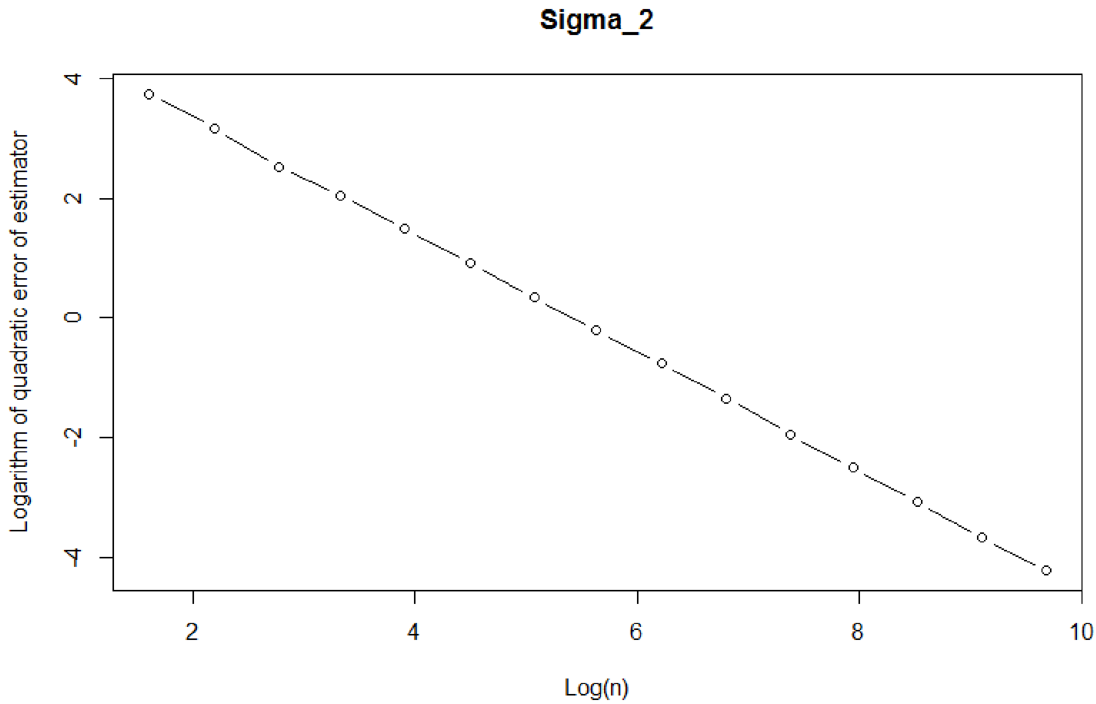
5. Teiknið dreifni metils $\hat{\sigma_2}$ sem fall af n. Notið lograskala fyrir n og dreifnina og notið sömu skilyrði og að ofan.

   Svar: Keyrum fyrra forrit með
   `estimatorAssignment(nVector, 96, 13, 10000, 2, 2)` og fáum:



6. Teiknið meðalferskekkju metils $\hat{\sigma_2}$ sem fall af n. Notið lograskala fyrir bæði n og dreifni metilsins. Notið sömu skilyrði og að ofan og reiknið meðalferskekkjuna fyrir hvert úrtak út frá bjaga og dreifni metilsins.

   Svar: Keyrum fyrra forrit með `estimatorAssignment(nVector, 96, 13, 10000, 3, 2)` og fáum:



7. Byggt á myndunum hér að ofan, hvorn metilinn teljið þið vera betri til að meta staðalfrávik? Útskýrið með hliðsjón af bjaga, dreifni og meðalferskekkju metlanna.

   Eins og sést á gröfunum er dreifni og meðalferskekkja tiltölulega svipuð á aðferðunum tveimur. Hins vegar er bjagi aðferðanna mjög ólíkur. Bjagi $\hat{\sigma_1}$ er mikill í fyrstu en minnkar ágætlega eftir því sem úrtak stækkar og virðist stefna á 0. Bjagi $\hat{\sigma_2}$ lækkar einnig með vaxandi úrtaksstærð en ekki jafn hratt og í tilfelli $\hat{\sigma_1}$. Því er $\hat{\sigma_1}$ heppilegari kostur sem metill.

8. Látum η_p tákna 100p-tu sætisstærð í normaldreifingu. Um η_p gildir $\eta_p = \mu + z_{1-p}\sigma$. Hermið gögn frá ofangreindri normaldreifingu þegar $n = 23$ og 1.000.000 ítrunum. $\eta_{0.31}$ er 31. sætisstærðin í normaldreifingunni. Notið eftirfarandi 95% öryggisbil:

$$(\bar{x} + z_{0.69} \cdot s) \pm z_{0.025} \frac{\sqrt{s^2 + 0.5z^2 + 0.69s^2}}{\sqrt{n}}$$

Metið öryggisstig öryggisbilsins fyrir $\eta_{0.31}$. Inniheldur öryggisbilið rétta gildið á $\eta_{0.31}$ í 95% tilfella? Öryggisbilið hér að ofan byggir á nálgun og ekki gefið að öryggisstig þess sé nákvæmlega 95%

Svar: Notum seinna forritið með skipun
`confidenceIntervalAssignment(23, 96, 13, 1000000)`
til að fá niðurstöðu. Það var óljóst hvort verið var að tala um staðalfrávik eða metil $\hat{\sigma_1} = s$ svo reiknað var með báðum stærðum. Ef reiknað var með staðalfráviki með skipun `sd()` í R fékkst að öryggisstig öryggisbilsins er 93.5779% og því inniheldur það ekki rétta gildi $\eta_{0.31}$ í 95% tilfella. Sé notast við formúlu fyrir s eins og kemur fyrir fremst í verkefninu er öryggisstig bilsins 95.4881 og inniheldur rétt gildi á $\eta_{0.31}$ í 95% tilfella.

2 Viðauki

```
# plotType = 1 plots estimator bias. plotType = 2 plots variance of estimators.
# plotType = 3 plots quadratic error of estimators.
# estimatorType = 1 uses sigma_1 method (sum of quadratic error). estimatorType = 2 uses
# sigma_2 method (IQR method)
estimatorAssignment <- function(sampleSizes, meanValue, stdDev, iterations, plotType, estimatorType){

  # Vector keeping the average bias of each estimator by sample size
  estimatorBiasVector = c(0)
  # Vector keeping the average variance of each estimator by sample size
  estimatorVarianceVector = c(0)
  # Vector keeping the average quadratic error of each estimator by sample size
  estimatorQuadraticVector = c(0)

  for(s in 1:length(sampleSizes)){
    # Vector containing every estimator for given sample size
    estimatorSum = c(0)
    # Creates estimator for each iteration
    for(j in 1:iterations){
      sampleValues = rnorm(sampleSizes[s], mean = meanValue, sd = stdDev)
      sampleMean = mean(sampleValues)
      sampleSum = 0
      # Calculation of quadratic error inside estimator formula (sigma_1)
      for(x in 1:sampleSizes[s]){
        sampleSum = sampleSum + (sampleValues[x] - sampleMean)^2
      }
      if(estimatorType == 1){
        # Calculation of estimator for given iteration
        estimatorSum[j] = sqrt((1/(sampleSizes[s] - 1)) * sampleSum)
      }
      else{
        # Calculation of estimator (sigma_2)
        estimatorSum[j] = IQR(sampleValues)/1.349
      }
    }
    estimatorBiasVector[s] = mean(estimatorSum) - stdDev
    estimatorVarianceVector[s] = var(estimatorSum)
    estimatorQuadraticVector[s] = estimatorVarianceVector[s] + estimatorBiasVector[s]^2
  }

  # Variables for plotting graph
  lnValues = log(sampleSizes)
  mainTitle = "Sigma_1"
  if(estimatorType == 2){
```

```
    mainTitle = "Sigma_2"
  }

  # Plotting of bias of estimators
  if(plotType == 1){
    plot(lnValues, estimatorBiasVector, type = "b", main = mainTitle, xlab = "Log(n)", ylab = "Bias of estimator")
  }

  # Plotting of variance of estimators
  if(plotType == 2){
    lnVariances = log(estimatorVarianceVector)
    plot(lnValues, lnVariances, type = "b", main = mainTitle, xlab = "Log(n)",
         ylab = "Logarithm of variance of estimator")
  }

  # Plotting of quadratic error of estimators
  if(plotType == 3){
    lnQuad = log(estimatorQuadraticVector)
    plot(lnValues, lnQuad, type = "b", main = mainTitle, xlab = "Log(n)",
        ylab = "Logarithm of quadratic error of estimator")
  }
}
```

estimatorAssignment.r

```
confidenceIntervalAssignment <- function(sampleSize, meanValue, stdDev, iterations){

  # 31. quantile for normal distribution
  quantileValue = meanValue + qnorm(1-0.31) * stdDev
  # Counter keeps track of how many samples are within safety interval
  counter = 0

  for(i in 1:iterations){
    # Generate random normal values with parameters
    normalValues = rnorm(sampleSize, meanValue, stdDev)
    # Quadratic error for given sample
    quadError = 0
    for(s in 1:sampleSize){
      quadError = quadError + (normalValues[s] - meanValue)^2
    }
    # Calculate estimator for given sample
    #estimator = sqrt((1/(sampleSize - 1)) * quadError)
     estimator = sd(normalValues)
    # Calculate confidence interval for given sample
    middleValue = mean(normalValues) + qnorm(0.69) * estimator
    shiftValue = qnorm(0.025)*(sqrt(estimator^2 + 0.5*qnorm(0.69)^2*estimator^2)/sqrt(sampleSize))
    confidenceUpper = middleValue + abs(shiftValue)
    confidenceLower = middleValue - abs(shiftValue)
```

```
    # Check if confidence interval contains quantile value
    if(confidenceLower <= quantileValue && quantileValue <=
       confidenceUpper){
      counter = counter + 1
    }
  }

  message((counter/iterations)*100)
}
```

confidenceIntervalAssignment.r