Non-Parametric Robustness Simulation Study Verification

Bryce Anderson

March 2024

# Introduction

The decommissioning of nuclear facilities is a critical environmental and safety undertaking. Sohn and Hong’s (2021) article presents a robust simulation study using the Sign test and Wilcoxon Rank Sum (WRS) test, for assessing the cleanliness of nuclear decommissioning sites. Sohn and Hong (2021) explore the ”MARSSIM” approach, which provides a structured methodology using non-parametric statistical tests for final status surveys of the nuclear sites that determine whether a given site is clean. They use a Monte Carlo simulation so they can experiment on a wide range of site-specific scenarios.

This paper aims to reproduce and validate their findings through a simulation study and verify the statistical analysis conducted. Additionally, it explores parametric alternatives, comparing their applicability and effectiveness against the original non-parametric approach. First, we will discuss the MARRSIM approach and dissect the methodology to simulate hypothetical surveys of radioactive sites.

# Summary

In the MARSSIM approach, the Sign test and the Wilcoxon Rank Sum Test are used in the hypothesis testing. These tests are used because environmental data is usually not normally distributed. *H*0 is phrased as the statement disqualifying the release of the survey unit of interest while *H*1 is phrased as the statement qualifying the release The goal of the hypothesis testing is to reject *H*0 (that the site is contaminated). To quantify this *H*0 = the mean of the net contamination distribution in a survey unit exceeds DCGL where the DCGL is the threshold for acceptable contamination. If the net contamination is below this level, the site is considered to meet the decommissioning criterion.

We are going to look into the power of the tests based on several varying parameters which include: designed Type I error (null hypothesis is incorrectly

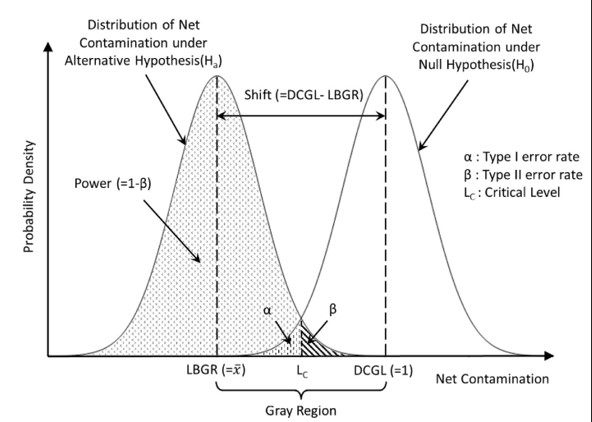


Figure 1: Relationship between parameters of statistical conditions and radiological characteristics of a survey unit

rejected when it is actually true) and Type II errors (when the null hypothesis is incorrectly accepted when it is false), shifts in the data and the sample size of the data. To get an outline of all of this we can look at figure 1:

LBGR (Lower Boundary of Gray Region) - This is the average level of net contamination observed in the survey unit. It’s essentially the threshold below which a site is considered potentially clean. DCGL (Upper Boundary of Gray Region) - This is the standard for acceptable contamination. If the net contamination is below this level, the site is considered to meet the decommissioning criterion. DCGL is normalized to 1 for simplification, allowing for easy comparison of contamination levels against the regulatory threshold. This would be set to whatever threshold is deemed appropriate for the allowable amount of radiation in the survey cases. In a situation where the land would be used for something like farming, this would be very low, but in a case where the land would be used for something like a steel factory, it would be set higher. DCGL and LBGR are assigned a constant value of 1 and 0.5, respectively, resulting in the shift having constant value of 0.5. Shift (D = DCGL - LBGR) - The distance between the mean contamination level considered clean and the DCGL. It is crucial for determining the number of samples needed to reliably assess the site.

# Methods and Monte Carlo Simulation Outline

We are going simulate the radiological characteristics of survey units to account for the variability in contamination levels across different scenarios. We start with a hypothetical survey unit where it is assumed that the data is normally distributed around the mean. Then, three relative shifts are considered (1, 2, and 3). This parameter reflects the difficulty in distinguishing the mean contamination level from the DCGL. A relative shift of 1 suggests high difficulty (sd = 0.5). A relative shift of 2 suggests medium difficulty (sd = 0.25).A relative shift of 3 suggests low difficulty (sd 0.17). Then to cover all the plausible statistical conditions, three values (0.01, 0.025 and 0.05) are assigned to each of *α* and *β* which correspond to Type I and Type II errors respectively. Lastly, we will apply a scaling factor (0.8, 1, 1.2) on the sample to see how sample size affects the power of the tests. So all in all, we will have 27 statistical conditions and 3 scaling factors resulting in 81 overall cases we will be testing.

For each test (Sign, WRS, and T-test), random samples are generated according to specified means (0.4 through 1.2). Nested for loops are used to loop through the alpha,beta,standard deviation and mean values. We then calculate the test statistic and critical value and return whether the test passes or failed. If the test statistic is greater than the critical value, we reject the null hypothesis, i.e the test passes. We iterate over this 1000 times, in this case for simplicity. The power calculations are done by comparing the number of pass events against the total number of iterations. This process then repeats for every possible combination of parameters.

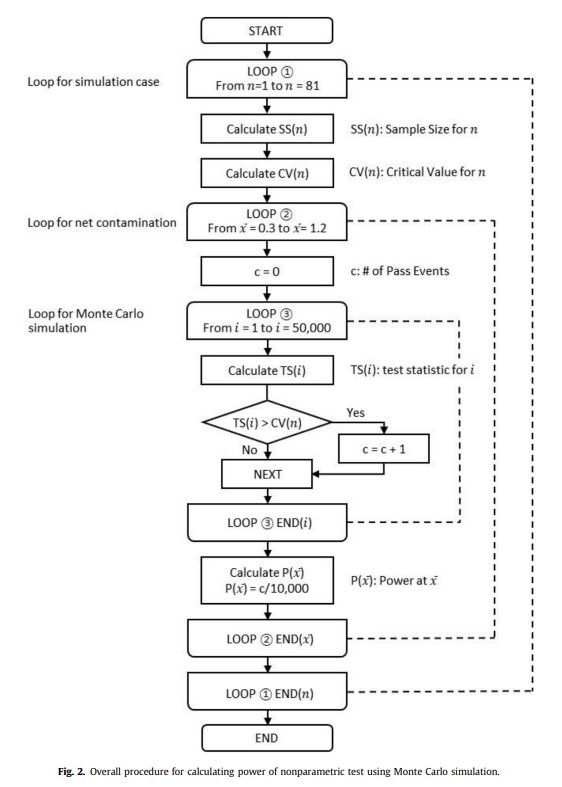


Figure 2: Overall procedure for calculating power of non-parametric test using

Monte Carlo simulation

# Tests and Assumptions

First, we must assume the data was chosen randomly and independently from a continuous distribution. In the sign test, we calculate the test-statistic by taking the difference between the mean and each sample then our *S*+ is equal to the sum of all the differences greater than 0. If any differences are equal to 0, we discard them. We calculate the critical value from the binomial distribution if the sample size is less than 50 and if it is greater than 50 then we use the large sample approximation. In the MARISSM approach, the survey tests for radionuclides which are naturally occurring radioactive isotopes of elements that emit radiation as they decay. If these are present, then an additional reference survey must be used. Thus, the WRS test must be used, because we must compare two samples. In the simulation the reference survey is set a constant mean of 0.1 for simplicity. The test statistic is calculated by combining the data from both samples and ranking them from smallest to largest. We then sum the ranks of each sample and the sum of the ranks from the reference sample is our test statistic. It must be noted that we adjust the reference area by adding the DCGL to each reference area measurement. Also, if there are ties, we use mid ranks. Lastly, the statistical sample size should be around double that of the sign test due to the fact we want two equal samples. All of the formula to calculate the test statistics and critical values are clearly detailed in my R code.

Lastly, since we are drawing from a normal distribution, it is appropriate to use the one sample T-Test. We can also use a two-sample t-test instead of the WRS test if the reference sample and survey unit are known to be normally distributed. The P-values are calculated directly in R using t.test and then compared with the alpha values. If *α > P* − *value* then we reject the null hypothesis i.e the test passes.

# Results

From my data I want to answer the following questions.

•At LBGR, how often does my test correctly say ’clean’?

•At DCGL, how often does my test correctly say ’dirty’?

•How far are my test’s estimated powers from what they should ideally be?

•How does the T-test compare?

First, we need to clarify the interpretation of powers. When the hypothetical mean is set less than 1 this is the probability of a correct decision. When it is set larger than 1, the power should be interpreted as the Type I error rate, incorrectly rejecting a true null hypothesis. The percentage error is the absolute difference between the estimated power and the designed power, divided by the designed power, multiplied by 100. If the estimated power is less than the designed power, the percentage error is negative, indicating under performance. If the estimated power exceeds the designed power, the percentage error is positive, indicating over performance.

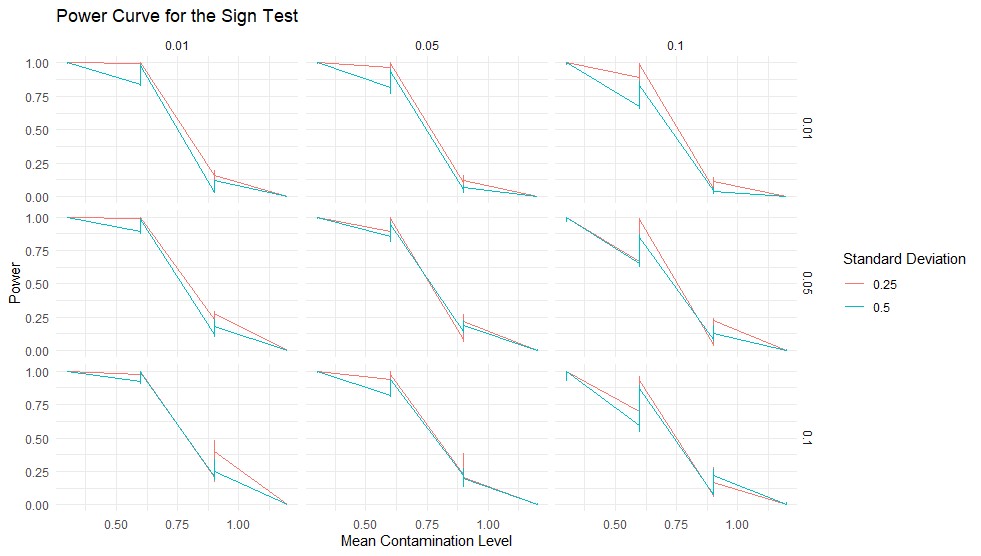


Figure 3: Power Curve of the Sign-Test. Note the values on the top and on the side (0.01,0.05,0.1) are the alpha and beta value interactions.

All in all, the same patterns arose as from the original paper. The study observed that the estimated power curves, except for exceptional cases with a standard deviation of 0.5 and a scaling factor of 0.8, reach the designed power level (1 *β*) at an LBGR value greater than 0.5. It was found that the power of each test was most effective at the LBGR and DCGL for the WRS and Sign test, however this was not true with the T-Test which I will get into later. When comparing the sample sizes, test statistics, and critical values with the given parameters, I got the same values as the original paper did. See the attached excel sheet for this.

I found that the Sign test over performed in many cases where the statistical sample size was scaled by 1.2. As mentioned in in Sohn and Hong’s (2021) paper, this is noteworthy as survey samples are very expensive to collect so it would be financially beneficial to reduce the required amount of samples. When we compare the T-test to our other tests, there were cases where the T-test and the sign test were comparable. When the alpha values were low and the beta values were high and the sample size was large (scaled of 1.2) and the shift was low difficulty. In most scenarios like this, the powers were nearly identical and their errors were both positive meaning that they were over performing. For example when *α* = 0*.*01 *β* = 0*.*05, sd = .25, and the scalar was 1.2. The power for the sign test was 1 and the power for the t-test was

0.988. Their percentage errors were 5.05 and 5.26 respectively meaning that it

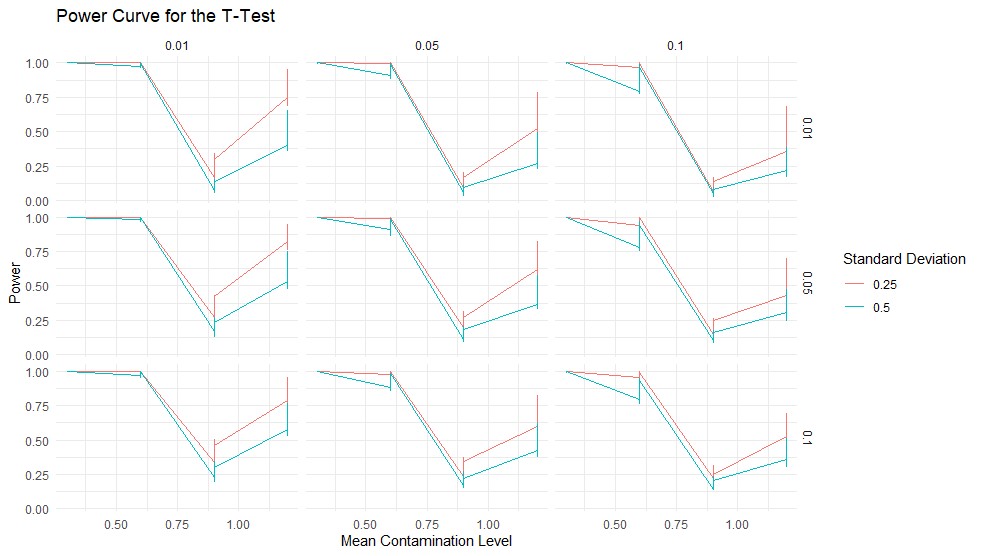


Figure 4: Power Curve of the T-Test. Note the values on the top and on the side (0.01,0.05,0.1) are the alpha and beta value interactions.

actually exceeded the designed power. To conclude, it would make sense that the tests are similar in parameters like these. The data is hardly skewed and it is an easily identifiable mean with a large sample size. Thus, the T-Test would be suitable in these cases. I have graphed the power curves of the T-Test below: However, the T-Test was very susceptible to Type I error. In cases where the set mean was above the DCGL, it had power from between 0.2 all the way up to 0.94 in some cases. It had the hardest time when the sample size was low and the standard deviation was high. This test suffered when sample size was low which is problematic for this field of work, because small samples are used in most cases due to the difficulty of collecting data. From this we can not only see that the sign-test is more robust than the T-test in several ways. Not only can it be used on non-normal distributions, but it is less effected by sample size.

# Discussion

I found it quite interesting to play around with all the different parameters of the samples and see how it affects the decision making of the tests. I don’t think I ever realized how hard it is sometimes to get good data in the environmental sciences. I’ve gained valuable insights into the prevalence of non-parametric tests in the field, which are particularly advantageous when sample sizes are small and data distribution is skewed. While coding the Monte Carlo Simulation, I learned a lot more about R and it’s functionality; I am definitely stronger in R, but I wish I took 445 first. If I were to do this project again, I would consider simulating data based on other distributions that might realistically describe contamination levels, such as log-normal or exponential distributions. Including different tests to find more exact thresholds for parameters that make the tests most effective would be useful, especially in the parametric case. I also learned a lot about spatial variability during this project which would be something I could implement into my simulations to reflect the heterogeneity of contamination across a site. To conclude, I think this project has deepened my understanding of non-parametric testing and the advantages of it compared to parametric testing.

# R-APPENDIX

# Parameters: alpha\_values <- c(0.01,0.05,0.1) # Significance level beta\_values <- c(0.01,0.05, 0.1) dcgl <- 1 sd\_values <- c(0.5,0.25, 0.17) # Standard deviation scalar\_factors <- c(.8,1,1.2) #Sample size scalars

# Set general parameters num\_cases <- 81 # Number of simulation cases num\_iterations <- 1000 # Number of Monte Carlo iterations mean\_range <- seq(0.3, 1.2, by=0.3) # Range of mean contamination levels background\_contamination = c(0.1)

# Initialize the results dataframe simulation\_results <- data.frame(

Case\_ID = integer(),

Alpha = numeric(),

Beta = numeric(),

SD = numeric(),

Scalar = numeric(),

Sample\_Size\_Sign = integer(),

Sample\_Size\_WRS = integer(),

Mean\_Contamination = numeric(),

TS\_Sign = numeric(),

TS\_WRS = numeric(),

Critical\_Value\_Sign = numeric(),

Critical\_Value\_WRS = numeric(),

Power\_Sign = numeric(), Power\_WRS = numeric(),

T\_Test\_P\_Value = numeric(),

T\_Test\_Power = numeric(),

Error\_LBGR\_Sign = numeric(),

Error\_DCGL\_Sign = numeric(),

Error\_LBGR\_WRS = numeric(),

Error\_DCGL\_WRS = numeric(),

Error\_LBGR\_T = numeric(),

Error\_DCGL\_T = numeric()

)

# Outer loop for simulation cases for (n in 1:num\_cases) {

# Calculate the sample sizes and critical values for this case

# Outer loops for alpha, beta, and sd values for (alpha in alpha\_values) { for (beta in beta\_values) { designed\_power <- 1 - beta for (sd in sd\_values) {

for (scalar in scalar\_factors){ s\_SS <- round(sign\_sample\_size(alpha,beta,sd) \* scalar) w\_SS <- round(WRS\_sample\_size(alpha, beta, sd) \* scalar) cvs <- calculate\_cvs(alpha, s\_SS) cvw <- calculate\_cvw(alpha,(w\_SS/2), w\_SS/2) # Loop over contamination levels for (mean\_contamination in mean\_range) { # Reset counter for pass events wilcox\_pass\_events <- 0 sign\_pass\_events <- 0 t\_test\_pass\_events <- 0

# Monte Carlo simulation loop for (i in 1:num\_iterations) {

# Generate random samples and calculate the test statistic

# need to scale samples sign\_sample <- rnorm(s\_SS, mean\_contamination, sd) tss <-calculate\_sign\_statistic(sign\_sample, 1)

WRS\_sample1 <- rnorm(w\_SS, mean\_contamination, sd) WRS\_sample2 <- rnorm(w\_SS, background\_contamination, sd) tsw <- calculate\_wrs\_statistic(WRS\_sample1,WRS\_sample2)

#Perform one-sample T-test comparing the sample mean to the DCGL t\_test\_results <- t.test(sign\_sample, mu = 1)

# Check if the p-value is less than alpha level if (t\_test\_results$p.value < alpha) {

t\_test\_pass\_events <- t\_test\_pass\_events + 1

}

# Check if the test statistic exceeds the critical value if (tsw > cvw) { wilcox\_pass\_events <- wilcox\_pass\_events + 1

}

if (tss > cvs) {

sign\_pass\_events <- sign\_pass\_events + 1 }

}

# Calculate the power for this mean contamination level sign\_power <- sign\_pass\_events / num\_iterations wrs\_power <- wilcox\_pass\_events / num\_iterations t\_test\_power <- t\_test\_pass\_events / num\_iterations

# Compute the percentage errors

Error\_LBGR\_Sign <- calculate\_percent\_error(sign\_power, 1 - beta)

Error\_DCGL\_Sign <- calculate\_percent\_error(sign\_power, alpha)

Error\_LBGR\_WRS <- calculate\_percent\_error(wrs\_power, 1 - beta)

Error\_DCGL\_WRS <- calculate\_percent\_error(wrs\_power, alpha)

Error\_LBGR\_T <- calculate\_percent\_error(t\_test\_power, 1 - beta)

Error\_DCGL\_T <- calculate\_percent\_error(t\_test\_power, alpha)

# Append results to the dataframe simulation\_results <- rbind(simulation\_results, c(

n, alpha, beta, s\_SS, w\_SS, scalar, sd, mean\_contamination, tss, tsw, cvs, cv

Error\_LBGR\_Sign, Error\_DCGL\_Sign, Error\_LBGR\_WRS, Error\_DCGL\_WRS, Error\_LBGR\_T

}

}

}

}

}

}

#Set column names for the results dataframe colnames(simulation\_results) <- c(

"Case\_ID", "Alpha", "Beta", "Sample\_Size\_Sign", "Sample\_Size\_WRS", "Scalar", "SD",

"Mean\_Contamination", "TS\_Sign", "TS\_WRS","CV\_Sign", "CV\_WRS",

"Power\_Sign", "Power\_WRS", "T\_Test\_P\_Value", "T\_Test\_Power", "Error\_LBGR\_Sign", "Error\_DCG

"Error\_DCGL\_WRS", "Error\_LBGR\_T", "Error\_DCGL\_T"

)

# Save the results to an Excel file write.xlsx(simulation\_results, file = "Monte\_Carlo\_Simulation\_Results.xlsx", rowNames = FALS

# Function to calculate percentage error

calculate\_percent\_error <- function(estimated\_value, designed\_value) {

percent\_error <- 100 \* (estimated\_value - designed\_value) / designed\_value return(ifelse(estimated\_value < designed\_value, -abs(percent\_error), abs(percent\_error)))

}

# Function to calculate the WRS test statistic calculate\_wrs\_statistic <- function(background\_values,set\_mean\_values) {

adj\_background\_contamination\_values <- background\_values + 1

full\_sample <- c(adj\_background\_contamination\_values, set\_mean\_values) ranks <- rank(full\_sample) ws <- sum(ranks[1:length(adj\_background\_contamination\_values)]) return(ws)

}

calculate\_sign\_statistic <- function(samples, mean) { # Obtain the differences by subtracting each sample from DCGL differences <- mean - samples

# Discard differences that are exactly zero differences <- differences[differences != 0]

# Calculate the value of the test statistic S by counting the number of positive differenc S\_plus <- sum(differences > 0) return(S\_plus)

}

# Function to calculate the sample size for the Sign test (NS) sign\_sample\_size <- function(alpha, beta, sd) {

# Z-scores from the standard normal distribution

Z\_alpha <- qnorm(1 - alpha)

Z\_beta <- qnorm(1 - beta)

# P is the probability of observing a value less than DCGL

# assuming the survey unit median is actually at LBGR

P <- pnorm(.5 / sd)

# The formula given for NS

NS <- ((Z\_alpha + Z\_beta)^2) / (4 \* (P - 0.5)^2)

return(round(NS))

}

# Function to calculate the sample size for the Wilcoxon Rank-Sum test (NW)

WRS\_sample\_size <- function(alpha, beta, sd) {

# Z-scores from the standard normal distribution

Z\_alpha <- qnorm(1 - alpha)

Z\_beta <- qnorm(1 - beta)

# Pr is similar to P but for the WRS test

Pr <- pnorm(.5 / (sqrt(2) \* sd))

# The formula given for NW

NW <- ((Z\_alpha + Z\_beta)^2) / (3 \* (Pr - 0.5)^2) return ((NW))

}

# Function to calculate the critical value for the Sign test (CVS) calculate\_cvs <- function(alpha, sample\_size) { if(sample\_size<50){

CVS <- qbinom(1 - alpha, sample\_size, prob = 0.5)

}else{

Z\_alpha <- qnorm(1 - alpha)

CVS <- sample\_size / 2 + Z\_alpha \* sqrt(sample\_size) / 2

} return(CVS)

}

# Function to calculate the critical value for the WRS test (CVW) when m or n > 20 calculate\_cvw <- function(alpha, m, n) {

Z\_alpha <- qnorm(1 - alpha)

CVW <- (m \* (m + n + 1)) / 2 + Z\_alpha \* sqrt(m \* (m + n + 1) / 12) return(CVW)

}