SIE512: Spatial Analysis
Fall 2019

Lab3: First and Second Order Spatial Data Analysis

Valerie Kamgue

9/29/2019

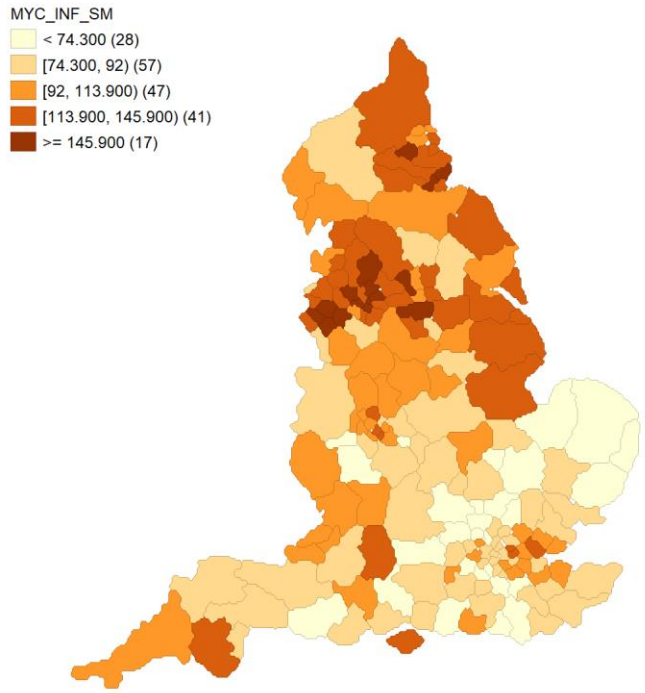
```
library(tmap)
library(ggplot2)
library(spdep)

library(raster)
library(stats)
library(foreign)
```

Check First Order Effects

Start Geoda. From the file menu select new project, click input file, specify shapefile and browse to where you copied the Lab3 data and select the dhapol.shp. A map will be displayed of the 190 UK health districts. You can expand or contract the map window as you like by clicking and dragging the edges. There is a bar between the map and legend that you can also adjust by clicking and dragging.

Clicking the table icon will open the data file so you can see the contents. Close the table and next create a map based on natural breaks classification. Use the menu option Map > Natural Breaks (5) to bring up the variable settings dialog. For the First Variable select MYC_INF_ SM (mycardial infarction i.e. heart attack). These data relate to males aged 35-64.

1) Add a copy of this Natural Breaks map to your rmarkdown file. (right clicking the map gives you an option to save image –save as a PNG). In your lab report describe any general spatial patterns you see in this map of heart attack rates.



Myocardial Infarction Rates

The map displays cases of heart attacks rate among male patients aged from 35 to 64 throughout the 190 districs using 5 different intervalles of values. The prevalence of the disease is more concentrated in northern region than the southern region with some few disparities. This might supposes that northern region has a high density of population compare to the south or the living conditions in the north greatly expose the population to the disease than in the south. There are 28 districts with lower cases and 17 districts with highest cases.

```
dha.nbr<-
read.gal("C:/Users/valer/Desktop/SIE512/Labs/Lab3/Lab3/dhapol.nb.gal")
```

To create the spatial moving average values, we need to convert the neighbor object into a spatial weights matrix with the function nb2mat. Use style = W to create a row standardized weights matrix and the zero.policy = TRUE is added to avoid an error for a polygon with no neighbors.

```
dha.mat<-nb2mat(dha.nbr, style="W", zero.policy=TRUE)

dim(dha.mat) #reports the dimension of the matrix

## [1] 190 190
```

This matrix has rows and columns corresponding to the Station ID's, with the weights for each station (polygon) in columns corresponding to the adjacent neighbors for that station (polygon). The weights are in rows corresponding to the adjacent neighbors of the stations of interest as columns.

```
dha.mat[1,] #displays the first row of the matrix
```

```
##   [1] 0.00 0.25 0.25 0.00 0.00 0.00 0.00 0.25 0.00 0.25 0.00 0.00 0.00
0.00
##  [15] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
##  [29] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
##  [43] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
##  [57] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
##  [71] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
##  [85] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
##  [99] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
## [113] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
## [127] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
## [141] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
## [155] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
## [169] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00
## [183] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```
dha.mat[,1] #display the first column of the matrix
```

```
##         1         2         3         4         5         6         7
## 0.0000000 0.2000000 0.3333333 0.0000000 0.0000000 0.0000000 0.0000000
##         8         9        10        11        12        13        14
## 0.2000000 0.0000000 0.1428571 0.0000000 0.0000000 0.0000000 0.0000000
##        15        16        17        18        19        20        21
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        22        23        24        25        26        27        28
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        29        30        31        32        33        34        35
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        36        37        38        39        40        41        42
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        43        44        45        46        47        48        49
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        50        51        52        53        54        55        56
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        57        58        59        60        61        62        63
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
```

```
##        64        65        66        67        68        69        70
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        71        72        73        74        75        76        77
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        78        79        80        81        82        83        84
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        85        86        87        88        89        90        91
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        92        93        94        95        96        97        98
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##        99       100       101       102       103       104       105
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       106       107       108       109       110       111       112
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       113       114       115       116       117       118       119
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       120       121       122       123       124       125       126
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       127       128       129       130       131       132       133
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       134       135       136       137       138       139       140
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       141       142       143       144       145       146       147
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       148       149       150       151       152       153       154
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       155       156       157       158       159       160       161
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       162       163       164       165       166       167       168
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       169       170       171       172       173       174       175
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       176       177       178       179       180       181       182
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       183       184       185       186       187       188       189
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##       190
## 0.0000000
```

We want to multiply the myocardial infarction data by this weights matrix.

To extract the myocardial infarction column, read in the DHA data from the .dbf file.

```
dha <- read.dbf("C:/Users/valer/Desktop/SIE512/Labs/Lab3/Lab3/dhapol.dbf")

## Field name: 'dha_spavg' changed to: 'dha_spavg.1'

dha.myc <- dha$MYC_INF_SM # extracts just the myocardial infarction data
column


dha.spavg<-dha.mat%*%dha.myc   #multiply the weights matrix by the dha.myc
```

```
vector

dha.spavg #displays the vector
```

```
##          [,1]
## 1   138.70000
## 2   131.32000
## 3   132.70000
## 4   117.06667
## 5   105.30000
## 6    96.90000
## 7   116.17500
## 8   137.50000
## 9   123.02500
## 10  130.65714
## 11  111.95000
## 12  124.24000
## 13  107.63333
## 14  113.36667
## 15  115.03333
## 16  125.35000
## 17  105.90000
## 18  120.87500
## 19  131.25000
## 20  108.86667
## 21  113.57778
## 22  116.36250
## 23   94.76667
## 24  107.40000
## 25  127.65000
## 26  127.90000
## 27  146.51667
## 28  142.56250
## 29  128.53333
## 30  128.37143
## 31  101.15000
## 32  141.10000
## 33  115.81667
## 34  110.85455
## 35   97.48333
## 36   92.06667
## 37  114.26000
## 38   86.18000
## 39  108.38333
## 40  105.46667
## 41   90.96667
## 42  111.71429
## 43  115.26667
## 44  122.20000
## 45  111.06667
```

```
## 46   73.92857
## 47   81.05000
## 48   64.48000
## 49   71.57500
## 50   67.52500
## 51   58.90000
## 52   76.56000
## 53   76.20000
## 54   69.81429
## 55   67.60000
## 56   72.17143
## 57   84.70000
## 58   64.41667
## 59   72.20000
## 60   74.06667
## 61   76.06667
## 62   76.12000
## 63   74.97500
## 64   73.30000
## 65   82.38000
## 66   88.55000
## 67   96.76000
## 68   92.85000
## 69   75.80000
## 70   81.25714
## 71   96.36667
## 72   96.26000
## 73   85.62000
## 74   78.23333
## 75   83.27500
## 76  100.98333
## 77   99.02500
## 78  100.96667
## 79  101.32500
## 80   89.40000
## 81  101.61429
## 82   90.24286
## 83   72.95000
## 84   70.11667
## 85   75.10000
## 86   78.70000
## 87   98.15000
## 88   92.67500
## 89   93.65000
## 90   88.00000
## 91   74.42000
## 92  104.52500
## 93   94.20000
## 94   75.90000
## 95   91.93333
```

```
## 96   86.18000
## 97  102.85000
## 98   81.70000
## 99   77.70000
## 100  75.15714
## 101  77.00000
## 102  73.90000
## 103  86.00000
## 104  72.08571
## 105  70.80000
## 106  75.96667
## 107  71.30000
## 108  76.03333
## 109  76.75000
## 110  81.53333
## 111  79.35000
## 112  83.95000
## 113  68.83333
## 114  80.75000
## 115  89.16667
## 116  76.24286
## 117  83.58571
## 118  83.98333
## 119 104.94286
## 120   0.00000
## 121  71.35000
## 122  79.66667
## 123  71.40000
## 124  70.78571
## 125  77.82500
## 126  78.51667
## 127  79.70000
## 128  81.77143
## 129  93.50000
## 130  98.02500
## 131 108.00000
## 132 100.65000
## 133  87.78000
## 134  95.92500
## 135  86.50000
## 136 102.50000
## 137  90.58333
## 138  98.03333
## 139  83.61667
## 140  88.44000
## 141  87.57500
## 142  93.15000
## 143  94.33333
## 144  94.56667
## 145 103.82857
```

```
## 146  98.50000
## 147 100.95000
## 148  79.08000
## 149  91.51667
## 150  86.82222
## 151 108.15000
## 152  92.95000
## 153 115.90000
## 154  99.20000
## 155 112.70000
## 156  86.25000
## 157  94.71667
## 158 109.45000
## 159  90.02000
## 160 100.36000
## 161 112.20000
## 162 117.67500
## 163 112.00000
## 164 127.16667
## 165 118.21429
## 166 126.40000
## 167 139.90000
## 168 138.24000
## 169 121.20000
## 170 127.75000
## 171 124.83333
## 172 117.55000
## 173 122.08000
## 174 121.50000
## 175 128.47500
## 176 135.07500
## 177 123.51667
## 178 128.25000
## 179 138.97500
## 180 154.07500
## 181 152.70000
## 182 146.18000
## 183 126.00000
## 184 146.37500
## 185 150.50000
## 186 137.12000
## 187 120.32000
## 188 129.22500
## 189 134.20000
## 190 137.13333
```

This vector provides the new spatially averaged myocardial infarction values. These need to be inserted back into the dbf file.

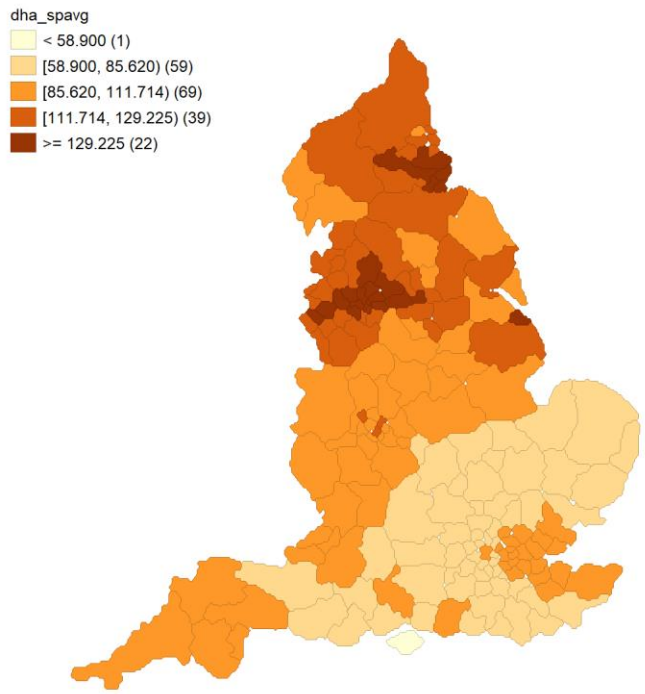
As a precaution, close GeoDa before altering the dhapol.dbf file. You may also want to create a backup copy of the dhapol.shp shapefile.

In Rstudio, these commands will create a new dbf file and attach the spatial moving average column to it:

```
dhatemp <-
read.dbf("C:/Users/valer/Desktop/SIE512/Labs/Lab3/Lab3/dhapol.dbf")

## Field name: 'dha_spavg' changed to: 'dha_spavg.1'

dhatemp2 <- cbind(dhatemp, dha.spavg) # merges the two files columnwise
write.dbf(dhatemp2,"dhapol.dbf") # writes the merged file back to a .dbf file
```

2) Open dhapol.shp in GeoDa again and create a Natural Breaks map using the new DHA_SPAVG column.

Add this map to your rmarkdown file (lab report). Describe this map relative to the first (unaveraged map) of heart attack rates. Examine this spatially averaged version of heart attack rates. In your lab report describe any first order (spatial trend) pattern(s) you see in this spatially averaged map of mycardial infarction.



Myocardial Infarction Rates

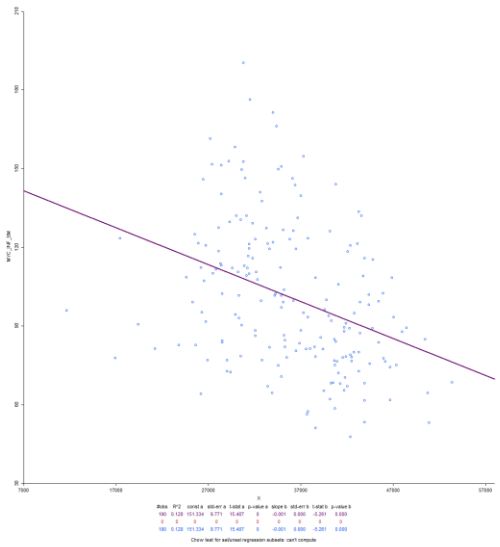
This second map has averaged the values of the data to a better representation of spatial patterns. The values are now clustered in 5 separate parts of the study area with regard to the prevalence. The trend of the distribution is spatially accessed with respect to the mean.

The lowest cases are still representated in the south as opposed to the highest cases in the north.

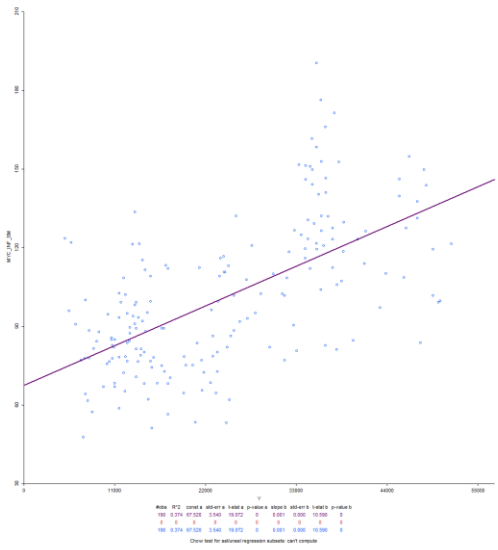
###Remove trend Use GeoDa for the next steps. We want to next remove a first order effect (a trend or pattern in the mean). We can do this by modeling a mean trend with a regression model.

In GeoDa create a scatterplot of myocardial infarction against X (the X coordinate centroids for the districts) and another scatter plot of myocardial infarction against Y (the Y coordinate centroids for the districts). Access this with the Scatterplot button (select X first as the independent variable and MYC_INF_SM as the dependent variable). A nice feature of GeoDa is that all data views are linked. If you select points in the scatterplot, the corresponding polygons are highlighted in the map. This is a useful way to identify where outliers in the scatterplot occur spatially.

3) Add images of the 2 scatterplots (right click gives the option to save to image) to your rmarkdown file and describe the relationship between myocardial infarction and X and myocardial infarction and Y.



Myocardial Infarction Rates



Myocardial Infarction Rates

According to the scatter plots, there is a negative correlation between Myocardial Infarction and X but there is a positive correlation between Myocardial Infarction and Y.

##Next compute a regression model for myocardial infarction using the X and Y coordinate values as explanatory variables. This is called a trend model (the explanatory variable are spatial coordinates or their powers).

In GeoDa use Regression and select MYC_INF_SM as the dependent variable and Y and X as the covariates. Check Classic model and click Run.

4) Copy the regression results to your lab report. To do this you can select the arrow in the upper left corner of the Regression report to save the result to a text file. This file can then be copied to a text block in your markdown file. Add your interpretation of the results (e.g how much variation does this regression model explain using X and Y as explanatory variables? Are the X and Y variables significant at the .05 significance level?

REGRESSION

SUMMARY OF OUTPUT: ORDINARY LEAST SQUARES ESTIMATION

Data set : dhapol

Dependent Variable : MYC_INF_SM Number of Observations: 190

Mean dependent var : 100.905 Number of Variables : 3

S.D. dependent var : 27.9136 Degrees of Freedom : 187

R-squared : 0.411039 F-statistic : 65.2543

Adjusted R-squared : 0.404740 Prob(F-statistic) :3.18466e-022

Sum squared residual: 87191 Log likelihood : -851.837

Sigma-square : 466.262 Akaike info criterion : 1709.67

S.E. of regression : 21.5931 Schwarz criterion : 1719.42

Sigma-square ML : 458.9

S.E of regression ML: 21.422

Variable	Coefficient	Std.Error	t-Statistic	Probability
CONSTANT	99.0065	9.76558	10.1383	0.00000
X	-0.000790753	0.000229509	-3.44542	0.00070
Y	0.00124691	0.000131613	9.47406	0.00000

REGRESSION DIAGNOSTICS

MULTICOLLINEARITY CONDITION NUMBER 13.692688

TEST ON NORMALITY OF ERRORS

TEST	DF	VALUE	PROB
Jarque-Bera	2	4.1437	0.12595

DIAGNOSTICS FOR HETEROSKEDASTICITY

RANDOM COEFFICIENTS

TEST	DF	VALUE	PROB
Breusch-Pagan test	2	9.2145	0.00998
Koenker-Bassett test	2	8.5767	0.01373

========== END OF REPORT
==========

When using X and Y as explanatory variables, the regression models ends up with a very small probability less than 0.05; therefore, we easily reject the null hypothesis with assurance that there is a significant relatioship between the variables X and Y.

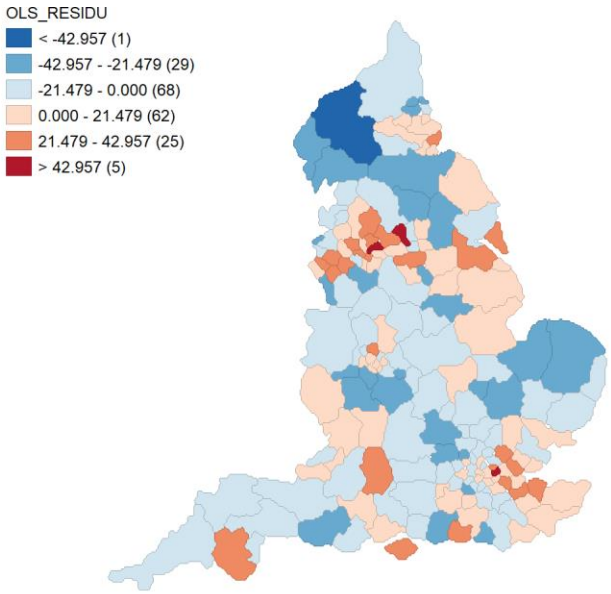
###GeoDa regression diagnostics include tests for mullticollinearity, normal distribution of residuals, and heteroskedasticity. Mullticollinearity indicates if explanatory variables are correlated. Multicollinearity does not reduce the predictive power of the model as a whole, but can affect significance of individual variables. Correlated predictors can indicate how well the entire bundle of predictors predicts the response variable but may not give valid results about any individual predictor, or about which predictors are redundant with respect to others. Values of 10 and above on this statistic indicate collinearity among the explanatory variables. The Jarque-Bera test is based on combined effects of skewness and kurtosis. It assumes a null hypothesis of a normal distribution (more specifically that skewness and excess kurtosis are zero) against the alternate that the residuals are not normally distributed. For a 95 percent confidence level we would reject the null hypothesis for a probability value less than .05. The diagnostics also include three tests for heteroskedasticity (non-constant variance). The null hypothesis is that the residual variance 2 is constant. Low probability values (< .05) suggest problems.

Close the result window. The Regression window should still be open - Click Save to Table. Check residuals and save the residuals using the default OLS_RESIDU name. They will be inserted at the end of the table.

Plot the residual values in GeoDa - check for second order effects

Create a standard deviations map of the residuals in GeoDa. Use Map > Standard Deviation and select the residuals field, OLS_RESIDU that was just added.

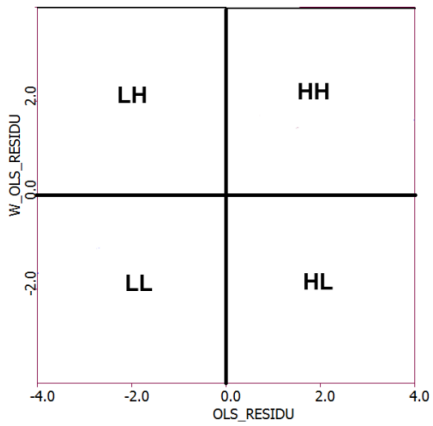
5) Add a copy of this map to your rmarkdown file. Describe any spatial pattern in the negative and positive residual values.



Myocardial Infarction Rates

Residuals are errors that the model can not explain. Residual = Observed value – predicted value. The negative values of the residuals (in blue on the map) are the over-predicted cases of heart attack, in other words there are actually less cases of myocarde infarctus than predicted. On the other hand, positive values (in pink on the map) of the residuals suggest an under-predicted cases of the disease, there are actually more cases than predicted. In this model, there is definitely a presence of second order effect due to the fact that residuals (negative and positive) are spatially almost evenely distributed throughout the 190 districts.

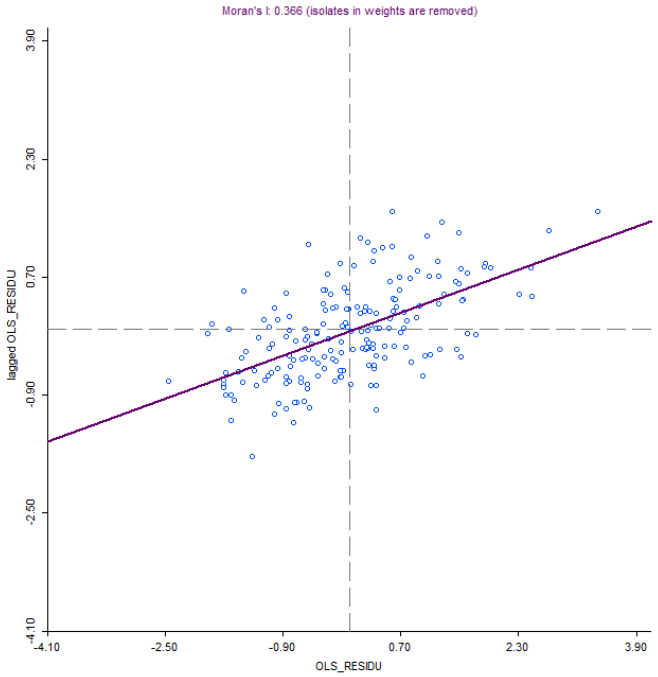
One way to see if there are second order effects present after we have removed a first order trend is with the Moran Scatterplot. This is a diagnostic plot which allows you to examine region values against neighboring values, in this case the residuals of the regression. It is similar to the lag plot you generated for Lab 2. The Moran's Scatterplot is divided into four quadrants. These plot regions indicate high values surrounded by high values (HH), low values surrounded by high value neighbors (LH), etc.



Create a Moran's Scatterplot in GeoDa:

Select Space > Univariate Moran's I and select the residuals as the variable.

6) Add the Moran's Scatterplot to your Rmarkdown file. In your report assess and briefly discuss the presence of second order effects using this plot and the residuals map created in the previous step.



Myocardial Infarction Rates

According to the Moran's Scatter plot, the HH and LL values are more significant than the HL and LH. Those HH and LL values suggest a positive correlation among the residuals and the HL/LH values define outliers.

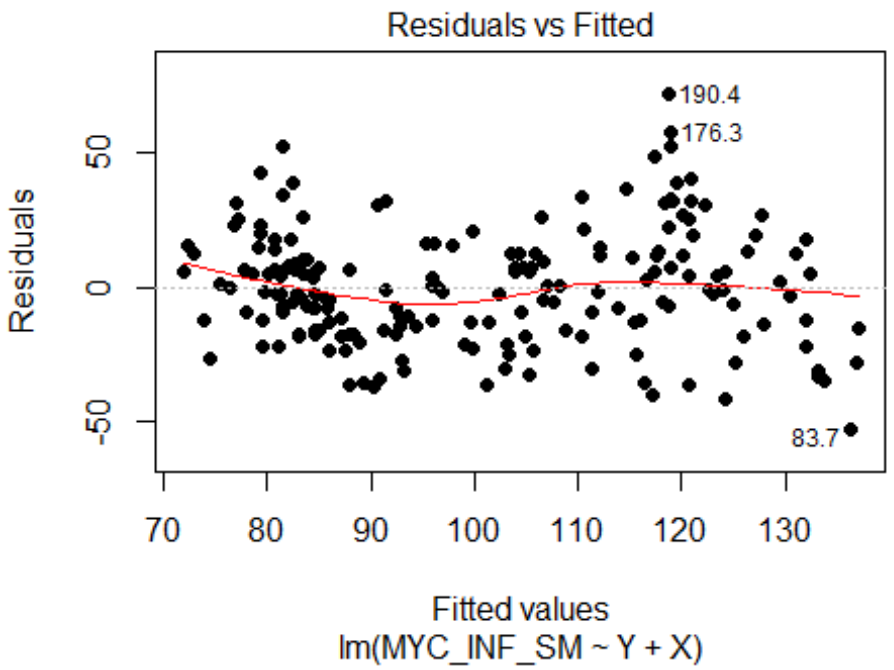
Run the same regression in R

```
dha.lm<-lm(MYC_INF_SM ~ Y+X, data = dhatemp)
```

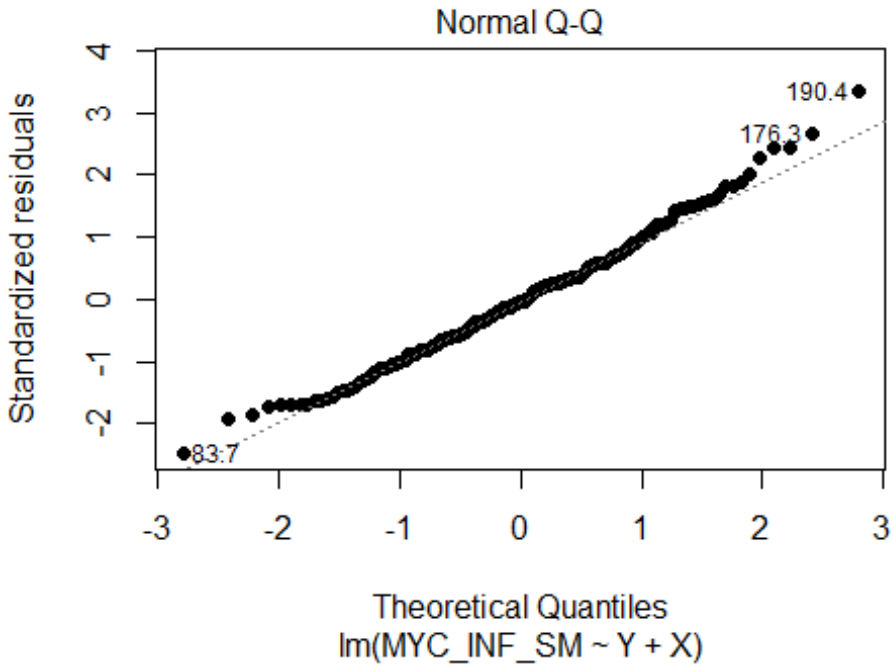
dha.lm<-lm(MYC_INF_SM ~ Y+X, data = dhatemp) # dhatemp <- read.dbf("dhapol.dbf") was created with this function above in case you need to recreate

7) Include these results along with the residual diagnostic plots (residual against fitted values and qq plots) in your lab report. The linear regression results should be the same as you obtained in GeoDa. Evaluate the assumptions of independence, constant variance, outliers, and normality of the residuals from the regression using the diagnostic tools of both GeoDa and R.

```
plot(dha.lm, which=1 ,pch=16, labels.id=dha.myc)
```




```r
plot(dha.lm, which=2 ,pch=16, labels.id=dha.myc)
```



8) In your lab report summarize your assessment of evidence of first order effects and any second order effects.

The plots of Residuals vs fitted and the QQ plot suggest a strong relationship among residuals with few outliers.