Exercise 1: Exploratory data analysis using R for the scallop data set. These data are the counts of scallops (marine bivalve mollusks) in the Atlantic Ocean, off the northeast coast of the U.S. (Ecker and Heltsche, 1994). You are to carry out some exploratory data analysis, including some rudimentary trend detection, as described below.

The data can be found in the file, `scallops.txt`, which is posted on Canvas. (This data is from the text book by Banerjee, Carlin, and Gelfand.) The columns we will be using are the 'lat', 'long', and 'tcatch' (total catch) columns.

1. Read the data file and display summary information for total catch and comment briefly.

Solution:
Reading in the data and observing the summary report we can see that the total catch varies wildely from place to place. With the maximum observation reporting 7084. However noticing that the mean is a lot larger that the median, and even the 3rd quartile it clear that the observation with the larges total catch is probably an outlier and warrants further investigation. In general I would expect the histogram of tcatch to be very right skewed, and would be interested to see if those large values have any spatial correlation.
Code:

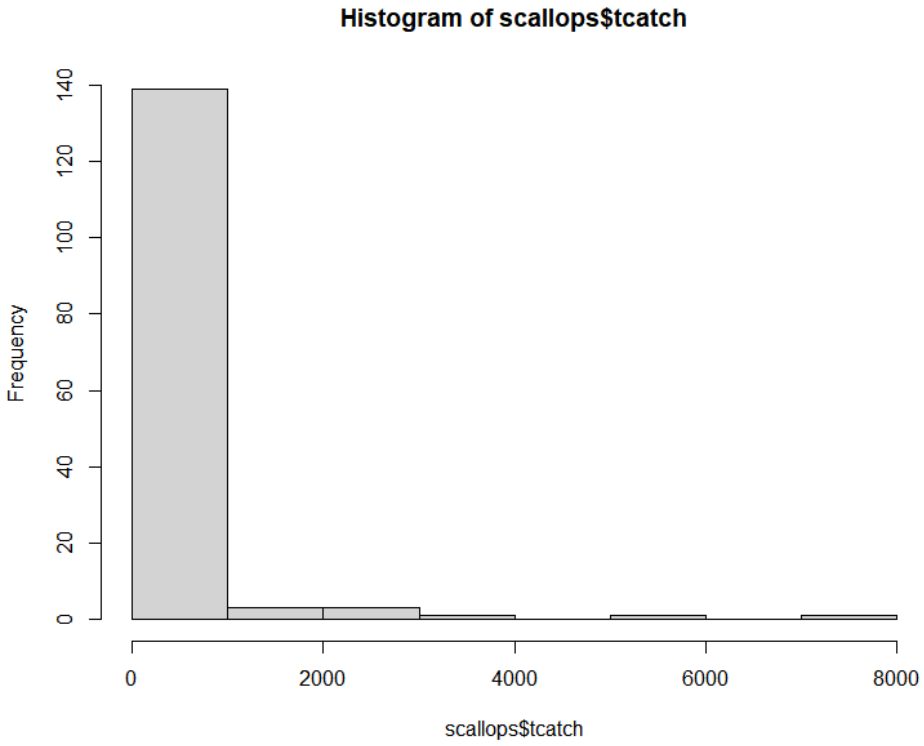
```
> scallops <- read.table("scallops.txt", header=TRUE)
```

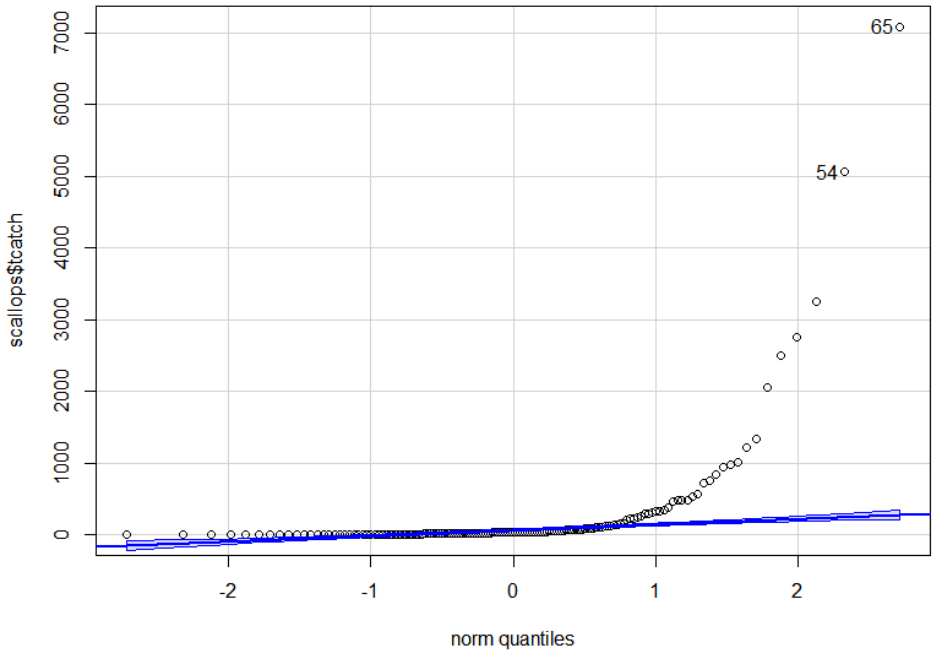
```
> names(scallops)
[1] "strata"   "sample"   "lat"      "long"     "tcatch"   "prerec"
[7] "recruits" "lgcatch"
```

```
> summary(scallops)
     strata          sample           lat             long
 Min.   :6220   Min.   :  1.0   Min.   :38.60   Min.   :-73.70
 1st Qu.:6260   1st Qu.:106.8   1st Qu.:39.46   1st Qu.:-73.14
 Median :6290   Median :147.0   Median :39.98   Median :-72.74
 Mean   :6288   Mean   :131.8   Mean   :39.91   Mean   :-72.72
 3rd Qu.:6310   3rd Qu.:185.2   3rd Qu.:40.41   3rd Qu.:-72.31
 Max.   :6350   Max.   :224.0   Max.   :40.92   Max.   :-71.52


     tcatch           prerec          recruits         lgcatch
 Min.   :   0.0   Min.   :   0.00   Min.   :   0.00   Min.   :0.000
 1st Qu.:   8.0   1st Qu.:   1.00   1st Qu.:   5.00   1st Qu.:2.197
 Median :  30.0   Median :   8.00   Median :  21.50   Median :3.434
 Mean   : 274.6   Mean   : 156.55   Mean   : 118.06   Mean   :3.483
 3rd Qu.: 115.2   3rd Qu.:  48.25   3rd Qu.:  73.75   3rd Qu.:4.756
 Max.   :7084.0   Max.   :4487.00   Max.   :2597.00   Max.   :8.866
```

2. Create a histogram and a normal probability plot of the total catch: Based on these plots, does it look plausible that the data are coming from a normal distribution? Explain briefly.

Solution:
Like we described in the summary report, the data is heavily right skewed with several relatively large values. Looking at the histogram and qq norm plot verifies that this data is likely not a situation where we have a normal distribution with some outliers, but rather something like a log-normal distribution.





Exercise 2: Log-transforming the scallops data.

1. Create a new column in the data set for a log-transformed total catch variable:
   `scallops$logcatch <- log( 1 + scallops$tcatch )`.
   Why did you need to add 1 before taking the logarithm?

   Solution:
   Firstly we must add 1 to the total catch data before performing the transformation because the log(0) goes to minus infinity, and that goes for any base. Beyond that shifting the data by one shouldn't have any appreciable difference in our EDA. After log transforming our data, looking at the summary report we can see that it clearly reflects a more normal distribution with the mean and median being very close to each other and near the middle of the range.
   Code:

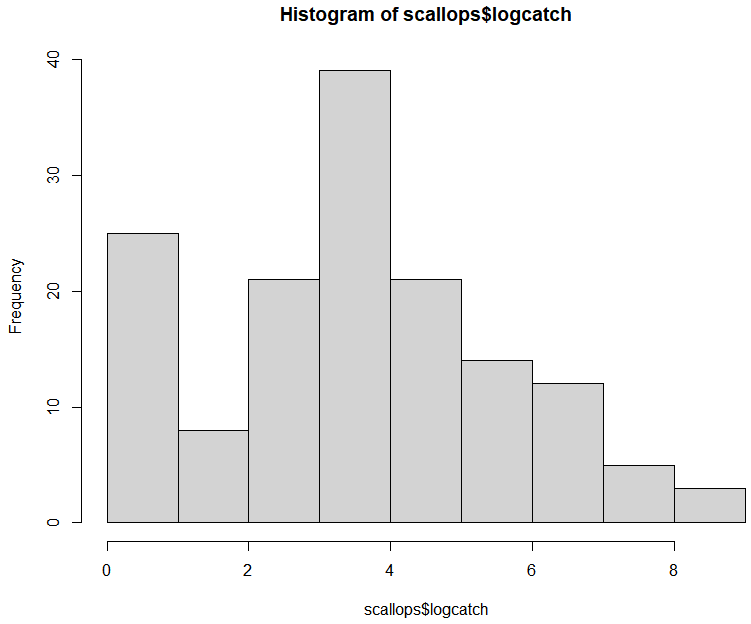
```
> scallops$logcatch <- log( 1 + scallops$tcatch)

> summary(scallops$logcatch)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.197   3.434   3.483   4.756   8.866
```

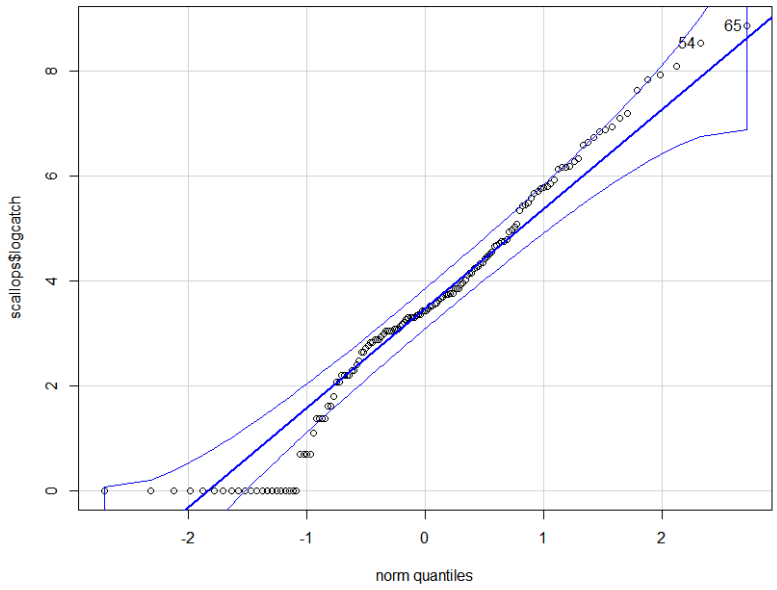
2. Is this natural log or log base 10? Would your plot differ in any meaningful way if you used the other base for the logarithm? Explain briefly.

   Solution:
   Looking at the documentation we can see that when we use the log() function without the base parameter the default is to use the natural log to transform the data. As discussed in exercise 7 from homework 2 that converting between the natural log and log base 10 transformation, involves multiplying the whole data set by a constant. Therefore in terms of describing the histogram, the log base 10 transformation will produce a more compact histogram, with a smaller range.

3. Create a histogram and a normal probability plot of logcatch; do the data appear more normally distributed than before? Explain briefly.

   Solution:
   Looking at the histogram we can see that we definitely have a more normal looking distribution. This conclusion is corroborated by the fit of the qq plot. The large quantities of 0 catch observations are something that we just have to deal with. When we go out and sample locations we are not guaranteed to sample the signal(is the P(catching scallops = 0) or is our observation outside of the support of the scallop catching probability distribution?).





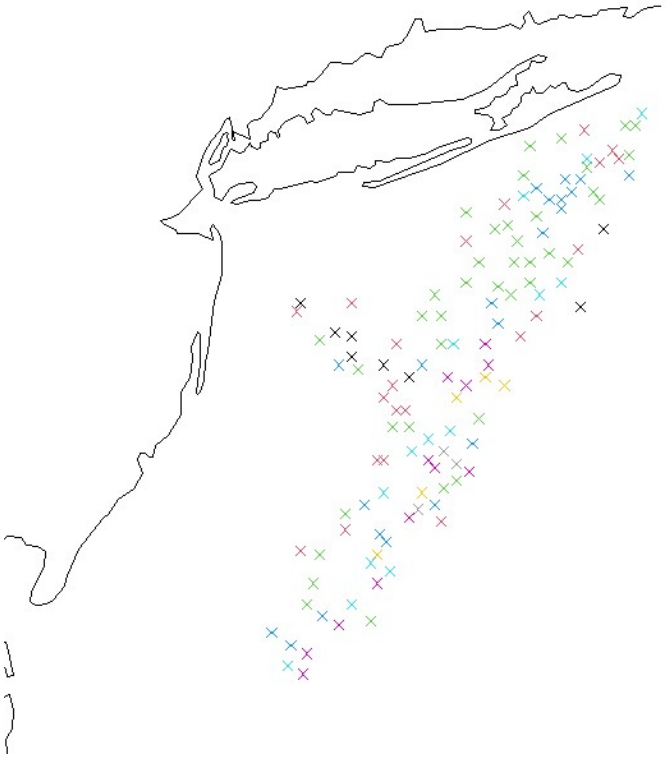
4. Plot the data on top of a map, as follows.

```
install.packages("maps")
library(maps)
```

```
map("usa",xlim=c(-74,-71),ylim=c(38.0,41.5))
points(scallops$long,scallops$lat, pch=4, col="gray")
```

Solution:
Plotting the data, we get the following (I added log catch data to the col parameter, out of curiosity.)

Figure 1: Scallop Locations



Exercise 3: MLR with the scallops data.

1. Simple trend detection, using multiple linear regression (MLR). Fit the following MLR model, which assumes independent N(0,σ^2) errors. Be sure to state the fitted model. (It's not enough to simply include the computer output, although you should

be doing that, too.) The model allows for linear and quadratic terms in latitude (y) or longitude (x).

$$\mathbb{E}(\text{logcatch}) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy.$$

The following R code fits this model and summarizes the parameter estimates and their standard errors, as well as p-values for the parameters:

```
scallops$clons <- scallops$long - mean(scallops$long)
scallops$clats <- scallops$lat - mean(scallops$lat)
myfit <- lm( logcatch ~ clons+clats + I(clons^2) +
             I(clats^2) + I(clats*clons), data=scallops )
summary(myfit)
```

Does it appear that any trend terms might be appropriate? Explain. (I am looking for a formal hypothesis test or tests.)

Solution:
Looking at the initial model summary, it appears as though the first order predictors have low significance. It could be that the second order terms are capturing a larger portion of the variance, beyond that adding those terms makes the model harder to interpret especially the interaction term. Performing partial F-tests with the anova() function firstly on a first order model, and the second order model with no interaction, we found that the second order predictors were significant. Testing the full second order model against the second order model with no interaction we found that the interaction term was also significant.
Code:

```
> scallops$clons <- scallops$long - mean(scallops$long)
> scallops$clats <- scallops$lat - mean(scallops$lat)
> myfit <- lm( logcatch ~ clons+clats + I(clons^2) +
             I(clats^2) + I(clats*clons), data=scallops )
> summary(myfit)

Call:
lm(formula = logcatch ~ clons + clats + I(clons^2) + I(clats^2) +
    I(clats * clons), data = scallops)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5381 -1.2812 -0.0044  1.2508  4.9780

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.4198     0.2520  17.541  < 2e-16 ***
clons        -0.9275     0.5419  -1.712 0.089139 .
clats        -0.1321     0.4550  -0.290 0.771917
```

```
I(clons^2)          -5.0811     0.8023  -6.333 2.97e-09 ***
I(clats^2)          -3.8192     0.9868  -3.870 0.000165 ***
I(clats * clons)     7.9139     1.5812   5.005 1.63e-06 ***

Residual standard error: 1.884 on 142 degrees of freedom
Multiple R-squared:  0.2739,    Adjusted R-squared:  0.2484
F-statistic: 10.71 on 5 and 142 DF,  p-value: 9.425e-09
```

```
> myfit1 <- lm( logcatch ~ clons + clats, data=scallops )
> myfit2 <- lm( logcatch ~ clons+clats + I(clons^2) + I(clats^2), data=scallops )


> anova(myfit1, myfit2)
Analysis of Variance Table

Model 1: logcatch ~ clons + clats
Model 2: logcatch ~ clons + clats + I(clons^2) + I(clats^2)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    145 653.72
2    143 592.72  2    60.994 7.3577 0.0009089 ***
```

```
> anova(myfit2, myfit)
Analysis of Variance Table

Model 1: logcatch ~ clons + clats + I(clons^2) + I(clats^2)
Model 2: logcatch ~ clons + clats + I(clons^2) + I(clats^2) + I(clats *
    clons)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    143 592.72
2    142 503.84  1    88.884 25.051 1.628e-06 ***
```

---

2. Why did I center the latitudes and longitudes? What would happen if I fitted the model without first centering the latitudes and longitudes? (Try it.)

Solution:

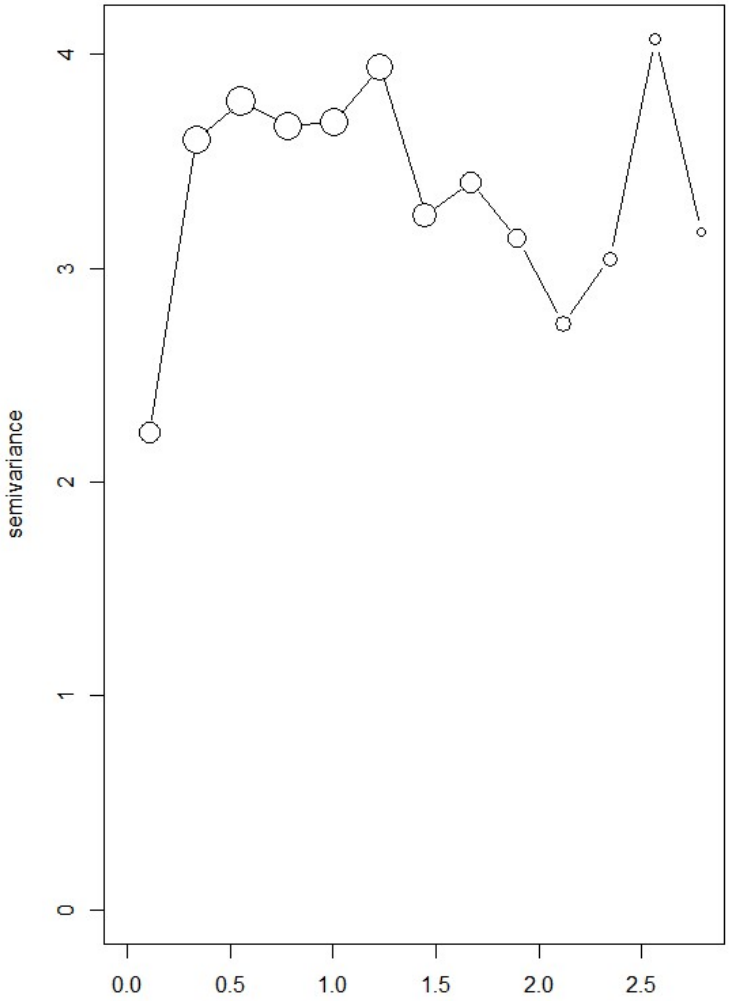
Exercise 4: Choosing and fitting a (semi)variogram for the scallops data.

1. Plot the robust estimator for the variogram (after detrending, if necessary). You may need to tweak this code:

```
geo.scallops <- as.geodata(
  cbind(scallops$clons, scallops$clats, scallops$logcatch ))
robust_est <- variog( geo.scallops,trend="2nd", estimator.type="modulus")
plot(robust_est,pts.range=c(1,3),type='b')
```

Solution:
From the previous problem we found that the second order model explained a significant amount of the variance. Plotting the robust estimated semi variogram we get the following,

Figure 2: Robust estimator for the 2nd order semivariogram



2. Which theoretical semi-variogram appears to be most appropriate for these data? (Exponential, gaussian, spherical, or other?) Explain. (It's entirely possible that there is no clear 'winner'.) Does there appear to be a nugget?

Solution:
Using the eyefit() function to get a sence of which semi-variogram model might be the most apporpriate we get the following plots of the potential exponential, gaussian, and spherical semi-variograms. In every case we had a nugget of around 1. From the shape of the model and the behavior of the parameters I feel as though the gaussian model is worth further exploration.
Code:

```
> eyefit(robust_est)
    cov.model sigmasq  phi tausq kappa kappa2     practicalRange
1 exponential    2.42 0.15   1.1  <NA>   <NA> 0.449362704023068

> eyefit(robust_est)
  cov.model sigmasq  phi tausq kappa kappa2     practicalRange
1  gaussian    2.42 0.23   1.1  <NA>   <NA> 0.398089900877426

> eyefit(robust_est)
  cov.model sigmasq  phi tausq kappa kappa2 practicalRange
1 spherical    2.64 0.98  0.99  <NA>   <NA>           0.98
```

Figure 3: eyefit() Exponential Semi-variogram

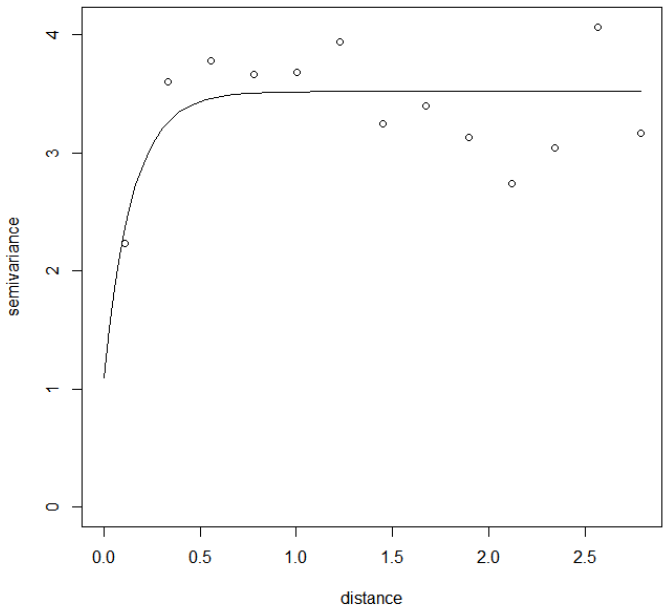


Figure 4: eyefit() Gaussian Semi-variogram

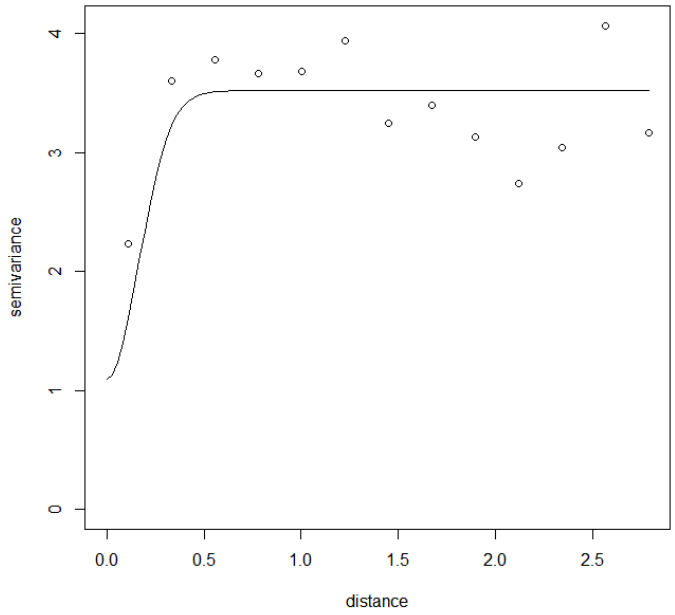
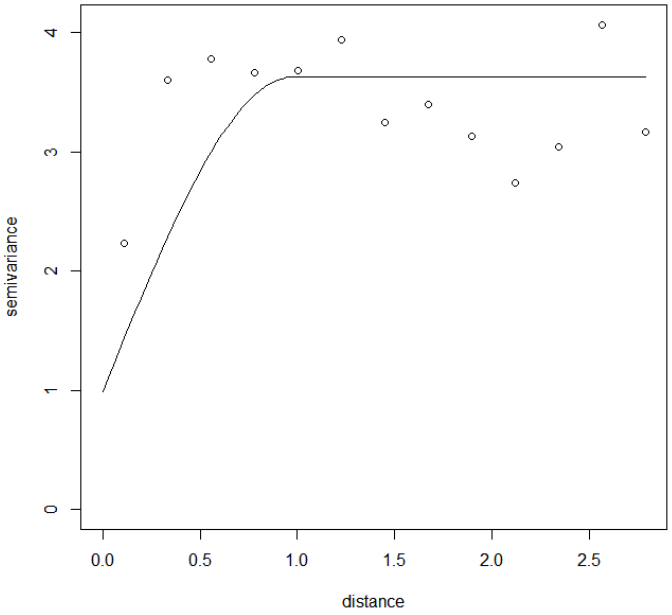


Figure 5: eyefit() Spherical Semi-variogram



3. Fit your 'winning' type of theoretical variogram using the weighted least squares

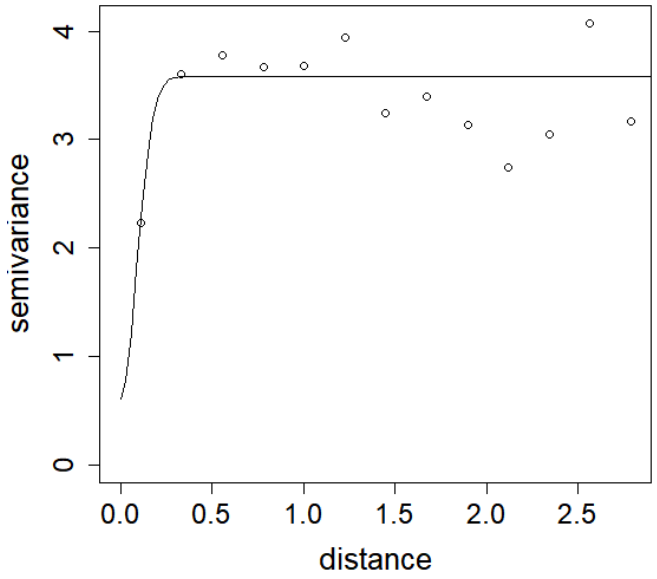
approach. State the resulting estimate for the theoretical variogram. You may need to tweak this code:

```
my_WLS_fit <- variofit( robust_est,
                        ini.cov.pars=c(2.0,.16),
                        cov.model="exponential",
                        fix.nugget=FALSE, nugget=2,
                        max.dist=1.9)
my_WLS_fit$cov.pars; my_WLS_fit$nugget
```

Solution:
Fitting the WLS estimator for a gaussian semi-variaogram we get the following semi-variogram,
Code:

```
> my_WLS_fit <- variofit( robust_est,
                          ini.cov.pars=c(2.42,.23),
                          cov.model="gaussian",
                          max.dist=3)
variofit: covariance model used is gaussian
variofit: weights used: npairs
variofit: minimisation function used: optim

> my_WLS_fit$cov.pars; my_WLS_fit$nugget
[1] sigamsq = 2.9763894    phi = 0.1256335
[1] tausq = 0.6073302


> plot(robust_est, main="", cex.lab=1.6, cex.axis=1.5, cex.main=1.5 )
> lines(my_WLS_fit)
```

Figure 6: WLS estimator for Gaussian semi-variogram



4. Fit your 'winning' theoretical variogram type using maximum likelihohod (ML). State the resulting estimates for the theoretical variogram.

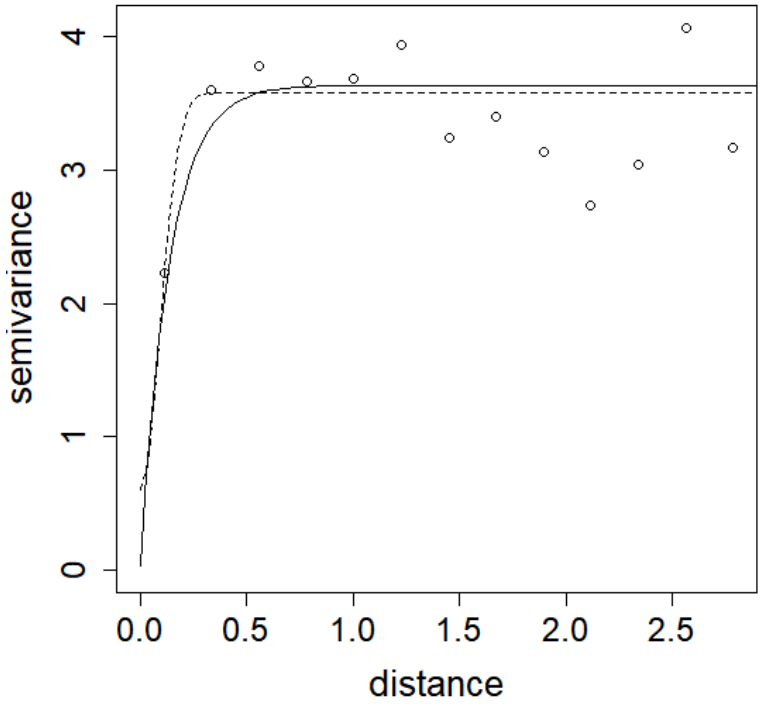
```
ini_sigsq <- my_WLS_fit$cov.pars[1]
ini_phi <- my_WLS_fit$cov.pars[2]
ini_tausq <- my_WLS_fit$nugget
my_ML_fit <- likfit( logCatch, trend="1st",
                     ini.cov.pars=c(ini_sigsq,ini_phi),
                     nugget=ini_tausq, lik.method="ML")
c( my_ML_fit$sigmasq, my_ML_fit$phi, my_ML_fit$tausq)
```

Solution:
Fitting the gaussian semi-variogram with the maximum likelihood algorithm we get the following,
Code:

```
%Using Initial Values from WLS Fit
ini_sigsq <- my_WLS_fit$cov.pars[1]
ini_phi <- my_WLS_fit$cov.pars[2]
ini_tausq <- my_WLS_fit$nugget

%Calling likfit() for the maximum liklehood estimtar
my_ML_fit <- likfit( geo.scallops , trend="2nd",
```

```
                    ini.cov.pars=c(ini_sigsq ,ini_phi),
                    nugget=ini_tausq , lik.method="ML")


%Printing the results
print(c(my_ML_fit$sigmasq , my_ML_fit$phi , my_ML_fit$tausq))
[1] sigamasq = 3.60472588 phi = 0.13487626 tausq = 0.03059721


%Plotting the results
plot(robust_est , main="", cex.lab=1.6, cex.axis=1.5, cex.main=1.5 )
lines(my_WLS_fit , lty = 2)
lines(my_ML_fit)
```

Figure 7: ML estimator for Gaussian semi-variogram (WLS is dashed)



5. Fit your winning theoretical variogram using REML. State the resulting estimates for the theoretical variogram. Finaly graph all three semivariograms (WLS, ML, REML) on a single plot.

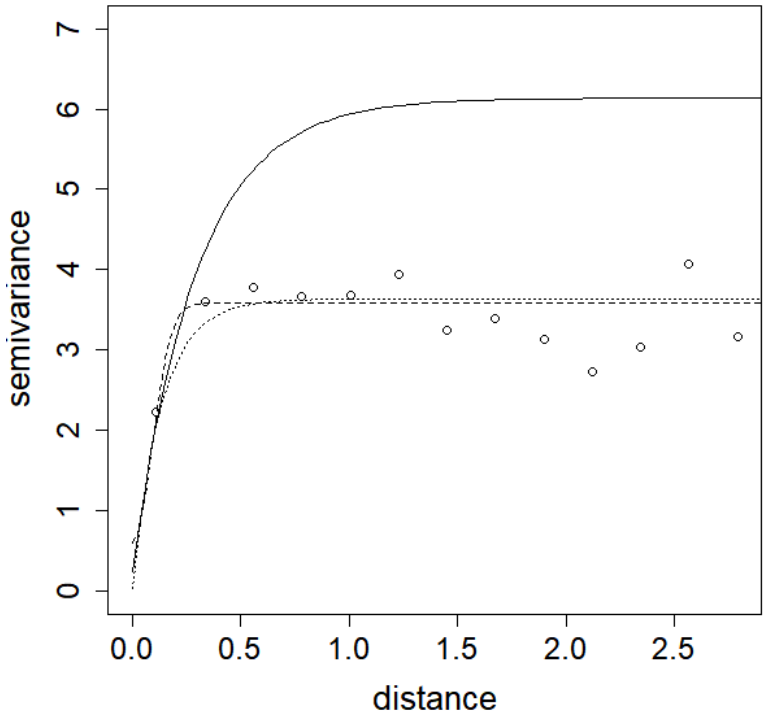
```
my_REML_fit <- likfit( logCatch, trend="2nd",
                       ini.cov.pars=c(ini_sigsq,ini_phi),
```

```
                                nugget=ini_tausq, lik.method="REML")
    c( my_REML_fit$sigmasq, my_REML_fit$phi, my_REML_fit$tausq)
```

Solution:
Fitting the gaussian semi-variogram with the residual maximum likelihood algorithm we get the following,
Code:

```
my_REML_fit <- likfit( geo.scallops , trend="2nd",
                       ini.cov.pars=c(ini_sigsq ,ini_phi),
                       nugget=ini_tausq , lik.method="REML")

print(c(my_REML_fit$sigmasq , my_REML_fit$phi , my_REML_fit$tausq))
[1] sigmasq = 5.8967643 phi = 0.2951893 tausq = 0.2380306

plot(robust_est , main="", cex.lab=1.6, cex.axis=1.5, cex.main=1.5,
ylim = c(0, 7))
lines(my_WLS_fit, lty = 2)
lines(my_ML_fit)
lines(my_REML_fit, lty = 3)
```

Figure 8: REML estimator for Gaussian semi-variogram (WLS is dashed)(ML is dotted)



Exercise 5: Simulating data, experimenting with estimating parameters for semivariograms.
I have posted R code on Canvas for this assignment. The code carries out the following steps:

- It simulates a spatial data set using the geoR function, `grf`.
- It calculates and plots an empirical estimate of the semivariogram using the robust estimator.
- It then carries out a randomization test to see whether a spatial model is necessary (which it should be, since we're using code to simulate spatial data!), and plots the results.
- It overlays a curve that is the true semivariogram (the one that's used to simulate the original data)
- It overlays a curve that is an estimate of the semivariogram, using the WLS method.

Your instructions are to run the code 8 times, which results in 8 plots. The first four times you run the code, use $n = 25$ observations; the second four times you run the code, use $n = 100$ observations.

1. What type of semivariogram is the "truth"? What are its parameters? Is there a nugget?

   Solution:
   The "truth" semi-variogram is defined in the first block of code,
   Code:

```
sigsq <- 1.5
phi <- 10.0

tmp <- grf(25, grid="irreg",
           xlims=c(70,110), ylims=c(20,50),
           cov.model = "exponential",
           cov.pars = c(sigsq,phi) )
mydata <- as.geodata( cbind( tmp$coords,tmp$data) )
```

   It describes the generation of random data with an exponential covariance function. This function has parameters $\sigma^2 = 1.5$, $\phi = 10$, and $\tau^2 = 0$(The nugget parameter is set by default. This can be found in the grf() documentation.)

2. Summarize your results. Include a table for the parameter estimates for the $n = 25$ simulations and another table for the $n = 100$ simulations.

Solution:
Here are the results of our simulations,

Figure 9: Simulation results for $n = 25$

Trial	σ^2	ϕ	τ^2
1	177.6862	3218.7565	0
2	0.7664049	2.868407	0
3	1.093023	6.151739	0
4	27.77649	435.54335	0

Figure 10: Simulation results for $n = 100$

Trial	σ^2	ϕ	τ^2
1	0.7267336	2.6374292	0
2	1.541939	8.777938	0
3	2.06919	12.03973	0
4	1.104447	4.464463	0

Since we know the true values of $\sigma^2 = 1.5$ and $\phi = 10$. It seems as though the first and fourth trials in the $n = 25$ simulation pulled a sample of observations which resulted in extremely large parameters in the WLS estimator. Generally it seems as though the WLS estimator for the $n = 100$ simulation performed better than the $n = 25$, this is as expected. When we look at the randomization tests, there is a lot less variation in the $n = 100$ simulation, it's easier to capture the signal of spatial correlation when we have more data.

Figure 11: Resultant Plots from $n = 25$. Trials are plotted left to right, Trial 1 is top left ...

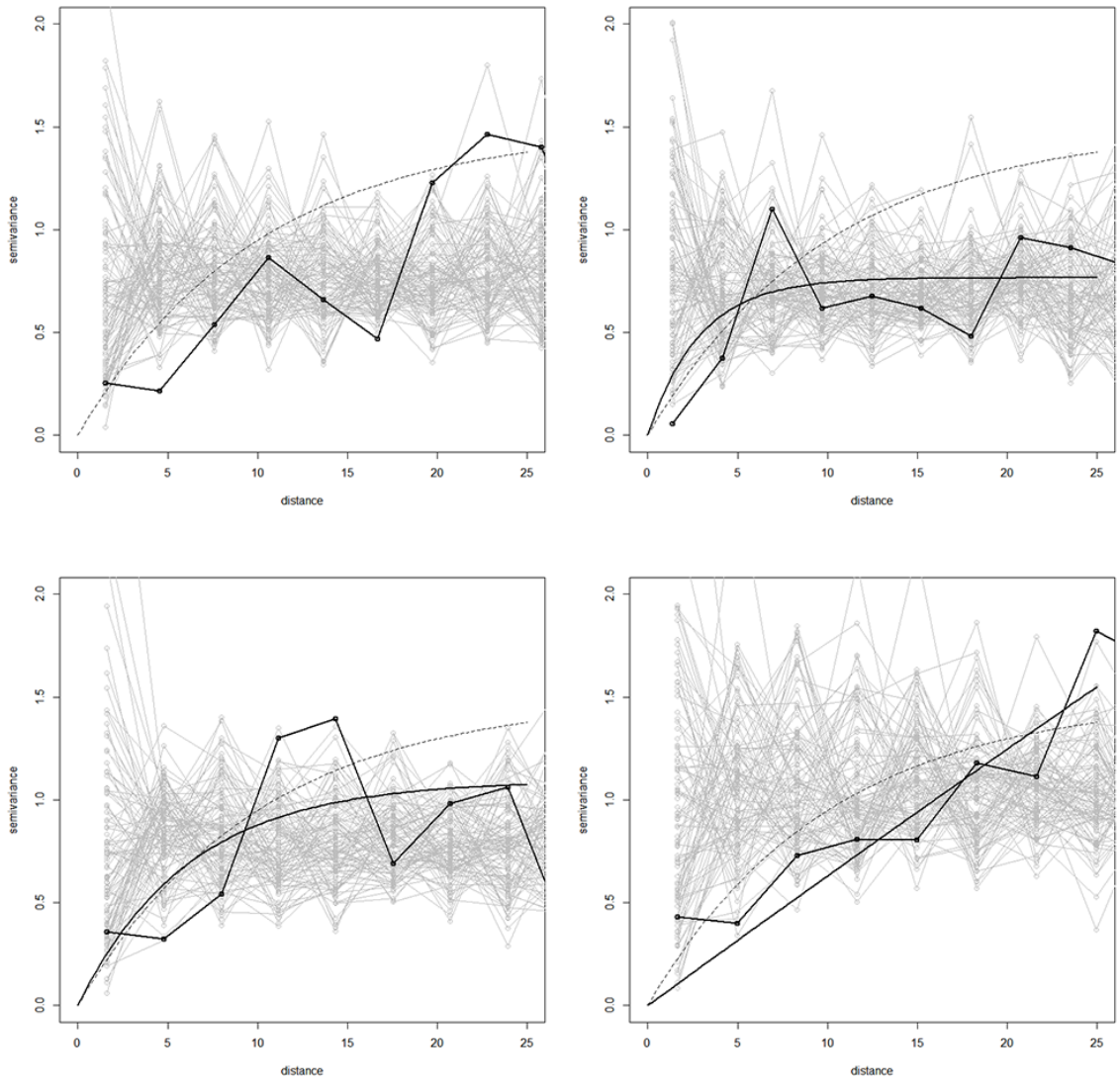
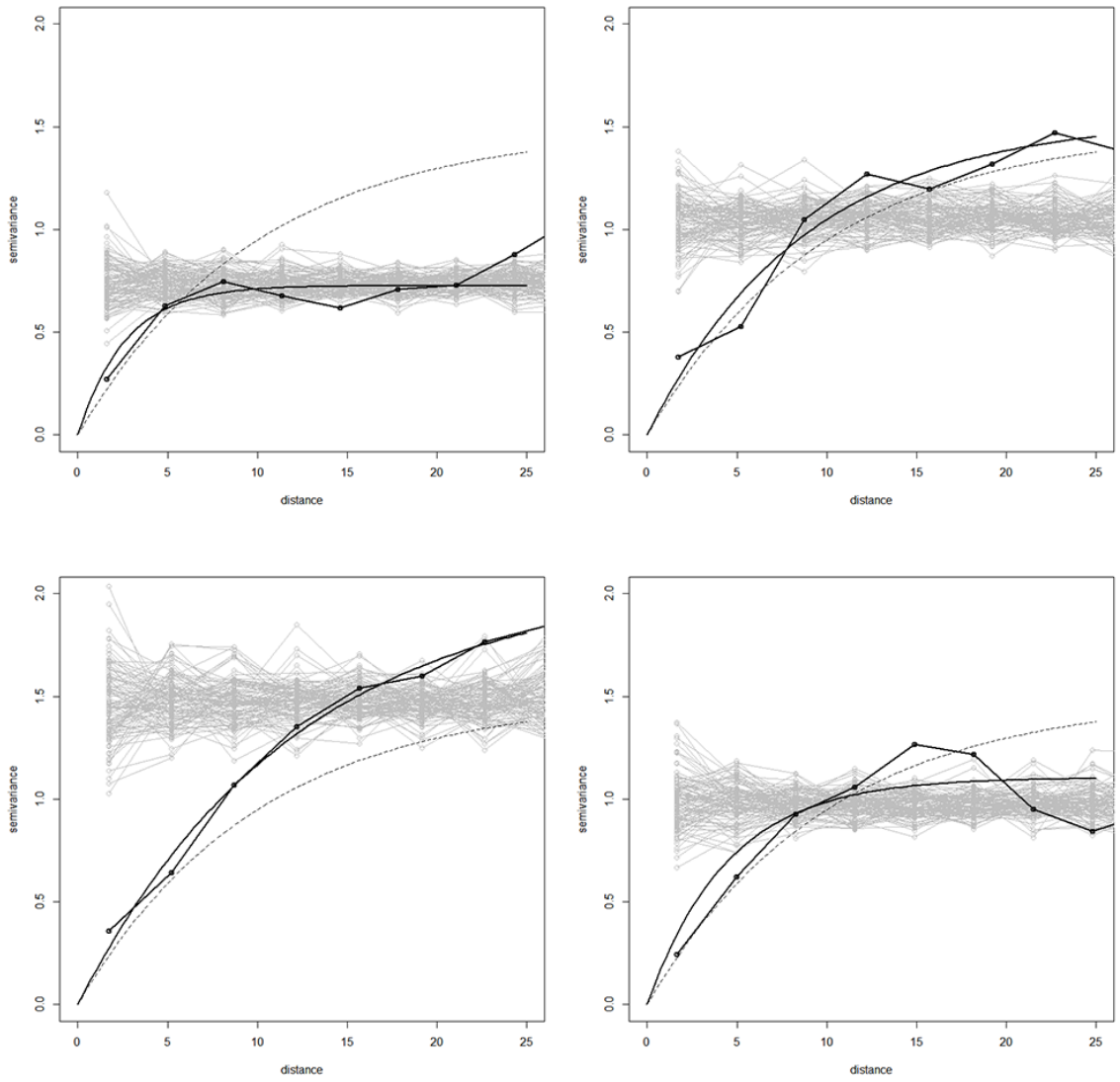


Figure 12: Resultant Plots from $n = 100$. Trials are plotted left to right, Trial 1 is top left ...



3. Do the empirical semivariograms appear to give a good match to the true semivariograms? Do the WLS estimated semiovariograms give a good match to the "truth"? Do you get appreciably better results with the larger sample size? (One hopes that bigger sample sizes yield better estimates. Does this appear to be the case?)

Solution:
Comparing the two simulations it seems that when we have a larger sample size the empirical semi-variogram will be closer to the WLS estimated semi-variogram, not necessarily close to the true semi-variogram. Trials 1 and 3 of the $n = 100$

simulation did a poor job of estimating the true semi-variogram even though we had more observations. I guess a possible conclusion is not that more observations is better, but having observations that capture the spatial signal is better. It seems like trials 1 and 3 of the $n = 100$, ended up with a sample which failed to capture the spatial signal as well as the other trials in the simulation.

4. Do the randomization tests appear to suggest that a spatial model is necessary in each case? Explain briefly.

Solution:
The randomization tests across both simulations suggest to me that a spatial model is necessary. Across all trials we see that the empirical semi-variogram exhibits a trend that goes across the random semi-variograms. In the $n = 100$ simulation this trend is clearer, since we have more observations there is less variance among the randomized semi-variograms.