Lab Exercise #6

This week, we will explore spatial autocorrelation in R, including global and local moran's I, global and local getis G, Geary's C. We will use spatial neighborhood and weights constructed in Lab 5.

The dataset that will be used for this lab is the "sids2" downloaded in Lab5. This data set is for the 100 counties of North Carolina, and includes counts of numbers of live births (also non-white live births) and numbers of sudden infant deaths, for the July 1, 1974 to June 30, 1978 and July 1, 1979 to June 30, 1984 periods. Variables include BIR74 (total number of live births during 1974-1978), SID74 (numbers of sudden infant deaths during 1974-1978), NWBIR74 (number of non-white live births during 1974-1978), SIDR74 (sudden infant deaths rate during 1974-1978).

There are 10 questions with 10 points each.

```
#R libraries we'll use
install.packages("ctv")
library("ctv")
library(maptools)
install.packages("spdep")
library(spdep)
library(rgdal)
install.packages("RColorBrewer")
install.packages("classInt")
library(RColorBrewer) ## Visualization
library(classInt)     ## Class intervals
library(sp)
library(sf)
library(terra)

#import and project the shapefile
sids<-readShapePoly("path to shapefile/shpfile.shp")
proj4string(sids)<-CRS("+proj=longlat +ellps=WGS84")


#difine contiguity based neighbors
library(spdep)
sids_nbq<-poly2nb(sids)
sids_nbr<-poly2nb(sids, queen=FALSE)

#Row-standardized weights matrix
sids_nbq_w<- nb2listw(sids_nbq)
```

```
#Binary Weights
sids_nbq_wb<-nb2listw(sids_nbq, style="B")

#examine the dataset
summary (sids)

#Examine spatial autocorrelation for variable SIDR74 (sudden infant deaths rate during 1974-1978). First we need to define a new “lagged” variable which measures the same thing for each observations neighbors.  Then we will look at the relationship between these two variables.

spplot(sids, "SIDR74") #Before examining the spatial autocorrelation, we can first plot the SIDR74 from 1974-1978

sids$SIDR74_lag1 <- lag(sids_nbq_w, sids$SIDR74) #define a new lagged variable using row standardized weights
lm2 <- lm(sids$SIDR74_lag1~sids$SIDR74)
lm2
plot(y= sids$SIDR74_lag1, x= sids$SIDR74)
title("The relationship between a variable \n and the lag of itself is moran's I")
```

Q1. Take a screenshot of the plot. What is the coefficient for variable SIDR74. What does this coefficient mean?

```
#Global moran’s I test
moran.test (sids$SIDR74, sids_nbq_w, randomisation = FALSE) # Moran I test under normality
moran.test(sids$SIDR74, sids_nbq_wb) # Moran I test under randomization using binary weight
```

Q2. What are the results (pattern)? Compare their test statistic, standard deviate, p-value, and variance.

```
# Monte-Carlo simulation of Moran’s I
set.seed(1234)
permu<-moran.mc(sids$SIDR74, sids_nbq_w, nsim = 999)
permu
graph999 <- hist(permu$res,freq=TRUE,col="light blue",main="Permutation Test for Moran's I - 999 permutations")
lines(permu$statistic,max(graph999$counts),type="h",col="red",lwd=2)
```

Q3. Take a screenshot of the histogram. Explain it.

```
# For rate-based data, there is a larger variance associated with rate estimates for counties with small populations at risk compared with counties with large populations at risk. Empirical Bayes estimates place more credence on the raw rates of counties with large populations at risk, and modify them much less than they modify rates for small counties. In the case of small populations at risk, more confidence is placed in either the global rate for the study area as a
```

whole, or for local Empirical Bayes estimates, in rates for a larger moving window including the neighbors of the county being estimated. The function used for this in spdep is EBlocal().

```
set.seed(1234)
EBImoran.mc(n = sids$SID74, x = sids$BIR74, sids_nbq_wb, nsim = 999)
```

Q4. How is the result from this method different from previous results?

```
#Explore the Correlogram
cor8<-sp.correlogram(sids_nbq, sids$SIDR74, order=8, method="I")
cor8
plot(cor8)
```

Q5. Take a screen shot of your plot and explain it.

```
#Next, we will explore the local Moran’ I
#First plot the moran scatterplot
moran.plot(sids$SIDR74, sids_nbq_w)
```

Q6. Take a screenshot of your plot and explain it.

```
#LISA.
locm <- localmoran(sids$SIDR74, sids_nbq_w) #Calculate the local Moran’s I
locm
summary(locm)
sids$sSIDR74<-scale(sids$SIDR74) #Standardize it
sids$lag_sSIDR74<-lag(sids_nbq_w, sids$sSIDR74) #Define a lag variable
sids$sSIDR74
sids$lag_sSIDR74
sids$quad_sig <- NA  #Define a new variable quad_sig and set the initial value to be NA
#Next define five types of results: High-High, Low-Low, High-Low, Low-High, and not significant
sids@data[(sids$sSIDR74>= 0 & sids$lag_sSIDR74>= 0) & (locm[,5]<= 0.05),"quad_sig"] <- 1
sids@data[(sids$sSIDR74<= 0 & sids$lag_sSIDR74<= 0) & (locm[,5]<= 0.05),"quad_sig"] <- 2
sids@data[(sids$sSIDR74>= 0 & sids$lag_sSIDR74<= 0) & (locm[,5]<= 0.05),"quad_sig"] <- 3
sids@data[(sids$sSIDR74<= 0 & sids$lag_sSIDR74>= 0) & (locm[,5]<= 0.05),"quad_sig"] <- 4
sids@data[locm[,5]> 0.05, "quad_sig"] <- 5
```

```
#Plot the LISA map
breaks <- seq(1, 5, 1)
labels <- c("High-High", "Low-Low", "High-Low", "Low-High", "Not Signif.")
np <- findInterval(sids$quad_sig, breaks)
colors <- c("red", "blue", "lightpink", "skyblue2", "white")
plot(sids, col = colors[np])  #colors[np] manually sets the color for each county
mtext("Local Moran's I", cex = 1.5, side = 3, line = 1)
```

```
legend("topleft", legend = labels, fill = colors, bty = "n")
```

Q7. Take a screenshot of your plot and explain the pattern.

Next explore the Getis G

```
globalG.test(sids$SIDR74, sids_nbq_wb)
```

Q8. What are your results and explain the pattern (clustered? Hotspot or cold spot?)

```
#Local Getis G

G <- localG(sids$SIDR74, sids_nbq_wb)
par.G <- par(mfrow=c(2,1))
# set the graphics parameters to display to maps on top of each other
#    in the same window
brks <- seq(-4,6,1)
# creates 10 intervals between -4 and +5
# summary(G) the variable to be plotted has a low of -3.6 and high of 5.7
cm.col <- cm.colors(length(brks)-1)
# picks colors corresponding for each interval out of the "cm" color palette
plot(sids,border="lightgray",col=cm.col[findInterval(G,brks,all.inside=TRUE)])
# maps the shape file and fills each polygon with a color corresponding to
# the level of G at that location
text(coordinates(sids)[,1],coordinates(sids)[,2], round(G, digits=1), cex=0.5)
# writes the G-statistic in each polygon
title("Getis-Ord G statistic for sids")
```

Q9. Take a screenshot of your results. Is it different from the LISA map in Q7? Why?

#Geary’s C

```
library(spdep)
geary.test(sids$SIDR74, sids_nbq_wb, randomisation=FALSE)
gc_perm <- geary.mc(sids$SIDR74, sids_nbq_wb, nsim=99)
gc_perm
```

Q10. What is the pattern? Are these results different? Explain the differences if any.