Project 2

Spatial Statistics

Jim og Christian

03 02 2022

Contents


```
library(geoR)
library(akima)
library(fields)
library(MASS)
library(viridis)
library(tidyverse)
library(spatstat)
library(scatterplot3d)
library(plotly)
```

2 GRF - Real Data

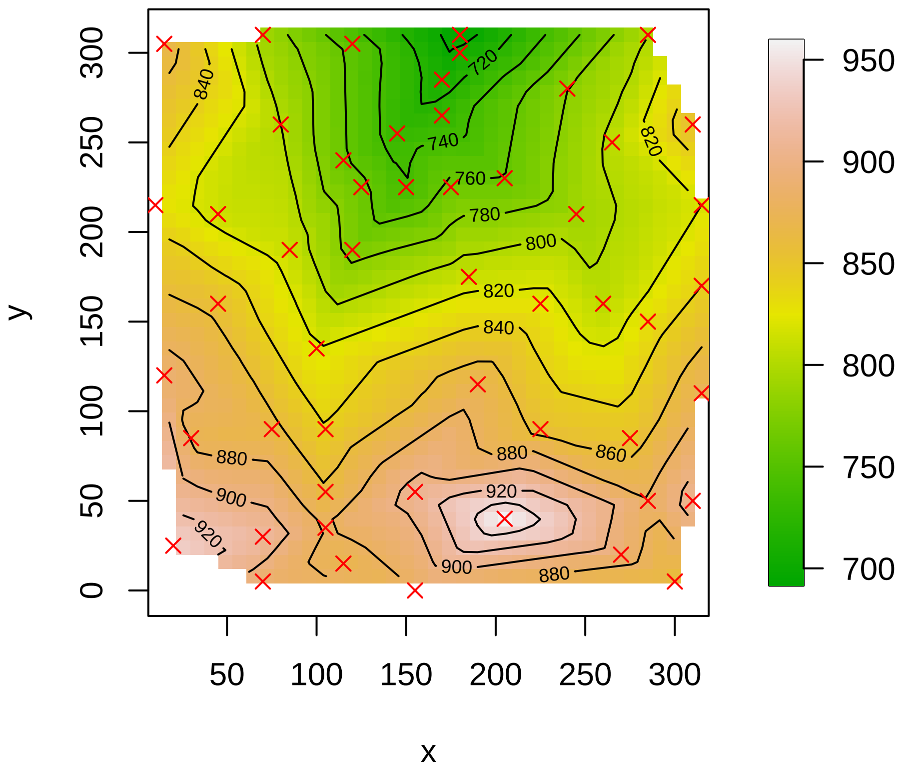
a) Visualization

Our dataset contain $n = 52$observations of terrain elevation located in the domain $\mathcal{D} = [0, 315]^2 \subset \mathbb{R}^{\not\in}$.

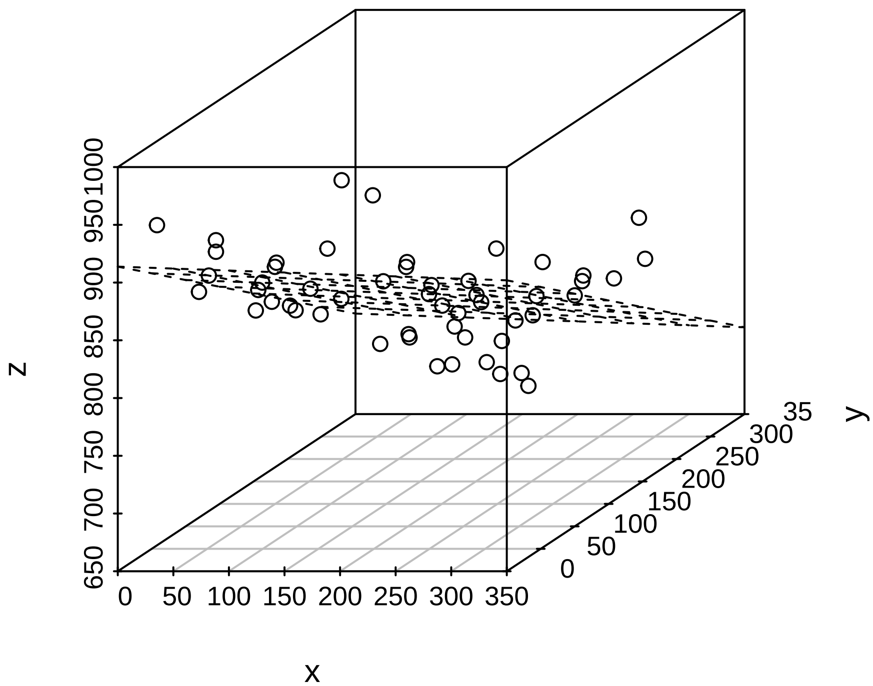
Visualization Vet ikke om d trengs noe mer intro til visualization?

```
figPath = "./Fysmat/8 Semester V2022/RomStat/romstat/project1/Problem2Figures/"

dt <- read.table(file = "~/Fysmat/8 Semester V2022/RomStat/Project1/Prj1Code_romstat/topo.dat")

setwd("C:/Users/chris/OneDrive/Documents/Fysmat/8 Semester V2022/RomStat/romstat/project1/Problem2Figur

# Setup
gd = as.geodata(dt)
```

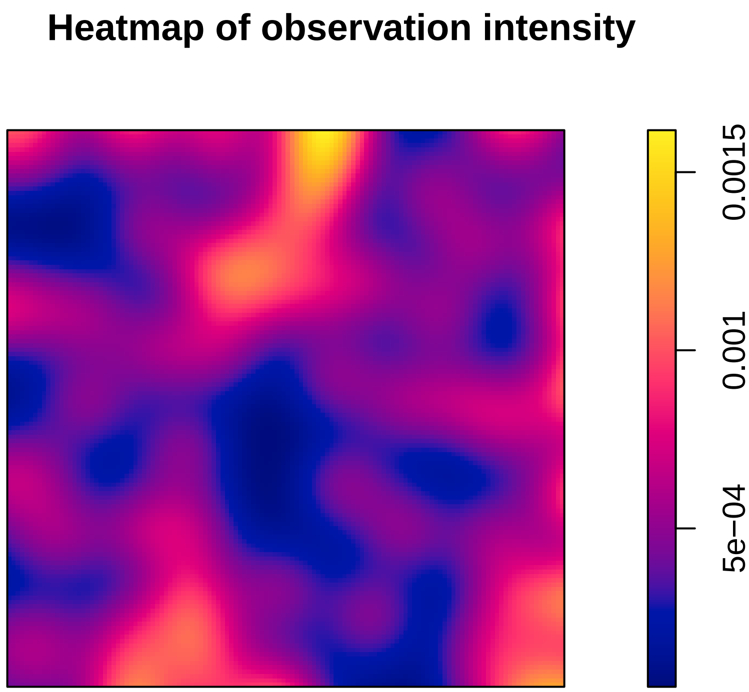
```
grid <- interp(dt$x, dt$y, dt$z)
griddf <- subset(data.frame(x = rep(grid$x, nrow(grid$z)), y = rep(grid$y, each = ncol(grid$z)),
    z = as.numeric(grid$z)), !is.na(z))
# pdf('2a_contour.pdf')
image.plot(grid, col = terrain.colors(200), asp = 1, xlab = "x", ylab = "y")
contour(grid, add = T)
points(dt, pch = 4, col = "red", )
```




```
# dev.off()

# surface2a = s3dPlot(grid, fn='2a_surface.pdf')

# Linear regression to see linear data trend pdf('2a_linearPlane.pdf')
plane = lm(dt$z ~ dt$x + dt$y)
planeplot = scatterplot3d(dt$x, dt$y, dt$z, angle = 35, xlab = "x", ylab = "y", zlab = "z")
planeplot$plane3d(plane)
```




```
# dev.off()

# pdf('2a_obsDensity.pdf')
heat <- as.ppp(cbind(dt$x, dt$y), c(0, 315, 0, 315))
plot(density(heat, adjust = 0.5), main = "Heatmap of observation intensity")
```




```
# dev.off() df = as.data.frame(dt)
```

Explain what we see and answer if a stationary grf is a suitable model

Here we see a decreasing trend in positive y direction, which suggest a non-constant mean function m. Thus, a stationary GRF model may not be suitable.

b) Universal Kriging

Let $X = (X(s_1), ..., X(s_{52}))^T$ be the vector of exact observations at locations $s_i, \quad i \in [1, 52]$. Further assume that X is a GRF modeled by

$$\begin{aligned} E[X(s)] &= g(s)^T \beta \\ Var[X(s)] &= \sigma^2 \\ Corr[X(s), X(s')] &= \rho(||s - s'||), s, s' \in \mathcal{D}. \end{aligned}$$

Since correlation function is clearly isotropic, then so is the covariance function. (means invariant to shift and rotation. Furthermore, cov fnc is the stationary cov fnc. This is a powered exponential model.)

refer to text setup or write in full as done below?

By following the notation in Spatial Statistics and Modeling[1], where

[1]Gaetan & Guyon (2009), p.44. not sure if needed, but an example of footnote.

$$
\begin{aligned}
\vec{z_i} &= (g_1(\vec{s_i}), ..., g_p(\vec{s_i}))^T \\
Z &= [z_1^T, ..., z_{52}^T]^T \\
z_0 &= (g_1(\vec{s_0}), ..., g_p(\vec{s_0}))^T,
\end{aligned}
$$

1. $$\hat{\vec{\beta}} = (Z^T \Sigma^{-1} Z)^{-1} Z^T \Sigma \vec{X}$$

2. $$\hat{\vec{X}_0} = \vec{z}_0 \hat{\vec{\beta}} + \vec{c}^T \Sigma^{-1} (\vec{X} - Z \hat{\vec{\beta}})$$

c is covar fnc mentioned in task 1. Need to mention here?

$$
\begin{aligned}
\Sigma = Cov(\vec{X}) &= \begin{pmatrix} \Sigma_{00} & \Sigma_{01} \\ \Sigma_{10} & \Sigma_{11} \end{pmatrix} \\
\vec{c} = Cov(\vec{X}, X_0) &= \sigma^2 \rho?? \\
\rho(h) = \exp(-(0.05)^{1.5})
\end{aligned}
$$

universal Kriging prediction variance. Does σ^2 need to change for different parameterizations of the expectation function?

Written stuff in overleaf

c) Ordinary kriging

Model as in universal kriging, but with mean $E[X(s)] = g(s)\beta_1 = \beta_1 \Leftrightarrow g(s) = 1, \quad \forall s \in \mathcal{D}$. For the code we chose the range by where the correlation function is $\rho(h) = 0.1 \Leftrightarrow h = 174$ (typically 10-13%)

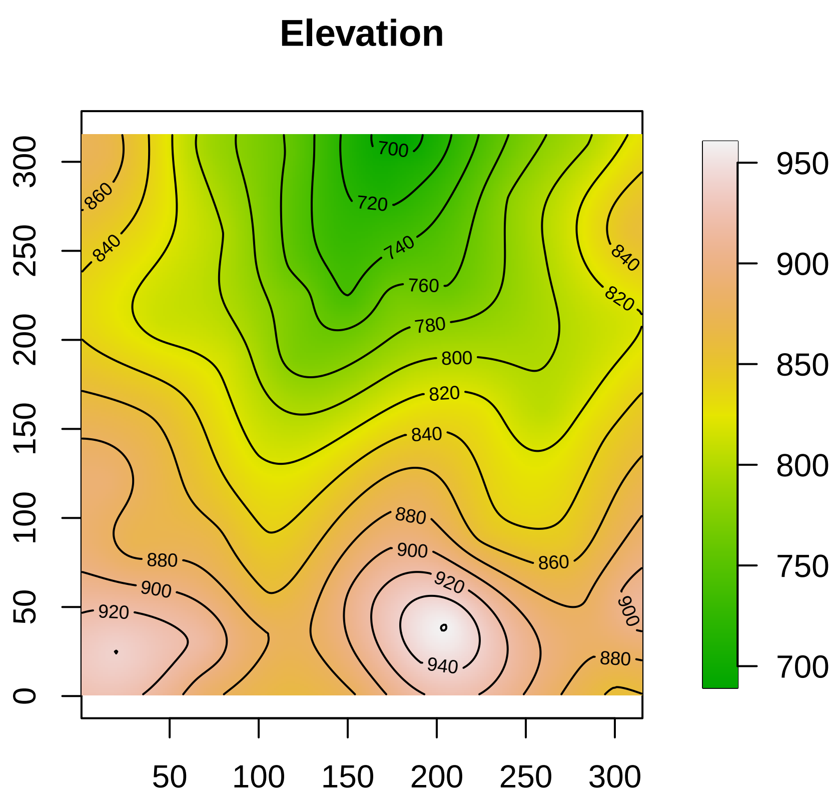
```r
D = seq(1,315,1)
gridOrd = expand.grid(x = D, y = D)
# Spatial Prediction - Ordinary Kriging
spOK = krige.conv(gd, locations=gridOrd,
           krige = krige.control(
             # defaults to ordinary Kriging and constant mean
             cov.model = "powered.exponential",
             cov.pars = c(50^2, 100), # (partial sill, range)
               kappa = 1.5 # Smoothness
           )
           )
```

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

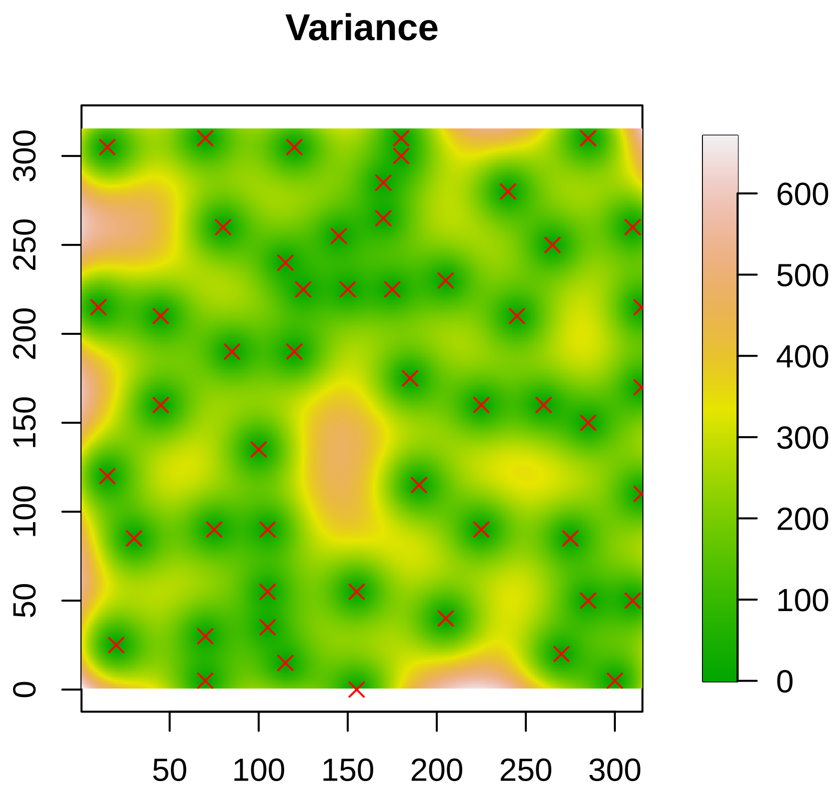
```r
# Beta estimate
spOK$beta.est
```

```
##     beta
## 845.2479
```

```r
predOK = matrix(spOK$predict, ncol = 315)  # to fit image.plot structure
varOK = matrix(spOK$krige.var, ncol = 315)
expImageOK = list(x = D, y = D, z = predOK)
VarImageOK = list(x = D, y = D, z = varOK)
# par(mfrow=(c(1,2)))

# Kriging predictor pdf('2c_ordKrigPred.pdf')
image.plot(expImageOK, col = terrain.colors(500), main = "Elevation", asp = 1)
contour(expImageOK, add = T)
```




```r
# dev.off()

# Associated prediction variance pdf('2c_ordKrigVar.pdf')
image.plot(VarImageOK, col = terrain.colors(500), main = "Variance", asp = 1)
points(dt$x, dt$y, col = "red", pch = 4)
```




```
# dev.off() par(mfrow=(c(1,1)))
```

Comment on results.

d) Universal Kriging

We denote the site $s = (s_1, s_2)$, and we set $n_g = 6$ to be the total number of known functions in the vector-valued function g. Furthermore, we specify each function in g to be all polynomials $s_1^k s_2^l$ for $(k, l) \in (0,0), (1,0), (0,1), (1,1), (2,0), (0,2)$. Thus, the 6-dimensional vector is use array instead

$$g(s) = [1, s_1, s_2, s_1 s_2, s_1^2, s_2^2] \tag{1}$$

The expected value of $X(s)$ is then

$$E[X(s)] = g(s)^T \beta = \beta_1 + \beta_2 s_1 + \beta_3 s_2 \beta_4 s_1 s_2 + \beta_5 s_1^2 + \beta_6 s_2^2 \tag{2}$$

```
gridUniv = expand.grid(x = D, y = D)
spUK     = krige.conv(
  geodata=gd,
  # coords     = cbind(x=x,y=y),
```

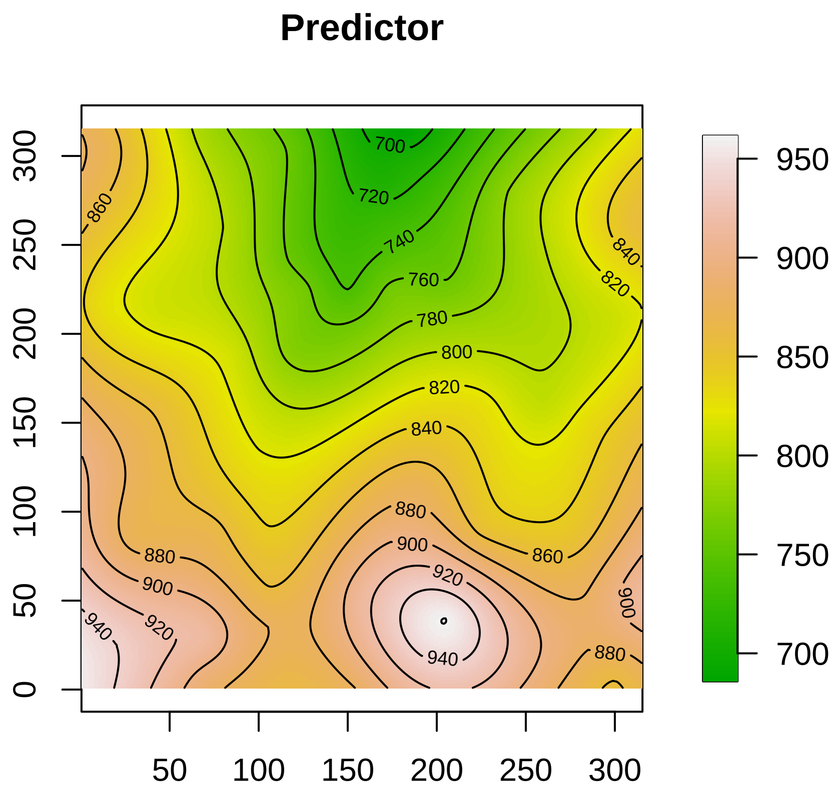
```r
  # data      = z,
  locations = gridOrd,
  krige     = krige.control(
    type.krige = "ok",
#"2nd": beta0 + beta1*x1 + beta2*x2 + beta3*(x1)^2 + beta4*(x2)^2 + beta5*x1*x2
    trend.d    = "2nd",
    trend.l    = "2nd",
    cov.model  = "powered.exponential",
    cov.pars   = c(50^2, 100),
    kappa      = 1.5
  )
)
```

```
## krige.conv: model with mean given by a 2nd order polynomial on the coordinates
## krige.conv: Kriging performed using global neighbourhood
```

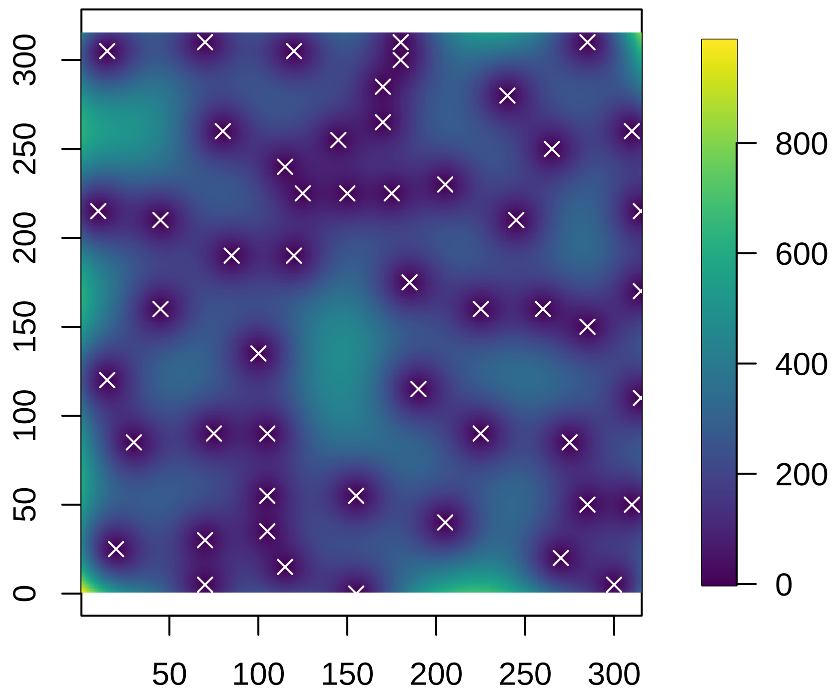
```r
# Estimated beta Universal Kriging
betaHat = (spUK$beta.est)
o2 = c(1, 2, 3, 6, 4, 5)  # order of betas corresponding to task description
betaHat[o2]
```

```
##         beta0         beta1         beta2         beta5         beta3
##  9.416681e+02 -1.109082e+00 -5.181555e-02  1.326022e-04  3.005938e-03
##         beta4
## -8.672447e-04
```

```r
# Universal Kriging predictor
predUK = matrix(spUK$predict, ncol = 315)
imageUK = list(x = D, y = D, z = predUK)
# pdf('2d_univKrigPred.pdf')
image.plot(imageUK, col = terrain.colors(100), main = "Predictor", asp = 1)
contour(list(x = D, y = D, z = predUK), add = T)
```




```r
# points(dt$x,dt$y, pch=4, col = 'red') dev.off()

# Associated variance predictions
sigmaHat = matrix(spUK$krige.var, ncol = 315)  # Predicted variance
# pdf('2d_univKrigVar.pdf')
image.plot(list(x = D, y = D, z = sigmaHat), col = viridis(200), asp = 1)
points(dt$x, dt$y, pch = 4, col = "white")
```




```
# dev.off() surface3d2 = s3dPlot(imageUK, fn='2d_univKrigSurface.pdf')
```

e)

```
D = seq(1,315,1)
gridUniv = expand.grid(x = D, y = D)
spUK     = krige.conv(geodata = gd,
  locations = gridUniv,
  krige     = krige.control(
    type.krige = "ok",
    #"2nd": beta0 + beta1*x1 + beta2*x2 + beta3*(x1)^2 + beta4*(x2)^2 + beta5*x1*x2
    trend.d    = "2nd",
    trend.l    = "2nd",
    cov.model  = "powered.exponential",
    cov.pars   = c(50^2, 100),
    kappa      = 1.5
  )
)
```

```
## krige.conv: model with mean given by a 2nd order polynomial on the coordinates
## krige.conv: Kriging performed using global neighbourhood
```

```r
# These are not needed???!!!
predUK = matrix(spUK$predict, ncol = 315)
imageUKe = list(x = D, y = D, z = predUK)
# # pdf('2e_ordKrigPred.pdf') # image.plot(D,D,z=predUK,
# col=terrain.colors(200), asp=1) # contour(D,D,predUK, add = T) #
# points(dt$x,dt$y, col='red', pch=4) dev.off() # Associated prediction
# variance sigmaHat = matrix(spUK$krige.var,ncol = 315)
# pdf('2e_ordKrigVar.pdf') image.plot(x=D,y=D,z=sigmaHat, col=viridis(200),
# asp=1) points(gd[[1]][,1],gd[[1]][,2], pch=4, col = 'white') dev.off()
```

```r
# deleteResults = c(q1= pnorm(850, mean = r0_hat, sd = sqrt(sigma_r_hat2),
# lower.tail = FALSE), q2= qnorm(0.9, mean = r0_hat, sd = sqrt(sigma_r_hat2)) )


vars0 = matrix(spOK$krige.var, ncol = 315)[100, 100]
x0 = matrix(spOK$predict, ncol = 315)[100, 100]
s0 = predOK[100, 100]
# P(X((100,100))>850)
pnorm(850, mean = x0, sd = sqrt(vars0), lower.tail = F)
```

```
## [1] 0.04221563
```

```r
# critical value for which the true val is below with a prob of 0.9
qnorm(0.9, x0, sqrt(vars0))
```

```
## [1] 847.1265
```

We have assumed normally distributed errors ε for the linear regression model. Thus, our predictions $\hat{X}(s_o) = \hat{X}((100,100)^T) =$ are normally distributed about the mean $E\hat{X}_0$. Specifically for $s_0 = (100,100)^T$ we have $\hat{X}((100,100)^T) =$