Spatial autocorrelation

Eddie Wu

2023-08-09

Introduction

This .Rmd file is to test the spatial autocorrelation in our black carp data file, and try to find the best distance that reduces the spatial autocorrelation when doing subsampling. In this file, we:

1. Show the distribution of our data points on a map.
2. Use moran's I test to test the global spatial autocorrelation in our dataset.
3. Use the correlog plots to examine how spatial autocorrelation changes with distances.
4. Use the variogram to further test point 2.
5. Subsample at 250 km and 550 km, and check the local moran's I after sub-sampling to see if spatial autocorrelation has been reduced.
6. Conduct laditudinal stratification analysis.
7. Look at the Chinese dataset - 17 data points.

```r
library(gstat)
library(ggplot2)
library(dplyr)
library(spdep)
library(sp)
library(nlme)
library(ape)
library(MuMIn)
library(raster)
library(ncf)
library(knitr)
library(rnaturalearth)
library(sf)
```

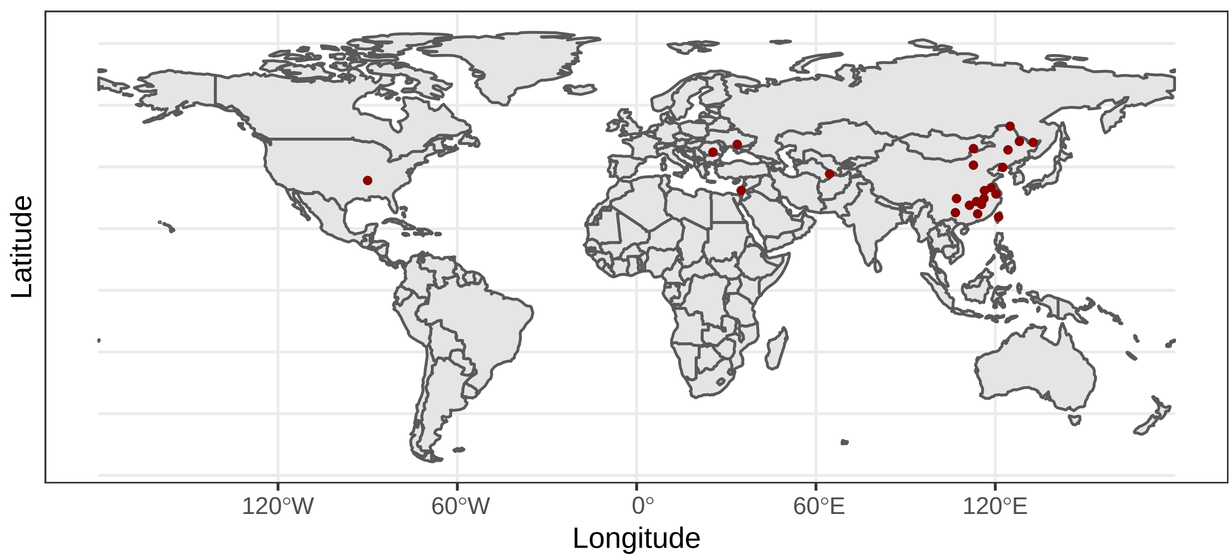
```r
## Import location data
location <- read.csv("location_no_temps.csv")
location <- unique(location) # remove duplicating locations

# Clean the data
carp <- read.csv("eddie_carp_new.csv")
carp.r <- carp %>%
  filter(sex != "male") %>% # keep the non-male data points
  distinct(location, .keep_all = TRUE) # remove all repeating locations

## Download one file to get the spatial points
# (this defines what projection to use when converting to spatial objects)
tmin.1979 <- brick("cpc/tmin.1979.nc", varname = "tmin")
```

```
## Loading required namespace: ncdf4
```

```
tmin.1979<-rotate(tmin.1979)
```

Data distribution

```
## Get the world data
world <- ne_countries(scale = "small", returnclass = "sf")

## Plots (global plot)
world %>%
  filter(admin != "Antarctica") %>%
  # remove antarctica
  ggplot()+
  geom_sf()+
  geom_point(aes(x=Longitude, y=Latitude), data = location,
             size = 1, color = "darkred")+
  theme_bw()+
  xlab("Longitude") + ylab("Latitude")
```



Calculate the residuals from the linear model

```
# Define the model
lm.annual <- lm(log(carp.r$AAM)~carp.r$AnnualTemp) # Annual temperature
lm.cold <- lm(log(carp.r$AAM)~carp.r$ColdTemp) # Cold temperature
lm.warm <- lm(log(carp.r$AAM)~carp.r$WarmTemp) # Warm temperature
```

```r
# See the results
summary(lm.annual)
```

```
##
## Call:
## lm(formula = log(carp.r$AAM) ~ carp.r$AnnualTemp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45033 -0.15088 -0.02442  0.11877  0.54619
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.027038   0.088922  22.796  2.7e-16 ***
## carp.r$AnnualTemp -0.018853   0.006468  -2.915  0.00828 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2131 on 21 degrees of freedom
## Multiple R-squared:  0.288,  Adjusted R-squared:  0.2541
## F-statistic: 8.495 on 1 and 21 DF,  p-value: 0.008285
```

```r
summary(lm.cold)
```

```
##
## Call:
## lm(formula = log(carp.r$AAM) ~ carp.r$ColdTemp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41846 -0.12055 -0.02502  0.09514  0.57767
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.79047    0.04397  40.723  < 2e-16 ***
## carp.r$ColdTemp -0.01190    0.00389  -3.059  0.00596 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.21 on 21 degrees of freedom
## Multiple R-squared:  0.3083, Adjusted R-squared:  0.2753
## F-statistic: 9.359 on 1 and 21 DF,  p-value: 0.005955
```

```r
summary(lm.warm)
```

```
##
## Call:
## lm(formula = log(carp.r$AAM) ~ carp.r$WarmTemp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48388 -0.15016  0.03813  0.11838  0.54909
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      2.26257    0.30290    7.47 2.43e-07 ***
## carp.r$WarmTemp -0.01943    0.01262   -1.54    0.139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2394 on 21 degrees of freedom
## Multiple R-squared:  0.1015, Adjusted R-squared:  0.05867
## F-statistic: 2.371 on 1 and 21 DF,  p-value: 0.1385
```

Global spatial autocorrelation

```r
## Make spatial dataframe
coords <- data.frame("long"=location[,3],"lat"=location[,2])
df <- data.frame(a = 1:nrow(location[3]))
spatial.data <- SpatialPointsDataFrame(coords,df,proj4string = tmin.1979@crs)

# Get a distance matrix from all points
dists <- spDists(spatial.data, longlat = TRUE)

## Run the Moran.I test on the residuals
Moran.annual <- Moran.I(lm.annual$residuals, dists)
Moran.cold <- Moran.I(lm.cold$residuals, dists)

global.moran <- data.frame(
  Model = c("Moran.annual", "Moran.cold"),
  Observed = c(Moran.annual$observed, Moran.cold$observed),
  Expected = c(Moran.annual$expected, Moran.cold$expected),
  sd = c(Moran.annual$sd, Moran.cold$sd),
  p.value = c(Moran.annual$p.value, Moran.cold$p.value)
)

kable(global.moran)
```

Model	Observed	Expected	sd	p.value
Moran.annual	-0.0276980	-0.0454545	0.0316840	0.5751899
Moran.cold	-0.0330776	-0.0454545	0.0317925	0.6970518

There is no global spatial autocorrelation for the entire dataset.

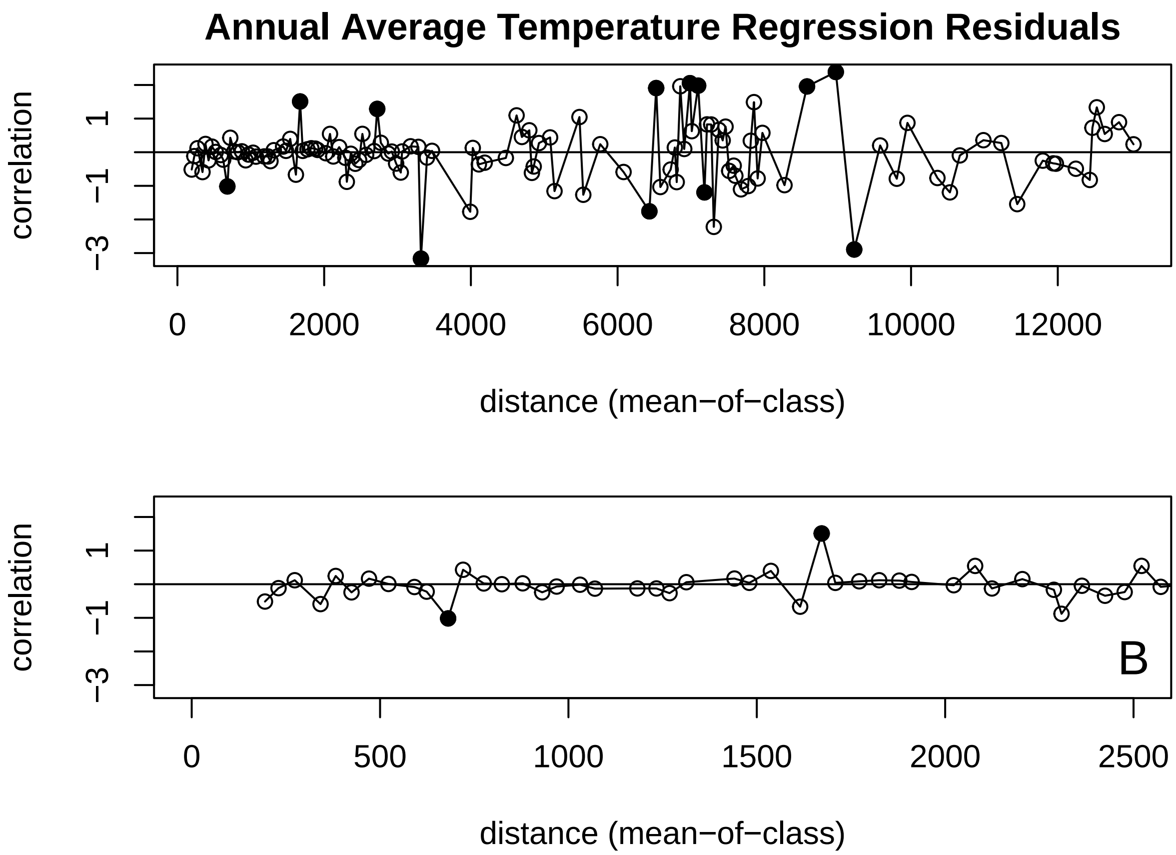
Local spatial autocorrelation

```r
## Annual temperature
par(mfrow=c(2,1),mar=c(4,4,2,2))

# Use the coorelog function to develop the relationship
test <- correlog(coords$long, coords$lat, lm.annual$residuals,
                 increment=50, resamp=500, latlon=T)
```

```
## 50  of  500 100  of  500 150  of  500 200  of  500 250  of  500 300  of  500 350  of  500 400  of  5
```

```r
# Plot with the entire distance range
plot(test, main="Annual Average Temperature Regression Residuals")
abline(h=0)
```

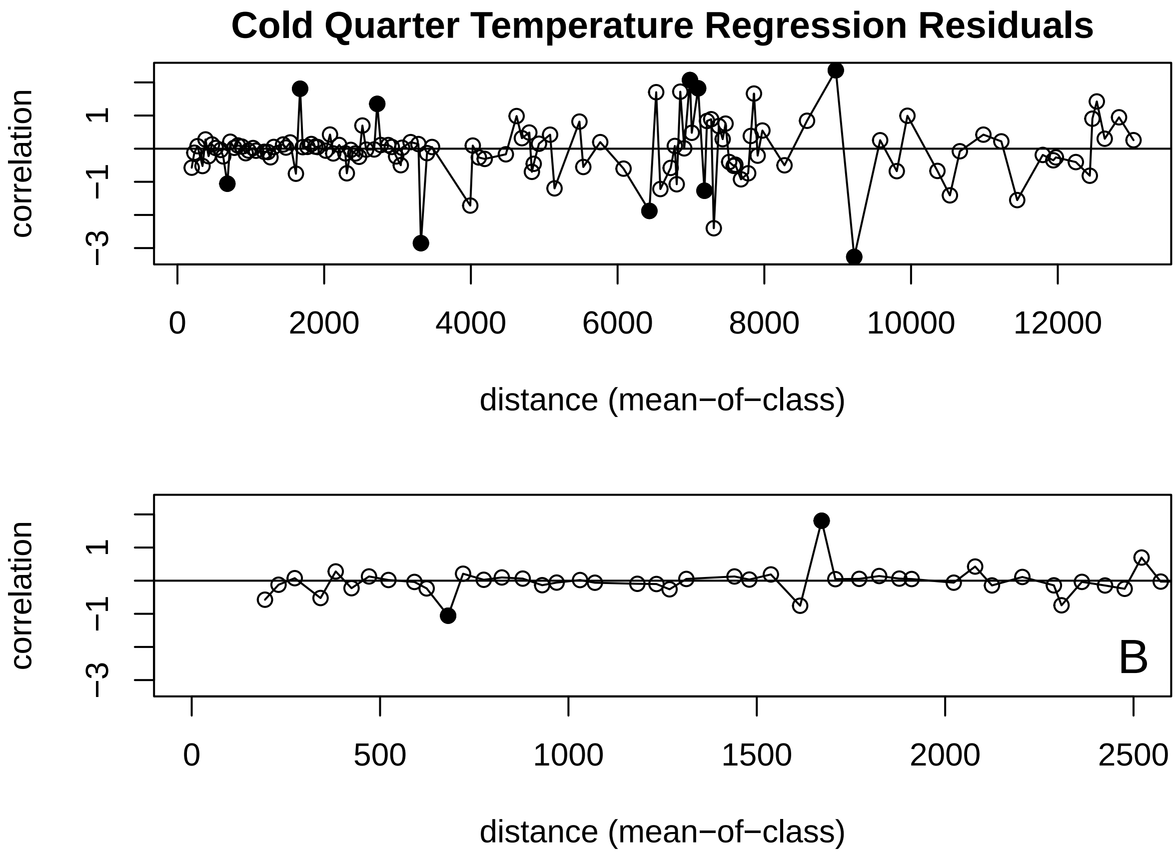
```r
text(17400, min(test$correlation)+1, "A", cex=1.5)

# Reduce the distance range to 2500 km
plot(test, main="", xlim=c(0,2500))
abline(h=0)
text(2500, min(test$correlation)+1, "B", cex=1.5)
```




```r
## Cold temperature
par(mfrow=c(2,1),mar=c(4,4,2,2))

# Use the coorelog function to develop the relationship
test <- correlog(coords$long, coords$lat, lm.cold$residuals,
                 increment=50, resamp=500, latlon=T)
```

```
## 50  of  500 100  of  500 150  of  500 200  of  500 250  of  500 300  of  500 350  of  500 400  of  5
```

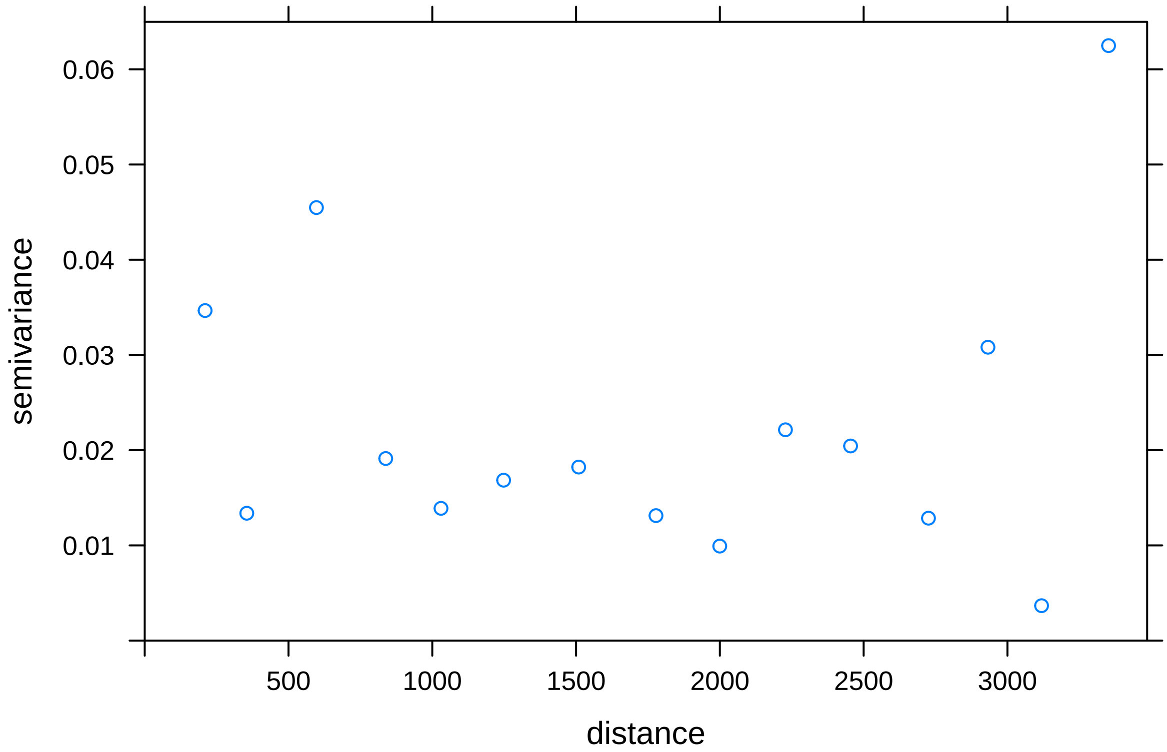
```r
# Plot with the entire distance range
plot(test, main="Cold Quarter Temperature Regression Residuals")
abline(h=0)
text(17400, min(test$correlation)+1, "A", cex=1.5)

# Reduce the distance range to 2500 km
plot(test, main="", xlim=c(0,2500))
abline(h=0)
text(2500, min(test$correlation)+1, "B", cex=1.5)
```



In general, the correlog plots suggest that for both temperature, 600 km is a good distance to reduce local spatial autocorrelation.

Variogram

```
## Create a data frame to store the necessary information
vario.data <- data.frame(residuals = lm.annual$residuals,
                         x = coords$long, y = coords$lat)

# Turn them to spatial points using our projection
geo.spatial<-SpatialPoints(coords, proj4string = tmin.1979@crs)


## Variogram
vario <- variogram(vario.data$residuals~1, data = geo.spatial)
plot(vario)
```



If we pick 250 km as the cutoff distance for sub-sampling, we would have 20 location points.

If we pick 550 km as the cutoff distance for sub-sampling, we would have only 12 location points.

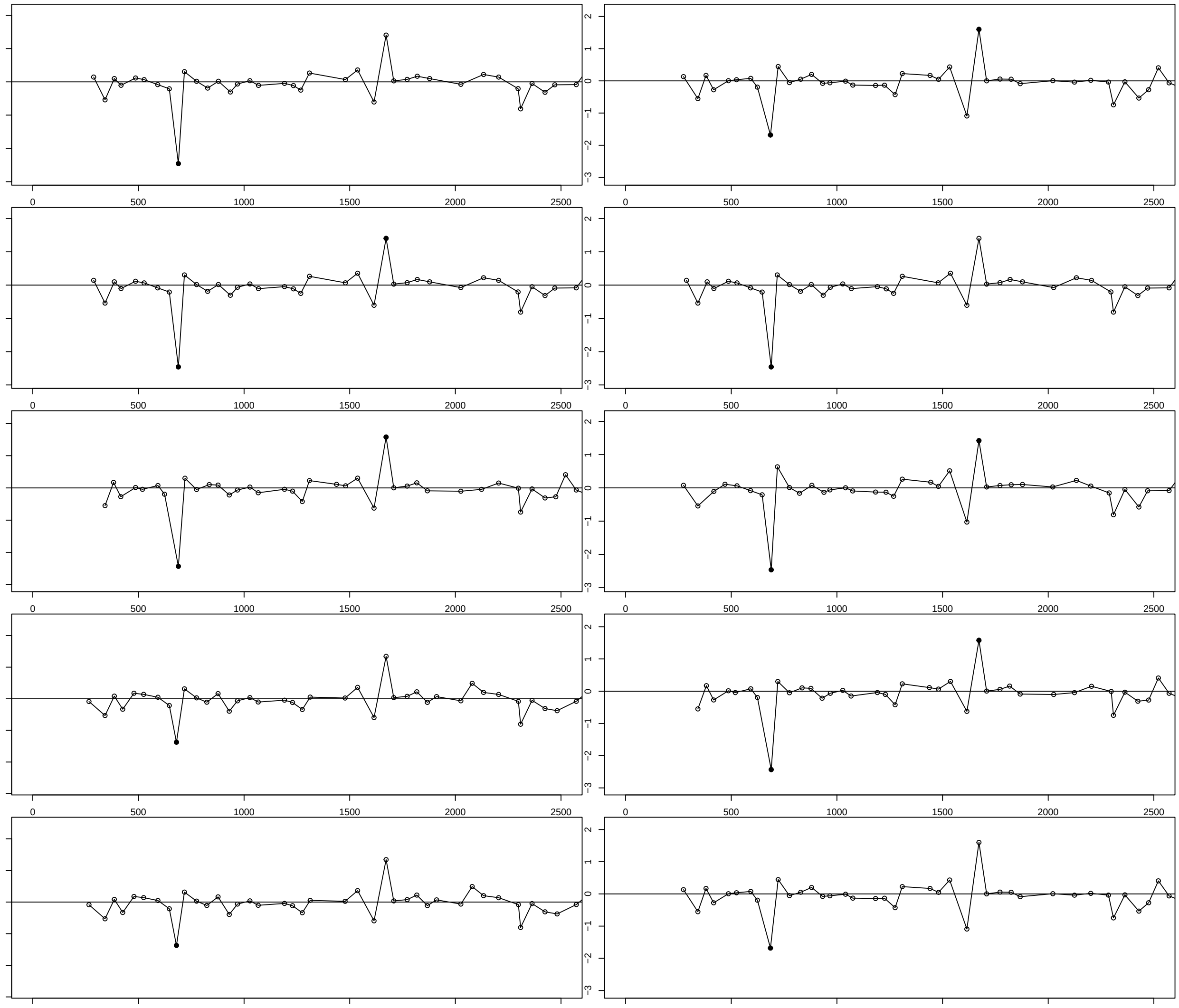
Subsampling at 250km

Here, we try to sub-sample at 250km and check the local moran's I (correlog plot) after sub-sampling.

Local moran's I - annual 250

```
# Store the r2 values
r2.250.raw <- matrix(NA,10, 2)
colnames(r2.250.raw) <- c("annual","cold")

table(carp.r$spatial.code.250)

par(mfrow = c(5, 2))
par(mar=c(1,1,1,1))

## Check the local moran's results after sub-sampling
for (i in 1:10){
  sub <- carp.r %>% group_by(spatial.code.250) %>% sample_n(size=1)
  reg.sub.annual <- lm(log(sub$AAM)~sub$AnnualTemp)
  test <- correlog(sub$longitude, sub$latitude, reg.sub.annual$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,2500))
  abline(h=0)
```

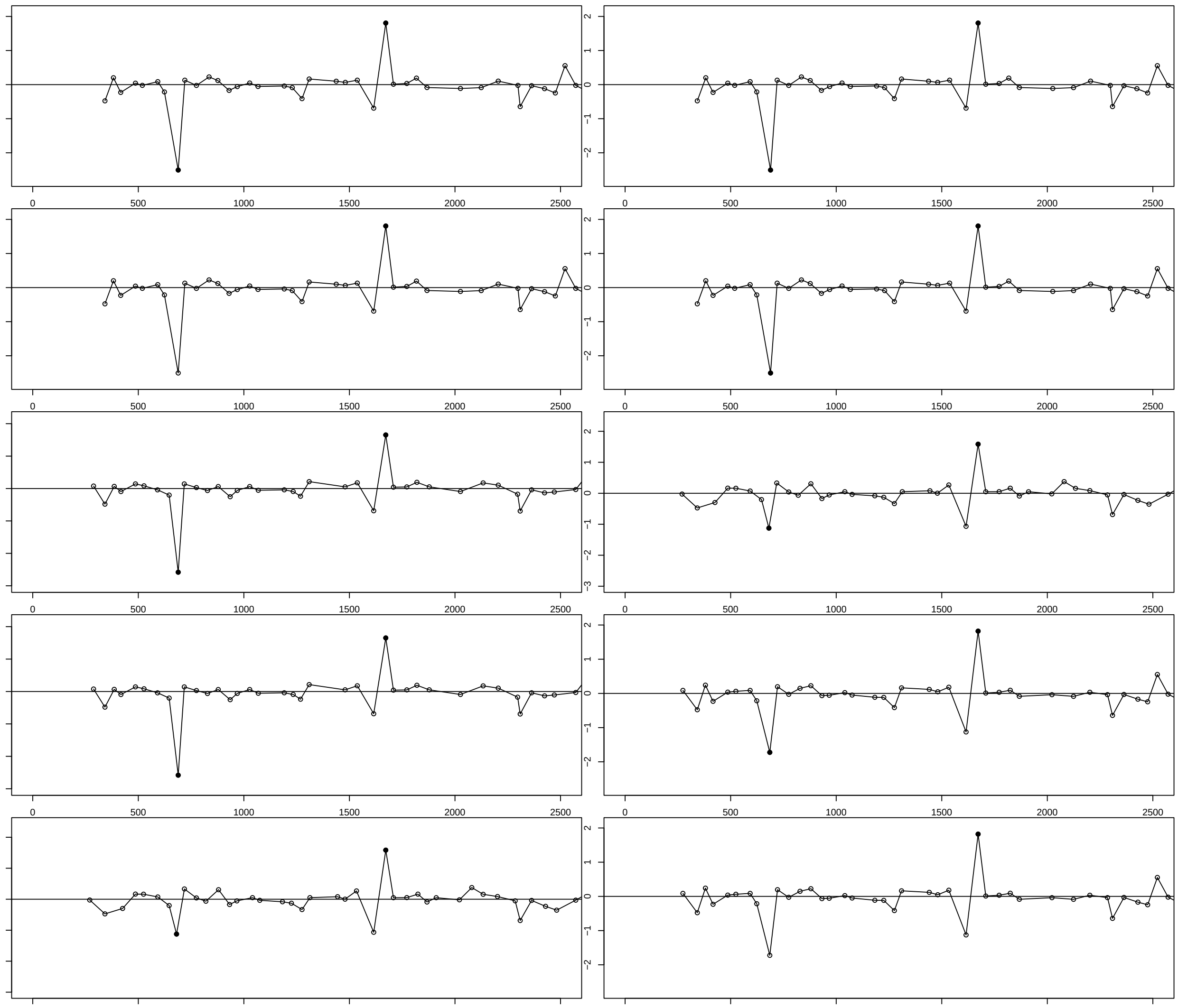
```
  r2.250.raw[i,1] <- summary(reg.sub.annual)$adj.r.squared #get the r2
}
```



Local moran's I - cold 250

```
table(carp.r$spatial.code.250)

par(mfrow = c(5, 2))
par(mar=c(1,1,1,1))

## Check the local moran's results after sub-sampling
for (i in 1:10){
  sub <- carp.r %>% group_by(spatial.code.250) %>% sample_n(size=1)
  reg.sub.cold <- lm(log(sub$AAM)~sub$ColdTemp)
  test <- correlog(sub$longitude, sub$latitude, reg.sub.cold$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,2500))
  abline(h=0)
  r2.250.raw[i,2] <- summary(reg.sub.cold)$adj.r.squared #get the r2
```

```
}
```



Subsampling at 250 km does not reduce the spatial autocorrelation in our dataset.

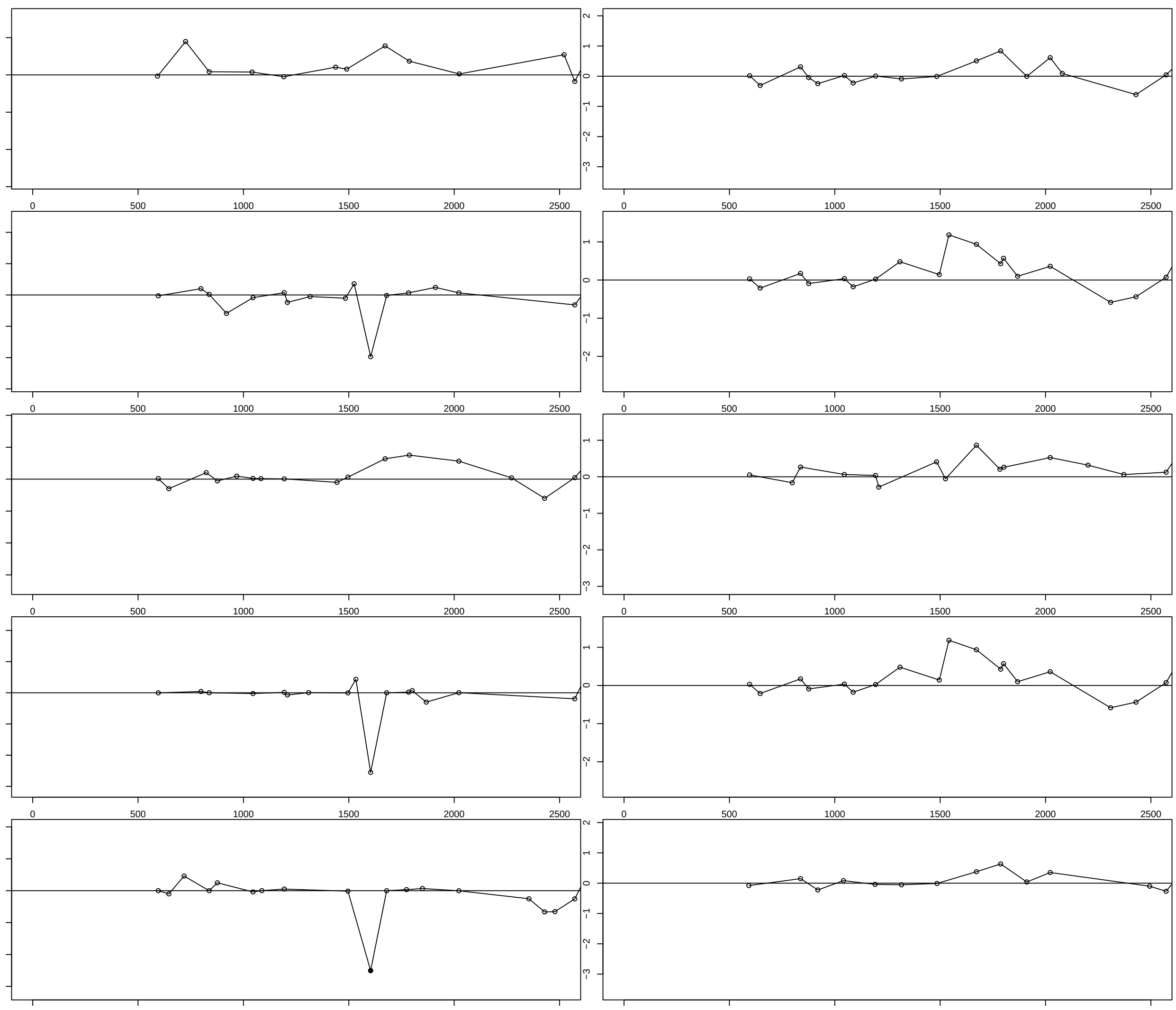
Subsampling at 550km

Now, we try to subsample at a larger distance - 550km.

Local moran's I - annual 550

```
table(carp.r$spatial.code.550)

# Store the r2 values
r2.550.raw <- matrix(NA,10, 2)
colnames(r2.550.raw) <- c("annual","cold")

par(mfrow = c(5, 2))
par(mar=c(1,1,1,1))

## Check the local moran's results after sub-sampling
```

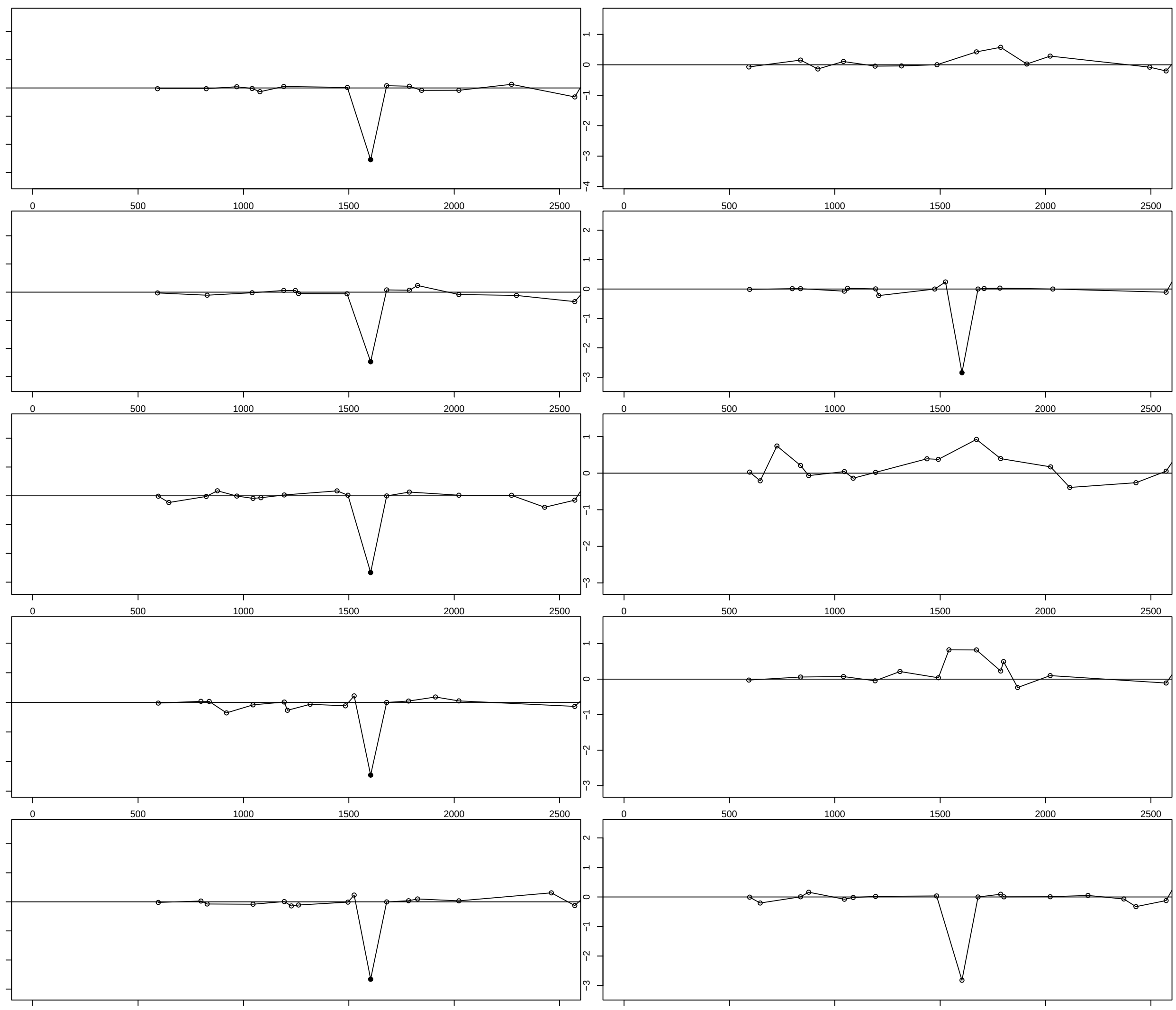
```r
for (i in 1:10){
  sub <- carp.r %>% group_by(spatial.code.550) %>% sample_n(size=1)
  reg.sub.annual <- lm(log(sub$AAM)~sub$AnnualTemp)
  test <- correlog(sub$longitude, sub$latitude, reg.sub.annual$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,2500))
  abline(h=0)
  r2.550.raw[i,1] <- summary(reg.sub.annual)$adj.r.squared #get the r2
}
```



Local moran's I - cold 550

```r
table(carp.r$spatial.code.550)

par(mfrow = c(5, 2))
par(mar=c(1,1,1,1))

## Check the local moran's results after sub-sampling
for (i in 1:10){
```

```
  sub <- carp.r %>% group_by(spatial.code.550) %>% sample_n(size=1)
  reg.sub.cold <- lm(log(sub$AAM)~sub$ColdTemp)
  test <- correlog(sub$longitude, sub$latitude, reg.sub.cold$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,2500))
  abline(h=0)
  r2.550.raw[i,2] <- summary(reg.sub.cold)$adj.r.squared #get the r2
}
```



We have seen that subsampling at 550 km would largely reduce the spatial autocorrelation in our dataset at low distances, even though we are only left with 12 data points.

Compare the r2 values

```
r2 <- data.frame(
  Time = c("Original", "Sub 250","Sub 550"),
  Annual = c(summary(lm.annual)$adj.r.squared, mean(unique(r2.250.raw[,1])),
             mean(unique(r2.550.raw[,1]))),
  Cold = c(summary(lm.cold)$adj.r.squared, mean(unique(r2.250.raw[,2])),
```

```r
           mean(unique(r2.550.raw[,2])))
)
kable(r2)
```

Time	Annual	Cold
Original	0.2541185	0.2753462
Sub 250	0.2550401	0.2766553
Sub 550	0.1127730	0.1564808

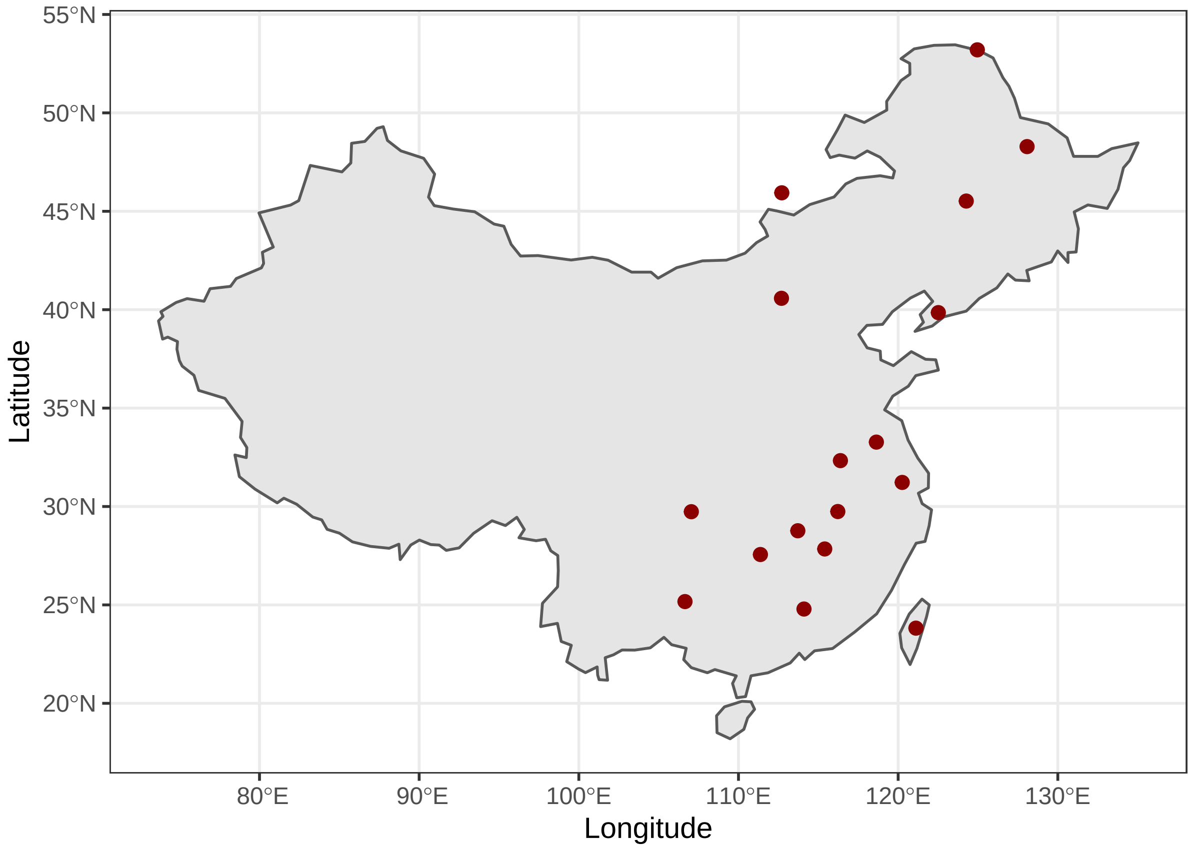
However, subsampling at 550 km would reduce our R2 value as well.

Latitudinal stratification

China Dataset analyses

Now let's look at the datapoints in China. There are 17 datapoints.

```r
## Subset the Chinese dataset
black.china <- carp.r[carp.r$china == "y",]

## Get the world data
world <- ne_countries(scale = "small", returnclass = "sf")

## Plots (china plot)
world %>%
  filter(admin == "China" | admin =="Taiwan") %>%
  # select only China
  ggplot()+
  geom_sf()+
  geom_point(aes(x=longitude, y=latitude), data = black.china,
             size = 2, color = "darkred")+
  theme_bw()+
  xlab("Longitude") + ylab("Latitude")
```



Note that the datapoint outside of the Chinese region is "Northern China". It is calculated through geographical center.

Global moran's I

```r
# Run the models
lm.annual.china <- lm(log(AAM)~AnnualTemp, data = black.china)
summary(lm.annual.china)
```

```
##
## Call:
## lm(formula = log(AAM) ~ AnnualTemp, data = black.china)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21857 -0.10327 -0.01010  0.09564  0.18908
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.998599   0.057669  34.656 9.83e-16 ***
## AnnualTemp  -0.019144   0.004173  -4.588 0.000355 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1235 on 15 degrees of freedom
## Multiple R-squared:  0.5839, Adjusted R-squared:  0.5562
```

```
## F-statistic: 21.05 on 1 and 15 DF,  p-value: 0.0003553
```

```r
lm.cold.china <- lm(log(AAM)~ColdTemp, data = black.china)
summary(lm.cold.china)
```

```
##
## Call:
## lm(formula = log(AAM) ~ ColdTemp, data = black.china)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21312 -0.04385 -0.00453  0.08316  0.18188
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.755918   0.027736   63.31  < 2e-16 ***
## ColdTemp    -0.012452   0.002372   -5.25 9.79e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1136 on 15 degrees of freedom
## Multiple R-squared:  0.6476, Adjusted R-squared:  0.6241
## F-statistic: 27.57 on 1 and 15 DF,  p-value: 9.785e-05
```

```r
## Make spatial dataframe
coords <- data.frame("long"=black.china[,14],"lat"=black.china[,13])
df <- data.frame(a = 1:nrow(black.china[14]))
spatial.data <- SpatialPointsDataFrame(coords,df,proj4string = tmin.1979@crs)

# Get a distance matrix from all points
dists <- spDists(spatial.data, longlat = TRUE)

## Run the Moran.I test on the residuals
Moran.annual <- Moran.I(lm.annual.china$residuals, dists)
Moran.cold <- Moran.I(lm.cold.china$residuals, dists)

global.moran <- data.frame(
  Model = c("Moran.annual", "Moran.cold"),
  Observed = c(Moran.annual$observed, Moran.cold$observed),
  Expected = c(Moran.annual$expected, Moran.cold$expected),
  sd = c(Moran.annual$sd, Moran.cold$sd),
  p.value = c(Moran.annual$p.value, Moran.cold$p.value)
)

kable(global.moran)
```

Model	Observed	Expected	sd	p.value
Moran.annual	-0.0523886	-0.0625	0.0381448	0.7909481
Moran.cold	-0.0474083	-0.0625	0.0379203	0.6906404

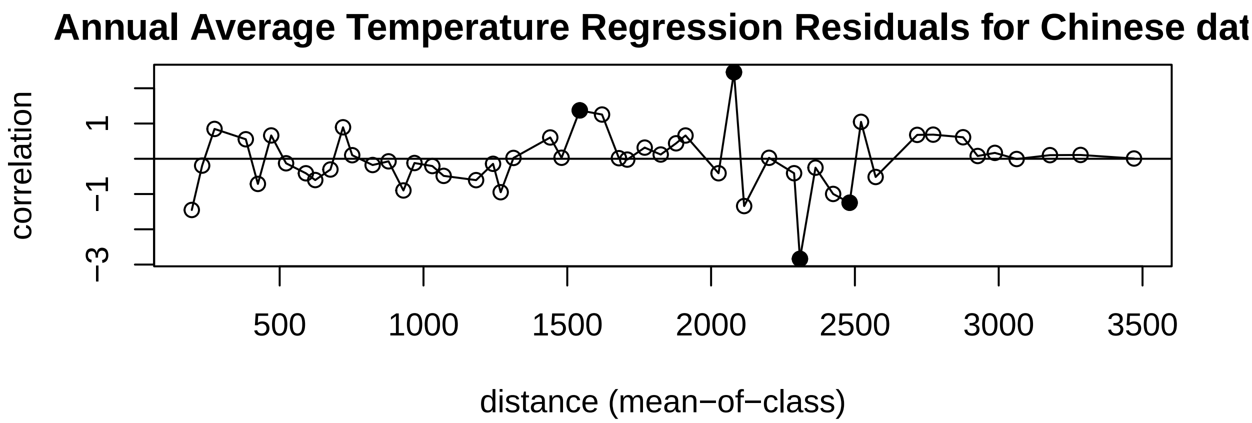
No global spatial autocorrelation in the Chinese dataset.

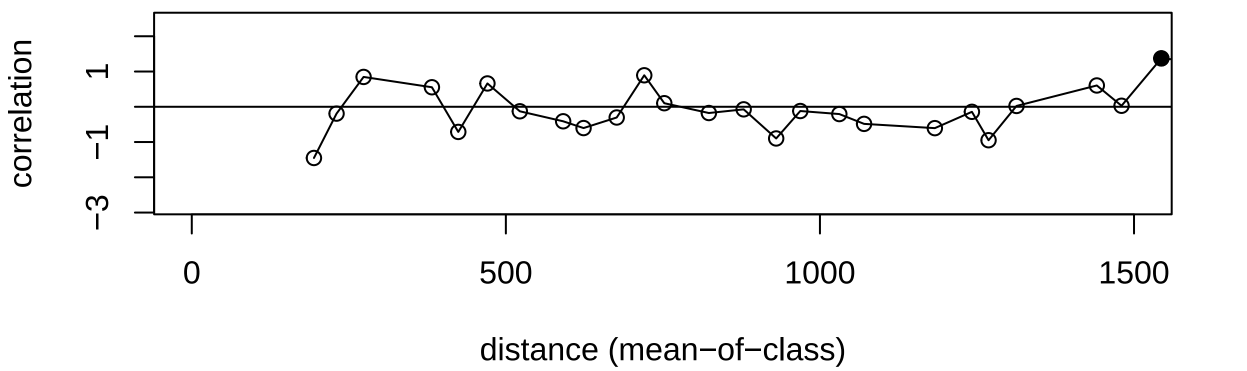
Local moran's I - correlog

```r
## Annual temperature
par(mfrow=c(2,1),mar=c(4,4,2,2))

# Use the coorelog function to develop the relationship
test <- correlog(black.china$longitude,
                 black.china$latitude, lm.annual.china$residuals,
                 increment=50, resamp=500, latlon=T)
```

```
## 50  of  500 100  of  500 150  of  500 200  of  500 250  of  500 300  of  500 350  of  500 400  of  50
```

```r
# Plot with the entire distance range
plot(test,
     main="Annual Average Temperature Regression Residuals for Chinese data")
abline(h=0)
# Reduce the distance range to 1500 km
plot(test, main="", xlim=c(0,1500))
abline(h=0)
```

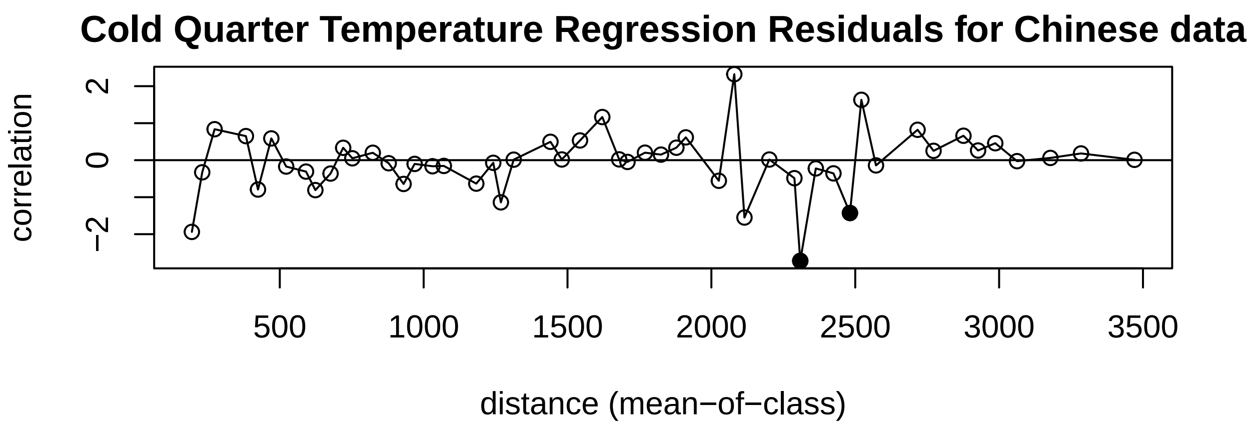


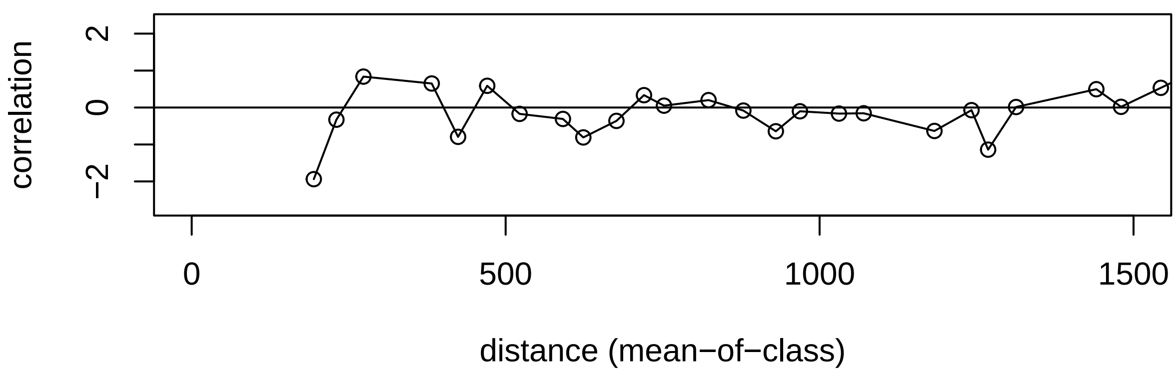



```r
## Cold temperature
par(mfrow=c(2,1),mar=c(4,4,2,2))
# Use the coorelog function to develop the relationship
test <- correlog(black.china$longitude,
                 black.china$latitude, lm.cold.china$residuals,
                 increment=50, resamp=500, latlon=T)
```

```
## 50  of  500 100  of  500 150  of  500 200  of  500 250  of  500 300  of  500 350  of  500 400  of  50
```

```r
# Plot with the entire distance range
plot(test,
     main="Cold Quarter Temperature Regression Residuals for Chinese data")
abline(h=0)

# Reduce the distance range to 1500 km
plot(test, main="", xlim=c(0,1500))
abline(h=0)
```

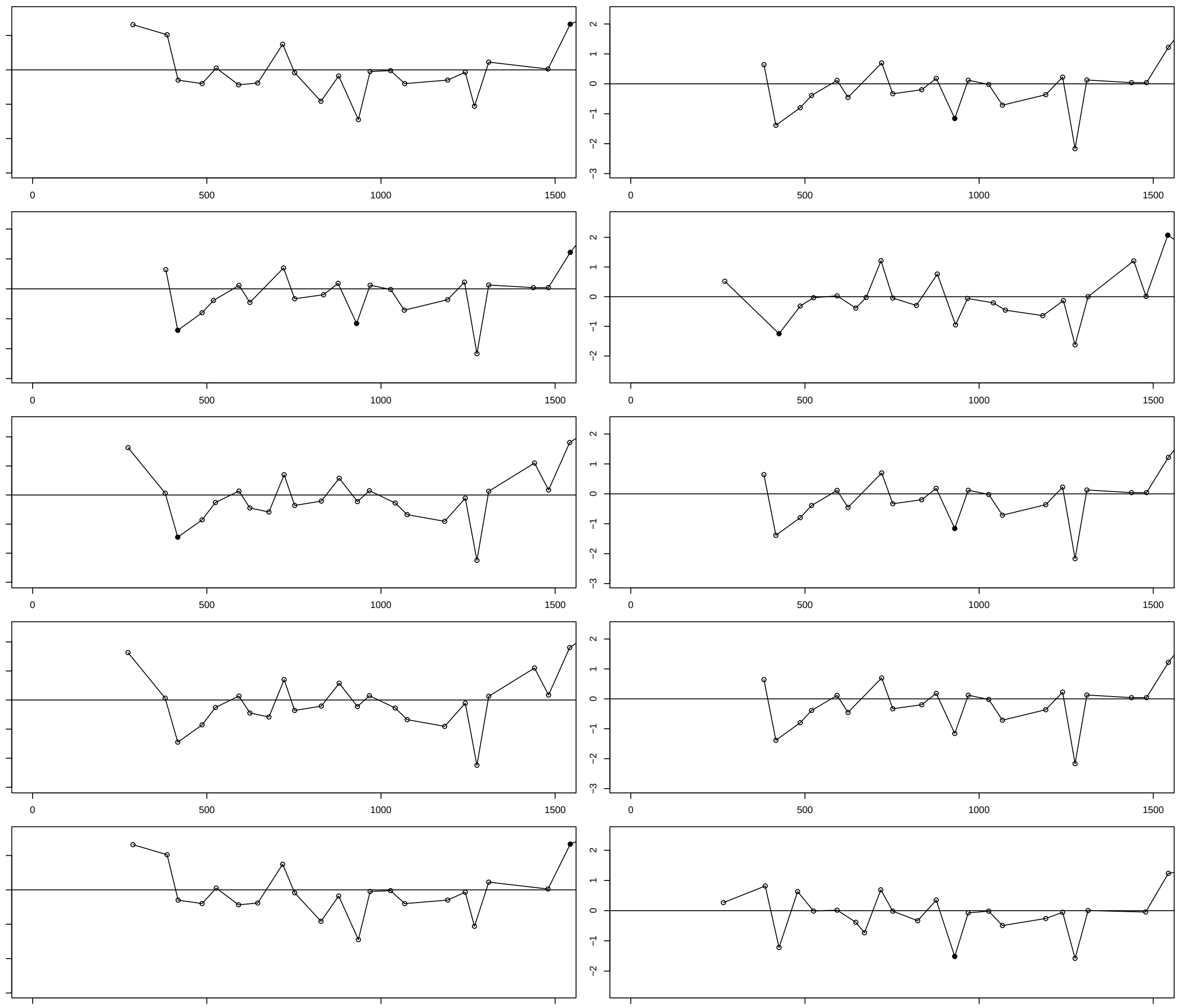




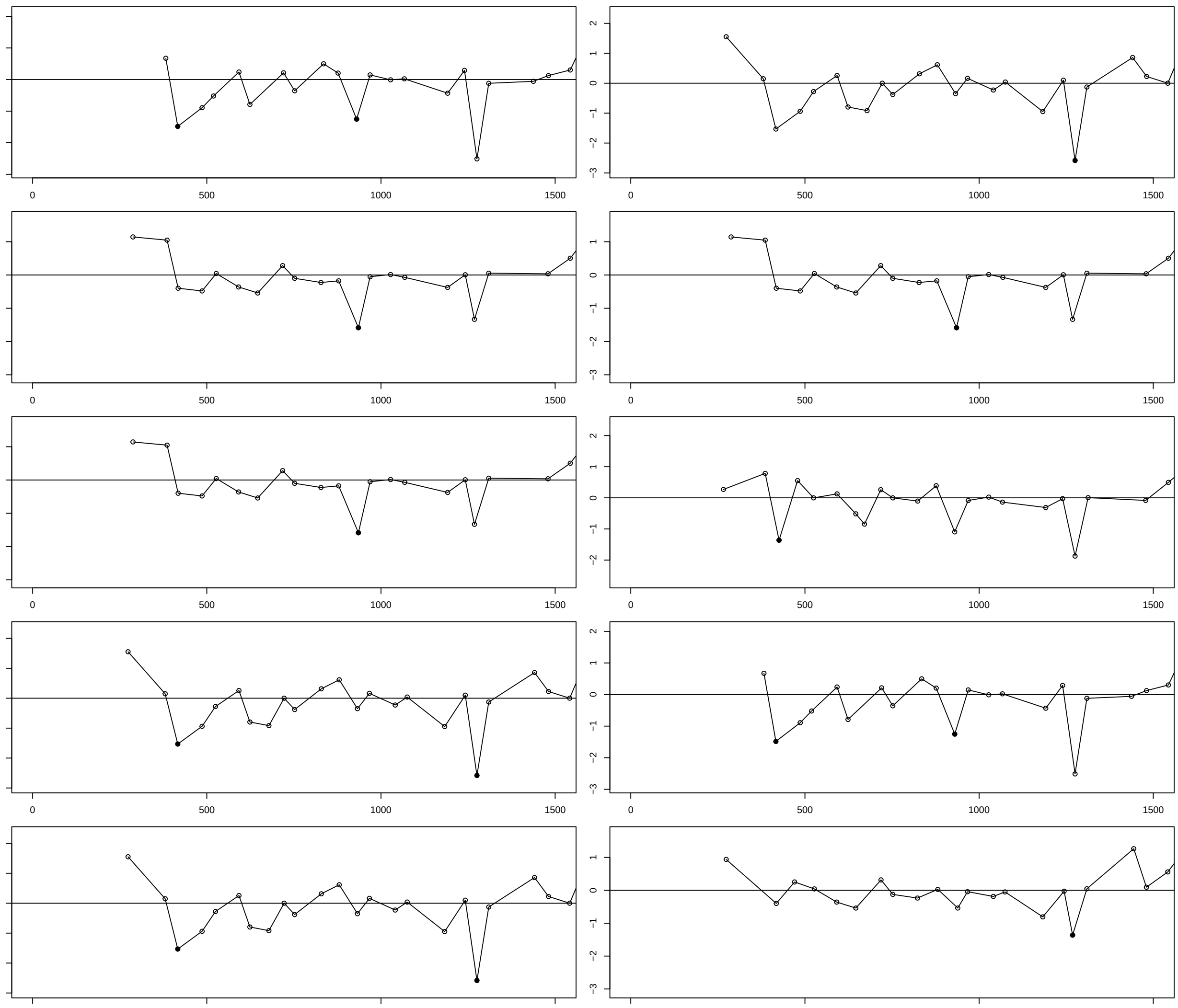
Subsample at 250km for Chinese dataset

Now we subsample at 250km to reduce spatial autocorrelation.

```r
table(black.china$spatial.code.250)

## Annual temperature
par(mfrow=c(5,2), mar=c(1,1,2,2))
for (i in 1:10){
  sub <- black.china %>% group_by(spatial.code.250) %>% sample_n(size=1)
  reg.sub.annual <- lm(log(sub$AAM)~sub$AnnualTemp)

  test <- correlog(sub$longitude, sub$latitude, reg.sub.annual$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,1500))
  abline(h=0)
}
```




```r
## Cold temperature
par(mfrow=c(5,2), mar=c(1,1,2,2))
for (i in 1:10){
  sub <- black.china %>% group_by(spatial.code.250) %>% sample_n(size=1)
  reg.sub.cold <- lm(log(sub$AAM)~sub$ColdTemp)

  test <- correlog(sub$longitude, sub$latitude, reg.sub.cold$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,1500))
  abline(h=0)
}
```


0
500
1000
1500
−3
−2
−1
0
1
2