Project 3

Blake Pappas

5 May 2022

Problem 1

The ERA-Interim is a global atmospheric reanalysis dataset. Reanalysis is an approach to produce spatially and temporally gridded datasets via data assimilation for climate monitoring and analysis. The R data file `NA_MaxT_ERA.RData` provides daily values of maximum temperature from 1979 to 2017 with the spatial resolution around 80km. The original global extent has been subsetted to a region covering the contiguous United States and extending into Canada and Mexico. In this problem, we would like to analyze an ERA-Interim annual average temperature time series at a spatial location of your choice. For example you can choose the spatial location closest to where you live now. Below you can find a script to get the spatial location closest to Clemson, SC:

1. First, use the following R script to load the R data object:

```
load("NA_MaxT_ERA.RData")

NA_MaxT <- NA_MaxT - 273.15 # Change from Kelvin to Celsius

# Numbers of Latitude and Longitude Grid Points
nlon <- length(NA_lon); nlat <- length(NA_lat)

nmon <- 12; nyr <- 39 # Numbers of months/years
```

```
library(fields)
```

```
## Loading required package: spam

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis
## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
Clemson.lon.lat <- c(-82.8374, 34.6834)

# You Could Get the Latitude and Longitude Information from Google
library(fields)
dist2Clemson <- rdist.earth(matrix(Clemson.lon.lat, 1, 2),
                            expand.grid(NA_lon, NA_lat), miles = F)
id <- which.min(dist2Clemson)
(lon_id <- id %% nlon)
```

```
## [1] 58
```

```
(lat_id <- id %/% nlon + 1)
```

```
## [1] 15
```

```
# Check
(rdist.earth(matrix(Clemson.lon.lat, 1, 2),
             matrix(c(NA_lon[lon_id], NA_lat[lat_id]), 1, 2), miles = F)
  == min(dist2Clemson))
```

```
##      [,1]
## [1,] TRUE
```

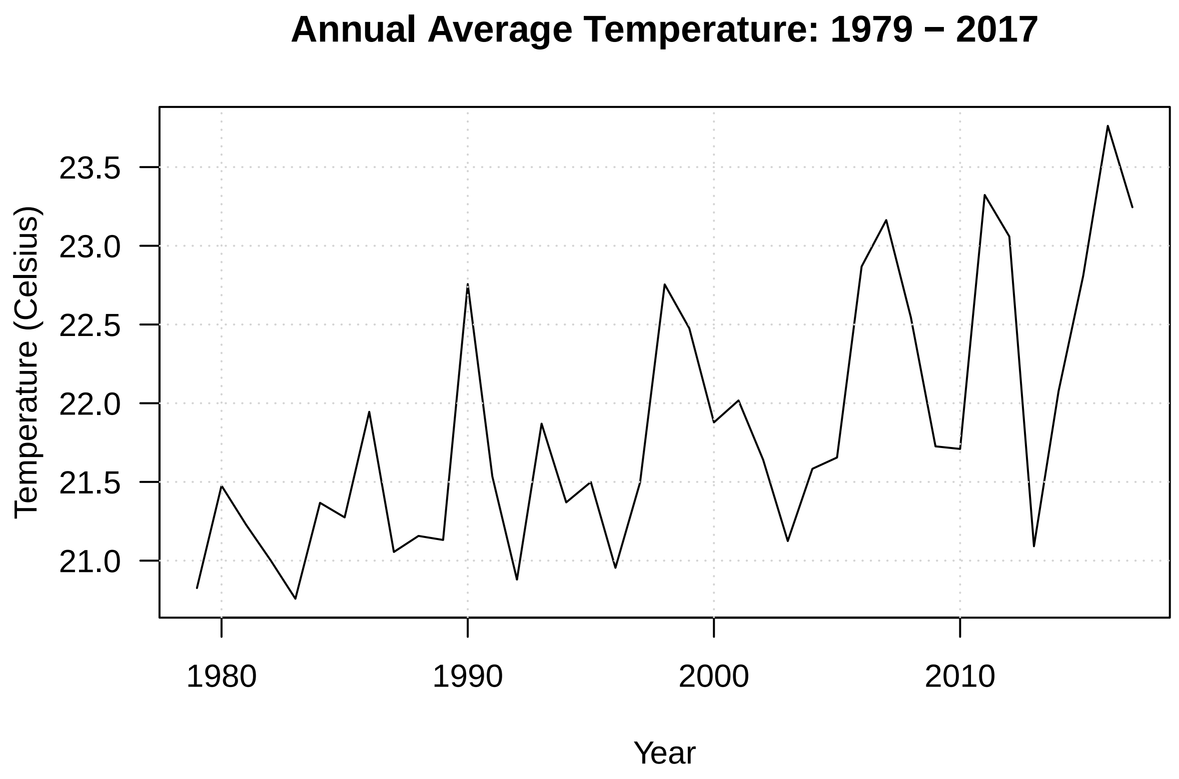
2. Second, aggregate the daily data to monthly average.

```
# Monthly Average Temperature
avg_by_mon <- array(dim = c(nlon, nlat, nmon, nyr))
for (i in 1:nlon) {
  for (j in 1:nlat) {
    dat <- cbind(NA_MaxT[i, j, ], as.factor(mon), as.factor(yr))
    avg_by_mon[i, j, , ] <- tapply(dat[, 1], list(dat[, 2], dat[, 3]), mean)
  }
}
```

3. Compute and plot the *annual average temperature* values using the *monthly average* data from 2.

```
# Annual Average Temperature
dat <- cbind(c(avg_by_mon[lon_id, lat_id, , ]),
as.factor(rep(1979:2017, each = 12)))
ann_mean <- tapply(dat[, 1], list(dat[, 2]), mean)
year <- 1979:2017

plot(year, ann_mean, type = "l", las = 1, xlab = "Year",
     ylab = "Temperature (Celsius)",
     main = "Annual Average Temperature: 1979 - 2017")
grid()
```



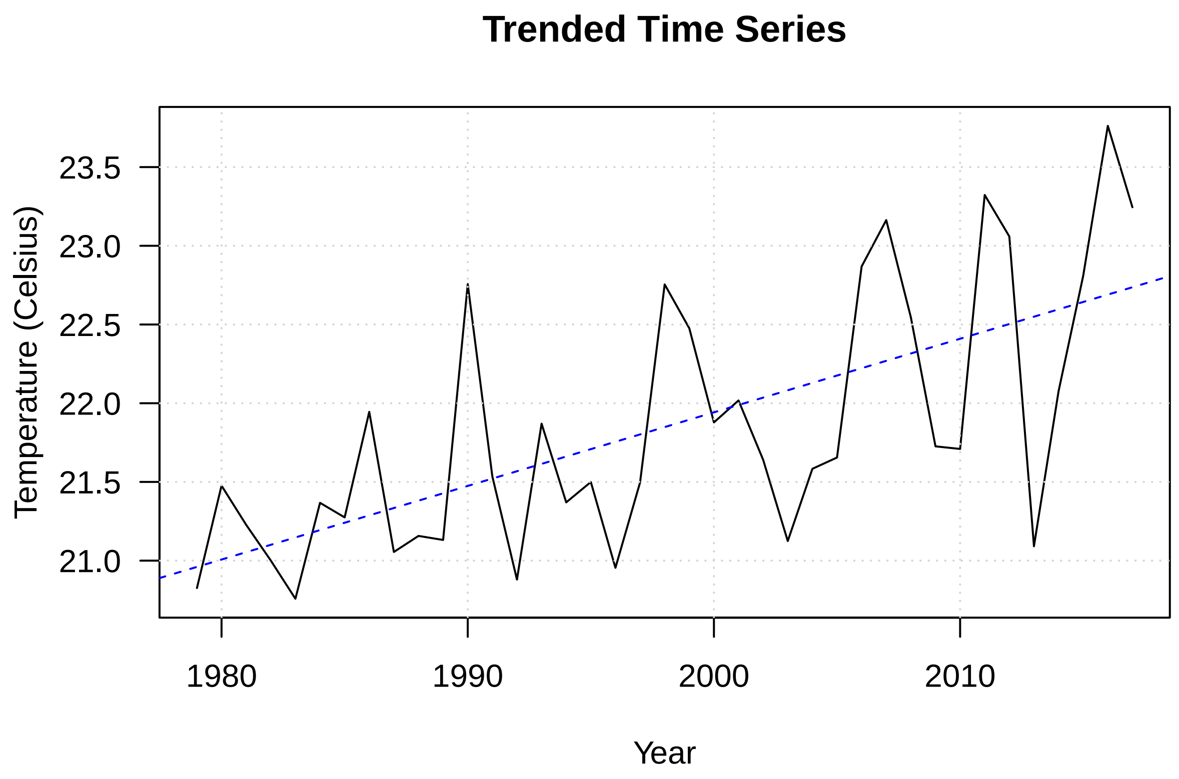
4. Describe the main features of the annual time series data in 3.

Answer: The time series analysis in 3 is a line graph which displays the average temperature (in degrees Celsius) of Clemson, SC from the years 1979 to 2017. During these 39 years, the average temperature was as low as 20.758 degrees Celsius in 1983 and as high as 23.76194 degrees Celsius in 2016. Looking at the graph, we see much volatility in average temperature from year to year. However, over the course of this time series, there is a general positive trend in the average temperature.

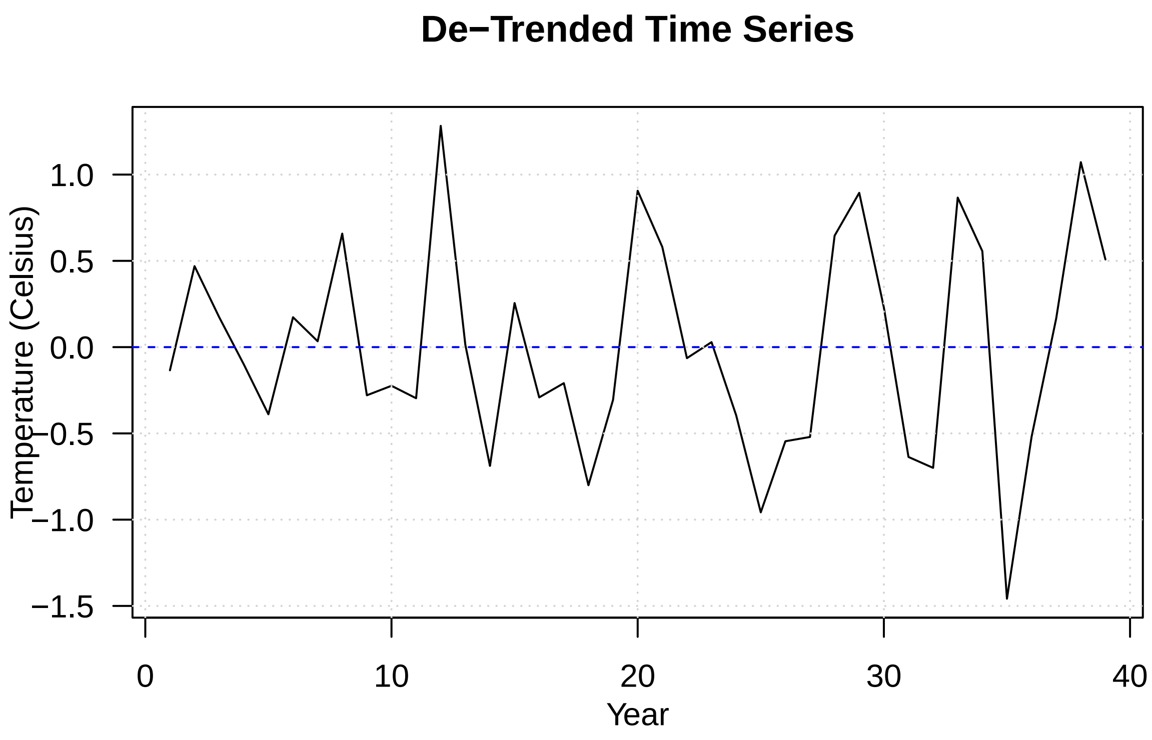
5. Is it reasonable to assume that there is a linear trend? If so, estimate and remove the trend to get the de-trended time series.

Answer: Yes, it is reasonable to assume that there is a linear trend. As I stated in the previous step, there is a general positive trend in the average temperature over the course of this time series. Please see below for the trended and de-trended time series plots.

```r
# Time Series with Linear Trend
lm <- lm(ann_mean ~ year)
plot(year, ann_mean, type = "l", las = 1, xlab = "Year",
     ylab = "Temperature (Celsius)",
     main = "Trended Time Series")
grid()
abline(lm, col = "blue", lty = 2)
```

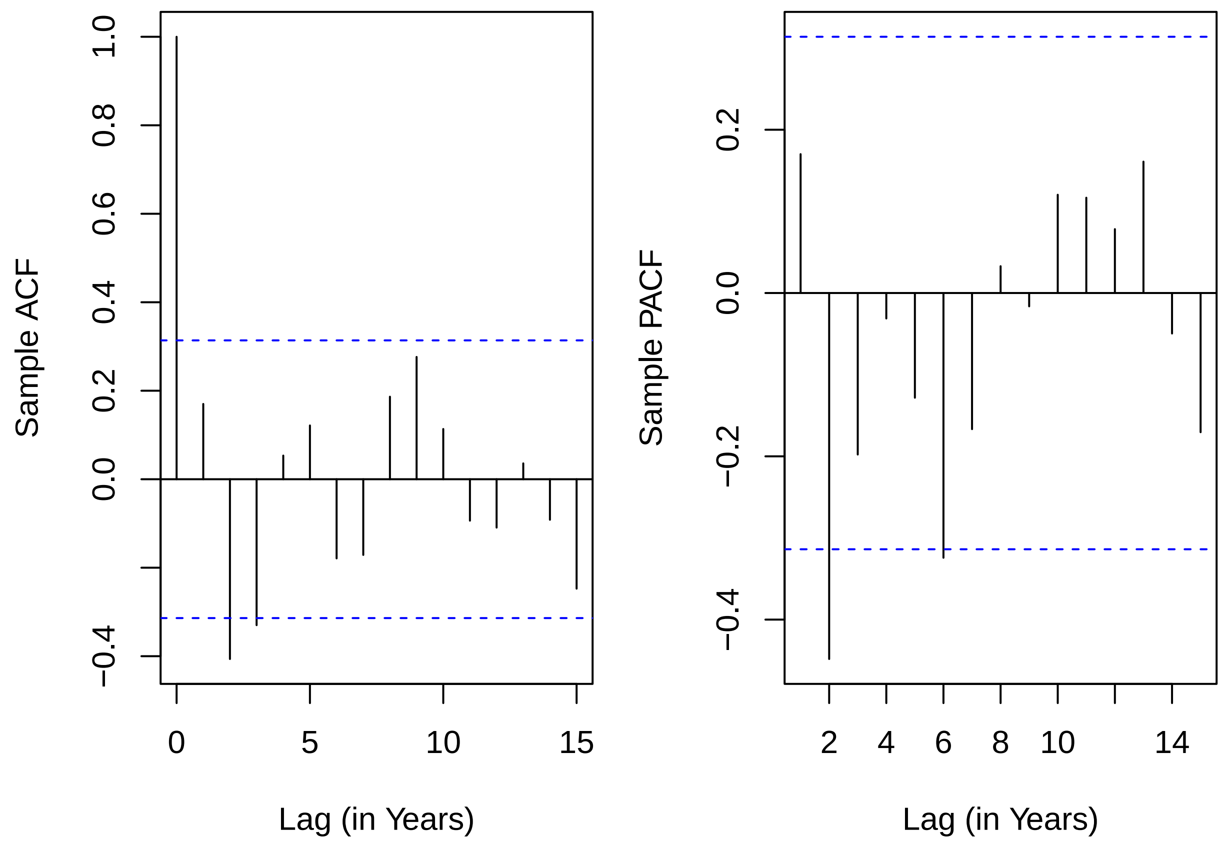


```
# De-Trended Time Series
deTrend <- resid(lm)
plot(deTrend, type = "l", ylab = "", xlab = "", las = 1,
     main = "De-Trended Time Series")
grid()
abline(h = 0, col = "blue", lty = 2)
mtext("Year", 1, line = 2)
mtext("Temperature (Celsius)", 2, line = 2.4)
```



6. Make ACF and PACF plots for the de-trended time series.

```
## ACF and PACF
par(mfrow = c(1, 2), mar = c(4, 4, 1, 1))
acf(deTrend, xlab = "Lag (in Years)", ylab = "Sample ACF", main = "")
pacf(deTrend, xlab = "Lag (in Years)", ylab = "Sample PACF", main = "")
```

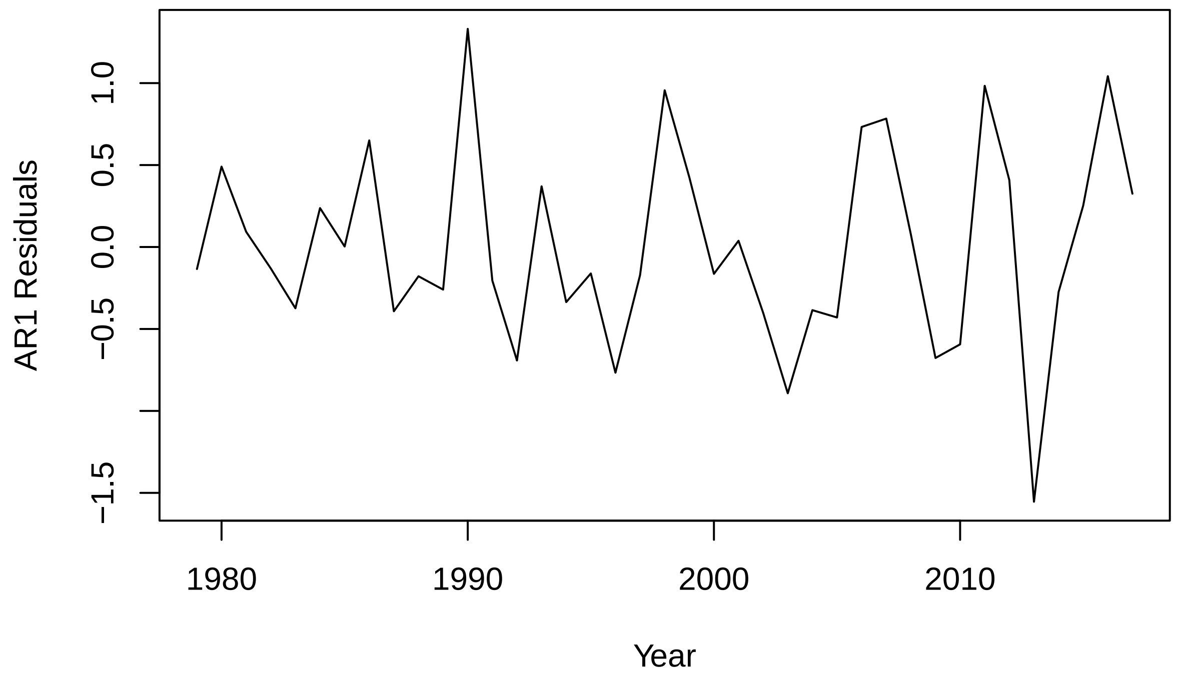


7. Fit some ARMA models and perform a model selection to determine the "best" model.

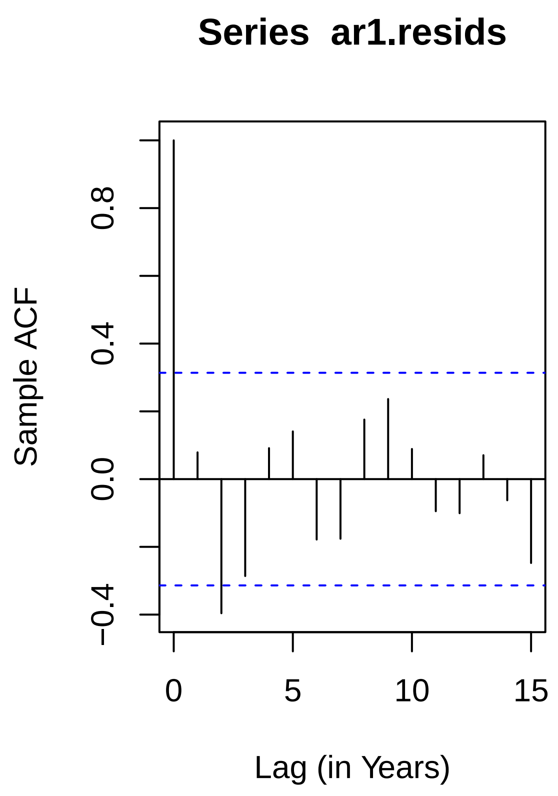
```
## AR(1)
(ar1.model <- arima(deTrend, order = c(1, 0, 0)))
```

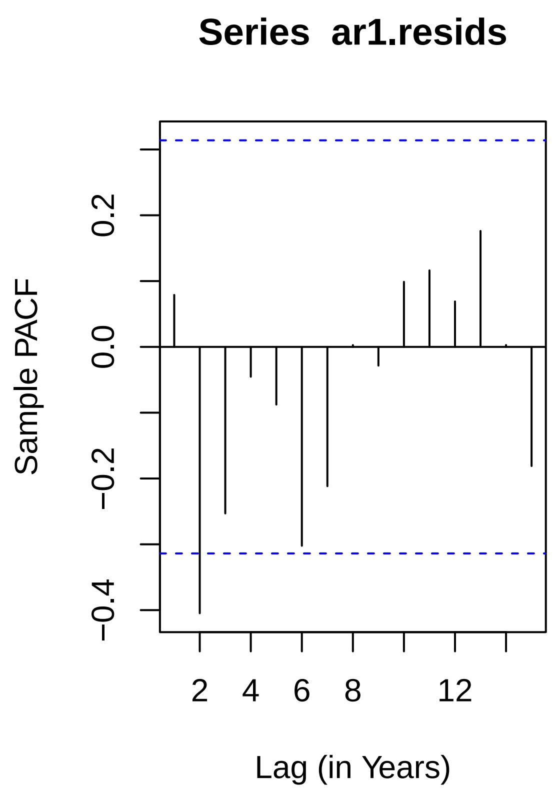
```
##
## Call:
## arima(x = deTrend, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.1692     0.0019
## s.e.  0.1572     0.1133
##
## sigma^2 estimated as 0.3489:  log likelihood = -34.82,  aic = 75.64
```

```
ar1.resids <- resid(ar1.model)
plot(1979:2017, ar1.resids, type = "l", xlab = "Year", ylab = "AR1 Residuals")
```



```
## Sample ACF and PACF of the Residuals
par(mfrow = c(1, 2))
acf(ar1.resids, ylab = "Sample ACF", xlab = "Lag (in Years)")
pacf(ar1.resids, ylab = "Sample PACF", xlab = "Lag (in Years)")
```





```
## Normal Q-Q Plot for the Residuals
qqnorm(ar1.resids, main = ""); qqline(ar1.resids, col = "blue")

## Test for Time Dependence for the Residuals
Box.test(ar1.resids, type = "Ljung-Box")
```

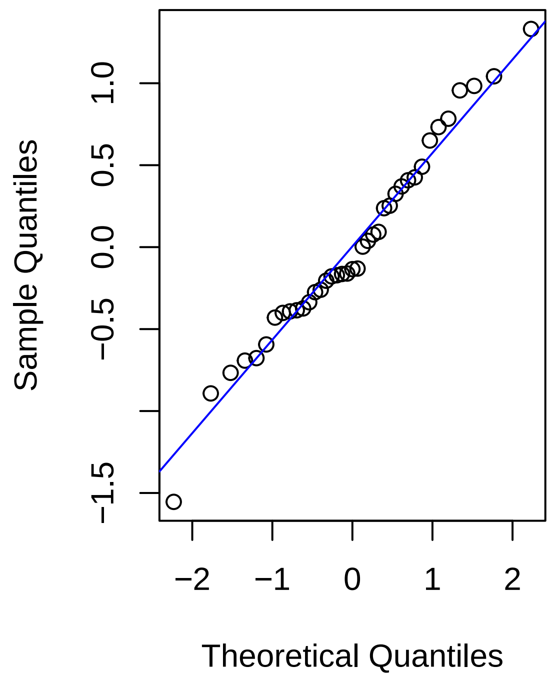
```
##
##  Box-Ljung test
##
## data:  ar1.resids
## X-squared = 0.26239, df = 1, p-value = 0.6085
```

```
## AR(2)
(ar2.model <- arima(deTrend, order = c(2, 0, 0)))
```

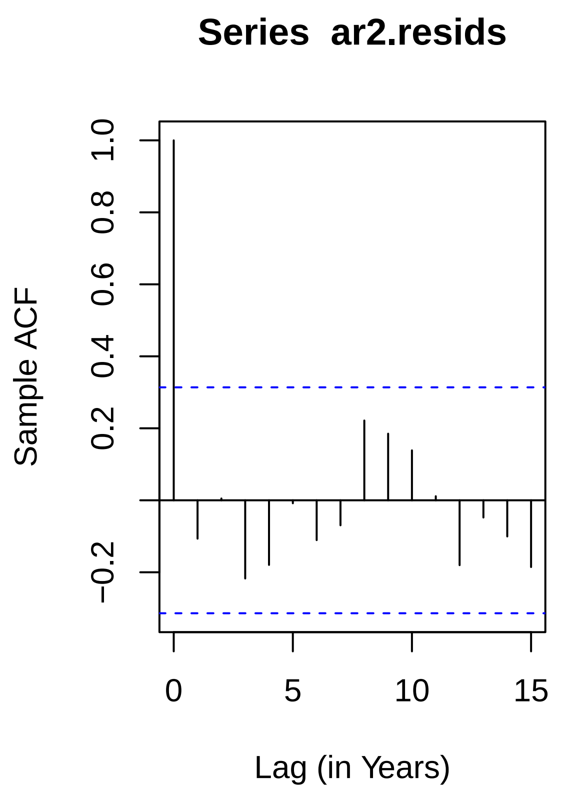
```
##
## Call:
## arima(x = deTrend, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       0.2363  -0.4737    -0.0175
## s.e.  0.1413   0.1441     0.0686
##
## sigma^2 estimated as 0.2717:  log likelihood = -30.19,  aic = 68.39
```

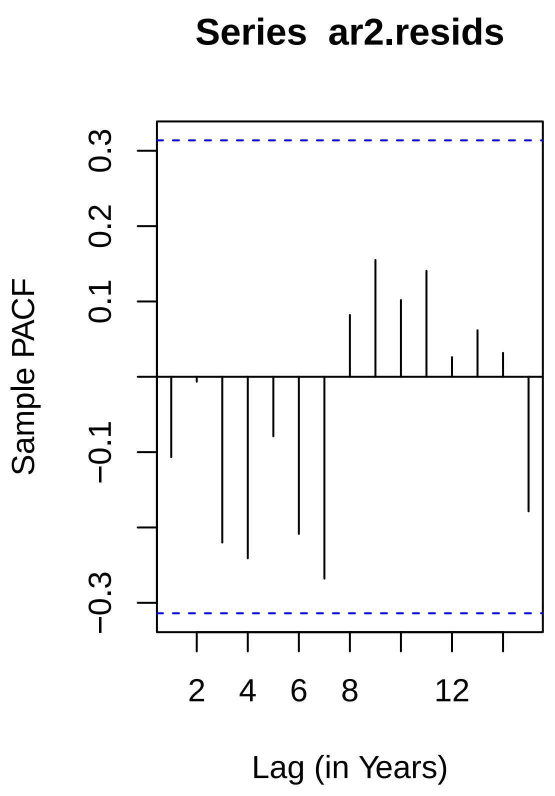
```r
## Calculate the Residuals
ar2.resids <- resid(ar2.model)

## Sample ACF and PACF of the residuals
par(mfrow = c(1, 2))
```



```r
acf(ar2.resids,  ylab = "Sample ACF", xlab = "Lag (in Years)")
pacf(ar2.resids,  ylab = "Sample PACF", xlab = "Lag (in Years)")
```





```
## Test for Time Dependence for the Residuals
Box.test(ar2.resids, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar2.resids
## X-squared = 0.47877, df = 1, p-value = 0.489
```

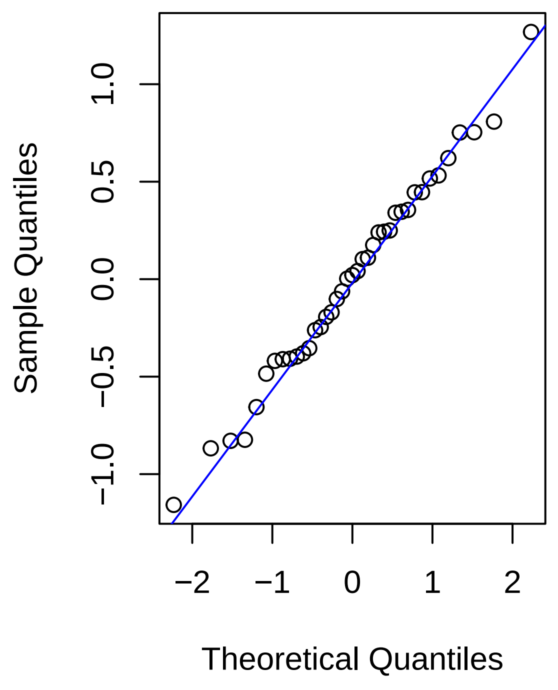
```
## Normal Q-Q Plot for the Residuals
qqnorm(ar2.resids, main = ""); qqline(ar2.resids, col = "blue")

## Fit the ARMA(2, 1) Model
(arma21.model <- arima(deTrend, order = c(2, 0, 1)))
```

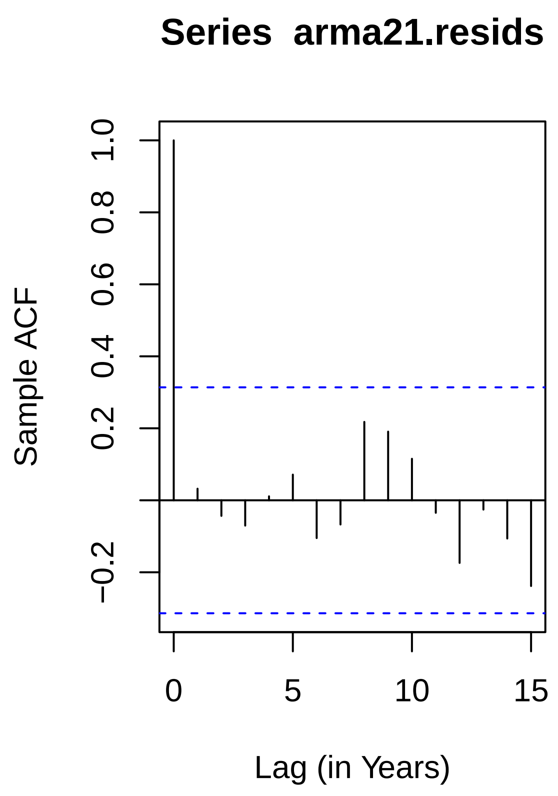
```
##
## Call:
## arima(x = deTrend, order = c(2, 0, 1))
##
## Coefficients:
##          ar1      ar2      ma1  intercept
##       0.7627  -0.5281  -0.7891    -0.0314
## s.e.  0.1732   0.1425   0.1771     0.0249
##
## sigma^2 estimated as 0.2342:  log likelihood = -27.67,  aic = 65.35
```

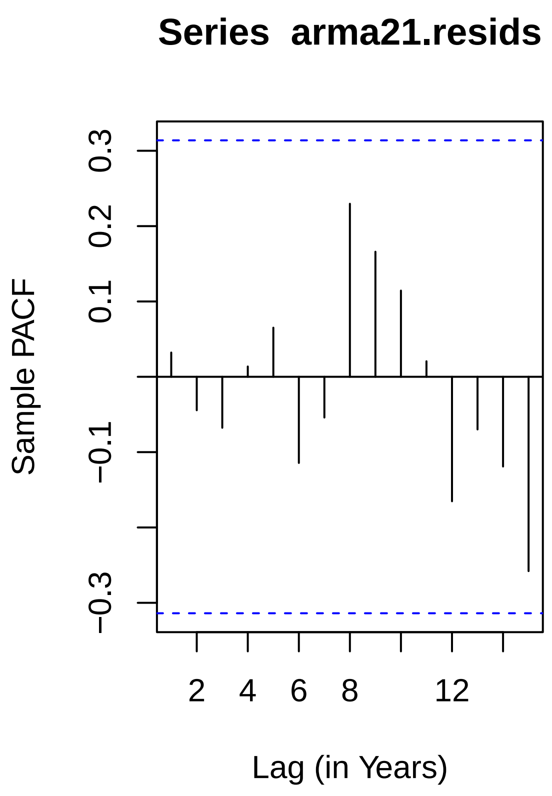
```r
## Calculate the Residuals
arma21.resids <- resid(arma21.model)

## Sample ACF and PACF of the Residuals
par(mfrow = c(1, 2))
```



```r
acf(arma21.resids,  ylab = "Sample ACF", xlab = "Lag (in Years)")
pacf(arma21.resids,  ylab = "Sample PACF", xlab = "Lag (in Years)")
```





```r
## Normal Q-Q Plot for the Residuals
qqnorm(arma21.resids, main = ""); qqline(arma21.resids, col = "blue")

## Test
Box.test(arma21.resids, type = "Ljung-Box")
```

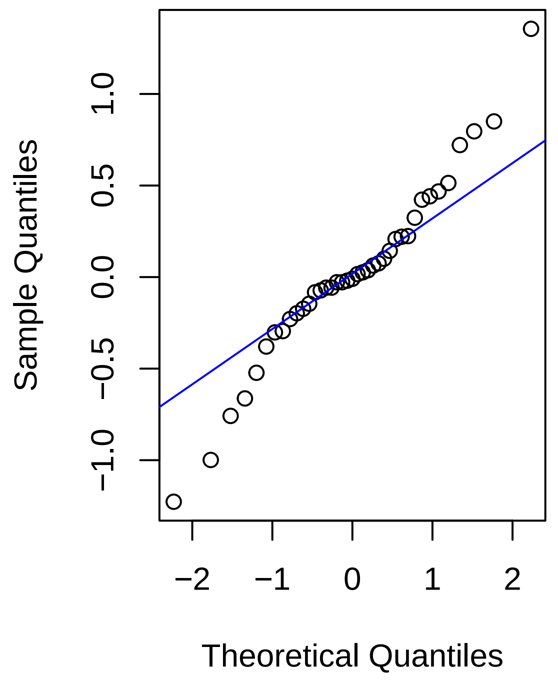
```
##
##  Box-Ljung test
##
## data:  arma21.resids
## X-squared = 0.043668, df = 1, p-value = 0.8345
```

```r
# Model Selection Using AIC
AIC(ar1.model); AIC(ar2.model); AIC(arma21.model)
```

```
## [1] 75.63519
```

```
## [1] 68.38739
```

```
## [1] 65.34855
```



Answer: The ARMA(2, 1) model is the best of the three ARMA models. This specific model had the highest p-value (0.8345) and smallest AIC value (65.3486).

8. Perform a one-year-ahead forecast. Calculate the prediction and provide a 95% prediction interval.

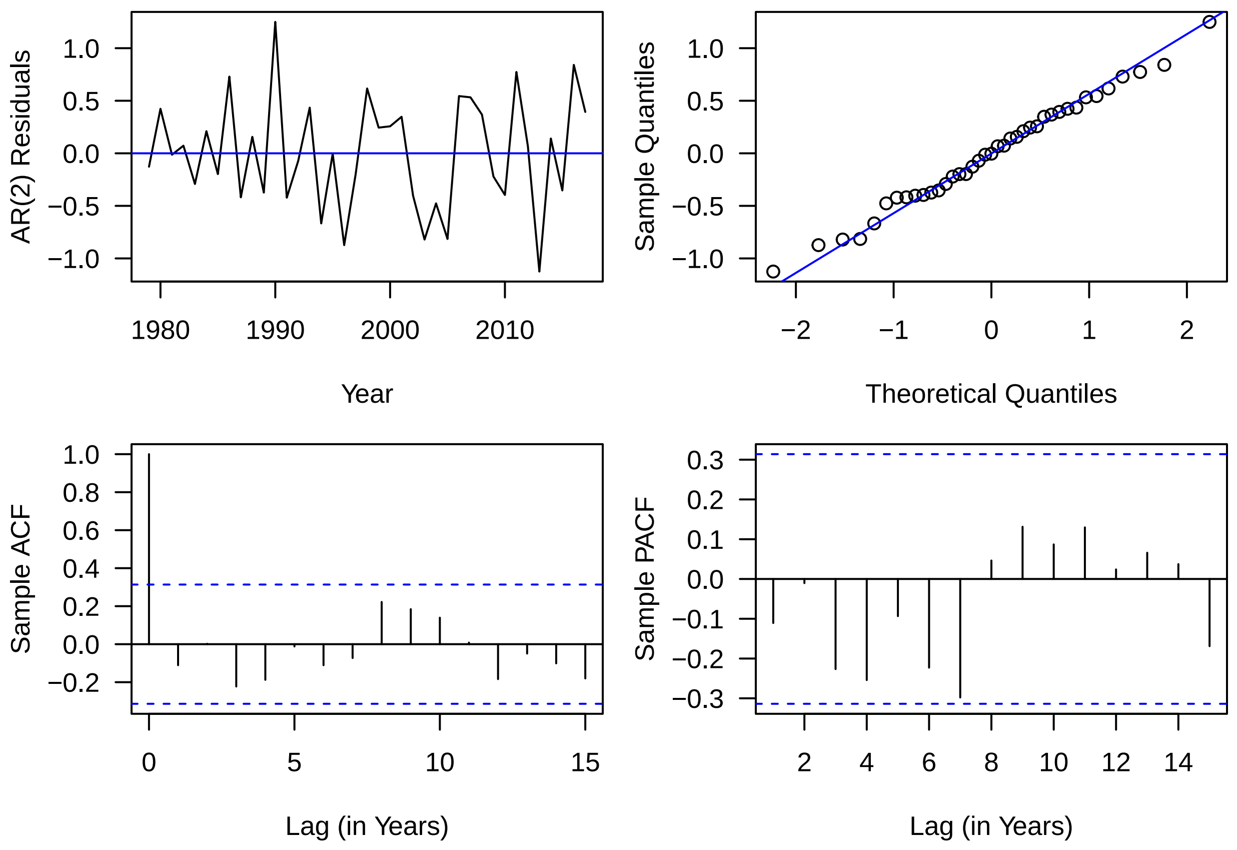
```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

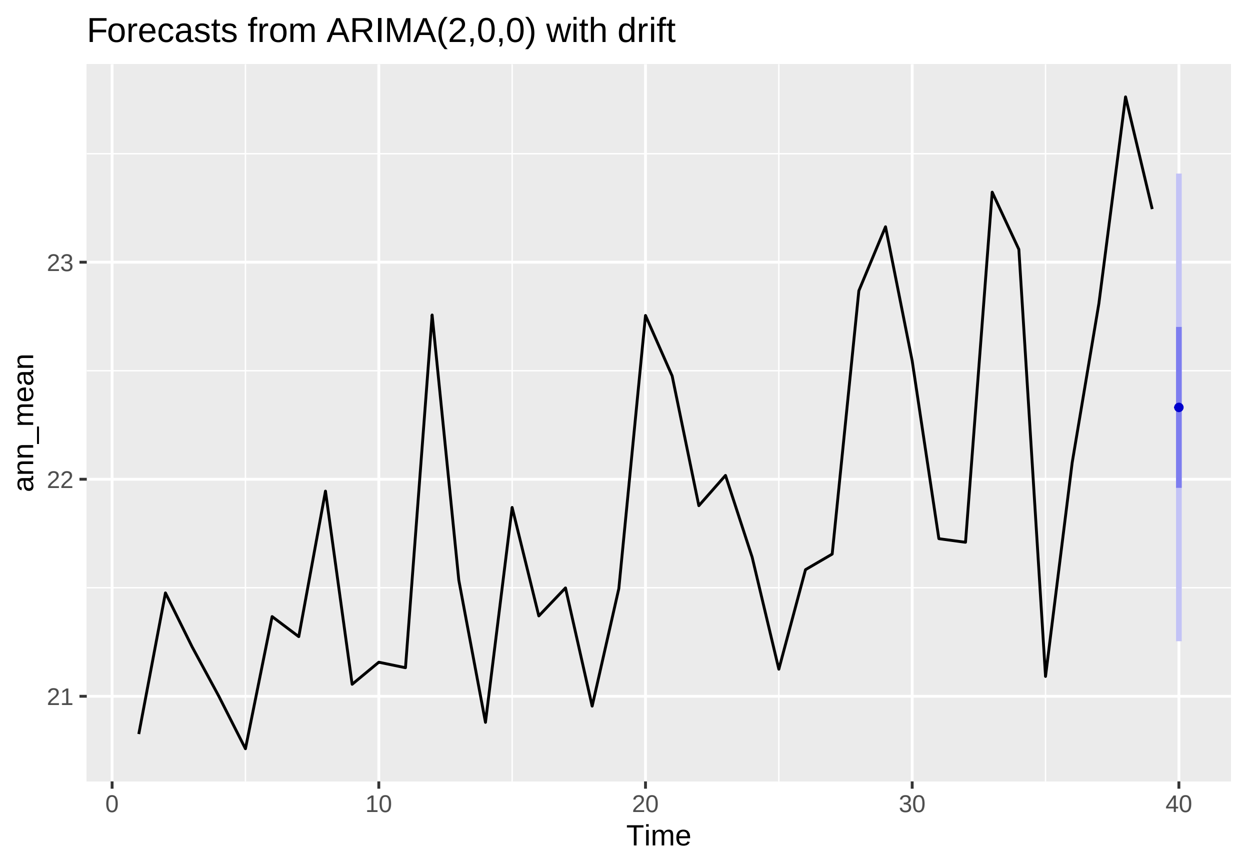
```
(fit <- Arima(ann_mean, order = c(2, 0, 0), include.drift = T))
```

```
## Series: ann_mean
## ARIMA(2,0,0) with drift
##
## Coefficients:
##          ar1      ar2  intercept   drift
##       0.2350  -0.4773    20.9278  0.0452
## s.e.  0.1413   0.1446     0.1423  0.0063
##
## sigma^2 = 0.3022:  log likelihood = -30.16
## AIC=70.32   AICc=72.14   BIC=78.64
```

```r
par(mfrow = c(2, 2), mar = c(4.1, 4, 1, 0.8), las = 1)
res <- fit$residuals
plot(1979:2017, res, type = "l", xlab = "Year", ylab = "AR(2) Residuals", las = 1)
abline(h = 0, col = "blue")
qqnorm(res, main = ""); qqline(res, col = "blue")
acf(res,  ylab = "Sample ACF", xlab = "Lag (in Years)")
pacf(res,  ylab = "Sample PACF", xlab = "Lag (in Years)")
```



```r
# One-Year Ahead Forecast
autoplot(forecast(fit, h = 1, level = c(50, 95)))
```



```r
forecast(fit, h = 1, level = c(50, 95))
```

```
##    Point Forecast    Lo 50    Hi 50    Lo 95    Hi 95
## 40       22.33101 21.96026 22.70177 21.25366 23.40837
```

Answer: The one-year-ahead prediction for the year 2018 is 22.33101 degrees Celsius, with a 95% prediction interval of (21.25366, 23.40837).

Problem 2

This problem uses the `fields` package to conduct spatial interpolation using a Gaussian process model. The `ozone2` dataset in the `fields` package interpolates average ozone levels.

1. Use the following script to load the data:

```r
library(fields)
data(ozone2)
loc <- ozone2$lon.lat
rg <- apply(loc, 2, range)
y <- ozone2$y[16, ]
good <- !is.na(y)
```

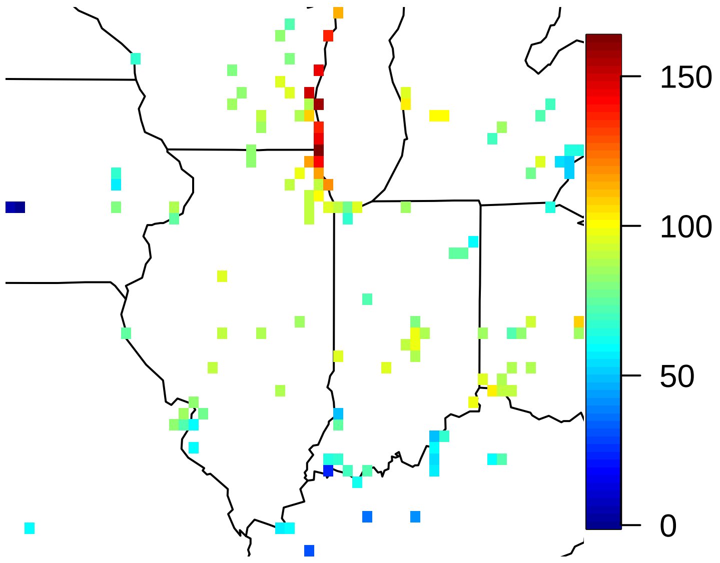
2. Visualize the spatial data and describe the findings:

```r
library(maps)
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:viridis':
##
##     unemp
```

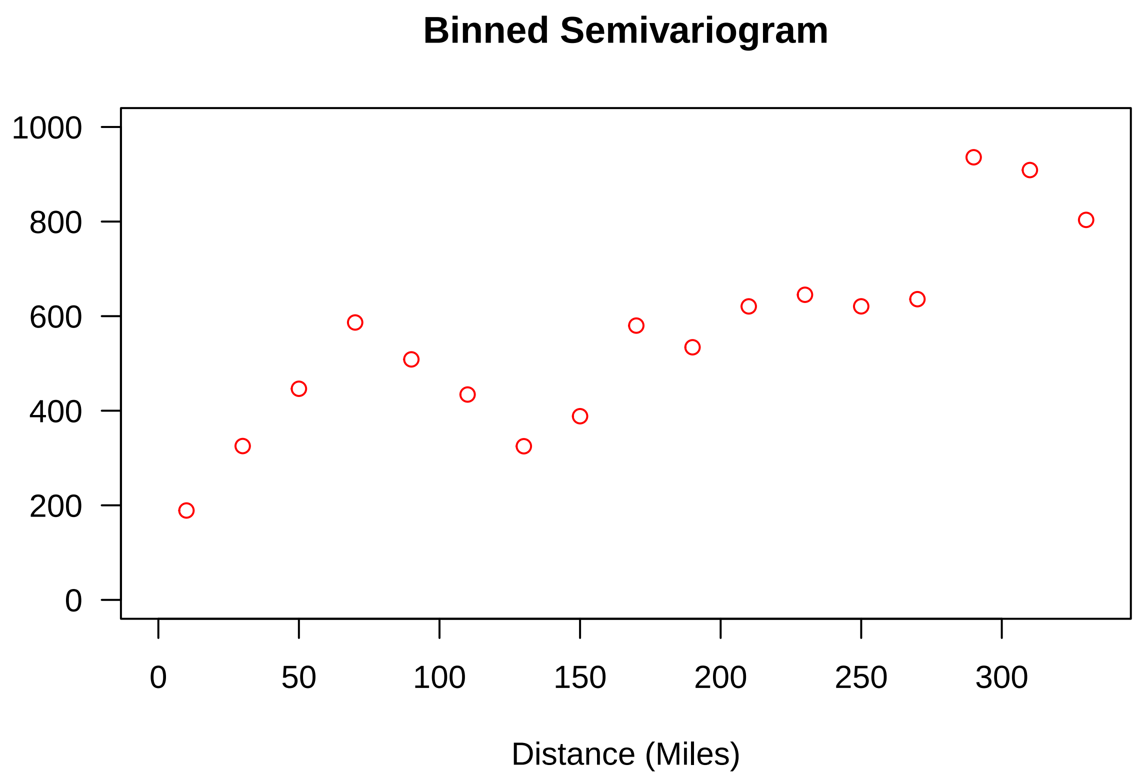
```r
map("state", xlim = rg[, 1], ylim = rg[, 2])
quilt.plot(loc[good, ], y[good], nx = 60, ny = 48, add = T)
```



Answer: The following map visualizes the average ozone levels in the Great Lakes region of the United States, specifically as it applies to the states of Michigan, Ohio, Indiana, Kentucky, Illinois, Wisconsin, Iowa, Missouri, and Minnesota. Based on the findings, there appears to be extremely high average ozone levels (100-150 on the scale) along the western shores of Lake Michigan, in the states of Illinois and Wisconsin. These higher than normal metrics can be attributed to the fact that this specific region is known as the Greater Chicago Metropolitan area. This is the most populous, developed, and industrialized area on the map, likely contributing to these high average ozone levels. Comparably, most of the regions outside of the Greater Chicago Metropolitan area exhibit lower average ozone levels (100 or lower on the scale).

3. Make two variograms: one without spatial trend and another one with linear spatial trend. Explain the differences between these two variograms.

```r
# With Spatial Trend
d <- rdist.earth(loc[good, ]); maxd <- max(d)
vgram <- vgram(loc = loc[good, ], y = y[good], N = 30, lon.lat = T)
plot(vgram$stats["mean", ] ~ vgram$centers, main = 'Binned Semivariogram',
     las = 1, ylab = "", xlab = "Distance (Miles)", col = "red",
     xlim = c(0, 0.5 * maxd), ylim = c(0, 1000))
```

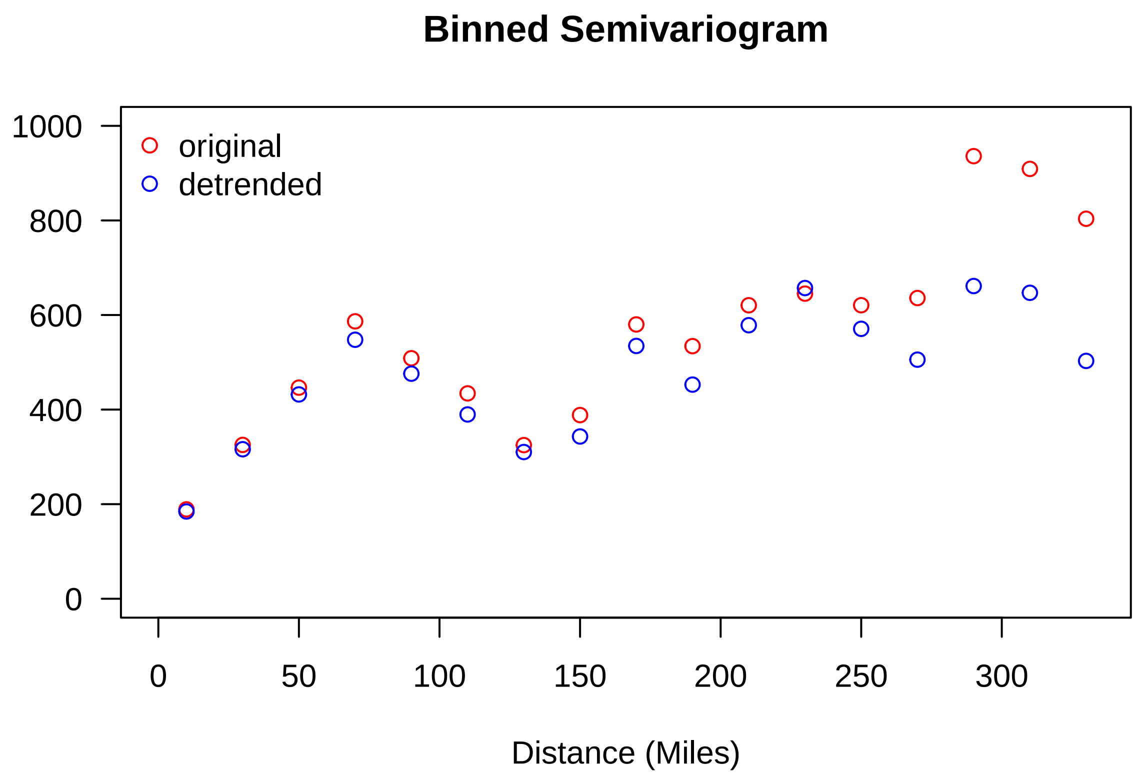


```r
# Without Spatial Trend
d <- rdist.earth(loc[good, ]); maxd <- max(d)
vgram <- vgram(loc = loc[good, ], y = y[good], N = 30, lon.lat = T)
plot(vgram$stats["mean", ] ~ vgram$centers, main = 'Binned Semivariogram',
     las = 1, ylab = "", xlab = "Distance (Miles)", col = "red",
     xlim = c(0, 0.5 * maxd), ylim = c(0, 1000))

# Remove Linear Spatial Trend
lm <- lm(y[good] ~ loc[good, ])

vgram <- vgram(loc = loc[good, ], y = lm$residuals, N = 30, lon.lat = TRUE)

# Use 30 bins
points(vgram$stats["mean", ] ~ vgram$centers, col = "blue")
legend("topleft", legend = c("original", "detrended"),
       col = c("red", "blue"), bty = "n", pch = 1)
```



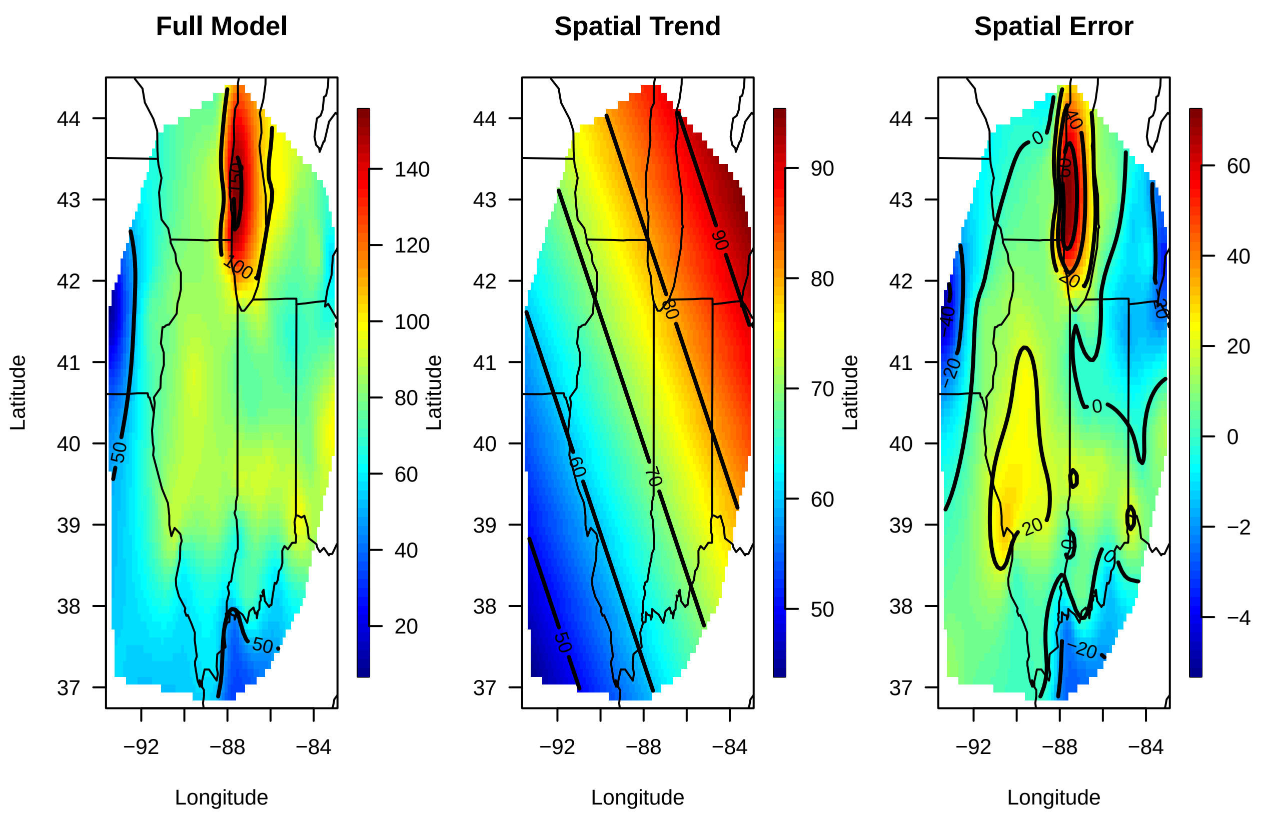
Answer: Looking at the two variograms , it appears that the plot of the variogram with a spatial trend exhibits a positive, linear trend in variance as distance increases. On the other hand, the plot of the variogram with no spatial trend does not appear to exhibit a clear positive or negative trend. However, the trend is definitely still linear. In most cases, eliminating the spatial trend decreased the variation.

4. Fit a Gaussian process to the data and decompose the prediction into a spatial trend part and a spatially correlated error part:

```
fit <- spatialProcess(loc[good, ], y[good])
out.full <- predictSurface(fit)
out.poly <- predictSurface(fit, just.fixed = T)
out.spatial <- out.full
out.spatial$z <- out.full$z - out.poly$z
set.panel(1, 3)
```

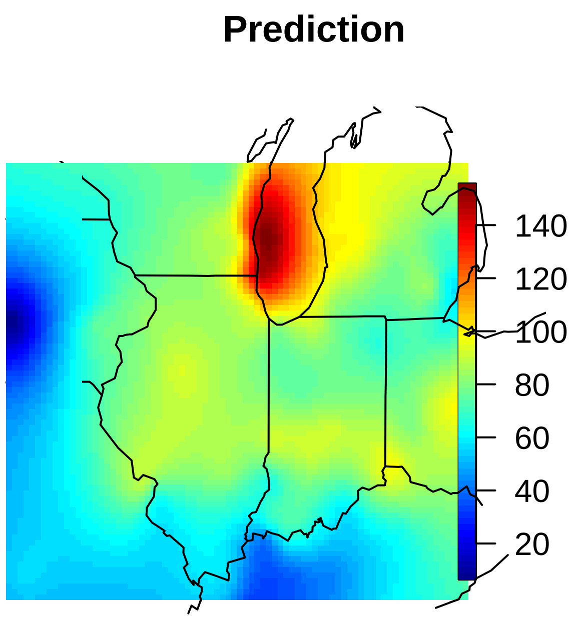
```
## plot window will lay out plots in a 1 by 3 matrix
```

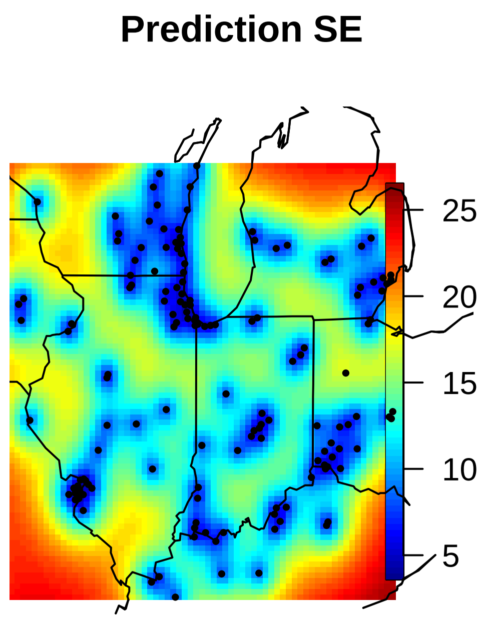
```
surface(out.full, las = 1, xlab = "Longitude", ylab = "Latitude", col = tim.colors())
title("Full Model")
map("state", add = T)
surface(out.poly, las = 1, xlab = "Longitude", ylab = "Latitude", col = tim.colors())
map("state", add = T)
title("Spatial Trend")
surface(out.spatial, las = 1, xlab = "Longitude", ylab = "Latitude", col = tim.colors())
map("state", add = T)
title("Spatial Error")
```



5. Make prediction map and the associated prediction uncertainty map. Explain the spatial variation in the uncertainty map.

```
xg <- fields.x.to.grid(loc[good, ])
Pred <- predict.mKrig(fit, xnew = expand.grid(xg))
SE <- predictSE(fit, xnew = expand.grid(xg))
par(mfrow = c(1, 2), las = 1)
map("state", xlim = rg[, 1], ylim = rg[, 2])
image.plot(xg[[1]], xg[[2]], matrix(Pred, 80), add = T)
map("state", xlim = rg[, 1], ylim = rg[, 2], add = T)
title("Prediction")
map("state", xlim = rg[, 1], ylim = rg[, 2])
image.plot(xg[[1]], xg[[2]], matrix(SE, 80), add = T)
map("state", xlim = rg[, 1], ylim = rg[, 2], add = T)
title("Prediction SE")
points(loc, pch = 16, cex = 0.5)
```





Answer: The dark blue areas have lower standard errors which lead to higher levels of certainty, as opposed to the red areas that have higher standard errors which lead to lower levels of certainty. It appears that there is less spatial variation towards the middle of the map and more spatial variation towards its edges. Perhaps there is more spatial variation towards the map edges because of the unknown that lays on the other side.