Statistics 421: *Estimation of mean and standard deviation of normal distribution via QQ methodology*

Jina Ryu, Feng Yuan

December 20, 2024

Contents

A Introduction

The estimation of the mean μ and standard deviation σ of a normal distribution is a fundamental problem in statistics. There are methods such as moments and maximum likelihood estimation to provide efficient estimators for these parameters. However, in this project, we'll discuss the quantile - quantile QQ approach method. QQ methodology offer insights and diagnostic capabilities by exploiting the relationship between the observed sample order statistics and theoretical quantiles of the assumed distribution. It is particularly useful in validating the assumption of normality and provides novel estimators for μ and σ.

In this project, QQ methodology utilizes the linear relationship between the quantiles of the normal parent distribution $N(\mu, \sigma^2)$ and its standard counterpart $N(0, 1)$. For the 100 x p-th quantiles of $N(\mu, \sigma^2), x_p$, the relationship is expressed as:

$$x_p = \mu + \sigma z_p$$

where z_p is the corresponding quantile of $N(0, 1)$. This linear structure allows to evaluate normality by plotting the sample order statistics $Y_1 \leq \cdots \leq Y_n$ against the Hazen scores $z_{(j-0.5)/n}$, which are approximations of the corresponding theoretical quantiles. When parent distribution is normal, the QQ plot exhibits a linear trend with a positive slope.

When fitting an ordinary least squares (OLS) regression line to the QQ plot, we obtain the estimators for μ and σ:

$$\mu_{QQ} = \overline{Y} - \sigma_{QQ}\overline{z}, \sigma_{QQ} = \frac{\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})(Y_j - \overline{Y})}{\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})^2}$$

,

where $\overline{Y}$ and $\overline{z}$ are the means of the order statistics and the Hazen scores, respectively. The corresponding estimator for the variance is $\sigma^2_{QQ} = \sigma^2_{QQ}$.

This study investigates the finite sample properties of μ_{QQ}, σ_{QQ} and σ^2_{QQ} through the Monte Carlo simulations, comparing the performance against the minimum variance unbiased estimators (MVUEs):

$$\mu_{MVUE} = \overline{X}, \quad \sigma_{MVUE} = C_n S, \quad \sigma^2_{MVUE} = S^2,$$

where S^2 is the unbiased sample variance, and C_n is a correction factor derived from the gamma function. The analysis will assess the biases and variances of these estimators, providing insights into the effectiveness of the QQ-based approach for parameter estimation in normal distribution.

B Dependence of variables

In this section, we are going to simulate R= 10000 random samples from the Gaussian parent, with n= 5, 10, 40, 50, 75, and 100, using the R function **rnorm()**. R-code used for this section can be found under appendix.

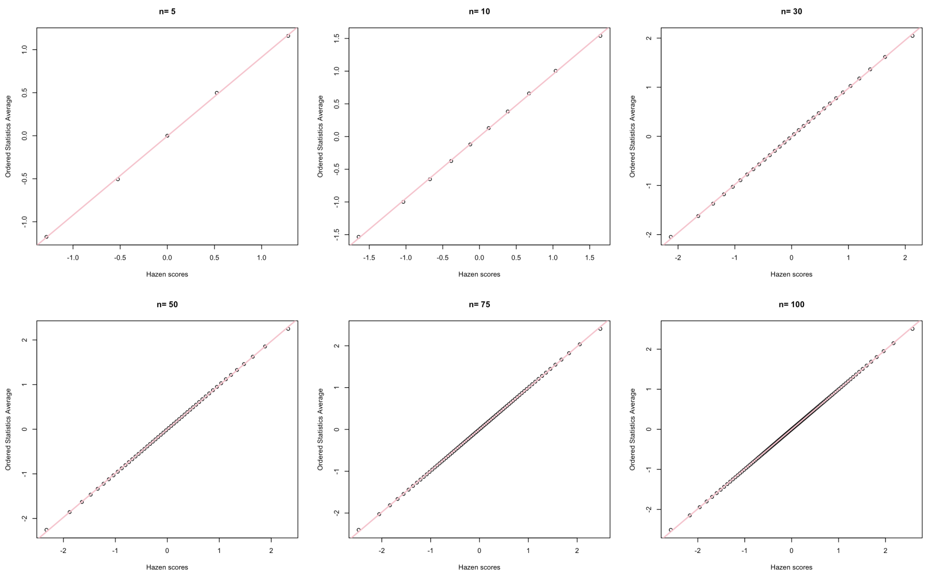

Figure 1: Scatter plot created at each values of n, given x-axes are values of Hazen scores, and y-axes are average value of ordered statistics.

Now, let's discuss about the **"average" effectiveness** of the QQ- plot in validating the normality assumption for a parent. As n increases, the average QQ-plot becomes more effective in validating normality, as the variability in the order statistics diminishes. Larger sample sizes reduce this variability, enhancing the precision of the estimated order statistics. Consequently, the scatter points align more closely with the regression line for larger n. Conversely, for smaller n, the greater variability in the order statistics reduces the validation's reliability, though the linear trend remains evident.

C Estimator and Unbiasness

Let's take a look at the estimator $\widehat{\mu}_{QQ}$ and $\widehat{\sigma}_{QQ}$ and show that they are approximately unbiased for μ and σ. That is, show that

$$E_\theta(\widehat{\mu}_{QQ}) \approx \mu, \quad \forall \mu \in \mathbb{R},$$

$$E_\theta(\widehat{\sigma}_{QQ}) \approx \sigma, \quad \forall \sigma > 0.$$

First, we rewrite σ_{QQ} as a linear combination of $Y_1, ..., Y_n$.

Given,

$$\sigma_{QQ} = \frac{\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})(Y_j - \overline{Y})}{\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})^2}$$

where

$$\overline{z} = \frac{1}{n}\sum_{j=1}^{n} z_{(j-0.5)/n} \quad \text{is the mean of the Hazen scores, and}$$

$$\overline{Y} = \frac{1}{n}\sum_{j=1}^{n} Y_j \quad \text{is the mean of the order statistics.}$$

Let,

$$a_j = \frac{z_{(j-0.5)/n} - \overline{z}}{\sum_{j=1}^{n}(z_{(j'-0.5)/n} - \overline{z})^2}, \quad j = 1, ..., n.$$

We want to rewrite σ_{QQ} as $\sigma_{QQ} = \sum_{j=1}^{n} a_j Y_j$.

Second step, we need to verify the properties of the coefficients a_j. Show $\sum_{j=1}^{n} a_j = 0$:

$$\sum_{j=1}^{n} a_j = \frac{\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})}{\sum_{j'=1}^{n}(z_{(j'-0.5)/n} - \overline{z})^2}$$

Recall,

$$\overline{z} = \frac{1}{n}\sum_{j=1}^{n} z_{(j-0.5)/n}$$

Then it follows that:

$$n\overline{z} = \sum_{j=1}^{n} z_{(j-0.5)/n}$$

We then got,

$$\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z}) = \sum_{j=1}^{n} z_{(j-0.5)/n} - n\overline{z} = n\overline{z} - n\overline{z} = 0,$$

the numerator is zero. Thus,

$$\sum_{j=1}^{n} a_j = 0.$$

Now, show $\sum_{j=1}^{n} a_j z_{(j-0.5)/n} = 1$:

$$\sum_{j=1}^{n} a_j z_{(j-0.5)/n} = \frac{\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})z_{(j-0.5)/n}}{\sum_{j'=1}^{n}(z_{(j'-0.5)/n} - \overline{z})^2}.$$

Lets expand the numerator,

$$\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})z_{(j-0.5)/n} = \sum_{j=1}^{n} z^2_{(j-0.5)/n} - \overline{z}\sum_{j=1}^{n} z_{(j-0.5)/n}.$$

From here, $\sum_{j=1}^{n} z_{(j-0.5)/n} = n\overline{z}$, so we can simplify the numerator as,

$$\sum_{j=1}^{n} z^2_{(j-0.5)/n} - n\overline{z}^2.$$

Since the dominator is,

$$\sum_{j'=1}^{n}(z^2_{(j'-0.5)/n} - \overline{z})^2 = \sum_{j=1}^{n} z^2_{(j-0.5)/n} - n\overline{z}^2,$$

thus,

$$\sum_{j=1}^{n} a_j z_{(j-0.5)/n} = 1.$$

Now, show $\sum_{j=1}^{n} a_j^2 = \frac{1}{\sum_{j=1}^{n}(z_{(j-0.5)/n}-\overline{z})^2}$: Using the definition,

$$a_j^2 = \frac{z_{(j-0.5)/n} - \overline{z}}{(\sum_{j=1}^{n}(z_{(j'-0.5)/n} - \overline{z})^2)^2}$$

Let's sum over j:

$$\sum_{j=1}^{n} a_j^2 = \frac{\sum_{j=1}^{n}(z_{(j=1)/n} - \overline{z})^2}{(\sum_{j'=1}^{n}(z_{(j'-0.5)/n} - \overline{z})^2)^2}$$

Thus,

$$\sum_{j=1}^{n} a_j^2 = \frac{1}{\sum_{j=1}^{n}(z_{(j-0.5)/n} - \overline{z})^2}.$$

Now, let's approximate Unbiasness of μ_{QQ} and σ_{QQ} The estimator for μ_{QQ} is $\mu_{QQ} = \overline{Y} - \sigma_{QQ}\overline{z}$. To calculate the bias-ness, we need to take the expectation of the estimator.

Let's rewrite $\sigma_{QQ} = \sum_{j=1}^{n} a_j Y_j$, and take the expectation of the estimator:

$$E[\sigma_{QQ}] = \sum_{j=1}^{n} a_j E[Y_j]$$

Using the approximation $E[Y_j] \approx \mu + \sigma z_{(j-0.5)/n}$,

$$E[\sigma_{QQ}] = \sum_{j=1}^{n} a_j(\mu + \sigma z_{(j-0.5)/n}).$$

Now,

$$E[\sigma_{QQ}] = \mu \sum_{j=1}^{n} a_j + \sigma \sum_{j=1}^{n} a_j z_{(j-0.5)/n} \quad \text{where} \quad \sum_{j=1}^{n} a_j = 0, \quad \text{and} \quad \sum_{j=1}^{n} a_j z_{(j-0.5)/n} = 1.$$

Thus,

$$E[\sigma_{QQ}] \approx \sigma.$$

Now, let's take expectation of the μ:

$$E[\mu_{QQ}] = E[\overline{Y}] - E[\sigma_{QQ}]\overline{z}.$$

Using the approximation $E[Y_j] \approx \mu + \sigma z_{(j-0.5)/n}$, now we have,

$$E[\overline{Y}] \approx \mu + \sigma\overline{z}$$

.

Let's substitute into the expression for $E[\mu_{QQ}]$:

$$E[\mu_{QQ}] \approx \mu + \sigma\overline{z} - E[\sigma_{QQ}]\overline{z}.$$

Since $E[\sigma_{QQ}] = \sigma$, thus,

$$E[\mu_{QQ}] \approx \mu.$$

D Monte Carlo Simulations

To start, we first begin with Monte Carlo simulations of Y by generating 10^4 random samples of $Y_1, \ldots, Y_n \sim N(0,1)$ to see how different sample sizes $n = 5, 10, 30, 50, 75, 100$ effects the estimation of the mean μ and standard deviation σ^2. Although we can visually see how they may be effected using the QQ-methodology via QQ-plots, to be more rigorous we check the biases of the estimators. This will allow us to see the effects that sample size has on μ and σ^2, then by plotting the bias of μ, σ, and σ^2 on a line plot to see how far they vary from 0, this allows us to visually check how much sample size effects our estimators.

Table 2 is a table of values of $B_n(\hat{\theta}, \theta)$ for the 3 estimators and for each sample size of $n = 5, 10, 30, 50, 75, 100$

n	Biasμ_{QQ}	Biasσ_{QQ}	Biasσ^2_{QQ}
5	0.09525756	9.104384	6.123619
10	0.42896073	5.194997	4.821353
30	0.04855181	2.029333	2.358443
50	0.07991231	1.288090	1.546149
75	0.01762104	1.059635	1.444096
100	0.03232369	0.875246	1.236728

Table 1: Bias of estimators at n

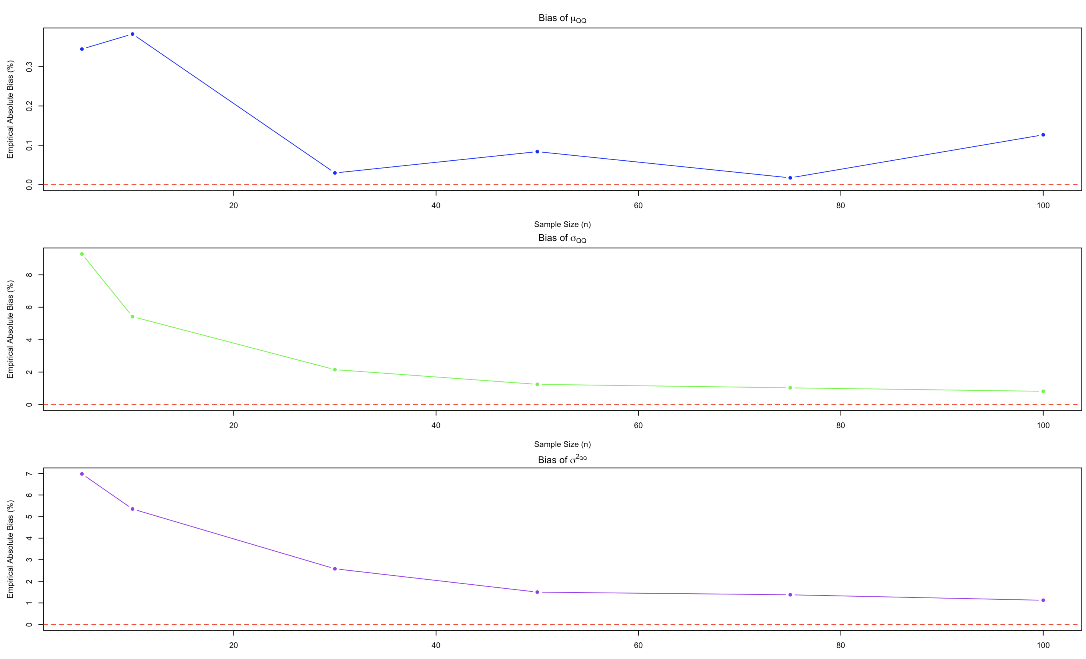

Figure 2: Bias $B_n(\widehat{\theta}, \theta)$ against sample size n for the estimators $\mu_{QQ}, \sigma_{QQ},$ and σ^2_{QQ}

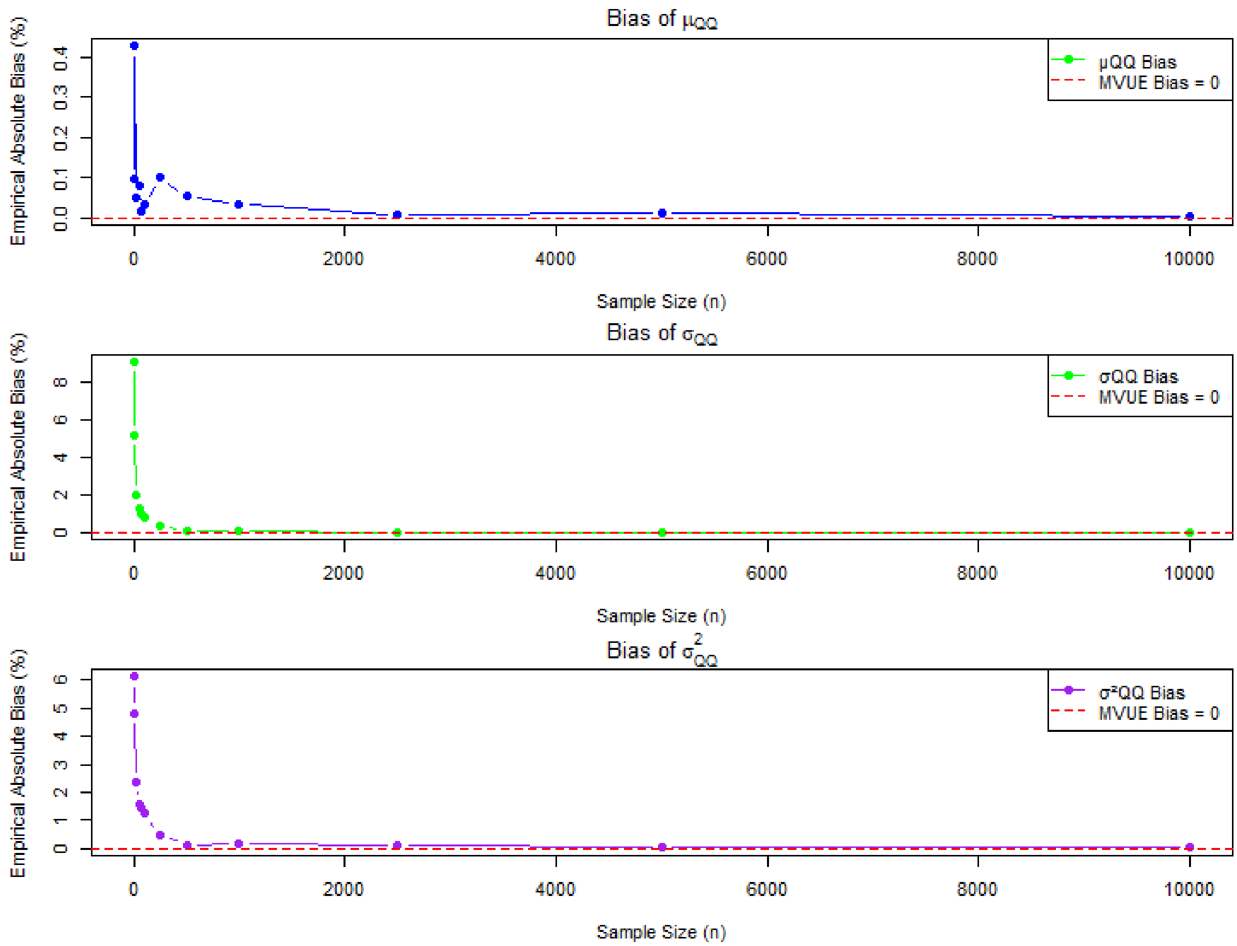

Figure 3: Expanded Bias $B_n(\widehat{\theta}, \theta)$ plots with $n = 250, 500, 1000, 2500, 5000, 10000$

Each plot includes a horizontal line at 0 for the 0-bias of the MVUE.
As we can see from **Figure 2** in general, when the sample size n increases, we can see that the bias for each estimator approaches 0. This means that our Monte Carlo simulations shows that the bias of the estimators are reduced as we use larger sample sizes. To address the anomaly seen in the bias of μ_{QQ} where at $n = 100$ the bias increases, we continued our simulation with even larger samples and in **Figure 3** found that it was indeed an outlier like at $n = 10, 50$. As the sample size increases, even the bias of μ_{QQ} approaches 0. It follows that the effectiveness of the QQ-based approach for parameter estimation of a normal distribution performs better at larger sample sizes, where the results confirm the approximate unbiasedness of $\hat{\mu}_{QQ}$, $\hat{\sigma}_{QQ}$, $\hat{\sigma^2}_{QQ}$. R-code used in this section can be found under appendix.

E Variance of MVUEs

Now, let's find the variance of MVUEs. We want to find $Var(\mu_{MVUE})$, $Var(\sigma_{MVUE})$, and $Var(\sigma^2_{MVUE})$. According to Lecture 6, pg.2, MVUE is minimum variance unbiased estimator. The sample mean is the MVUE of the population mean μ for a normal distribution because the expectation of the sample mean is always the population mean.

Therefore, the MVUE of μ is,

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Since $X_i \sim N(\mu, \sigma^2)$, sample mean follows:

$$\overline{X} \sim N\left(\mu, \frac{1}{n}\sigma^2\right)$$

Thus, the variance of $\mu_{MVUE} = \frac{\sigma^2}{n}$

Now, let's find the variance of σ_{MVUE}

The MVUE of σ is $\hat{\sigma} = \sqrt{\frac{S^2}{n-1}}$, where S^2 is the sample variance,

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

Recall from Student's theorem (Lecture 2, Proof od Student's Theorem)

$$U = \frac{(n-1)S^2}{\sigma^2} \sim X^2_{n-1}.$$

Now, this is equivalent to:

$$S^2 \sim \text{Gamma}\left(\alpha = \frac{n-1}{2}, \beta = \frac{2\sigma^2}{n-1}\right)$$

For $\hat{\sigma} = \sqrt{S^2} = S$, the distribution is Nakagami:

$$S \sim \text{Nakagami}(m = \frac{n-1}{2}, \Omega = \sigma^2)$$

From the properties of the Nakagami distribution (Nakagami Distribution, Wikipedia),

$$Var_{Nakagami} = \Omega\left(1 - \frac{1}{m}\left(\frac{\Gamma(m+\frac{1}{2})}{\Gamma(m)}\right)^2\right)$$

However, this can be simplified to:

$$Var_{Nakagami} = \Omega\left(1 - \frac{1}{m}\right)$$

Substitute m and Ω from $S \sim$ Nakagami$(m = \frac{n-1}{2}, \Omega = \sigma^2)$, we got:

$$Var_\theta(\hat{\sigma}) = \sigma^2\left(1 - \frac{2}{n-1}\right).$$

Let's move on to variance of σ^2_{MVUE}. Let's recall that the sample variance S^2 is the MVUE for the population variance σ^2.

$$S^2 \sim \text{Gamma}\left(\alpha = \frac{n-1}{2}, \beta = \frac{2\sigma^2}{n-1}\right)$$

The variance of a Gamma random variable is given by:

$$Var_\theta(S^2) = \alpha\beta^2$$

Now, substitute $\alpha = \frac{n-1}{2}$ and $\beta = \frac{2\sigma^2}{n-1}$,

$$Var_\theta(S^2) = \frac{n-1}{2}\left(\frac{2\sigma^2}{n-1}\right)^2 = \frac{2\sigma^4}{n-1}$$

Thus,

$$Var_\theta(\sigma^2_{MVUE}) = \frac{2\sigma^4}{n-1}.$$

We have found all the variance of the each MVUEs. We can use Monte Carlo simulation to obtain the variances of μ_{QQ}, σ_{QQ}, and σ^2_{QQ}. To obtain them via Monte Carlo simulation, we are using the same random samples generated in question 1. Now, let's compute the empirical relative efficiencies, r-code for this section can be found under appendix.

Table 2 is a table of values of calculated **empirical relative efficiency** for each value of n.

n	$\text{eff}_n\mu$	$\text{eff}_n\sigma$	$\text{eff}_n\sigma^2$
5	$4.537455e^{-}06$	1.652568	1.762731
10	$1.840257e^{-}04$	1.155714	4.076946
30	$7.072542e^{-}06$	1.031026	13.825499
50	$3.193308e^{-}05$	1.015970	23.750619
75	$2.328991e^{-}06$	1.006212	35.942679
100	$1.044926e^{-}05$	1.002931	48.288039

Table 2: Empirical relative efficiency $\text{eff}_n(\hat{\theta}, \hat{\theta}_{MVUE}|n)$ for each n

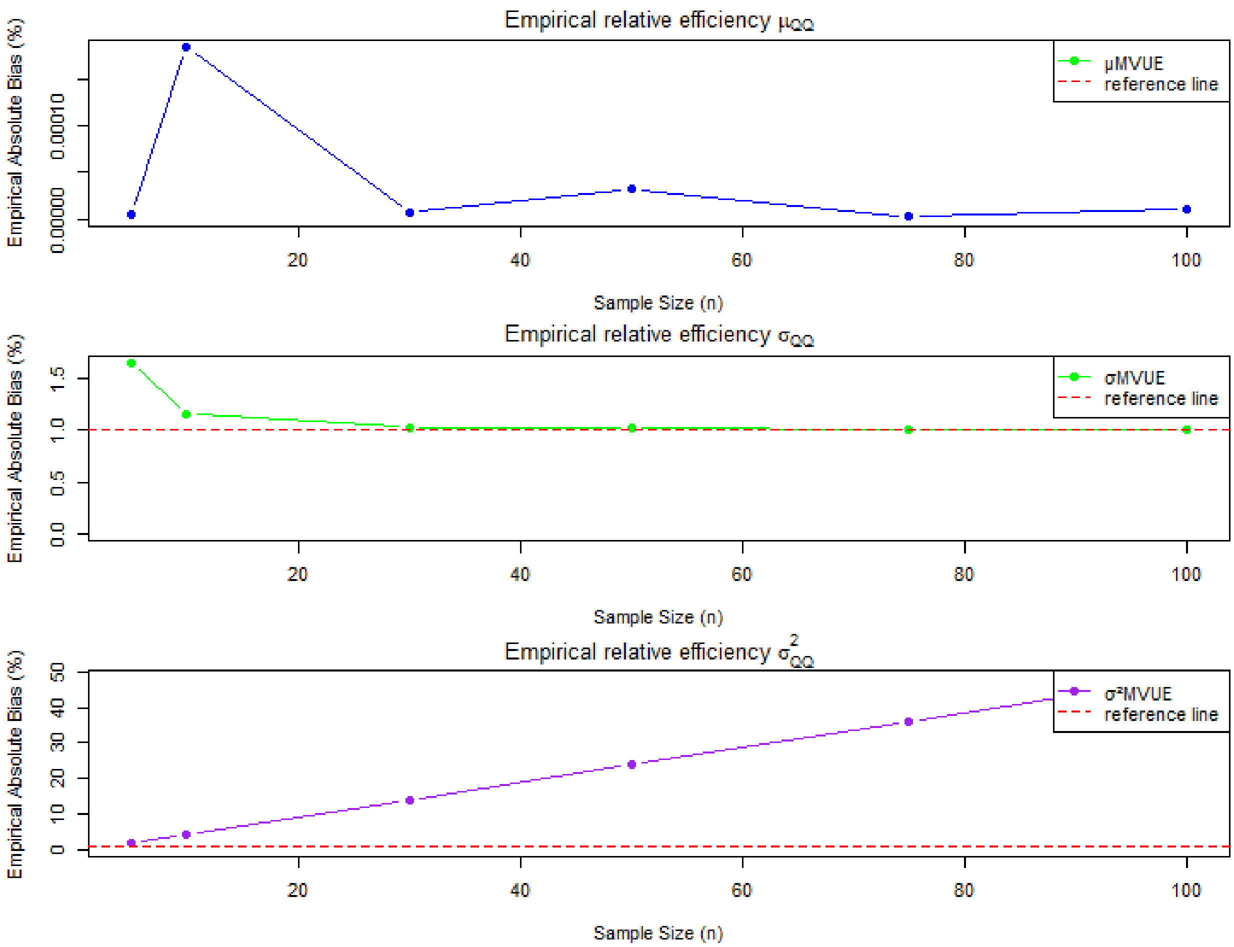

Figure 4: Empirical relative efficiency $\text{eff}_n(\hat{\theta}, \hat{\theta}_{MVUE})$ plots

Based off these plots, since the higher the value the more efficient it is, we can see that only the estimator σ^2 is more efficient as sample size increases, while the other estimators efficiencies get worse as the sample size increase.

F Conclusion

From this project, we can conclude that the QQ-methodology provides reliable unbiased estimators of μ, σ, and σ^2 particularly when the sample size is sufficiently large. As the sample size increases, the bias of the estimator approaches zero, demonstrating their ability to achieve approximately unbiased estimation fo the parameters. However, this accuracy comes with a trade-off in terms of empirical relative efficiency. If the QQ estimator for μ and σ perform well, their efficiencies relative to the optimal MVUEs tend to decrease as sample size n increases.

In contrast, the QQ estimator for σ^2 performs exceptionally well, maintaining high empirical relative efficiency even at larger sample sizes. This makes the QQ-methodology particularly advantageous for variance estimation, as it combines both accuracy and efficiency.

Overall, the QQ methodology proves to be an effective approach for parameter estimation, demonstrating strong performance across multiple metrics. However, it underscores the need to balance accuracy and efficiency, which can vary depending on the specific parameter being estimated and the sample size.

G Reference

"Nakagami Distribution." Wikipedia, Wikimedia Foundation, 7 Dec. 2024,

`en.wikipedia.org/wiki/Nakagami_distribution.`

De Leon, Alex. R. "Lecture 2 Parametric Families of Distributions ." 2024.

De Leon, Alex. R. "Lecture 6 Optimality Criteria for Estimation ." 2024.

Hogg, Robert V., et al. Introduction to Mathematical Statistics. Pearson, 2019., Theorem 3.6.1, p. 210-216

H Appendix

Appendix A for question 1

```
set.seed(4212024)
values_n<- c(5, 10, 30, 50, 75, 100)
R<- 10000
mu<- 0
variance<- 1
results<- list()
for ( n in values_n) {
  # simulate R samples of size n
  samples<- replicate (R, sort( rnorm(n)))
  # compute hazen scores
  hazenscore<- qnorm((1:n - 0.5)/n)
  avg_orderstats<- rowMeans(samples)
  results[[as.character(n)]]<- list(hazenscore= hazenscore,
     avg_orderstats= avg_orderstats)
}
par(mfrow = c(2,3))

for ( n in values_n) {
  res <- results[[as.character(n)]]
  hazenscore<- res$hazenscore
  avg_orderstats<- res$avg_orderstats

  #regression line (OLS)
  fit <- lm(avg_orderstats~ hazenscore)

  #scatterplot
  plot(hazenscore, avg_orderstats, main = paste("n=", n), xlab=
     "Hazen scores", ylab= "Ordered Statistics Average")

  #regression line
  abline(fit, col="pink", lwd = 2)
}
```

```r
set.seed(4212024)

mu = 0
sigma_squared = 1
random_samples = 10000
n_vals = c(5,10,30,50,75,100)

par(mfrow=c(2,3))
hazen_scores <- list()
simulated_normals <- list()
averages <- c()
estimates_mu <- list()
estimates_sigma <- list()
estimates_sd <- list()
for (n in n_vals){
  estimated_sigma_qqs <- numeric(random_samples)
  estimated_sd_qqs <- numeric(random_samples)
  estimated_mu_qqs <- numeric(random_samples)
  hazen_score <- c()
  simulated_normal <- replicate(random_samples, sort(rnorm(n, mean =
     mu, sd = sigma_squared)))
  simulated_normals[[as.character(n)]] <- simulated_normal
  #average = rowMeans(simulated_normal)
  j = seq(1,n)
  for (values in j){
    hazen_score <- qnorm((j-0.5)/n)
  }
  hazen_scores[as.character(n)] <- list(hazen_score)
  #OLM
  for (i in 1:random_samples){
    average = sort(rnorm(n, mean = mu, sd = sigma_squared))
    fit <- lm(average~hazen_score)
    estimated_mu_qqs[i] <- coef(fit)[1]
    estimated_sigma_qqs[i] <- coef(fit)[2]
    estimated_sd_qqs[i] <- (coef(fit)[2])^2
  }
  averages <- append(averages,sum(average))

  estimates_mu[as.character(n)] = list(estimated_mu_qqs)
  estimates_sigma[as.character(n)] = list(estimated_sigma_qqs)
  estimates_sd[as.character(n)] = list(estimated_sd_qqs)

  #plot
  plot(hazen_score, average, main = "n=5,10,30,50,75,100", xlab=
     "hazen scores", ylab= "averages")
  abline(fit)
}
```

Appendix B for question 2

```
#Question 2 Part a

#Obtain mean hazen score \overline{z}
mean_hazen_scores <- c()
for (j in n_vals){
  mean_hazen_scores <- append(mean_hazen_scores,
      sum(hazen_scores[[as.character(j)]])/n)
}
mean_z = sum(mean_hazen_scores)/length(mean_hazen_scores)

#obtain the sum of a_j from j=1 to n
sum_aj <- list()
for (j in n_vals){
  a_numerator = sum(hazen_scores[[as.character(j)]]) - mean_z
  a_denominator = (sum(hazen_scores[[as.character(j)]]) - mean_z)^2
  print(a_denominator)
  sum_aj[as.character(j)] <- a_numerator/a_denominator
}

#calculate emperical biases
bias_mean <- c()
bias_sigma <- c()
bias_sd <- c()
for (i in n_vals){
  bias_mean <- append(bias_mean,
      abs(mean(estimates_mu[[as.character(i)]]) - mu)*100)
  bias_sigma <- append(bias_sigma,
      abs(mean(estimates_sigma[[as.character(i)]]) -
      sqrt(sigma_squared))*100)
  bias_sd <- append(bias_sd,
      abs(mean(estimates_sd[[as.character(i)]]) - sigma_squared)*100)
}
# End Question 2 part a
```

```
#Question 2 part b
par(mfrow = c(3, 1), mar = c(4, 4, 2, 1), oma = c(0, 0, 2, 0))
plot(n_vals, bias_mean, type = "b", col = "blue", pch = 16, lty = 1,
     xlab = "Sample␣Size␣(n)", ylab = "Empirical␣Absolute␣Bias␣(%)",
     main = expression(paste("Bias␣of␣", mu[QQ])),
     ylim = c(0, max(bias_mean)))
abline(h = 0, col = "red",lty="dashed")
legend("topright", legend = c(" QQ ␣Bias", "MVUE␣Bias␣=␣0"),
       col = c("green", "red"), lty = c(1, 2), pch = c(16, NA))

plot(n_vals, bias_sigma, type = "b", col = "green", pch = 16, lty =
   1,
     xlab = "Sample␣Size␣(n)", ylab = "Empirical␣Absolute␣Bias␣(%)",
     main = expression(paste("Bias␣of␣", sigma[QQ])),
     ylim = c(0, max(bias_sigma)))
abline(h = 0, col = "red",lty="dashed")
legend("topright", legend = c(" QQ ␣Bias", "MVUE␣Bias␣=␣0"),
       col = c("green", "red"), lty = c(1, 2), pch = c(16, NA))

plot(n_vals, bias_sd, type = "b", col = "purple", pch = 16, lty = 1,
     xlab = "Sample␣Size␣(n)", ylab = "Empirical␣Absolute␣Bias␣(%)",
     main = expression(paste("Bias␣of␣", sigma[QQ]^2)),
     ylim = c(0, max(bias_sd)))
abline(h = 0, col = "red",lty="dashed")
legend("topright", legend = c("  QQ  ␣Bias", "MVUE␣Bias␣=␣0"),
       col = c("purple", "red"), lty = c(1, 2), pch = c(16, NA))
#end question 2 part b
```

Appendix C for question 3

```r
# begin Question 3 part a:
eff_means <- c()
eff_sigmas <- c()
eff_sds <- c()
for (i in n_vals){
  var_mean = 0
  var_sigma = 0
  var_sd = 0
  for (r in random_samples){
    var_mean = var_mean +
        (((abs(mean(estimates_mu[[as.character(i)]]) -
        mu)*100)^2)/(random_samples-1))
    var_sigma = var_sigma +
        (((abs(mean(estimates_sigma[[as.character(i)]]) -
        mu)*100)^2)/(random_samples-1))
    var_sd = var_sd + (((abs(mean(estimates_sd[[as.character(i)]]) -
        mu)*100)^2)/(random_samples-1))
  }
  eff_means <- append(eff_means, var_mean/(sigma_squared/i))
  eff_sigmas <- append(eff_sigmas,
      var_sigma/(sigma_squared*(1-(2/(i-1)))))
  eff_sds <- append(eff_sds, var_sd*(i-1)/(2*sigma_squared^2))
}
# end Question 3 part a
```

```
# Begin Question 3 part b:
par(mfrow = c(3, 1), mar = c(4, 4, 2, 1), oma = c(0, 0, 2, 0))
plot(n_vals, eff_means, type = "b", col = "blue", pch = 16, lty = 1,
     xlab = "Sample␣Size␣(n)", ylab = "Empirical␣Absolute␣Bias␣(%)",
     main = expression(paste("Empirical␣relative␣efficiency␣",
        mu[QQ])),
     ylim = c(0, max(eff_means)))
abline(h = 1, col = "red",lty="dashed")
legend("topright", legend = c(" MVUE ", "reference␣line"),
       col = c("green", "red"), lty = c(1, 2), pch = c(16, NA))

plot(n_vals, eff_sigmas, type = "b", col = "green", pch = 16, lty =
   1,
     xlab = "Sample␣Size␣(n)", ylab = "Empirical␣Absolute␣Bias␣(%)",
     main = expression(paste("Empirical␣relative␣efficiency␣",
        sigma[QQ])),
     ylim = c(0, max(eff_sigmas)))
abline(h = 1, col = "red",lty="dashed")
legend("topright", legend = c(" MVUE ", "reference␣line"),
       col = c("green", "red"), lty = c(1, 2), pch = c(16, NA))

plot(n_vals, eff_sds, type = "b", col = "purple", pch = 16, lty = 1,
     xlab = "Sample␣Size␣(n)", ylab = "Empirical␣Absolute␣Bias␣(%)",
     main = expression(paste("Empirical␣relative␣efficiency␣",
        sigma[QQ]^2)),
     ylim = c(0, max(eff_sds)))
abline(h = 1, col = "red",lty="dashed")
legend("topright", legend = c("  MVUE  ", "reference␣line"),
       col = c("purple", "red"), lty = c(1, 2), pch = c(16, NA))
# End Question 3 part b
```