

HW6

Mitch Oshima

February 17, 2016

1.) After loading in the BCI data set I used the colSums function to determine how many of each species was recorded and selected *Virola.multiflora* as a rare species and *Tabernaemontana.arborea* as a common species (the call for colSums() was left out of the code to eliminate unnecessary output in the PDF). I isolated these species from the data set and plotted the BCI_xy coordinates using the count of species to scale the size of the points. I was able to see that the *virola* species grows more along the bottom of the plot. Meanwhile the *tabernaemontana* species grows more in the upper right quadrant of the plot.

```
library(nlme)
```

```
## Warning: package 'nlme' was built under R version 3.1.3
```

```
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 3.1.3
```

```
## Loading required package: permute
```

```
## Warning: package 'permute' was built under R version 3.1.3
```

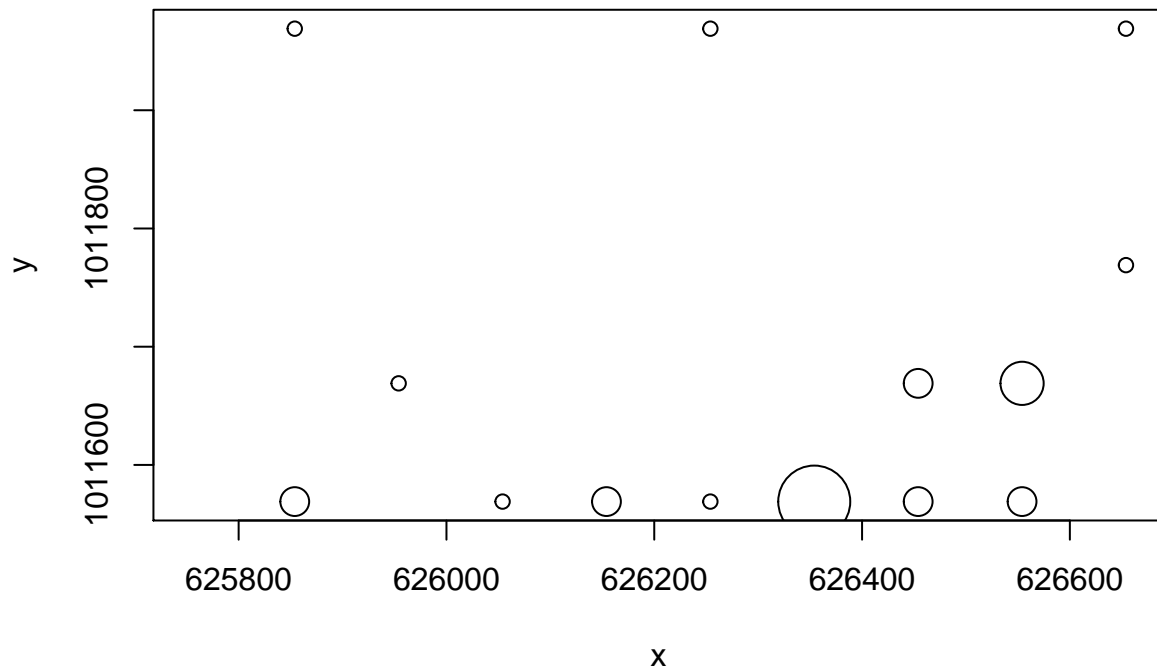
```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 3.1.3
```

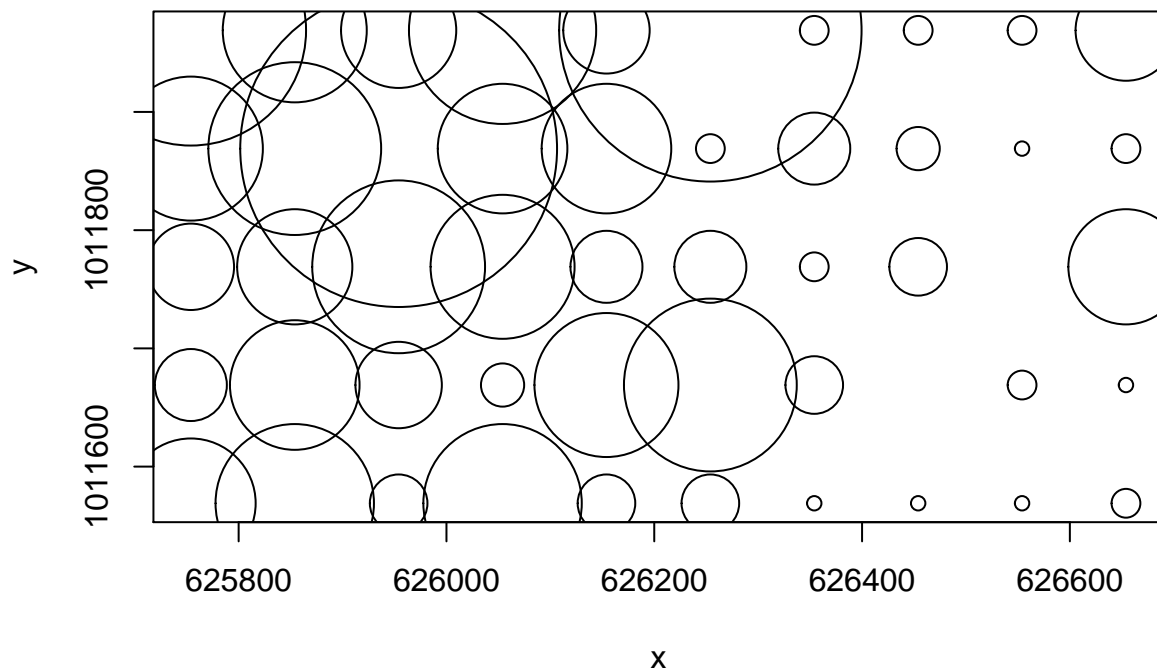
```
## This is vegan 2.3-3
```

```
data(BCI)
## UTM Coordinates (in metres)
BCI_xy = data.frame(x = rep(seq(625754, 626654, by=100), each=5),
                    y = rep(seq(1011569, 1011969, by=100), len=50))

#Virola.multiflora is rare species. Tabernaemontana.arborea is common species.
virMult = BCI$Virola.multiflora
tabArb = BCI$Tabernaemontana.arborea
virPlot = plot(BCI_xy, cex = virMult)
```



```
tabPlot = plot(BCI_xy, cex = tabArb)
```



2.) First, I decided to use the *Cordia lasiocalyx* species as the initial predictor. The second model uses each of the tree species as predictors. For the remainder of the analysis the models designated with a “1” uses the single predictor while the models ending with a “2” represents the models using multiple predictors. I then plotted the variogram for each of these models. The first plot fits the data pretty well. However, there is a point that seems to be pulling the model towards it causing the fit line to flatten out near the top. When looking at the variogram plot for the second model we see a similar effect as in the first plot. The line of fit is being pulled down by the outlier point at the end. However, I limited the maximum distance which greatly improved the model’s performance.

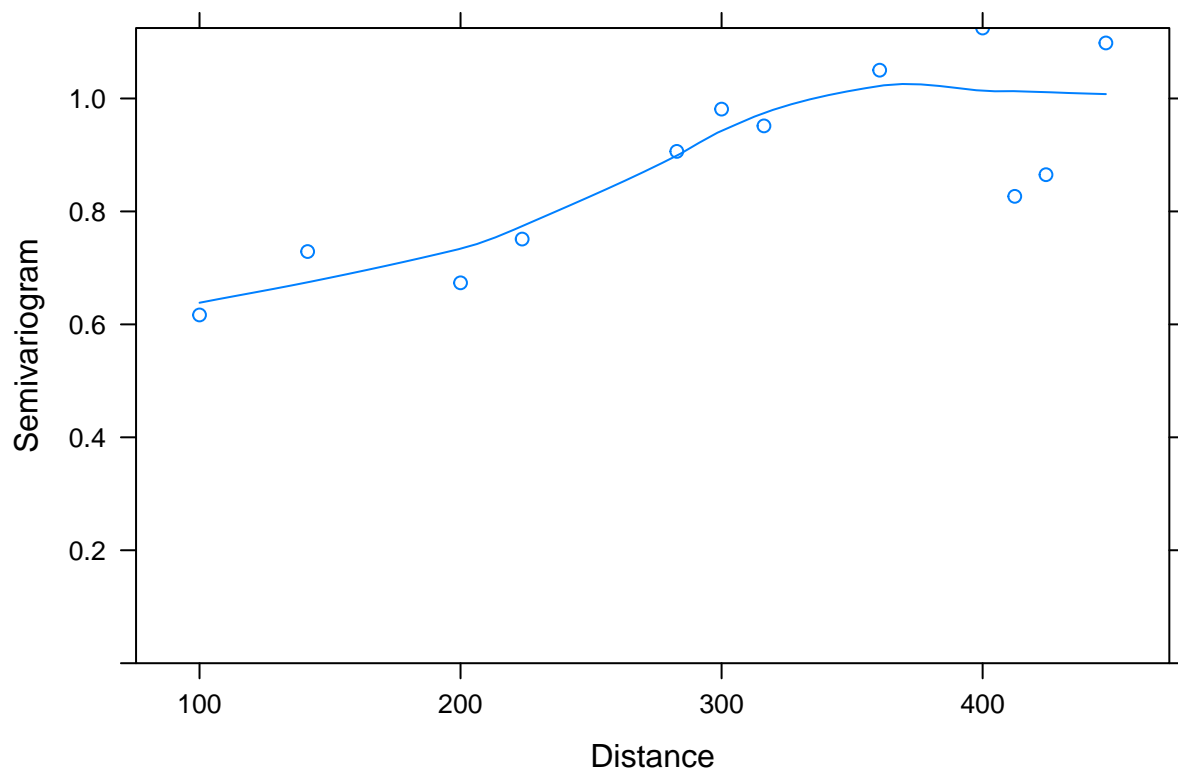
```

sp_ids = c("Drypetes.standleyi", "Cordia.lasiocalyx", "Hirtella.triandra",
           "Picramnia.latifolia", "Quassia.amara",
           "Tabernaemontana.arborea", "Trattinnickia.aspera",
           "Xylopia.macrantha")
BCI_dat = data.frame(BCI[,sp_ids], BCI_xy)

xy_dist = dist(BCI_xy)
max_dist = max(xy_dist)/2

sp_lm1 = gls(Drypetes.standleyi~Cordia.lasiocalyx, data = BCI_dat)
par(mfrow=c(1,1))
plot(Variogram(sp_lm1, form= ~ x + y, maxDist = max_dist))

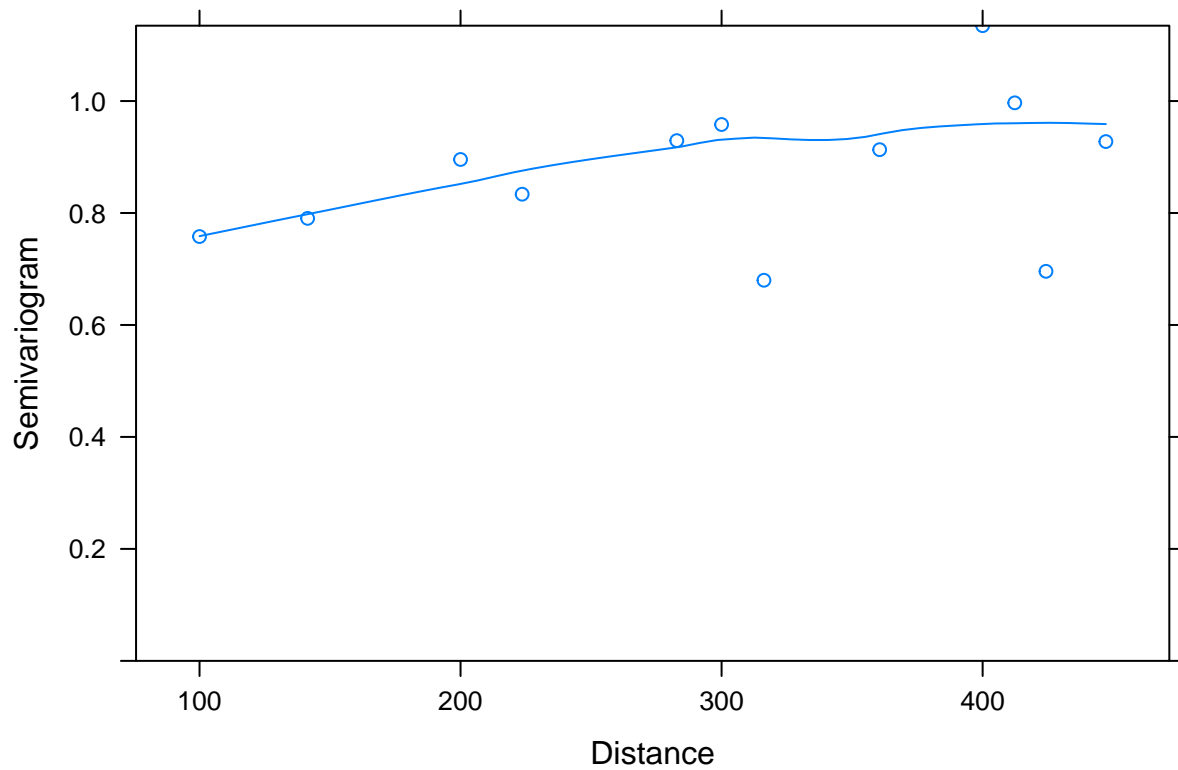
```



```

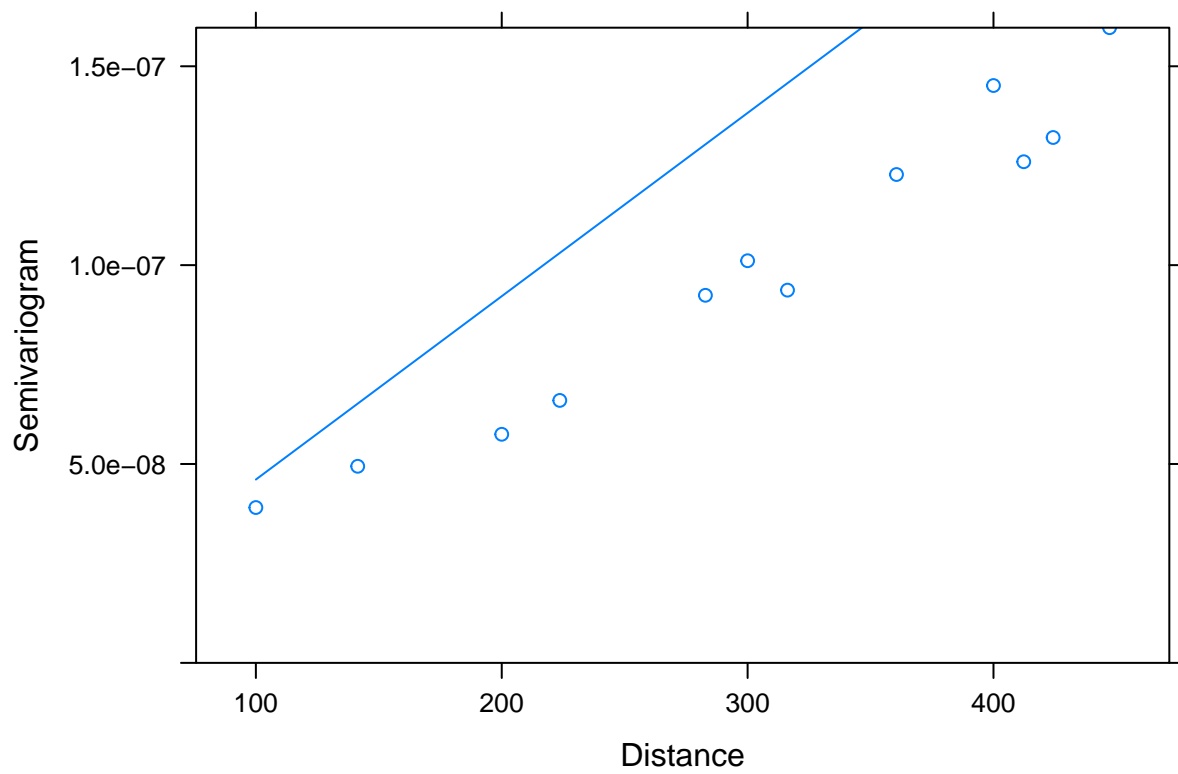
sp_lm2 = gls(Drypetes.standleyi~Cordia.lasiocalyx+Hirtella.triandra+Picramnia.latifolia+Quassia.amara+T
par(mfrow=c(1,1))
plot(Variogram(sp_lm2, form= ~ x + y, maxDist = max_dist))

```

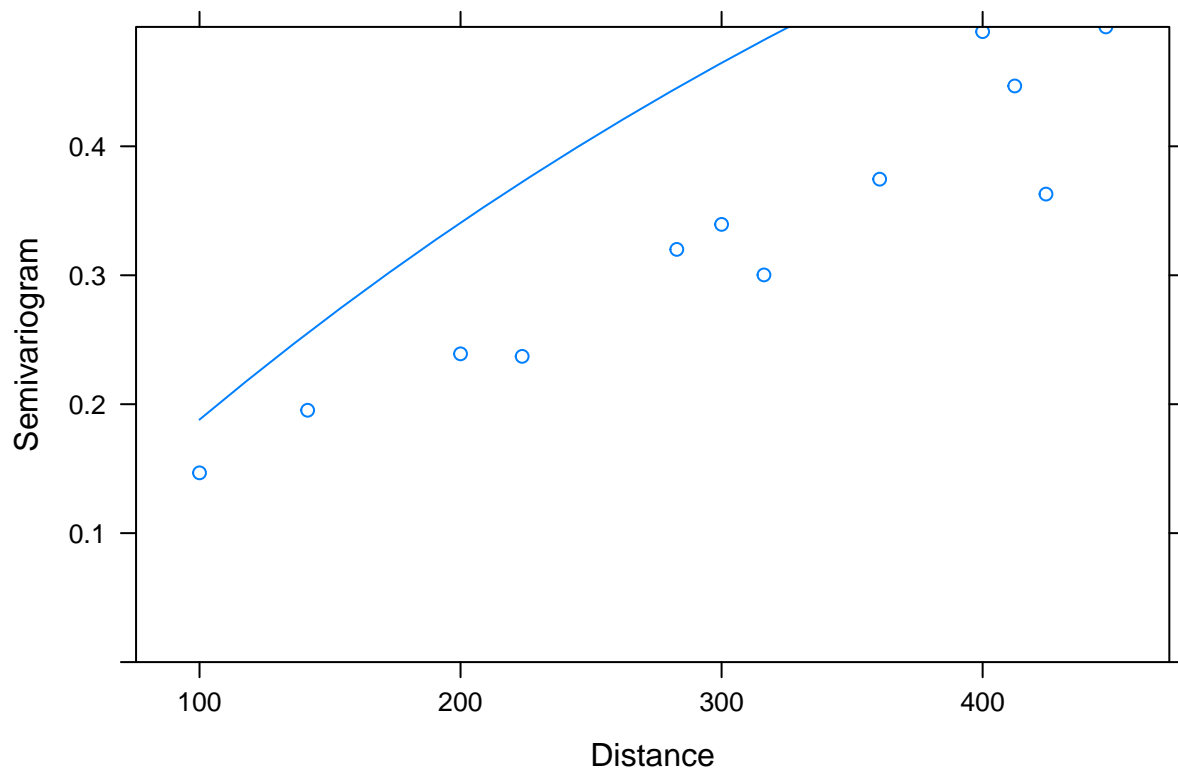


After including the error term in the model the model seems to be weaker. I applied two different error structures to each of the model. First, I used an exponential error model and while the shape seems to fit the data there once again seems to be an influential point that is shifting the fit line upwards. However, once the nugget was included in the model the fit was much stronger with this error structure particularly in the single predictor model.

```
sp_exp1 = update(sp_lm1, corr=corExp(form=~x + y))  
plot(Variogram(sp_exp1, maxDist = max_dist))
```



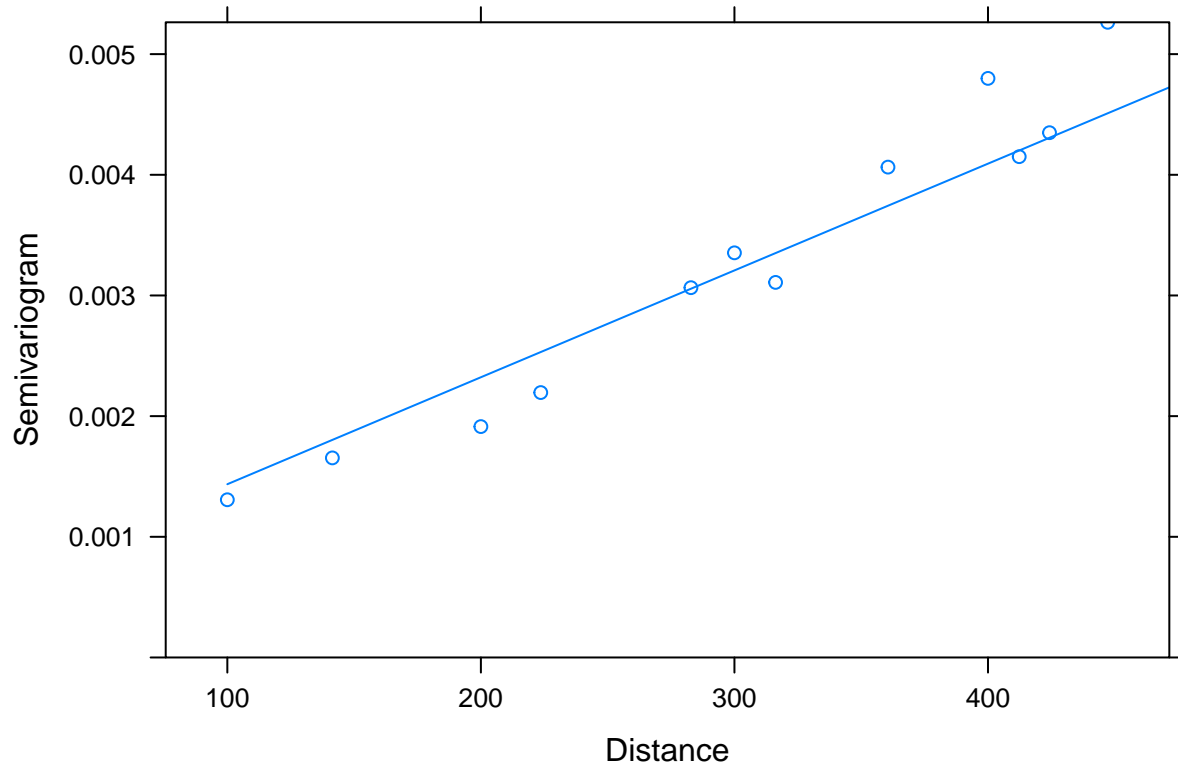
```
sp_exp2 = update(sp_lm2, corr=corExp(form=~x + y))
plot(Variogram(sp_exp2, maxDist = max_dist))
```



```
sp_expNug1 = update(sp_lm1, corr=corExp(form=~x + y, nugget = T),control=glControl(opt='optim', msVer
```

```
## initial value 182.562784  
## iter 10 value 175.347024  
## final value 175.329770  
## converged
```

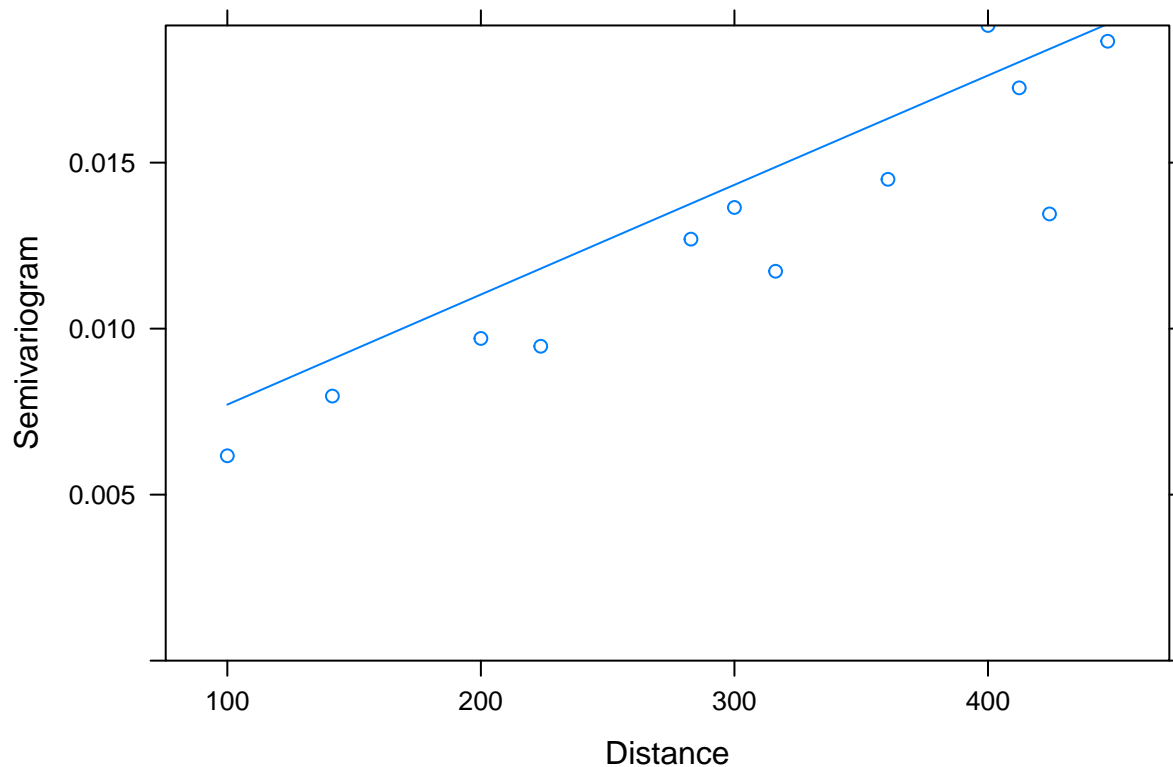
```
plot(Variogram(sp_expNug1, maxDist = max_dist))
```



```
sp_expNug2 = update(sp_lm2, corr=corExp(form=~x + y, nugget = T),control=glControl(opt='optim', msVer
```

```
## initial value 161.334238  
## iter 10 value 158.897416  
## iter 20 value 158.881464  
## final value 158.879778  
## converged
```

```
plot(Variogram(sp_expNug2, maxDist = max_dist))
```



I then used the anova function to compare these models to one another. With both the single preidictor and the multiple predictors including the error term greatly improved the model as did the inclusion of the nugget as evidenced in the change in the log of likelihood value. The inclusion of the error terms improved these models by shifting the points to a more linear shape and reducing the variation at the upper extremes of the distance variable.

```
anova(sp_lm1, sp_exp1, sp_expNug1)
```

##	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
##	sp_lm1	1	3	335.1246	340.7382	-164.5623		
##	sp_exp1	2	4	310.6438	318.1286	-151.3219	1 vs 2	26.480806 <.0001
##	sp_expNug1	3	5	311.0600	320.4160	-150.5300	2 vs 3	1.583808 0.2082

```
anova(sp_lm2, sp_exp2, sp_expNug2)
```

##	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
##	sp_lm2	1	9	307.1163	322.7554	-144.5582		
##	sp_exp2	2	10	301.6062	318.9829	-140.8031	1 vs 2	7.510175 0.0061
##	sp_expNug2	3	11	301.9683	321.0826	-139.9841	2 vs 3	1.637897 0.2006