

Honour Project

RENJIE PENG

B00819047

Supervised by

Professor Bruce Smith

Introduction:

Matern covariance function: in statistics, the Matern covariance (named after Swedish forestry statistician Bertil Matern) is a covariance function for spatial statistics, geographic statistics, machine learning, image analysis, and other applications of multivariate statistical analysis in metric Spaces. It is commonly used to define the statistical covariance between two measurements at points d units apart. And since the covariance only depends on the distance between the points, it is stationary. Our goal of this paper is to assess the performance of maximum likelihood estimators of the parameters of the Matern covariance function.

Defination:

The Matern covariance between two points with d distance unit is given by:

$$C_v(d) = \sigma^2 \frac{2^{(1-v)}}{\Gamma(v)} (\sqrt{2v} \frac{d}{\rho})^v K_v(\sqrt{2v} \frac{d}{\rho})$$

where Γ is the gamma function, K_v is the modified Bessel function of the second kind, and ρ and v are non-negative parameters of the covariance

Plot of 1-dimensional matern covariance function.

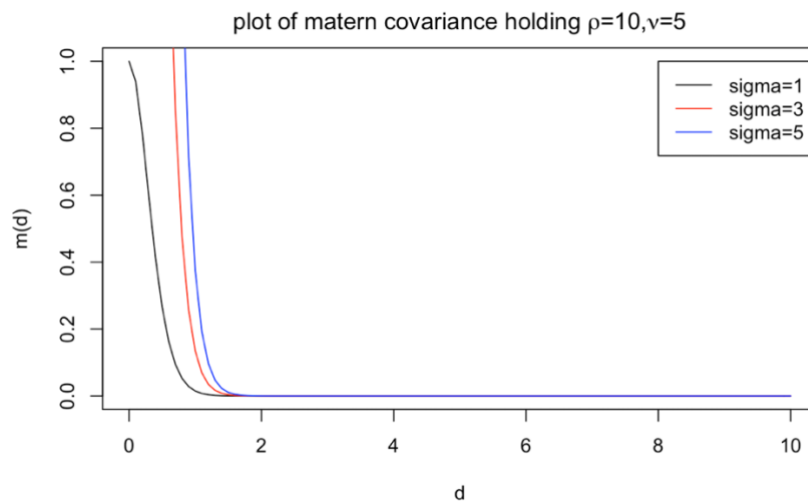
At first, we would like to see the effect of each of the three parameters in the matern covariance. So, each time, we hold the values of two parameters and only change the value of the parameter left.

Here we use the `matern.covariance()` function from the package `rSPDE` in `r` to simulate data of the Matern covariance and then use `plot()` and `lines()` function to plot.

It's worthy to mention that there are three parameters in `matern.covariance()` function corresponding to the parameters in the definition given above. One is κ which refers to the parameter ρ in the $C_v(d)$, the other two parameters are v and σ , which are as same as the definition of the $C_v(d)$.

The relative r-code is in the appendix at the end of the report.

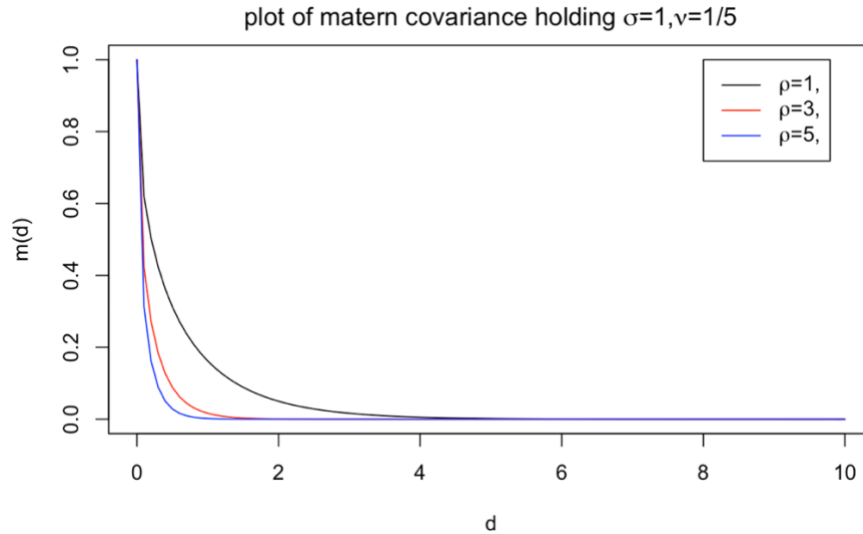
First, we fix ρ and ν , change the value of σ .



According to the function of covariance function, we can see σ looks like a constant and scales the covariance function.

Also, from the plot, we can see holding the d , ρ and ν same, when σ is larger, the value of the Matern covariance is larger at the same distance d . It is consistent with the formula. So, in the following study, to simplify the question, we just fix $\sigma = 1$.

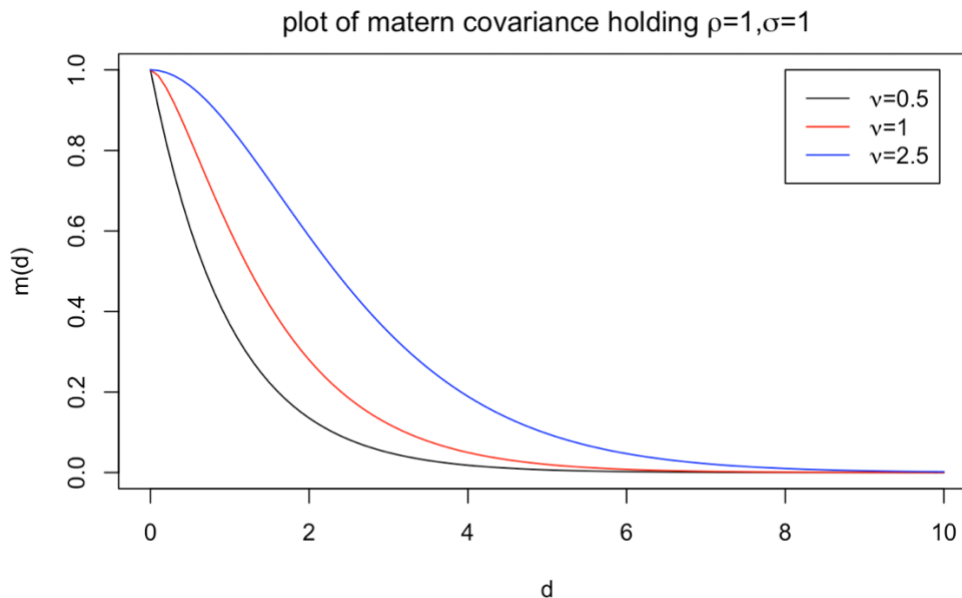
Then, we fix ν , σ and change the value of ρ .



According to the formula of the covariance function, the ρ is denominator and scales the distance d .

From the plot, we can see holding the σ and ν same, when ρ is larger, the value of the Matern covariance is smaller at the same distance d . And the covariance function becomes more concentrated near 0 with increasing ρ .

At last, we fix ρ, σ and change the value of ν .



According to the formula, we can also see when $\nu = \frac{1}{2}, C(d) = \sigma^2 \exp -\frac{d}{\rho}$. So, ν is a shape parameter of this function.

From the graph, it looks as though as ν increases with other parameters fixed, the covariance between two points at a fixed distance increase. And when $\nu = \frac{1}{2}$, the shape of Matern covariance looks like an exponential distribution.

Research question:

The usual practice, when fitting the Matern covariance function to observed data, needs to fix the parameter ν . This raises questions such as follows: What is the effect of using the wrong ν on the estimates of σ and ρ ? What is the effect on the estimated covariance function when the wrong value of ν is used? Is it possible to estimate ν in addition to σ and ρ ?

Method:

The main functions used in r:

`matern.image(fields)`: Given two sets of locations defined on a 2-d grid efficiently multiplies a cross covariance with a vector.

`sim.rf {fields}`: Simulates a stationary Gaussian random field on a regular grid with unit marginal variance.

`likfit(geoR)`: Maximum likelihood (ML) or restricted maximum likelihood (REML) parameter estimation for (transformed) Gaussian random fields.

Relative bias: $Bias = \frac{\sum_{i=1}^n (\hat{X}_i - X_i)}{n}$, where X_i is true value and \hat{X}_i is estimated value.

Relative mean square error: $MSE = \frac{\sum_{i=1}^n (\hat{X}_i - X_i)^2}{n}$, where X_i is true value and \hat{X}_i is estimated value.

it's worthy to mention that if we would like to use `likfit()` function to get the maximum likelihood estimation for matern covariance function. we need to fix parameter ν which is shown as kappa in this function. And the question we study is to find a reasonable value to plug in.

Idea:

First, we use `matern.image()` function and `sim.rf()` function from package `fields` in r to simulate a set of data with specific true parameters set by us. So, we know what exactly the true values of these parameters are.

Then, we set a sequence (0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,2,3,4,5,6,7,8,9,10,15,20) working as the potential value of the parameter ν , and then use `likfit()` function from package `geoR` to do maximum likelihood estimation using each of them as fixed value separately. After that, We pick the value with the largest `log_likihood` as the picked value of

ν in the `likfit()` function and do maximum likelihood estimation again to record the estimation for the other two parameters.

At last, we also use `likfit()` function in R to get the maximum likelihood estimations for σ, ρ, ν based on using true value of ν and wrong value of ν as fixed value separately and record the estimations.

Repeat this procedure 100 times.

To make this study more convincing, we set the true value of (ν, ρ, σ) equal $(0.5, 1, 1), (2.5, 1, 1), (0.5, 3, 1)$ and $(2.5, 3, 1)$ separately corresponding to different shape of pdf of matern covariance.

We also calculate the relative bias and relative mean squared errors to see the performances of the sequence method compared with the others.

Process:

Here are some specific codes we used to realize the idea we talked above.

As mentioned before, we use 4 different sets initial parameters to simulate data. Here, we set true value of ν equal 0.5 and true value of ρ and σ equal 1.

```
7 ▾ `` `{r}
8 library(fields)
9 sigmasq_true <- 1
10 mu_true <- 0
11
12 nu_true <- 1/2
13
14 rho_true <- 1
15 nu_wrong <- 2.5
16
17 sigmasq_est <- c()
18 nu_est <- c()
19 rho_est <-c()
20
21 sigmasq_est_mle <- c()
22 nu_est_mle <- c()
23 rho_est_mle <-c()
24
25 sigmasq_est_wrong <- c()
26 nu_est_wrong <- c()
27 rho_est_wrong <-c()
28
29 ▾ for (a in 1:100){
30
31 #simulate data
32 grid<-list( x= seq( 0,25,,20), y= seq(0,25,,20))
33 obj<-matern.image.cov( grid=grid, theta=rho_true, smoothness=nu_true,setup=TRUE)
34 look<- sim.rf( obj)
35
36 x = seq(0,25,,20)
37 y = seq(0,25,,20)
38 y1 = rep(y,20)
39 x1 = c()
40 ▾ for (i in 1:20){
41 x1 = c(x1,rep(x[i],20))
42 }
43
44 obs <- c()
45 ▾ for (i in 1:20){
46 obs <- c(obs,look[,i])
47 }
48 coords<- matrix(cbind(x1,y1),ncol = 2)
49 datalist<-list(coords,obs)
50 names(datalist)<-c('coords','obs')
51
```

```

52 #estimate with true kappa
53 ml1 <- likfit(datalist,coords = datalist$coords,data = datalist$obs,
54 cov.model = 'mat',kap = nu_true,ini = c(1,rho_true),nug = 0.2)
55
56 sigmasq_est <- c(sigmasq_est,ml1$sigmasq)
57 nu_est <- c(nu_est,ml1$kappa)
58 rho_est <- c(rho_est,ml1$phi)
59
60 # find the kappa estimate that makes the loglikelihood maximum
61 kappa = c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,2,3,4,5,6,7,8,9,10,15,20)
62 output = matrix(rep(0,2*length(kappa)),byrow = T,ncol = 2)
63 output[,1] = kappa
64
65 for(j in 1:length(kappa)){
66 output[j,2]=likfit(datalist,coords = datalist$coords,data = datalist$obs,
67 cov.model = 'mat',kap=kappa[j],ini = c(1,rho_true),nug = 0.2)$loglik
68
69 }
70 kappa_est = output[which(output[,2]==max(output[,2])),1]
71
72 ml2 <- likfit(datalist,coords = datalist$coords,data = datalist$obs,
73 cov.model = 'mat',kap = kappa_est,ini = c(1,rho_true),nug = 0.2)
74
75 sigmasq_est_mle <- c(sigmasq_est_mle,ml2$sigmasq)
76 nu_est_mle <- c(nu_est_mle,ml2$kappa)
77 rho_est_mle <- c(rho_est_mle,ml2$phi)
78
79 ml3 <- likfit(datalist,coords = datalist$coords,data = datalist$obs,
80 cov.model = 'mat',kap = nu_wrong,ini = c(1,rho_true),nug = 0.2)
81 sigmasq_est_wrong <- c( sigmasq_est_wrong ,ml3$sigmasq)
82 nu_est_wrong <- c(nu_est_wrong,ml3$kappa)
83 rho_est_wrong <- c(rho_est_wrong,ml3$phi)
84
85 output = cbind(nu_est,rho_est,sigmasq_est,nu_est_mle,rho_est_mle,
86 sigmasq_est_mle,nu_est_wrong,rho_est_wrong,sigmasq_est_wrong)
87 write(t(output),file = paste0('test',a, '.txt'),ncol= 9)
88
89 }
90
91 ...

```

Cause, the parameterization used in the functions are different with the definition of covariance we used, here is a comparison table to make it clear.

covariance function $C_v(d)$	matern.image.cov()	likfit()
ρ	θ	Don't need
ν	smoothness	kappa(fixed parameter)
σ	scale=1	Don't need

We only need to change the initial values to get the whole four sets of data.

Analysis

After collecting the data, we would like to do some simple plot first. In the following boxplots, the true parameters in the title represent the values of the parameter we used to simulate the data at first.

The parameters in the x-axis represent the estimators we got with different fixed ν in `likfit()`.

ν_0, ρ_0, σ_0 represent the estimations we got setting fixed ν equal true value.

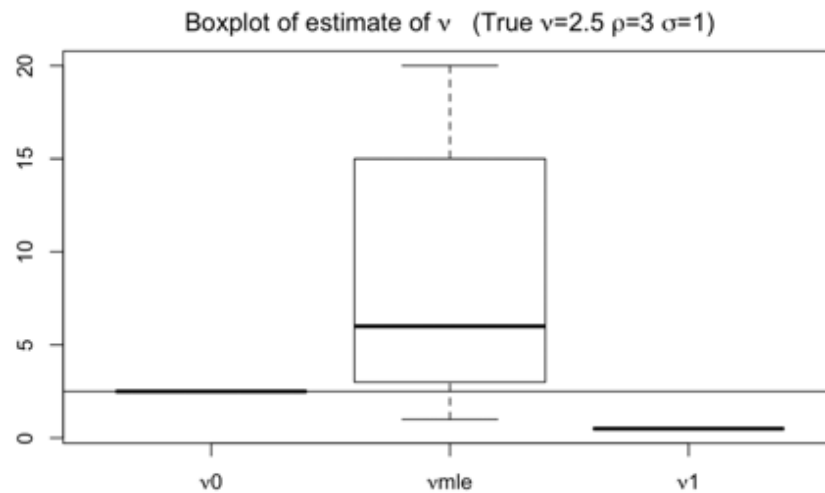
ν_1, ρ_1, σ_1 represent the estimations we got setting fixed ν equal wrong value.

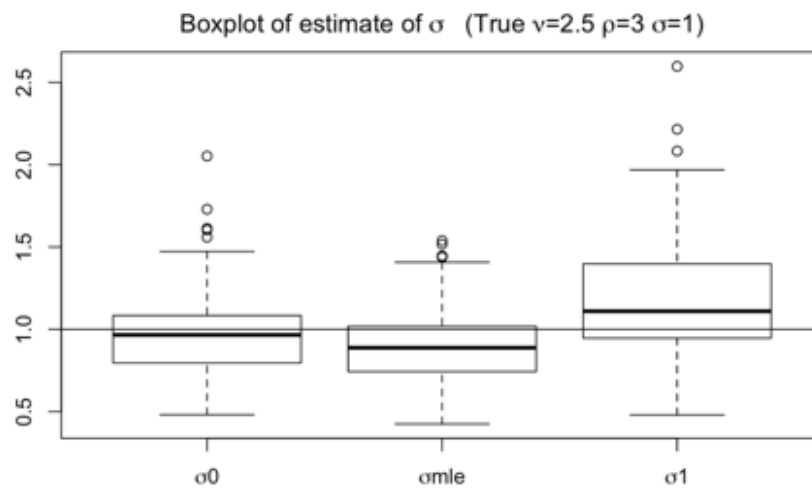
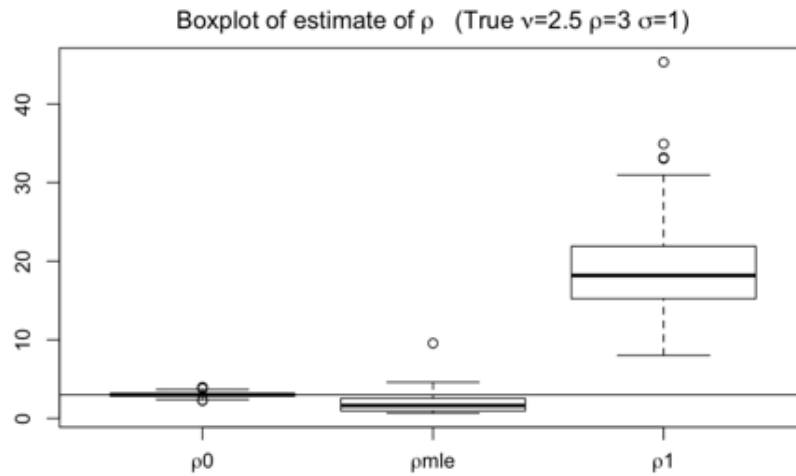
$\nu_{MLE}, \rho_{MLE}, \sigma_{MLE}$ represent the estimations we got setting fixed ν equal picked value from the sequence.

The relative r-code is in the appendix at the end of the report.

Case I:

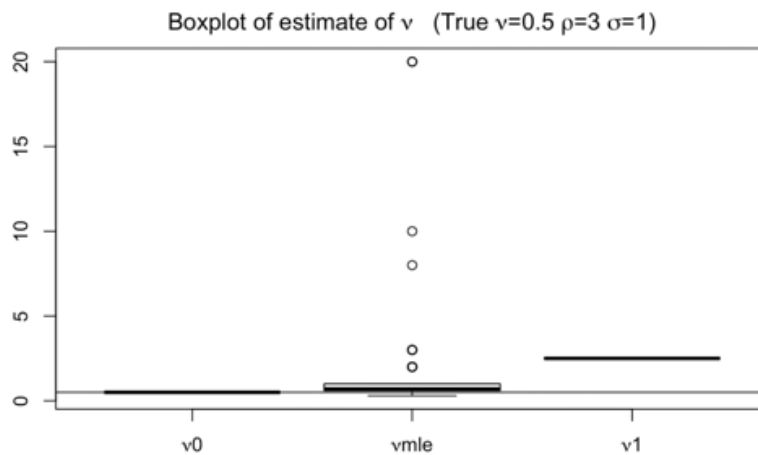
$(\nu = 2.5, \rho = 3, \sigma = 1)$

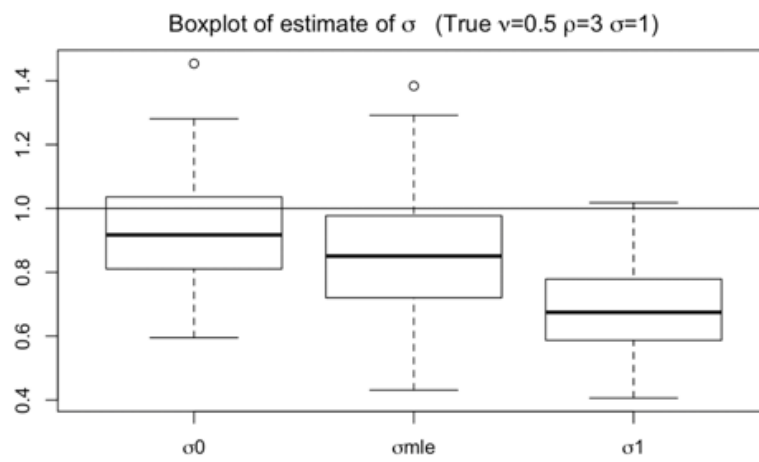
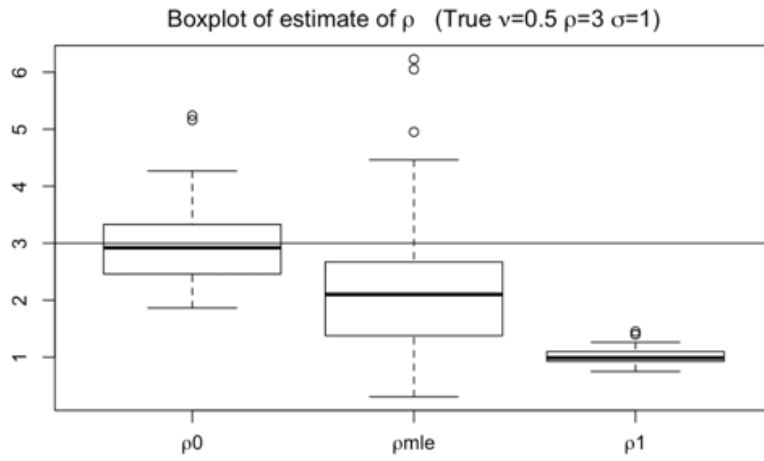




Case II:

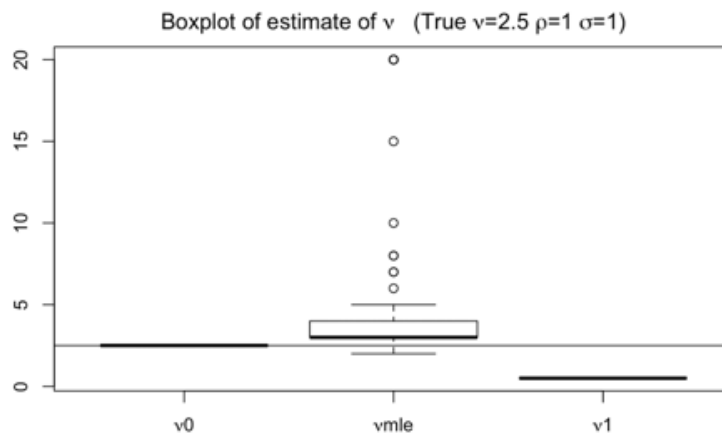
$(v = \frac{1}{2}, \rho = 3, \sigma = 1)$

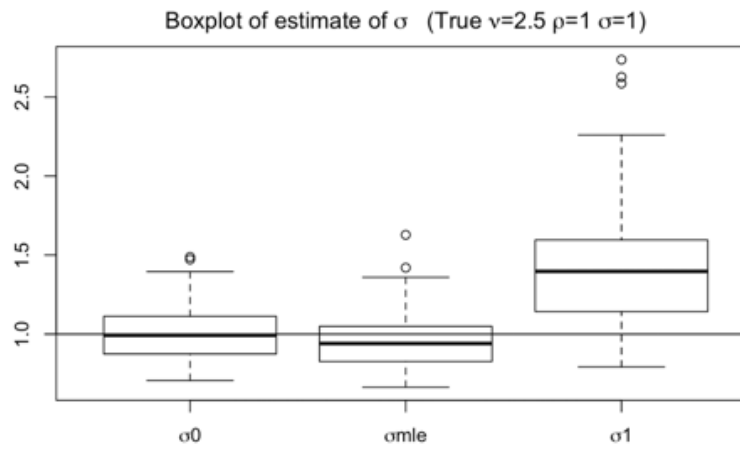
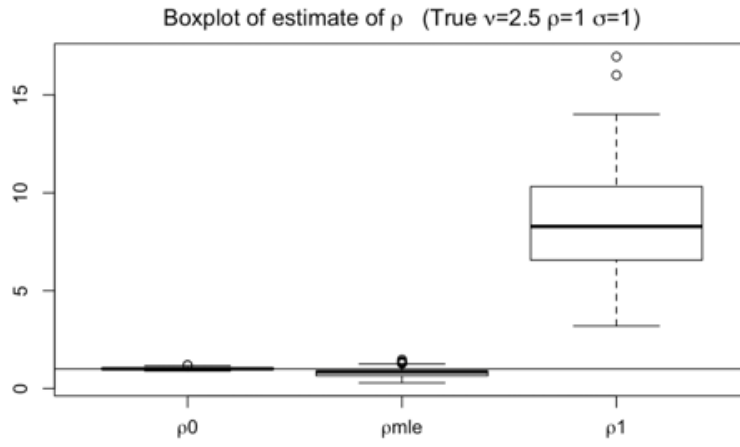




Case III

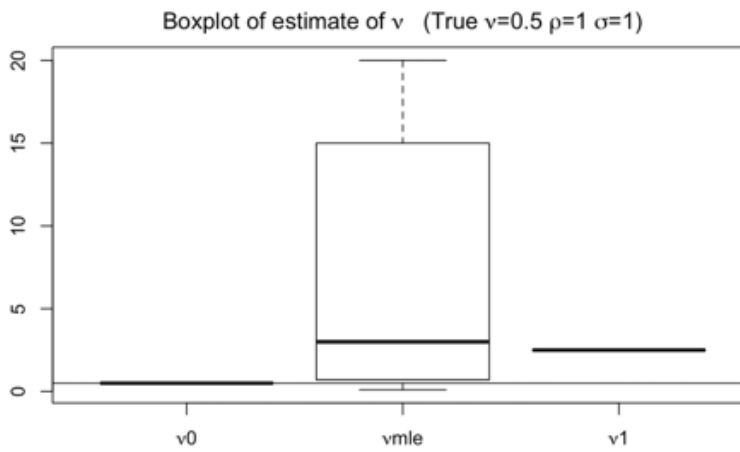
$$\left(v = \frac{5}{2}, \rho = 1, \sigma = 1\right)$$

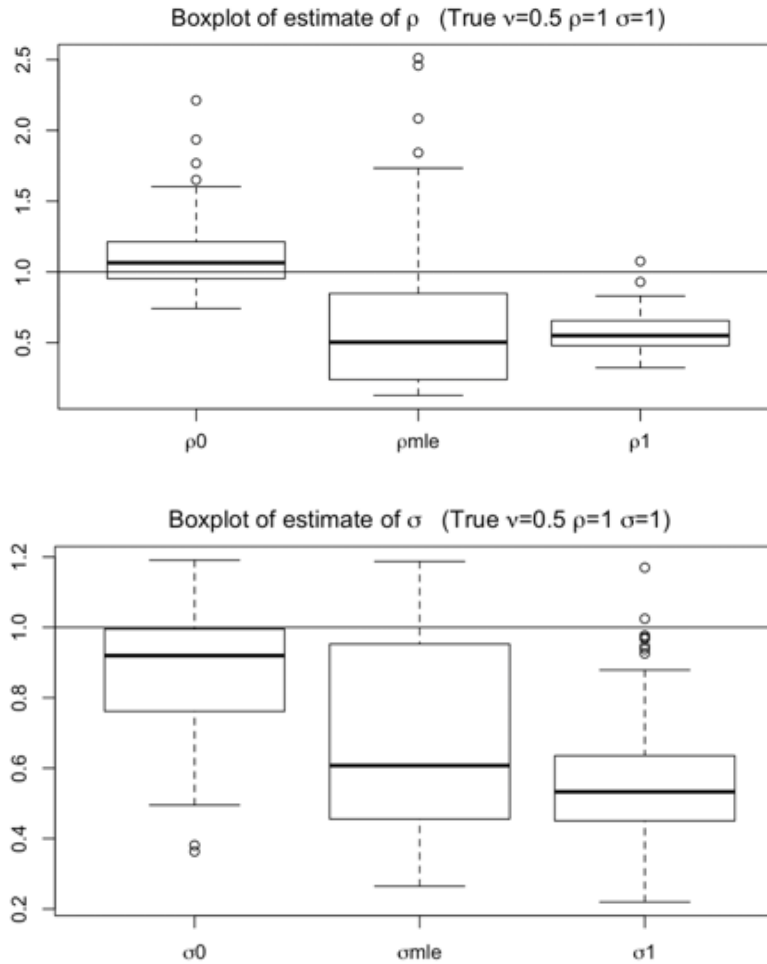




Case IV:

$$(v = \frac{1}{2}, \rho = 1, \sigma = 1)$$





From the boxplots above, we can see in the first three cases (case I ($\nu = 2.5, \rho = 3, \sigma = 1$), case II ($\nu = 0.5, \rho = 3$ and $\sigma = 1$) and case III ($\nu = 2.5, \rho = 1$ and $\sigma = 1$)) the estimates of ρ and σ when ν is fixed at the picked value have a better performance than when ν is fixed at the wrong value.

In the case IV ($\nu = \frac{1}{2}, \rho = 1, \sigma = 1$), the estimates of σ when ν is fixed at the picked value are better than when ν is fixed at the wrong value. However, the estimates of ρ when ν is fixed at the wrong value are better than when ν is fixed at the picked value in general.

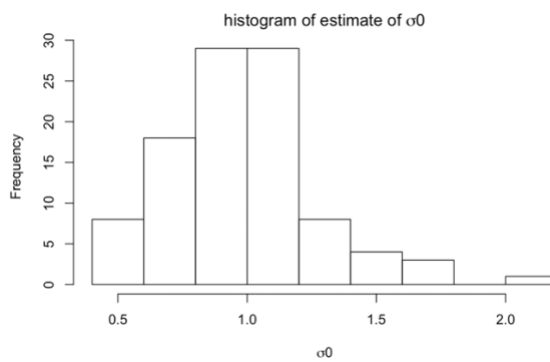
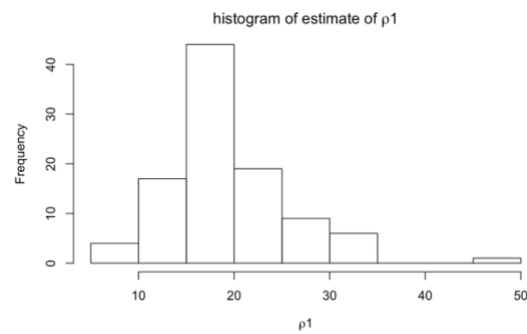
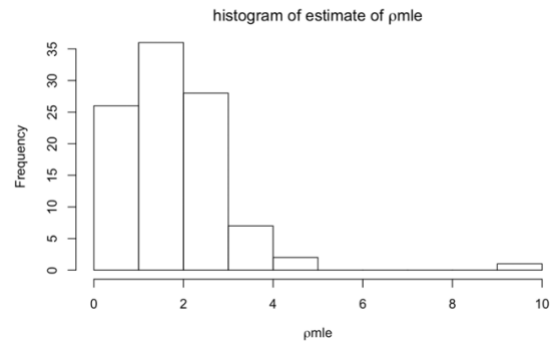
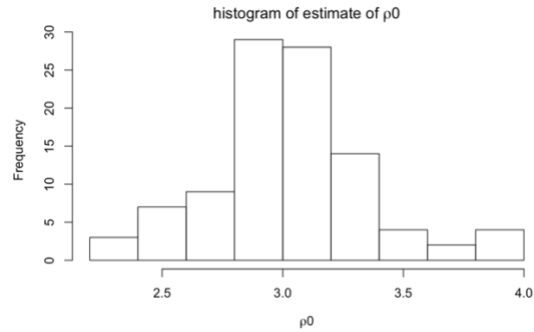
For the variance of the estimates, we can say in case II ($\nu = 0.5, \rho = 3$ and $\sigma = 1$) and case IV ($\nu = 0.5, \rho = 1, \sigma = 1$), the performance of the estimates of σ and ρ when ν is fixed at the picked value is higher than ν is fixed at the wrong value.

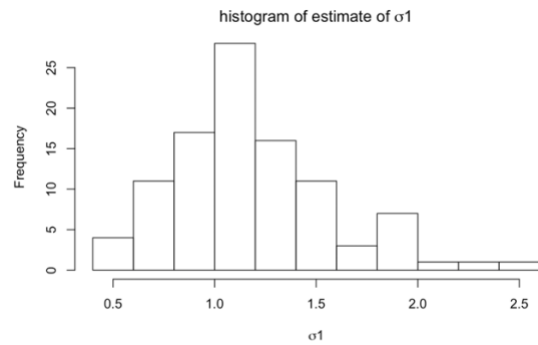
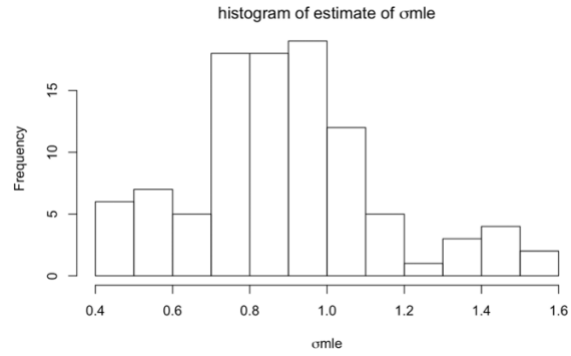
it's also worthy to point out that in case II and case III, the estimates of ν have a number of large outliers, but apart from the outliers, the majority of estimates can be better than using incorrect ν .

Histogram of the estimates with a large number of outliers.

The relative r-code is in the appendix at the end of the report.

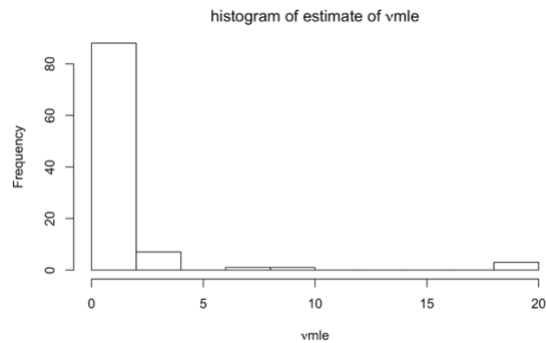
In case I ($\nu = 2.5, \rho = 1, \sigma = 1$), we make the histograms for the estimates of ρ and σ .



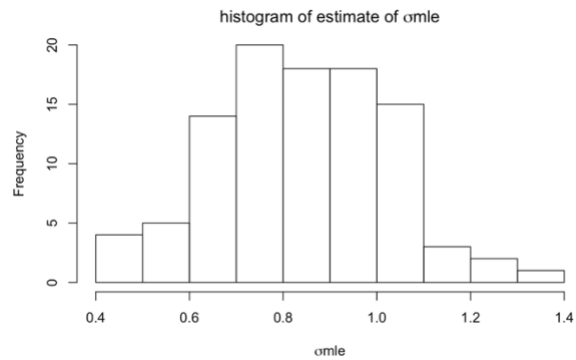
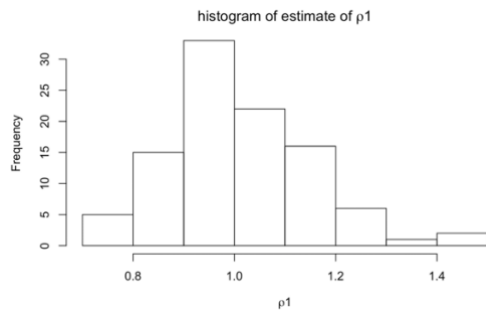
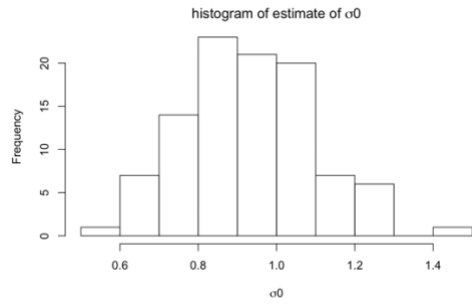
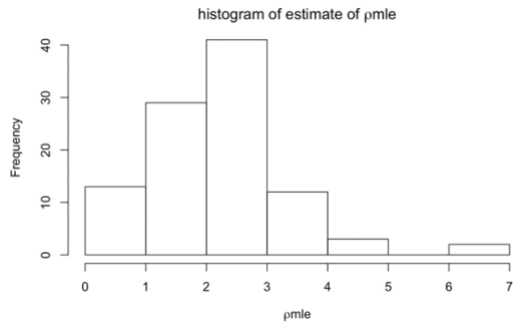
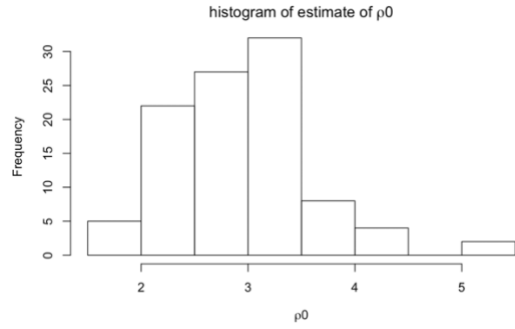


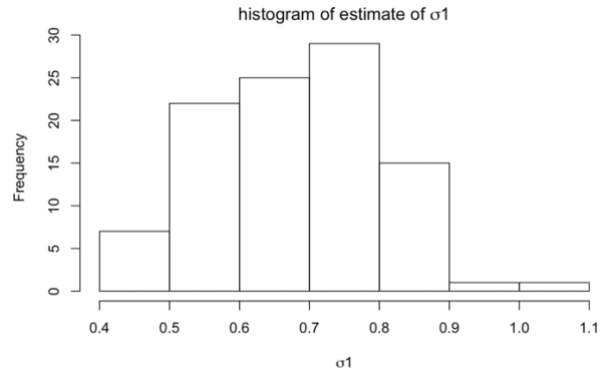
From the histograms, we can see the histogram of ρ_{mle} has a positive skewness. And the distributions of σ_{mle} look like normal.

In case II ($\nu = 0.5$, $\rho = 3$ and $\sigma = 1$), we make the histograms for the estimates of ν , ρ and σ .

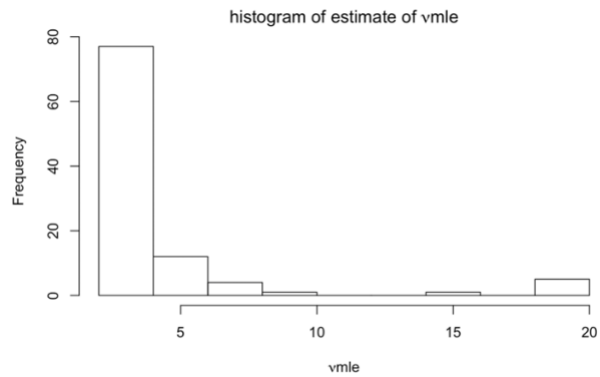
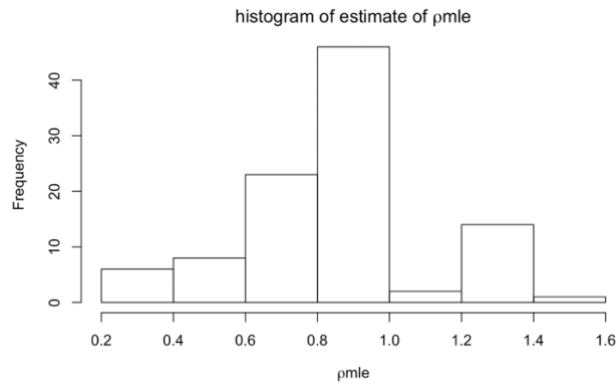
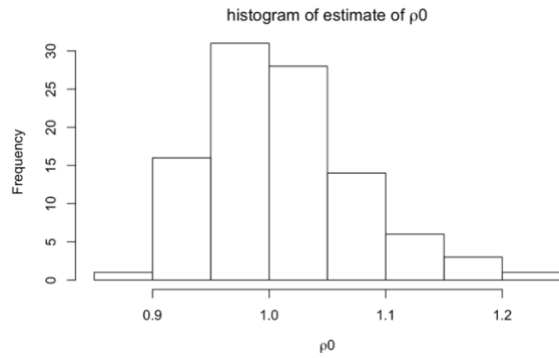


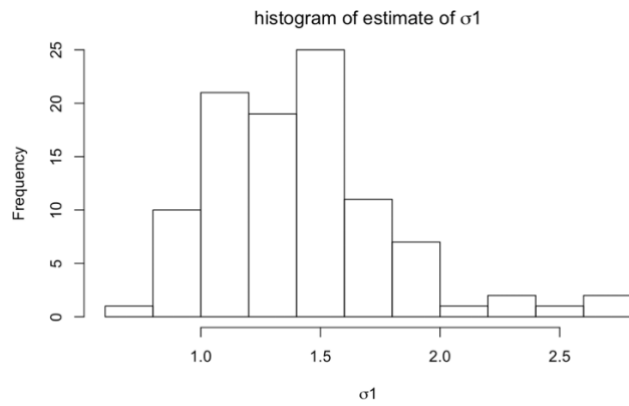
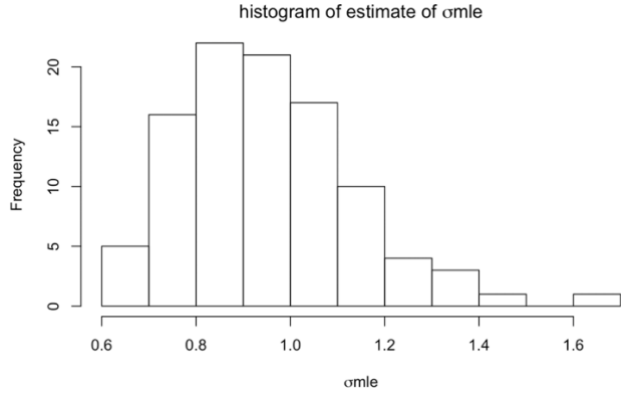
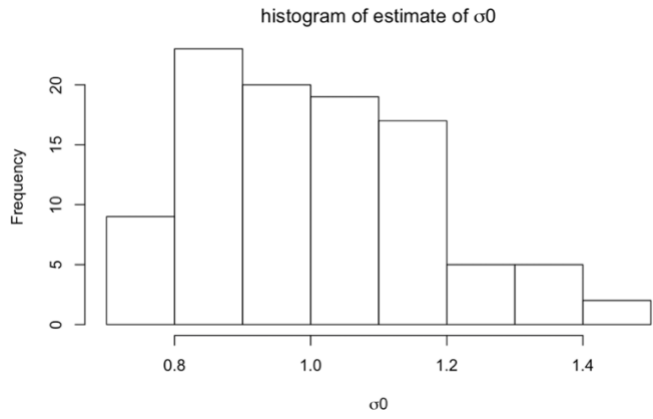
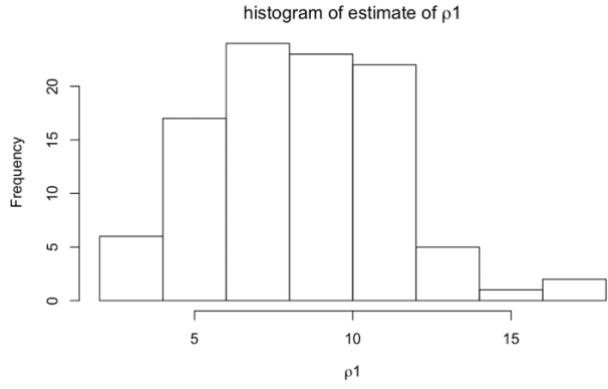
It doesn't make too much sense to make a histogram for estimates for ν when we use the true ν and the wrong ν . Cause the estimates for them are a constant as the value we use.



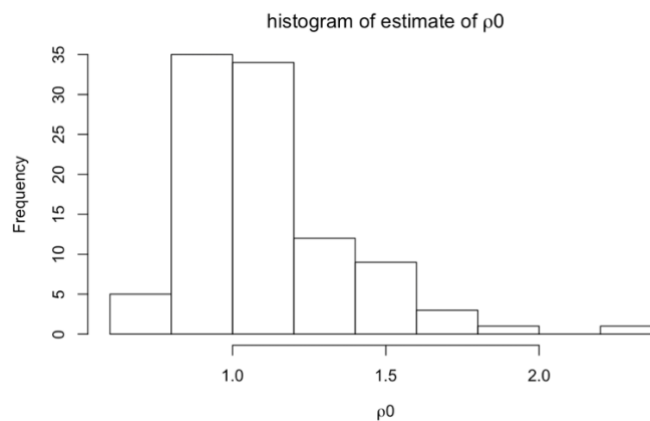
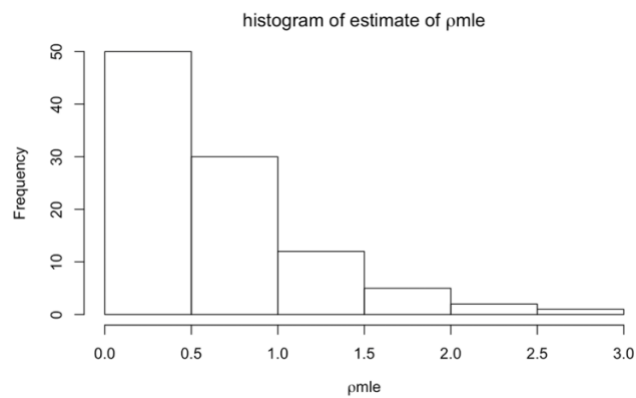
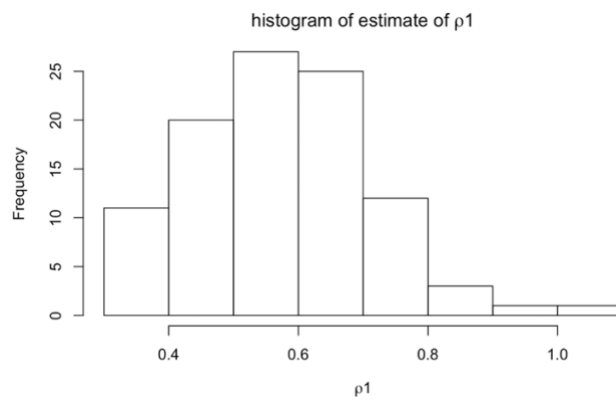
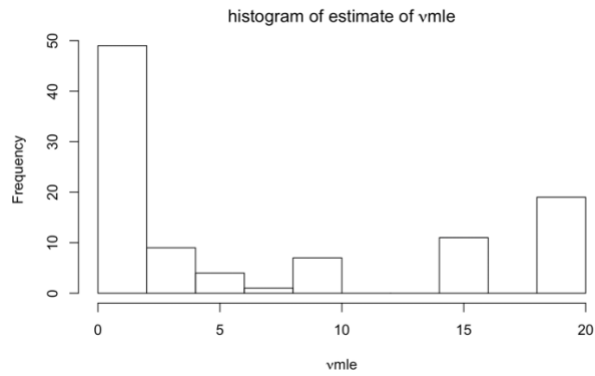


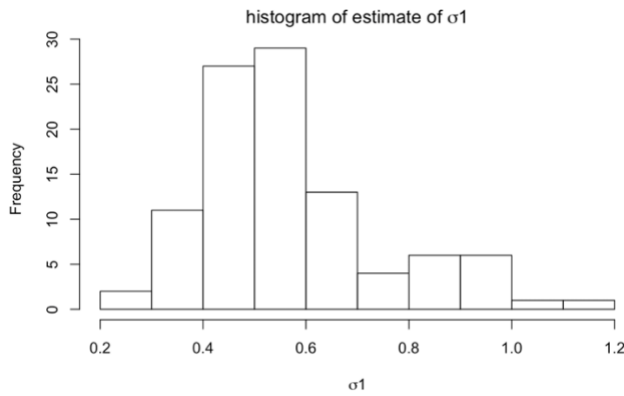
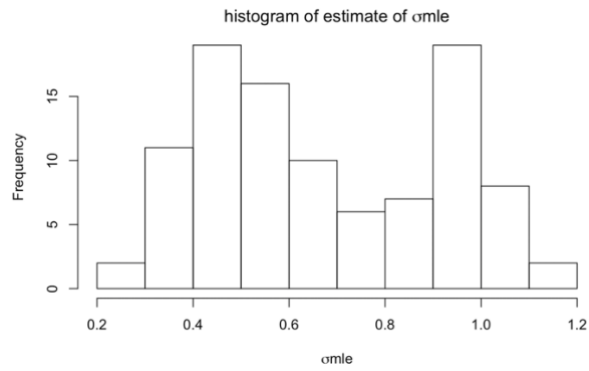
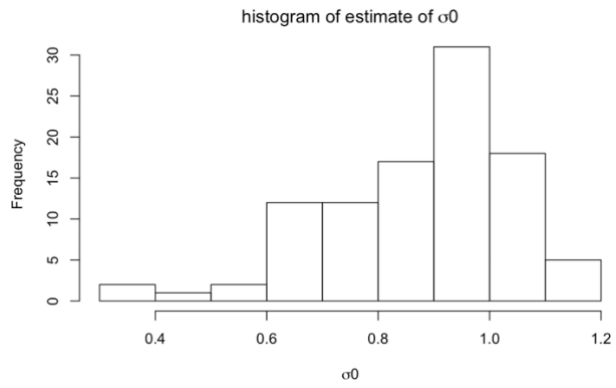
In case III ($\nu = 2.5, \rho = 1$ and $\sigma = 1$), we make the histograms for the estimates of ν, ρ and σ .





In case IV ($\nu = 0.5, \rho = 1, \sigma = 1$), we make the histograms for the estimates of ν, ρ and σ .





Relative bias and MSE

In this part, we calculate the relative bias and MLE for estimations and also summarize the output in the table.

The relative r-code is in the appendix at the end of the report.

The order of the elements in bracket in the following two tables:(true ν ,picked ν ,wrong ν)

Table of relative bias

	ν	ρ	σ
$(\nu = \frac{5}{2}, \rho = 3)$	(0, 2.516, -0.8)	(0.011, -0.3596, 5.376)	(-0.0266, -0.108, 0.178)
$(\nu = \frac{1}{2}, \rho = 3)$	(0, 2.416, 4)	(-0.0179, -0.279, 0.663)	(-0.067, -0.155, -0.318)
$(\nu = \frac{5}{2}, \rho = 1)$	(0, 0.756, -0.8)	(0.0115, -0.165, 7.34)	(0.0061, -0.042, 0.422)
$(\nu = \frac{1}{2}, \rho = 1)$	(0, 13.396, 4)	(0.118, -0.347, -0.431)	(-0.12, -0.325, -0.4327)

Table of relative MSE

	ν	ρ	σ
$(\nu = \frac{5}{2}, \rho = 3)$	(0, 13.912, 0.64)	(0.0118, 0.289, 33.26)	(0.077, 0.071, 0.187)
$(\nu = \frac{1}{2}, \rho = 3)$	(0, 54.3352, 16)	(0.0458, 0.2028, 0.441)	(0.031, 0.0573, 0.117)
$(\nu = \frac{5}{2}, \rho = 1)$	(0, 3.137, 0.64)	(0.004, 0.094, 61.934)	(0.0297, 0.034, 0.321)
$(\nu = \frac{1}{2}, \rho = 1)$	(0, 416.572, 16)	(0.078, 0.384, 0.2044)	(0.043, 0.1684, 0.2198)

The red part: relative bias/MSE of the estimates when we use picked larger than when we used wrong value.

Comment:

(1) For the ρ and σ , we can see most of relative bias and relative MSE based on the picked ν are smaller than the values based on the wrong ν , except the relative MSE when ($\nu = \frac{1}{2}, \rho = 1$). However, these two values are 0.384 and 0.2044 are also closed. And there are some outliers in the estimates of ρ when we use picked ν . The performance of majority part of estimates when we used picked values are not worse than the performance of the estimates when we used the wrong values.

(2) For the ν , half of relative bias and relative MSE based on the picked ν are bigger than the values based on the wrong ν . That's a little bit surprised. But it's reasonable, cause our sequence is from 0.1 to 20, and the true value of ν we picked are 0.5 and 2.5. if we pick larger ν , or decrease the range of sequence of ν , it's likely to become better. Also, in the real case, if we can predict the approximately range of ν through plotting the data, and set the corresponding sequence of ν , it is likely to be a better method.

Conclusion:

(1) Generally, the estimates for ν and σ are better when we used the picked value as fixed ν than when we used the wrong value. A special case is case 4 when the $\nu = 0.5$ and $\rho = 1$, there are some outliers for the estimates of ρ , and the two sets of estimates are very similar whatever we use picked ν or wrong ν .

(2) One interesting thing is when the true ν is small, even the performance of estimates when we use picked ν is better, however a large variance appears.

(3) For estimates for ν , it's hard to say something, cause the range of our sequence is too large and the value of wrong ν is very closed to the true ν . Sometimes the estimates when we use picked value is better. If we have enough time maybe, we should try $\nu = 0.5$ and 10 to probe further.

APPENDIX

The r-code for Plot of 1-dimensional matern covariance function.

```
library(rSPDE)
## Loading required package: Matrix
##
## Attaching package: 'rSPDE'
```

```

## The following object is masked from 'package:stats':
##
##      simulate

x = seq(from = 0, to = 10, length.out = 101)

#sigma
y1 = matern.covariance(x, kappa = 10, nu = 5, sigma = 1)
plot(x,y1,type = "l", ylab = "m(d)", xlab = "d",main=expression(paste('plot of
matern covariance holding ',rho,"=10,",nu,"=5' )))
y2 = matern.covariance(x, kappa = 10, nu = 5, sigma = 3)
lines(x,y2,col = 'red')
y3 = matern.covariance(x, kappa = 10, nu = 5, sigma = 5)
lines(x,y3,col = 'blue')
legend(8,1,legend = c(expression(paste(sigma,"=1," )),expression(paste(sigma,
"=3," )),
expression(paste(sigma,"=5," ))),col=c('black','red','blue'),lty=1:1)

#rho
y1 = matern.covariance(x, kappa = 1, nu = 1/5, sigma = 1)
plot(x,y1,type = "l", ylab = "m(d)", xlab = "d",main=expression(paste('plot o
f matern covariance holding ',sigma,"=1,",nu,"=1/5' )))
y2 = matern.covariance(x, kappa = 3, nu = 1/5, sigma = 1)
lines(x,y2,col = 'red')
y3 = matern.covariance(x, kappa = 5, nu = 1/5, sigma = 1)
lines(x,y3,col = 'blue')
legend(8,1,legend = c(expression(paste(rho,"=1," )),expression(paste(rho,"=3,
" )),expression(paste(rho,"=5," ))),col=c('black','red','blue'),lty=1:1)

#nu
y1 = matern.covariance(x, kappa = 1, nu = 1/2, sigma = 1)
plot(x,y1,ylab = "m(d)", xlab = "d",col = 'black',type= 'l',main=expression(p
aste('plot of matern covariance holding ',rho,"=1,",sigma,"=1' )))
y2 = matern.covariance(x, kappa = 1, nu = 1, sigma = 1)
lines(x,y2,col = 'red')
y3 = matern.covariance(x, kappa = 1, nu = 5/2, sigma = 1)
lines(x,y3,col = 'blue')
legend(8,1,legend = c(expression(paste(nu,"=0.5" )),expression(paste(nu,"=1"
)),expression(paste(nu,"=2.5" ))),col=c('black','red','blue'),lty=1:1)

```

The r-code for plot for the boxplots of the estimations for same parameter.

```

#nu = 2.5 rho = 3
nu0_1 <- data1[,1]
nu_mle_1 <- data1[,4]
nu1_1 <- data1[,7]
data_11 <- cbind(nu0_1,nu_mle_1,nu1_1)
boxplot(data_11,main= expression(paste('Boxplot of estimate of ',nu,' (True
',nu,"=2.5 ',rho,"=3 ',sigma,"=1' )))
abline(h = 2.5)

```

```

rho0_1 <- data1[,2]
rho_mle_1 <- data1[,5]
rho1_1 <- data1[,8]
data_12 <- cbind(rho0_1,rho_mle_1,rho1_1)
boxplot(data_12,main= expression(paste('Boxplot of estimate of ',rho,' (True
e ',nu,'=2.5 ',rho,'=3 ',sigma,'=1)' )))
abline(h = 3)

sigma0_1 <- data1[,3]
sigma_mle_1 <- data1[,6]
sigma1_1 <- data1[,9]
data_13 <- cbind(sigma0_1,sigma_mle_1,sigma1_1)
boxplot(data_13,main= expression(paste('Boxplot of estimate of ',sigma,' (T
rue ',nu,'=2.5 ',rho,'=3 ',sigma,'=1)' )))
abline(h = 1)

```

The r-code for making the histograms.

```

hist(sigma0,main= expression(paste('histogram of estimate of ',sigma,'0')),xlab =c(expression(
on(paste(sigma,'0'))))

hist(sigma_mle,main= expression(paste('histogram of estimate of ',sigma,'mle')),xlab =c(ex
pression(paste(sigma,'mle'))))

hist(sigma1,main= expression(paste('histogram of estimate of ',sigma,'1')),xlab =c(express
ion(paste(sigma,'1'))))

hist(rho0,main= expression(paste('histogram of estimate of ',rho,'0')),xlab =c(expression(p
aste(rho,'0'))))

hist(rho_mle,main= expression(paste('histogram of estimate of ',rho,'mle')),xlab =c(express
ion(paste(rho,'mle'))))

hist(rho1,main= expression(paste('histogram of estimate of ',rho,'1')),xlab =c(expression(
paste(rho,'1'))))

```

The r-code for calculating the MLE and relative bias.

```

#true parameter
nu1_true <- 2.5
rho1_true <- 3
sigma1_true <- 1

#relative bias
nu0_1 <- as.vector(nu0_1)
rela_bias11 <- (mean(nu0_1 - nu1_true))/nu1_true
nu_mle_1 <- as.vector(nu_mle_1)
rela_bias12 <- (mean(nu_mle_1 - nu1_true))/nu1_true
nu1_1 <- as.vector(nu1_1)

```

```

rela_bias13 <- (mean(nu1_1 - nu1_true))/nu1_true

rho0_1 <- as.vector(rho0_1)
rela_bias14 <- (mean(rho0_1 - rho1_true))/rho1_true
rho_mle_1 <- as.vector(rho_mle_1)
rela_bias15 <- (mean(rho_mle_1 - rho1_true))/rho1_true
rho1_1 <- as.vector(rho1_1)
rela_bias16 <- (mean(rho1_1 - rho1_true))/rho1_true

sigma0_1 <- as.vector(sigma0_1)
rela_bias17 <- (mean(sigma0_1 - sigma1_true))/sigma1_true
sigma_mle_1 <- as.vector(sigma_mle_1)
rela_bias18 <- (mean(sigma_mle_1 - sigma1_true))/sigma1_true
sigma1_1 <- as.vector(sigma1_1)
rela_bias19 <- (mean(sigma1_1 - sigma1_true))/sigma1_true

cat('The relative bias for nu based on true value is ',rela_bias11,'\n' )
## The relative bias for nu based on true value is 0

cat('The relative bias for nu based on best value picked is ',rela_bias12,'\n' )
## The relative bias for nu based on best value picked is 2.516

cat('The relative bias for nu based on wrong value is ',rela_bias13,'\n' )
## The relative bias for nu based on wrong value is -0.8

cat('The relative bias for rho based on true value is ',rela_bias14,'\n' )
## The relative bias for rho based on true value is 0.01068387

cat('The relative bias for rho based on best value picked is ',rela_bias15,'\n' )
## The relative bias for rho based on best value picked is -0.3595894

cat('The relative bias for rho based on wrong value is ',rela_bias16,'\n' )
## The relative bias for rho based on wrong value is 5.376361

cat('The relative bias for sigma based on true value is ',rela_bias17,'\n' )
## The relative bias for sigma based on true value is -0.02663211

cat('The relative bias for sigma based on best value picked is ',rela_bias18,'\n' )
## The relative bias for sigma based on best value picked is -0.1076705

cat('The relative bias for sigma based on wrong value is ',rela_bias19,'\n' )

```



```
## The relative bias for sigma based on wrong value is 0.1783664
```

Calculate relative MSE

```
#relative MLE
```

```
rela_MSE11 <- mean(((nu0_1/nu1_true)-1)^2)
```

```
rela_MSE12 <- mean(((nu_mle_1/nu1_true)-1)^2)
```

```
rela_MSE13 <- mean(((nu1_1/nu1_true)-1)^2)
```

```
rela_MSE14 <- mean(((rho0_1/rho1_true)-1)^2)
```

```
rela_MSE15 <- mean(((rho_mle_1/rho1_true)-1)^2)
```

```
rela_MSE16 <- mean(((rho1_1/rho1_true)-1)^2)
```

```
rela_MSE17 <- mean(((sigma0_1/sigma1_true)-1)^2)
```

```
rela_MSE18 <- mean(((sigma_mle_1/sigma1_true)-1)^2)
```

```
rela_MSE19 <- mean(((sigma1_1/sigma1_true)-1)^2)
```

```
cat('The relative MSE for nu based on true value is ',rela_MSE11,'\n' )
```

```
## The relative MSE for nu based on true value is 0
```

```
cat('The relative MSE for nu based on best value picked is ',rela_MSE12,'\n' )
```

```
## The relative MSE for nu based on best value picked is 13.912
```

```
cat('The relative MSE for nu based on wrong value is ',rela_MSE13,'\n' )
```

```
## The relative MSE for nu based on wrong value is 0.64
```

```
cat('The relative MSE for rho based on true value is ',rela_MSE14,'\n' )
```

```
## The relative MSE for rho based on true value is 0.01175599
```

```
cat('The relative MSE for rho based on best value picked is ',rela_MSE15,'\n' )
```

```
## The relative MSE for rho based on best value picked is 0.2890243
```

```
cat('The relative MSE for rho based on wrong value is ',rela_MSE16,'\n' )
```

```
## The relative MSE for rho based on wrong value is 33.26092
```

```
cat('The relative MSE for sigma based on true value is ',rela_MSE17,'\n' )
```

```
## The relative MSE for sigma based on true value is 0.07707117
```

```
cat('The relative MSE for sigma based on best value picked is ',rela_MSE18,'\n' )
```

```
## The relative MSE for sigma based on best value picked is 0.07105509
```

```
cat('The relative MSE for sigma based on wrong value is ',rela_MSE19,'\n' )  
## The relative MSE for sigma based on wrong value is 0.1865675
```