Shane Clement
March 4, 2023

Project 4: Programming with R and GeoDa

The purpose of this project is to demonstrate spatial autocorrelation through the implementation of spatial weights, Moran's I, and Local Indicators of Spatial Autocorrelation (LISA).

```
setwd <-("C:/R Files/Unit 4/calenv4")

calenv <-st_read("C:/R Files/Unit4/calenv4/calenv4.shp")


## Reading layer `calenv4' from data source
##   `C:\\UCLA\MAGIST\GEOG 413\R Files\Unit 4\calenv4\calenv4.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 8035 features and 67 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -124.4096 ymin: 32.53416 xmax: -114.1344 ymax:
42.00952
## Geodetic CRS:  WGS 84
```

Group the data by county and summarize some of the variables:

```
calenvcty <-calenv %>% group_by(County) %>% summarize(unemploy =
mean(Unemployme), ciscore_p = mean(CIscoreP) , avg_poverty_p =
mean(Poverty_Pc), geometry = st_union(geometry))

calmap <-as(calenvcty,"Spatial")
```

Fix errors in the geometry

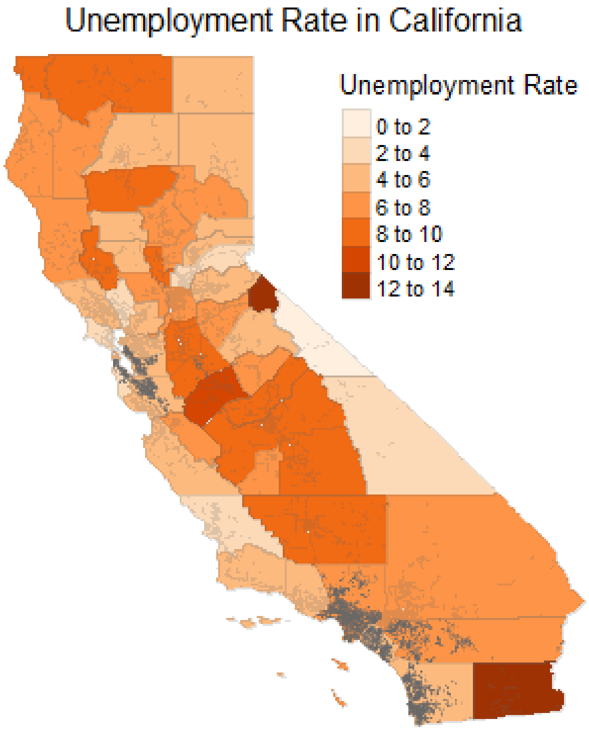
```
calmap_sf <- st_as_sf(calmap)

# check if the object is valid
# make the object valid
calmap_sf_valid <- st_make_valid(calmap_sf)

# check if the object is valid now
st_is_valid(calmap_sf_valid)
```

Reassign the variable:

```
calmap1 <-calmap_sf_valid
```

Generate a plot of the study region.

```
tm_shape(calmap1) + tm_fill("unemploy", palette = 'Oranges',
title='Unemployment Rate') + tm_layout(legend.outside = FALSE, frame = FALSE,
main.title = "Unemployment Rate in California", main.title.position =
"center", main.title.size = 1)+tm_borders(alpha=.2)
```



```
summary(calmap1)

##     County             unemploy        ciscore_p     avg_poverty_p
##  Length:58          Min.   : 0.100   Min.   :13.92   Min.   :22.70
##  Class :character   1st Qu.: 4.654   1st Qu.:28.90   1st Qu.:43.61
##  Mode  :character   Median : 6.010   Median :36.69   Median :56.71
##                     Mean   : 6.369   Mean   :40.80   Mean   :54.48
##                     3rd Qu.: 7.915   3rd Qu.:51.10   3rd Qu.:67.85
##                     Max.   :13.577   Max.   :79.74   Max.   :79.11
##           geometry
##  MULTIPOLYGON :58
##  epsg:4326    : 0
##  +proj=long...: 0
##
##
##
```

Use the functions within spdep to generate queen and rook contiguity neighbors (order 1) for the study region. Provide summary statistics for rook and queen neighbor objects and plot two neighbor maps.

Generate a set of queen contiguity neighbors for each county in California and provide a summary:

```
cawm_q <-poly2nb(calmap1, queen=TRUE)
summary(cawm_q)

## Neighbour list object:
## Number of regions: 58
## Number of nonzero links: 270
## Percentage nonzero weights: 8.026159
## Average number of links: 4.655172
## Link number distribution:
##
##   2  3  4  5  6  7  8
##   6  5 14 17 12  2  2
## 6 least connected regions:
## 8 13 21 25 38 42 with 2 links
## 2 most connected regions:
## 10 34 with 8 links
```

Find the neighbor of the county:

```
cawm_q[[1]]

## [1]  7 39 43 50
```

Find the name of the county:

```
calmap1$County[1]

## [1] "Alameda"
```

Find the neighbors of the zip codes:

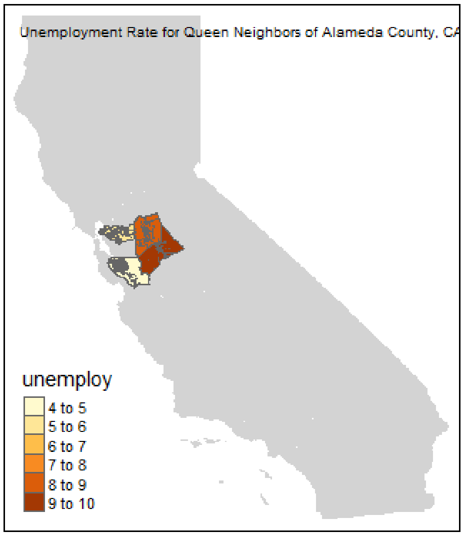
```
calmap1$County[c(7,39,43,50)]

## [1] "Contra Costa" "San Joaquin"  "Santa Clara"  "Stanislaus"
```

Subset the neighboring zip codes:

```
calmap1_asheq <-subset(calmap1, calmap1$County == 'San
Joaquin'|calmap1$County == 'Santa Clara'|calmap1$County == 'Contra
Costa'|calmap1$County == 'Stanislaus')
```

Map of queen neighbors:

```
tm_shape(calmap1) + tm_fill("lightgrey") + tm_shape(calmap1_asheq, edgecolor
= 'black', edgeborder = 4)  + tm_fill("unemploy") + tm_legend(show=TRUE) +
tm_borders() + tm_layout(title = "Unemployment Rate for Queen Neighbors of
Alameda County, CA",legend.position = c("left", "bottom"))
```



Generate rook contiguity neighbors and summarize:

```
cawm_r <- poly2nb(calmap1, queen=FALSE)
summary(cawm_r)

## Neighbour list object:
## Number of regions: 58
## Number of nonzero links: 264
## Percentage nonzero weights: 7.8478
## Average number of links: 4.551724
## Link number distribution:
##
##  2  3  4  5  6  7  8
##  6  8 12 17 12  1  2
## 6 least connected regions:
## 8 13 21 25 38 42 with 2 links
## 2 most connected regions:
## 10 34 with 8 links
```

Find rook neighbors:

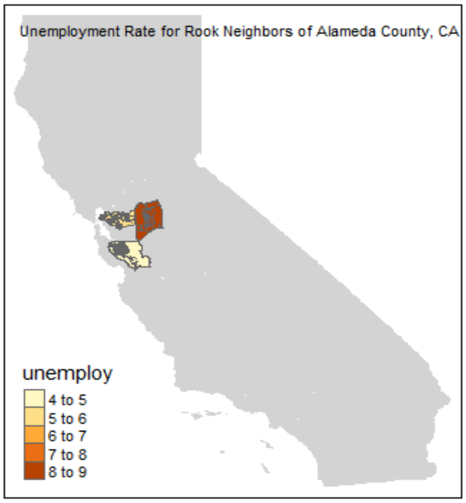
```
cawm_r[[1]]

## [1]  7 39 43

calmap1$County[c(7,39,43)]

## [1] "Contra Costa" "San Joaquin"  "Santa Clara"
```

Subset the rook neighboring zip codes:

```
calmap1_asher <-subset(calmap1, calmap1$County == 'San
Joaquin'|calmap1$County == 'Santa Clara'|calmap1$County == 'Contra Costa')
```

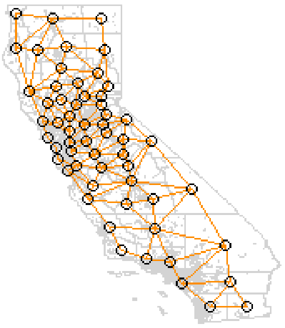
Map of rook neighbors with unemployment variable:

```
tm_shape(calmap1) + tm_fill("lightgrey") + tm_shape(calmap1_asher, edgecolor
= 'black', edgeborder = 4)  + tm_fill("unemploy") + tm_legend(show=TRUE) +
tm_borders() + tm_layout(title = "Unemployment Rate for Rook Neighbors of
Alameda County, CA",legend.position = c("left", "bottom"))
```



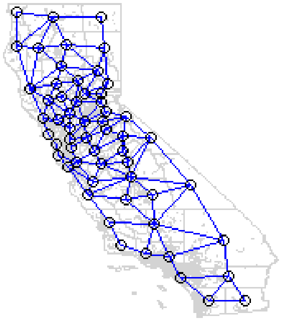
Plot the queen contiguity neighbors:

```
plot(calmap, border="lightgrey")
plot(cawm_q, coordinates(calmap),add=TRUE, col="darkorange")
```



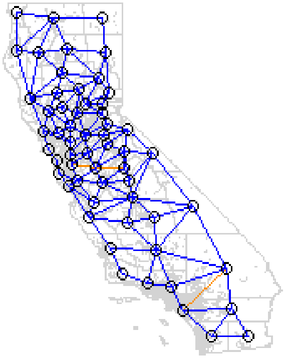
Plot rook contiguity neighbors:

```
plot(calmap, border="lightgrey")
plot(cawm_r,coordinates(calmap), add=TRUE, col='blue')
```



Plot overlay:

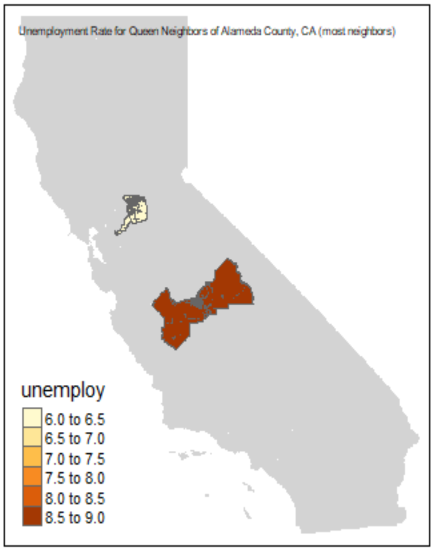
```
plot(calmap, border="lightgrey")
plot(cawm_q, coordinates(calmap),add=TRUE, col="darkorange")
plot(cawm_r,coordinates(calmap), add=TRUE, col='blue')
```



Examine the queen contiguity with the most and least connected regions:

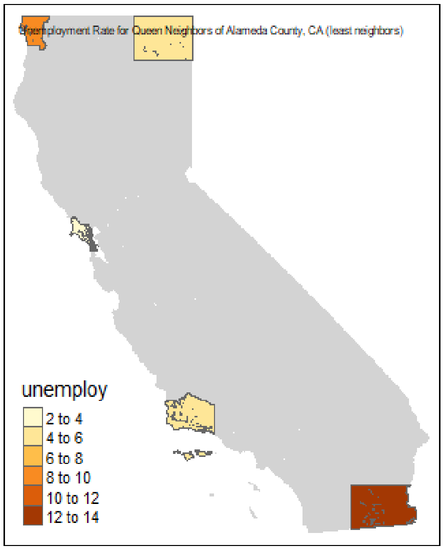
Subset of the two most connected regions:

```
calmap1$County[c(10,34)]

## [1] "Fresno"     "Sacramento"

calmap1_asheqM <-subset(calmap1, calmap1$County == 'Fresno'|calmap1$County ==
'Sacramento')

tm_shape(calmap1) + tm_fill("lightgrey") + tm_shape(calmap1_asheqM, edgecolor
= 'black', edgeborder = 4)  + tm_fill("unemploy") + tm_legend(show=TRUE) +
tm_borders() + tm_layout(title = "Unemployment Rate for Queen Neighbors of
Alameda County, CA (most neighbors)",legend.position = c("left", "bottom"))
```


Unemployment Rate for Queen Neighbors of Alameda County, CA (most neighbors)
unemploy
6.0 to 6.5
6.5 to 7.0
7.0 to 7.5
7.5 to 8.0
8.0 to 8.5
8.5 to 9.0

Queen contiguity with least connections: 8 13 21 25 38 42

```
calmap1$County[c(8,13,21,25,38,42)]

## [1] "Del Norte"     "Imperial"      "Marin"         "Modoc"
## [5] "San Francisco" "Santa Barbara"

calmap1_asheqL <-subset(calmap1, calmap1$County == 'Del Norte'|calmap1$County
== 'Imperial'|calmap1$County == 'Marin'|calmap1$County ==
'Modoc'|calmap1$County == 'San Francisco'|calmap1$County == 'Santa Barbara')

tm_shape(calmap1) + tm_fill("lightgrey") + tm_shape(calmap1_asheqL, edgecolor
= 'black', edgeborder = 4)  + tm_fill("unemploy") + tm_legend(show=TRUE) +
tm_borders() + tm_layout(title = "Unemployment Rate for Queen Neighbors of
Alameda County, CA (least neighbors)",legend.position = c("left", "bottom"))
```


Unemployment Rate for Queen Neighbors of Alameda County, CA (least neighbors)
unemploy
2 to 4
4 to 6
6 to 8
8 to 10
10 to 12
12 to 14

Generate and plot the first-order nearest neighbors for the study region.

To find the distance-based neighbors, start with finding the coordinates:

```
coords <-coordinates(calmap)
head(coords)

##        [,1]     [,2]
## 1 -121.8886 37.64618
## 2 -119.8207 38.59719
## 3 -120.6511 38.44639
## 4 -121.6007 39.66693
## 5 -120.5541 38.20461
## 6 -122.2370 39.17757
```

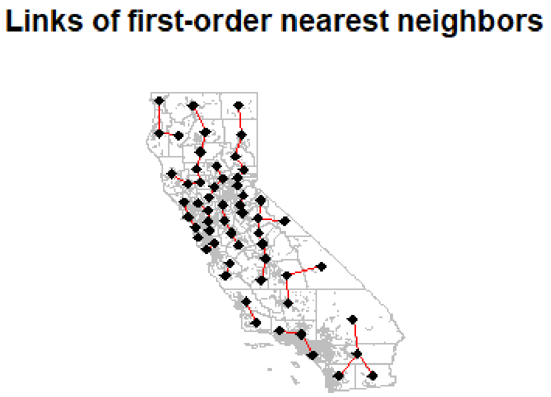
Find nearest neighbor:

```
k1 <-knn2nb(knearneigh(coords))
k1dists <-unlist(nbdists(k1, coords,longlat = TRUE))
summary(k1dists)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   26.91   42.28   49.40   60.98   81.03  128.83
```

The largest first-order nearest neighbor distance is 128.83. Therefore, the upper bound can be set to 129.

Plot links of first-order nearest neighbors:

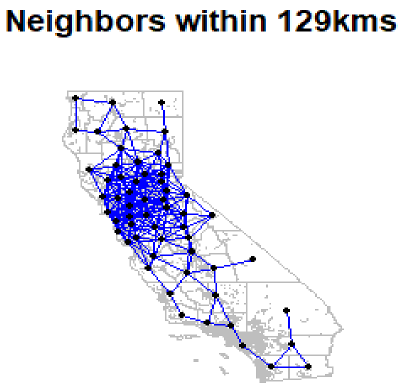
```
par(fig=c(0, 1, 0, 1))
plot(calmap, border = "grey")
plot(k1, coords, add=TRUE, col = "red",pch=18)
title(main = "Links of first-order nearest neighbors", cex=.6)
```



Links of first-order nearest neighbors

Plot links of neighbors at max distance:

```
cawm_d129 <-dnearneigh(coords, 0, 129, longlat=TRUE)
plot(calmap, border = "grey")
plot(cawm_d129, coords, add=TRUE, col='blue',pch=19, cex=.6)
title(main="Neighbors within 129kms", cex=.6)
```



Neighbors within 129kms

To find the full set of inverse distance weights between all pairs of counties, first generate the list of possible neighbors with a large upper bound:

```
cawm_d1000 <-dnearneigh(coords, 0, 1000, longlat=TRUE)
cawm_d1000

## Neighbour list object:
## Number of regions: 58
## Number of nonzero links: 3272
## Percentage nonzero weights: 97.26516
## Average number of links: 56.41379
```

Find the inverse distance weights:

```
dist <-nbdists(cawm_d1000, coords, longlat=TRUE)
idw <-lapply(dist,function(x)1/(x))
idw[1]
```

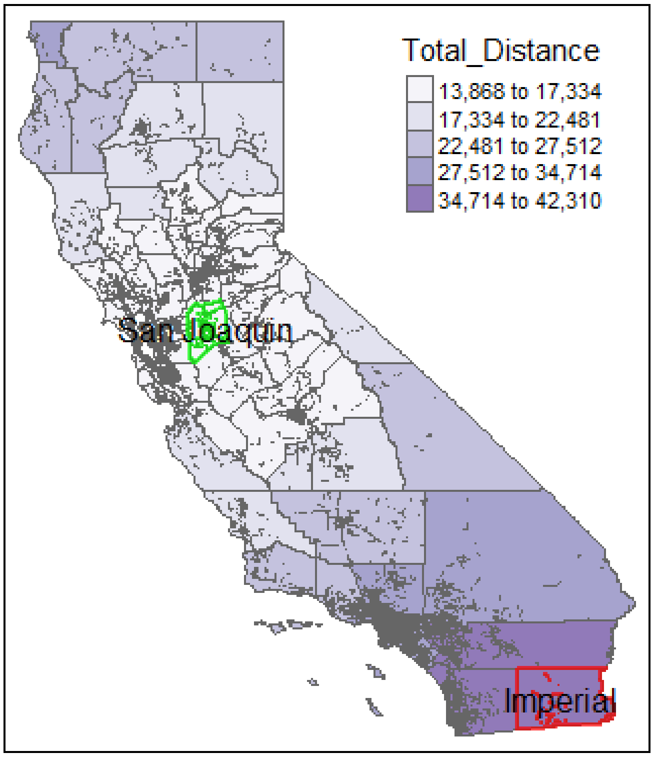
Queen contiguity neighbors of first six counties:

```
head(cawm_q)

## [[1]]
## [1]  7 39 43 50
##
## [[2]]
## [1]  3  5  9 26 55
##
## [[3]]
## [1]  2  5  9 34 39
##
## [[4]]
## [1]  6 11 32 51 52 58
##
## [[5]]
## [1]  2  3 39 50 55
##
## [[6]]
## [1]  4 11 17 51 57
```

Generate a weight for each neighbor to tell how much each neighbor might influence the area. Create weights for queen contiguity neighbors:

```
rscawm_q <-nb2listw(cawm_q, style='W', zero.policy = TRUE)

coords <-coordinates(calmap)
dists <-spDists(coords, longlat =TRUE) %>% rowSums(.)
dists

min_d <- min(dists)
max_d <- max(dists)
min_i <- which(dists == min_d)
```

```
max_i <- which(dists == max_d)

cal_data <- as.data.frame(calmap1)
access_cty <-cal_data[min_i, c("County")]
remote_cty <-cal_data[max_i, c("County")]
access_cty

## [1] "San Joaquin"

remote_cty

## [1] "Imperial"

for (i in 1: length(dists)){
  calmap1[i, c("Total_Distance")] <-dists[i]
}

tm_shape(calmap1)+
  tm_fill(col = "Total_Distance", n=5, palette = "Purples", alpha = .7, style
= "fisher")+
  tm_borders()+
  tm_shape(calmap1[min_i, ]) +
  tm_borders(col='green', lwd=2, alpha =.7)+
  tm_shape(calmap1[max_i, ]) +
  tm_borders(col='red', lwd=2, alpha = .7)+
  tm_shape(calmap1[c(max_i,min_i),]) +
  tm_text("County", size=1, col='black')
```

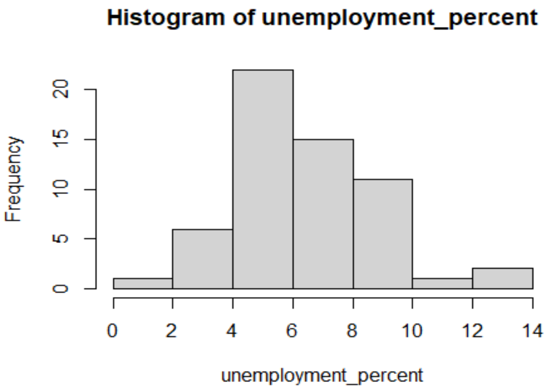


Selecting a variable of interest and mapping the variable with tmap.

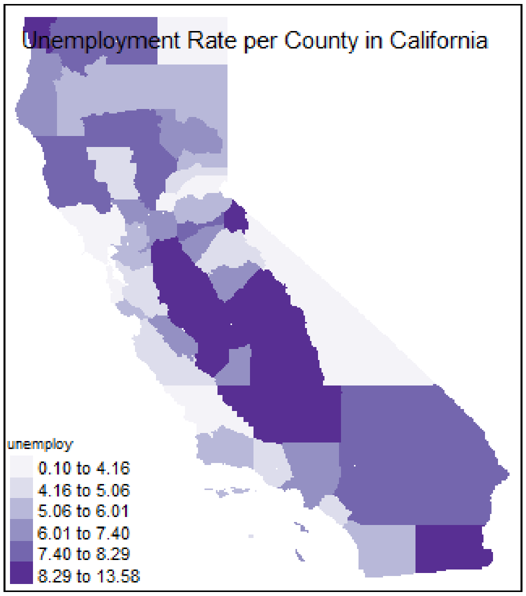
```
unemployment_percent <-calmap1$unemploy
summary(unemployment_percent)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.100   4.654   6.010   6.369   7.915  13.577
```

The mean unemployment for the California counties is 6.39 percent.

```
hist(unemployment_percent)
```



```
tm_shape(calmap1)+tm_fill("unemploy", style="quantile", n=6,palette =
"Purples")+tm_layout(legend.position =
c("LEFT","BOTTOM"),title=("Unemployment Rate per County in California"),
legend.title.size = .65)
```



Counties in middle of California with the dark purple have a higher percentage of unemployment compared to counties along the coastline. These high unemployment areas tend to be clustered together, but also some low/high unemployment areas in other areas of California.

Using the queen contiguity weights, generate a global Moran's I scatterplot for the variable of interest. Show and interpret the scatterplot.

Standard Deviation of Unemployment:

```
sd(calmap1$unemploy)

## [1] 2.442868
```

Mean of Unemployment:

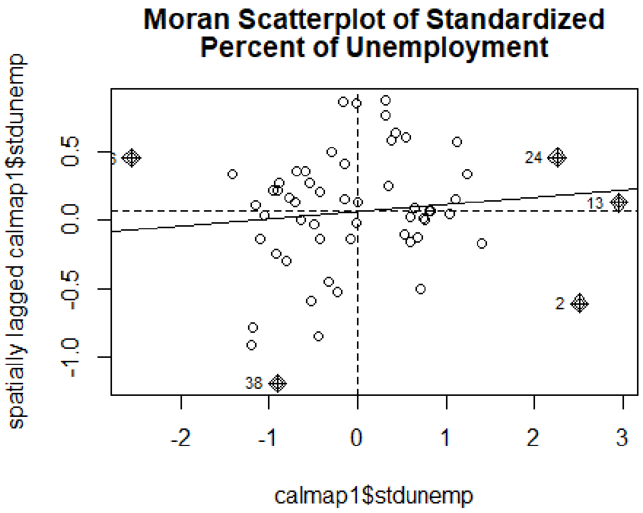
```
mean(calmap1$unemploy)

## [1] 6.369119
```

Create standardized values:

```
calmap1$stdunemp <-(calmap1$unemploy -
mean(calmap$unemploy))/sd(calmap1$unemploy)
```

Create a Moran scatterplot with the standardized values and the spatial lag values of the unemployment variable:

```
moran.plot(calmap1$stdunemp, listw = rscawm_q, main=c("Moran Scatterplot of
Standardized", "Percent of Unemployment"))
```



The scatterplot shows that there is a minimal positive spatial autocorrelation. There are 5 outliers that are affecting the slope coefficient.

Perform a test of the hypothesis that the spatial distribution of the values of the variable of interest is random.

State the hypothesis:

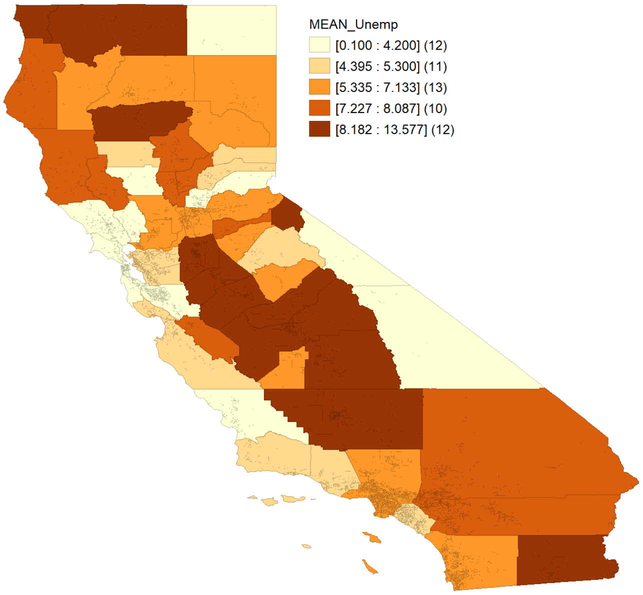
H-null: The percentage of unemployment is randomly distributed in California.

H-alternative: The percentage of unemployment is not randomly distributed in California

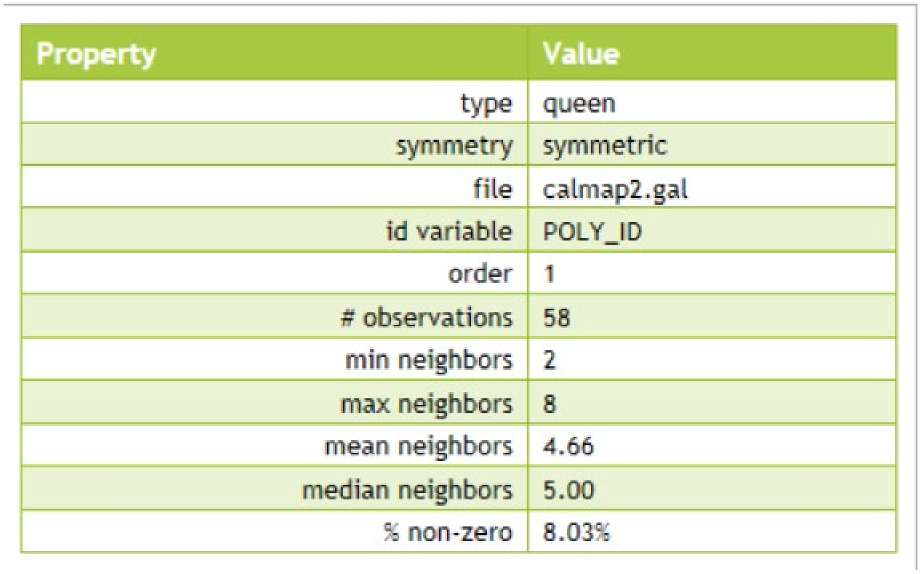
```
moran.test(calmap1$stdunemp,rscawm_q)

## 
##  Moran I test under randomisation
## 
## data:  calmap1$stdunemp  
## weights: rscawm_q  
## 
## Moran I statistic standard deviate = 0.81334, p-value = 0.208
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance 
##       0.051930232      -0.017543860       0.007296254
```

The p-value > alpha = .05, therefore, do not reject the null hypothesis that that unemployment is randomly distribution. There is weak evidence of positive spatial correlation.

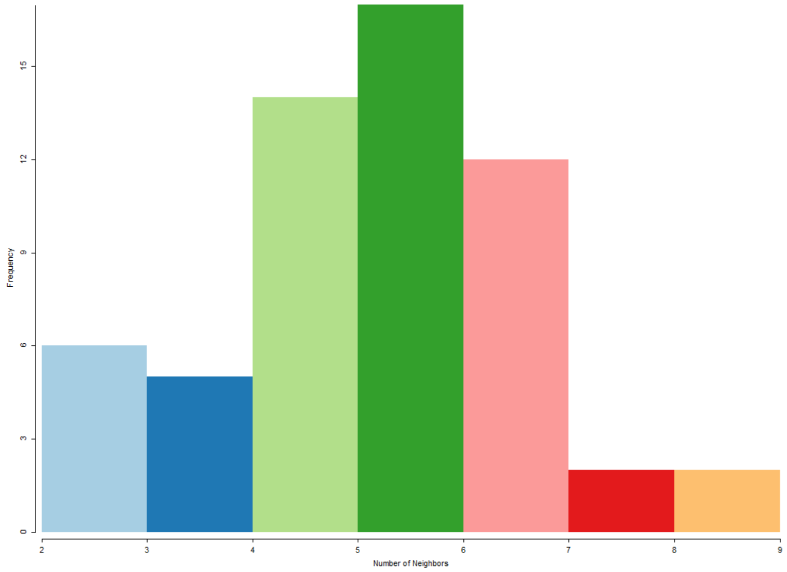
For the shape file and variable used in questions 1 and 2, read the data (shape file) into GeoDa.

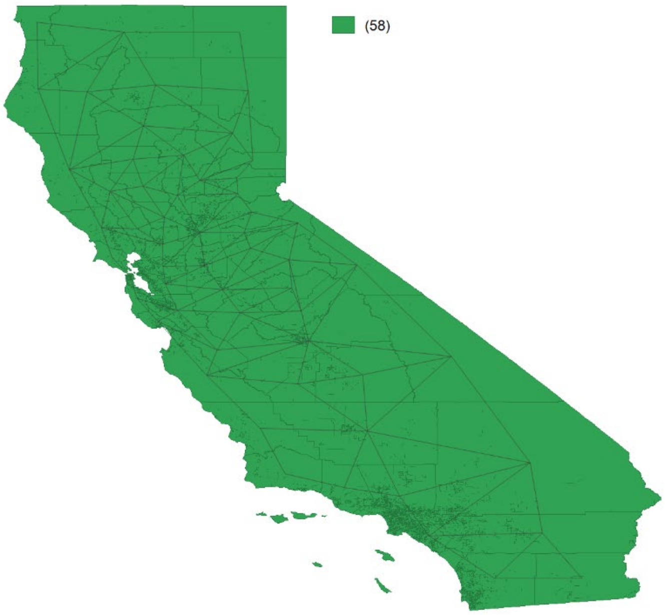


Generate a set of queen contiguity weights in GeoDa (order 1).



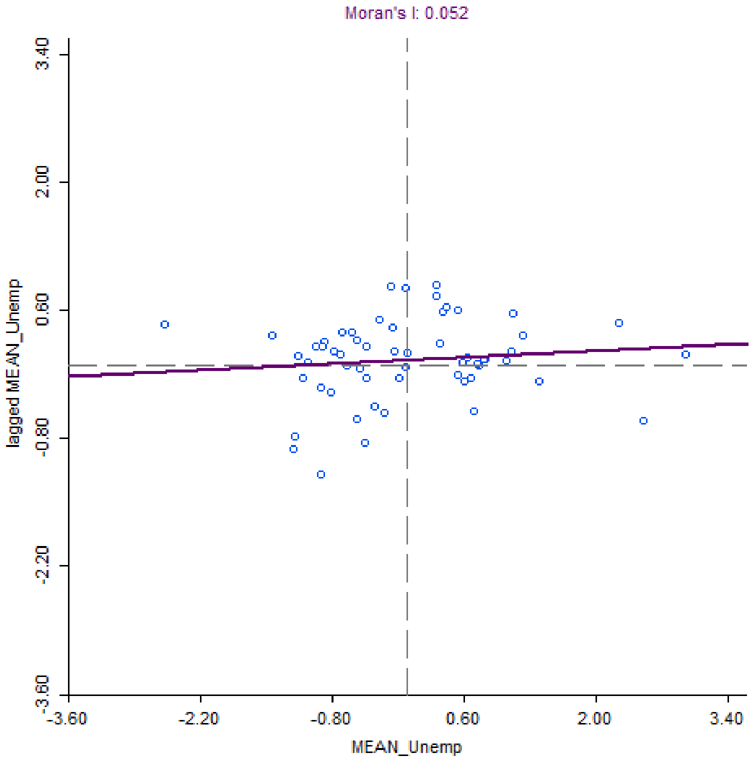
Property	Value
type	queen
symmetry	symmetric
file	calmap2.gal
id variable	POLY_ID
order	1
# observations	58
min neighbors	2
max neighbors	8
mean neighbors	4.66
median neighbors	5.00
% non-zero	8.03%





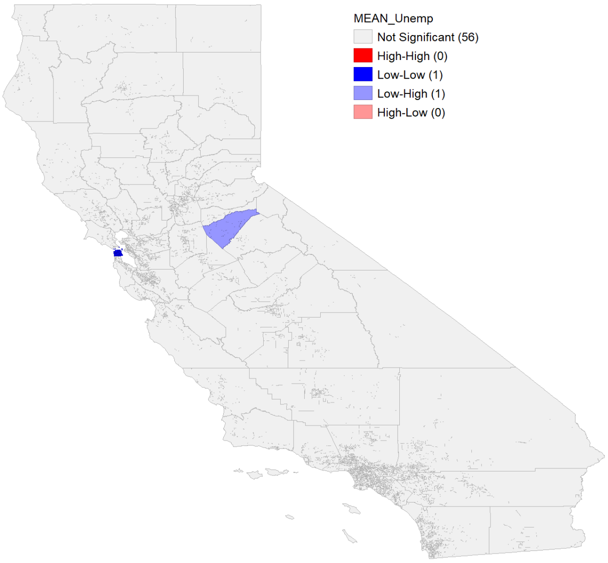
The histogram shows the number of neighbors for each of the counties using queen contiguity. The minimum connecting neighbors is two and the maximum is nine. The average is five to six neighbors for each county. The above map displays queen contiguity connectivity.

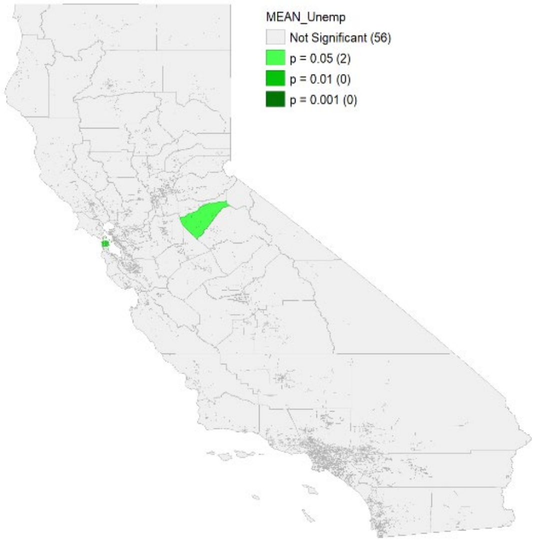
Generate the global Moran's I scatterplot (again) for the variable of interest and then generate a cluster map of LISA statistics for the same variable. Show the scatterplot and LISA cluster map. For any obvious clusters, discuss the positions of the polygons in the Global Moran I plot and interpret.



The Moran's I is used to evaluate spatial autocorrelation and to which the degree the observed values of a variable are similar or dissimilar from their neighboring values. The unemployment rate in the Cal Enviro dataset returns a Moran's I of .052 which indicates weak positive spatial autocorrelation. This Moran's I was also obtained in R for the unemployment rate.

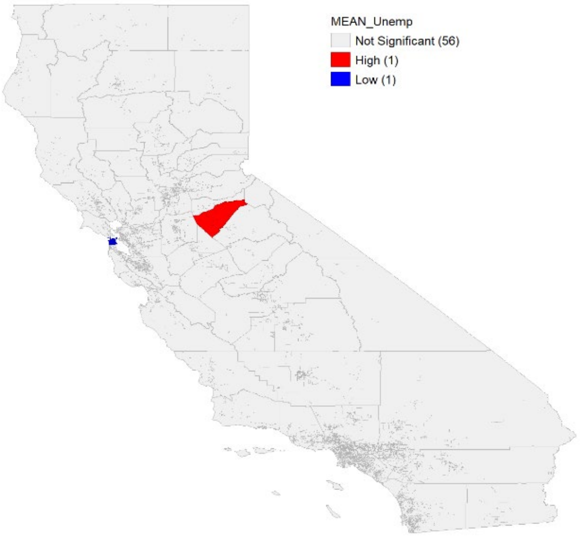
The below LISA clustering map has one county that is high-low and statistically significant at 5%. The observed value for the variable is less that its mean and the observed value for the spatial lag (neighboring counties) is greater than the mean, yielding negative correlation. There is one county that is low-low and statistically significant at 5%. This county is positively correlated and both observations are below their respective means.

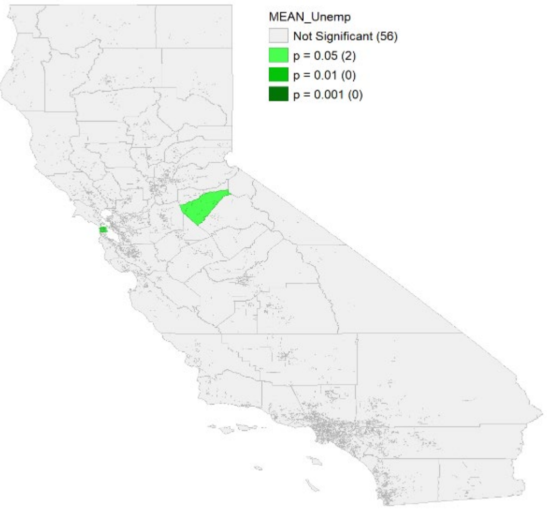




Generate Getis-Ord local G-statistics (not G*), map and show the resulting clusters and relate those to the LISA plots.

The Local G is a hot spot analysis that measures the concentration of high and low values in a given region. Both values found in the LISA plots are also statistically significant at 5%. The low-high LISA value is a "high' value in the Local G, meaning that the unemployment rate variable is greater than the mean value of all the observations for the same variable. The county that yielded low-low in the LISA plot is a 'low' value in the Local G, meaning that the variable is less than the mean of all the other observed values for the unemployment rate in the Cal Enviro dataset.


MEAN_Unemp
□ Not Significant (56)
🟥 High (1)
🟦 Low (1)


MEAN_Unemp
□ Not Significant (56)
🟩 p = 0.05 (2)
🟩 p = 0.01 (0)
🟩 p = 0.001 (0)