**Reproduction/Analysis of ecological field surveys**

**Mitsuaki Maruta**

**St. Lawrence University**

**Abstract**

In this paper, I sought to reproduce the simulation of data and statistical analyses in the paper, “The consequences of spatial structure for the design and analysis of ecological field surveys” (Legendre et al. 2002). Even though the authors of the paper provide the general setups and methods for simulating data and analyses, they did not provide any code for their simulations. This made it challenging for me in reconstructing the simulations and analyses. The bulk of the results in the reference paper could be reproduced. However, there are some differences between the original results and the reproduced values due to the modification of methods and structures. I provided the R code used for the simulation to make them reproducible at the end.

**Introduction**

In order to investigate the relationship between biological response variables (e.g., the abundance of a species) and explanatory environmental variables (e.g., soil characteristics) in ecological field surveys, ecologists collect observations at different spatial locations. They usually depend on systematic or random sampling designs if they have no prior knowledge about the spatial structures of variables. The reference paper seeks to answer whether researchers can use the information, obtained from previous surveys, to revise the sampling design to maximize the ability to identify the relationship between the response and explanatory variables. The authors measure the frequency of type 1 error (the rejection of the null hypothesis when in fact there is no effect of the environment variable on the response variable) and estimate the power for the different combinations of sampling designs and methods of statistical analysis. Power is measured by the rate of rejection of the null hypothesis when an effect of the environment variable on the response variable is present.

For the statistical analysis, a modified t-test developed by Dutilleul for correlation coefficients is compared to a t-test on the correlation coefficient assuming independence of observations. They are compared to see if we can eliminate or control for the effect of spatial autocorrelation when it is present.

In this paper, I sought to reproduce the simulation of spatial data that would be sampled and the statistical analyses for the combinations of chosen spatial autocorrelation and sampling designs. All the simulations, illustrations, and analyses are coded in R because of the strength of existing libraries for generating spatial data and conducting statistical analyses.

**Spatial autocorrelation**

In our simulations, spatial autocorrelation is present both in the response and explanatory variables. Spatial autocorrelation is the term used to describe the presence of systematic spatial variation in a variable and the degree to which values at spatial locations are similar to each other ("Spatial Autocorrelation" 2020). When generating spatial autocorrelation, the extent of the spatial influence is determined by the range of variogram. The range is the distance beyond which observations are no longer spatially correlated ("Variogram And Spatial Autocorrelation – Aspexit" 2020). The nugget effect represents the small-scale spatial variations within the fields usually due to measurement errors from a man-made or sensor measurement (Galton 1973). The sill is the magnitude of variation of the variable while the partial sill is the variation that is spatial in nature. A higher partial sill means a stronger spatial structure. In the paper, spatial autocorrelation for the errors was generated by the following exponential covariance model defined with different values of the range, nugget, and partial sill:

where is the nugget value, is partial sill, ρ is the range, and is obtained from the distance matrix.

**Models**

The implementation of my paper is based on the model description of the original implementation. I attempted to follow the structure and the setting. However, as we do not have access to the code of the original implementation, the simulation codes were developed originally without the dependence on the reference paper. In our simulations, a response variable consists of the sum of 3 separate effects. The effects are the influence of an explanatory environmental variable (), spatial autocorrelation in the response variable (), and a spatially unstructured random error component () taking independent values for each observation i:

The environmental variable consists of a deterministic structure, a spatially autocorrelated error component, and a spatially unstructured random error:

The model is assumed to be a linear function of the ecological to the environmental variables. The linear effect is modelled by multiplying E by an effect-size parameter (transfer parameter). By substituting the environmental variable, the model for the response variable R can be written as follows:

**Methods and Modifications**

First, I generate an experimental surface in a field containing 100 by 100 units and obtain a distance matrix. Before simulating an explanatory environmental surface and a response surface, I chose the spatial locations that would be sampled. In the original paper, the authors generate an explanatory environmental surface and a response surface, then extract the explanatory variable and response variable. However, the issue with generating an entire variable surface, including unsampled sites, is that a great deal of time would be spent just on 1 simulation. Instead, I select sites that I would sample, then take only the rows and the columns of the distance matrix corresponding to those sites. Then, I simulate the response and explanatory surface on those selected sites. This process saves a lot of time as we simulate 1000 times for each sampling design. Then, the relationship between explanatory and response value is analyzed by conducting a correlation analysis and producing a probability associated with the t-statistics. Pairs of surfaces (E, R) are replicated 1000 times, and results are accumulated over all simulations of a run.

For the final statistic, the proportion of rejections of the null hypothesis was computed. For my simulation, the significance level is set to be 0.05. The null hypothesis for the test is that there is no effect of the environmental variable on the response variable. The alternative hypothesis for the test is that there is an effect of the environmental variable on the response variable. For correlation analysis, I followed the reference paper and used a regular t-test of the Pearson correlation coefficient. Dutilleul’s modified t-test was also used to examine how well the test can compensate for SA in the environmental and response variables. Modified t-test corrects the variance of the test statistics as well as the degrees of freedom in the presence of spatial autocorrelation. I used modified.ttest function from SpatialPack in R to compute the p-value.

When finding the proportion of type 1 error, I generate surfaces so that the null hypothesis is true. The transfer parameter β was set to 0 to get rid of the effect of environmental variable on the response variable. In the simulation runs for studying the power, the transfer parameter β was set to 0.3 so that there is a relationship between the environmental and response variable.

**Setup (Sampling Design, Structure, and Range)**

The reference paper used thirteen different choices for the sampling designs. Out of 13 choices, I focused on 5 sampling designs, which are random, systematic, vertical stratified, vertical transact, vertical transact with two sampling intervals. For random sampling, 100 spatial points were chosen randomly. For systematic sampling, I select every 70th point on the surface until the sample size reaches 100. The interval for systematic sampling is not specified in the reference paper. The interval of 70 was chosen so that the obtained proportion of type 1 error is close to the result in the paper. For vertically stratified sampling, the surface is stratified vertically into 2 strata. Then, we sample 50 points randomly in each stratum (100 points in total). For vertical transect sampling, I chose the column randomly in the field and selected every 2nd point within the selected column. The transect starts on row 1 and the sample size is 50. Lastly, for the vertical transect with two sampling intervals, I chose the column randomly in the field again and select the points with alternating intervals of 1 and 2 points. For example, I select the first row and second row because of the interval of 1 but skip the third row and grab the fourth row for the interval of 2. The transect started on row 1 and reached down to row 74 in the chosen column. The sample size for this sampling design is 50 as well.

Even though six types of underlying spatial structures were available in the program, I have implemented three deterministic structures (no deterministic, linear gradients, and two zones). No deterministic structure has random normal error only. Hence, is set to 0 for this structure. The second structure is linear gradients from north to south and from west to east. The lowest values are in the upper left-hand corner and the highest values are in the lower righthand corner of the map. The linear gradient structure is defined by the equation D = 0.03x + 0.03y. The third structure is two zones deterministic structure. The surface was separated into north and south potion by a discontinuity. All the values in the north potion are set to be 6 while values in the south portion are 1.

The ranges, which determines the extent of the zone of spatial influence, were {0, 20, 50} points for random, systematic, and vertical stratified sampling designs. They were {0, 4, 16} points for vertical sampling design and vertical sampling design with two intervals. I have summarized all the simulations carried out to study the rate of type 1 error and power in Table 1.

Table 1. List of all the simulations

|  |  |  |
| --- | --- | --- |
| Deterministic Structure in the environmental variable | Sampling Design | Ranges |
| No deterministic structure | Random | {0, 20, 50} |
| No deterministic structure | Systematic | {0, 20, 50} |
| No deterministic structure | Vertical stratified | {0, 20, 50} |
| No deterministic structure | Vertical | {0, 4, 16} |
| No deterministic structure | Two Vertical | {0, 4, 16} |
| Gradients in two directions | Random | {0, 20, 50} |
| Gradients in two directions | Systematic | {0, 20, 50} |
| Gradients in two directions | Vertical stratified | {0, 20, 50} |
| Gradients in two directions | Vertical | {0, 4, 16} |
| Gradients in two directions | Two Vertical | {0, 4, 16} |
| Two Zones | Random | {0, 20, 50} |
| Two Zones | Systematic | {0, 20, 50} |
| Two Zones | Vertical stratified | {0, 20, 50} |
| Two Zones | Vertical | {0, 4, 16} |
| Two Zones | Two Vertical | {0, 4, 16} |

**Results (No deterministic structure in the environmental variable)**

In this section, I present the results of my reproduction of the simulation with no deterministic structure. The results are very similar as shown in Figure 1. The rate of type 1 error is inflated when autocorrelation is present in both environmental and response variables. In other words, the test is valid when there is spatial autocorrelation in one of the variables. Dutilleul's modified t-test successfully corrects rate of type 1 error with all sampling designs.

For the power study, the simulation shows that there is not much difference in power for all sampling designs. The power is reduced for vertical and vertical sampling design with two intervals due to smaller sample size of 50. The presence of SA in the response variable reduces the power of tests as shown in Figure 2. In my simulation results, the power is reduced for modified t-test in the presence of SA in the response variable. This reduction in power was not observed in the reference paper. Overall, the similar results could be reproduced.

Chart

Description automatically generated

Graphical user interface

Description automatically generated

Figure 1. Type 1 error rates for increasing values of ranges.

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Figure 2. Power study for increasing values of ranges.

**Results (Gradients in two directions in the environmental variable)**

If a broad-scale gradient in the environmental variable is present, Figure 3 shows that the regular t-test is valid only when SA is not present in the response variable. Rate of type 1 error inflated for all other cases. Dutilleul’s modified t-test can corrects the rate of type 1 error by reacting to the deterministic structure.

In power study, power is high for ordinary t-tests when the test is valid. Vertical sampling design has a smaller power due to the smaller size of sampling design. For modified t-test, power is small in the presence of SA in the environmental and response variables. The results are similar to that in the reference paper.

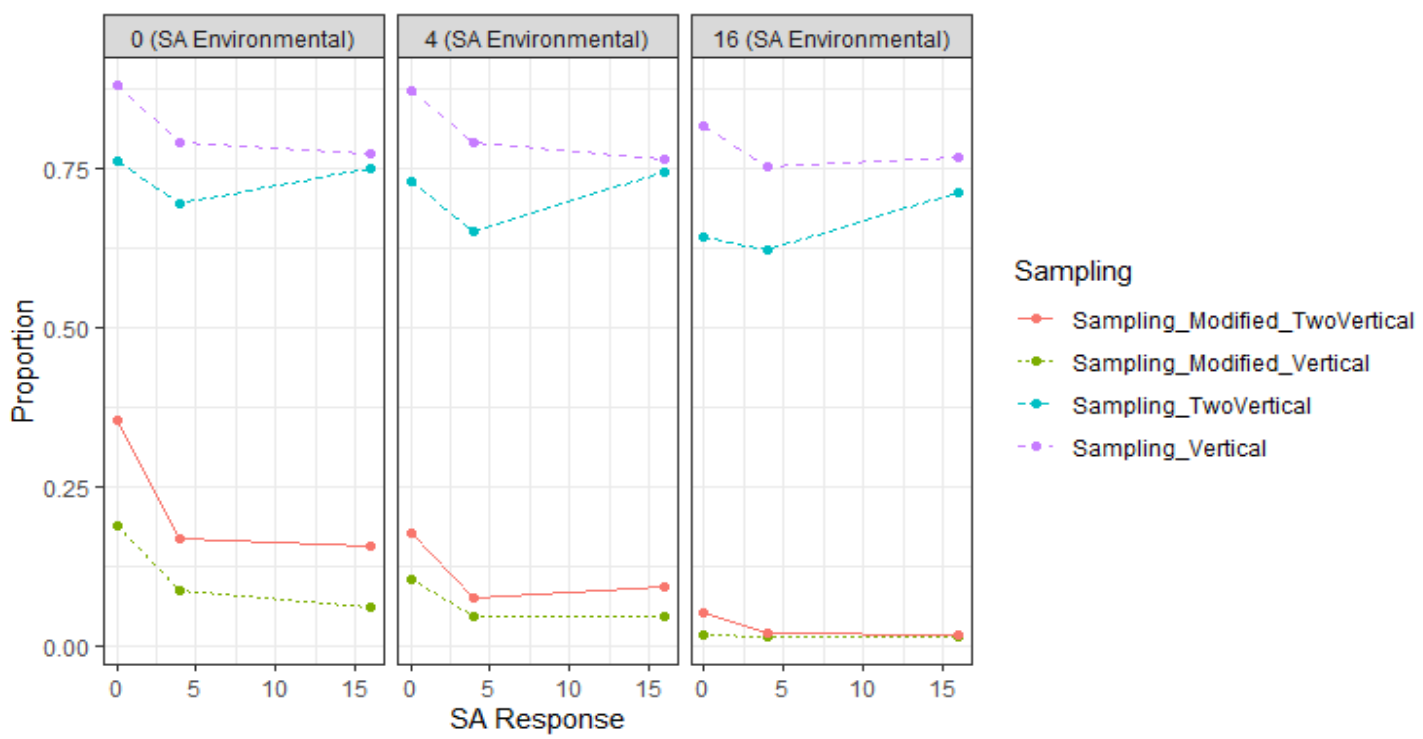
Chart

Description automatically generated

Chart, line chart

Description automatically generated

Figure 3. Type 1 error rates of the ordinary t-test and modified t-test



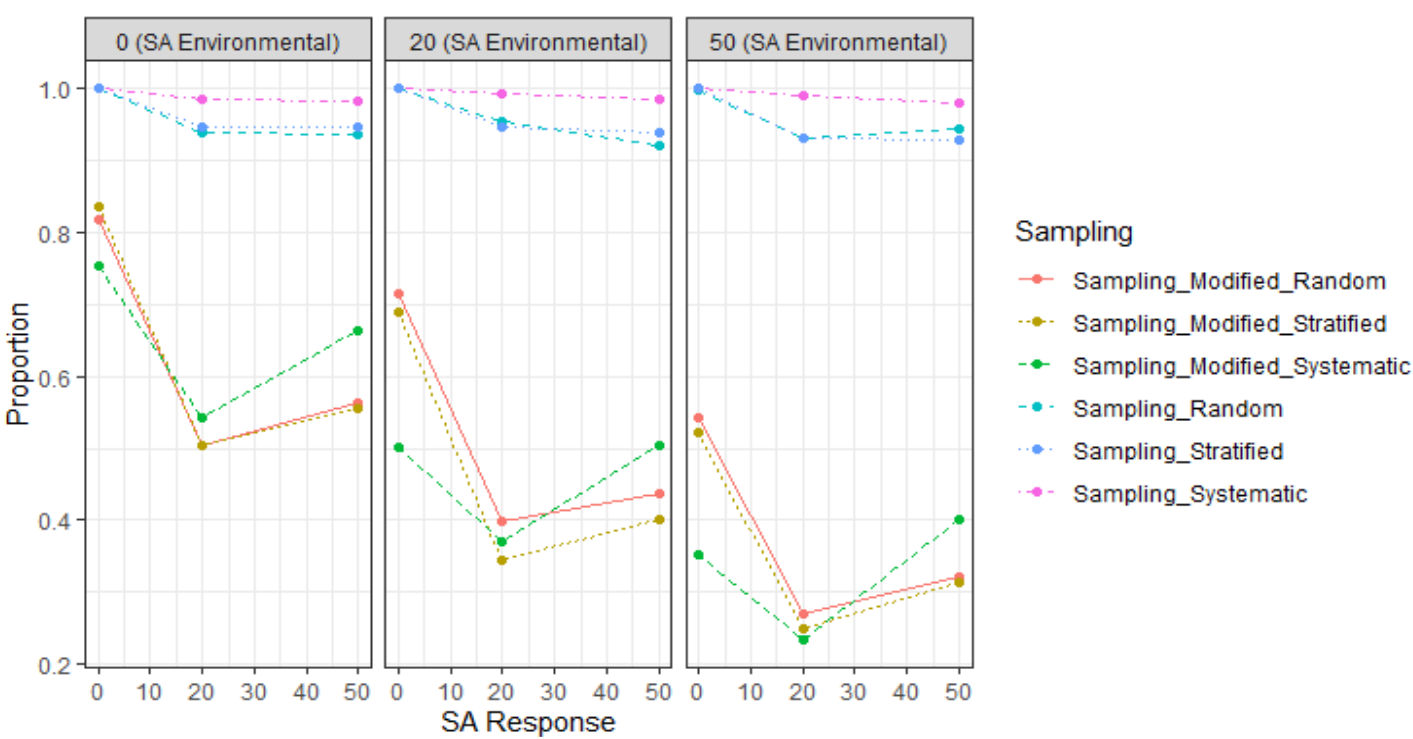


Figure 4. Power study for gradients in two directions

**Results (Two zones)**

Figure 5 shows that rate of type 1 error inflated whenever SA in response variable is present. The observations are the same as in the case of a gradient. Dutilleul’s modified t-test can corrects the rate of type 1 error by accounting for the presence of the deterministic structure.

Power study shows that equally strong power was obtained for random, systematic, and stratified sampling designs. For vertical transect with a single sampling interval, power is almost zero as shown in Figure 6. Power is also small for the vertical transect with two sampling intervals.

**Chart, line chart

Description automatically generated**

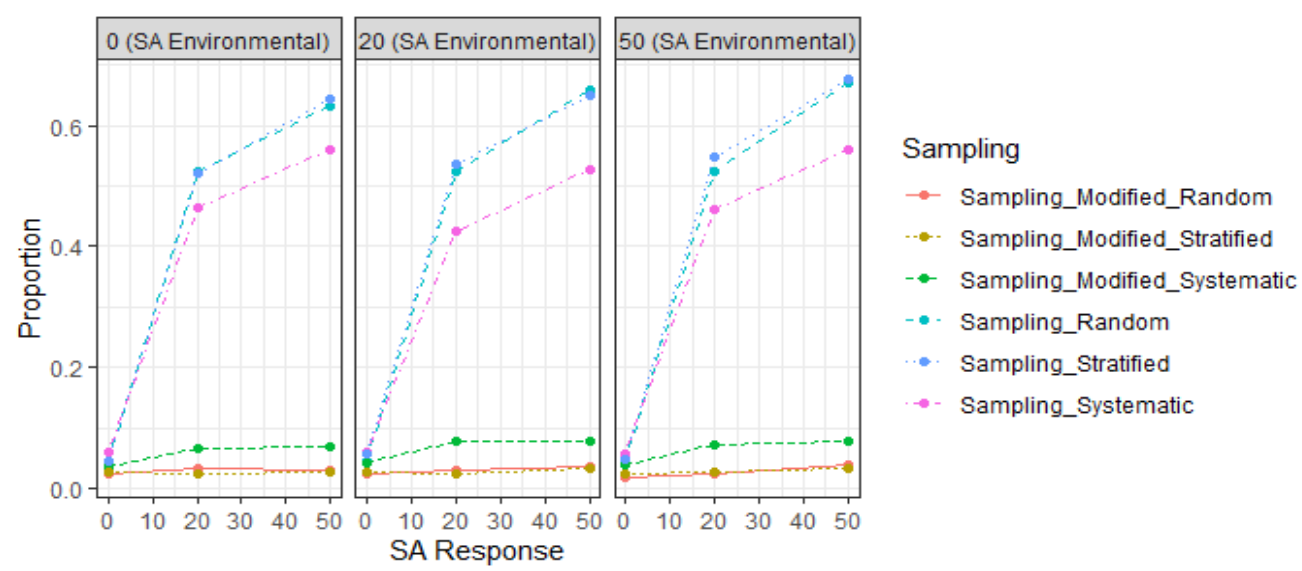


Figure 5. Type 1 error rates of the ordinary t-test and modified t-test for two zones

Chart, line chart

Description automatically generated

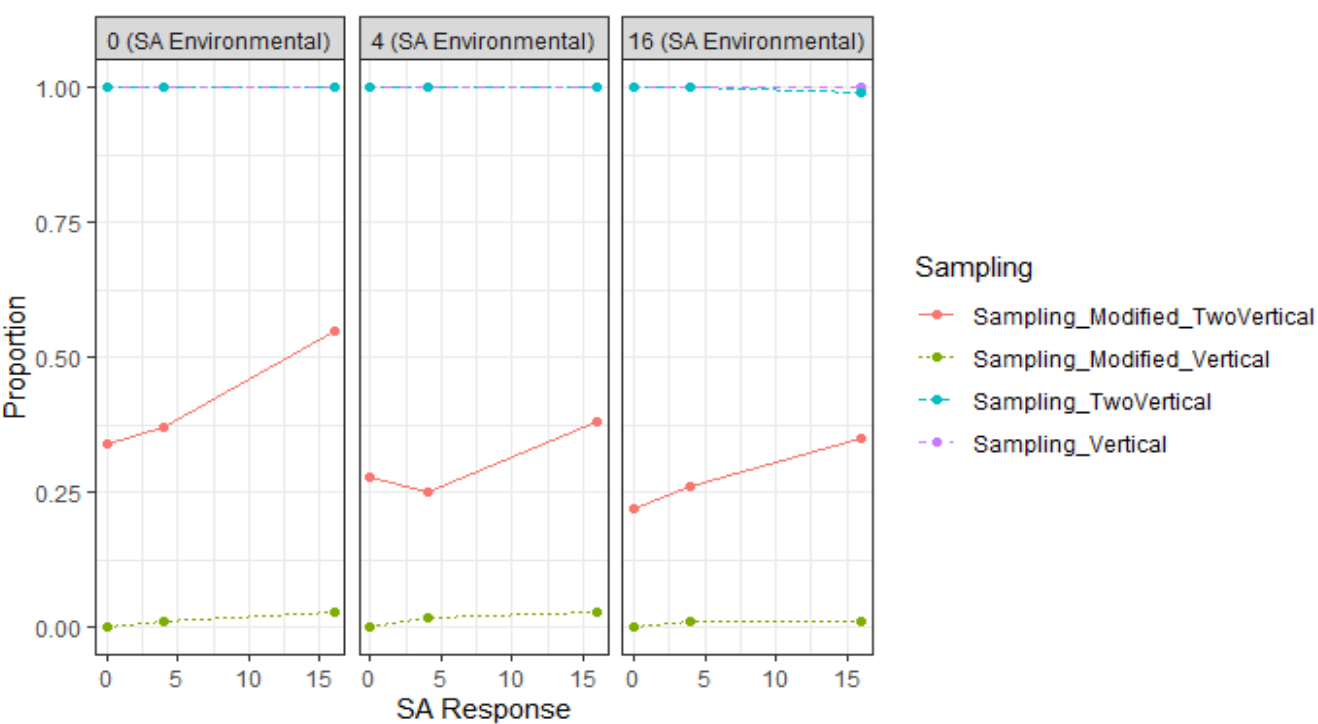


Figure 6. Power study for two zones

**Conclusion**

Overall, the results are comparable to that reported in the reference article. In the presence of spatial autocorrelation in both environmental and response variable, a normal t-test on the correlation coefficient becomes invalid. The test of significance is not disturbed when there is spatial autocorrelation in one of the variables. The tests of significance also inflate rate of type 1 error when there are a broad-scale spatial structure present and spatial autocorrelation in one of the variables. I could confirm that Dutillel’s modified t-test could take the effect of spatial autocorrelation into account and reduce the rate of type 1 error. It could also effectively correct for the presence of a broad-scale spatial structure in the errors.

**References**

1. Legendre, Pierre, Mark R. T. Dale, Marie-Josée Fortin, Jessica Gurevitch, Michael Hohn, and Donald Myers. 2002. "The Consequences Of Spatial Structure For The Design And Analysis Of Ecological Field Surveys". Ecography 25 (5): 601-615. doi:10.1034/j.1600-0587.2002.250508.x.
2. Galton, Francis. 1973. Natural Inheritance. New York: AMS Pr.
3. "Spatial Autocorrelation — R Spatial". 2020. Rspatial.Org. https://rspatial.org/raster/analysis/3-spauto.html.
4. "Variogram And Spatial Autocorrelation – Aspexit". 2020. Aspexit.Com. https://www.aspexit.com/en/variogram-and-spatial-autocorrelation/.

**Appendix A: R function used for vertical, and vertical with two intervals**

SAT1 <- function (Beta=0, rangeE = 1.5, rangeR = 2.5, psill = 0.05, Nsimulation =10, Nrow = 5, Ncol = 5, deterministic = "Random", Nsample =50){  
 # store pvalues from each sampling design  
   
 Vertical\_pvalues = c()  
 ModifiedVertical\_pvalues =c()  
 TwoVertical\_pvalues = c()  
 ModifiedTwoVertical\_pvalues = c()  
  
 # generating points grids  
 xy <- expand.grid(1:Nrow, 1:Ncol)  
 names(xy) <- c('x', 'y')  
 distance <- as.matrix(dist(xy, method = "euclidean"))  
 npoints <- nrow(xy)  
   
 # For vertical stratified sampling, this is the right part of xy  
 right\_xy <- xy[xy$x<Nrow/2+1,]  
 right\_names <- rownames(right\_xy)  
 # Left part of xy for vertical stratified sampling  
 left\_xy <- xy[xy$x>Nrow/2,]  
 left\_names <- rownames(left\_xy)  
 for(i in 1:Nsimulation){  
 col <- sample(1:Ncol, 1)  
 # Starting index when extracting in the column  
 Vertical\_site <- xy[xy$x==col, ][seq(1, Nrow, 2),]  
 Vertical\_Selectdistance <- distance[, rownames(Vertical\_site)]  
 Vertical\_Selectdistance <- Vertical\_Selectdistance[rownames(Vertical\_site), ]  
   
 Vertical\_Two <- as.numeric(rownames(Vertical\_site))  
 Vertical\_Two[Vertical\_Two< npoints/2] <- 5  
 Vertical\_Two[Vertical\_Two> npoints/2] <- 1  
  
 Vertical\_D<- switch(deterministic, "Random" = rep(0, 50),  
 "XYgradient" = 0.03\*xy[rownames(Vertical\_site), ][, 1]+0.03\*xy[rownames(Vertical\_site), ][, 2],  
 "Two Zone" = Vertical\_Two)  
 Vertical\_muE <- Vertical\_D ## mu for Explanatory variable  
 # Simulating data for explanatory variable  
 Vertical\_SAE <- mvrnorm(1, mu = Vertical\_muE, Sigma = psill\*exp(-Vertical\_Selectdistance/rangeE)+  
 diag(x = 0, nrow=50, ncol = 50))   
   
 #simulating response values  
 Vertical\_muR <- Beta\*Vertical\_SAE ## mu for response  
 Vertical\_SAR<- mvrnorm(1, mu = Vertical\_muR, Sigma = psill \* exp(-Vertical\_Selectdistance / rangeR)+  
 diag(x = 0, nrow =50, ncol = 50 ))  
   
 Vertical\_rho <- cor(Vertical\_SAE, Vertical\_SAR)  
 Vertical\_Nsample <- Ncol/2  
 Vertical\_t<-Vertical\_rho\*sqrt(Vertical\_Nsample-2)/sqrt(1-Vertical\_rho^2)  
 Vertical\_p<- (1-pt(abs(Vertical\_t), Vertical\_Nsample-2))\*2  
   
 Vertical\_pvalues[i] <-Vertical\_p  
   
 Vertical\_coords <- xy[c(names(Vertical\_SAE)),]  
 Modified\_Vertical\_p <- modified.ttest(Vertical\_SAE,  
 Vertical\_SAR, Vertical\_coords)$p.value  
   
 ModifiedVertical\_pvalues[i] <- Modified\_Vertical\_p  
  
 # Two Vertical Sampling  
 # Choosing the site  
 TwoVertical\_site <- head(xy[xy$x==col, ][-seq(0, Nrow, 3),], Nrow/2)  
 TwoVertical\_Selectdistance <- distance[, rownames(TwoVertical\_site)]  
 TwoVertical\_Selectdistance <- TwoVertical\_Selectdistance[rownames(TwoVertical\_site), ]  
 TwoVertical\_Two <- as.numeric(rownames(TwoVertical\_site))  
 TwoVertical\_Two[TwoVertical\_Two< npoints/2] <- 5  
 TwoVertical\_Two[TwoVertical\_Two> npoints/2] <- 1  
   
 TwoVertical\_D <- switch(deterministic, "Random" = rep(0, 50),  
 "XYgradient" = 0.03\*xy[rownames(TwoVertical\_site), ][,1] + 0.03\*xy[rownames(TwoVertical\_site), ][, 2],  
 "Two Zone" = TwoVertical\_Two)  
   
 TwoVertical\_muE <- TwoVertical\_D  
 TwoVertical\_SAE <- mvrnorm(1, mu = TwoVertical\_muE, Sigma =psill\*exp(-TwoVertical\_Selectdistance/rangeE)+  
 diag(x = 0, nrow=50, ncol = 50) )  
   
 #simulating response values  
 TwoVertical\_muR <- Beta\*TwoVertical\_SAE ## mu for response  
 TwoVertical\_SAR<- mvrnorm(1, mu = TwoVertical\_muR, Sigma = psill \* exp(-TwoVertical\_Selectdistance / rangeR)+  
 diag(x = 0, nrow =50, ncol = 50 ))  
 TwoVertical\_rho <- cor(TwoVertical\_SAE, TwoVertical\_SAR)  
 TwoVertical\_Nsample <- Ncol/2  
 TwoVertical\_t<-TwoVertical\_rho\*sqrt(TwoVertical\_Nsample-2)/sqrt(1-TwoVertical\_rho^2)  
 TwoVertical\_p<- (1-pt(abs(TwoVertical\_t), TwoVertical\_Nsample-2))\*2  
 TwoVertical\_pvalues[i] <- TwoVertical\_p  
 TwoVertical\_coords <- xy[c(names(TwoVertical\_SAE)),]  
 Modified\_TwoVertical\_p <- modified.ttest(TwoVertical\_SAE,  
 TwoVertical\_SAR, TwoVertical\_coords)$p.value  
 ModifiedTwoVertical\_pvalues[i] <- Modified\_TwoVertical\_p  
   
  
 }  
   
 # list of lists of p-values from each sampling desgin  
 output <- list(Vertical\_pvalues,  
 ModifiedVertical\_pvalues,  
 TwoVertical\_pvalues,   
 ModifiedTwoVertical\_pvalues,  
 rangeE,  
 rangeR)  
 # name the sampling design  
 names(output) <- c("Sampling\_Vertical",  
 "Sampling\_Modified\_Vertical",  
 "Sampling\_TwoVertical",   
 "Sampling\_ModifiedTwoVertical",  
 "rangeE",  
 "rangeR")  
 output  
}  
## In the paper, we have total of 9 combinations for variogram ranges. They use 0, 20, and 50.  
rangeEvec = c(0.0000001, 4, 16) ## Variogram ranges for E  
rangeRvec = c(0.0000001, 4, 16) ## Variaogram ranges for R  
params <- expand.grid(rangeEvec, rangeRvec)  
# All combinations of variogram ranges.  
# For the future, we are going to add sampling designs in the params vectors.  
names(params) <- c('rangeE', 'rangeR')  
Results\_total<-data.frame()## empty dataframe. Use this dataframe to add proportion later.  
for(i in 1:nrow(params)){  
 ## new dataframe for each function call.  
 Result <-SAT(Beta=0.3,  
 rangeE = params[i, 1],  
 rangeR = params[i, 2],  
 psill = 0.2,  
 Nsimulation =100,  
 Nrow = 100,  
 Ncol =100,  
 deterministic = "Two Zone",  
 Nsample = 100)  
 # create a dataframe that contains proportoin of errors or  
 # powers(when beta is not 0) from each sampling design  
 Results <- data.frame(length(which(Result$Sampling\_Vertical <0.05))/length(Result$Sampling\_Vertical),  
 length(which(Result$Sampling\_Modified\_Vertical <0.05))/length(which(!is.na(Result$Sampling\_Modified\_Vertical))),  
 length(which(Result$Sampling\_TwoVertical <0.05))/length(Result$Sampling\_TwoVertical),  
 length(which(Result$Sampling\_ModifiedTwoVertical<0.05))/length(which(!is.na(Result$Sampling\_ModifiedTwoVertical))),  
 Result$rangeE,  
 Result$rangeR  
 )  
 ## Add new dataframe to total results  
 Results\_total<-rbind(Results, Results\_total)  
 print(i)  
}  
# name the data frame  
  
names(Results\_total)[1] <- "Sampling\_Vertical"  
names(Results\_total)[2] <- "Sampling\_Modified\_Vertical"  
names(Results\_total)[3] <- "Sampling\_TwoVertical"  
names(Results\_total)[4] <- "Sampling\_Modified\_TwoVertical"  
names(Results\_total)[5] <- "rangeE"  
names(Results\_total)[6] <- "rangeR"  
Results\_total<-  
 Results\_total %>%  
 pivot\_longer(  
 cols = starts\_with("Sampling"),  
 names\_to = "Sampling",  
 values\_to = "Proportion",  
 values\_drop\_na = TRUE  
 )  
# Proportion <- (SimSat1$Proportion+SimSat2$Proportion)/2  
# SimSat2$Proportion <- Proportion  
  
variable\_names <- list("0 (SA Environmental)",   
 "4 (SA Environmental)",   
 "16 (SA Environmental)")  
  
variable\_labeller <- function(variable,value){  
 return(variable\_names[value])  
}  
  
ggplot(data=Results\_total, aes(x=rangeR, y=Proportion, group=Sampling,color = Sampling)) +  
 geom\_line(aes(linetype=Sampling))+  
 geom\_point()+  
 facet\_grid(. ~ rangeE, labeller = as\_labeller(variable\_labeller))+   
 theme\_bw()+  
 xlab("SA Response")

**Appendix B: R function used for random, systematic, and stratified sampling**

SAT <- function (Beta=0, rangeE = 1.5, rangeR = 2.5, psill = 0.05, Nsimulation =10, Nrow = 5, Ncol = 5, deterministic = "Random", Nsample =50){  
 # store pvalues from each sampling design  
 Random\_pvalues = c()  
 ModifiedRandom\_pvalues = c()  
   
 Systematic\_pvalues = c()  
 ModifiedSystematic\_pvalues =c()  
   
 Stratified\_pvalues = c()  
 ModifiedStratified\_pvalues = c()  
 # generating points grids  
 xy <- expand.grid(1:Nrow, 1:Ncol)  
 names(xy) <- c('x', 'y')  
 distance <- as.matrix(dist(xy, method = "euclidean"))  
 npoints <- nrow(xy)  
   
 # For vertical stratified sampling, this is the right part of xy  
 right\_xy <- xy[xy$x<Nrow/2+1,]  
 right\_names <- rownames(right\_xy)  
 # Left part of xy for vertical stratified sampling  
 left\_xy <- xy[xy$x>Nrow/2,]  
 left\_names <- rownames(left\_xy)  
 for(i in 1:Nsimulation){   
 RandomSites <- sample\_n(data.frame(rownames(xy)), size = 100, replace = FALSE)  
 names(RandomSites) <- "chosen sites"  
 Selectdistance <- distance[, RandomSites$`chosen sites`]  
 Selectdistance <- Selectdistance[RandomSites$`chosen sites`, ]  
   
 # For two zone  
 Random\_Two <- as.numeric(RandomSites$`chosen sites`)  
 Random\_Two[Random\_Two< npoints/2] <- 6  
 Random\_Two[Random\_Two> npoints/2] <- 1  
  
   
 # simulating explanatory valuesF  
 # defining deterministic. Random means there is no deterministic.  
 # XYgradient is linear gradients from north to south and from west to east.  
  
 D<- switch(deterministic, "Random" = rep(0, 100),  
 "XYgradient" = 0.03\*xy[RandomSites$`chosen sites`,][,1]+0.03\*xy[RandomSites$`chosen sites`,][,2],  
 "Two Zone" = Random\_Two)  
 muE <- D ## mu for Explanatory variable  
 # Simulating data for explanatory variable  
 SAE <- mvrnorm(1, mu = muE, Sigma = psill\*exp(-Selectdistance/rangeE)+  
 diag(x = 0, nrow=100, ncol = 100))   
   
 #simulating response values  
 muR <- Beta\*SAE ## mu for response  
 SAR<- mvrnorm(1, mu = muR, Sigma = psill \* exp(-Selectdistance / rangeR)+  
 diag(x = 0, nrow =100, ncol = 100 ))  
 ## Correlation between response values and explanatory values  
 Random\_rho <- cor(SAE, SAR)  
 Random\_t<-Random\_rho\*sqrt(Nsample-2)/sqrt(1-Random\_rho^2) ## t stat  
 Random\_p<- (1-pt(abs(Random\_t), Nsample-2))\*2 ##p-value  
   
 # store the pvalue  
 Random\_pvalues[i] <-Random\_p  
   
 # XY coordinates for Random Modified t-test  
 Random\_coords <- xy[c(names(SAE)),]  
 Modified\_Random\_p <- modified.ttest(SAE,  
 SAR, Random\_coords)$p.value  
   
 ModifiedRandom\_pvalues[i] <- Modified\_Random\_p  
 # Systematic sampling design  
 # Systematic\_site <- xy[seq(1, nrow(xy), npoints/Nsample),]  
 Systematic\_site <- head(xy[seq(1, nrow(xy), 70),], 100)  
 Systematic\_Selectdistance <- distance[, rownames(Systematic\_site)]  
 Systematic\_Selectdistance <- Systematic\_Selectdistance[rownames(Systematic\_site), ]  
 Systematic\_Two <- as.numeric(rownames(Systematic\_site))  
 Systematic\_Two[Systematic\_Two< npoints/2] <- 6  
 Systematic\_Two[Systematic\_Two> npoints/2] <- 1  
  
   
 # simulating explanatory valuesF  
 # defining deterministic. Random means there is no deterministic.  
 # XYgradient is linear gradients from north to south and from west to east.  
  
 Systematic\_D<- switch(deterministic, "Random" = rep(0, 100),  
 "XYgradient" = 0.05\*xy[rownames(Systematic\_site),][,1]+0.05\*xy[rownames(Systematic\_site),][,2],  
 "Two Zone" = Systematic\_Two)  
 Systematic\_muE <- Systematic\_D ## mu for Explanatory variable  
 # Simulating data for explanatory variable  
 Systematic\_SAE <- mvrnorm(1, mu = Systematic\_muE, Sigma = psill\*exp(-Systematic\_Selectdistance/rangeE)+  
 diag(x = 0, nrow=100, ncol = 100))   
   
 #simulating response values  
 Systematic\_muR <- Beta\*Systematic\_SAE ## mu for response  
 Systematic\_SAR<- mvrnorm(1, mu = Systematic\_muR, Sigma = psill \* exp(-Systematic\_Selectdistance / rangeR)+  
 diag(x = 0, nrow =100, ncol = 100 ))  
 ## Correlation between response values and explanatory values  
 Systematic\_rho <- cor(Systematic\_SAE, Systematic\_SAR)  
 Systematic\_t<-Systematic\_rho\*sqrt(Nsample-2)/sqrt(1-Systematic\_rho^2) ## t stat  
 Systematic\_p<- (1-pt(abs(Systematic\_t), Nsample-2))\*2 ##p-value  
 Systematic\_pvalues[i] <- Systematic\_p   
 Systematic\_coords <- xy[c(names(Systematic\_SAE)),]  
 Modified\_Systematic\_p <- modified.ttest(Systematic\_SAE,  
 Systematic\_SAR, Systematic\_coords)$p.value  
   
 ModifiedSystematic\_pvalues[i] <- Modified\_Systematic\_p  
  
 # Stratified sampling  
 sample\_right <-sample\_n(as.data.frame(right\_names), size = Nsample/2)  
 # Randomly sample from left part of xy  
 sample\_left <-sample\_n(as.data.frame(left\_names), size = Nsample/2)  
   
 sample\_right <- unname(sample\_right)  
 sample\_left <-unname(sample\_left)  
 names(sample\_right) <-"Position"  
 names(sample\_left) <-"Position"  
 # # combine the randomly selected by rows  
 right\_left <- rbind(sample\_right, sample\_left)  
   
 Stratified\_Selectdistance <- distance[, right\_left$Position]  
 Stratified\_Selectdistance <- Stratified\_Selectdistance[right\_left$Position, ]  
  
 Stratified\_Two <- as.numeric(right\_left$Position)  
 Stratified\_Two[Stratified\_Two< npoints/2] <- 6  
 Stratified\_Two[Stratified\_Two> npoints/2] <- 1  
   
 Stratified\_D<- switch(deterministic, "Random" = rep(0, 100),"XYgradient" = 0.03\*xy[right\_left$Position, ][,1]+0.03\*xy[right\_left$Position, ][,2], "Two Zone" =Stratified\_Two ) # defining deterministic. Random means there is no deterministic. XYgradient is linear gradients from north to south and from west to east.   
   
 Stratified\_muE <- Stratified\_D ## mu for Explanatory variable  
 Stratified\_SAE <- mvrnorm(1, Stratified\_muE, Sigma = psill\*exp(-Stratified\_Selectdistance/rangeE)+diag(x = 0, nrow= 100, ncol = 100)) # simulating data for explanatory variable   
  
#simulating response values  
 Stratified\_muR <- Beta\*Stratified\_SAE ## mu for response   
 Stratified\_SAR<- mvrnorm(1, mu = Stratified\_muR, Sigma = psill \* exp(-Stratified\_Selectdistance / rangeR)+diag(x = 0, nrow =100, ncol = 100 ))  
   
 Stratified\_rho <- cor(Stratified\_SAE, Stratified\_SAR)  
 Stratified\_t<-Stratified\_rho\*sqrt(Nsample-2)/sqrt(1-Stratified\_rho^2)  
 Stratified\_p<- (1-pt(abs(Stratified\_t), Nsample-2))\*2  
   
 Stratified\_pvalues[i] <- Stratified\_p  
   
 Stratified\_coords <- xy[right\_left$Position, ]  
 Modified\_Stratified\_p <- modified.ttest(Stratified\_SAE,  
 Stratified\_SAR, Stratified\_coords)$p.value  
   
   
 ModifiedStratified\_pvalues[i] <- Modified\_Stratified\_p  
 }  
   
 # list of lists of p-values from each sampling desgin  
 output <- list(Random\_pvalues,  
 ModifiedRandom\_pvalues,  
 Systematic\_pvalues,  
 ModifiedSystematic\_pvalues,  
 Stratified\_pvalues,  
 ModifiedStratified\_pvalues,  
 rangeE,  
 rangeR)  
 # name the sampling design  
 names(output) <- c("Sampling\_Random",  
 "Sampling\_Modified\_Random",  
 "Sampling\_Systematic",  
 "Sampling\_Modified\_Systematic",  
   
 "Sampling\_Stratified",  
 "Sampling\_ModifiedStratified",  
 "rangeE",  
 "rangeR")  
 output  
}  
## In the paper, we have total of 9 combinations for variogram ranges. They use 0, 20, and 50.  
rangeEvec = c(0.0000001, 20, 50) ## Variogram ranges for E  
rangeRvec = c(0.0000001, 20, 50) ## Variaogram ranges for R  
params <- expand.grid(rangeEvec, rangeRvec)  
# All combinations of variogram ranges.  
# For the future, we are going to add sampling designs in the params vectors.  
names(params) <- c('rangeE', 'rangeR')  
Results\_total<-data.frame()## empty dataframe. Use this dataframe to add proportion later.  
for(i in 1:nrow(params)){  
 ## new dataframe for each function call.  
 Result <-SAT(Beta=0.3,  
 rangeE = params[i, 1],  
 rangeR = params[i, 2],  
 psill = 0.3,  
 Nsimulation =1000,  
 Nrow = 100,  
 Ncol =100,  
 deterministic = "Two Zone",  
 Nsample = 100)  
 # create a dataframe that contains proportoin of errors or  
 # powers(when beta is not 0) from each sampling design  
 Results <- data.frame(length(which(Result$Sampling\_Random <0.05))/length(Result$Sampling\_Random),  
 length(which(Result$Sampling\_Modified\_Random<0.05))/length(which(!is.na(Result$Sampling\_Modified\_Random))),  
 length(which(Result$Sampling\_Systematic <0.05))/length(Result$Sampling\_Systematic),  
 length(which(Result$Sampling\_Modified\_Systematic<0.05))/length(which(!is.na(Result$Sampling\_Modified\_Systematic))),  
 length(which(Result$Sampling\_Stratified <0.05))/length(Result$Sampling\_Stratified),  
 length(which(Result$Sampling\_ModifiedStratified <0.05))/length(which(!is.na(Result$Sampling\_ModifiedStratified))),  
 Result$rangeE,  
 Result$rangeR  
 )  
 ## Add new dataframe to total results  
 Results\_total<-rbind(Results, Results\_total)  
 print(i)  
}  
# name the data frame  
names(Results\_total)[1] <- "Sampling\_Random"  
names(Results\_total)[2] <- "Sampling\_Modified\_Random"  
  
names(Results\_total)[3] <- "Sampling\_Systematic"  
names(Results\_total)[4] <- "Sampling\_Modified\_Systematic"  
  
names(Results\_total)[5] <- "Sampling\_Stratified"  
names(Results\_total)[6] <- "Sampling\_Modified\_Stratified"  
  
names(Results\_total)[7] <- "rangeE"  
names(Results\_total)[8] <- "rangeR"  
Results\_total<-  
 Results\_total %>%  
 pivot\_longer(  
 cols = starts\_with("Sampling"),  
 names\_to = "Sampling",  
 values\_to = "Proportion",  
 values\_drop\_na = TRUE  
 )  
# Proportion <- (SimSat1$Proportion+SimSat2$Proportion)/2  
# SimSat2$Proportion <- Proportion  
  
variable\_names <- list("0 (SA Environmental)",   
 "20 (SA Environmental)",   
 "50 (SA Environmental)")  
  
variable\_labeller <- function(variable,value){  
 return(variable\_names[value])  
}  
  
ggplot(data=Results\_total, aes(x=rangeR, y=Proportion, group=Sampling,color = Sampling)) +  
 geom\_line(aes(linetype=Sampling))+  
 geom\_point()+  
 facet\_grid(. ~ rangeE, labeller = as\_labeller(variable\_labeller))+   
 theme\_bw()+  
 xlab("SA Response")