1. *Let γ(si,sj)= γ(hij) be a semivariogram for a second-order stationary spatial process.*

   (a) *Show that*

   (b) *Show that*

2. *Matherons semivariogram estimator is*

   where N(h)= (si,sj):h=si-sj and —N(h)— is the number of pairs in the set N(h).

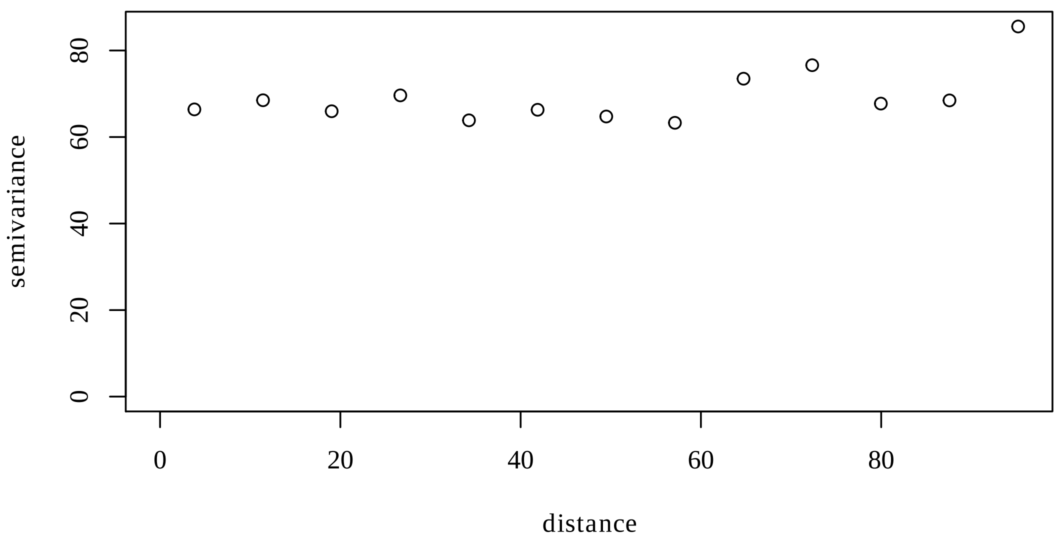
(a) *Let Z(s) = μ + e(s) where E [e(s)] = 0 with Z (h) = e(h) (adding a constant to the random error terms does not change the variance/covariance properties of the process). Show that $\hat{\gamma}$(h) is unbiased for γ_z(h). That is, show E [$\hat{\gamma}$(h)] = γ_z(h). Hint: Show that under an assumption of a constant mean*

(b) *We pointed out in class that Matherons Estimator is biased in the presence of trend. Let Z(s) = μ(s) + e(s) with E [e(s)] = 0 and γ_z(h) = γ_e(h). Show*

(c) *Consider the (very) simple model Zi = 10 + ei where the eis are independent normally distributed error terms with variance σ^2 = 81. We have a pure nugget effect model $\gamma_z(h) = \gamma_e(h)$ = 81. Simulate 100 observations of Zi and calculate the empirical semi- variogram assuming the observations are on a one-dimensional transect.*

Compare what you see in the plot to the true $\gamma_z(h)$. Is the result consistent with part (a) above? Why or why not?

```
set.seed(123)
library(geoR)
    Zdat<-10+rnorm(100,0,9)
    i<-1:100
    xycoord<-cbind(c(rep(1,100)),i)
    Zvgram<-variog(coords=xycoord,data=Zdat)

variog: computing omnidirectional variogram

    plot(Zvgram)
```




```
    mean.Zvgram <- mean(Zvgram$v)
```

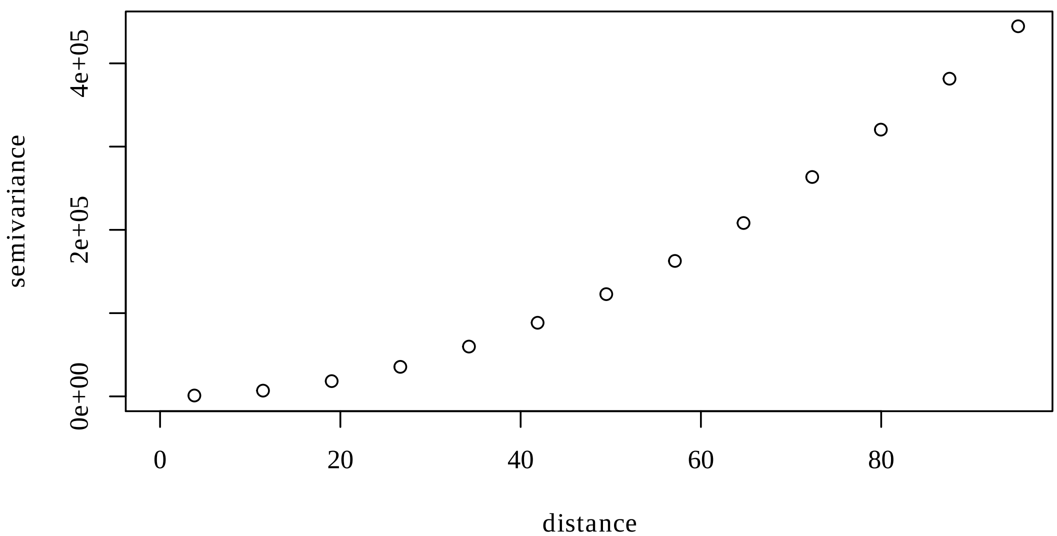
The true γ_z(h) = γ_e(h) = 81. Visually, it appears Materon's estimator in this realization is not on target which is verified by finding the mean of the binned semi-variances, which is 69.2679. On average over many realizations Materon's estimator should be unbiased, but again we only have one realization here. The semi-variances appear to slightly increase towards greater distances

(d) *Redo the above calculations based on the (still) simple model Zi = 10 + 10i + ei, i.e. there is now a linear trend and the process is no longer stationary.*

Compare the empirical semivariogram Zvgram2 to Zvgram. Are the results consistent with part (b) above? Justify your answer.

The empirical semivariogram Zvgram2 is on a much larger scale than Zvgram and increases with distance. This is consistent with the result in (b), which shows in the presence of a linear spatial trend, the empirical semivariogram is the semivariogram in the presence of no spatial linear trend, plus an extra term that is a function of the difference in means, which in this case is a function of the distance.

```
set.seed(123)
Zdat2<-10 +10*i + rnorm(100,0,9)
Zvgram2<-variog(coords=xycoord,data=Zdat2)

variog: computing omnidirectional variogram

plot(Zvgram2)
```



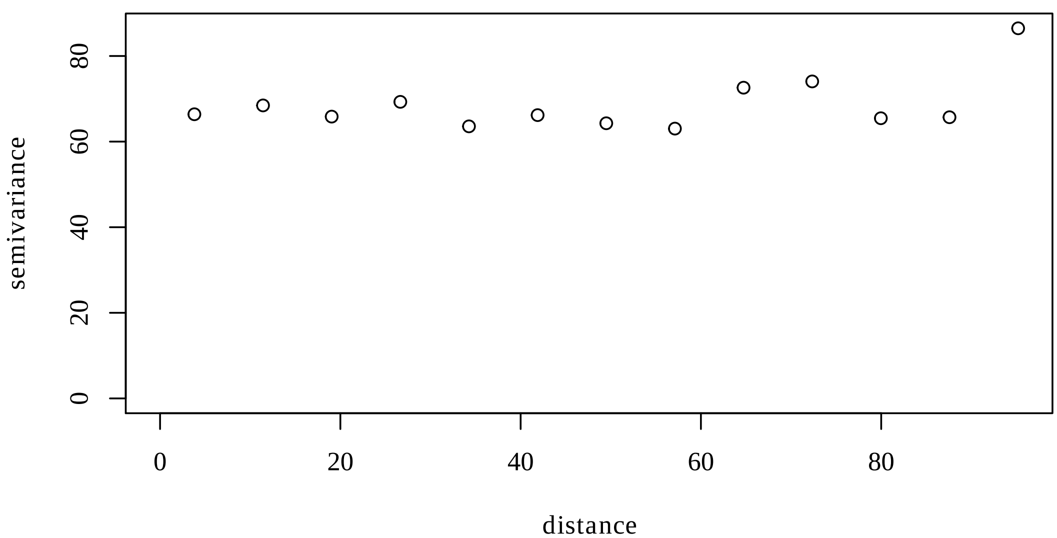
(e) *Fit a linear model to the data in (d), extract the residuals, and compute the empirical semivariogram for the residuals. Note that what you are doing is removing the trend.*

Compare the 3 empirical semivariograms.

Once the spatial linear trend is removed, the empirical semi-variogram of the residuals appears to be very similar to the semi-variogram in (c), both in terms of bias and in terms of variation in semi-variances.

Together, relative to the empirical semi-variogram in (c), the semi-variogram of the residuals is very similar. Relative to the empirical semi-variogram in (d), the semi-variogram of the residuals is unbiased with a smaller variance.

```
e.resid <- residuals(lm(Zdat2~i))

evgram <- variog(coords=xycoord,data=e.resid)

variog: computing omnidirectional variogram

plot(evgram)
```



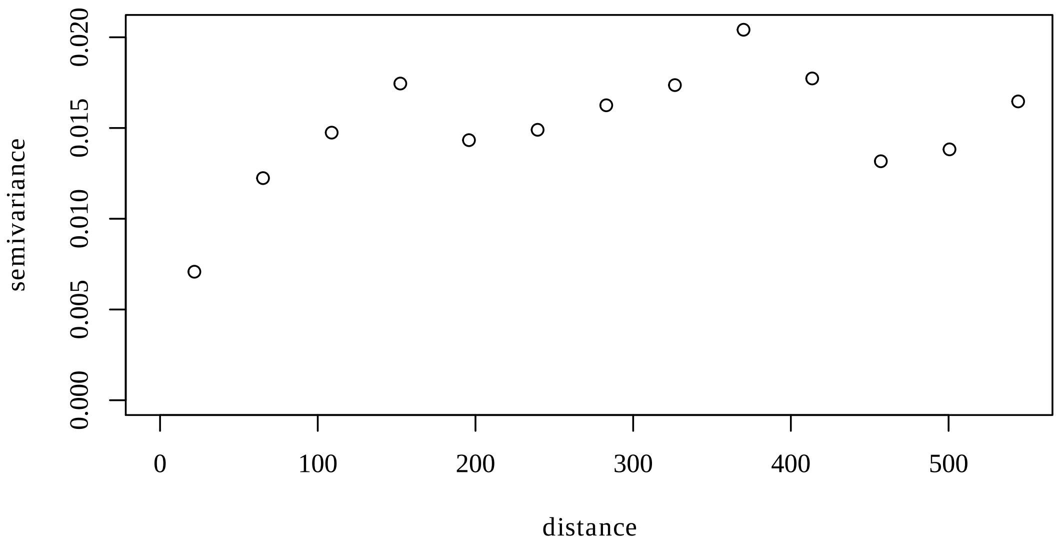
3. *Attached is a data set containing the carbon nitrogen values used in the carbon/nitrogen data set. The first 2 columns contain the coordinates, total nitrogen is in the third column, total carbon is in the 4th column and the ratio is in the last column. We will work with the total carbon data. Use geoR for the analysis. It will be easiest if you convert the data into a geodata object as follows.*

   (a) *Calculate the empirical semivariogram. Give initial eyeball estimates of the nugget effect, sill, and (effective) range.*

   The effective range is close to 150, the nugget effect is close to 0.005, and the sill is close to 0.016. The semi-variance does not stabilize for distances greater than a specific point, however, the semi-variance pattern first changes at a distance of 150.

```
require(geoR)
CN.dat <- read.table("CN.dat", header = TRUE)

# this code created a very different looking semi-variogram
#data.frame(read.table("CN.dat", header = TRUE))
#CN.dat[,1] <- as.numeric(CN.dat[,1])
#CN.dat[,2] <- as.numeric(CN.dat[,2])
#CN.dat[,2] <- as.numeric(CN.dat[,4])
#TC.geodata<-as.geodata(CN.dat)#,coords.col=c(1,2),data.col=4)
#plot(TC.var <- variog(TC.geodata))
```

```
TC.geodata<-as.geodata(CN.dat,coords.col=c(1,2),data.col=4)

plot(TC.var <- variog(coords = TC.geodata$coords, data = as.numeric(TC.geodata$data)))

variog: computing omnidirectional variogram
```

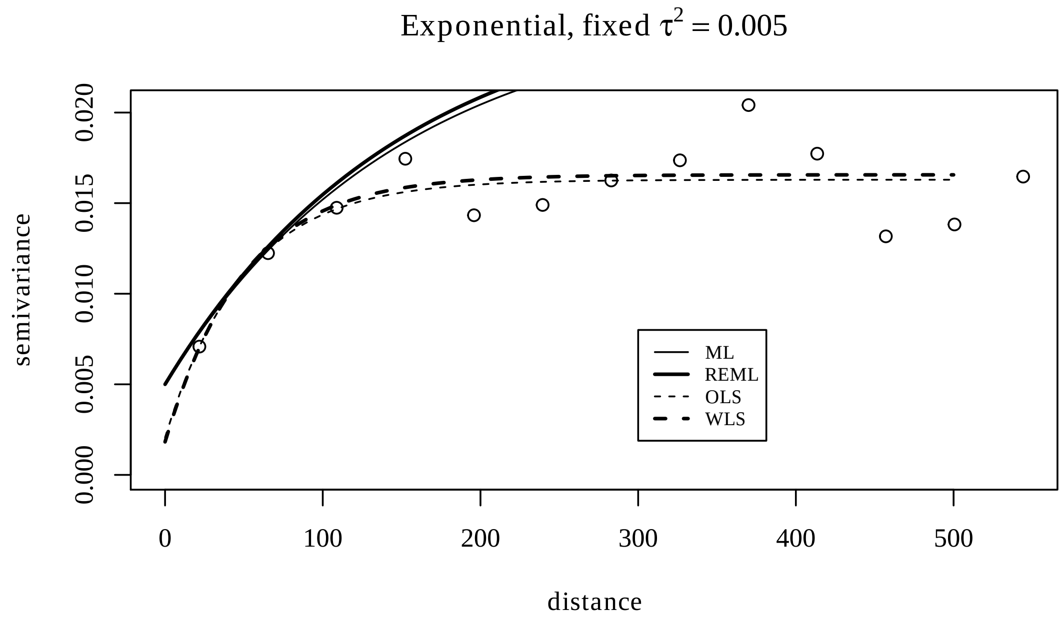


(b) *Fit an exponential semivariogram to the carbon data using OLS, WLS, MLE, and REML methods. Specify a nugget effect in each case, i.e. you do not need to consider models without a nugget. Plot the fitted functions and comment on which one you like best.*

ML and REML both overestimate the sill, OLS and WLS are on target to what I specified as the sill. OLS is simpler, and so is preferred.

```
# Fitting models estimated nugget

# partial sill = 0.016-0.005=0.011
ml.n <- likfit(TC.geodata, ini = c(0.011,150), nugget = 0.005, lik.method = "ML", cov.model = "exponential",
               fix.nug = TRUE)

kappa not used for the exponential correlation function
---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
         arguments for the maximisation function.
        For further details see documentation for optim.
likfit: It is highly advisable to run this function several
        times with different initial values for the parameters.
likfit: WARNING: This step can be time demanding!
---------------------------------------------------------------
likfit: end of numerical maximisation.

reml.n <- likfit(TC.geodata, ini = c(0.011,150), nugget = 0.005, lik.method = "REML",
```

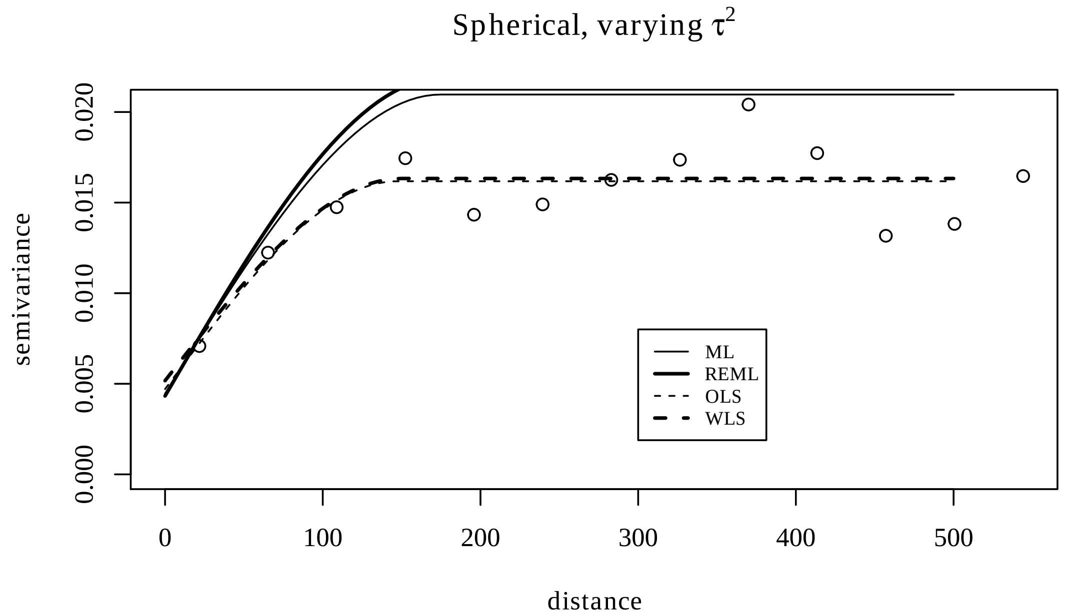
```
                  cov.model = "exponential",
               fix.nug = TRUE)

kappa not used for the exponential correlation function
---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
         arguments for the maximisation function.
        For further details see documentation for optim.
likfit: It is highly advisable to run this function several
        times with different initial values for the parameters.
likfit: WARNING: This step can be time demanding!
---------------------------------------------------------------
likfit: end of numerical maximisation.

# i chose cressie's weights
wls.n <- variofit(TC.var, ini = c(0.011,150/3), weights = "cressie",
                  nugget=0.005, cov.model = "exponential")

variofit: covariance model used is exponential
variofit: weights used: cressie
variofit: minimisation function used: optim

# equal weights are ols
ols.n <- variofit(TC.var, ini = c(0.011,150/3), weights = "equal",
                  nugget=0.005, cov.model = "exponential")

variofit: covariance model used is exponential
variofit: weights used: equal
variofit: minimisation function used: optim

# Now, plotting fitted models against empirical variogram
par(mfrow = c(1,1))
plot(TC.var, main = expression(paste("Exponential, fixed ", tau^2 == 0.005)))
lines.variomodel(ml.n, max.dist = 500)
lines(reml.n, lwd = 2, max.dist = 500)
lines(ols.n, lty = 2, max.dist = 500)
lines(wls.n, lty = 2, lwd = 2, max.dist = 500)
legend(300, 0.008, legend=c("ML","REML","OLS","WLS"),lty=c(1,1,2,2),lwd=c(1,2,1,2), cex=0.7)
```



(c) With the spherical covariance structure WLS and OLS appear again to do the best at matching the sill I specified and so OLS is againpreferred.

```
#range itself spherical, range/number exponential

ml.s <- likfit(TC.geodata, ini = c(0.011,150), nugget = 0.005, lik.method = "ML", cov.model = "spherical",
               fix.nug = FALSE)

kappa not used for the spherical correlation function
---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
         arguments for the maximisation function.
        For further details see documentation for optim.
likfit: It is highly advisable to run this function several
        times with different initial values for the parameters.
likfit: WARNING: This step can be time demanding!
---------------------------------------------------------------
likfit: end of numerical maximisation.

reml.s <- likfit(TC.geodata, ini = c(0.011,150), nugget = 0.005, lik.method = "REML",
                  cov.model = "spherical",
               fix.nug = FALSE)

kappa not used for the spherical correlation function
---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
         arguments for the maximisation function.
        For further details see documentation for optim.
likfit: It is highly advisable to run this function several
        times with different initial values for the parameters.
likfit: WARNING: This step can be time demanding!
---------------------------------------------------------------
```

```
likfit: end of numerical maximisation.

# i chose cressie's weights
wls.s <- variofit(TC.var, ini = c(0.011,150), weights = "cressie",
                  nugget=0.005, cov.model = "spherical")

variofit: covariance model used is spherical
variofit: weights used: cressie
variofit: minimisation function used: optim

# equal weights are ols
ols.s <- variofit(TC.var, ini = c(0.011,150), weights = "equal",
                  nugget=0.005, cov.model = "spherical")

variofit: covariance model used is spherical
variofit: weights used: equal
variofit: minimisation function used: optim

# Now, plotting fitted models against empirical variogram
par(mfrow = c(1,1))
plot(TC.var, main = expression(paste("Spherical, varying ", tau^2)))
lines.variomodel(ml.s, max.dist = 500)
lines(reml.s, lwd = 2, max.dist = 500)
lines(ols.s, lty = 2, max.dist = 500)
lines(wls.s, lty = 2, lwd = 2, max.dist = 500)
legend(300, 0.008, legend=c("ML","REML","OLS","WLS"),lty=c(1,1,2,2),lwd=c(1,2,1,2), cex=0.7)
```



(d) *Summarize your results in table format. Compare the results and discuss.*

The practical range and partial sill are overestimated for the ML and REML methods with the spherical covariance structure. OLS and WLS approximate the practical range and partial sill very closely with the spherical covariance model. All methods do well in estimating the nugget effect based on what I specified.

```
var.outn <- data.frame(cbind(c(ml.n$parameters.summary[c(2,3,4),2]),
                             c(reml.n$parameters.summary[c(2,3,4),2]),
                             c(t(summary(wls.s)$estimated.pars)),
                             c(t(summary(wls.s)$estimated.pars))))
colnames(var.outn) <- c("ML Exp", "REML Exp", "OLS Exp", "WLS Exp")
rownames(var.outn) <- c("nugget", "part. sill", "pract. range")


var.out <- data.frame(cbind(c(ml.s$parameters.summary[c(2,3,4),2]),
                            c(reml.s$parameters.summary[c(2,3,4),2]),
                            c(t(summary(ols.s)$estimated.pars)),
                            c(t(summary(ols.s)$estimated.pars))))
colnames(var.out) <- c("ML Sphere", "REML Sphere", "OLS Sphere", "WLS Sphere")
rownames(var.out) <- c("nugget","part. sill", "pract. range")

print(xtable(var.outn, align = "||l|l|l|l|l||", digits = c(6,6,6,6,6)))
```

	ML Exp	REML Exp	OLS Exp	WLS Exp
nugget	0.005000	0.005000	0.005171	0.005171
part. sill	0.021000	0.021500	0.011160	0.011160
pract. range	150.000000	150.000000	150.000000	150.000000

```
print(xtable(var.out, align = "||l|l|l|l|l||", digits = c(6,6,6,6,6)))
```

	ML Sphere	REML Sphere	OLS Sphere	WLS Sphere
nugget	0.004500	0.004300	0.004705	0.004705
part. sill	0.016500	0.017600	0.011475	0.011475
pract. range	174.823700	176.065700	147.305888	147.305888