

TITLE HERE

A Thesis Proposal Presented to
The Faculty of the SS
univ

In Partial Fulfillment
of the Requirements for the Degree of
degree
1st Semester A.Y. 2025-2026

by
Nmae

Abstract

Contents

Abstract	2
1 Methodology	4
1.1 Parametric bootstrap	4
1.2 Nonrank-based method	5
2 Introduction	7
2.1 Background of the Study	7
2.2 Statement of the Problem	7
2.3 Objective of the Study	7
2.4 Study Hypothesis	7
2.5 Significance of the Study	7
2.6 Scope and Limitation	7
2.7 Definition of Terms	7
3 Background	7
References	7
Appendices	7
Codes for CI calculation	7
Codes for T_1, T_2, T_3 calculation	11

1 Methodology

Let $\theta_1, \theta_2, \dots, \theta_K$ be the true parameter values and $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_K$ be the estimates obtained.

1.1 Parametric bootstrap

Let $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_K$ be independent but not identically distributed estimates. For this study, it is assumed that $\hat{\theta}_k \sim N(\theta_k, \sigma_k^2)$, $k = 1, 2, \dots, K$, where σ_k^2 is known. Denote the corresponding ordered values by $\hat{\theta}_{(1)}, \hat{\theta}_{(2)}, \dots, \hat{\theta}_{(K)}$.

Algorithm 1 Computation of Joint Confidence Region via Parametric Bootstrap

1: **for** $b = 1, 2, \dots, B$ **do**

2: Generate $\hat{\theta}_{bk}^* \sim N(\hat{\theta}_k, \sigma_k^2)$, $i = 1, 2, \dots, K$ and let $\hat{\theta}_{b(1)}, \hat{\theta}_{b(2)}, \dots, \hat{\theta}_{b(K)}$ be the corresponding ordered values

	$k = 1$	$k = 2$	\dots	$k = K$
$b = 1$	$\hat{\theta}_{1(1)}^*$	$\hat{\theta}_{1(2)}^*$	\dots	$\hat{\theta}_{1(K)}^*$
$b = 2$	$\hat{\theta}_{2(1)}^*$	$\hat{\theta}_{2(2)}^*$	\dots	$\hat{\theta}_{2(K)}^*$
\vdots	\vdots	\vdots	\dots	\vdots
$b = B$	$\hat{\theta}_{B(1)}^*$	$\hat{\theta}_{B(2)}^*$	\dots	$\hat{\theta}_{B(K)}^*$

3: Compute

$$\hat{\sigma}_{b(k)}^* = \sqrt{\text{kth ordered value among } \{\hat{\theta}_{b1}^{*2} + \sigma_1^2, \hat{\theta}_{b2}^{*2} + \sigma_2^2, \dots, \hat{\theta}_{bK}^{*2} + \sigma_K^2\} - \hat{\theta}_{(k)}^{*2}}$$

4: Compute $t_b^* = \max_{1 \leq k \leq K} \left| \frac{\hat{\theta}_{b(k)}^* - \hat{\theta}_k^*}{\sigma_{b(k)}^*} \right|$

5: **end for**

6: Compute the $(1 - \alpha)$ -sample quantile of $t_1^*, t_2^*, \dots, t_B^*$, call this \hat{t} .

7: The joint confidence region of $\theta_{(1)}, \theta_{(2)}, \dots, \theta_{(K)}$ is given by

$$\mathfrak{R} = [\hat{\theta}_{(1)} \pm \hat{t} \times \hat{\sigma}_{(1)}] \times [\hat{\theta}_{(2)} \pm \hat{t} \times \hat{\sigma}_{(2)}] \times \dots \times [\hat{\theta}_{(K)} \pm \hat{t} \times \hat{\sigma}_{(K)}]$$

where $\hat{\sigma}_{(k)}$ is computed as

$$\hat{\sigma}_{(k)} = \sqrt{\text{kth ordered value among } \{\hat{\theta}_1^2 + \sigma_1^2, \hat{\theta}_2^2 + \sigma_2^2, \dots, \hat{\theta}_K^2 + \sigma_K^2\} - \hat{\theta}_{(k)}^2}$$

Algorithm 2 Computation of Coverage Probability for Parametric Bootstrap

For given values of $\theta_1, \theta_2, \dots, \theta_K$ and thus $\theta_{(1)}, \theta_{(2)}, \dots, \theta_{(K)}$

1: **for** replications = 1, 2, \dots , 5000 **do**

2: Generate $\hat{\theta}_k \sim N(\theta_k, \sigma_k^2)$, for $k = 1, 2, \dots, K$

3: Compute the rectangular confidence region \mathfrak{R} using Algorithm 1.

4: Check if $(\theta_{(1)}, \theta_{(2)}, \dots, \theta_{(K)}) \in \mathfrak{R}$ and compute

$$\begin{aligned} T_1 &= \frac{1}{K} \sum_{k=1}^K |\Lambda_{Ok}| \\ T_2 &= \prod_{k=1}^K |\Lambda_{Ok}| \\ T_3 &= 1 - \frac{K + \sum_{k=1}^K |\Lambda_{Ok}|}{K^2} \end{aligned}$$

5: **end for**

6: Compute the proportion of times that the condition in step 4 is satisfied and the average of T_1, T_2 , and T_3 .

1.2 Nonrank-based method

The nonrank-based method assumes that $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_K) \sim N(\boldsymbol{\theta}, \boldsymbol{\Sigma})$. It accounts for potential correlation among items being ranked. For this case, an exchangeable correlation, $\boldsymbol{\rho}$ (See Equation 1.1.), is assumed and used in the calculation of the variance covariance matrix (See Equation 1.2.).

$$\boldsymbol{\rho} = (1 - \rho) \mathbf{I}_K + \rho \mathbf{1}_K \mathbf{1}_K' \quad (1.1)$$

$$\boldsymbol{\Sigma} = \boldsymbol{\Delta}^{1/2} \boldsymbol{\rho} \boldsymbol{\Delta}^{1/2} \quad (1.2)$$

where $\boldsymbol{\Delta} = \text{diag}\{\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2\}$, with known σ_k 's and ρ is studied for 0.1, 0.5, 0.9.

Algorithm 3 Computation of Joint Confidence Region via Nonrank-based Method

Let the data consist of $\hat{\theta}_1, \dots, \hat{\theta}_K$ and suppose Σ is known

- 1: **for** $b = 1, 2, \dots, B$ **do**
- 2: Generate $\hat{\theta}_b^* \sim N_K(\hat{\theta}, \Sigma)$ and write $\hat{\theta}_b^* = (\hat{\theta}_{b1}^*, \hat{\theta}_{b2}^*, \dots, \hat{\theta}_{bK}^*)'$
- 3: Compute $t_b^* = \max_{1 \leq j \leq K} \left| \frac{\hat{\theta}_{bj}^* - \hat{\theta}_j}{\sigma_j} \right|$
- 4: **end for**
- 5: Compute the $(1 - \alpha)$ -sample quantile of $t_1^*, t_2^*, \dots, t_B^*$, call this \hat{t} .
- 6: The joint confidence region of $\theta_1, \theta_2, \dots, \theta_K$ is given by

$$\mathfrak{R} = [\hat{\theta}_1 \pm \hat{t} \times \sigma_1] \times [\hat{\theta}_2 \pm \hat{t} \times \sigma_2] \times \dots \times [\hat{\theta}_K \pm \hat{t} \times \sigma_K]$$

Algorithm 4 Computation of Coverage Probability for Nonrank-based Method

For given values of $\theta_1, \theta_2, \dots, \theta_K$ and Σ

- 1: **for** replications = 1, 2, \dots , 5000 **do**
 - 2: Generate $\hat{\theta} \sim N_K(\theta, \Sigma)$
 - 3: Compute the rectangular confidence region \mathfrak{R} using Algorithm 3.
 - 4: Check if $(\theta_1, \theta_2, \dots, \theta_K) \in \mathfrak{R}$ and compute T_1, T_2 , and T_3 .
 - 5: **end for**
 - 6: Compute the proportion of times that the condition in step 4 is satisfied and the average of T_1, T_2 , and T_3 .
-

2 Introduction

2.1 Background of the Study

2.2 Statement of the Problem

2.3 Objective of the Study

2.4 Study Hypothesis

2.5 Significance of the Study

2.6 Scope and Limitation

2.7 Definition of Terms

3 Background

THIS IS Rizzo (2008) and Klein et al. (2020)

References

Klein, M., Wright, T., & Wieczorek, J. (2020). *A joint confidence region for an overall ranking of populations.*

Rizzo, M. (2008). *Statistical computing with r.*

Appendices

Codes for CI calculation

```
get_independent_ci <- function(theta_hat,  
                                S,  
                                alpha){
```

```

K <- length(theta_hat)
gamma = 1-(1-alpha)^(1/K)
z = qnorm(1-gamma/2)
ci_lower <- theta_hat - z*S
ci_upper <- theta_hat + z*S
return(list(
  ci_lower = ci_lower,
  ci_upper = ci_upper
))
}

```

```

get_bonferroni_ci <- function(theta_hat,
                               S,
                               alpha){
  K <- length(theta_hat)
  z = qnorm(1-(alpha/K)/2)
  ci_lower <- theta_hat - z*S
  ci_upper <- theta_hat + z*S
  return(list(
    ci_lower = ci_lower,
    ci_upper = ci_upper
  ))
}

```

```

get_parametric_ci <- function(B,
                               theta_hat,
                               S,
                               alpha) {
  K <- length(theta_hat)

```



```

# step 1a =====
thetahat_star <- sapply(seq_len(K), function(i) {
  rnorm(B, mean = theta_hat[i], sd = S[i])
})
colnames(thetahat_star) <- paste0("thetahat_star",
                                   sprintf("%02d", 1:K))
sorted_thetahat_star <- t(apply(thetahat_star, 1, sort))
colnames(sorted_thetahat_star) <- paste0("sorted_thetahat_star",
                                           sprintf("%02d", 1:K))

# step 1b =====
variance_vector <- S^2
minuend <- thetahat_star^2 + rep(
  variance_vector, each = nrow(thetahat_star))
sigma_hat_star <- sqrt(
  t(apply(minuend, 1, sort)) - sorted_thetahat_star^2)

# step 1c =====
sorted_theta_hat <- sort(theta_hat)
t_star <- apply(
  abs(
    (
      sorted_thetahat_star - rep(
        sorted_theta_hat,
        each = nrow(sorted_thetahat_star)
      )
    )/sigma_hat_star
  ),
  1,
  max)

# step 2 =====

```

```

t_hat <- quantile(t_star, probs = 1 - alpha)

# step 3 =====

sigma_hat <- sqrt(
  sort(theta_hat^2 + variance_vector) - sorted_theta_hat^2)

# step 6 =====

ci_lower <- sorted_theta_hat - t_hat*sigma_hat
ci_upper <- sorted_theta_hat + t_hat*sigma_hat
return(list(
  ci_lower = ci_lower,
  ci_upper = ci_upper
))
}

```

```

get_nonrankbased_ci <- function(B,
                                theta_hat,
                                alpha,
                                varcovar_matrix) {
  K <- length(theta_hat)

  # step 1a =====

  generate_data <- function(){MASS::mvrnorm(n = 1,
                                             mu = theta_hat,
                                             Sigma = varcovar_matrix)}

  thetahat_star <- t(replicate(B, generate_data()))

  # step 1b =====

  t_star <- apply(thetahat_star,
                  1,
                  function(x) max(abs((x - theta_hat) / sqrt(
                    diag(varcovar_matrix))))))

  # step 2 =====

```

```

t_hat <- quantile(t_star, probs = 1 - alpha)

# step 3 =====
ci_lower <- theta_hat - t_hat*sqrt(diag(varcovar_matrix))
ci_upper <- theta_hat + t_hat*sqrt(diag(varcovar_matrix))

return(list(
  ci_lower = ci_lower,
  ci_upper = ci_upper
))
}

```

Codes for T_1, T_2, T_3 calculation

```

get_ranks <- function(k, tuple_list){
  Lambda_lk <- which(
    tuple_list[,2]<=tuple_list[k,1])
  Lambda_lk <- Lambda_lk[Lambda_lk != k]
  Lambda_Ok <- which(
    tuple_list[,2]>tuple_list[k,1] & tuple_list[k,2] > tuple_list[,1])
  Lambda_Ok <- Lambda_Ok[Lambda_Ok != k]
  ranks <- seq(
    length(unique(Lambda_lk)) + 1,
    length(unique(Lambda_lk)) + length(unique(Lambda_Ok)) + 1,
    1
  )
  return(list(
    ranks = ranks,
    Lambda_Ok = Lambda_Ok
  ))
}

```

```
get_t1 <- function(v) mean(v)
```

```
get_t2 <- function(v) prod(v)^(1/length(v))
```

```
get_t3 <- function(v) {  
  1 - ((length(v)+sum(v))/(length(v)^2))  
}
```