Test 1 MATH 6344 SP 2019

Laha Ale

March 1, 2019

Note: This file is produced by RMarkdown , and the lines start with ## are the outputs of R codes.

Prepare Data

```
library(spBayes)
```

```
## Loading required package: coda
```

```
## Loading required package: magic
```

```
## Loading required package: abind
```

```
## Loading required package: Formula
```

```
library(classInt)
library(RColorBrewer)
library(geoR)
```

```
## ----------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-5.2.1 (built on 2016-05-02) is now loaded
## ----------------------------------------------------------
```

```
#Pollution ppm
ppm <- c(14.3, 10.05,9.91,
         10.15,12.31,10.39,
         9.61,11.10,9.16,
         9.39,12.05)

#X Coordinate
X <- c(0.35,0.80,0.41,
       0.05,0.21,0.04,
       0.55,0.11,0.49,
       0.15,0.38)
#Y Coordinate
Y <- c(0.85,0.44,0.79,
       0.08,0.58,0.92,
       0.32,0.85,0.13,
       0.08,0.83)
# Construct dataframe
ppm.df <- data.frame("ppm"=ppm,"X"=X,"Y"=Y)
ppm.df
```

```
##      ppm    X    Y
## 1  14.30 0.35 0.85
## 2  10.05 0.80 0.44
## 3   9.91 0.41 0.79
## 4  10.15 0.05 0.08
```

```
## 5  12.31 0.21 0.58
## 6  10.39 0.04 0.92
## 7   9.61 0.55 0.32
## 8  11.10 0.11 0.85
## 9   9.16 0.49 0.13
## 10  9.39 0.15 0.08
## 11 12.05 0.38 0.83
```

```
coords <- as.matrix(ppm.df[,c("X","Y")])
```

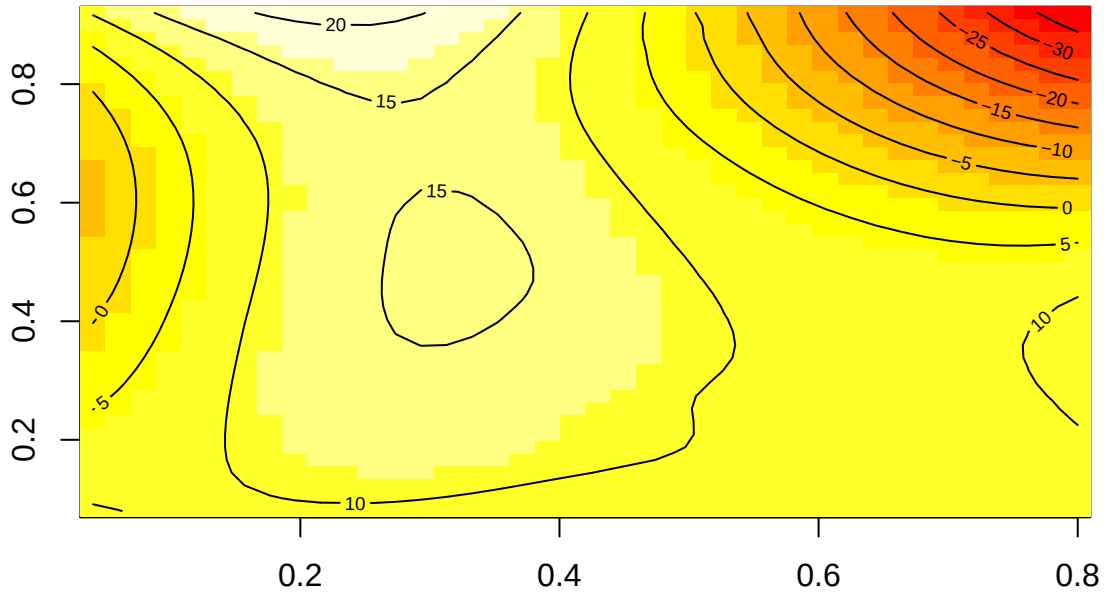
1) Show a 3d graph for these data and contour plot

method 1

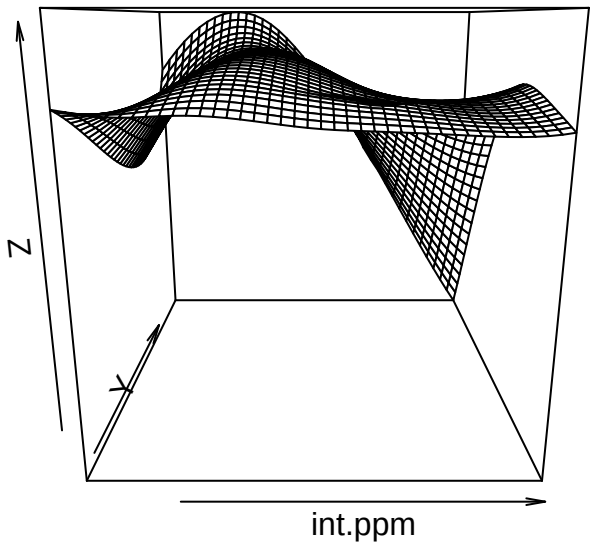
```
library(akima)
int.ppm <-interp.new(ppm.df$X,ppm.df$Y,ppm.df$ppm,extrap=TRUE)
```

```
## Warning in interp.new(ppm.df$X, ppm.df$Y, ppm.df$ppm, extrap = TRUE):
## interp.new() is deprecated, use interp()
```

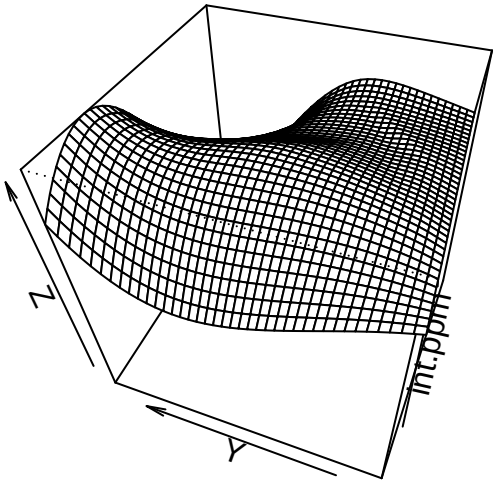
```
image(int.ppm)
contour(int.ppm, add=T)
```



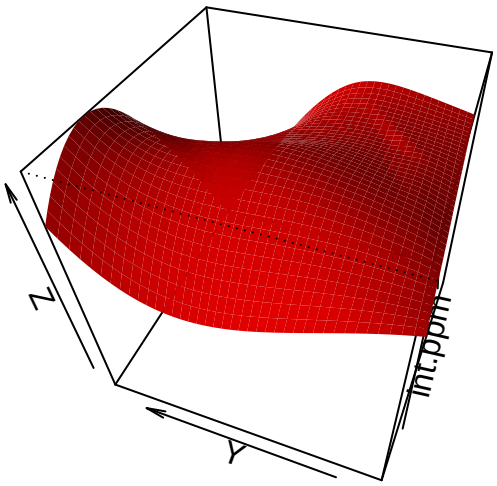
```
persp(int.ppm)
```




```
persp(int.ppm,theta = -70, phi = 45, d=1)
```




```
persp(int.ppm,theta = -70, phi = 45, d=1,col="red",shade=0.75,border=NA)
```



method 2

```
library(MBA)
library(fields) ## For using the image.plot function
```

```
## Loading required package: spam
```

```
## Loading required package: dotCall64
```

```
## Loading required package: grid
```

```
## Spam version 2.2-1 (2018-12-20) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

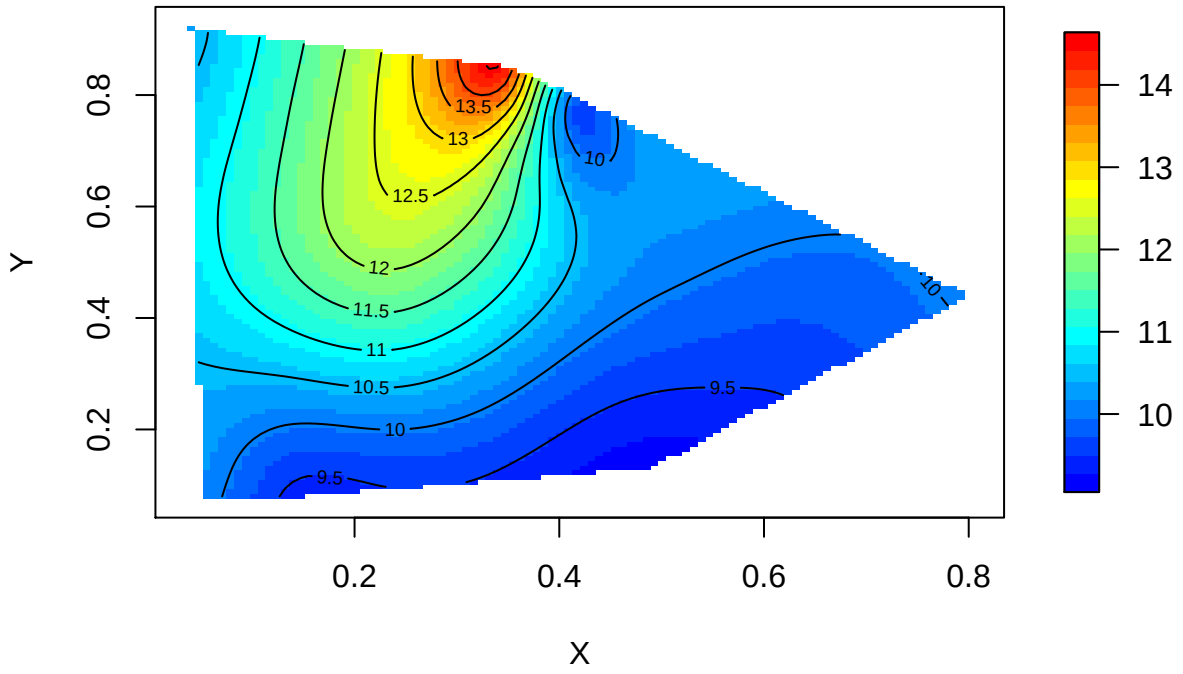
```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: maps
```

```
## See www.image.ucar.edu/~nychka/Fields for
##  a vignette and other supplements.
```

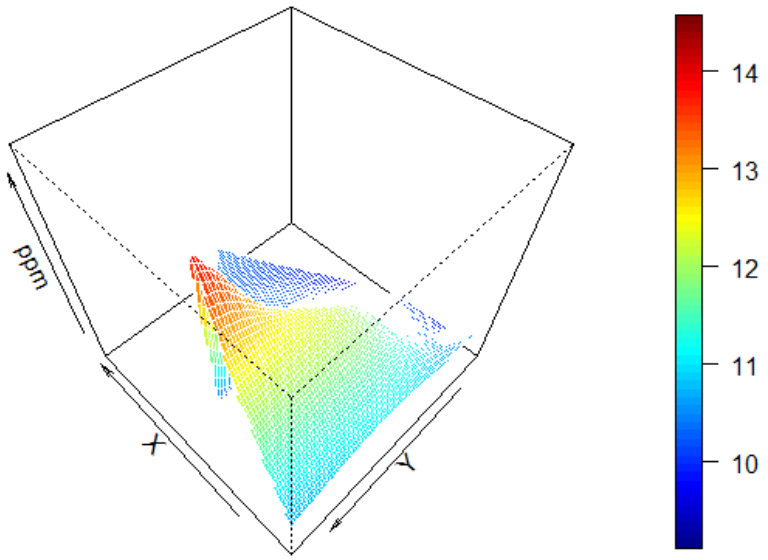
```
x.res <- 100
y.res <- 100
col.br <- colorRampPalette(c("blue", "cyan", "yellow", "red"))
col.pal <- col.br(5)


surf <- mba.surf(cbind(coords, ppm),
                 no.X = x.res, no.Y = y.res, h = 5,
                 m = 2, extend = FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           xlab = "X", ylab = "Y", col = col.br(25))
contour(surf, add=T)
```




```
library(rgl)
col <- rbind(0, cbind(matrix(drape.color(surf[[3]],
                                         col = col.br(25)),
                             x.res - 1, y.res - 1), 0))
surface3d(surf[[1]], surf[[2]], surf[[3]], col = col)
axes3d()

title3d(main = "PPM", xlab = "X", ylab = "Y", zlab = "ppm")
drape.plot(surf[[1]], surf[[2]], surf[[3]], col = col.br(150),
           theta = 225, phi = 50, border = FALSE, add.legend = FALSE,
           xlab = "X", ylab = "Y", zlab = "ppm")
image.plot(zlim = range(surf[[3]], na.rm = TRUE),
           legend.only = TRUE, horizontal = FALSE)
```



Although the second method is fancier than the previous one, it would take a long time to plot the 3D plot.

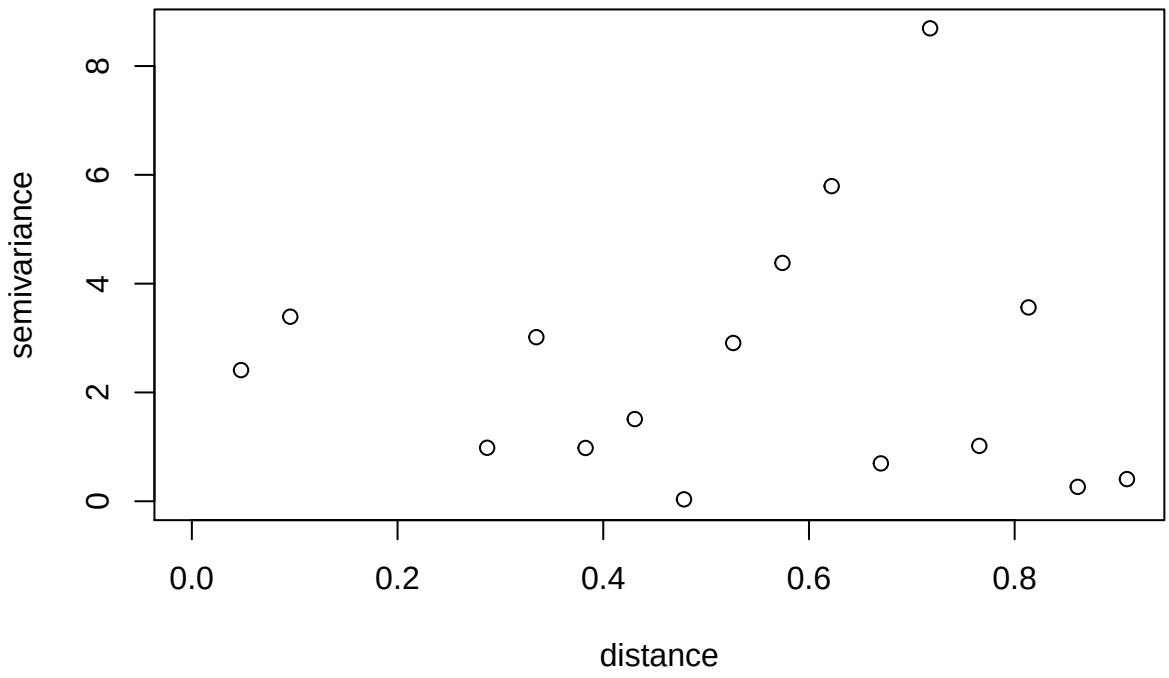
2) Fit a variogram model of your choice, use any kind of estimation(MLE, OLS, MOM) or if the results still do not agree with the empirical variogram you can also use eyefit.

```
## Load some libraries we will use

bins = 20
max.dist <- max(iDist(coords))
ppm.vario <- variog(coords = coords, data = ppm,
                        uvec = (seq(0, max.dist, length = bins)))
```

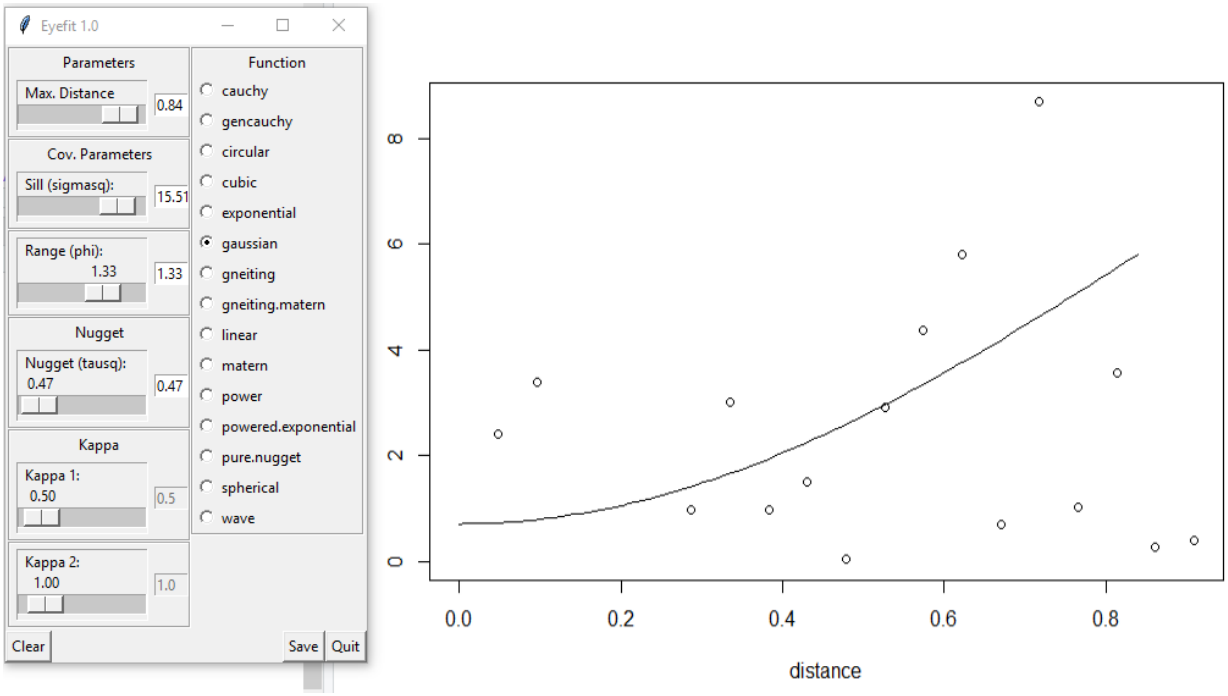
```
## variog: computing omnidirectional variogram
```

```
plot(ppm.vario)
```



As we can see from above the data is not really organized and it hardly use MLE, OLS, MOM to fit. Therefore, let's try to useing eyefit.

```
eyefit(ppm.vario,silent=TRUE)
```



We will choose data given by above eyefit. Sill $\sigma^2 = 15.51$, Range $\Phi = 1.33$ and Nugget $\tau^2 = 0.47$. I know those parameters and models are not reasonably fit the data. The data seems not suitable for spatial statistics analysis.

```
## variofit: covariance model used is gaussian
## variofit: weights used: npairs
## variofit: minimisation function used: optim

## variofit: model parameters estimated by WLS (weighted least squares):
## covariance model is: gaussian
## parameter estimates:
##   tausq sigmasq     phi
##  2.2530 14.8725  5.6778
## Practical Range with cor=0.05 for asymptotic range: 9.827172
##
## variofit: minimised weighted sum of squares = 194.1483
```