Geostatistics

Alexander J Ohrt, Mikel, Victor (add apellidos and order alphabetically)

22 oktober, 2021

This is the first assignment in Spatial Epidemiology at UPC, fall 2021. It is in the topic of Geostatistics. Don't know if we should write anything else in this "intro".

Load libraries and data

```
library(geoR)
library(sm)
library(gstat)

coordinates <- read.table("poly84.txt", head = TRUE, sep = "\t", dec = ".")
elevs <- read.table("elevationsIslet.txt", head = TRUE, sep = "\t", dec = ".")
head(elevs)
```

```
#>           x          y data
#> 1  98.57754  0.8906123 17.6
#> 2 113.59737 19.5085930 27.1
#> 3 110.53511 39.6851815 16.4
#> 4 105.27965 28.3568305 -9.5
#> 5  95.20204  8.4319902  6.1
#> 6 102.96788 53.0629008 -0.5
```

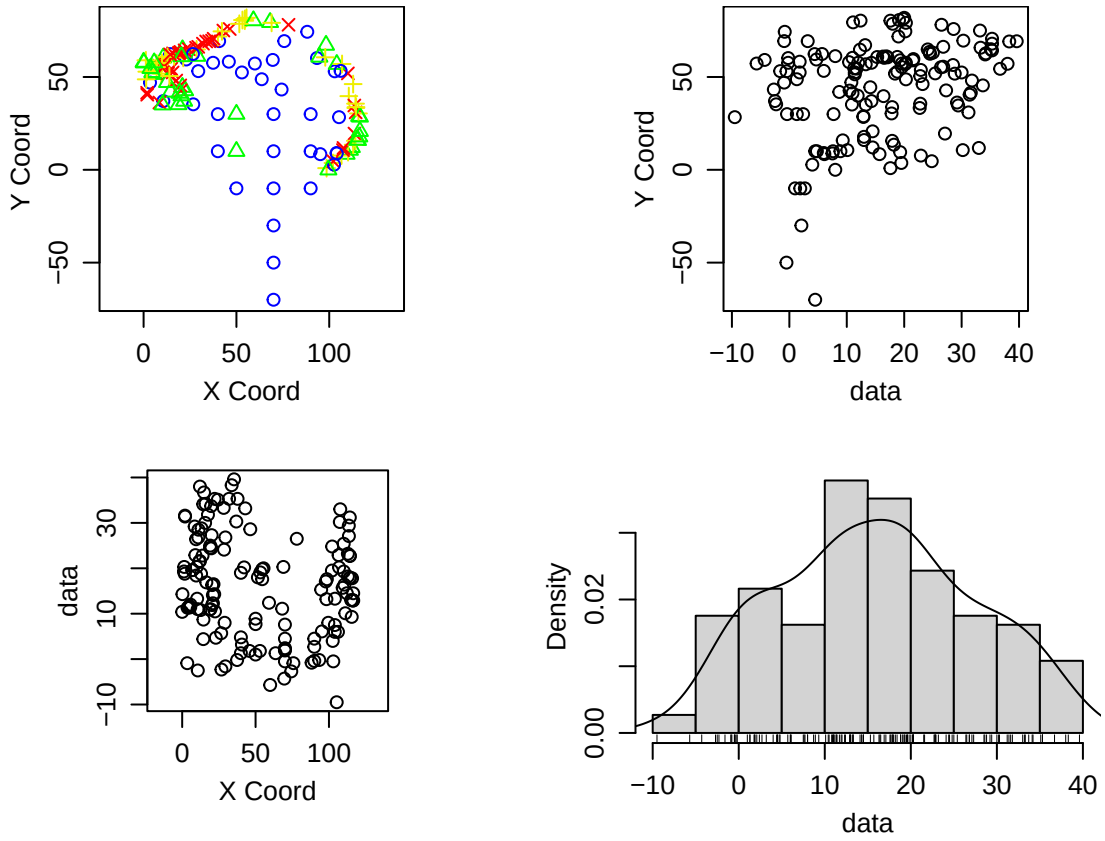
1. Exploration of the large scale variability of the elevation

In this problem we will explore the large scale variability of the elevation data.

```
geoelevs <- as.geodata(elevs)
summary(geoelevs)
```

```
#> Number of data points: 148
#>
#> Coordinates summary
#>            x         y
#> min   0.0000 -70.00000
#> max 116.5471  81.88936
#>
#> Distance summary
#>         min         max
#>   0.2104305 152.5866269
#>
#> Data summary
#>     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
#> -9.50000  7.67500 16.35000 15.97095 24.17500 39.60000
```

```
plot(geoelevs)
```

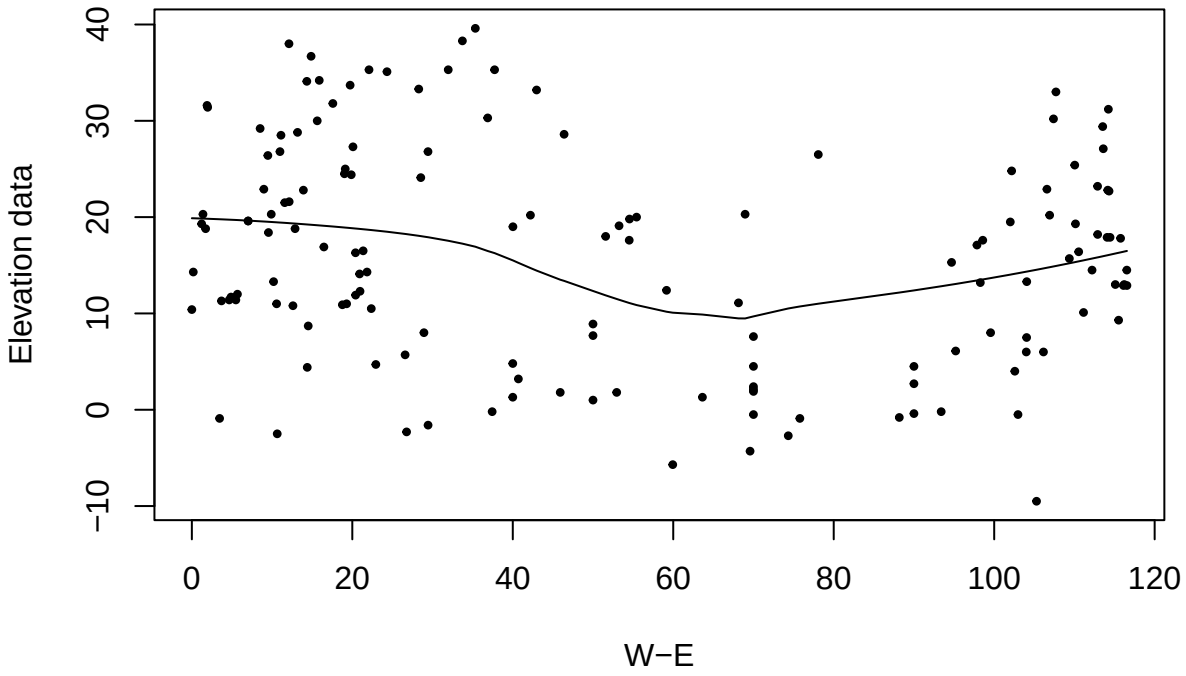



```
# Remember that:
###   (circles) : 1st quartile
###   (triangles) : 2nd quartile
###   (plus)  : 3rd quartile
###   (crosses)  : 4th quartile
```

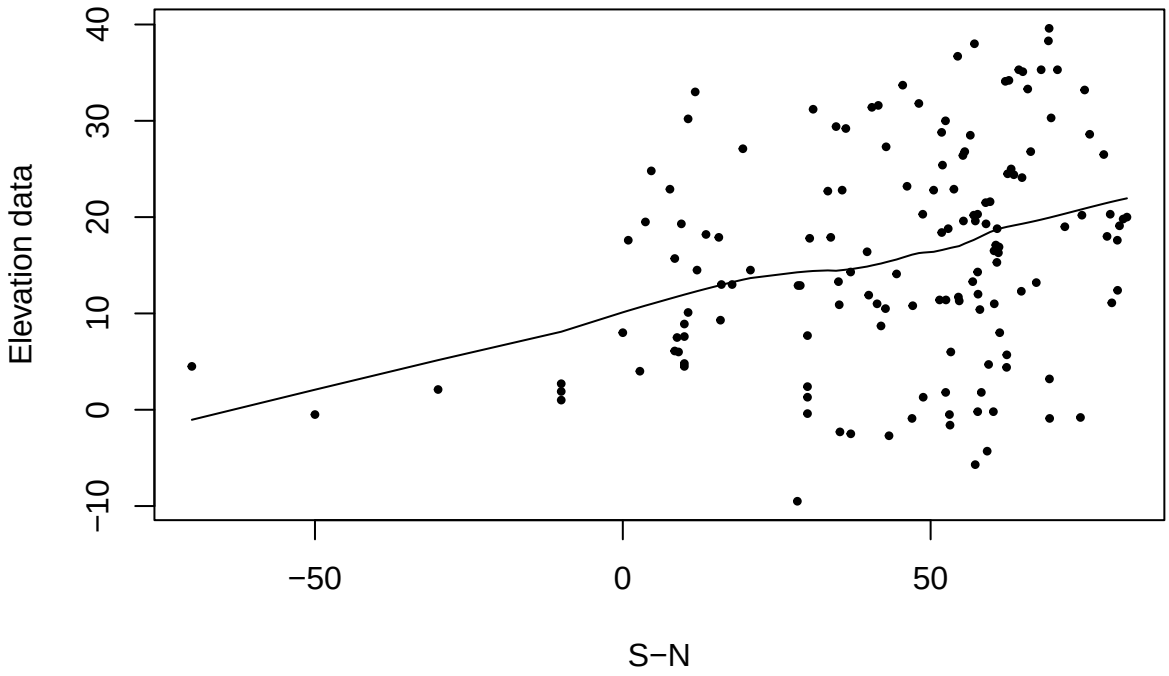
The histogram of the elevations shows that the data is nicely distributed, resembling a normal distribution, which means that we will not need to apply a transformation at this stage. Moreover, the plot in the upper left corner shows that most of the larger values are in the upper left part of the islet, based on the red crosses, which make up the largest 25% of values in the data set. In general, the first quartile in the data is more widespread on the islet, while the rest of the data is mostly spread out around the border of the islet. Note that the plot in the lower left shows a trend in the latitude, perhaps following a second order polynomial (upside down U). There are also some indications of a rising trend in the longitude, based on the plot in the upper right corner. An attempt at removing these trends will be done later, using a linear model.

The plots below further ground the initial theories about the trends in the latitude and longitude.

```
with(geoelevs, plot(coords[, 1], data, xlab = "W-E",
              ylab = "Elevation data", pch = 20, cex = 0.7))
lines(with(geoelevs, lowess(data ~ coords[, 1])))
```




```
with(geoelevs, plot(coords[, 2], data, xlab = "S-N",
              ylab = "Elevation data", pch = 20, cex = 0.7))
lines(with(geoelevs, lowess(data ~ coords[, 2])))
```



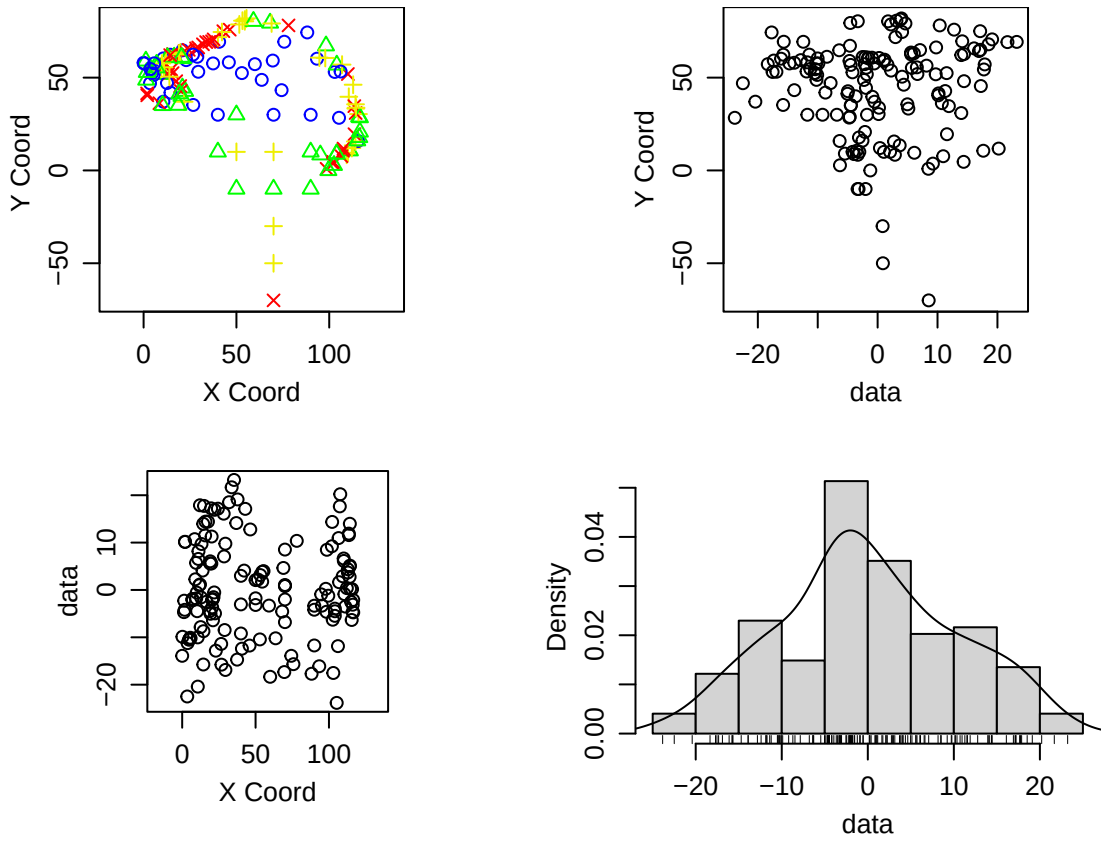
Attempting to remove the trends with a linear model. The residuals of the linear regression are added to a new dataframe, with the original data.

```
lm.fit <- lm(data ~ y + poly(x,2), data = elevs)
summary(lm.fit)
```

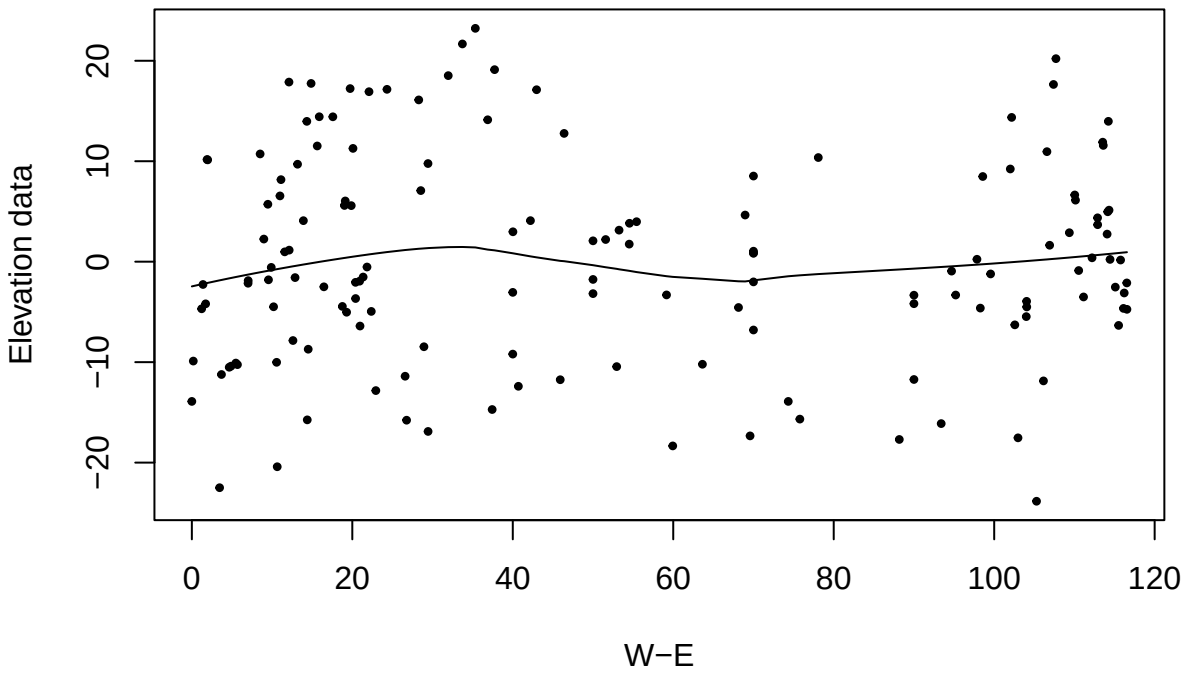
```
#>
#> Call:
#> lm(formula = data ~ y + poly(x, 2), data = elevs)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -23.845  -6.301  -1.079   6.571  23.226
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 10.18780    1.81117   5.625 9.35e-08 ***
#> y            0.13223    0.03658   3.615 0.000415 ***
#> poly(x, 2)1 -7.12030   11.65722  -0.611 0.542291
#> poly(x, 2)2 39.06998   10.41746   3.750 0.000255 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 10.33 on 144 degrees of freedom
#> Multiple R-squared:  0.1754, Adjusted R-squared:  0.1583
#> F-statistic: 10.21 on 3 and 144 DF,  p-value: 3.862e-06
```

```
# lm.fit <- lm(data ~ poly(y,2) + poly(x,2), data = elevs)
# summary(lm.fit)

# Add residuals to a new data frame.
elevs2 <- data.frame(elevs, residuals = lm.fit$residuals)
geoelevs2 <- as.geodata(elevs2, data.col = 4)
```

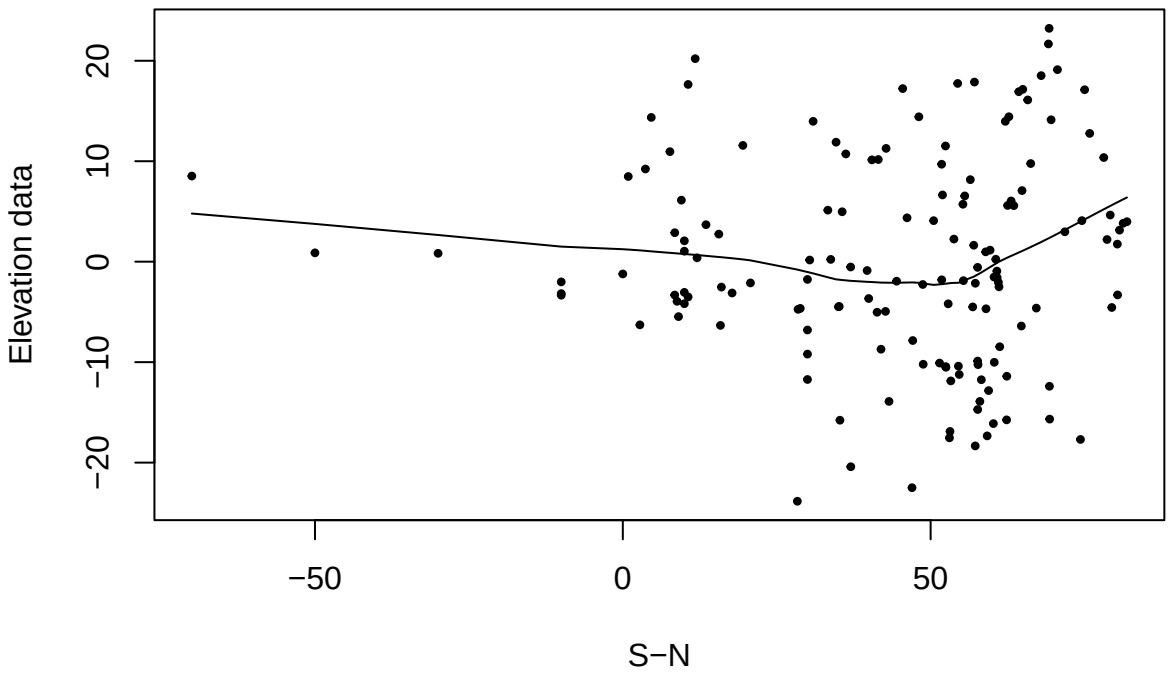
```
plot(geoelevs2)
```




```
with(geoelevs2, plot(coords[, 1], data, xlab = "W-E",
                     ylab = "Elevation data", pch = 20, cex = 0.7))
lines(with(geoelevs2, lowess(data ~ coords[, 1])))
```




```
with(geoelevs2, plot(coords[, 2], data, xlab = "S-N",
                     ylab = "Elevation data", pch = 20, cex = 0.7))
lines(with(geoelevs2, lowess(data ~ coords[, 2])))
```



It looks like some of the trend has been removed, i.e. the linear model has explained some of the variations in the data. Note that we also attempted different combinations of dropping the second order term in latitude and adding a second order term in longitude, but found that this model gives the best results, i.e. the linear model we have used above is the simplest model that explains the data to approximately the same extent as the most complicated model (among these combinations) and the residuals have the least amount of trend. Note that this is by no means a rigorous analysis applying the linear model, but we have chosen the simplest model that qualitatively seems to remove some of the trend in the data. Therefore, we will use the residuals from the linear regression in the rest of the analysis. What do you think about which of the linear models above we should use? The one with poly(x,2) and poly(y,2) has larger adjusted R-squared, but I think the plots look nicer with the one I have chosen here. Thoughts?

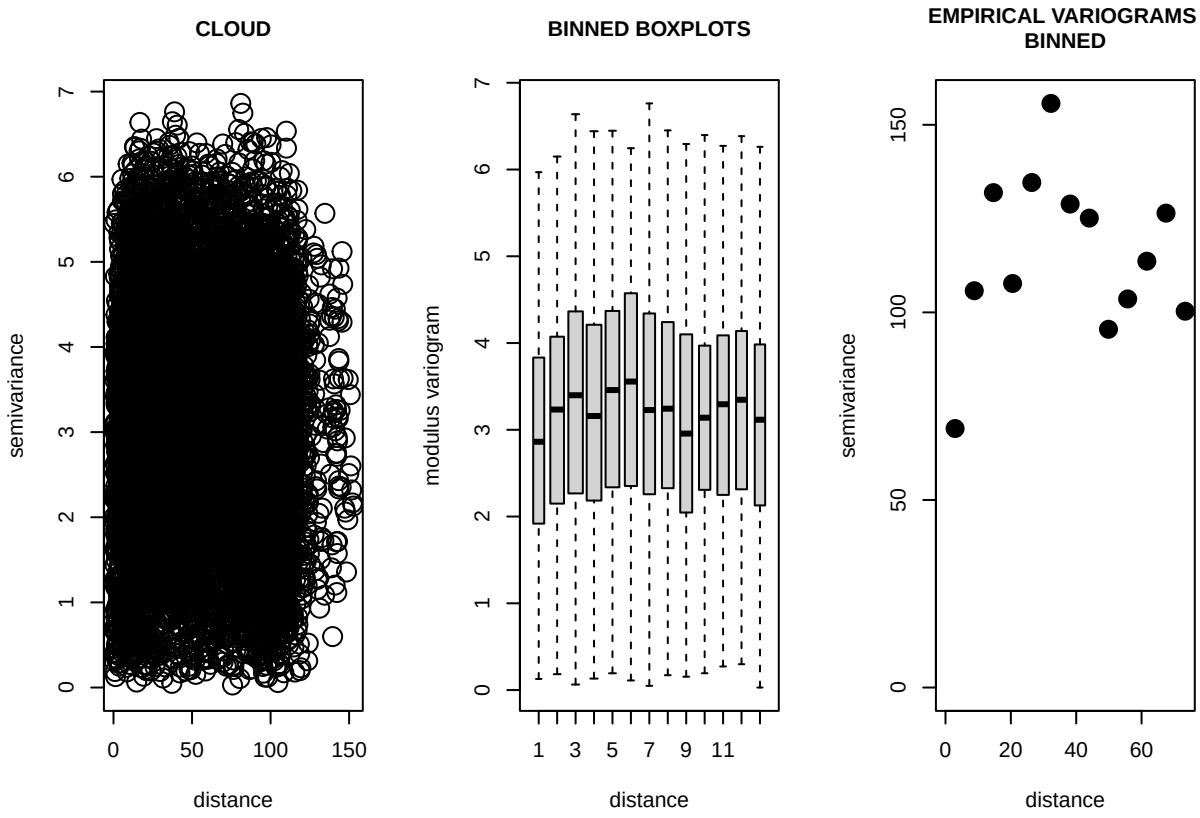
2. Exploration of the small scale variability of the elevation

In this problem we will explore the small scale variability of the elevation data.

First we calculate and plot the variogram cloud together with the empirical variogram. Remember that we now are analyzing the residuals after the linear regression presented earlier. Note that the robust estimator (`modulus`, Hawkins and Cressie 1993) is used when estimating the empirical variogram.

```
# Practical rules: lags only up to half of maxdist. That is why maxdist is saved here.
maxdist <- max(dist(cbind(elevs$x,elevs$y)))
cloud <- variog(geoelevs2, option = "cloud", estimator.type = "modulus")
bp <- variog(geoelevs2, option = "bin", bin.cloud = T,
               pairs.min=30, max.dist=maxdist/2,
               estimator.type = "modulus") # Not sure about all these arguments.
bin <- variog(geoelevs2, option = "bin", pairs.min=30, max.dist=maxdist/2,
                estimator.type = "modulus")
```

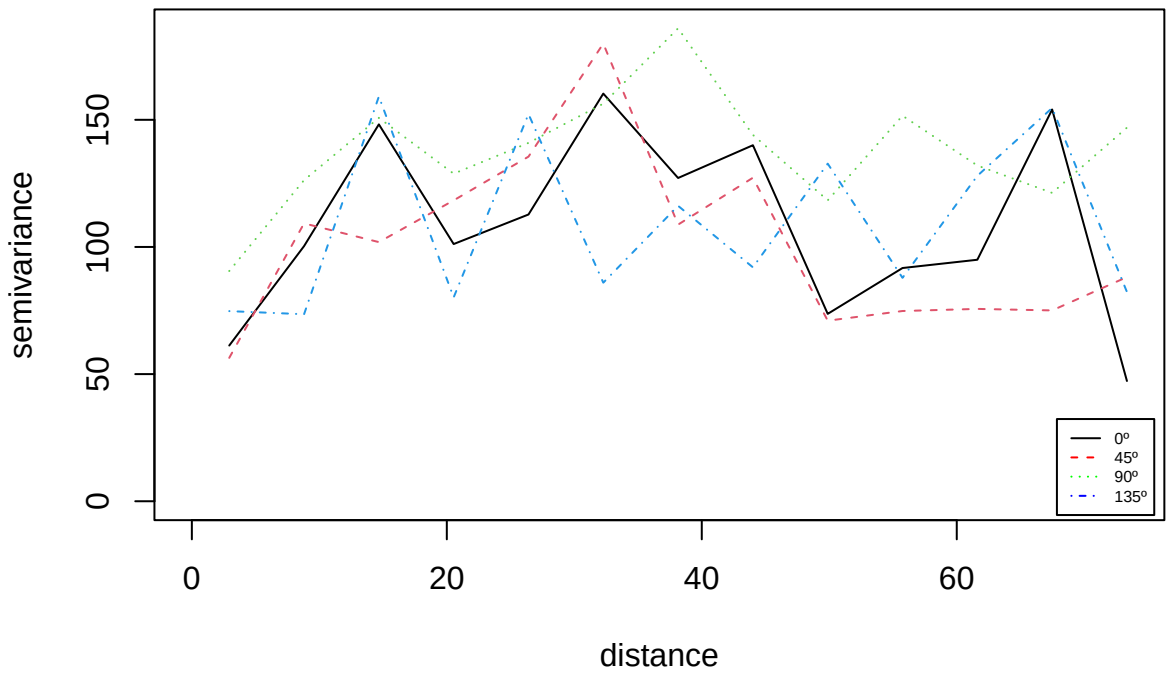
```
# Also: pairs,min = 30 since the sample variogram should only be considered for lags
# that have more than 30 pairs.

par(mfrow=c(1,3))
plot(cloud,main="CLOUD", cex.main=1, cex.lab=1, cex=2)
plot(bp, bin.cloud=T, cex.lab=1)
title(main = "BINNED BOXPLOTS", cex.main=1)
plot(bin, main="EMPIRICAL VARIOGRAMS \nBINNED",cex.main=1, cex.lab=1, cex=2, pch=16)
```



From these variogram plots it looks like the range of a variogram is approx. 30 (distance), the sill around 150 (semivariance) and the nugget might be around 30. Thus, one might say that there is spatial correlation, at least in a range of 30.

Not sure if this should be done in this problem or in problem 3. What do you think? Will calculate directional variograms to study isotropic/anistropic properties also.

```
par(mfrow=c(1, 1))
variod <- variog4(geoelevs2,max.dist=maxdist/2, pairs.min=30,estimator.type = "modulus")
plot(variod,lyt=2,legend=FALSE)
legend(x="bottomright", inset=0.01, lty=c(1,2,3,4), col=c("black", "red", "green","blue"),
       legend=c("0º", "45º", "90º","135º"), cex=0.5)
```

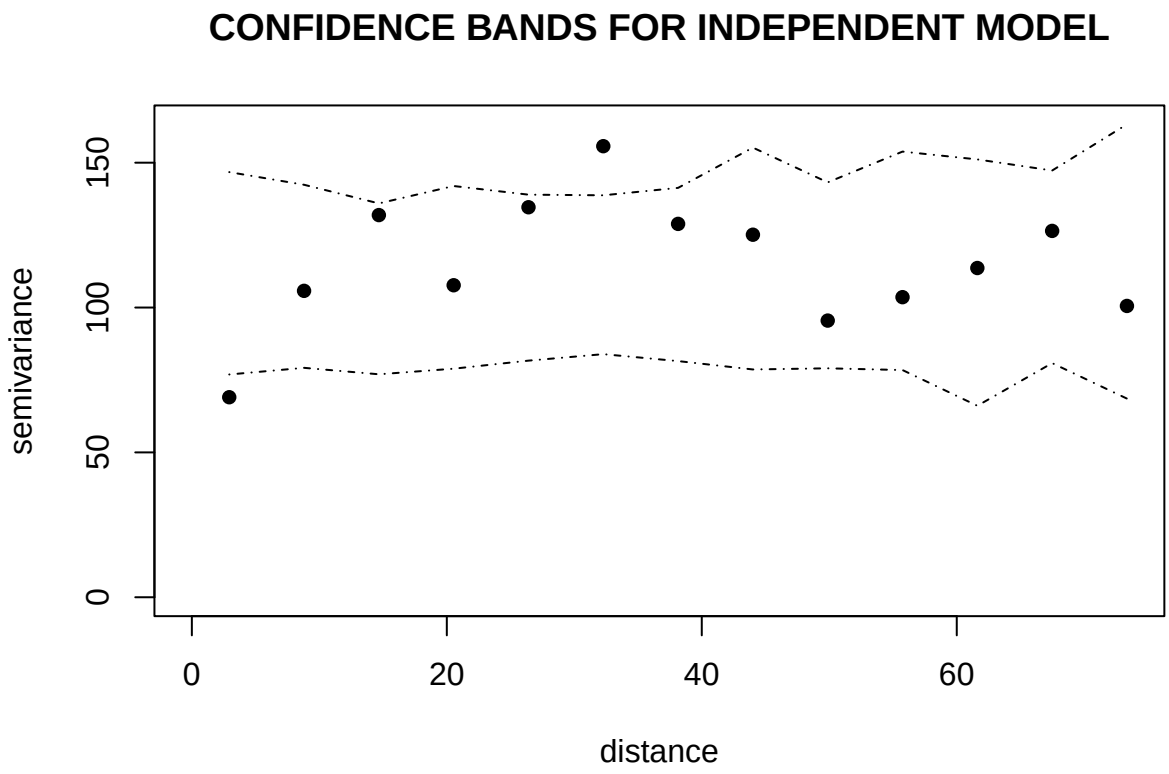


We think the data looks relatively isotropic, as all 4 directions checked look relatively similar to each other.

3. Exploration of the spatial independence

In this problem we will further explore the spatial independence of the process. As specified in the problem description, we will set the seed to 1000.

```
set.seed(1000)
par(mfrow=c(1,1))
indep.env <- variog.mc.env(geoelevs2, coords=geoelevs2$coords, data=geoelevs2$data, obj.variog=bp, nsim=
plot(bp, envelope = indep.env, main="CONFIDENCE BANDS FOR INDEPENDENT MODEL", lwd=2, pch=16)
```



The confidence bands show a range of estimated variograms when the measurements are randomly permuted in the spatial points. The lower confidence band is the minimum value of the variogram for the simulated data in each point and, similarly, the upper confidence band is the maximum value. In order for the hypothesis that the process is independent, i.e. that the apparent increase in the estimated variogram with distance (from earlier) might be attributed to chance, the entire variogram should fall inside these confidence bands. It is not clear what to conclude based on this plot only, but two points are outside the confidence bands, which indicates that complete spatial randomness in the underlying process may be unplausible.

We will also add a Rose diagram, to study the anisotropy further. The Rose diagram will be plotted used the code supplied in the lectures (`RoseDiagram.R`).

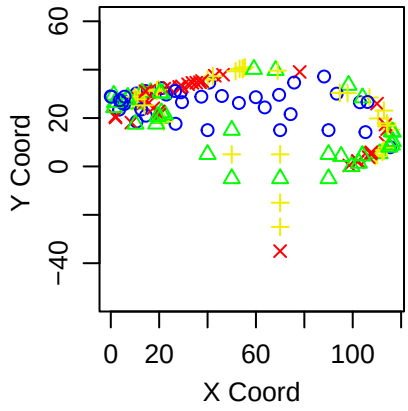
```
source("../Chapter3/RoseDiagram.R") # This has to point at the RoseDiagram-file on your computer.
rose.diagram(data.var=geoelevs2$data,data.cds=geoelevs2$coord,max.dist=maxdist/2,numcases=10,numdirec=4
```

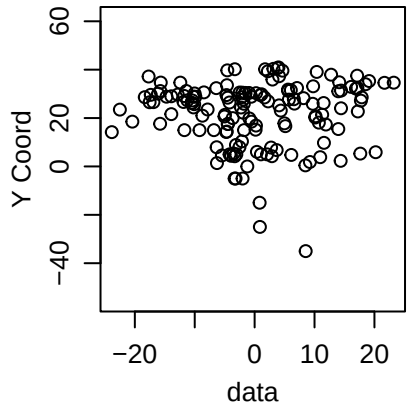
The idea behind this plot is that one can see if the data looks anisotropic. If this is the case, one can try to apply a anisotropic coordinate correction (rotational transformation) and then make the same plot as above again (with the estimated variogram in four different directions), to see if the data looks more isotropic.

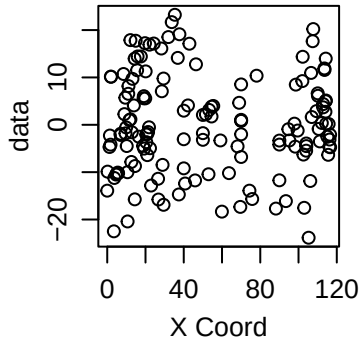
\textcolor{red}{This is done in "exploreSmallScaleDep.R", at the bottom of the file. Done in "study_isotropy.R" also.}

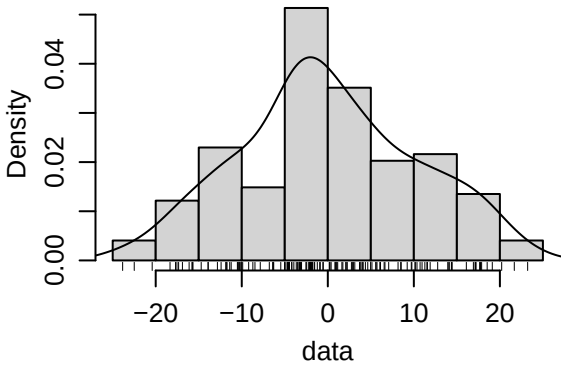
Not sure why I do not get any output now. Does it work on your computer? From the Rose Diagram plotted above, it looks like the Anisotropy angle is 0 (angle between y-axis and the direction with the maximum range). Moreover, it looks like the Anisotropy ratio is 2 (ratio between maximum and minimum ranges). Thus we can try the rotational transformation below and observe if it looks more isotropic Even though it looked pretty good to begin with IMO. What do you think?.

```
angle <- 0 # No rotation in this case, only "squishing".
ratio <- 2
elevs2.ani <- cbind(coords.aniso(geoelevs2$coords, c(angle,ratio),reverse = FALSE),geoelevs2$data)
geoelevs2.ani <- as.geodata(elevs2.ani)
plot(geoelevs2.ani)
```

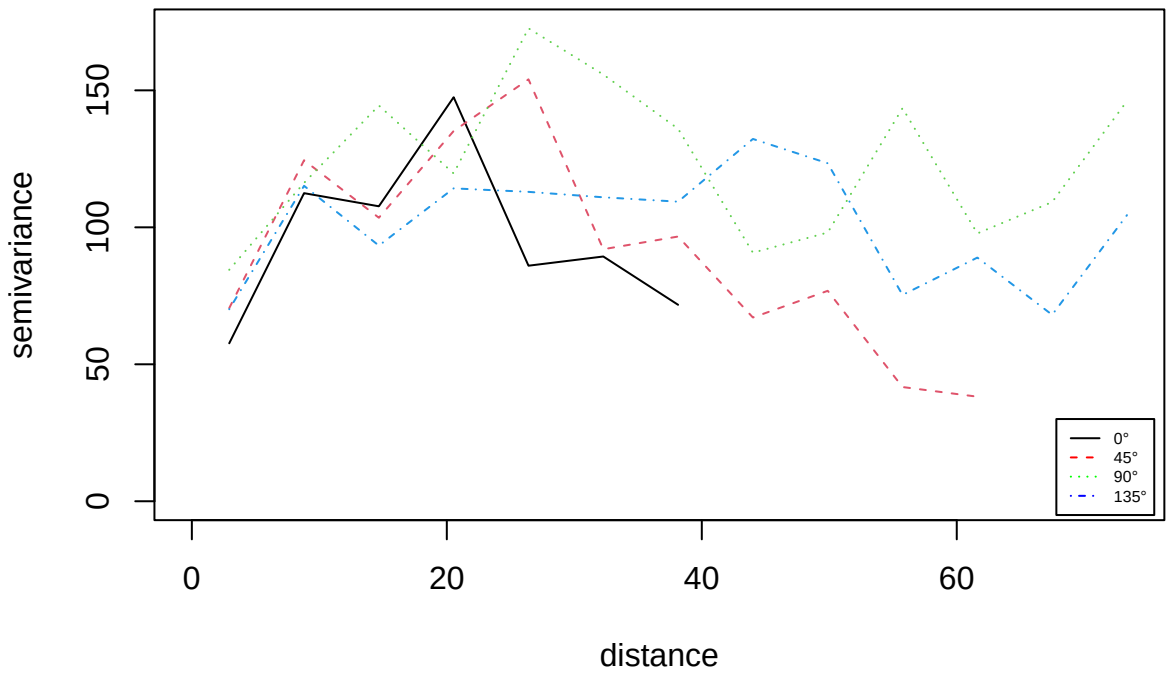








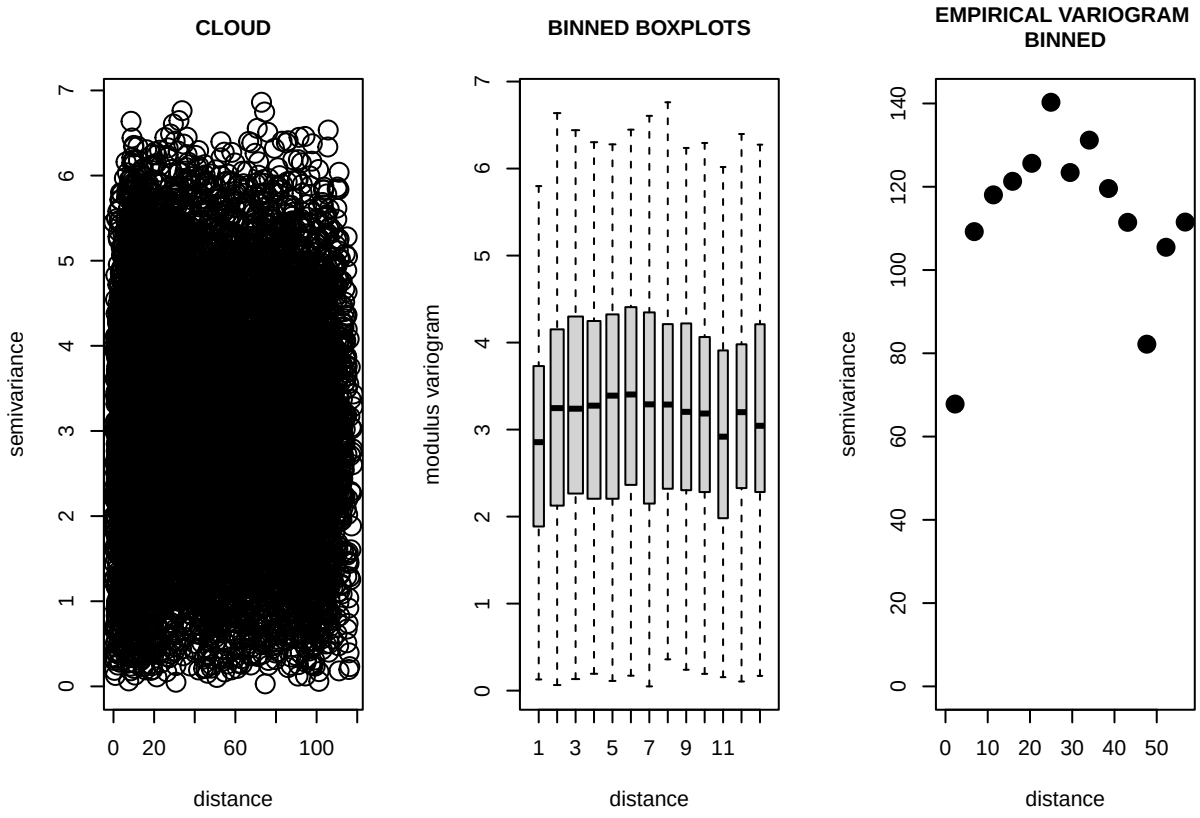

```
par(mfrow=c(1,1))
variod.ani <- variog4(geoelevs2.ani, max.dist=maxdist/2, pairs.min=30,estimator.type = "modulus");
plot(variod.ani,lyt=2,legend=FALSE,main="Anisotropy angle=0,ratio=2")
legend(x="bottomright", inset=0.01, lty=c(1,2,3,4), col=c("black", "red", "green","blue"),
       legend=c("0\u00B0", "45\u00B0", "90\u00B0","135\u00B0"), cex=0.5)
```



These variograms look about the same as earlier IMO. This tranformation-process only makes sense for geometric anisotropies?

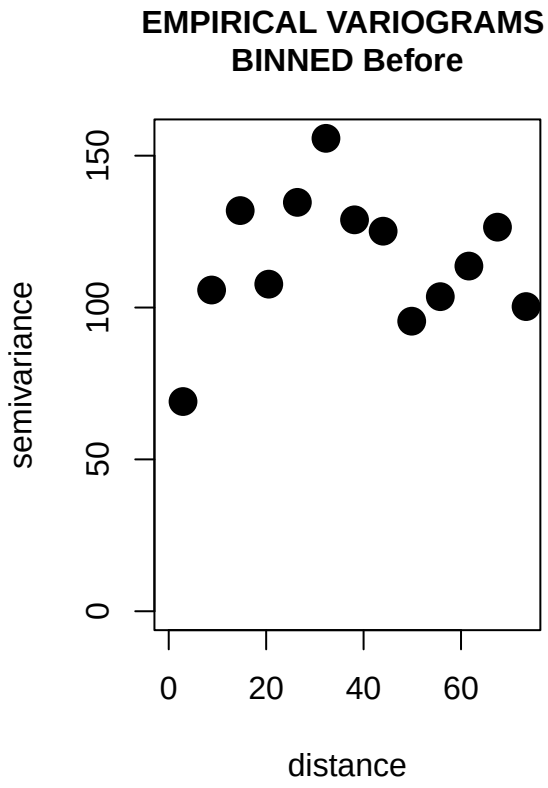
Making the same plots as earlier, to see if the process looks more isotropic.

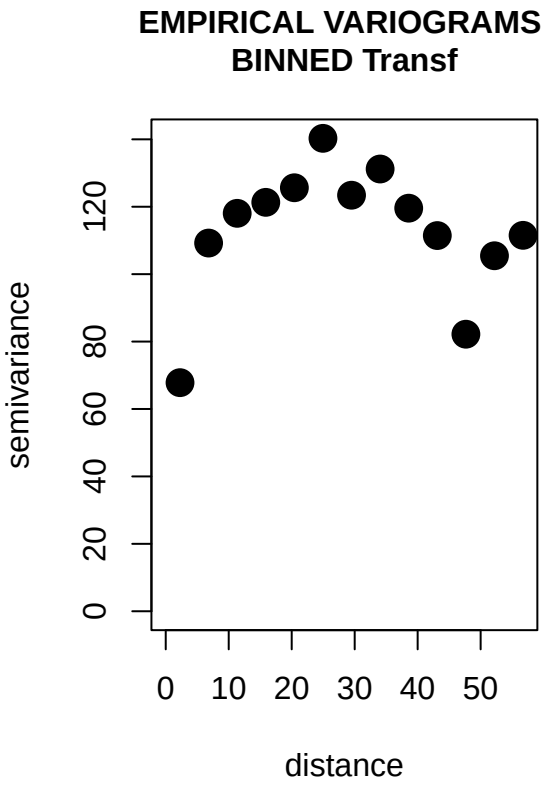
```
maxdist2 <- max(dist(cbind(elevs2.ani[,1],elevs2.ani[,2]))) # Practical rules: lags only up to half of
cloud2 <- variog(geoelevs2.ani, option = "cloud", estimator.type = "modulus")
bp2 <- variog(geoelevs2.ani, option = "bin", bin.cloud = T,
              pairs.min=30, max.dist=maxdist2/2,
              estimator.type = "modulus") # Not sure about all these arguments.
bin2 <- variog(geoelevs2.ani, option = "bin", pairs.min=30, max.dist=maxdist2/2,
               estimator.type = "modulus")
# Also: pairs.min = 30 since the sample variogram should only be considered for lags that have more tha

par(mfrow=c(1,3))
plot(cloud2,main="CLOUD", cex.main=1, cex.lab=1, cex=2)
plot(bp2, bin.cloud=T, cex.lab=1)
title(main = "BINNED BOXPLOTS", cex.main=1)
plot(bin2, main="EMPIRICAL VARIOGRAM \nBINNED",cex.main=1, cex.lab=1, cex=2, pch=16)
```



Below we compare the sample variograms before and after the transformation.

```
par(mfrow=c(1,2))
plot(bin, main="EMPIRICAL VARIOGRAMS \nBINNED Before",cex.main=1, cex.lab=1, cex=2, pch=16)
plot(bin2, main="EMPIRICAL VARIOGRAMS \nBINNED Transf",cex.main=1, cex.lab=1, cex=2, pch=16)
```





It looks like the sills are different. Do we have to conclude on whether this is a geometric, zonal or combined anisotropy?

4. Theoretical variograms and estimations of their parameters

In this problem we will propose four theoretical variograms and estimate them via restricted maximum likelihood (REML) Only applicable to processes with second-order stationary errors?. Later we will select the two variograms that best fit the data and explain the parameters of the chosen variogram. We propose the exponential, Gaussian, spherical and Matérn models. Note that this is done with 'bin' for now, which is the empirical variogram estimate with the un-transformed data (referring to the transformation done above). This can easily be changed by swapping 'bin' for 'bin2' or 'geoelevs2' for 'geoelevs2.ani'. Which data set do you think should be used here? In order to find the initial values for optimization of the restricted maximum likelihoods, we used the function below, which is a graphical tool.

```
eyefit(bin, silent = FALSE)
```

From this tool, we arrived at the initial values inserted into the functions below. Please check that you also agree with these values :)

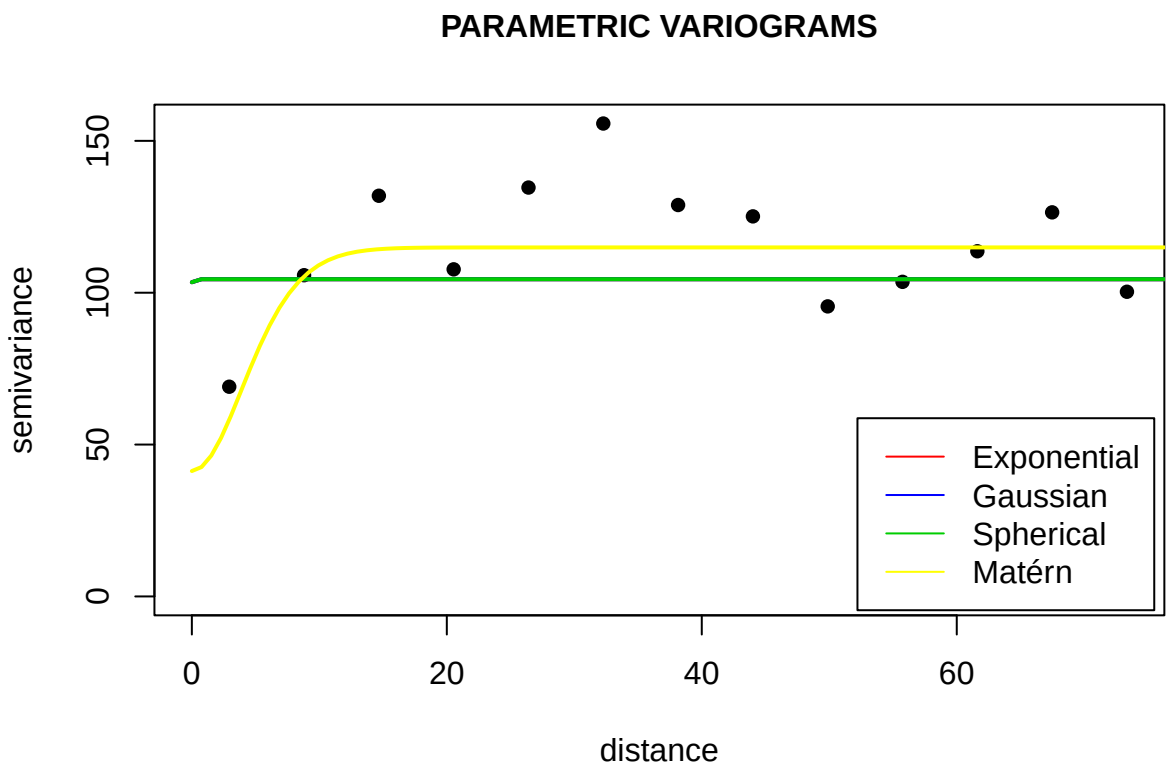
In the functions `likfit` below, the first initial value is for the partial sill and the second initial value is for the range. We have used the default trend of `cte` in `likfit`. What do you think about this?

```
#Fit an exponential model with nugget effect.
lk1 <- likfit(geoelevs2, cov.model = "exponential", ini =c(3,0.19),
              fix.nugget = F, nugget =0.18 ,lik.method = "REML")

# Fit a Gaussian model.
lk2 <- likfit(geoelevs2, cov.model = "gaussian", ini = c(3,0.19),
```

```
              fix.nugget = F, nugget =0.09 ,lik.method = "REML")

# Fit a spherical model with nugget effect.
lk3 <- likfit(geoelevs2, cov.model = "spherical", ini = c(1.29,0.41),
              fix.nugget = F, nugget =0.41,lik.method = "REML")

# Fit a Matérn model.
lk4 <- likfit(geoelevs2, cov.model = "matern", ini = c(2.03,0.41),
              fix.nugget = F, nugget =0.19, fix.kappa = FALSE, kappa=0.95,lik.method = "REML")
# Kappa is the smoothing parameter for the Matérn model.

# Increase number of bins to estimate the variograms (as done in 'variogramaEstimationScallops3.R')?
# This estimates a new sample variogram I think. Should this be done earlier, in problem 2/3 already pe
```

The parametric variograms are plotted together with the empirical variogram below. LOOKS LIKE SHIT WITH THE MAXIMUM LIKELIHOOD.

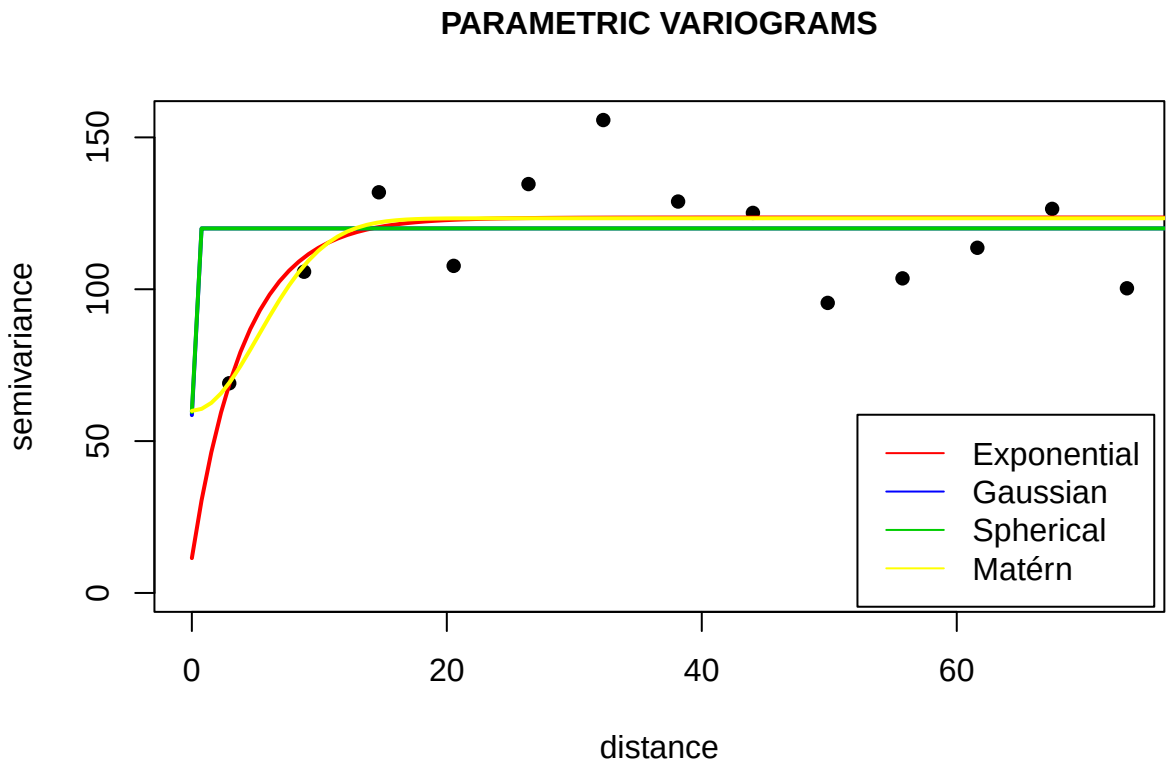
```
par(mfrow=c(1, 1))
plot(bin,  main = "PARAMETRIC VARIOGRAMS",cex.main = 1, pch = 16)
lines(lk1, lwd = 2, col = "red", max.dist = maxdist/2)
lines(lk2, lwd = 2, col = "blue", max.dist = maxdist/2)
lines(lk3, lwd = 2, col = "green3", max.dist = maxdist/2)
lines(lk4, lwd = 2, col = "yellow", max.dist = maxdist/2)
legend(x = "bottomright", inset = 0.01, lty = c(1, 1), col = c("red", "blue", "green3","yellow"),
       legend = c("Exponential", "Gaussian", "Spherical","Matérn"), cex = 1)
```



I AM TRYING WITH LEAST SQUARES AS WELL: comment on what Cressie is.

```r
# Exponential variogram.
wls1 <- variofit(bin, cov.model = "exponential", ini = c(3,0.19),
                 fix.nugget = F, nugget =0.18 , weights="cressie")

# Gaussian variogram.
wls2 <- variofit(bin, cov.model = "gaussian", ini = c(3,0.19),
                 fix.nugget = F, nugget =0.09, weights="cressie")

# Spherical.
wls3 <- variofit(bin, cov.model = "spherical", ini = c(1.29,0.41),
                 fix.nugget = F, nugget =0.4, weights="cressie")

# Matérn variogram.
wls4 <- variofit(bin, cov.model = "matern", ini = c(2,0.4),
                 fix.nugget = F, nugget =0.2, fix.kappa = FALSE, kappa=0.95,weights="cressie")
```

```r
par(mfrow=c(1, 1))
plot(bin,  main = "PARAMETRIC VARIOGRAMS",cex.main = 1, pch = 16)
lines(wls1, lwd = 2, col = "red", max.dist = maxdist/2)
lines(wls2, lwd = 2, col = "blue", max.dist = maxdist/2)
lines(wls3, lwd = 2, col = "green3", max.dist = maxdist/2)
lines(wls4, lwd = 2, col = "yellow", max.dist = maxdist/2)
legend(x = "bottomright", inset = 0.01, lty = c(1, 1), col = c("red", "blue", "green3","yellow"),
       legend = c("Exponential", "Gaussian", "Spherical","Matérn"), cex = 1)
```



THE LEAST SQUARED LOOKS WAY BETTER, HENCE WILL CONTINUE WORKING WITH THOSE

FOR NOW. MAYBE NEED TO USE BIN2 ?

Selecting the variograms that best fit the data can be done by a combination of inspecting the semivariance plot (qualitative), comparing the sum of squares from the minimizations and comparing the AIC AIC only seems to exist for maximum likelihood. From inspection it looks like the Exponential and the Matérn are the two best models. The quantiative measures are summarized in the table below.

```
comparison <- data.frame("Model" = c("Exponential", "Gaussian", "Spherical","Matérn"),
                "Sum of Squares" = c(summary(wls1)$sum.of.squares, summary(wls2)$sum.of.squares,
                    summary(wls3)$sum.of.squares, summary(wls4)$sum.of.squares),
                        "AIC" = c(0,0,0,0))
knitr::kable(comparison)
```

Model	Sum.of.Squares	AIC
Exponential	114.3839	0
Gaussian	200.7675	0
Spherical	200.7675	0
Matérn	112.0823	0

From the table we can see that the visual insights are further substantiated by the sum of squares. Hence, we choose the Exponential and Matérn models. Now, let us explain the parameters from the models. First of all, the parameters are the following

```
# looks like the maximum likelihood also gives a lot more parameters ?
# Maybe we should use the max likelihood if we can get it to work properly!
wls1$nugget
```

```
#> [1] 11.46801
```

```
wls4$nugget
```

```
#> [1] 59.94185
```

In general the parameters of the variogram represent the following quantities:

- Nugget: The nugget effect presents micro-scale variation or measurement error. This can be seen graphically as the y-value where the variogram crosses the y-axis.
- Sill: The sill represents the variance of the random field. Note that a quantity called the *partial sill* is defined as $\sigma^2 = \text{sill} - \text{nugget}$.
- Range: The range represents the distance at which the data no longer are autocorrelated. This can be seen graphically by noting the distance, i.e. the x-value, where the variogram stops increasing or becomes approximately parallell to the x-axis.

Now, how about the parameters of the chosen variograms?

5. Predictions of elevations along the area of study

In this final problem we will use kriging to predict elevations along the area of study. This will be done using the two best variaograms among the four proposed theoretical variograms in problem 4.

```
# Make grid for kriging and do it.
```

General questions to profe:

- Should we somehow comment on if the random process is Stationary, Weak Stationary or Intrinsic Stationary?