

STA5001_Assignment_3

2024-04-29

#a. Produce a sample variogram on the interval [0,1] using 20 bins.

```
library(geoR)

## Warning: package 'geoR' was built under R version 4.3.3

## -----
## Analysis of Geostatistical Data
## For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
## geoR version 1.9-4 (built on 2024-02-14) is now loaded
## -----

library(sp)

## Warning: package 'sp' was built under R version 4.3.3

data(s256i)

bin <- variog(s256i, uvec=seq(0,1,l=20))

## variog: computing omnidirectional variogram
```

#b. Fit the spherical variogram to the sample variogram by using ordinary least squares.
Use the initial values (1, 0.5) and nugget = 0.5

```
ols.n <- variofit(bin, ini = c(1,0.5), nugget=0.5, weights="equal", cov.model = "sph")

## variofit: covariance model used is spherical
## variofit: weights used: equal
## variofit: minimisation function used: optim

summary(s256i)

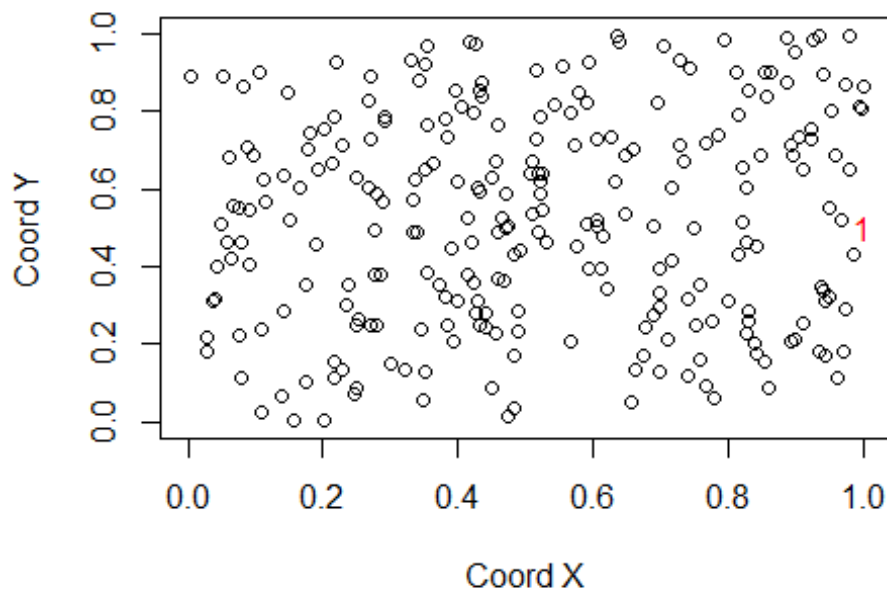
## Number of data points: 256
##
## Coordinates summary
##           x           y
## min 0.003999991 0.005934719
## max 0.999277622 0.996592965
##
## Distance summary
##           min           max
## 0.004018807 1.301652573
##
## Data summary
```

```
##      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
## -3.1835322 -0.9664729 -0.3294795 -0.4056549  0.1531377  1.3515990
```

#c.Consider the location (1, 0.5). Plot locations of the data in black and this location in red in the same image.

```
plot(s256i$coords,xlab="Coord X", ylab="Coord Y",xlim=c(0,1),ylim=c(0,1))

loci <- matrix(c(1,0.5),ncol=2)
text(loci,as.character(1), col="red")
```



#d.Use the kriging method to compute the predicted value and the variance at the point (1, 0.5). Round the answers with 4 decimal places.

```
kc <- krige.conv(s256i, locations = loci,krige = krige.control(obj.m =
ols.n))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

kc

## $predict
##      data
## 0.8541704
##
## $krige.var
```

```
## [1] 0.2525041
##
## $beta.est
##      beta
## -0.2445487
##
## $distribution
## [1] "normal"
##
## $message
## [1] "krige.conv: Kriging performed using global neighbourhood"
##
## $call
## krige.conv(geodata = s256i, locations = loci, krige = krige.control(obj.m
= ols.n))
##
## attr(,"sp.dim")
## [1] "2d"
## attr(,"prediction.locations")
## loci
## attr(,"parent.env")
## <environment: R_GlobalEnv>
## attr(,"data.locations")
## s256i$coords
## attr(,"class")
## [1] "kriging"
```

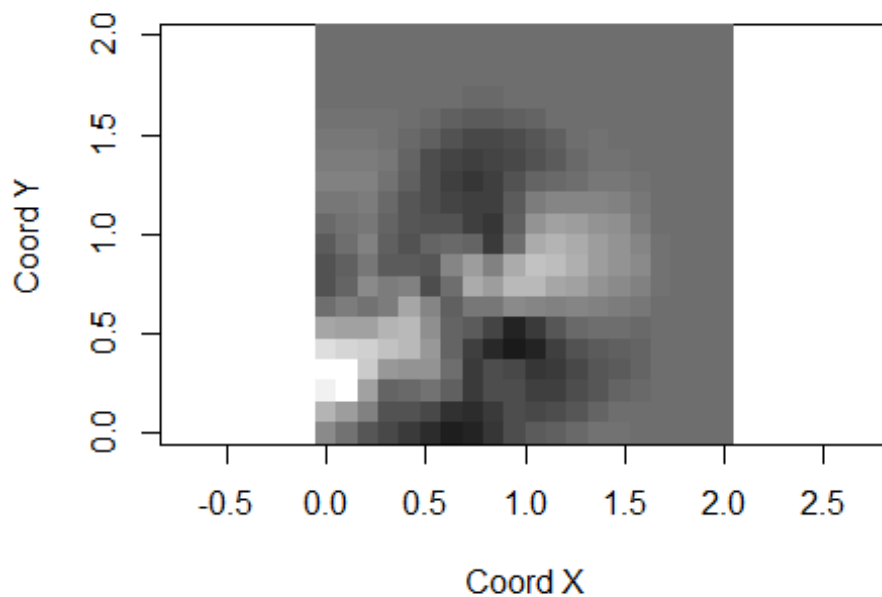
#Answer: #Predicted value:0.8542 #Variance: 0.2525

#e. Perform a prediction(kriging) on a grid covering the area [0,2]x[0,2]. Plot the result.

```
pred.grid <- expand.grid(seq(0,2, l=20), seq(0,2, l=20))
kc <- krige.conv(s256i, loc = pred.grid, krige = krige.control(obj.m =
ols.n))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

image(kc, loc = pred.grid, col=gray(seq(1,0.1,l=50)), xlab="Coord X",
ylab="Coord Y")
```



#f.Explain the obtained plot. #The grey level indicates values of s256i at the corresponding locations.

#g. To prepare your data for cross-validation, use the R commands ##where `ols.n` is the variogram fitted by the ordinary least squares method.

```
library(gstat)
```

```
## Warning: package 'gstat' was built under R version 4.3.3
```

```
a <- as.data.frame(s256i$data)
s <- SpatialPointsDataFrame(s256i$coords, a,
proj4string=CRS(proj4args=as.character(NA)), match.ID=TRUE)
v.fit <- as.vgm.variomodel(ols.n)
```

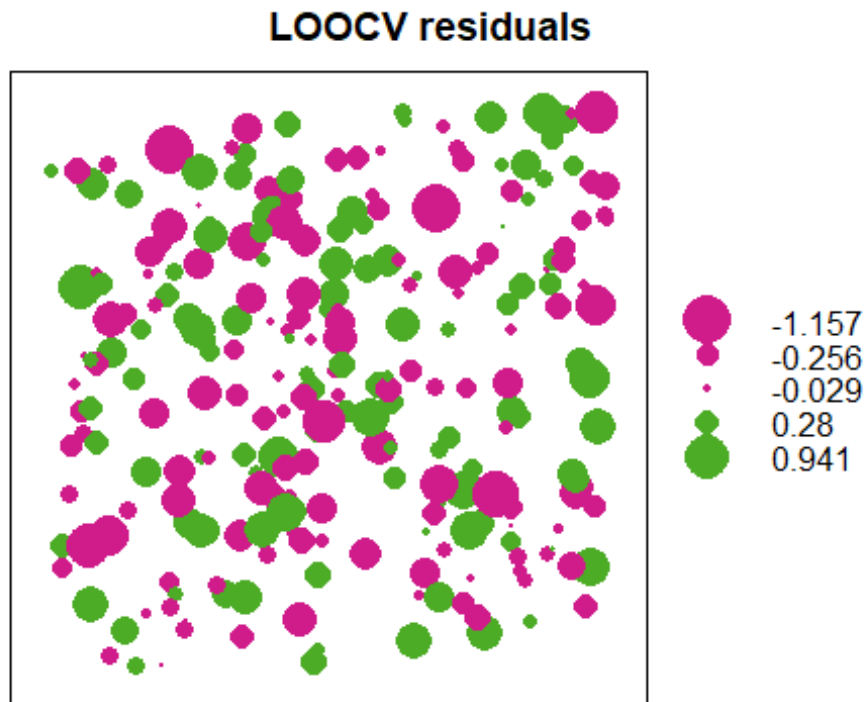
#Cross-validate your model by using leave-one-out cross-validation and a bubble plot of the result.

```
#Cross-validate your model
```

```
cvL00CV <- krige.cv(s256i.data~1, s, v.fit, nfold=nrow(s))
```

```
#Bubble plot of the result
```

```
bubble(cvL00CV, "residual", main = "L00CV residuals")
```




#h.Explain the obtained plot. #By definition: visits a data point, and predicts the value at that location by leaving out the observed value, and proceeds with the next data point.
#COME BACK LATER!

#i.Cross-validate your model by using 10-fold cross-validation and a bubble plot of the result.

```
library(gstat)
cv256 <- krige.cv(s256i.data~1, s, v.fit, nfold=10)

bubble(cv256, "residual", main = "s256i.data: 10-fold CV residuals")
```



#N-fold cross validation, as i understand it, means we partition our data in N random equal sized subsamples. A single subsample is retained as validation for testing and the remaining N-1 subsamples are used for training. #More to add!

```
m1 <- sapply(1:256, function(x) mean(krige(s256i.data~1, s[-x,], s,
v.fit)$var1.var))
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

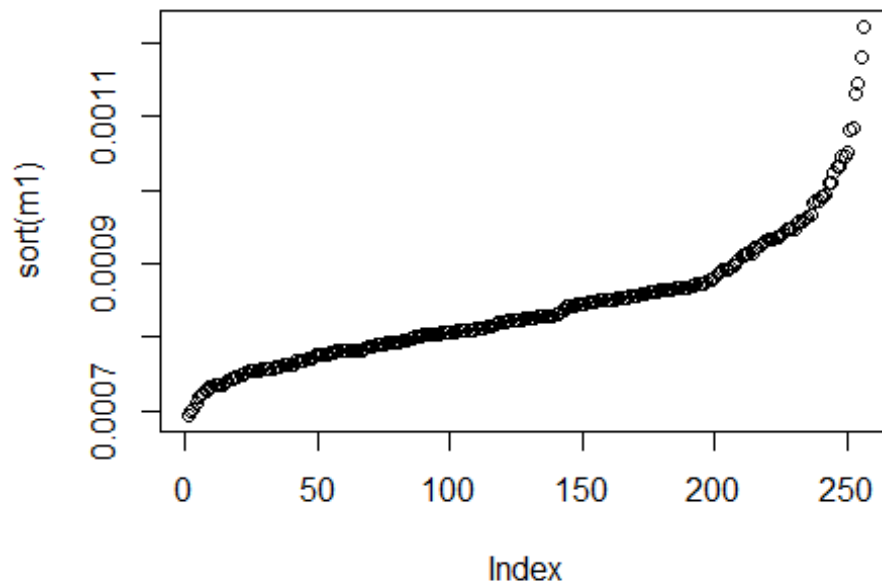

```
which(m1 %in% sort(m1)[255])
```

```
## [1] 254
```

#Answer: 254

#l. Plot the sorted mean kriging variances. Explain the obtained plot.

```
plot(sort(m1))
```



#First 5 candidate points for removal.

```
which(m1 %in% sort(m1)[1:5])
```

```
## [1] 5 11 16 79 139
```

#Explanation: #From the obtained plot, It points to observation 139 as the first candidate for removal. Where 79, 16, 11 and 5 for example are the next candidates for removal.

#m. Plot points with top and bottom 5% mean kriging variance (use different point shapes for top and bottom points).

#To compare the obtained results we plot top and bottom 5% candidate points for removal for both criteria:

```
cutoff <- 0.1
```

```
x <- seq(0,1,by=0.001)
```

```
y <- seq(0,1,by=0.001)
```

```
sgrid <- data.frame(x,y)
```

```
coordinates(sgrid) <- ~x+y
```

[illegible]

[illegible]

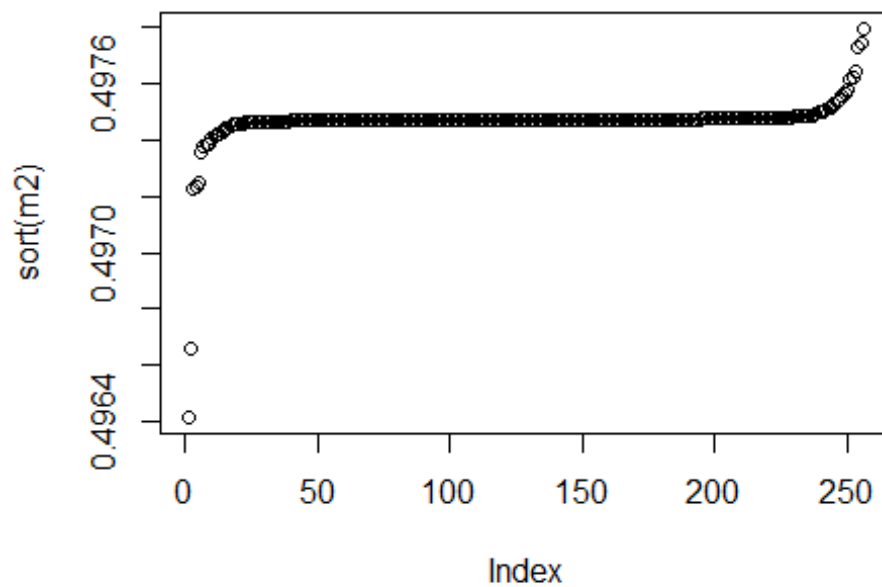
[illegible]

[illegible]

[illegible]

```
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
```

```
plot(sort(m2))
```



```
layout(matrix(1:2,1,2))
```

```
#Left for m1
```

```
plot(s256i$coords)
points(s256i$coords[m1<quantile(m1,.05),],pch=1)
points(s256i$coords[m1>quantile(m1,.95),],pch=16)
```

```
#Right for m2
```

```
plot(s256i$coords)
points(s256i$coords[m2<quantile(m2,.05),],pch=16)
points(s256i$coords[m2>quantile(m2,.95),],pch=1)
```

