Spatial autocorrelation

Eddie Wu

2023-08-03

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.

```
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)
```

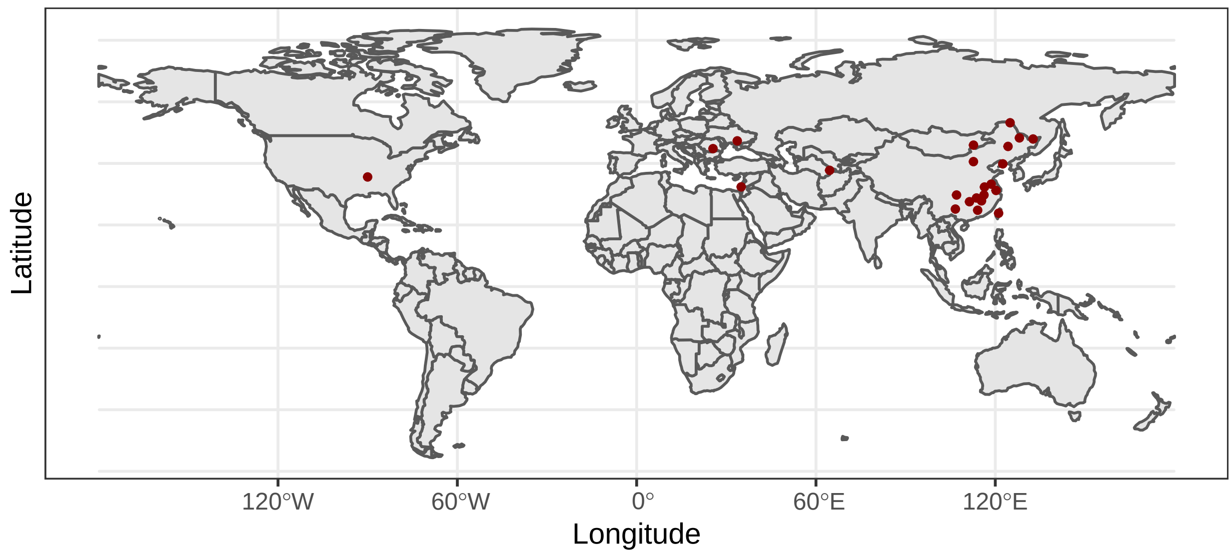
```
## Import location data
location <- read.csv("location_no_temps.csv")
location <- unique(location) # remove duplicating locations

# 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

```r
## 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

```r
# 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

# 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

```
## 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.

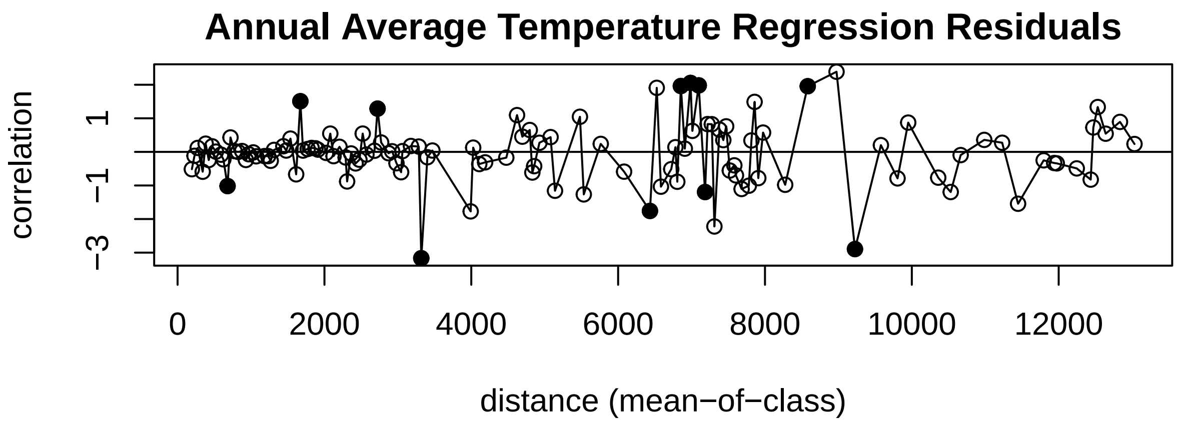
Correlog plots - local spatial autocorrelation

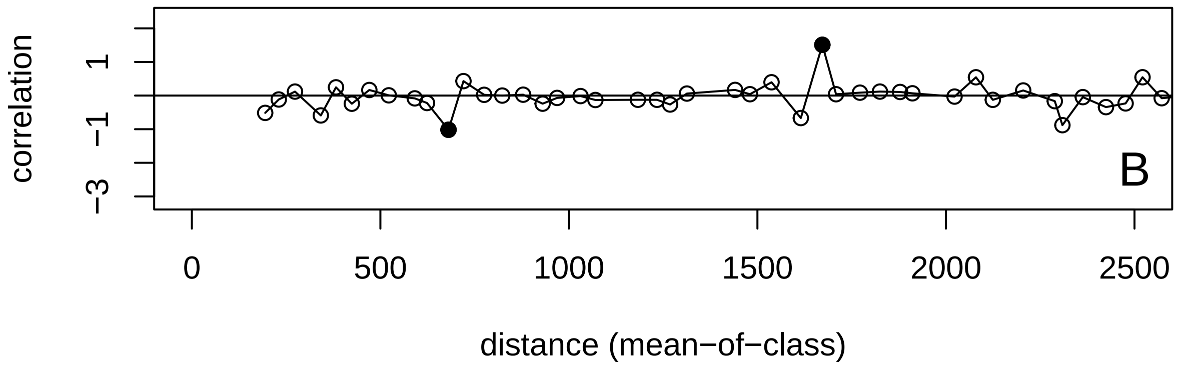
```
## Annual temperature
par(mfrow=c(2,1),mar=c(4,4,2,2))

# 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
```

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

# Reduce the distance range to 2500 km
```

```r
plot(test, main="", xlim=c(0,2500))
abline(h=0)
text(2500, min(test$correlation)+1, "B", cex=1.5)
```

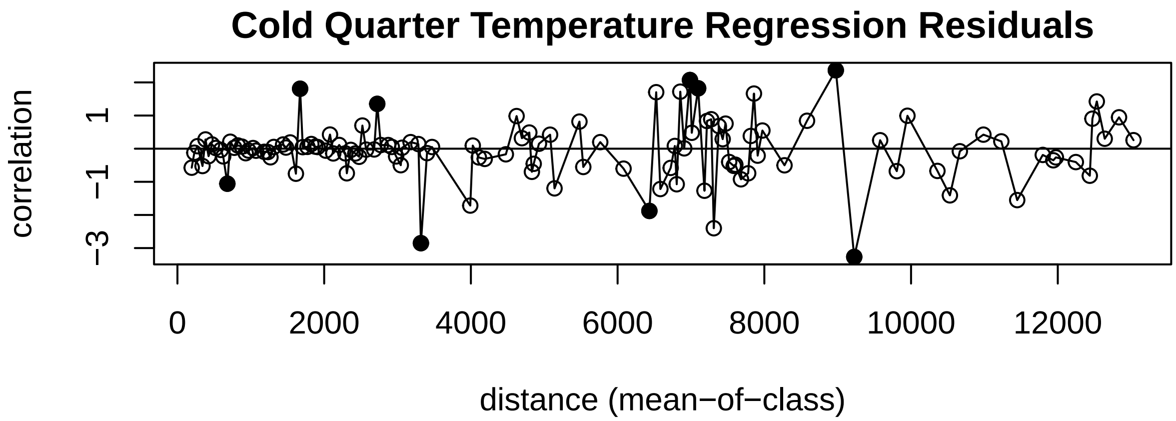


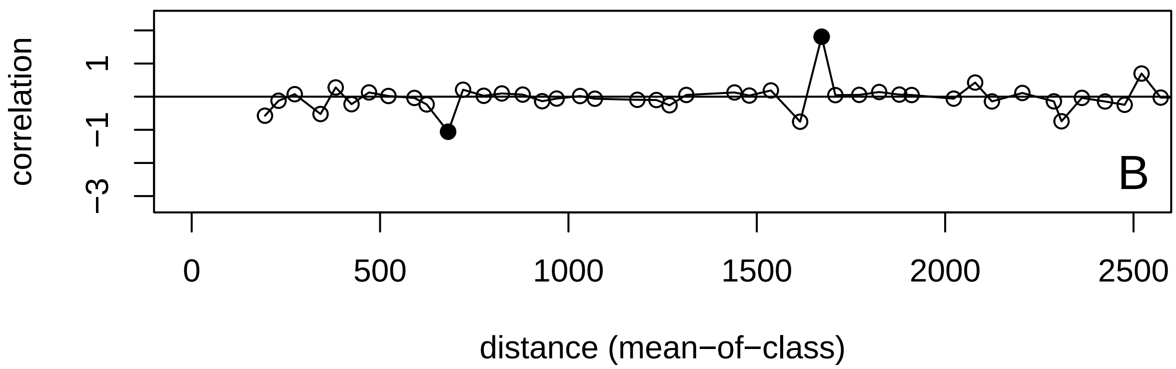



```r
## Cold temperature
par(mfrow=c(2,1),mar=c(4,4,2,2))

# 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)
```

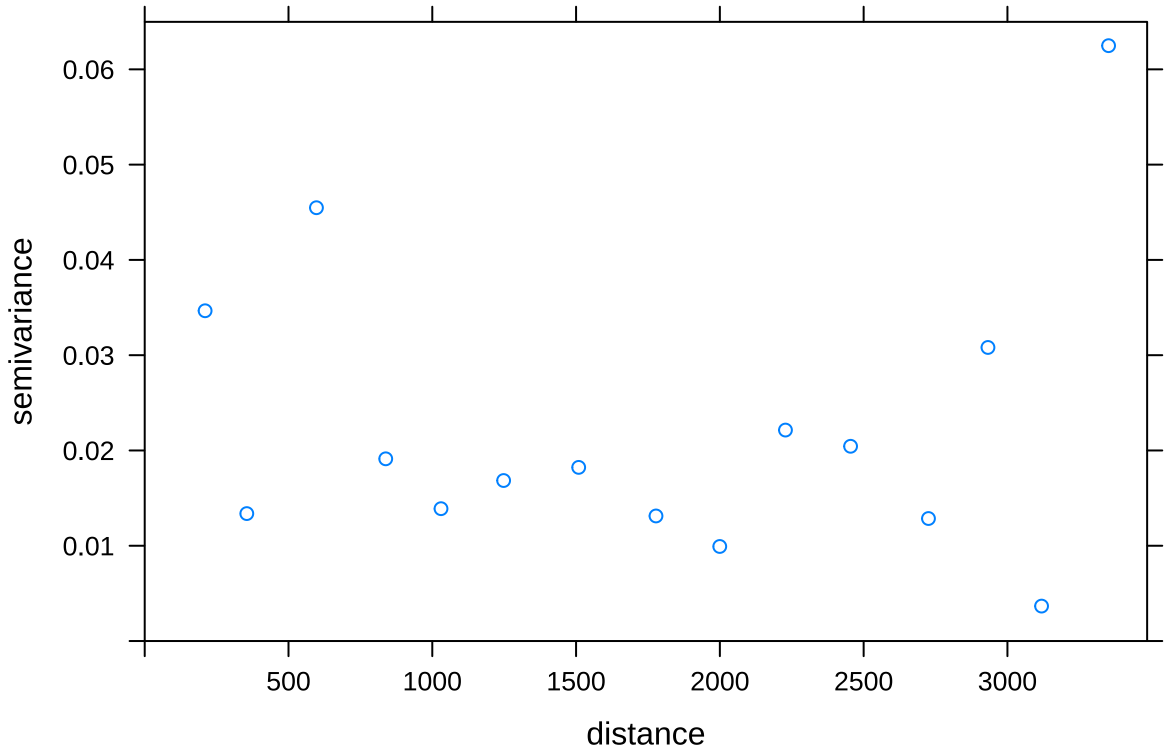




In general, the correlog plots suggest that for both temperature, 600 km is a good distance to reduce local spatial autocorrelation.

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.

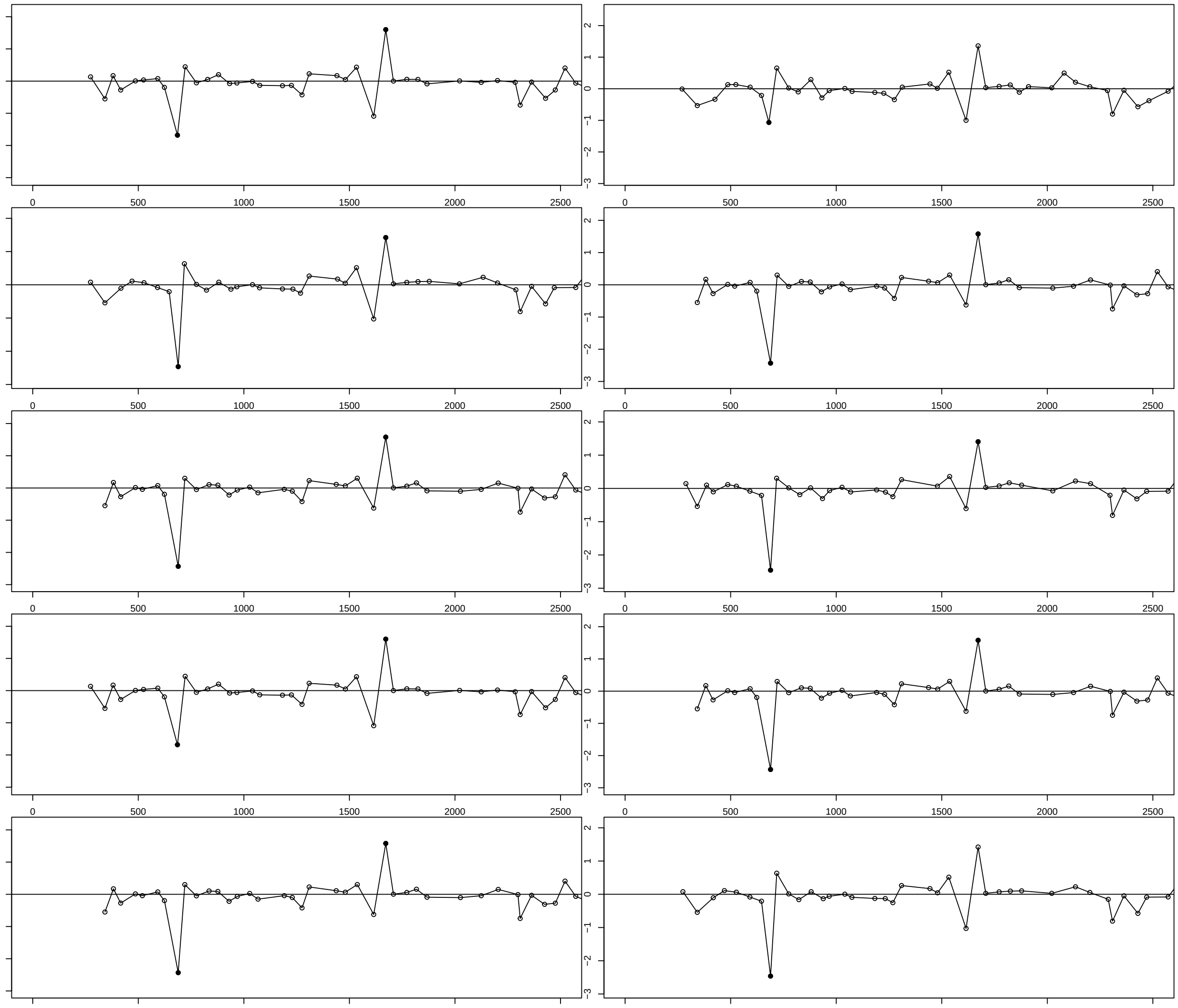
If we pick 500 km as the cutoff distance for sub-sampling, we would have only 13 location points.

Subsampling examination (at 250km)

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

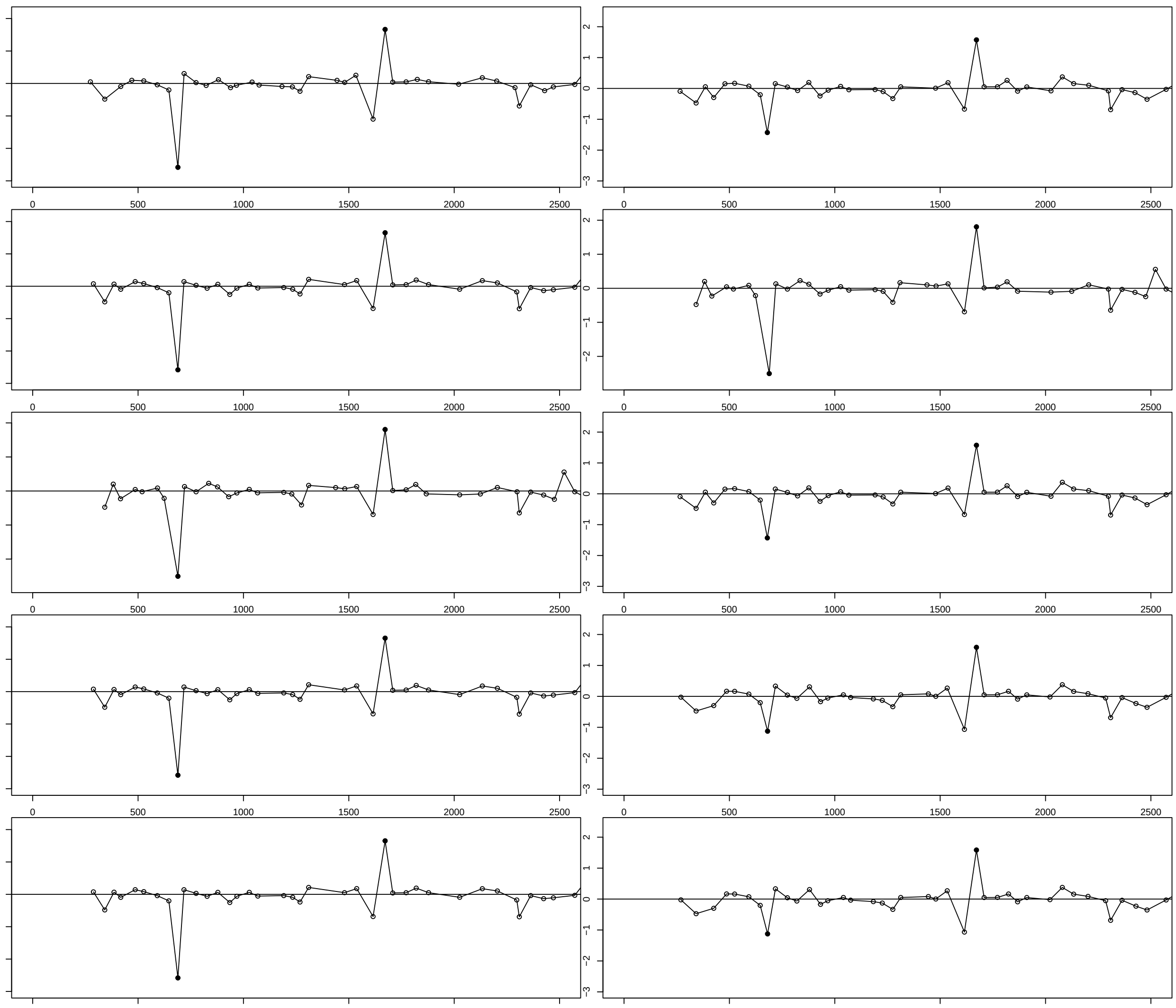
Local moran's I - annual

```
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$AAM)~sub$AnnualTemp)
  test <- correlog(sub$longtitude, sub$latitude, reg.sub.cold$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,2500))
  abline(h=0)
}
```



Local moran's I - 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.cold <- lm(log(sub$AAM)~sub$ColdTemp)
  test <- correlog(sub$longtitude, sub$latitude, reg.sub.cold$residuals,
                   increment=50, resamp=500, latlon=T)
  plot(test, main="", xlim=c(0,2500))
  abline(h=0)
}
```



Slope and intercept - annual

```
## Create a matrix to store the results over the sub-sampling
results.raw <- matrix(NA,10,4)
colnames(results.raw) <- c("slope", "intercept", "p value", "R^2")

results.final.annual <- matrix(NA,2,4)
colnames(results.final.annual) <- c("slope", "intercept", "p value", "R^2")
rownames(results.final.annual) <- c("sub-sampling at 250km", "no sub-sampling")


## Sub-sample and run regression for 10 times
for(i in 1:nrow(results.raw)) {
  # sub-sample by location
  sub <- carp.r %>% group_by(spatial.code.250) %>% sample_n(size=1)
  # run regression
  reg <- lm(log(sub$AAM)~sub$AnnualTemp)
  values <- summary(reg)
  results.raw[i,1]<-values$coef[2,1] # slope for artificial
```

```r
  results.raw[i,2]<-values$coef[1,1] # intercept for artificial
  results.raw[i,3]<-values$coef[2,4] # p value for the AAM ~ temp relationship
  results.raw[i,4]<-values$adj.r.squared # R2
}

# Take the mean of only the unique possibilities for slope, intercept, R2
results.final.annual[1,1] <- mean(unique(results.raw[,"slope"]))
results.final.annual[1,2] <- mean(unique(results.raw[,"intercept"]))
results.final.annual[1,4] <- mean(unique(results.raw[,"R^2"]))

# Count the number of times when p value is greater than 0.05
results.final.annual[1,3] <- sum(results.raw[,"p value"] > 0.05)

## Compare to without sub-sampling
results.final.annual[2,1] <- summary(lm.annual)$coef[2,1]
results.final.annual[2,2] <- summary(lm.annual)$coef[1,1]
results.final.annual[2,3] <- summary(lm.annual)$coef[2,4]
results.final.annual[2,4] <- summary(lm.annual)$adj.r.squared

results.final.annual
```

```
##                             slope intercept     p value       R^2
## sub-sampling at 250km -0.01950233  2.026544 0.000000000 0.2553525
## no sub-sampling       -0.01885274  2.027038 0.008284632 0.2541185
```

Slope and intercept - cold

```r
## Create a matrix to store the results over the sub-sampling
results.raw <- matrix(NA,10,4)
colnames(results.raw) <- c("slope", "intercept", "p value", "R^2")

results.final.cold <- matrix(NA,2,4)
colnames(results.final.cold) <- c("slope", "intercept", "p value", "R^2")
rownames(results.final.cold) <- c("sub-sampling at 250km", "no sub-sampling")

## Sub-sample and run regression for 10 times
for(i in 1:nrow(results.raw)) {
  # sub-sample by location
  sub <- carp.r %>% group_by(spatial.code.250) %>% sample_n(size=1)
  # run regression
  reg <- lm(log(sub$AAM)~sub$ColdTemp)
  values <- summary(reg)
  results.raw[i,1]<-values$coef[2,1] # slope for artificial
  results.raw[i,2]<-values$coef[1,1] # intercept for artificial
  results.raw[i,3]<-values$coef[2,4] # p value for the AAM ~ temp relationship
  results.raw[i,4]<-values$adj.r.squared # R2
}

# Take the mean of only the unique possibilities for slope, intercept, R2
results.final.cold[1,1] <- mean(unique(results.raw[,"slope"]))
results.final.cold[1,2] <- mean(unique(results.raw[,"intercept"]))
results.final.cold[1,4] <- mean(unique(results.raw[,"R^2"]))

# Count the number of times when p value is greater than 0.05
```

```r
results.final.cold[1,3] <- sum(results.raw[,"p value"] > 0.05)

## Compare to without sub-sampling
results.final.cold[2,1] <- summary(lm.cold)$coef[2,1]
results.final.cold[2,2] <- summary(lm.cold)$coef[1,1]
results.final.cold[2,3] <- summary(lm.cold)$coef[2,4]
results.final.cold[2,4] <- summary(lm.cold)$adj.r.squared

results.final.cold
```

```
##                              slope intercept     p value       R^2
## sub-sampling at 250km -0.01202992  1.785916 0.000000000 0.2723027
## no sub-sampling       -0.01190156  1.790470 0.005955222 0.2753462
```

For both annual temperature and cold temperature, the slope, intercept and R^2 value before and after subsampling is not significantly different from each other.