Homework 6

Laha Ale

April 22, 2019

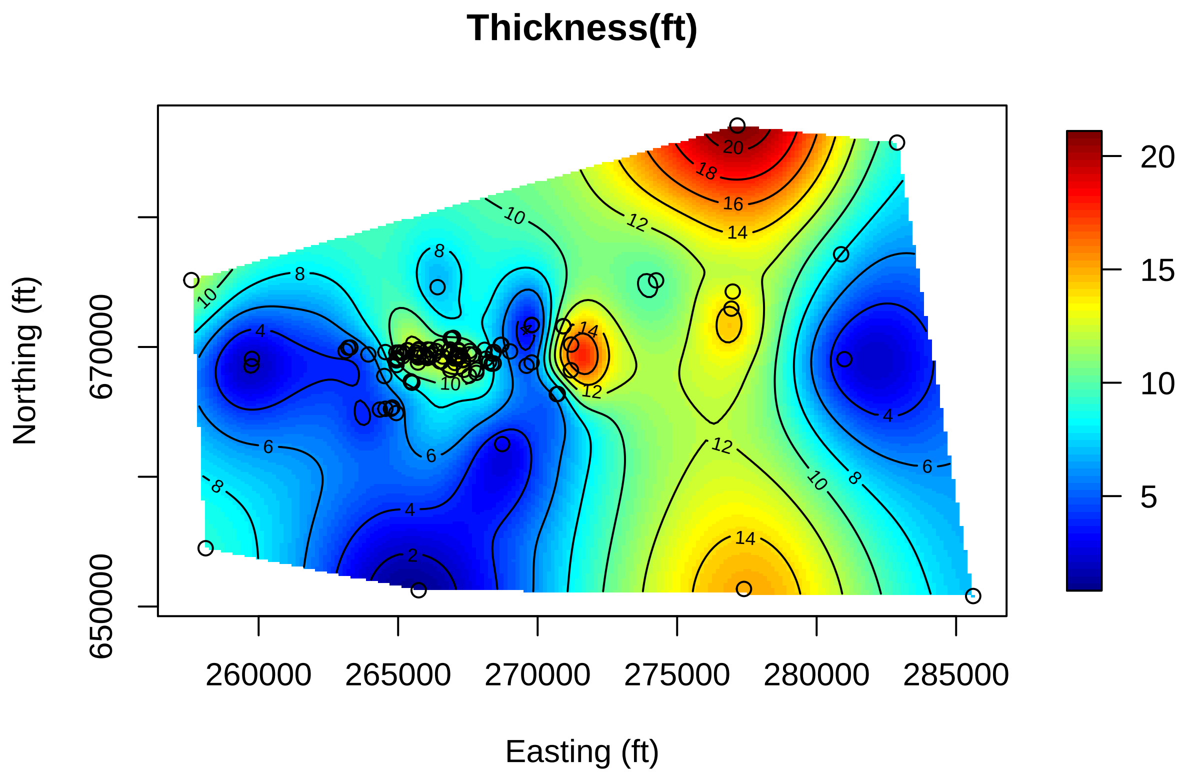
Note: This file is produced by RMarkdown , and the lines start with ## are the outputs of R codes.

Excercise 6

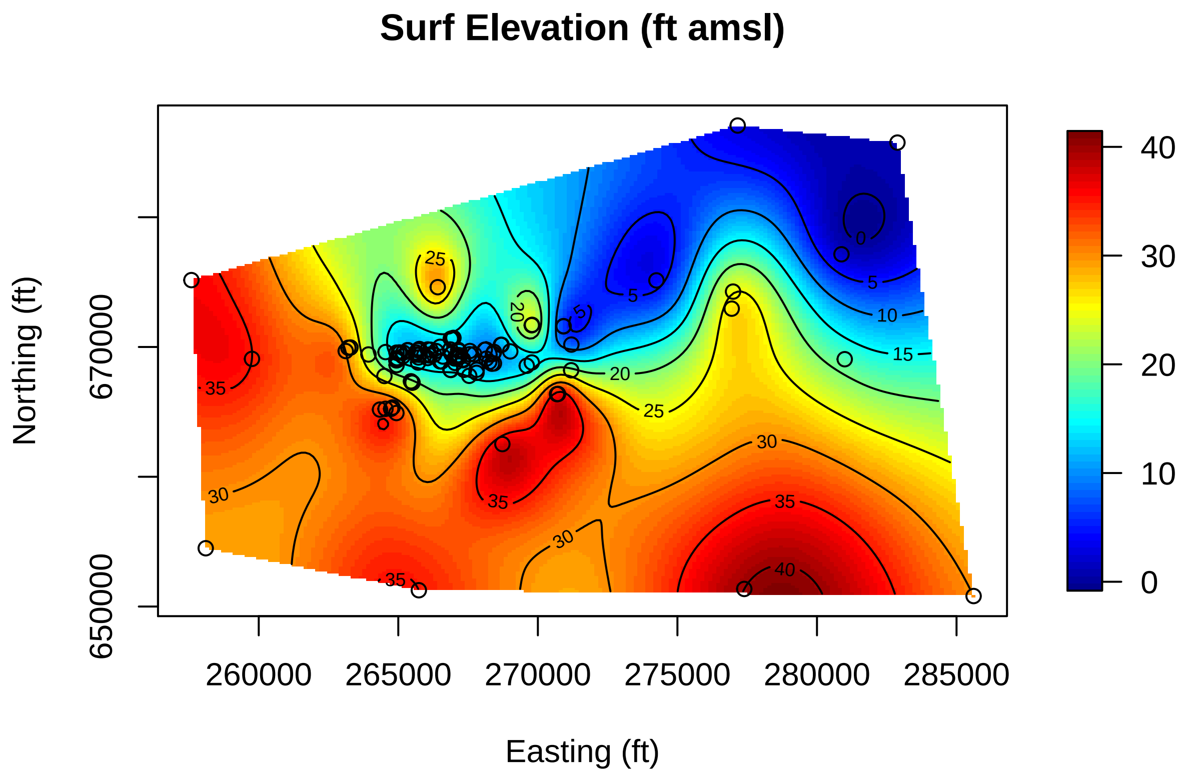
```
#load libraries
library(spBayes)
library(MBA)
library(geoR)
library(fields)
library(sp)
library(maptools)
library(rgdal)
library(classInt)
library(lattice)
library(dplyr)
```

a)

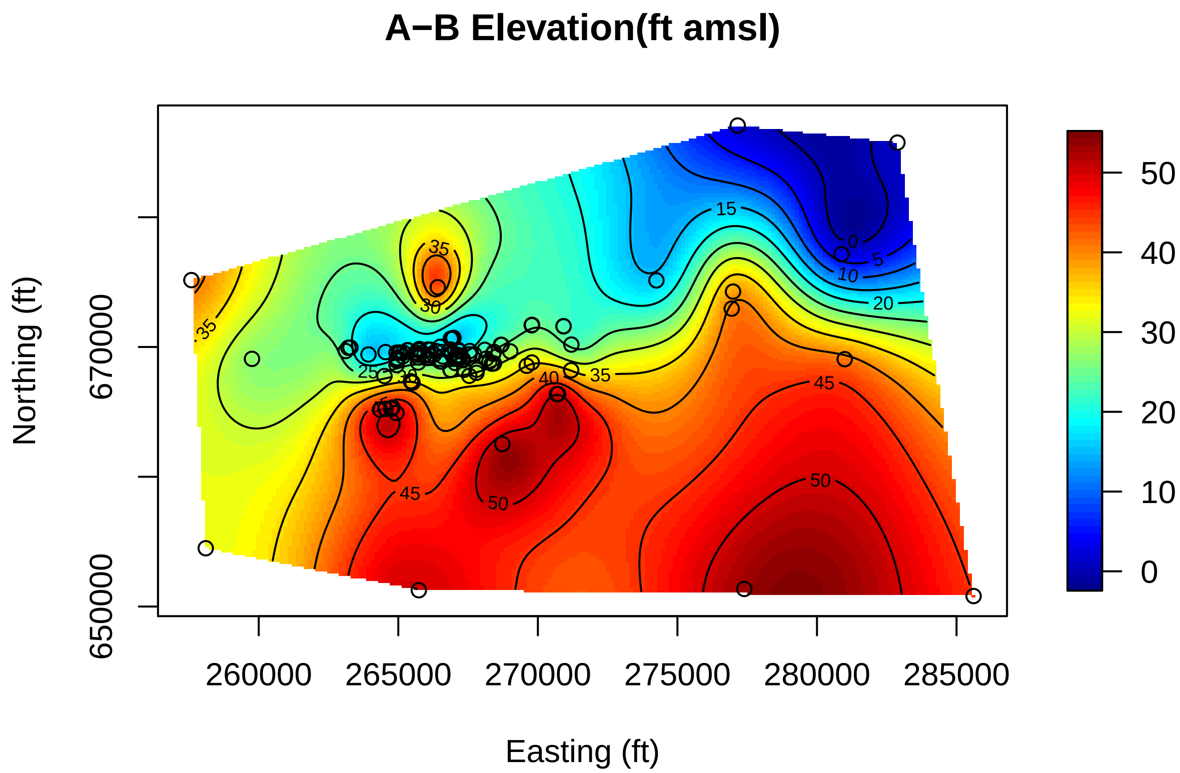
```
lithology_df <- read.csv("lithology.csv",header = T)
# drop the NA or miss data
lithology_df <- lithology_df %>% filter( !is.na(Thickness_ft) &
                                         !is.na(Surf_Elevation_ft_amsl) &
                                         !is.na(A_B_Elevation_ft_amsl))
lithology_df$Surf_Elevation_ft_amsl <- as.numeric(lithology_df$Surf_Elevation_ft_amsl)
lithology_df$A_B_Elevation_ft_amsl <- as.numeric(lithology_df$A_B_Elevation_ft_amsl)

# Extract the coordinates
coords <- as.matrix(lithology_df[,c("Easting_ft","Northing_ft")])

x.res <- 200; y.res <- 200

surf <- mba.surf(cbind(coords,
                       lithology_df$Thickness_ft),
                 no.X=x.res, no.Y=y.res, h=5,
                 m=2, extend=FALSE)$xyz.est
zlim.Thickness_ft <- range(surf[[3]], na.rm=TRUE)
image.plot(surf, xaxs = "r",
           yaxs = "r", xlab="Easting (ft)",
           ylab="Northing (ft)",
           main="Thickness(ft)" )
points(coords)
contour(surf,add = T)
```




```
surf <- mba.surf(cbind(coords,
                       lithology_df$Surf_Elevation_ft_amsl),
                 no.X=x.res, no.Y=y.res, h=5,
                 m=2, extend=FALSE)$xyz.est
zlim.Surf_Elevation_ft_amsl <- range(surf[[3]], na.rm=TRUE)
image.plot(surf, xaxs = "r",
           yaxs = "r", xlab="Easting (ft)",
           ylab="Northing (ft)",
           main="Surf Elevation (ft amsl)")
points(coords)
contour(surf,add = T)
```




```
surf <- mba.surf(cbind(coords,
                       lithology_df$A_B_Elevation_ft_amsl),
                 no.X=x.res, no.Y=y.res, h=5,
                 m=2, extend=FALSE)$xyz.est
zlim.A_B_Elevation_ft_amsl <- range(surf[[3]], na.rm=TRUE)
image.plot(surf, xaxs = "r",
           yaxs = "r", xlab="Easting (ft)",
           ylab="Northing (ft)",
           main="A-B Elevation(ft amsl)")
points(coords)
contour(surf,add = T)
```



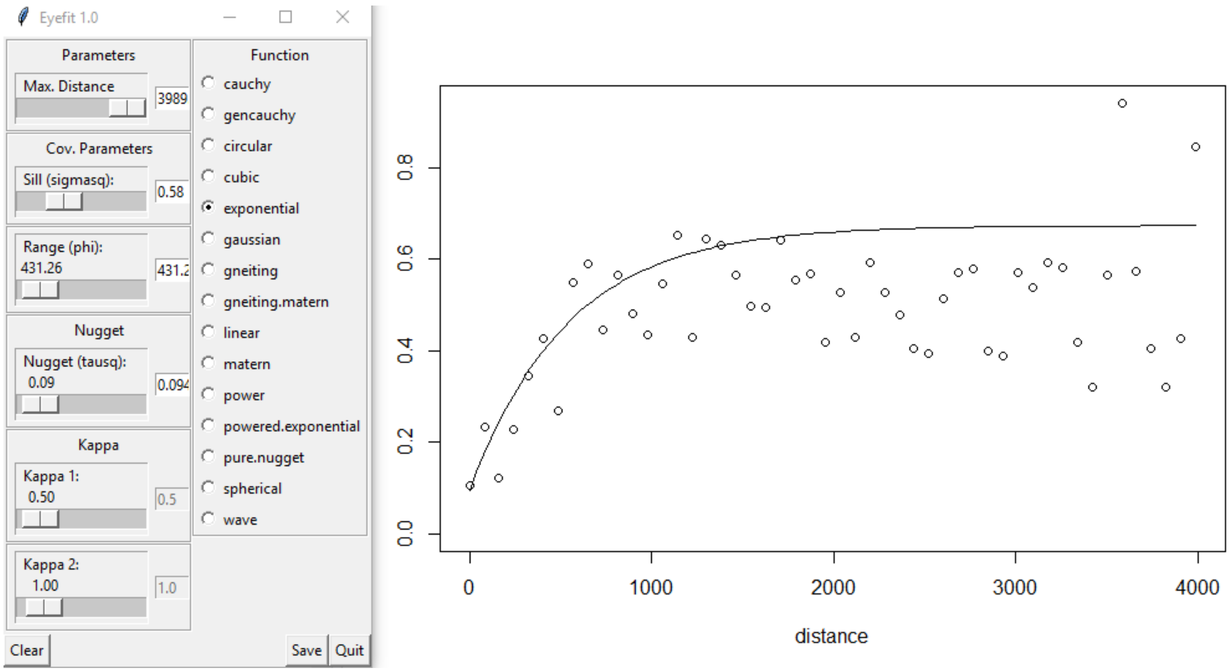
As we can see from the above results, the thickness IQR range about 5 feet to 20 feet whereas Surf Elevation is about 10 to 40 feet. Most of the points are located around Easting 265000 feet and Northing 67000 feet.

b)

```
# compute the variogram for exponential
log.thickness <- log(lithology_df$Thickness_ft)
bins = 50
max.dist <- 0.1*max(iDist(coords))
log.thick.vario <- variog(coords = coords, data = log.thickness,
                          uvec = (seq(0, max.dist, length = bins)))
```

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

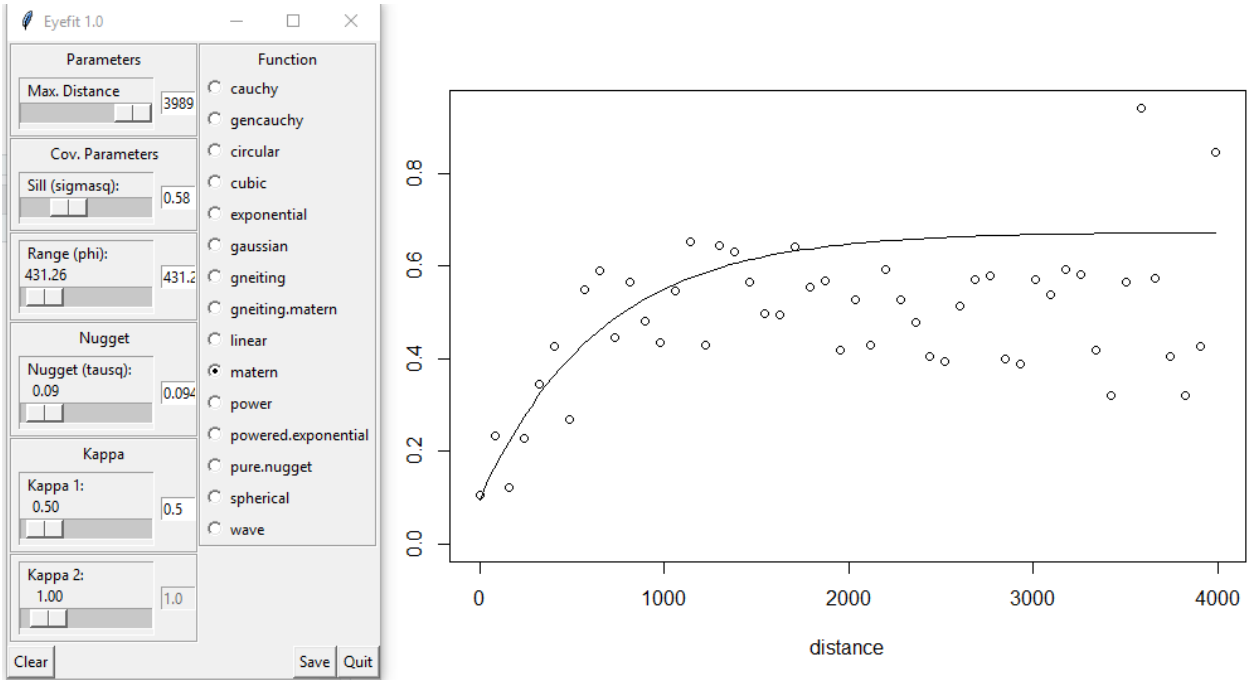
```
plot(log.thick.vario)
eyefit(log.thick.vario,silent=TRUE)
```




```r
# compute the variogram for matern
log.thick.vario <- variog(coords = coords, data = log.thickness,
                          uvec = (seq(0, max.dist, length = bins)))
```

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

```r
plot(log.thick.vario)
eyefit(log.thick.vario,silent=TRUE)
```




```
log.thickness <- log(lithology_df$Thickness_ft)

p <- 3 # This is the number of columns in the design matrix
# Set the prior mean and precision for the regression
beta.prior.mean <- as.matrix(rep(0, times=p))
beta.prior.precision <- matrix(0, nrow=p, ncol=p)

# For use with bayesGeostatExact, do the following
phi <- 0.014 ## Set the spatial range (from the variogram)
alpha <- 0.016/0.08 ## Set the nugget/partial-sill ratio
sigma.sq.prior.shape <- 0.1 ## Set IG shape for sigma.sq (partial sill)
sigma.sq.prior.rate <- 0.1 ## Set IG scale for sigma.sq (partial sill)

# Run bayesGeostatExact to deliver exact posterior samples
sp.exact <- bayesGeostatExact(
  log.thickness ~ Surf_Elevation_ft_amsl + A_B_Elevation_ft_amsl,
  data=lithology_df, coords=coords, n.samples=1000,
  beta.prior.mean=beta.prior.mean,
  beta.prior.precision=beta.prior.precision,
  cov.model="exponential",
  phi=phi, alpha=alpha,
  sigma.sq.prior.shape=sigma.sq.prior.shape,
  sigma.sq.prior.rate=sigma.sq.prior.rate,
  sp.effects=FALSE)
```

```
## -------------------------------------------------
```

```
##  General model description
## -------------------------------------------------
## Model fit with 113 observations.
## Number of covariates 3 (including intercept if specified).
## Using the exponential spatial correlation model.
##
## -------------------------------------------------
##      Sampling
## -------------------------------------------------
## Sampled: 1000 of 1000, 100%
```

```
#Produce the posterior summaries
round(summary(sp.exact$p.samples)$quantiles,3)
```

```
##                          2.5%    25%    50%    75%  97.5%
## (Intercept)             2.316  2.526  2.635  2.736  2.957
## Surf_Elevation_ft_amsl -0.048 -0.037 -0.031 -0.024 -0.012
## A_B_Elevation_ft_amsl  -0.015 -0.006 -0.001  0.003  0.012
## sigma.sq                0.274  0.321  0.353  0.388  0.465
## tau.sq                  0.055  0.064  0.071  0.078  0.093
```

```
phi <- 1/1100 ## Set the spatial range (from the variogram)
alpha <- 0.094/0.58 ## Set the nugget/partial-sill ratio
nu <- 0.5

# Run bayesGeostatExact to deliver exact posterior samples
sp.exact <- bayesGeostatExact(
  log.thickness ~ Surf_Elevation_ft_amsl + A_B_Elevation_ft_amsl,
  data=lithology_df, coords=coords, n.samples=1000,
  beta.prior.mean=beta.prior.mean,
  beta.prior.precision=beta.prior.precision,
  cov.model="matern",
  phi=phi, alpha=alpha,
  nu=nu,
  sigma.sq.prior.shape=sigma.sq.prior.shape,
  sigma.sq.prior.rate=sigma.sq.prior.rate,
  sp.effects=FALSE)
```

```
## -------------------------------------------------
##  General model description
## -------------------------------------------------
## Model fit with 113 observations.
## Number of covariates 3 (including intercept if specified).
## Using the matern spatial correlation model.
##
## -------------------------------------------------
##      Sampling
## -------------------------------------------------
## Sampled: 1000 of 1000, 100%
```

```
round(summary(sp.exact$p.samples)$quantiles,3)
```

```
##                          2.5%    25%    50%    75% 97.5%
## (Intercept)             1.920  2.343  2.568  2.810 3.285
## Surf_Elevation_ft_amsl -0.042 -0.022 -0.013 -0.004 0.015
## A_B_Elevation_ft_amsl  -0.029 -0.019 -0.013 -0.008 0.002
```

```
## sigma.sq                  0.546  0.637  0.693  0.764 0.907
## tau.sq                    0.089  0.103  0.112  0.124 0.147
```

As we can see from the above results, the log Thickness has a negative correlation. For exponential, the model can draw as $log.thickness = 2.315 - 0.048 * Surf\_Elevation\_ft\_amsl - 0.015 * A\_B\_Elevation\_ft\_amsl$, and $\sigma^2 = 0.276$, $\tau^2 = 0.055$. For exponential, the model can draw as $log.thickness = 1.926 - 0.039 * Surf\_Elevation\_ft\_amsl - 0.028 * A\_B\_Elevation\_ft\_amsl$, and $\sigma^2 = 0.544$, $\tau^2 = 0.088$.

c)

```
# Run spLM to deliver MCMC samples from marginal posterior distributions
n.samples <- 1000
thickness.sp <- spLM(log.thickness ~ Surf_Elevation_ft_amsl + A_B_Elevation_ft_amsl,
               data=lithology_df, coords=coords,
               starting=list("phi"=3/1100,"sigma.sq"=0.08,"tau.sq"=0.02),
               tuning=list("phi"=0.1, "sigma.sq"=0.05, "tau.sq"=0.05),
               priors=list("phi.Unif"=c(3/1500, 3/50),
                           "sigma.sq.IG"=c(0.1,0.1),"tau.sq.IG"=c(0.1, 0.1)),
               cov.model="exponential",n.samples=n.samples)
```

```
## ----------------------------------------
##  General model description
## ----------------------------------------
## Model fit with 113 observations.
##
## Number of covariates 3 (including intercept if specified).
##
## Using the exponential spatial correlation model.
##
## Number of MCMC samples 1000.
##
## Priors and hyperpriors:
##  beta flat.
##  sigma.sq IG hyperpriors shape=0.10000 and scale=0.10000
##  tau.sq IG hyperpriors shape=0.10000 and scale=0.10000
##  phi Unif hyperpriors a=0.00200 and b=0.06000
## -------------------------------------------------
##      Sampling
## -------------------------------------------------
## Sampled: 100 of 1000, 10.00%
## Report interval Metrop. Acceptance rate: 58.00%
## Overall Metrop. Acceptance rate: 58.00%
## -------------------------------------------------
## Sampled: 200 of 1000, 20.00%
## Report interval Metrop. Acceptance rate: 49.00%
## Overall Metrop. Acceptance rate: 53.50%
## -------------------------------------------------
## Sampled: 300 of 1000, 30.00%
## Report interval Metrop. Acceptance rate: 56.00%
## Overall Metrop. Acceptance rate: 54.33%
## -------------------------------------------------
## Sampled: 400 of 1000, 40.00%
## Report interval Metrop. Acceptance rate: 58.00%
## Overall Metrop. Acceptance rate: 55.25%
## -------------------------------------------------
```

```
## Sampled: 500 of 1000, 50.00%
## Report interval Metrop. Acceptance rate: 59.00%
## Overall Metrop. Acceptance rate: 56.00%
## -------------------------------------------------
## Sampled: 600 of 1000, 60.00%
## Report interval Metrop. Acceptance rate: 56.00%
## Overall Metrop. Acceptance rate: 56.00%
## -------------------------------------------------
## Sampled: 700 of 1000, 70.00%
## Report interval Metrop. Acceptance rate: 52.00%
## Overall Metrop. Acceptance rate: 55.43%
## -------------------------------------------------
## Sampled: 800 of 1000, 80.00%
## Report interval Metrop. Acceptance rate: 60.00%
## Overall Metrop. Acceptance rate: 56.00%
## -------------------------------------------------
## Sampled: 900 of 1000, 90.00%
## Report interval Metrop. Acceptance rate: 59.00%
## Overall Metrop. Acceptance rate: 56.33%
## -------------------------------------------------
## Sampled: 1000 of 1000, 100.00%
## Report interval Metrop. Acceptance rate: 53.00%
## Overall Metrop. Acceptance rate: 56.00%
## -------------------------------------------------
```

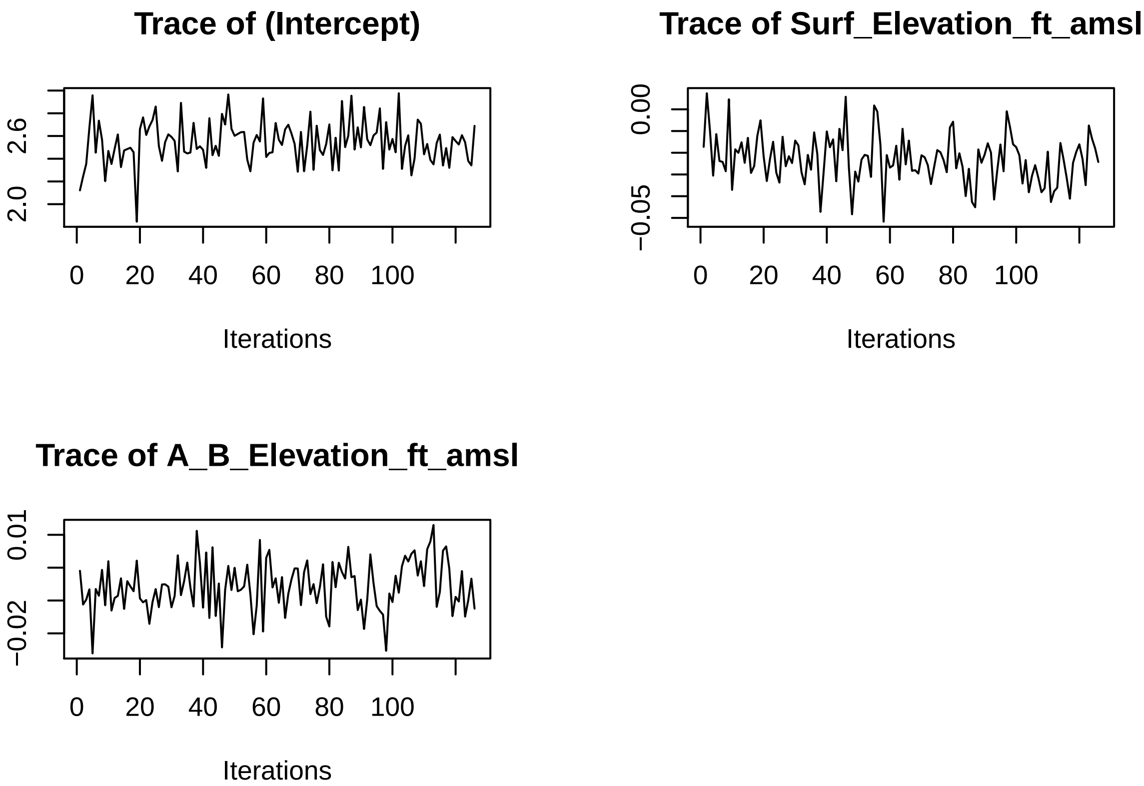
```r
round(summary(mcmc(thickness.sp$p.theta.samples))$quantiles,3)
```

```
##           2.5%   25%   50%   75% 97.5%
## sigma.sq 0.107 0.320 0.383 0.449 0.572
## tau.sq   0.038 0.066 0.088 0.128 0.221
## phi      0.002 0.003 0.005 0.007 0.011
```

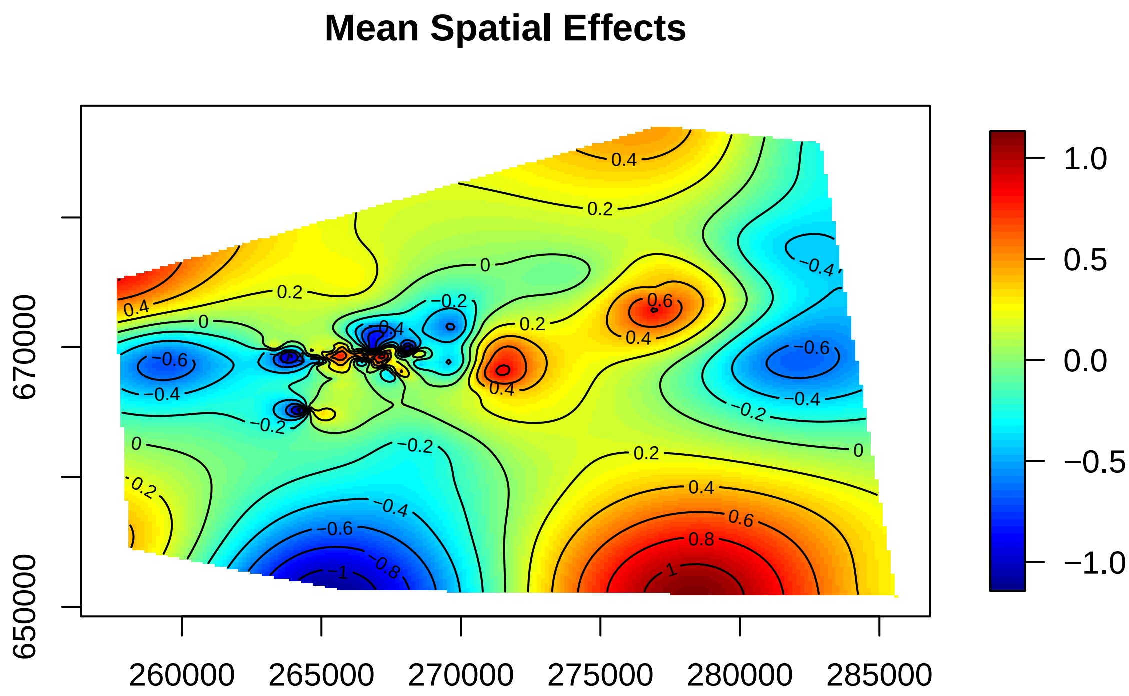
```r
# Recover spatial residuals using spRecover
burn.in <- floor(0.75*n.samples)
thickness.sp <- spRecover(thickness.sp, start=burn.in, thin=2)
```

```
## -------------------------------------------------
##      Recovering beta and w
## -------------------------------------------------
## Sampled: 99 of 126, 78.57%
```

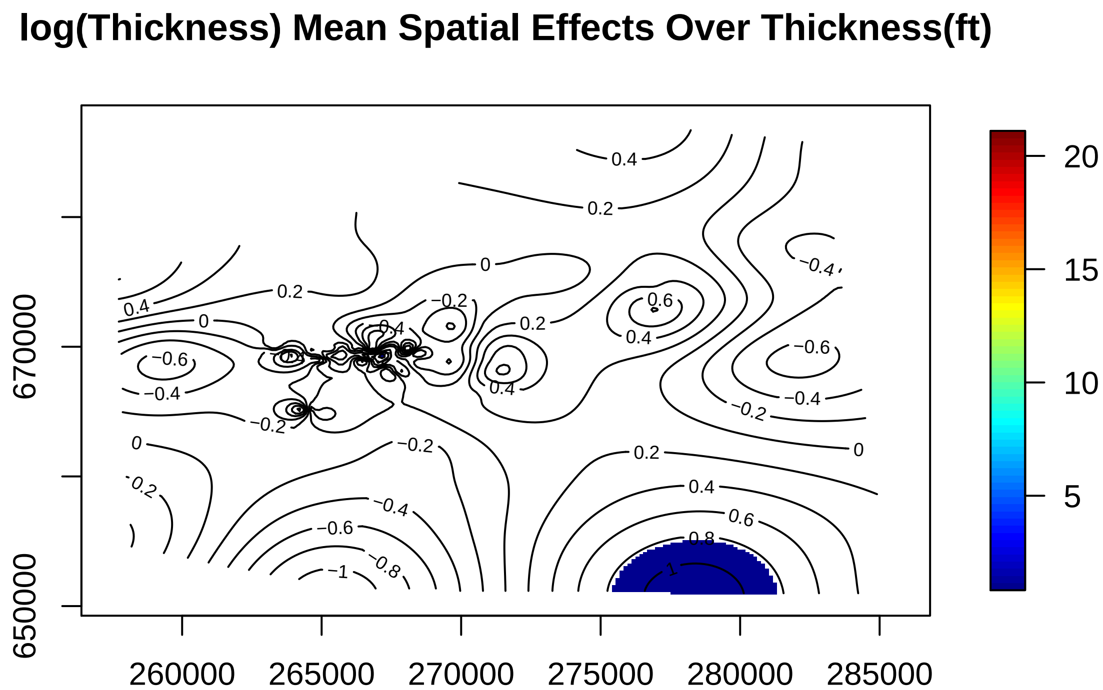
```r
# The posterior samples of the regression coefficients and the spatial effects can then be obtained as
beta.samples = thickness.sp$p.beta.recover.samples
w.samples = thickness.sp$p.w.recover.samples

# Obtain trace plots for regression coefficients
par(mfrow=c(2,2))
plot(beta.samples, auto.layout=TRUE, density=FALSE)
```



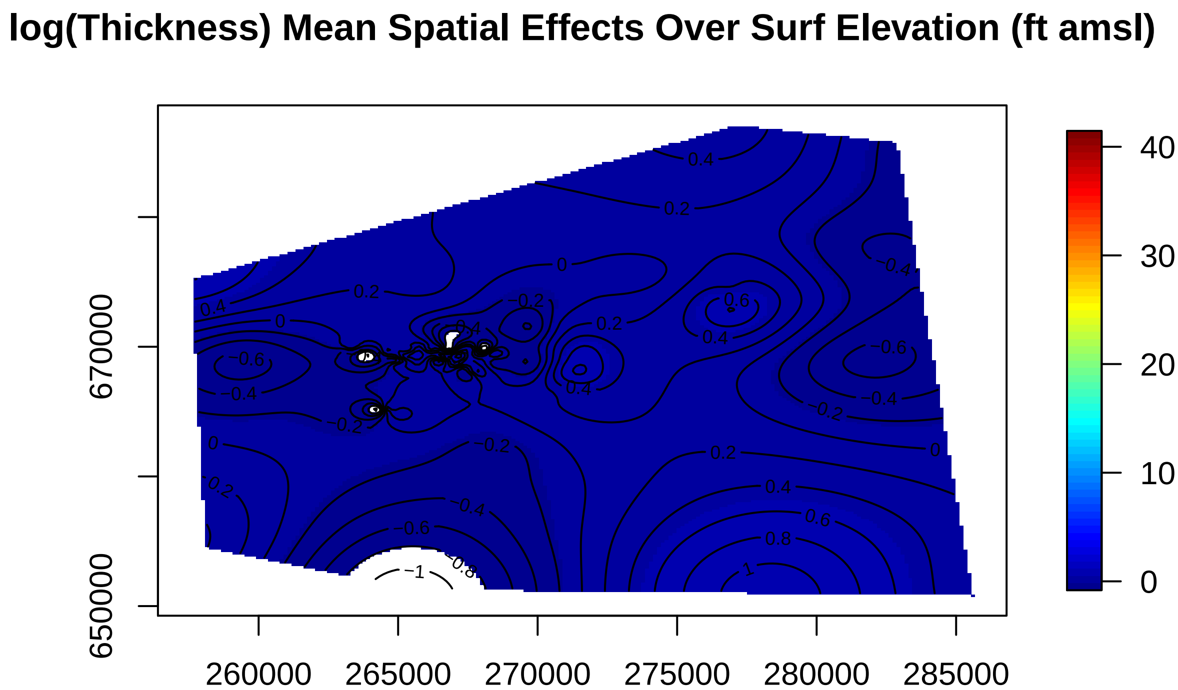

```r
par(mfrow=c(1,1))
# Obtain posterior means and sd's of spatial residuals for each location
w.hat.mu <- apply(w.samples,1,mean)
w.hat.sd <- apply(w.samples,1,sd)


surf <- mba.surf(cbind(coords, w.hat.mu),
                 no.X=x.res, no.Y=y.res, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           main="Mean Spatial Effects")
contour(surf,add = T)
```



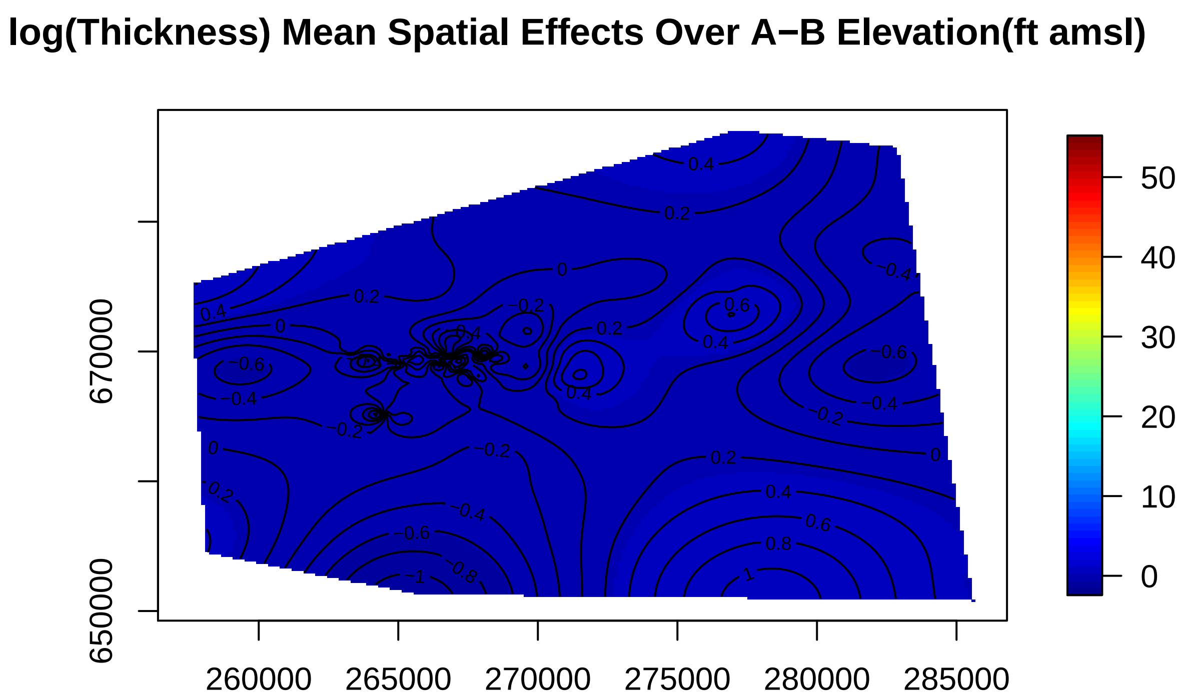

```
surf <- mba.surf(cbind(coords, w.hat.mu),
                 no.X=x.res, no.Y=y.res, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           zlim = zlim.Thickness_ft,
           main="log(Thickness) Mean Spatial Effects Over Thickness(ft)")
contour(surf,add = T)
```




```r
surf <- mba.surf(cbind(coords, w.hat.mu),
                 no.X=x.res, no.Y=y.res, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           zlim = zlim.Surf_Elevation_ft_amsl,
           main="log(Thickness) Mean Spatial Effects Over Surf Elevation (ft amsl)")
contour(surf,add = T)
```




```r
surf <- mba.surf(cbind(coords, w.hat.mu),
                 no.X=x.res, no.Y=y.res, extend=FALSE)$xyz.est
image.plot(surf, xaxs = "r", yaxs = "r",
           zlim = zlim.A_B_Elevation_ft_amsl,
           main="log(Thickness) Mean Spatial Effects Over A-B Elevation(ft amsl) ")
contour(surf,add = T)
```



d)

```r
# run model with purely spatial
# note: set tau.sq=0 will work on spRecover and spDiag, therefore we set it a value close to 0
thickness.sp2 <- spLM(log.thickness ~ Surf_Elevation_ft_amsl + A_B_Elevation_ft_amsl,
               data=lithology_df, coords=coords,
               starting=list("phi"=3/1100,"sigma.sq"=0.08,"tau.sq"=1E-10),
               tuning=list("phi"=0.1, "sigma.sq"=0.05, "tau.sq"=1E-10),
               priors=list("phi.Unif"=c(3/1500, 3/50),
                           "sigma.sq.IG"=c(0.1,0.1),
                           "tau.sq.IG"=c(1E-10, 1E-10)),
               cov.model="exponential",n.samples=n.samples)
```

```
## ----------------------------------------
##  General model description
## ----------------------------------------
## Model fit with 113 observations.
##
## Number of covariates 3 (including intercept if specified).
##
## Using the exponential spatial correlation model.
##
## Number of MCMC samples 1000.
##
## Priors and hyperpriors:
##  beta flat.
```

```
##  sigma.sq IG hyperpriors shape=0.10000 and scale=0.10000
##  tau.sq IG hyperpriors shape=0.00000 and scale=0.00000
##  phi Unif hyperpriors a=0.00200 and b=0.06000
## -------------------------------------------------
##      Sampling
## -------------------------------------------------
## Sampled: 100 of 1000, 10.00%
## Report interval Metrop. Acceptance rate: 42.00%
## Overall Metrop. Acceptance rate: 42.00%
## -------------------------------------------------
## Sampled: 200 of 1000, 20.00%
## Report interval Metrop. Acceptance rate: 55.00%
## Overall Metrop. Acceptance rate: 48.50%
## -------------------------------------------------
## Sampled: 300 of 1000, 30.00%
## Report interval Metrop. Acceptance rate: 52.00%
## Overall Metrop. Acceptance rate: 49.67%
## -------------------------------------------------
## Sampled: 400 of 1000, 40.00%
## Report interval Metrop. Acceptance rate: 48.00%
## Overall Metrop. Acceptance rate: 49.25%
## -------------------------------------------------
## Sampled: 500 of 1000, 50.00%
## Report interval Metrop. Acceptance rate: 39.00%
## Overall Metrop. Acceptance rate: 47.20%
## -------------------------------------------------
## Sampled: 600 of 1000, 60.00%
## Report interval Metrop. Acceptance rate: 46.00%
## Overall Metrop. Acceptance rate: 47.00%
## -------------------------------------------------
## Sampled: 700 of 1000, 70.00%
## Report interval Metrop. Acceptance rate: 41.00%
## Overall Metrop. Acceptance rate: 46.14%
## -------------------------------------------------
## Sampled: 800 of 1000, 80.00%
## Report interval Metrop. Acceptance rate: 53.00%
## Overall Metrop. Acceptance rate: 47.00%
## -------------------------------------------------
## Sampled: 900 of 1000, 90.00%
## Report interval Metrop. Acceptance rate: 50.00%
## Overall Metrop. Acceptance rate: 47.33%
## -------------------------------------------------
## Sampled: 1000 of 1000, 100.00%
## Report interval Metrop. Acceptance rate: 54.00%
## Overall Metrop. Acceptance rate: 48.00%
## -------------------------------------------------
```

```r
round(summary(mcmc(thickness.sp2$p.theta.samples))$quantiles,3)
```

```
##           2.5%   25%   50%   75% 97.5%
## sigma.sq 0.132 0.440 0.493 0.563 0.686
## tau.sq   0.000 0.000 0.000 0.000 0.000
## phi      0.004 0.007 0.009 0.012 0.020
```

```r
thickness.sp2 <- spRecover(thickness.sp2, start=burn.in, thin=2)
```

```
## -------------------------------------------------
##      Recovering beta and w
## -------------------------------------------------
## Sampled: 99 of 126, 78.57%
```

```r
Dic1 = spDiag(thickness.sp,start=burn.in,verbose=FALSE)
Dic1
```

```
## $DIC
##                  value
## bar.D       -149.22028
## D.bar.Omega -218.12442
## pD            68.90414
## DIC          -80.31614
##
## $GP
##       value
## G  3.833089
## P 20.463483
## D 24.296571
##
## $GRS
## [1] 173.8616
```

```r
Dic2 = spDiag(thickness.sp2,start=burn.in,verbose=FALSE)
Dic2
```

```
## $DIC
##                  value
## bar.D       -2488.5174
## D.bar.Omega -2600.9881
## pD            112.4707
## DIC         -2376.0468
##
## $GP
##          value
## G 2.032464e-10
## P 2.306171e-08
## D 2.326495e-08
##
## $GRS
## [1] 2521.242
```

Compare the two models with above DIC values, the second model with much nugget is better than the first one with nugget.

Excercise 7

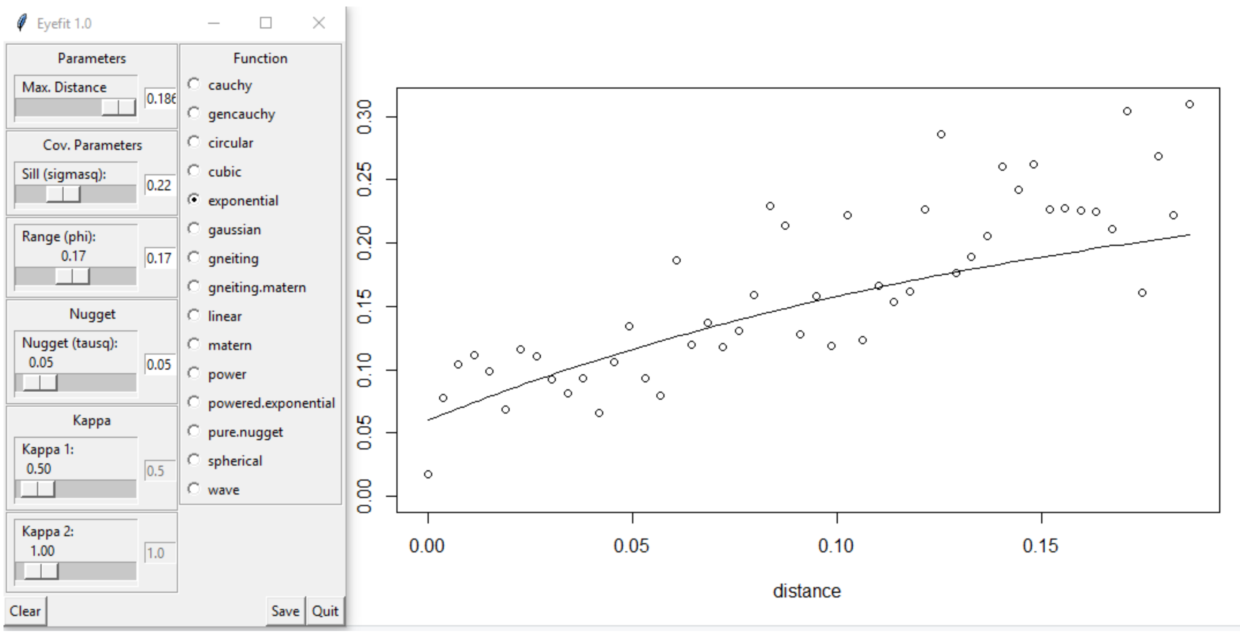
a)

```r
bat_df <- read.table("batonrouge.dat",header = T)

# Extract the coordinates
coords <- as.matrix(bat_df[,c("Longitude","Latitude")])
```

```r
# compute the variogram for exponential
bins = 50
max.dist <- 0.7*max(iDist(coords))
log.selling.vario <- variog(coords = coords, data = bat_df$logSellingPr,
                            uvec = (seq(0, max.dist, length = bins)))
```

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

```r
plot(log.selling.vario)
eyefit(log.selling.vario,silent=TRUE)
```



b)

```r
# OLS
lm.log.selling = lm(logSellingPr~LivingArea + Age + OtherArea + Bathrooms, data=bat_df)
lm.log.selling
```

```
##
## Call:
## lm(formula = logSellingPr ~ LivingArea + Age + OtherArea + Bathrooms,
##     data = bat_df)
##
## Coefficients:
## (Intercept)   LivingArea          Age    OtherArea    Bathrooms
##  10.2065918    0.0005892   -0.0099027    0.0001508   -0.0512299
```

```r
summary(lm.log.selling)
```

```
##
## Call:
## lm(formula = logSellingPr ~ LivingArea + Age + OtherArea + Bathrooms,
##     data = bat_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58154 -0.08757  0.00618  0.12072  0.53926
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.021e+01  1.617e-01  63.113  < 2e-16 ***
## LivingArea   5.892e-04  6.314e-05   9.331  1.3e-13 ***
## Age         -9.903e-03  2.960e-03  -3.346  0.00137 **
## OtherArea    1.508e-04  1.232e-04   1.225  0.22516
## Bathrooms   -5.123e-02  8.464e-02  -0.605  0.54713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1996 on 65 degrees of freedom
## Multiple R-squared:  0.7696, Adjusted R-squared:  0.7554
## F-statistic: 54.27 on 4 and 65 DF,  p-value: < 2.2e-16
```

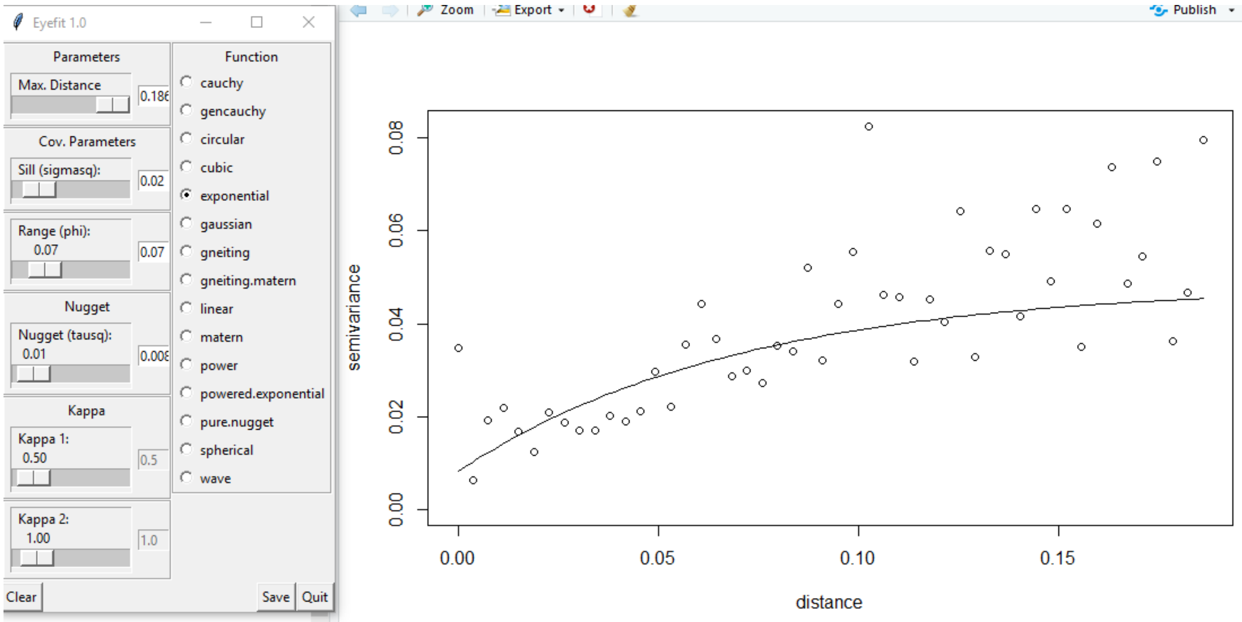
As we can see from the above results, the linear models is $logSellingPr = 10.2065918 + 0.0005892 * LivingArea - 0.0099027 * Age + 0.0001508 * OtherArea - 0.0512299 * Bathrooms$, the model log selling price has a negative relationship with age and bathrooms. It is easy to understand that the old house should be cheaper, but it is a little strange about the bathrooms. Mybe, the people, do not want to bathrooms occupy too much area of the house.

c)

```r
# Obtain OLS residuals
selling.resid = resid(lm.log.selling)
selling.resid.vario <- variog(coords = coords, data = selling.resid,
                              uvec = (seq(0, max.dist, length = bins)))
```

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

```r
plot(selling.resid.vario)
eyefit(selling.resid.vario,silent=TRUE)
```



d)

```
# the values of sigma square, tau sqaure  and nugget are from the above ressults
point<-krige.conv(coords = coords, data = bat_df$logSellingPr,loc=c(length(bat_df$logSellingPr),1),
                  krige=krige.control(cov.pars=c(0.22,0.17),cov.model="exponential",nugget=0.05))
```

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

```
point
```

```
## $predict
##     data
## 11.12649
##
## $krige.var
## [1] 0.3799696
##
## $beta.est
##     beta
## 11.12649
##
## $distribution
## [1] "normal"
##
## $message
## [1] "krige.conv: Kriging performed using global neighbourhood"
```

```
##
## $call
## krige.conv(coords = coords, data = bat_df$logSellingPr, locations = c(length(bat_df$logSellingPr),
##     1), krige = krige.control(cov.pars = c(0.22, 0.17), cov.model = "exponential",
##     nugget = 0.05))
##
## attr(,"sp.dim")
## [1] "2d"
## attr(,"prediction.locations")
## c(length(bat_df$logSellingPr), 1)
## attr(,"parent.env")
## <environment: R_GlobalEnv>
## attr(,"data.locations")
## coords
## attr(,"class")
## [1] "kriging"
```

```r
pred_low <-point$predict - 2*sqrt(point$krige.var)
pred_high <-point$predict + 2*sqrt(point$krige.var)
print(paste("The 95% confident PI is between",pred_low,"and",pred_high))
```

```
## [1] "The 95% confident PI is between 9.89365211713998 and 12.35931920472"
```

e)

Give above model: $logSellingPr = 10.2065918 + 0.0005892 * LivingArea - 0.0099027 * Age + 0.0001508 * OtherArea - 0.0512299 * Bathrooms$

```r
LivingArea <- 938
Age <- 25
OtherArea <- 332
Bathrooms <- 3
logSellingPr <- 10.2065918+0.0005892*LivingArea-0.0099027*Age+0.0001508*OtherArea-0.0512299*Bathrooms
print(paste("predict results=",logSellingPr))
```

```
## [1] "predict results= 10.4080698"
```

As we can see, the predict results is 10.40807, and the true value is 10.448, which is very good prediction.

f)

```r
n.samples = 1000

log.selling.mcmc <- spLM(logSellingPr~LivingArea + Age + OtherArea + Bathrooms,
                         data=bat_df,
                         coords=coords,
                         starting=list("phi"=0.17,"sigma.sq"=0.22,
                                       "tau.sq"=0.05),
                         tuning=list("phi"=0.1, "sigma.sq"=0.05, "tau.sq"=0.05),
                         priors=list("phi.Unif"=c(3/1500, 10),
                                     "sigma.sq.IG"=c(0.1, 0.1),
                                     "tau.sq.IG"=c(0.1, 0.1)),
                         cov.model="exponential",n.samples=n.samples)
```

```
## ----------------------------------------
##  General model description
```

```
## ----------------------------------------
## Model fit with 70 observations.
##
## Number of covariates 5 (including intercept if specified).
##
## Using the exponential spatial correlation model.
##
## Number of MCMC samples 1000.
##
## Priors and hyperpriors:
##  beta flat.
##  sigma.sq IG hyperpriors shape=0.10000 and scale=0.10000
##  tau.sq IG hyperpriors shape=0.10000 and scale=0.10000
##  phi Unif hyperpriors a=0.00200 and b=10.00000
## -------------------------------------------------
##      Sampling
## -------------------------------------------------
## Sampled: 100 of 1000, 10.00%
## Report interval Metrop. Acceptance rate: 65.00%
## Overall Metrop. Acceptance rate: 65.00%
## -------------------------------------------------
## Sampled: 200 of 1000, 20.00%
## Report interval Metrop. Acceptance rate: 58.00%
## Overall Metrop. Acceptance rate: 61.50%
## -------------------------------------------------
## Sampled: 300 of 1000, 30.00%
## Report interval Metrop. Acceptance rate: 66.00%
## Overall Metrop. Acceptance rate: 63.00%
## -------------------------------------------------
## Sampled: 400 of 1000, 40.00%
## Report interval Metrop. Acceptance rate: 68.00%
## Overall Metrop. Acceptance rate: 64.25%
## -------------------------------------------------
## Sampled: 500 of 1000, 50.00%
## Report interval Metrop. Acceptance rate: 62.00%
## Overall Metrop. Acceptance rate: 63.80%
## -------------------------------------------------
## Sampled: 600 of 1000, 60.00%
## Report interval Metrop. Acceptance rate: 64.00%
## Overall Metrop. Acceptance rate: 63.83%
## -------------------------------------------------
## Sampled: 700 of 1000, 70.00%
## Report interval Metrop. Acceptance rate: 58.00%
## Overall Metrop. Acceptance rate: 63.00%
## -------------------------------------------------
## Sampled: 800 of 1000, 80.00%
## Report interval Metrop. Acceptance rate: 58.00%
## Overall Metrop. Acceptance rate: 62.38%
## -------------------------------------------------
## Sampled: 900 of 1000, 90.00%
## Report interval Metrop. Acceptance rate: 65.00%
## Overall Metrop. Acceptance rate: 62.67%
## -------------------------------------------------
## Sampled: 1000 of 1000, 100.00%
```

```
## Report interval Metrop. Acceptance rate: 58.00%
## Overall Metrop. Acceptance rate: 62.20%
## -------------------------------------------------
```

```
round(summary(mcmc(log.selling.mcmc$p.theta.samples))$quantiles,3)
```

```
##           2.5%   25%   50%   75% 97.5%
## sigma.sq 0.041 0.116 0.198 0.485 0.992
## tau.sq   0.014 0.020 0.024 0.028 0.040
## phi      0.181 0.768 1.364 3.084 9.253
```

```
burn.in <- floor(0.75*n.samples)
log.selling.mcmc <- spRecover(log.selling.mcmc, start=burn.in, thin=2)
```

```
## -------------------------------------------------
##      Recovering beta and w
## -------------------------------------------------
## Sampled: 99 of 126, 78.57%
```

```
Dic1 = spDiag(log.selling.mcmc,start=burn.in,verbose=FALSE)
Dic1
```

```
## $DIC
##                  value
## bar.D       -196.00163
## D.bar.Omega -215.51265
## pD            19.51101
## DIC         -176.49062
##
## $GP
##      value
## G 1.056456
## P 2.435316
## D 3.491772
##
## $GRS
## [1] 209.1287
```

The DIC is considerably small, but I cannot compare with other models, because other model donot have DIC.

g)

```
beta.samples = log.selling.mcmc$p.beta.recover.samples
w.samples = log.selling.mcmc$p.w.recover.samples
## Obtain posterior means and sd's of spatial residuals for each location
w.hat.mu <- apply(w.samples,1,mean)
w.hat.mu
```

```
##  [1]  0.059380121 -0.057783702 -0.154705990  0.044407457  0.074972700
##  [6]  0.151889090  0.105070923  0.031154997 -0.136549721  0.107769010
## [11]  0.220069073  0.064932157  0.144751178  0.135657024 -0.014676985
## [16]  0.059892508 -0.049153522  0.111470221 -0.095788020  0.193963448
## [21]  0.037197459  0.018657754  0.048828284 -0.082925812 -0.010582583
## [26]  0.108630993  0.038668905  0.029137228  0.092292138 -0.226952531
## [31]  0.212668053  0.194204206  0.091237462  0.081959813  0.100602525
## [36] -0.016343719 -0.146324020  0.251782548 -0.019271370  0.074158578
```

```
## [41]  0.028791287  0.036314593  0.151410271  0.101581842  0.009193362
## [46]  0.164004095  0.094771741  0.180876543  0.104436327  0.114630638
## [51]  0.089007283  0.104573897  0.091394246 -0.007471693  0.096305287
## [56]  0.064681618 -0.047861686 -0.191604472 -0.200041520  0.063855564
## [61] -0.172385539 -0.014129917  0.030246250  0.147782021 -0.080022993
## [66]  0.068605052 -0.212021414  0.210112775  0.120569582 -0.245351516
```

```r
w.hat.sd <- apply(w.samples,1,sd)
w.hat.sd
```

```
##  [1] 0.6464418 0.6462885 0.6618975 0.6409119 0.6391850 0.6524066 0.6302877
##  [8] 0.6345496 0.6327039 0.6403917 0.6242578 0.6438275 0.6328420 0.6287909
## [15] 0.6509210 0.6423428 0.6444981 0.6477829 0.6541271 0.6358252 0.6382986
## [22] 0.6422854 0.6358628 0.6454636 0.6504003 0.6426646 0.6329815 0.6323629
## [29] 0.6425103 0.6613407 0.6454264 0.6420567 0.6442774 0.6321556 0.6424159
## [36] 0.6409272 0.6555002 0.6374222 0.6502316 0.6370852 0.6435921 0.6339109
## [43] 0.6465354 0.6507827 0.6567735 0.6381727 0.6384013 0.6512974 0.6305903
## [50] 0.6454409 0.6433519 0.6425580 0.6486471 0.6379522 0.6398144 0.6444340
## [57] 0.6492211 0.6519507 0.6534964 0.6439230 0.6398888 0.6487160 0.6359542
## [64] 0.6532251 0.6463981 0.6397340 0.6618009 0.6403509 0.6435176 0.6430949
```

```r
summary(w.hat.mu)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.24535 -0.01454  0.06427  0.03967  0.10709  0.25178
```

```r
summary(w.hat.sd)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6243  0.6382  0.6429  0.6432  0.6484  0.6619
```