ECOL 8890: Assignment 5

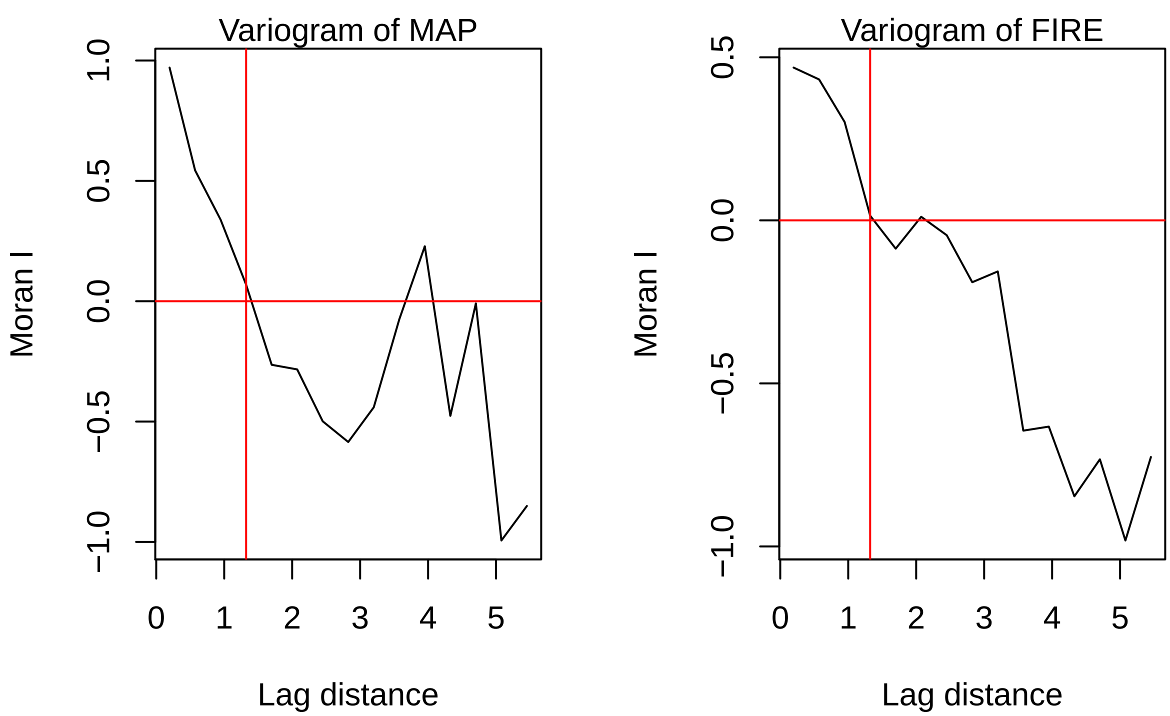
Michelle Evans

```
treeData <- read.csv("SNP_TC_change1.csv", stringsAsFactors = F)
#rescale location data
treeData$XScaled <- scale(treeData$X, center = T, scale = T)
treeData$YScaled <- scale(treeData$Y, center = T, scale = T)
```

1

Produce variograms for MAP and FIRE. Test for significant spatial autocorrelation using Moran's I. What is the range of the spatial autocorrelation of the model residuals?

```
par(mfrow = c(1,2))
MAPvar <- with(treeData, data.frame(correlog(coords = cbind(XScaled, YScaled), z = MAP, method = "Moran
plot(MAPvar$dist.class, MAPvar$coef, type = "l", xlab = "Lag distance", ylab = "Moran I")
mtext("Variogram of MAP")
abline(h=0, col="red")
abline(v=MAPvar$dist.class[which(MAPvar$coef<0.1)[1]], col="red")

FIREvar <- with(treeData, data.frame(correlog(coords = cbind(XScaled, YScaled), z = FIRE, method = "Mora
plot(FIREvar$dist.class, FIREvar$coef, type = "l", xlab = "Lag distance", ylab = "Moran I")
mtext("Variogram of FIRE")
abline(h=0, col="red")
abline(v=FIREvar$dist.class[which(FIREvar$coef<0.1)[1]], col="red")
```

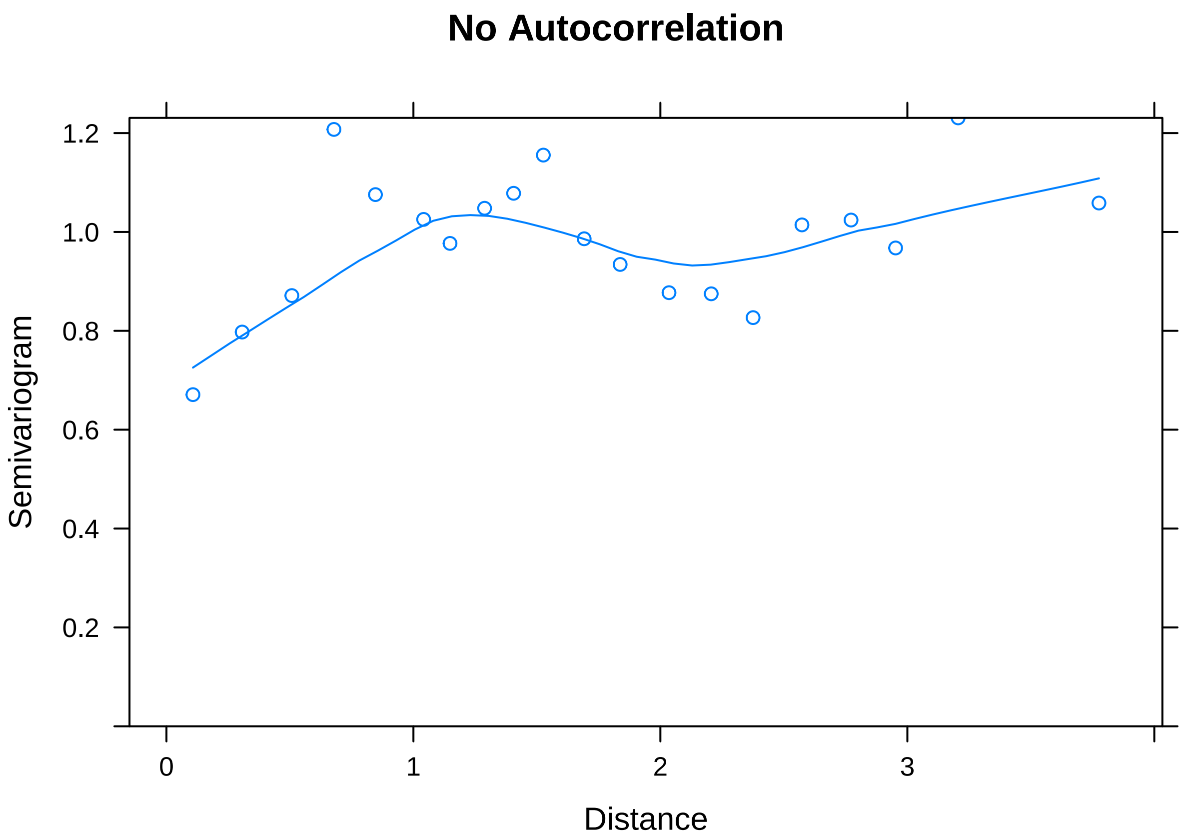


Both of the variables are autocorrelated until about 1.3 rescaled distance units away.

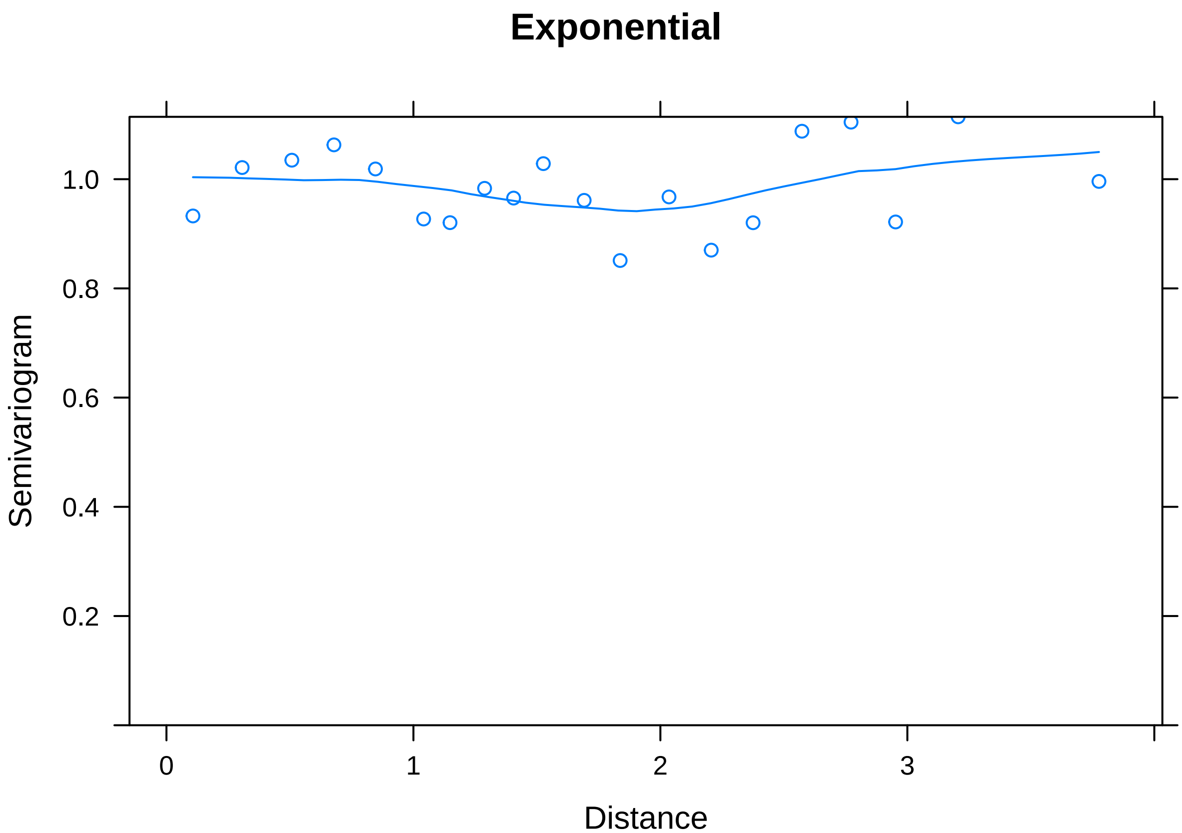
2

Fit two models relating woody cover change (WCI) to MAP and FIRE using gls, one with and one without spatial autocorrelation. Compare them with a Likelihood Ratio test.

```
mod1 <- gls(WCI ~ MAP + FIRE,
            data = treeData)

plot(Variogram(mod1, resType = "n", form = ~XScaled + YScaled), main = "No Autocorrelation")
```




```
modExp <- gls(WCI ~ MAP + FIRE,
              correlation = corExp(form = ~XScaled + YScaled),
              data = treeData)

plot(Variogram(modExp, resType = "n", form = ~XScaled + YScaled), main = "Exponential")
```




```
(anova(mod1, modExp))
```

```
##        Model df       AIC       BIC   logLik   Test  L.Ratio p-value
## mod1       1  4 -240.4739 -228.9732 124.2370
## modExp     2  5 -267.3302 -252.9542 138.6651 1 vs 2 28.85627  <.0001
```

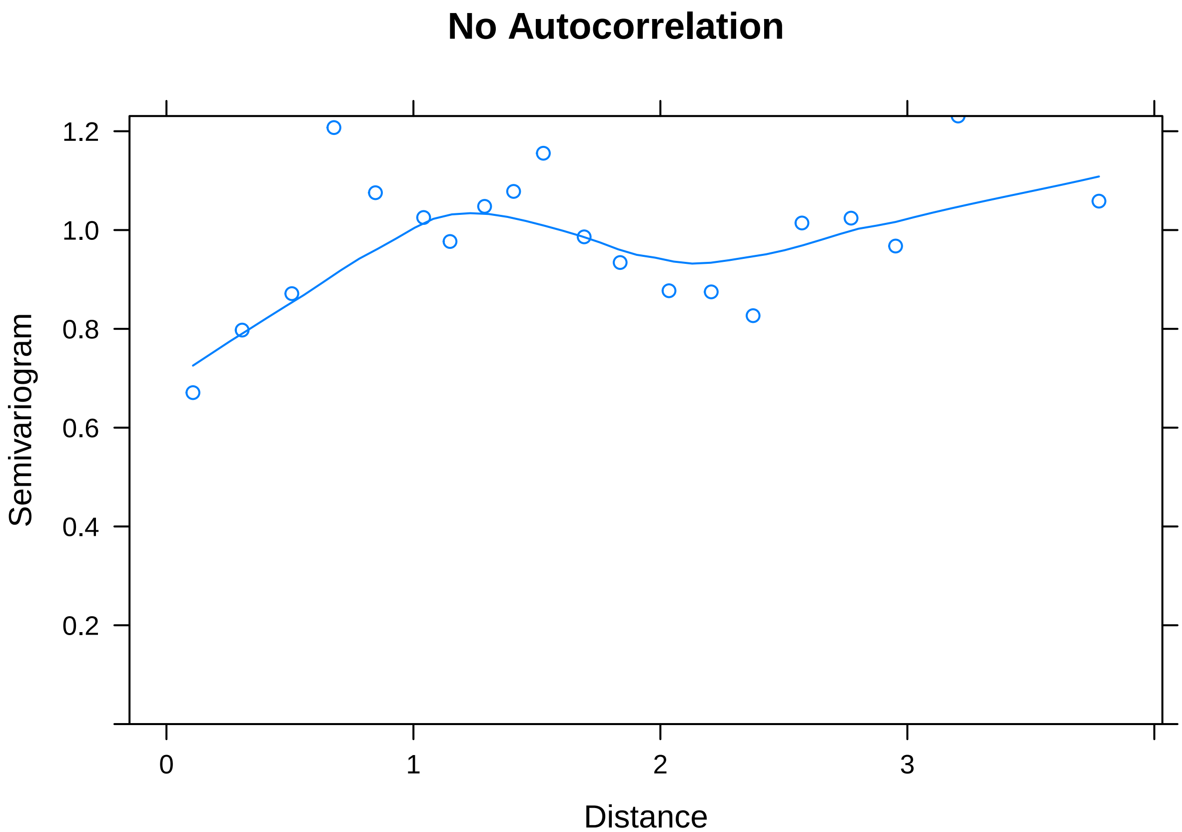
The model with spatial autocorrelation `modExp` has a higher log-Likelihood (138.67 vs. 124.270). It's semivariogram also begins at 1 and is flat, compared to the model without spatial autocorrelation that increases with distance until about 1.5 (agreeing with the results from # 1).

3

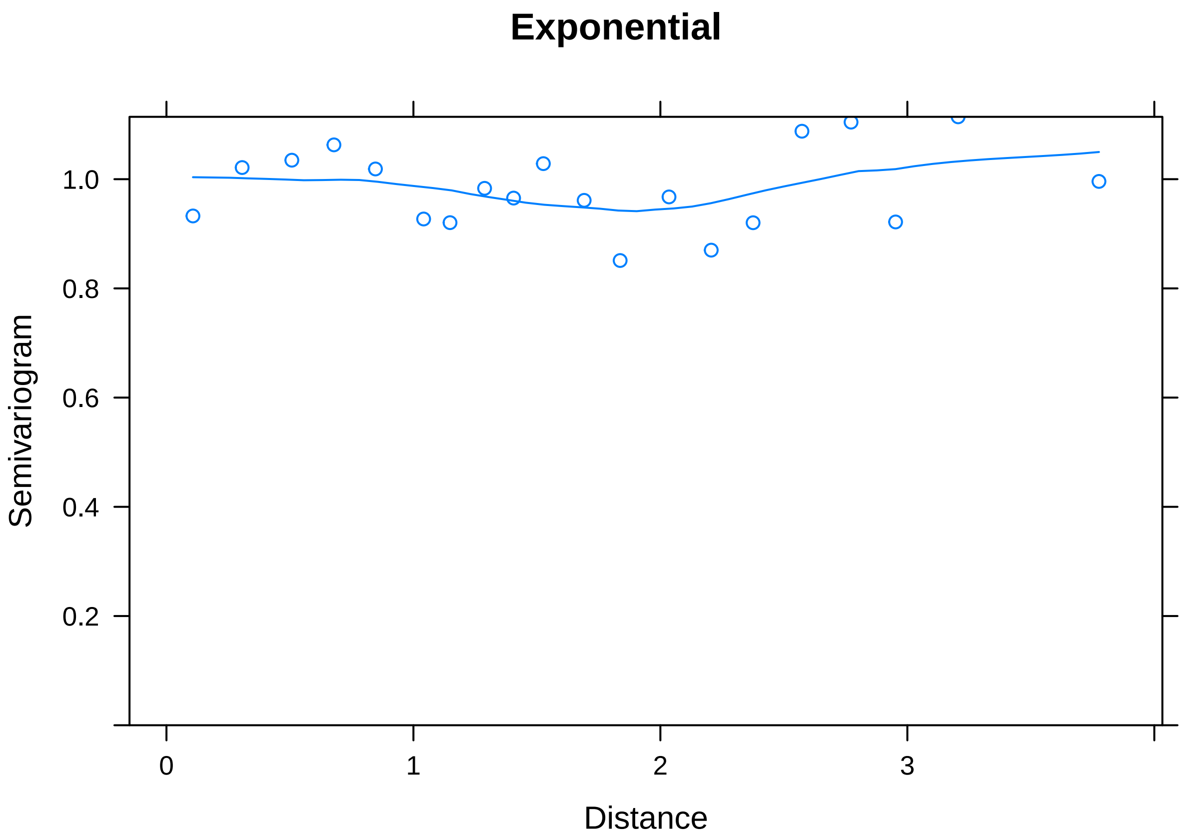
Fit the same model again, but test alternative autocorrelation functions (Spherical, Exponential). Do some fit better than others? How do the semivariograms change?

```
modLin <- gls(WCI ~ MAP + FIRE,
              correlation = corLin(form = ~XScaled + YScaled),
              data = treeData)

modSpher <- gls(WCI ~ MAP + FIRE,
              correlation = corSpher(form = ~XScaled + YScaled),
              data = treeData)

modGauss<- gls(WCI ~ MAP + FIRE,
              correlation = corGaus(form = ~XScaled + YScaled),
              data = treeData)
```

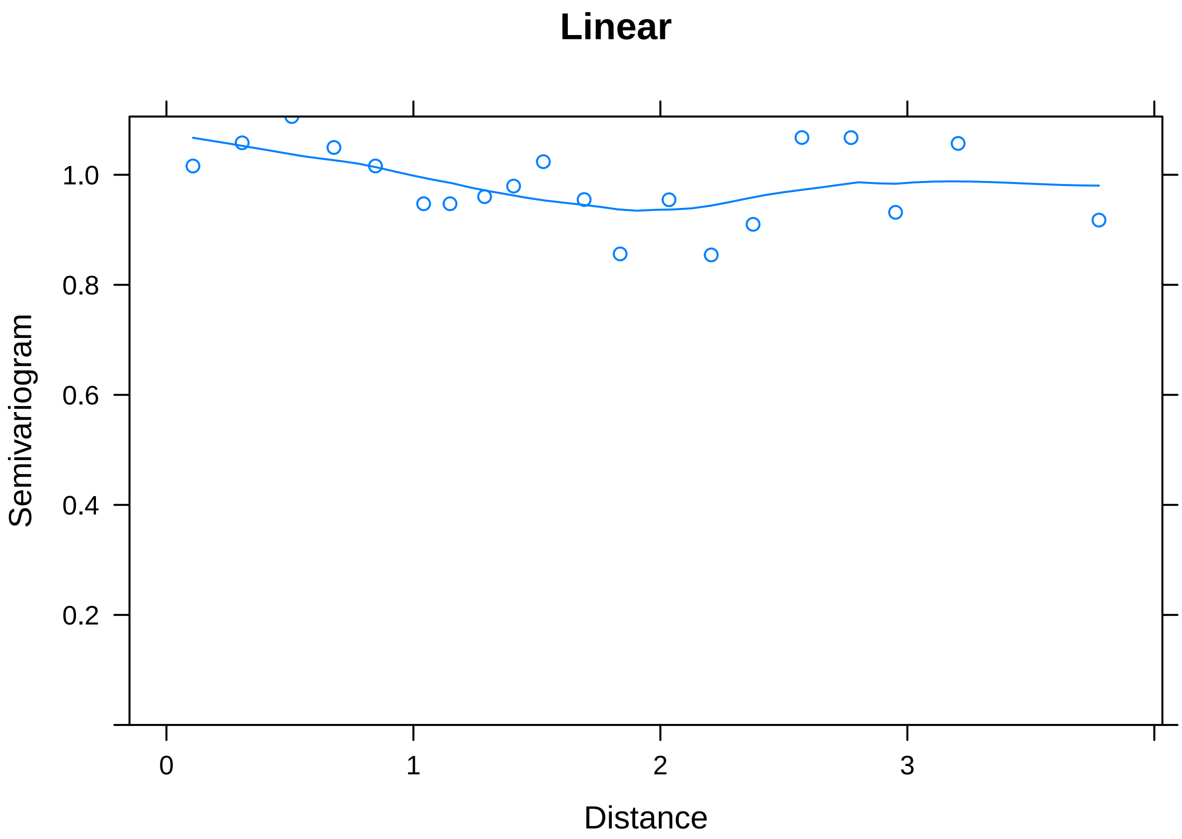
```r
par(mfrow=c(3,2))

plot(Variogram(mod1, resType = "n", form = ~XScaled + YScaled), main = "No Autocorrelation")
```



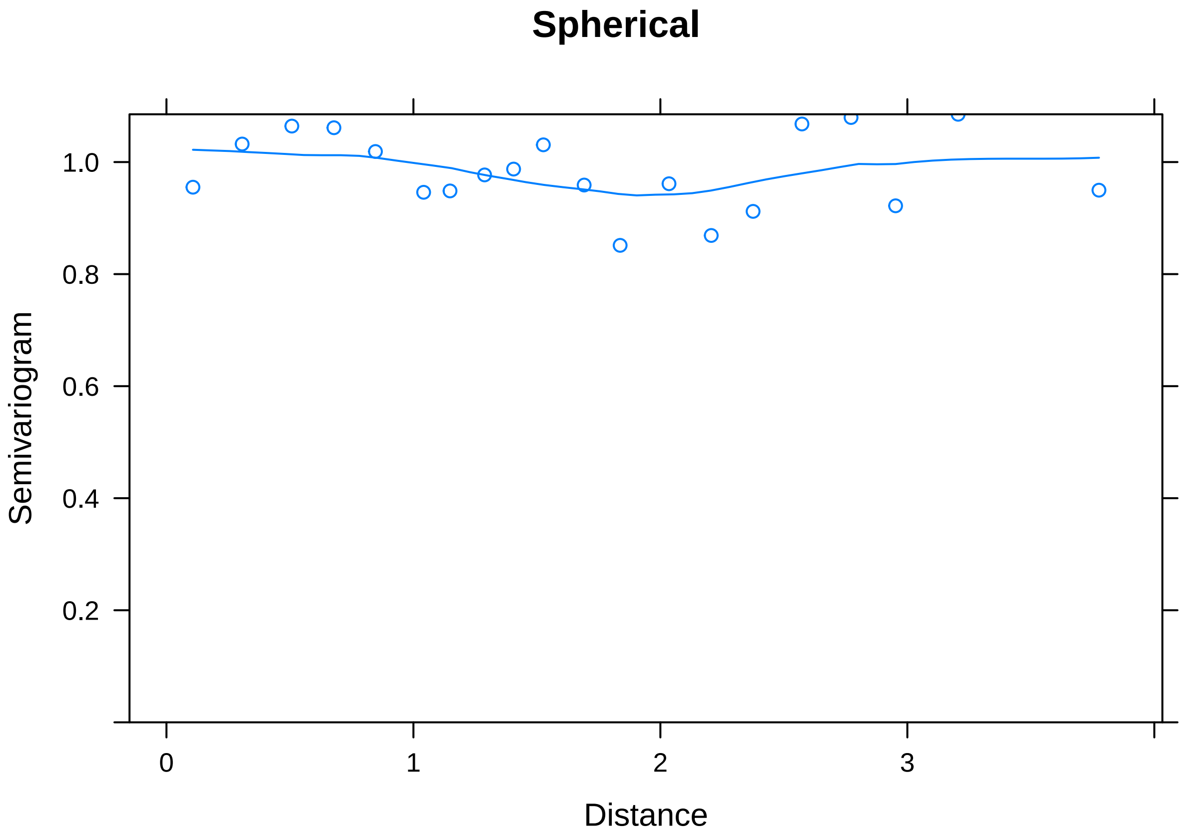

```r
plot(Variogram(modExp, resType = "n", form = ~XScaled + YScaled), main = "Exponential")
```



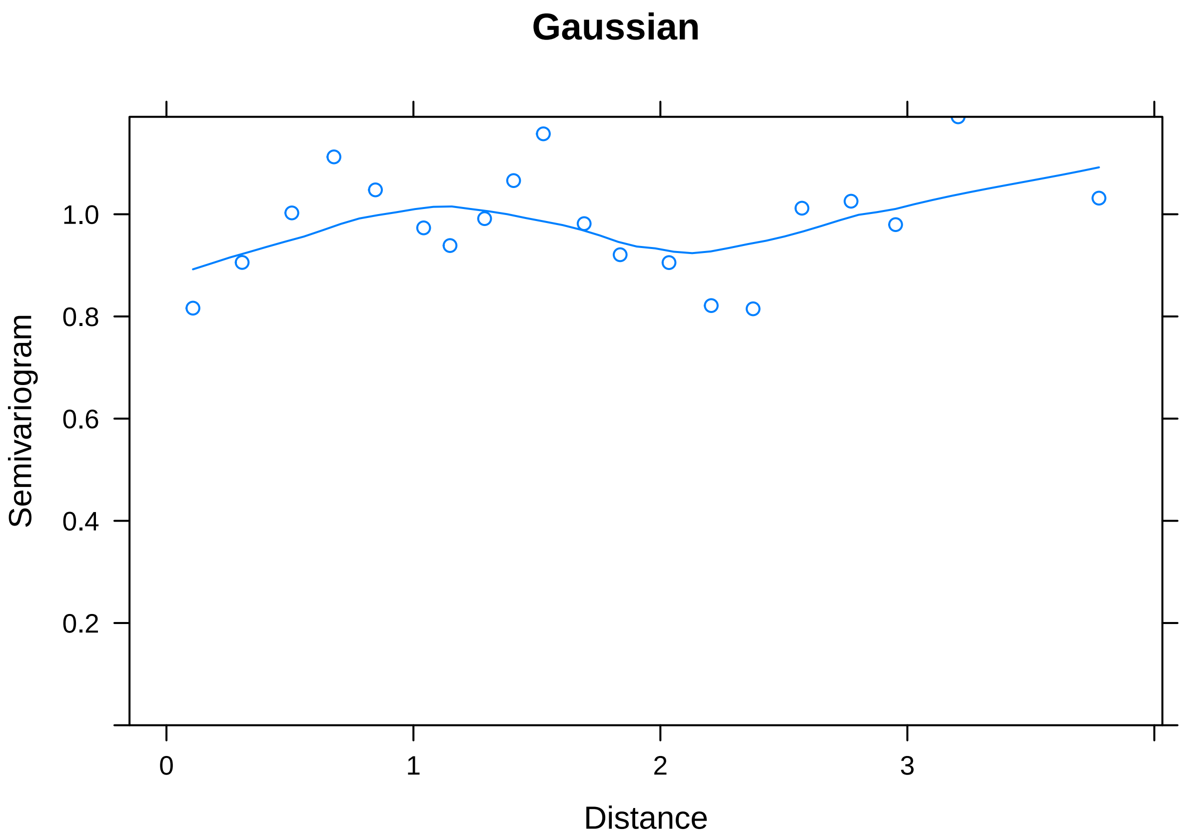

```r
plot(Variogram(modLin, resType = "n", form = ~XScaled + YScaled), main = "Linear")
```



```r
plot(Variogram(modSpher, resType = "n", form = ~XScaled + YScaled), main = "Spherical")
```




```r
plot(Variogram(modGauss, resType = "n", form = ~XScaled + YScaled), main = "Gaussian")
```




```r
anova(mod1, modExp, modLin, modSpher, modGauss)
```

```
##          Model df       AIC       BIC   logLik   Test  L.Ratio p-value
## mod1         1  4 -240.4739 -228.9732 124.2370
## modExp       2  5 -267.3302 -252.9542 138.6651 1 vs 2 28.85627  <.0001
## modLin       3  5 -263.2234 -248.8474 136.6117
## modSpher     4  5 -265.9155 -251.5395 137.9577
## modGauss     5  5 -255.8992 -241.5233 132.9496
```

There isn't a huge difference between most of the autocorrelation functions. The gaussian is visibly worse, and fits worse than the others, but the exponential and spherical fit similarly. The exponential fits marginally better, however.

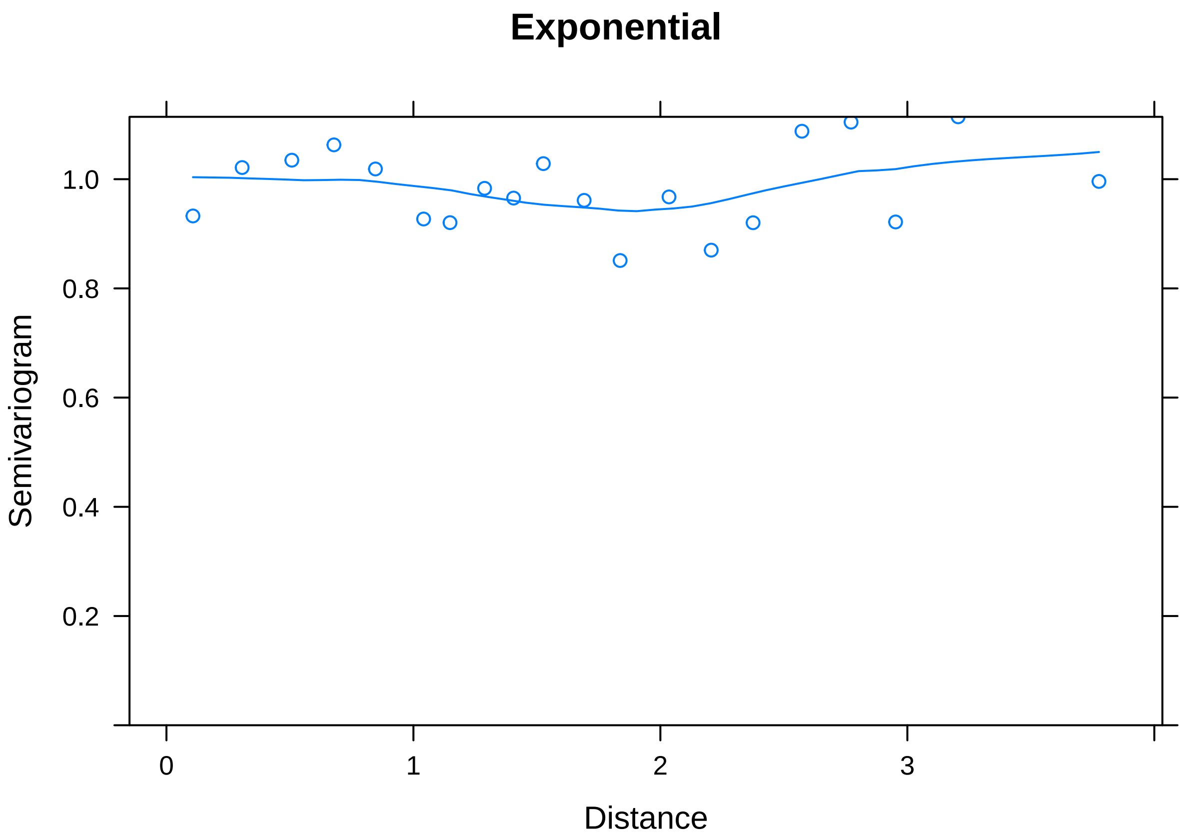
4

Conduct a model selection exercise using gls. What is the top model? Examine the semivariogram of your top model using normalized residuals. Does your model do a good job accounting for autocorrelation?

```r
MuMIn::Weights(AIC(mod1, modExp, modLin, modSpher, modGauss))
```

```
## [1] 9.067910e-07 6.155610e-01 7.897389e-02 3.034362e-01 2.028007e-03
```

```r
plot(Variogram(modExp, resType = "n", form = ~XScaled + YScaled), main = "Exponential")
```



An AIC weighting scheme identifies the model with the exponential spatial autocorrelation as having the best fit, receiving about 61% of the weight. The line is flat and starts at the maximum value, so I think this autocorrelation function performs well.

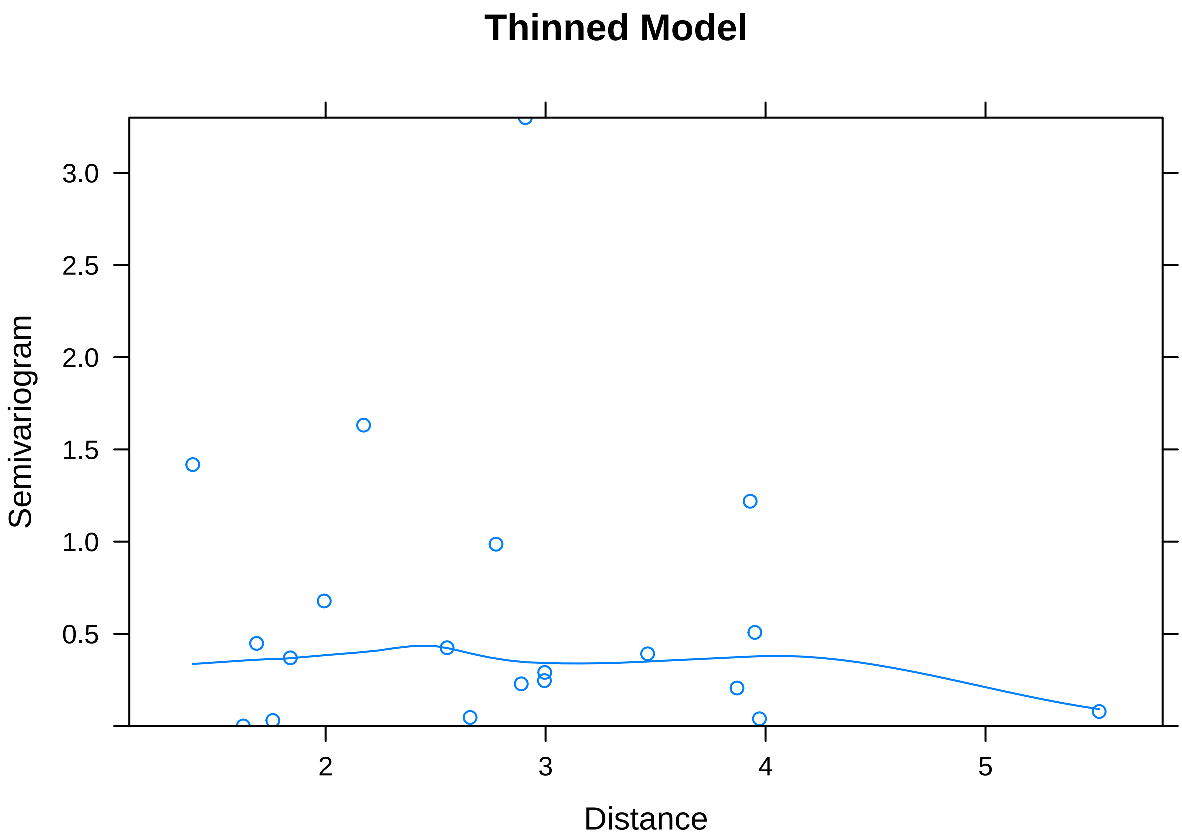
5

Now rerun your models in an OLS framework by thinning your data set. You previously estimated the range of the semivariogram for one of your models. Thin your data using sample so that on average any two points are at least this distance apart. How might you do this?

```
set.seed(5)
distance <- as.matrix(dist(cbind(treeData$XScaled, treeData$YScaled), diag = T))
diag(distance) <- NA

#based on thinning algorithm from 'spThin' package
treeThin <- treeData
while(min(distance, na.rm = T) < 1 & nrow(treeThin) > 1){ #loop until each one is gone
  nearNb <- which(distance < 1.3, arr.ind = T)[,1]
   #drop that point which is closest to the most points
  toRemove <- as.numeric(names(which(table(nearNb) == max(table(nearNb)))))
  if (length(toRemove) > 1){
    toRemove <- sample(toRemove, 1)
  }
  treeThin <- treeThin[-toRemove,]
  distance <- distance[-toRemove,-toRemove]
}
```

This results in a very small data frame of only 7 samples.

```
modThin <- gls(WCI ~ MAP + FIRE,
               data = treeThin)

plot(Variogram(modThin, resType = "n", form = ~XScaled + YScaled), main = "Thinned Model")
```




```
summary(modThin)
```

```
## Generalized least squares fit by REML
##   Model: WCI ~ MAP + FIRE
##   Data: treeThin
##        AIC      BIC    logLik
##   7.083316 4.628494 0.4583418
##
## Coefficients:
##                   Value  Std.Error   t-value p-value
## (Intercept)  0.9900642 0.16072436  6.160013  0.0035
## MAP         -0.0008880 0.00016147 -5.499373  0.0053
## FIRE        -0.5272556 0.11068871 -4.763409  0.0089
##
##  Correlation:
##      (Intr) MAP
## MAP  -0.983
## FIRE -0.370  0.238
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
```

```
## -1.26369948 -0.39136432 -0.09911495  0.42019394  1.30515520
##
## Residual standard error: 0.04922235
## Degrees of freedom: 7 total; 4 residual
```

```r
summary(modExp)
```

```
## Generalized least squares fit by REML
##   Model: WCI ~ MAP + FIRE
##   Data: treeData
##         AIC       BIC   logLik
##   -267.3302 -252.9542 138.6651
##
## Correlation Structure: Exponential spatial correlation
##  Formula: ~XScaled + YScaled
##  Parameter estimate(s):
##      range
## 0.05432723
##
## Coefficients:
##                  Value  Std.Error   t-value p-value
## (Intercept)  0.3488998 0.07926980  4.401421  0.0000
## MAP         -0.0003333 0.00009445 -3.529019  0.0006
## FIRE        -0.0965675 0.05459533 -1.768788  0.0793
##
##  Correlation:
##      (Intr) MAP
## MAP  -0.984
## FIRE  0.250 -0.360
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.39917755 -0.71467026 -0.04645037  0.74184278  2.29411309
##
## Residual standard error: 0.09084652
## Degrees of freedom: 134 total; 131 residual
```

The thinned model has a significant negative effect of fire, while the model including exponential correlation did not. The semiovariogram is flat, however, so it does seem that thinning has reduced autocorrelation. Due to the use of 'sample' in the thinning function, however, the dataset is randomly thinned, and rerunning this with different seeds will result in different thinned datasets, and some spurious looking semivariograms.

In this case, thinning the data resulted in a loss of nearly 95% of our data, losing a lot of other valuable information in the process. Therefore, I think fitting a model including autocorrelation is the better choice in this instance.