STAT 226 Homework 3

Sampson Mao

Problem 1

Extract the observations corresponding to the Dakotas and Minnesota for August 6 – 8, 2017.

```
library(rgdal); library(sp); library(maps); library(maptools); library(dplyr)
source("lonlat_smap_data_script.R")
# mydata contains the data
names(mydata) = c("longitude", "latitude", "aug6",
                  "aug7", "aug8", "aug9",
                  "aug10", "aug11", "aug12")
# The following code to convert coordinates to state names is from
# https://stackoverflow.com/questions/8751497/latitude-longitude-coordinates-to-state-code-in-r
lonlat_to_state_sp <- function(pointsDF) {
    # Prepare SpatialPolygons object with one SpatialPolygon
    # per state (plus DC, minus HI & AK)
    states <- map('state', fill=TRUE, col="transparent", plot=FALSE)
    IDs <- sapply(strsplit(states$names, ":"), function(x) x[1])
    states_sp <- map2SpatialPolygons(states, IDs=IDs,
                     proj4string=CRS("+proj=longlat +datum=WGS84"))
    # Convert pointsDF to a SpatialPoints object
    pointsSP <- SpatialPoints(pointsDF,
                    proj4string=CRS("+proj=longlat +datum=WGS84"))
    # Use 'over' to get _indices_ of the Polygons object containing each point
        indices <- over(pointsSP, states_sp)
    # Return the state names of the Polygons object containing each point
    stateNames <- sapply(states_sp@polygons, function(x) x@ID)
    stateNames[indices]
}
```

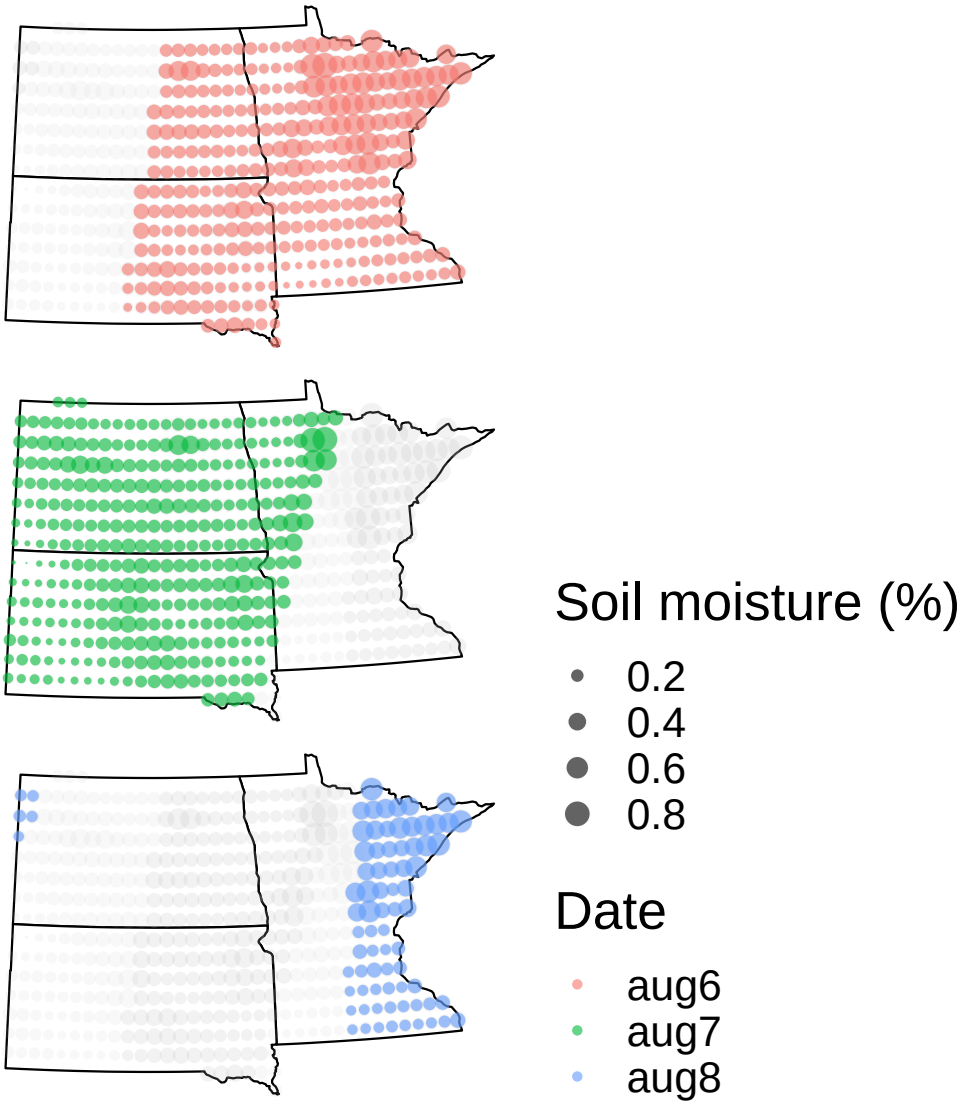
```
mydata$state = lonlat_to_state_sp(mydata[,1:2])

dak.min = mydata %>%
  dplyr::select(longitude, latitude, state, aug6, aug7, aug8) %>%
  filter(state %in% c("north dakota", "south dakota", "minnesota"))
head(dak.min)
```

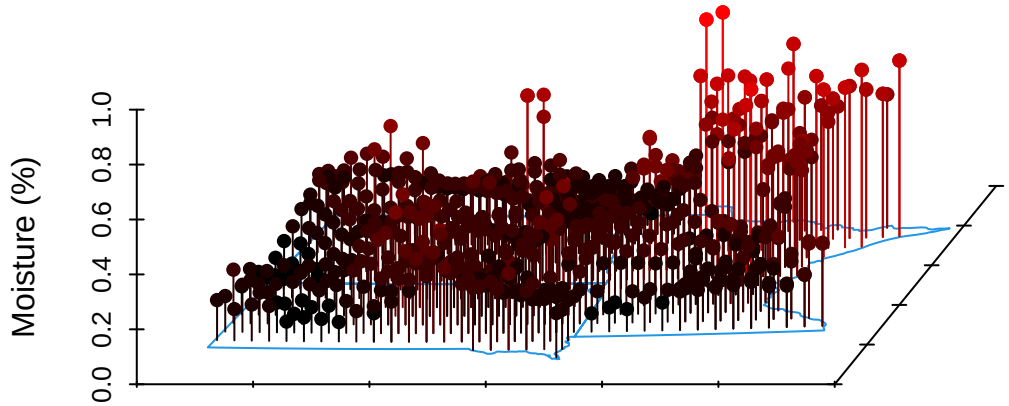
```
##        longitude latitude        state aug6      aug7      aug8
## 16965 -102.88382 49.00464 north dakota   NA 0.1473764        NA
## 16966 -102.51037 49.00464 north dakota   NA 0.1549300        NA
## 16967 -102.13693 49.00464 north dakota   NA 0.1395647        NA
## 16986  -95.04149 49.00464    minnesota   NA        NA        NA
## 17304 -104.00415 48.57916 north dakota   NA 0.1952672 0.1740514
## 17305 -103.63071 48.57916 north dakota   NA 0.1979792 0.1719063
```

Problem 2

Perform a graphical exploration of the data. Is there evidence of a first or second order trend function of location? Is there evidence that a transformation is needed in order to make the data closer to normality?

The code for this section is located at the end of this document

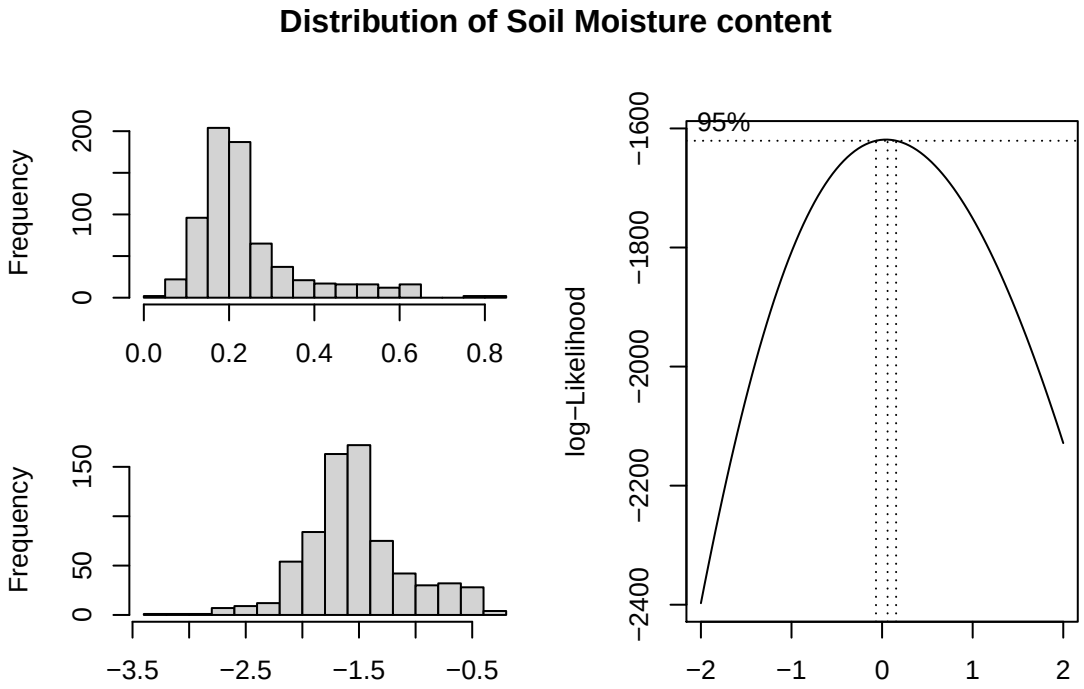




There looks to be evidence of a second order trend as a function of location. In the balloons plot, we can see that the northeastern part of minnesota has a much higher soil moisture content compared to the rest of the state and the Dakotas. For the rest of the region, there is also a high or low moisture content depending on location, though it is not as large as northeast Minnesota. A constant trend would not be able to explain the sudden increase as we move into the more northeastern regions. So for our covariates matrix D, we have

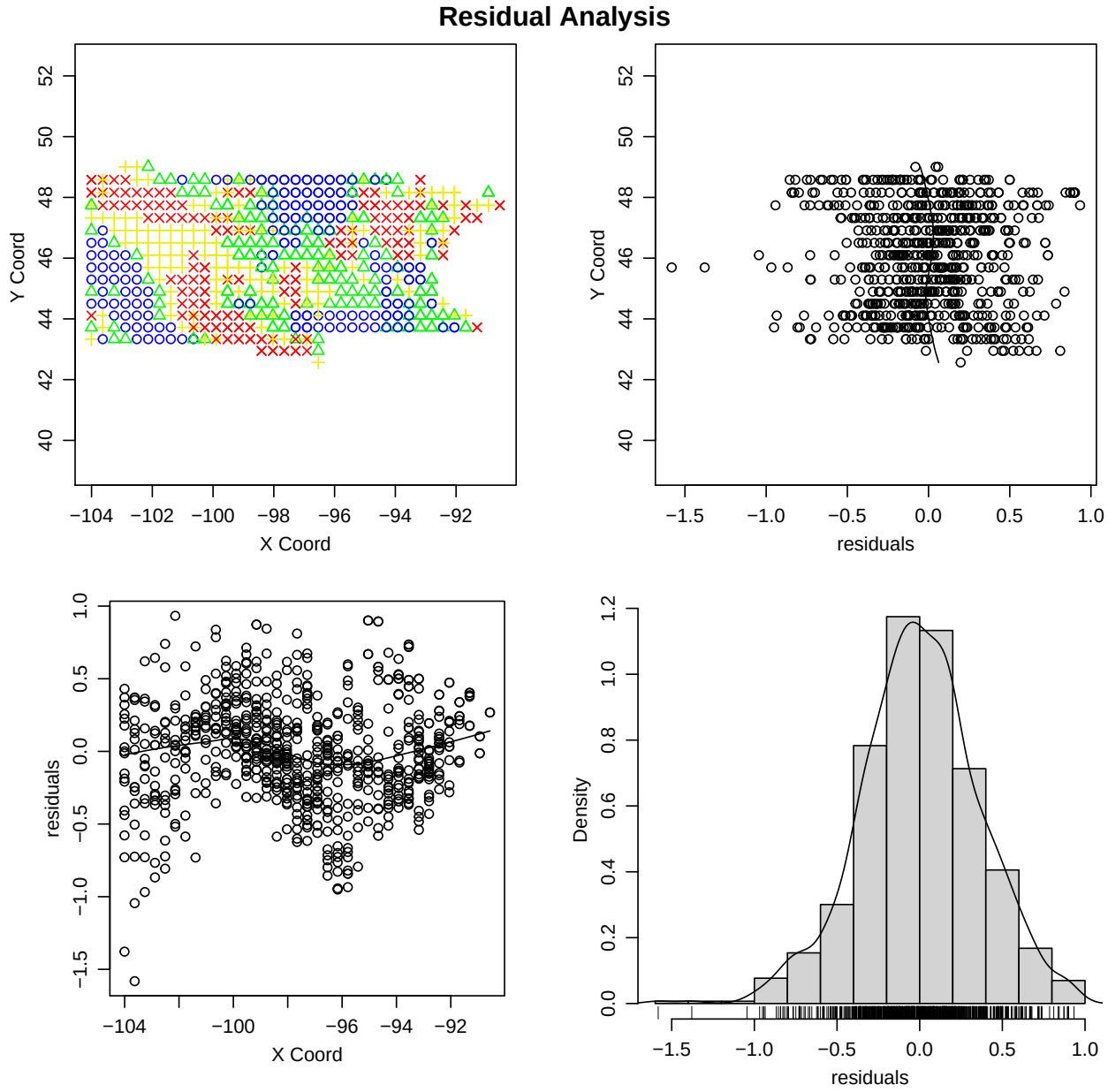
$$D = \begin{bmatrix} \mathbf{1} & \mathbf{d_1} & \mathbf{d_2} & \mathbf{d_1}^2 & \mathbf{d_2}^2 & \mathbf{d_1 d_2} \end{bmatrix}$$

where $\mathbf{d_1}, \mathbf{d_2}$ are the vector of longitudes and latitudes respectively. For the rest of the homework, the jittered coordinates are used, except for the variogram plots so that we can estimate the semivariance for 0 distance. Some matrix calculations may not be possible with un-jittered data because of having 0 distance. The jitter will be very small, with the maximum jitter value being 10^{-7}.



When plotting the histogram for the data, we can see that it is skewed to the right, which indicates non-normality (upper left hand plot). The right hand plot is the Box-Cox fit of the data, using D is the covariates matrix, for various values of λ. We can see that very small values of λ, including 0, will give a transformation

of the data that make it approximately normal. So for the rest of the analysis, I will use the log transform, which is plotted on the bottom left. The distribution is not as skewed as it was originally though the right tail is still thicker.



In the above plots created by geoR, we can see that the residuals (trend removed data) are normally distributed, which is a good sign for our model. Furthermore, there appears to be a near-costant variance in both the X and Y (longitude and latitude) directions, shown in the bottom left and top right plots. Across the states, there seem to be more variability than up and down the states based on these 2 plots. Finally, in the top left plot, we can see that the values are more consistent. The red x denotes the fourth quartile, which we see exists in the Dakotas now rather than just northeast Minnesota. Similarly for the yellow +, green triangles, and blue circles for the third, second, and first quartiles respectively.

Problem 3

Obtain the residuals after fitting the trend function resulting from the previous question, if any. Plot the variogram. Explore possible anisotropies using a directional variogram.

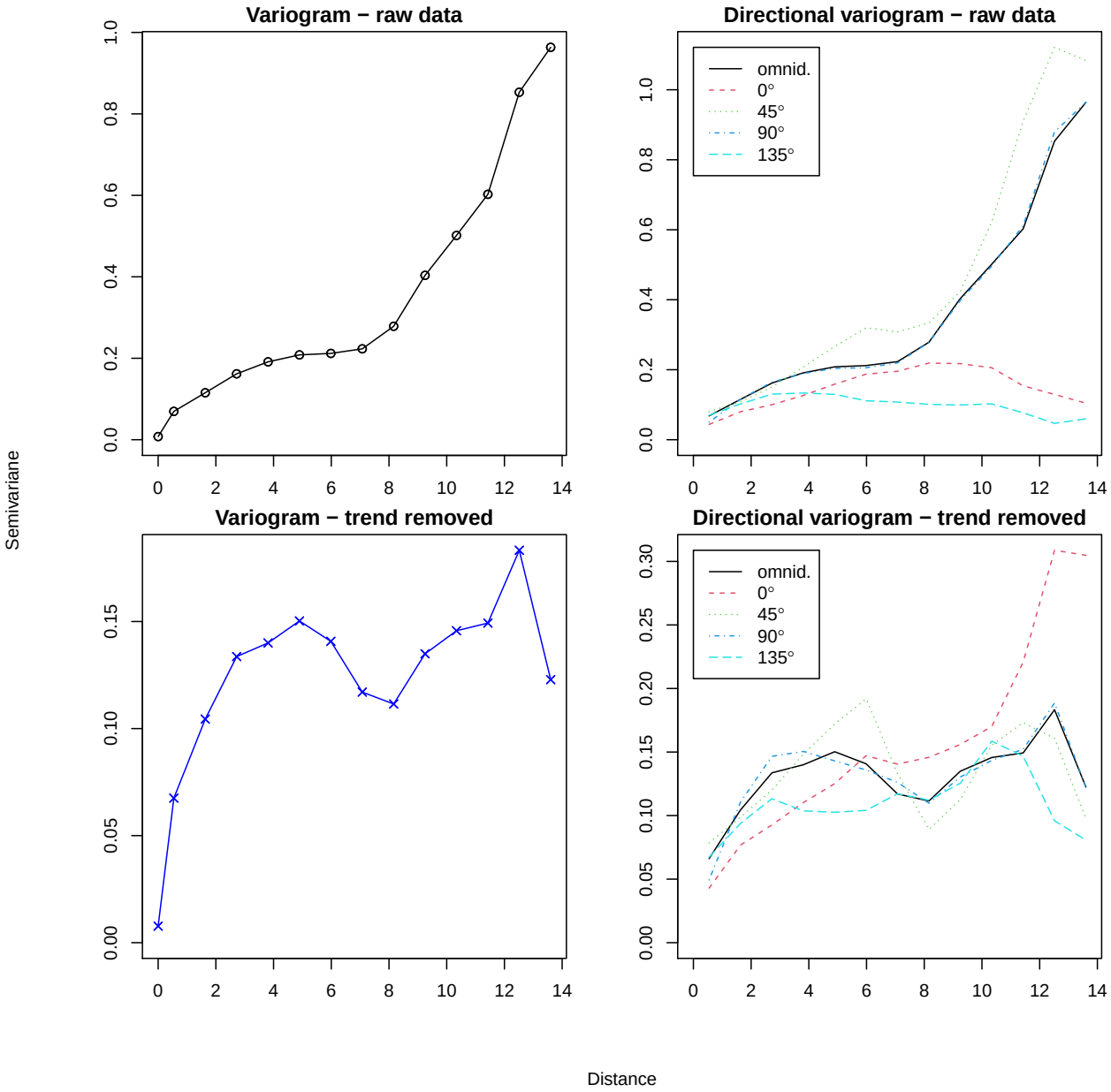
In classical parameter estimation we assume that the mean function is a linear combination of spatial explanatory variabiles. For our data, this would apply to the log of the mean function as we saw before. Here, we don't know about any existing covariance structure, so we can use ordinary least squares to estimate the trend.

$$\begin{aligned} log(\mu(s)) &= \beta_0 + \beta_1 \mathbf{d_1} + \beta_2 \mathbf{d_2} + \beta_3 \mathbf{d_1^2} + \beta_4 \mathbf{d_2^2} + \beta_5 \mathbf{d_1 d_2} \\ D^T D\hat{\beta} &= D^T \log(Y) \\ R &= \log(Y) - D\hat{\beta} \end{aligned}$$

```
beta.hat = solve(t(D) %*% D, t(D) %*% log(Y))
residuals = log(Y) - D %*% beta.hat
```

```
## variog: co-locatted data found, adding one bin at the origin
```

```
## variog: co-locatted data found, adding one bin at the origin
```



The plots above show the variograms and directional variograms for the trend-present (first row) and trend-removed (second row) data. Removing the trend decreased the semivariance significantly. With the trend, the variogram did not seem to plateau and reached as high as 1. After fitting the quadratic trend, the plot reaches the max at 13 with a semivariance of around 0.2, though the plot becomes flat early on. The fitted trend seems to account for much of the total variance in the data. Meanwhile, the directional variogram for the residuals show that the spatial variation is relatively similar at 45, 90, and 135 degrees above the horizontal axis in our map. Meanwhile, across states, or 0 degrees, there is potential anisotropy. For the trend-present plot, it is more clear that there is anisotropy at 45 and 90 degrees, which we have discussed on problem 2. In summary, anisotropy all but disappeared after accounting for trend.

Problem 4

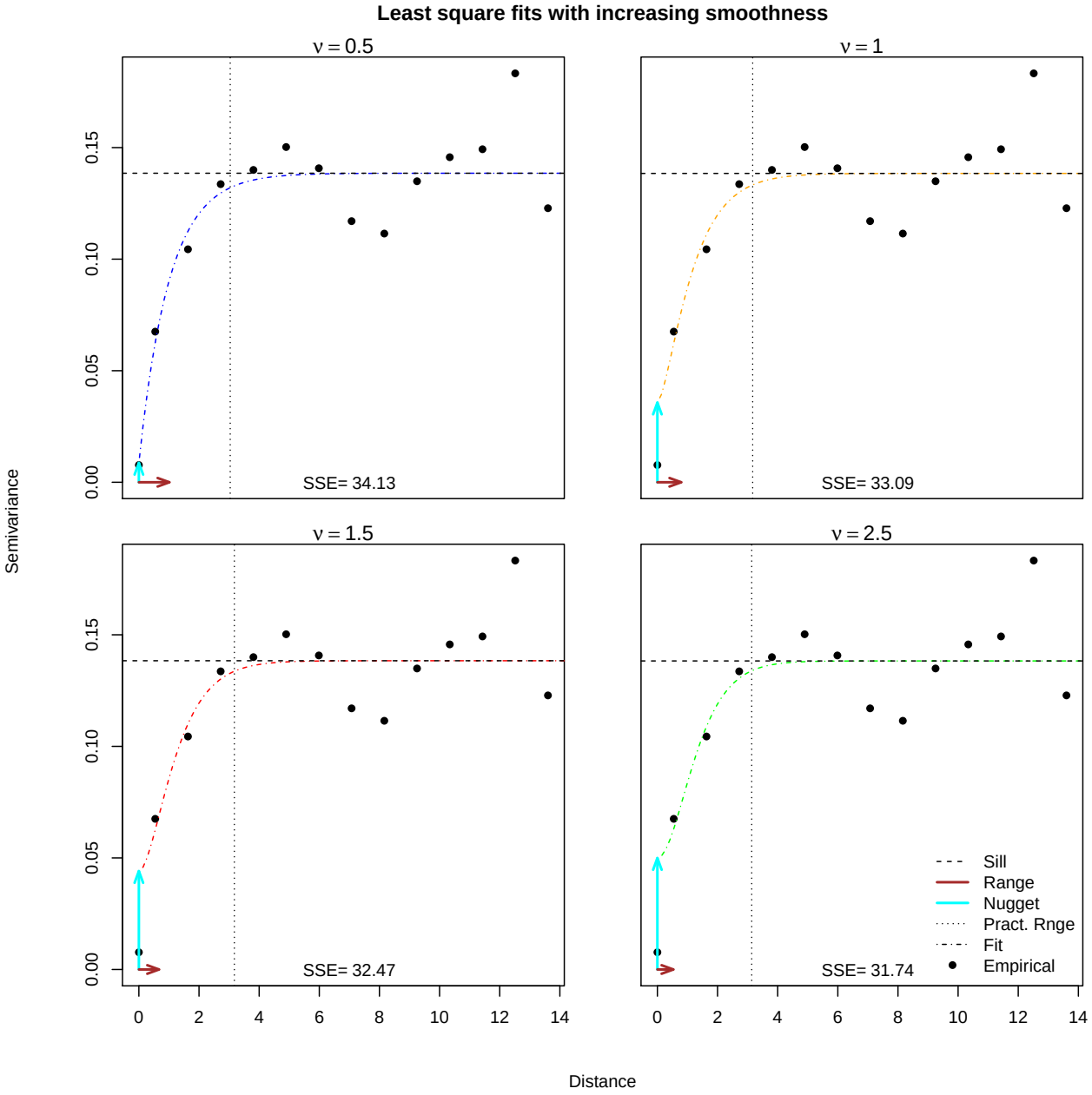
Use least squares to fit the covariograms in the Matèrn family with smoothness equal to .5; 1; 1.5; 2.5. Plot the results. Use the plots and the values of the LSE to select the best fit.

We have the residuals (binned) r_k^* and we want to fit a curve $V(x_k^*; \sigma^2, \phi, \tau^2)$, with corresponding distance x_k^*. A n_k-weighted objective function is

$$S(\sigma^2, \phi, \tau^2) = \sum_{k=1}^{n} n_k \left\{ r_k^* - V(x_k^*; \sigma^2, \phi, \tau^2) \right\}^2$$

where n_k is the number of points used to obtain the binned residual v_k. Minimizing S with respect to the parameters would give us the estimated curve for our data. In this case, we would like V to be the Matern covariance function with nugget τ^2. This is implemented in the `variofit` function in `geoR`

```
## variog: co-locatted data found, adding one bin at the origin
```




```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Feb 10, 2021 - 18:21:17

Table 1:

	Smoothness	Nugget	SigmaSq	Phi	Practical range
1	0.500	0.009	0.130	1.014	3.038
2	1	0.036	0.103	0.792	3.168
3	1.500	0.044	0.094	0.669	3.176
4	2.500	0.050	0.088	0.530	3.136

All of the plots look similar with longer distance. At shorter distances, they differ in the value of the nugget and range, with $\nu = 2.5$ having the largest nugget and smallest range, and $\nu = 0.5$ having the smallest nugget and largest range. The n_k-weighted sum of squares is the lowest for the smoothest option, though it differs from the roughest option only by 2.39. The smoothest option does seem to have the best fit based on this statistic. Fortunately, we have duplicate points, which from the plot shows us that the nuggest is not as large as estimated with the best fit curve. Based on this I would say the best fit is the case where $\nu = 0.5$.

Problem 5

Plot the likelihood function for the sill and the range corresponding to each of the correlations in the previous point. If a nugget is needed, you can plug an estimated value.

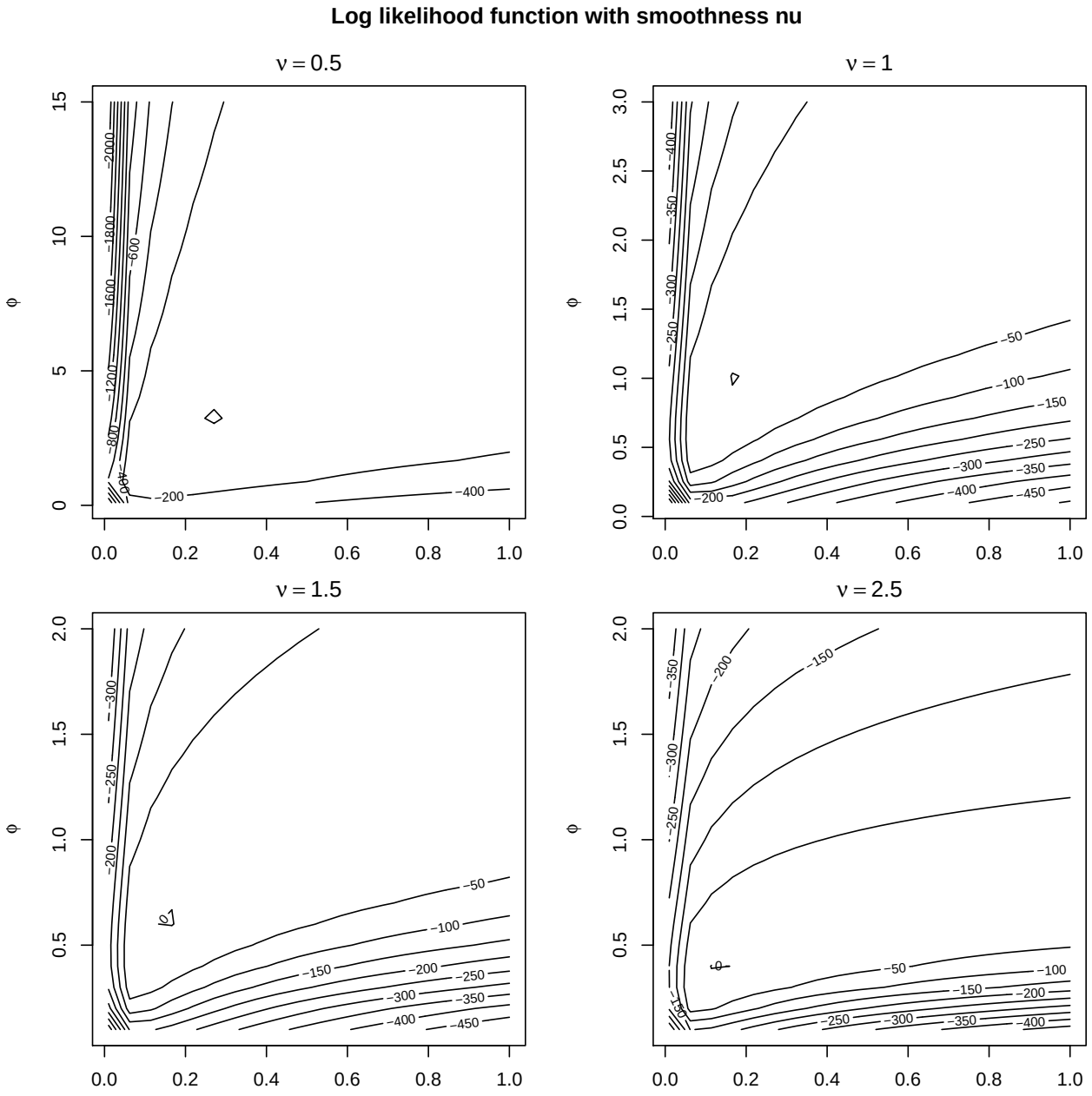
Since we removed the trend, the likelihood is

$$L(\psi) \propto |V(\psi)|^{-1/2} \exp\left\{-\frac{1}{2} R^T V(\psi)^{-1} R\right\}$$
$$\log(L(\psi)) \propto -\frac{1}{2}\log|V(\psi)| - \frac{1}{2} R^T V(\psi)^{-1} R$$

We have the entries of V given by $V(\psi)_{ij} = C(s_i, s_j; \sigma^2, \phi) + \tau^2 \delta_{ij}$, where C is the Matern covariance function. This is implemented in the R code below

```
logL = function(sigma2, phi, nu, tau2) {
  V.mat = varcov.spatial(in_geor_jitter$coords, cov.pars=c(sigma2, phi), nugget=tau2, kappa=nu)
  V = V.mat$varcov

  n = dim(D)[1]

  L = t(chol(V))
  Z = forwardsolve(L, residuals)

  G = t(Z) %*% Z

  log.det = 2*sum(log(diag(L)))

  density = -n/2*log(2*pi) -0.5*log.det - 0.5*G

  return(density)
}
```

```
vect.logL = Vectorize(logL, vectorize.args=c("sigma2", "phi"))
```



The estimated nugget from the previous problem was used. Like we saw in class, for higher smoothness, the likelihood plot becomes more banana shaped, indicating stronger correlation between the parameters. The function becomes flatter near the maximum point.

Problem 6

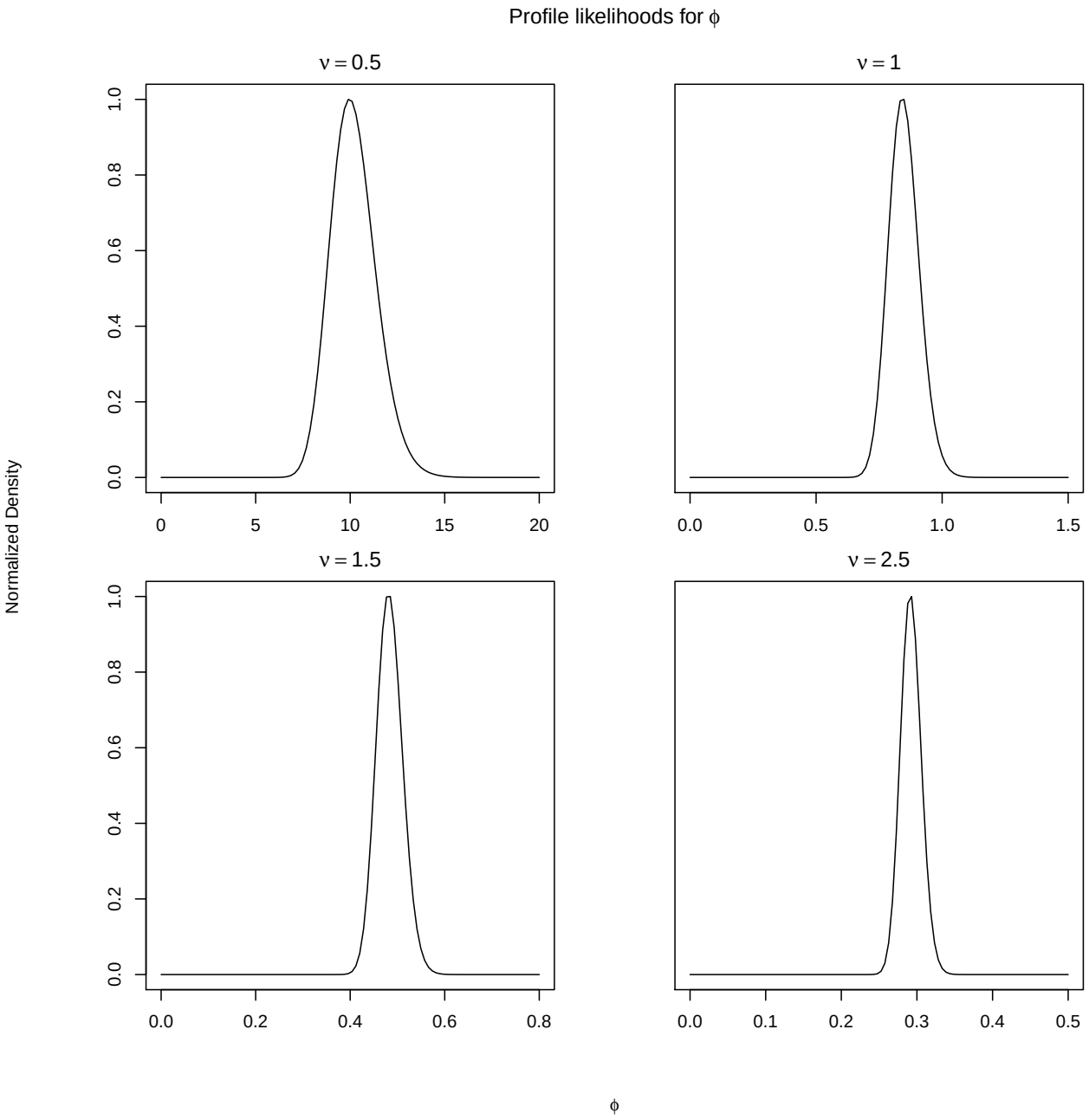
Plot the marginal likelihood for the range parameter for each of the examples above.

The profile likeilihood for ϕ is

$$L(\phi) \propto |V(\phi)|^{-1/2} |D^T V(\phi)^{-1} D|^{-1/2} (S(\phi)^2)^{-(n-k)/2}$$
$$\log(L(\phi)) \propto -\frac{1}{2} \log|V(\phi)| - \frac{1}{2} \log|D^T V(\phi)^{-1} D| - \frac{n-k}{2} \log(S(\phi)^2)$$
$$S(\phi)^2 = R V(\phi)^{-1} R$$

```
logprofL = function(phi, nu, tau2) {
  V.mat = varcov.spatial(in_geor_jitter$coords, cov.pars=c(1, phi), nugget=tau2, kappa=nu)
  V = V.mat$varcov

  n = dim(D)[1]
  k = dim(D)[2]

  L = t(chol(V))
  Z = forwardsolve(L, residuals)
  W = forwardsolve(L, D)

  S2 = t(Z) %*% Z
  DVD = t(W) %*% W

  E = t(chol(DVD))

  logdet.V = 2*sum(log(diag(L)))
  logdet.DVD = 2*sum(log(diag(E)))

  density = -0.5*logdet.V -0.5*logdet.DVD - (n-3)/2 * log(S2)

  return(density)
}

vect.logprofL = Vectorize(logprofL, "phi")
```

The plots below show the marginal likelihood for the range parameter where the nuggest is fixed at the estimated values from problem 4. The density has been normalized by the maximum value and exponentiated so that the peaks are more visible. Except for $\nu = 0.5$, all of the maxima seem to coincide with the ones we saw in the contour plots. Since we had replicate data, the 2D plot for the likelihood may look more irregular with multiple maximum points for $\nu = 0.5$. The profile likelihood is a more regular function. This could explain why the profile likelihood shows that the maximum for ϕ is around 10.



Appendix

Code for problem 2

```
library(usmap)
library(ggplot2)
library(reshape2)

transformed_data = usmap_transform(dak.min)
melted = melt(
  transformed_data,
  id.vars = c("longitude", "latitude", "state", "longitude.1", "latitude.1"),
```

```
  variable.name = "date"
)

plot_usmap(include = c("ND", "SD", "MN")) +
  geom_point(
    data = melted
    %>% dplyr::select(-date),
    aes(x = longitude.1, y = latitude.1, size = value),
    alpha = 0.1,
    color = "grey"
  ) +
  geom_point(
    data = melted,
    aes(
      x = longitude.1,
      y = latitude.1,
      color = date,
      size = value
    ),
    shape = 16,
    alpha = 0.6
  ) +
  facet_wrap( ~ date, ncol = 1) +
  labs(color = "Date") +
  labs(size = "Soil moisture (%)") +
  scale_size_continuous(range = c(1e-5, 4)) +
  theme(
    legend.position = "right",
    legend.title = element_text(size = 20),
    legend.text = element_text(size = 16),
    strip.background = element_blank(),
    strip.text.x = element_blank()
  )
```

```
library(scatterplot3d)
soil = melted[complete.cases(melted),]
x = soil$longitude.1
y = soil$latitude.1
z = soil$value
borders = us_map(include = c("ND", "SD", "MN"))
# https://stackoverflow.com/questions/9946630/colour-points-in-a-plot-differently-depending-on-a-vector
rbPal <- colorRampPalette(c('black','red'))
z.col = rbPal(10)[as.numeric(cut(z,breaks = 10))]
par(mar=c(1,2,0,1))
s = scatterplot3d(borders$x, borders$y, rep(0,length(borders$x)),
                  type="l", box=F, axis=T, grid=F,
                  zlim=c(0,1), color=4, angle=65,
                  x.ticklabs="", y.ticklabs="",
                  xlab="", ylab="", zlab="Moisture (%)")
s$points3d(x,y,z, type="h", pch=16, col=z.col)
title("Soil Moisture in the Dakota-Minnesota region", line=-1)
```

```
library(geoR)
# Raw data
```

```r
soil_raw = soil[,c(1,2,7)]

# Averaging replicates
avgd.data = data.frame(dak.min[,1:2], value = rowMeans(dak.min[,4:6], na.rm = T))
soil_avg = avgd.data[complete.cases(avgd.data),]

# Jittering replicates
soil_jitter = data.frame(jitterDupCoords(soil[,1:2], max=1e-7), value=soil$value)

# Creating design matrix D
Y = soil_raw$value
n = length(Y)
lon = soil_raw$longitude
lat = soil_raw$latitude
D = cbind(rep(1,n), lon, lat, lon^2, lat^2, lon*lat)

# Boxcox comparison
layout.matrix = matrix(c(1,2,3,3),2,2)
layout(layout.matrix)
par(oma = c(5, 4, 2, 0) + 0.1,
    mar = c(2, 4, 2, 1) + 0.1)
hist(Y, main="",
     xlab="Soil moisture (%)")
hist(log(Y), main="",
     xlab="log(Soil Moisture)",
     breaks=14)
MASS::boxcox(lm(Y ~ D))

title("Distribution of Soil Moisture content", outer=T)
```

```r
in_geor = list(coords=soil_raw[,1:2], data=soil_raw$value)
in_geor_avg = list(coords=soil_avg[,1:2], data=soil_avg$value)
in_geor_jitter = list(coords=soil_jitter[,1:2], data=soil_jitter$value)

par(oma = c(5, 4, 3, 0) + 0.1,
    mar = c(2, 4, 2, 1) + 0.1)
plot.geodata(in_geor, lowess=T, trend=D, lambda=0)
title("Residual Analysis", outer=T)
```

Code for problem 3

```r
par(pty="s", mfrow=c(2,2))

vgm.trend = variog(in_geor, lambda=0, messages=F)
vgm4.trend = variog4(in_geor_jitter, lambda=0, messages=F)
vgm.notrend = variog(in_geor, lambda=0, trend="2nd", messages=F)
vgm4.notrend = variog4(in_geor_jitter, lambda=0, trend="2nd", messages=F)

par(mfrow = c(2, 2),
    oma = c(5, 4, 2, 0) + 0.1,
    mar = c(2, 0.5, 1.5, 0.5) + 0.1)

plot(vgm.trend, main="Variogram - raw data")
```

```
lines(vgm.trend)
plot(vgm4.trend, omni=T)
title("Directional variogram - raw data")

plot(vgm.notrend, col="blue", pch=4, main="Variogram - trend removed")
lines(vgm.notrend, col="blue", pch=4)
plot(vgm4.notrend, omni=T)
title("Directional variogram - trend removed")
title(xlab="Distance", ylab="Semivariane", outer=T)
```

Code for problem 4

```
vgm = variog(in_geor, lambda=0, trend="2nd", nugget.tolerance = 0, messages=F)

init.pars = c(0.15, 2)

fit0.5 = variofit(vgm, kappa=0.5, cov.model="matern", ini.cov.pars = init.pars,
                  limits=c(0,10), messages=F)
fit1 = variofit(vgm, kappa=1, cov.model="matern", ini.cov.pars = init.pars,
                limits=c(0,10), messages=F)
fit1.5 = variofit(vgm, kappa=1.5, cov.model="matern", ini.cov.pars = init.pars,
                  limits=c(0,10), messages=F)
fit2.5 = variofit(vgm, kappa=2.5, cov.model="matern", ini.cov.pars = init.pars,
                  limits=c(0,10), messages=F)

par(mfrow = c(2, 2), pty="s",
    oma = c(5, 4, 2, 0) + 0.1,
    mar = c(1, 1, 1, 1) + 0.1)
plot(vgm.notrend, pch=16, main=bquote(nu==0.5), xlab="", ylab="", xaxt='n')
lines(fit0.5, col="blue", lty=4)
abline(v=fit0.5$practicalRange, lty=3)
abline(h=fit0.5$cov.pars[1]+fit0.5$nugget, lty=2)
arrows(0,0,0, fit0.5$nugget, angle=20, length=0.1, lwd=2, col="cyan")
arrows(0,0,fit0.5$cov.pars[2], 0, angle=20, length=0.1, lwd=2, col="brown")
text(7,0, labels=bquote("SSE="~.(round(fit0.5$value, 2))))

plot(vgm.notrend, pch=16, main=bquote(nu==1.0), xlab="", ylab="", xaxt='n', yaxt='n')
lines(fit1, col="orange", lty=4)
abline(v=fit1$practicalRange, lty=3)
abline(h=fit1$cov.pars[1]+fit1$nugget, lty=2)
arrows(0,0,0, fit1$nugget, angle=20, length=0.1, lwd=2, col="cyan")
arrows(0,0,fit1$cov.pars[2], 0, angle=20, length=0.1, lwd=2, col="brown")
text(7,0, labels=bquote("SSE="~.(round(fit1$value, 2))))

plot(vgm.notrend, pch=16, main=bquote(nu==1.5), xlab="", ylab="")
lines(fit1.5, col="red", lty=4)
abline(v=fit1.5$practicalRange, lty=3)
abline(h=fit1.5$cov.pars[1]+fit1.5$nugget, lty=2)
arrows(0,0,0,fit1.5$nugget, angle=20, length=0.1, lwd=2, col="cyan")
arrows(0,0,fit1.5$cov.pars[2], 0, angle=20, length=0.1, lwd=2, col="brown")
text(7,0, labels=bquote("SSE="~.(round(fit1.5$value, 2))))

plot(vgm.notrend, pch=16, main=bquote(nu==2.5), xlab="", ylab="", yaxt='n')
```

```
lines(fit2.5, col="green", lty=4)
abline(v=fit2.5$practicalRange, lty=3)
abline(h=fit2.5$cov.pars[1]+fit2.5$nugget, lty=2)
arrows(0,0,0,fit2.5$nugget, angle=20, length=0.1, lwd=2, col="cyan")
arrows(0,0,fit2.5$cov.pars[2], 0, angle=20, length=0.1, lwd=2, col="brown")
text(7,0, labels=bquote("SSE="~.(round(fit2.5$value, 2))))
title(main="Least square fits with increasing smoothness",
      xlab="Distance",
      ylab="Semivariance",
      outer=T)
legend("bottomright", legend=c("Sill", "Range", "Nugget", "Pract. Rnge", "Fit", "Empirical"),
       col=c("black", "brown", "cyan", "black", "black", "black"), lty=c(2,1,1,3,4,NA),
       pch=c(NA,NA,NA,NA,NA,16), lwd=c(1,2,2,1,1), bty="n")
```

```
library(stargazer)
smoothness = c(0.5,1,1.5,2.5)
nuggets = c(fit0.5$nugget, fit1$nugget, fit1.5$nugget, fit2.5$nugget)
sigma.phi = rbind(fit0.5$cov.pars, fit1$cov.pars, fit1.5$cov.pars, fit2.5$cov.pars)
p.range = c(fit0.5$practicalRange, fit1$practicalRange, fit1.5$practicalRange, fit2.5$practicalRange)

estimates = data.frame(cbind(smoothness,nuggets, sigma.phi, p.range))
names(estimates) = c("Smoothness", "Nugget", "SigmaSq", "Phi", "Practical range")
stargazer(estimates, summary=F)
```

Code for problem 5

```
sill0.5 = seq(0.01, 1, l=20)
range0.5 = seq(0.1, 15, l=20)

sill1 = seq(0.01, 1, l=20)
range1 = seq(0.1, 3, l=20)

sill1.5 = seq(0.01, 1, l=20)
range1.5 = seq(0.1, 2, l=20)

logdensity0.5 = outer(sill0.5, range0.5, function(x,y) vect.logL(x,y, nu=0.5, tau2=fit0.5$nugget))
logdensity1 = outer(sill1, range1, function(x,y) vect.logL(x,y, nu=1, tau2=fit1$nugget))
logdensity1.5 = outer(sill1.5, range1.5, function(x,y) vect.logL(x,y, nu=1.5, tau2=fit1.5$nugget))
logdensity2.5 = outer(sill1.5, range1.5, function(x,y) vect.logL(x,y, nu=2.5, tau2=fit2.5$nugget))
```

```
par(pty="s", mfrow=c(2,2),
    oma = c(5, 4, 2, 0) + 0.1,
    mar = c(2, 3, 2, 1) + 0.1)
contour(sill0.5, range0.5, logdensity0.5-max(logdensity0.5),
        xlab=expression(sigma^2),
        ylab=expression(phi),
        main=bquote(nu=="0.5"))
contour(sill1, range1, logdensity1-max(logdensity1),
        xlab=expression(sigma^2),
        ylab=expression(phi),
        main=bquote(nu=="1"))
contour(sill1.5, range1.5, logdensity1.5-max(logdensity1.5),
        xlab=expression(sigma^2),
```

```
        ylab=expression(phi),
        main=bquote(nu=="1.5"))
contour(sill1.5, range1.5, logdensity2.5-max(logdensity2.5),
        xlab=expression(sigma^2),
        ylab=expression(phi),
        main=bquote(nu=="2.5"))
title("Log likelihood function with smoothness nu", outer=T)
```

Code for problem 6

```
seq0.5 = seq(0,20,l=100)
seq1 = seq(0,1.5,l=100)
seq1.5 = seq(0, 0.8,l=100)
seq2.5 = seq(0, 0.5,l=100)

density0.5 = vect.logprofL(seq0.5, 0.5, fit0.5$nugget)
density1 = vect.logprofL(seq1, 1, fit1$nugget)
density1.5 = vect.logprofL(seq1.5, 1.5, fit1.5$nugget)
density2.5 = vect.logprofL(seq2.5, 2.5, fit2.5$nugget)
```

```
par(pty="s", mfrow=c(2,2),
    oma = c(5, 4, 2, 0) + 0.1,
    mar = c(2, 1, 2, 1) + 0.1)
plot(seq0.5, exp(density0.5 - max(density0.5)), type="l",
    xlab=expression(phi),
    main=bquote(nu=="0.5"),
    ylab="")
plot(seq1, exp(density1 - max(density1)), type="l",
    xlab=expression(phi),
    main=bquote(nu=="1"),
    yaxt="n",
    ylab="")
plot(seq1.5, exp(density1.5 - max(density1.5)), type="l",
    xlab=expression(phi),
    main=bquote(nu=="1.5"),
    ylab="")
plot(seq2.5, exp(density2.5 - max(density2.5)), type="l",
    xlab=expression(phi),
    main=bquote(nu=="2.5"),
    yaxt="n",
    ylab="")
title(bquote("Profile likelihoods for"~phi),
    ylab="Normalized Density",
    xlab=bquote(phi), outer=T)
```