Air Quality Index of Ozone

Stats C173 Applied Geostatistics Final Project

Huimin Zhang - March 12, 2021

Introduction

There are six common pollutants in the world, such as Ozone (O3), Particulate matter (PM10 and PM2.5), Carbon monoxide (CO), Nitrogen dioxide (NO2), Sulfur dioxide (SO2), and Lead (Pb). In this project, I mainly study the ozone.

Ozone can be “good” or “bad” for people. “Good” ozone is also called stratospheric ozone, which protects us from the sun’s harmful radiation. “Bad” ozone is also called ground-level ozone, it has health effects on people, particularly for children, the elderly, and people who have lung disease. It also has environmental effects on sensitive vegetation and ecosystems.

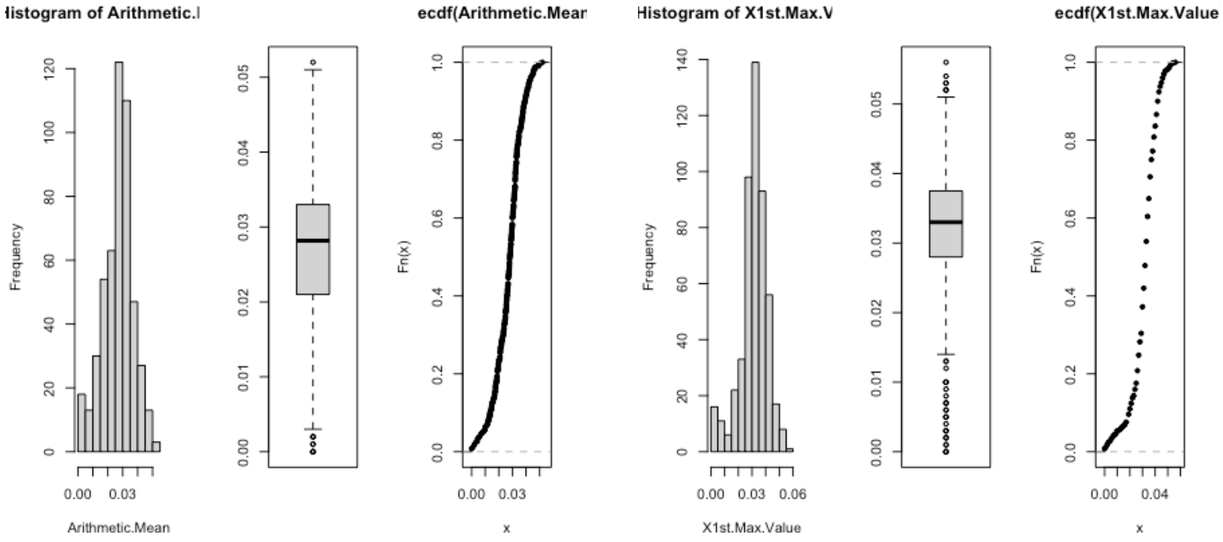
AQI is an index of overall daily air quality. There are six levels of health concern connect with AQI values. For example, "Good" AQI is 0 - 50. "Moderate" AQI is 51 - 100. "Unhealthy for Sensitive Groups" AQI is 101 - 150. "Unhealthy" AQI is 151 - 200. "Very Unhealthy" AQI is 201 - 300. "Hazardous" AQI is greater than 300.

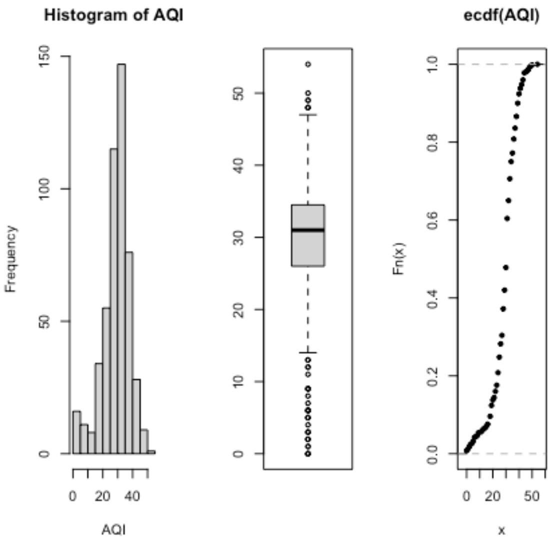
Data Discussion

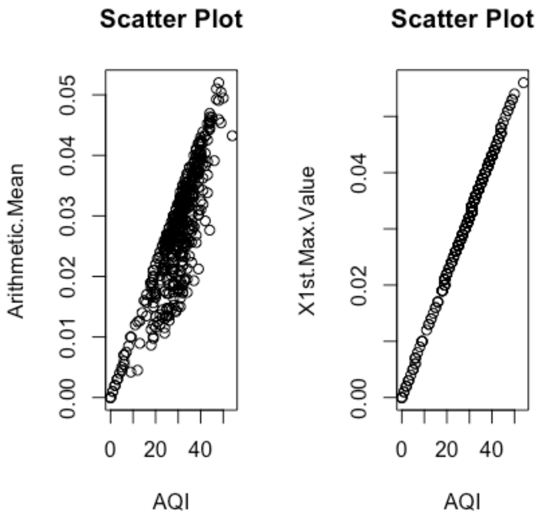
I downloaded the 2020 ozone daily summary dataset from United States Environmental Protection Agency. The original dataset has 247653 observations and 29 variables, from date 01/01/2020 to 11/04/2020 in 274 sites. After I removed the observations with duplicate x, y coordinates, there were 1211 observations left.

I randomly selected 500 observation with two coordinates variables “Longitude” and “Latitude”, one target variable “AQI”, and two other variables “Arithmetic.Mean” which is the mean of ozone value and “X1st.Max.Value” which is the first observed maximum value of ozone. “Arithmetic.Mean” has a high correlation of 0.88 with “AQI”, and “X1st.Max.Value” also has a high correlation of 0.99 with “AQI”.

Non spatial exploratory analysis



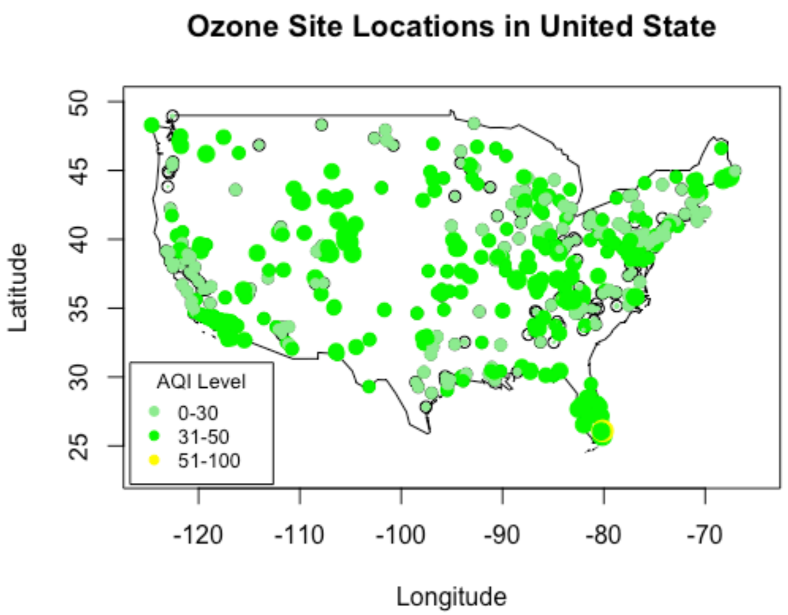




From the above figures, we can see that the distribution of "Arithmetic.Mean", "X1st.Max.Value", and "AQI" are approximately normal, therefore, we do not need to transfer the data.

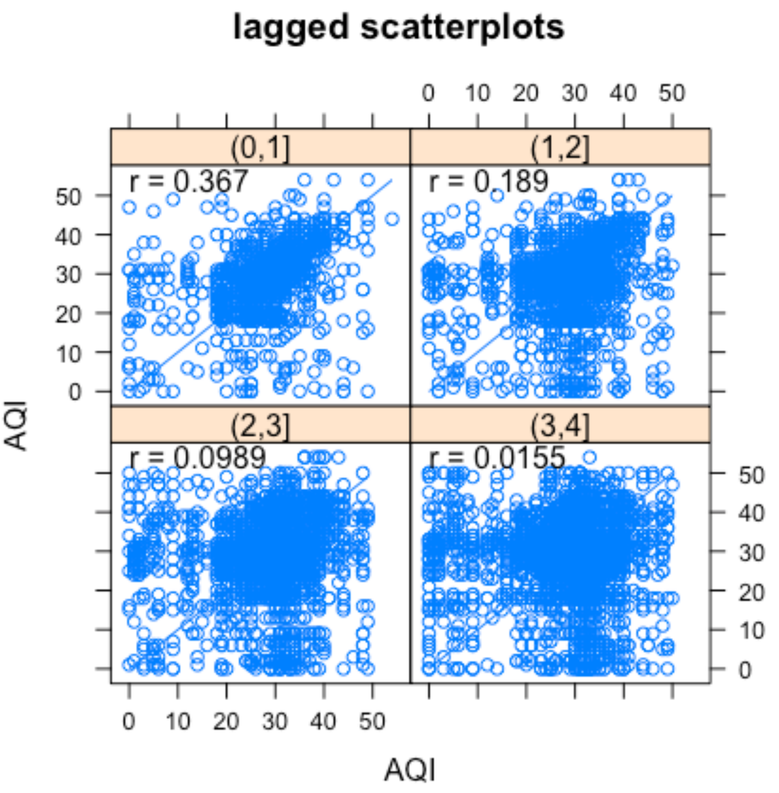
Also, we can see that Arithmetic.Mean" and "X1st.Max.Value" has a high correlation with "AQI" since the scatter plots show that points are in one line.

Spatial analysis



Location Plot

499 observations have an AQI level is under 50, which means the air quality is satisfactory. Only 1 observation has an AQI level of 54, which means the air quality is acceptable but maybe a risk for some people. From the plot, we can see that this location is a yellow point in Florida.

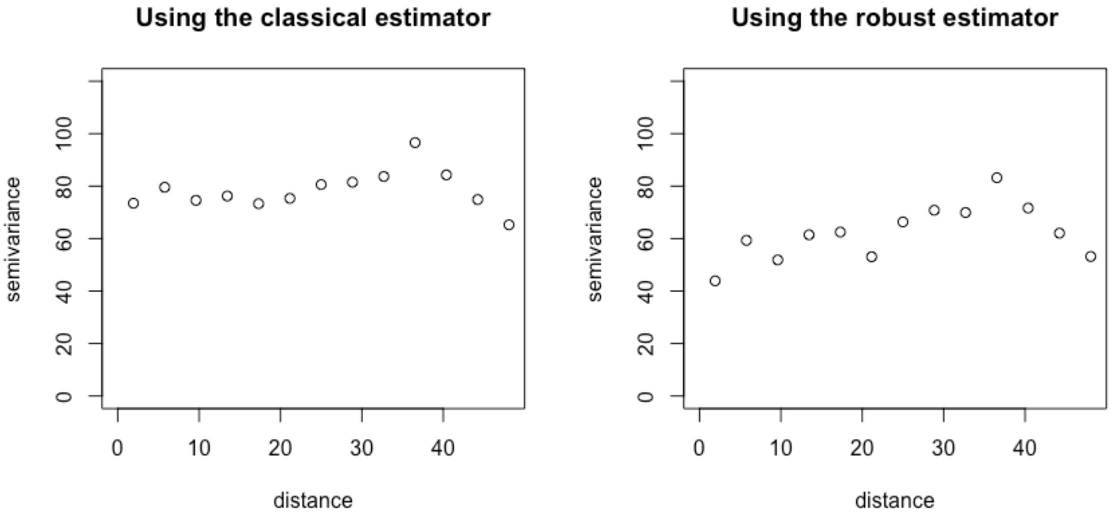


h-scatterplots

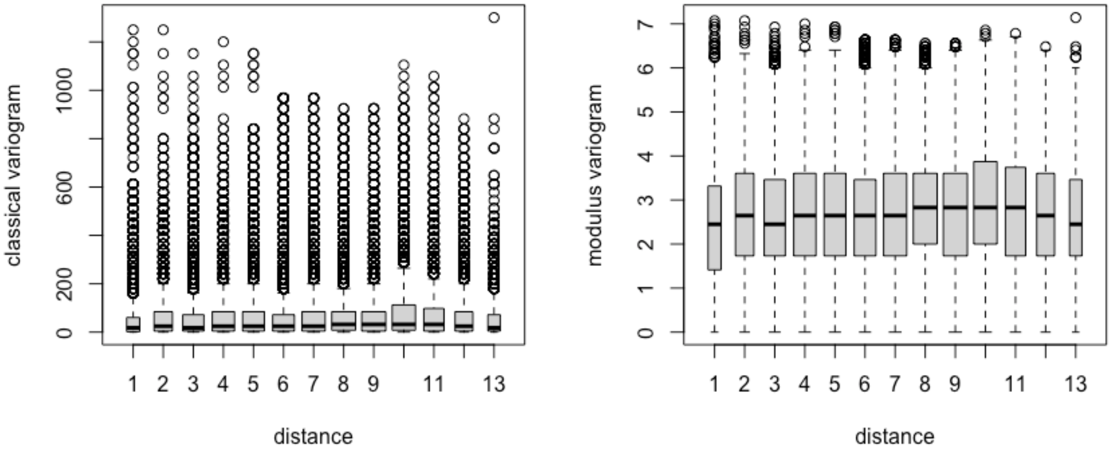
These scatterplots show a cloud of points distributed around the 45-degree line. If we increase the distance from 0 m to 1 m, 2 m, 3 m, and 4 m, we see that the cloud becomes “fatter” indicating that the values separated by a longer distance are not as close as with the 1 m case.

The spatial correlation between points within 1 m is 0.367. The spatial correlations decrease as the separation distances increase.

Semivariogram plots of the classical and robust estimators

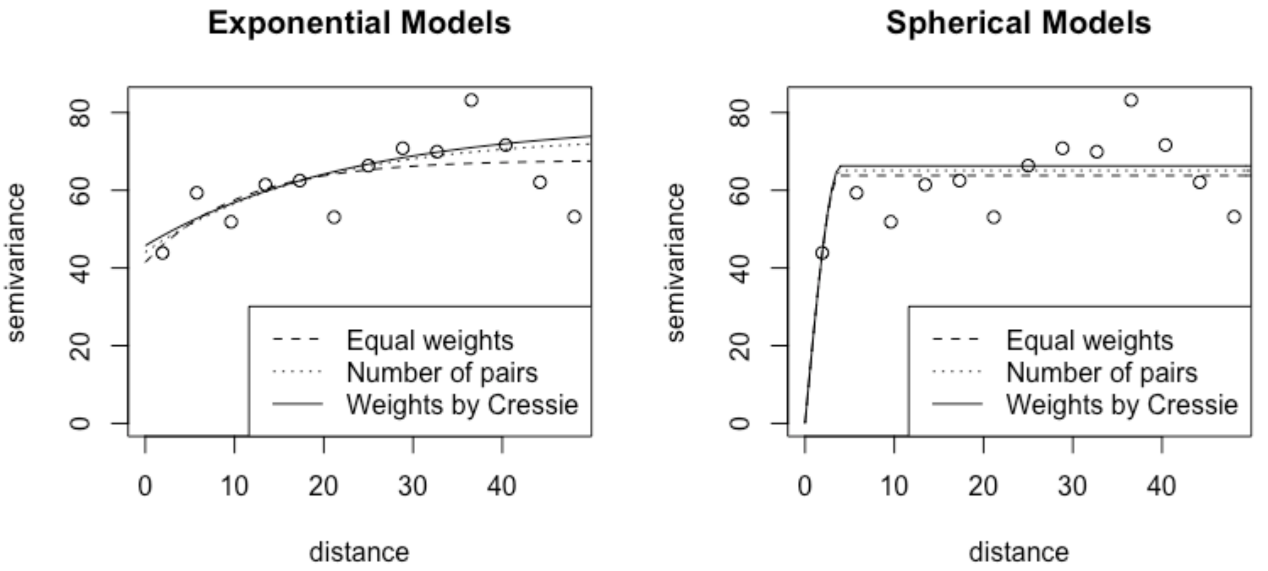


Box plots using the variogram cloud of the classical and robust estimators

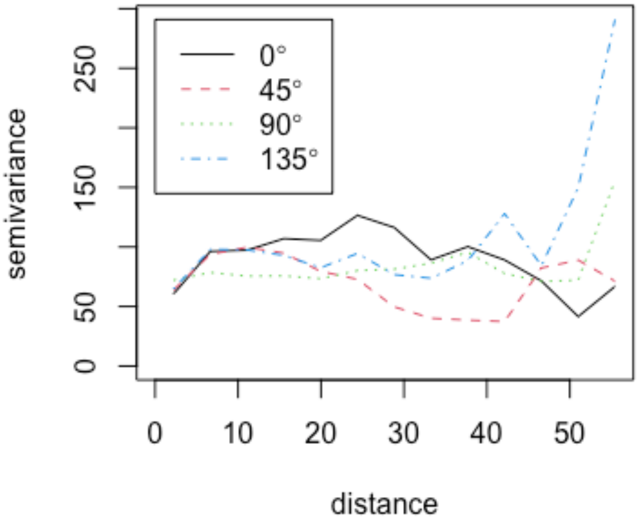


From the above plots, we can see that using robust estimators can reduce the effect of outliers.

Fitting Variogram Models using the geoR package

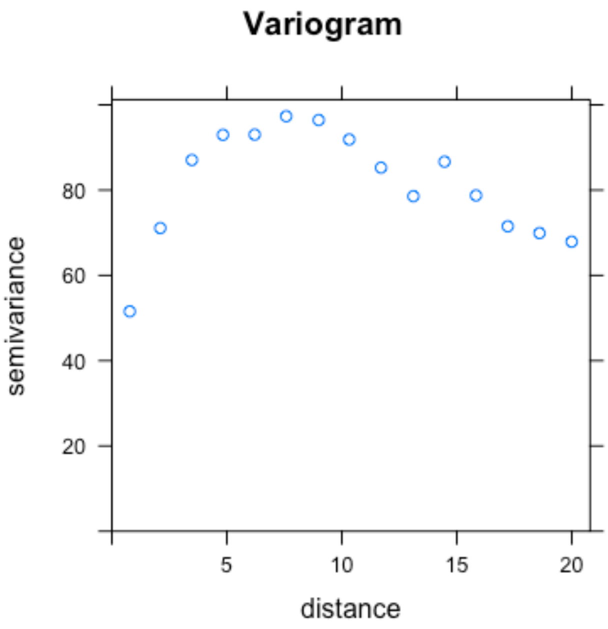


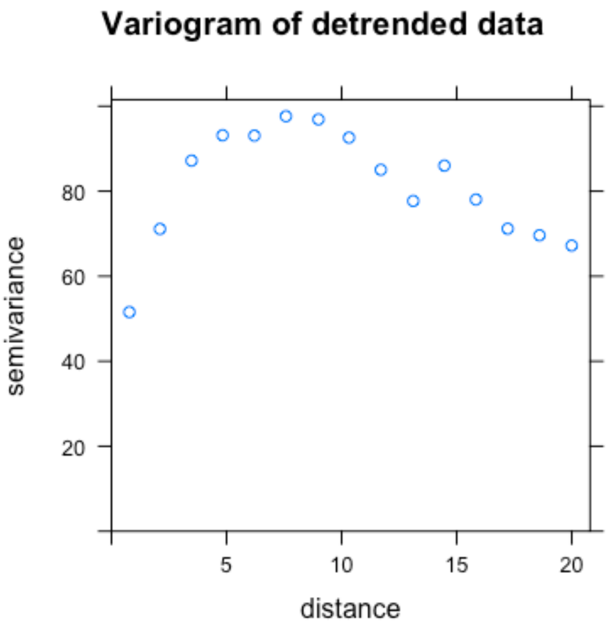
I fitted the exponential model and spherical model to the variogram using equal weights, number of pairs, and weights by Cressie to compare the difference. I think the exponential model fits better.



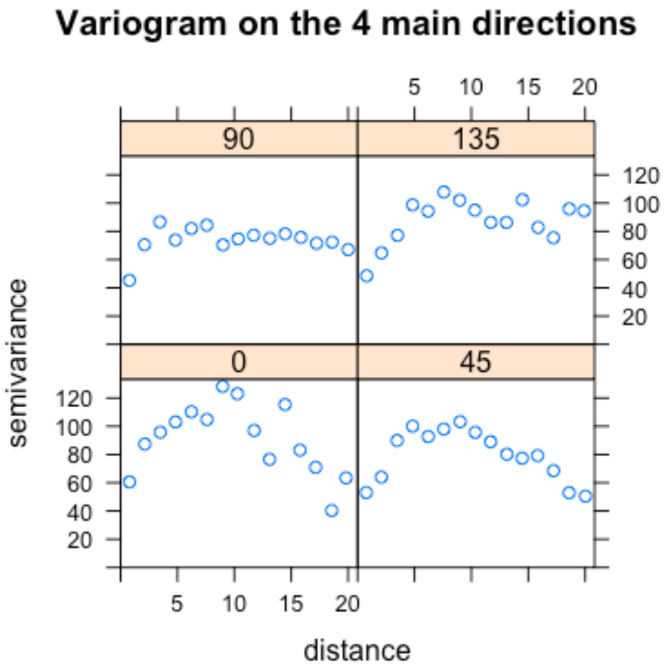
I computed the variogram using direction = pi/2, and max distance = 50 base on the 4 main direction of variogram.

Also, I used robust estimator to reduce the effect of outliers.

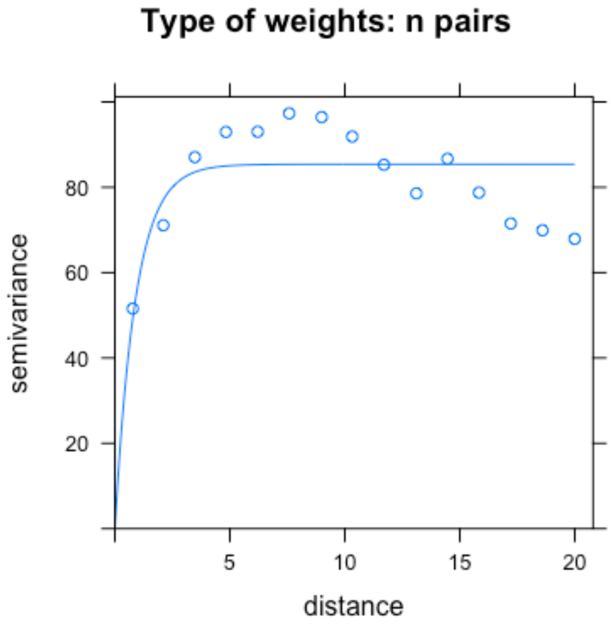




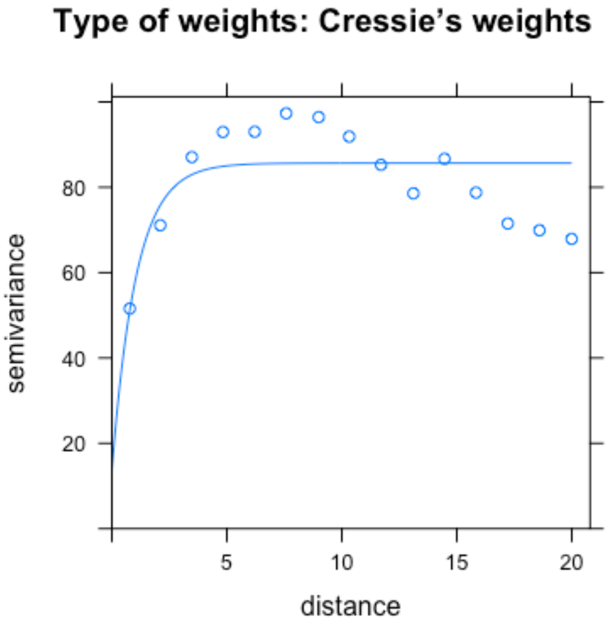
We can see that after we remove the trend in the data, the variogram does not change significant. Therefore, we can assume that there is no trend in the data.



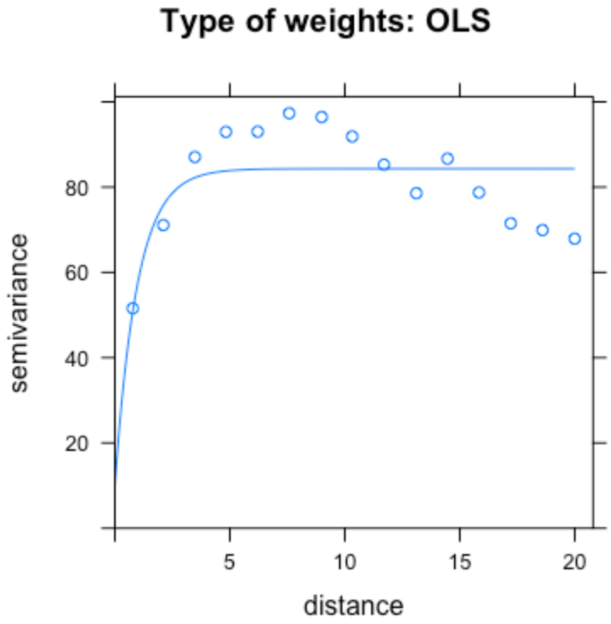
Base on the 4 main direction of variogram, we can see that variogram on 90 degree direction looks better than the others.



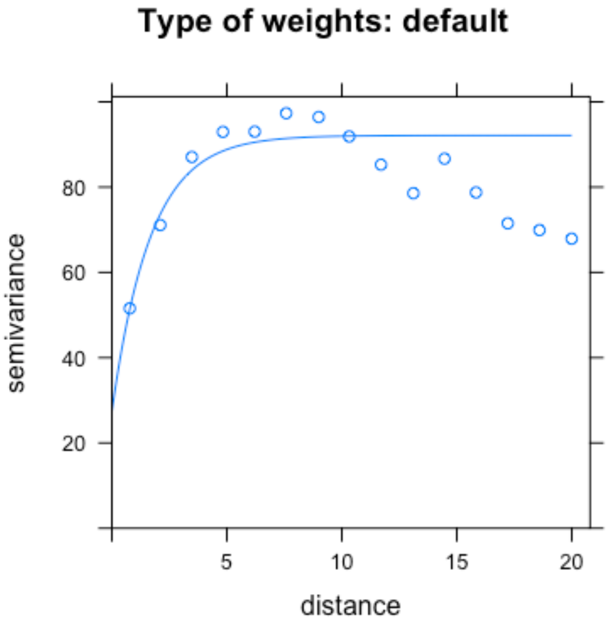
Type of weights: n pairs



Type of weights: Cressie’s weights



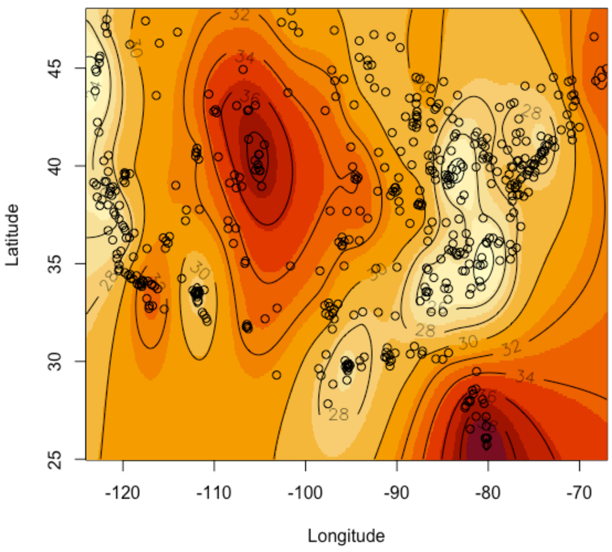
Type of weights: OLS

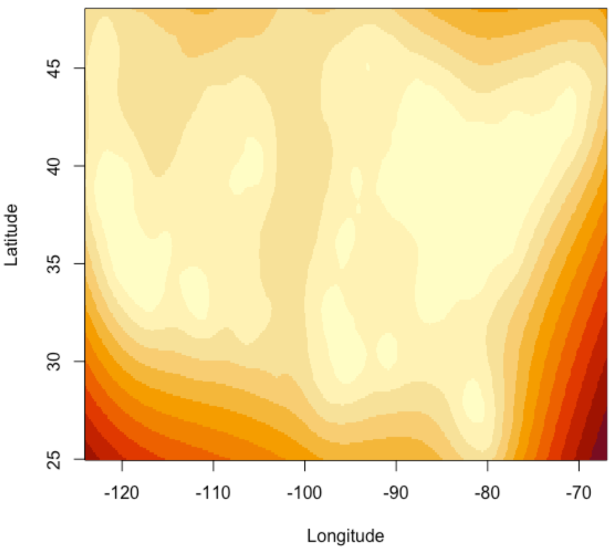


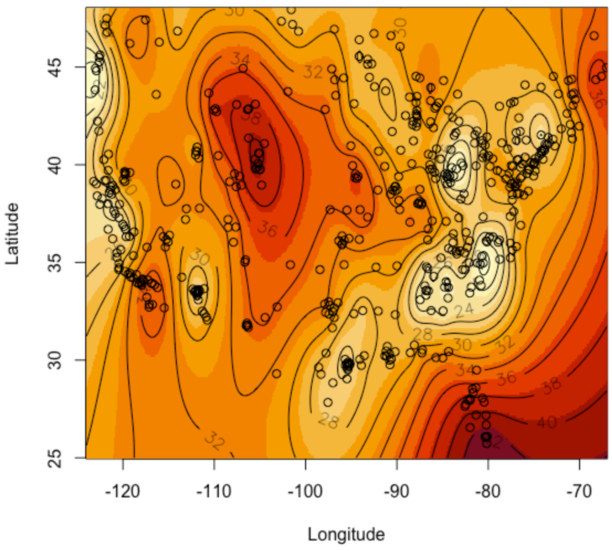
Type of weights: default

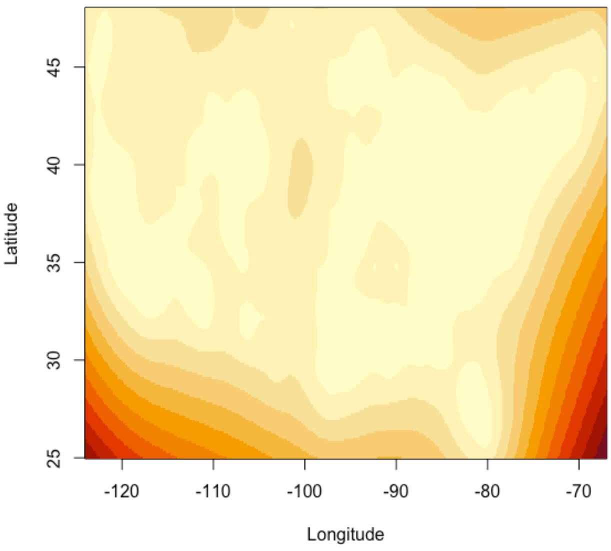
I fitted the exponential model to the variograms with partial sill = 20, range = 2, and nugget = 0. By comparing different weights of n pairs, Cressie’s, OLS, and default, the default weights fits better.

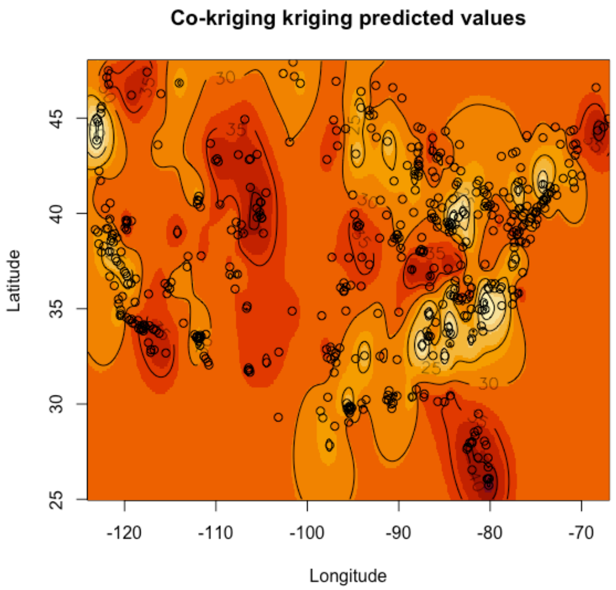
Spatial prediction using the kriging method

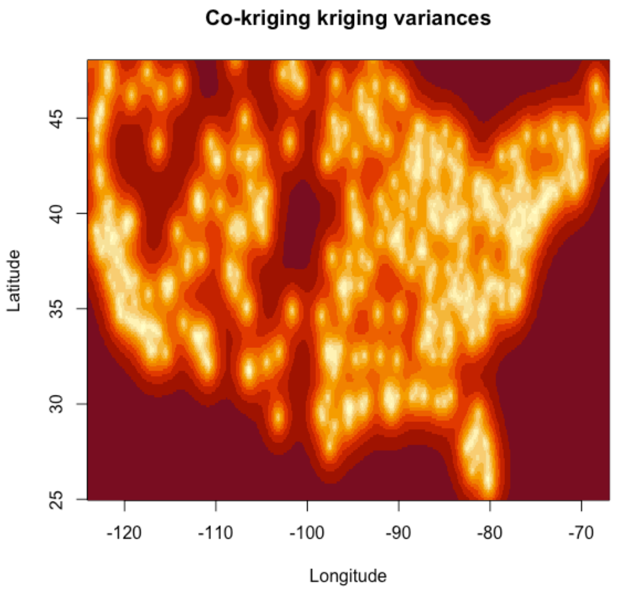

Ordinary kriging predicted values


Ordinary kriging variances


Universal kriging predicted values

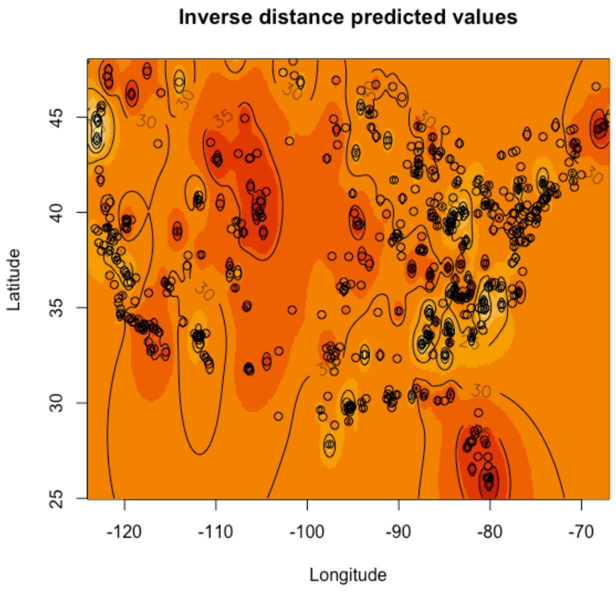

Universal kriging variances





I chose an exponential model variogram with partial sill = 20, range = 2, and nugget = 0 to predicted by using ordinary kriging, universal kriging, and co-kriging. Above are the raster maps of 3 different kriging method predicted values and variances.

Spatial prediction using the inverse distance method



There is no raster map of variances because inverse distance weighted method does not provide the variance of the predicted values.

Cross-validation using the geoR package

Cross-validation: omits one point at a time	
Model	**PRESS**
Fit1: Exponential model with equal weights	61.84
Fit2: Exponential model with n pair weights	63.52
Fit3: Exponential model with Cressie’s weights	64.32

Cross-validation: re-estimated each time a data point is omitted	
Model	**PRESS**
Fit1: Exponential model with equal weights	62.53

Cross-validation using the gstat package

Ordinary kriging: Cross-validation	
Model	**PRESS**
Fit1: Exponential model with n pairs weights	69.24
Fit2: Exponential model with Cressie’s weights	62.42
Fit6: Exponential model with OLS weights	63.47
Fit7: Exponential model with default weights	59.83

Universal kriging: Cross-validation	
Model	**PRESS**
Fit0: Exponential model with default weights	59.91

Co-kriging: Cross-validation	
Model	**PRESS**
Exponential model using “Arithmetic.Mean” and “X1st.Max.Value” as co-located variables	0.097

By using cross-validation, we can decide which prediction method gives better results.

From cross-validation using the geoR package, we can see that the exponential model with equal weights gives us the smallest PRESS 61.84. I used this model to perform cross-validation that re-estimated each time a data point is omitted, it took more time to run and the result was similar to the cross-validation which omits one point at a time.

From cross-validation using the gstat package, we can see that co-kriging gives a much smaller PRESS than ordinary kriging and universal kriging. I chose the best model from ordinary kriging to perform the universal kriging to compare the difference, it turned out that the PRESS of these two methods are similar. Ordinary kriging was slightly better than universal kriging in this case.

Conclusion

Co-kriging used two co-located variables “Arithmetic.Mean” and “X1st.Max.Value” could improve our predicting power to predict “AQI” a lot since these two co-located variables have a high correlation with “AQI”.

Summary of co-kriging predicted AQI values

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
8.55	29.48	30.72	30.67	32.34	45.56

Overall, the predicted AQI values of the 131901 grid points are under 50, which means the air quality is good and the ozone level is low in the United States.

In order to reduce air pollution and maintain a lower AQI, we can choose cleaner transportation, use environmentally safe paints and cleaning products whenever possible, etc.

Appendix: R codes

```
set.seed(173)
library(tidyverse)
oz_og <- read.csv(“daily_44201_2020.csv")

# Remove duplicated x, y coordinates.
oz_rm <- oz_og[-which(duplicated(oz_og$Longitude)), ]

# Select x,y coordinates, target variable, and other variables.
oz_sl <- oz_rm[sample(1:nrow(oz_rm), 500),]
oz <- oz_sl %>%  select(Longitude, Latitude, AQI, Arithmetic.Mean, X1st.Max.Value)

attach(oz)

#################################################
#################################################

# Non spatial exploratory analysis
hist(AQI); boxplot(AQI); plot(ecdf(AQI))
hist(X1st.Max.Value); boxplot(X1st.Max.Value); plot(ecdf(X1st.Max.Value))
hist(Arithmetic.Mean); boxplot(Arithmetic.Mean); plot(ecdf(Arithmetic.Mean))

plot(AQI, Arithmetic.Mean, main = "Scatter Plot")
plot(AQI, X1st.Max.Value, main = "Scatter Plot")

#################################################
#################################################

#Create the grid:
x.range <- as.integer(range(oz[,1]))
y.range <- as.integer(range(oz[,2]))
grd <- expand.grid(x=seq(from=x.range[1], to=x.range[2], by=0.1),
               y=seq(from=y.range[1], to=y.range[2], by=0.1))

#################################################
#################################################

# spatial analysis using gstat
```

```
library(gstat)
library(sp)

##################################################
# location plot
plot(Longitude, Latitude, xlim = c(-125, -65), ylim = c(23, 50), xlab  ="Longitude",
   ylab = "Latitude", main="Ozone Site Locations in United State")
map("world", "us", add = TRUE)
AQI_colors <- c("lightgreen", "green", "yellow")
AQI_levels <- cut(AQI, c(min(AQI),30, 50, max(AQI)))
points(Longitude, Latitude, cex = AQI/mean(AQI), pch = 19,
    col = AQI_colors[as.numeric(AQI_levels)])
legend("bottomleft", inset=.01, title="AQI Level", pch=19, col=AQI_colors, cex=0.8,
    c(“0-30","31-50","51-100"))

##################################################
# h-scatterplots
qq <- hscat(AQI ~ Longitude + Latitude, locations = ~ Longitude + Latitude,
       oz, c(seq(0, 4, 1)))
plot(qq, main = “h-scatterplots")

##################################################
# Compute and plot the semivariogram:
coordinates(oz) <- ~Longitude+Latitude
coordinates(grd) <- ~ x+y

g <- gstat(id = "AQI", formula = AQI~1, data = oz)
q <- variogram(g); plot(q, main = "Variogram")

# Remove the trend
g0 <- gstat(id = "AQI", formula = AQI~Longitude+Latitude, data = oz)
q0 <- variogram(g0); plot(q0, main = "Variogram of detrended data")

v.fit0 <- fit.variogram(q0, vgm(20,"Exp",2,0), fit.method=7)
plot(q0, v.fit0)

v <- variogram(g, alpha = c(0,45,90,135))
plot(v, main = "Variogram on the 4 main directions")

# Variogram modeling:
```

```
v.fit1 <- fit.variogram(q, vgm(20,"Exp",2,0), fit.method=1)
v.fit2 <- fit.variogram(q, vgm(20,"Exp",2,0), fit.method=2)
v.fit6 <- fit.variogram(q, vgm(20,"Exp",2,0), fit.method=6)
v.fit7 <- fit.variogram(q, vgm(20,"Exp",2,0), fit.method=7)

plot(q, v.fit1, main = "Type of weights: n pairs")
plot(q, v.fit2, main = "Type of weights: Cressie’s weights")
plot(q, v.fit6, main = "Type of weights: OLS")
plot(q, v.fit7, main = "Type of weights: default")

##################################################
# Perform inverse distance predictions:
idw.out <- idw(AQI~1, oz, grd)

#Collapse the predicted values into a matrix:
qqq <- matrix(idw.out$var1.pred, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
        length(seq(from=y.range[1], to=y.range[2], by=0.1)))

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1], to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude",  main = "Inverse distance predicted values")

#Add the data points:
points(oz)

#Add contours:
contour(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, add=TRUE, col="black", labcex=1)

##################################################
# Perform ordinary kriging predictions:
pr_ok <- krige(id = "AQI", formula = AQI~1, oz, newdata = grd, model = v.fit7)

#Collapse the vector of the predicted values into a matrix:
qqq <- matrix(pr_ok$AQI.pred, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
        length(seq(from=y.range[1], to=y.range[2], by=0.1)) )

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Ordinary kriging Predicted values")
```

```
#Add the data points:
points(oz)

#Add contours:
contour(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, add=TRUE, col="black", labcex=1)

#Collapse the vector of the variances into a matrix:
qqq <- matrix(pr_ok$AQI.var, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
        length(seq(from=y.range[1], to=y.range[2], by=0.1)) )

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Ordinary kriging variances")

#################################################
# Perform co-kriging predictions:
#Begin with the target variable AQI:
g1 <- gstat(id = "AQI", formula = AQI~1, data = oz)
#Append mean:
g1 <- gstat(g1,id="Mean Value", formula = Arithmetic.Mean~1, data = oz)
#Append max:
g1 <- gstat(g1,id="Max Value", formula = X1st.Max.Value~1, data = oz)

#Fit a model variogram to all the variograms:
vm <- variogram(g1); vm.fit <- fit.lmc(vm, g1, model=v.fit7)

#Perform co-kriging predictions:
ck <- predict(vm.fit, grd)

#Collapse the vector of the predicted values into a matrix:
qqq <- matrix(ck$AQI.pred, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
        length(seq(from=y.range[1], to=y.range[2], by=0.1)))

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1], to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Co-kriging kriging predicted
values")
```

```
#Add the data points:
points(oz)

#Add contours:
contour(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1], to=y.range[2],
by=0.1), qqq, add=TRUE, col="black", labcex=1)

#Collapse the vector of the variances into a matrix:
qqq <- matrix(ck$AQI.var, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
          length(seq(from=y.range[1], to=y.range[2], by=0.1)) )

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Co-kriging kriging variances")

################################################
# Perform cross-validation:
#Ordinary kriging:
cv_pr1 <- krige.cv(formula = AQI~1, oz, model = v.fit1, nfold=nrow(oz))
#Compute the prediction sum of squares:
cv_PRESS1 <- sum(cv_pr1$residual^2) / nrow(oz); cv_PRESS1 # 69.24327

cv_pr2 <- krige.cv(formula = AQI~1, oz, model = v.fit2, nfold=nrow(oz))
#Compute the prediction sum of squares:
cv_PRESS2 <- sum(cv_pr2$residual^2) / nrow(oz); cv_PRESS2 # 62.42498

cv_pr6 <- krige.cv(formula = AQI~1, oz, model = v.fit6, nfold=nrow(oz))
#Compute the prediction sum of squares:
cv_PRESS6 <- sum(cv_pr6$residual^2) / nrow(oz); cv_PRESS6 # 63.46811

cv_pr7 <- krige.cv(formula = AQI~1, oz, model = v.fit7, nfold=nrow(oz))
#Compute the prediction sum of squares:
cv_PRESS7 <- sum(cv_pr7$residual^2) / nrow(oz); cv_PRESS7 # 59.83212

####################
#Universal kriging:
cv_pr0 <- krige.cv(formula = AQI~Longitude+Latitude, oz, model = v.fit0, nfold=nrow(oz))
#Compute the prediction sum of squares:
cv_PRESS0 <- sum(cv_pr0$residual^2) / nrow(oz); cv_PRESS0 # 59.90906
```

```
####################
#Co-kriging:
#cv_ck <- gstat.cv(vm.fit)
#Compute the prediction sum of squares:
cv_PRESS <- sum(cv_ck$residual^2) / nrow(oz); cv_PRESS # 0.09700724

#################################################
#################################################

# spatial analysis using geoR

library(geoR)

b <-as.geodata(oz)
summary(b)
plot(b)

# Compute and plot the semivariogram:
# Using the classical estimator
variogram_classical <- variog(b, dir=pi/2, max.dist=50)
plot(variogram_classical, main="Using the classical estimator", ylim = c(0, 120))

# Using the robust estimator
variogram_robust <- variog(b, dir=pi/2, max.dist=50, estimator.type="modulus")
plot(variogram_robust, main="Using the robust estimator", ylim = c(0, 120))

# Compute the semivariogram cloud for both estimators and construct the box plot
# Using the classical estimator
variogram_classical_cloud <- variog(b, dir=pi/2, max.dist=50, option="cloud")

# Using the robust estimator
variogram_robust_cloud <- variog(b, dir=pi/2, max.dist=50, option="cloud",
estimator.type="modulus")

# box plot
classical_cloud <- variog(b, dir=pi/2, max.dist=50, bin.cloud=T)
robust_cloud <- variog(b, dir=pi/2, max.dist=50, bin.cloud=T, estimator.type="modulus")
plot(classical_cloud, bin.cloud=T) ; plot(robust_cloud, bin.cloud=T)

#################################################
```

```
# Variogram modeling:
var4 <- variog4(b)
plot(var4, main = "Variogram on the 4 main directions")
var2 <- variog(b, direction=pi/2, max.dist=50, estimator.type="modulus")
plot(var2)

initial.values <- expand.grid(seq(50, 100, by=1),seq(0, 10, by=0.1))

fit1 <- variofit(var2, cov.model="exp", ini.cov.pars=initial.values,
          fix.nugget=FALSE, nugget=0, wei="equal")

fit2 <- variofit(var2, cov.model="exp", ini.cov.pars=initial.values,
          fix.nugget=FALSE, nugget=0, wei="npairs")

fit3 <- variofit(var2, cov.model="exp", ini.cov.pars=initial.values,
          fix.nugget=FALSE, nugget=0, wei="cressie")

plot(var2, main = "Exponential Models")
lines(fit1, type="l", lty=2) ; lines(fit2, type="l", lty=3) ; lines(fit3, type="l")
legend("bottomright", legend=c("Equal weights", "Number of pairs", "Weights by Cressie"),
lty=c(2,3,1))

##################################################

# Perform ordinary kriging:
qq <- krige.conv(b, locations=grd, krige=krige.control(obj.model=fit3))

#Collapse the vector of the predicted values into a matrix:
qqq <- matrix(qq$predict, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
        length(seq(from=y.range[1], to=y.range[2], by=0.1)) )

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Ordinary kriging predicted values")

#Add the data points:
points(oz)

#Add contours:
```

```
contour(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, add=TRUE, col="black", labcex=1)


#Collapse the vector of the variances into a matrix:
qqq <- matrix(qq$krige.var, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
          length(seq(from=y.range[1], to=y.range[2], by=0.1)) )

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Ordinary kriging variances")

#################################################

#Perform universal kriging:
krig_trend <- ksline(b, cov.model="exp", cov.pars=c(40, 20), nugget=20, m0="kt", trend=1,
locations=grd)

#Perform universal kriging (geoR using the function krige.conv):
#kc <- krige.conv(b, locations=grd,krige=krige.control(type.krige="ok", cov.model="sph",
#cov.pars=c(40, 10), nugget=20, trend.l="1st", trend.d="1st"))

#Collapse the vector of the predicted values into a matrix:
qqq <- matrix(krig_trend$predict, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
          length(seq(from=y.range[1], to=y.range[2], by=0.1)) )

#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Universal kriging predicted values")

#Add the data points:
points(oz)

#Add contours:
contour(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, add=TRUE, col="black", labcex=1)

#Collapse the vector of the variances into a matrix:
qqq <- matrix(krig_trend$krige.var, length(seq(from=x.range[1], to=x.range[2], by=0.1)),
          length(seq(from=y.range[1], to=y.range[2], by=0.1)) )
```

```
#Use the image to create a raster map:
image(seq(from=x.range[1], to=x.range[2], by=0.1), seq(from=y.range[1],to=y.range[2],
by=0.1), qqq, xlab="Longitude", ylab="Latitude", main="Universal kriging variances")

#################################################
#Perform cross validation:
x_val1 <- xvalid(b, model=fit1)
#Compute the prediction sum of squares:
dif1 <- oz$AQI - x_val1$predicted
PRESS1 <- sum(dif1^2) / nrow(oz); PRESS1 # 61.84378

#Perform cross validation:
x_val2 <- xvalid(b, model=fit2)
#Compute the prediction sum of squares:
dif2 <- oz$AQI - x_val2$predicted
PRESS2 <- sum(dif2^2) / nrow(oz); PRESS2 # 63.52127

#Perform cross validation:
x_val3 <- xvalid(b, model=fit3)
#Compute the prediction sum of squares:
dif3 <- oz$AQI - x_val3$predicted
PRESS3 <- sum(dif3^2) / nrow(oz); PRESS3 # 64.32154

#Or re-estimating the variogram after we omit each data point:
x_val0 <- xvalid(b, model=fit1, reest=TRUE, variog.obj=var2)
#Compute the prediction sum of squares:
dif0 <- oz$AQI - x_val0$predicted
PRESS0 <- sum(dif0^2) / nrow(oz); PRESS0 # 62.53157
```