Lab #5 Spatial Regression  
(Due Date: 04/29/2017)

PURPOSES  
1. Learn to conduct alternative linear regression modeling on spatial data  
2. Learn to diagnose and take into account spatial autocorrelation in regression modeling

Before starting your lab, create a new directory named lab5 in your network drive (where you could organize files for this course), and create a Word document named lab5_yourusername.doc for your lab write up.

1. Linear Regression in R

To fit a linear model in R for regression using ordinary least square (OLS) is straightforward. We could use function lm().

\[
\text{lm(formula, data, subset, weights, na.action,}
\text{ method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,}
\text{ singular.ok = TRUE, contrasts = NULL, offset, ...)}
\]

Basically, we need to define the formula for our model. Say, we want to fit a linear model for response variable \( y \) and explanatory variable \( x \), then the formula could be defined as follows:

\[ y \sim x \]

Then, our model becomes

\[ \text{lm(y~x)} \]

If you want to incorporate weights for your linear model (i.e., using weighted least square), you just need to specify the parameter of weights in the \text{lm} function.

To know more about how to fit a linear model using \text{lm} function, check out the following link:

http://www.inside-r.org/r-doc/stats/lm

Besides \text{lm}, there are a set of functions that have been developed in R to support regression modeling. For example, \text{glm()} for generalized linear modeling (for logistic regression, binomial regression...). You could google the usage and tutorials for \text{glm()}.

http://www.inside-r.org/r-doc/stats/glm

2. The Boston Housing Dataset

We will use the (corrected) Boston Housing Dataset for this lab. This dataset is available in the \text{spdep} library (for spatial autocorrelation analysis). For the detailed description of the Boston Housing Dataset, refer to the manual of \text{spdep} library here (page 19-23):

http://cran.r-project.org/web/packages/spdep/spdep.pdf

To load the dataset, you need to use library \text{spdep}.

\[ > \text{library(spdep)} \]
> data(boston)

Basically, there are three datasets associated with boston data:

- **boston.c**: housing data
- **boston.utm**: x, y coordinates data
- **boston.soi**: a sphere of influence neighbors list

Simply, to see the spatial distribution of house records in this data, try the following command:

```r
> plot(boston.utm)
```

To see what specific data you have in the boston data, use the following command:

```r
> summary(boston.c)
```
Below is a Table that describes each variable in the corrected boston data (boston.c) (copied from the spdep library).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOWN:</td>
<td>a factor with levels given by town names</td>
</tr>
<tr>
<td>TOWNNO</td>
<td>a numeric vector corresponding to TOWN</td>
</tr>
<tr>
<td>TRACT</td>
<td>a numeric vector of tract ID numbers</td>
</tr>
<tr>
<td>LON</td>
<td>a numeric vector of tract point longitudes in decimal degrees</td>
</tr>
<tr>
<td>LAT</td>
<td>a numeric vector of tract point latitudes in decimal degrees</td>
</tr>
<tr>
<td>MEDV</td>
<td>a numeric vector of median values of owner-occupied housing in USD 1000</td>
</tr>
<tr>
<td>CMEDV</td>
<td>a numeric vector of corrected median values of owner-occupied housing in USD 1000</td>
</tr>
<tr>
<td>CRIM</td>
<td>a numeric vector of per capita crime</td>
</tr>
<tr>
<td>ZN</td>
<td>a numeric vector of proportions of residential land zoned for lots over 25000 sq. ft per town (constant for all Boston tracts)</td>
</tr>
<tr>
<td>INDUS</td>
<td>a numeric vector of proportions of non-retail business acres per town (constant for all Boston tracts)</td>
</tr>
<tr>
<td>CHAS</td>
<td>a factor with levels 1 if tract borders Charles River; 0 otherwise</td>
</tr>
<tr>
<td>NOX</td>
<td>a numeric vector of nitric oxides concentration (parts per 10 million) per town</td>
</tr>
<tr>
<td>RM</td>
<td>a numeric vector of average numbers of rooms per dwelling</td>
</tr>
<tr>
<td>AGE</td>
<td>a numeric vector of proportions of owner-occupied units built prior to 1940</td>
</tr>
<tr>
<td>DIS</td>
<td>a numeric vector of weighted distances to five Boston employment centres</td>
</tr>
<tr>
<td>RAD</td>
<td>a numeric vector of an index of accessibility to radial highways per town (constant for all Boston tracts)</td>
</tr>
<tr>
<td>TAX</td>
<td>a numeric vector full-value property-tax rate per USD 10,000 per town (constant for all Boston tracts)</td>
</tr>
<tr>
<td>PTRATIO</td>
<td>a numeric vector of pupil-teacher ratios per town (constant for all Boston tracts)</td>
</tr>
<tr>
<td>B</td>
<td>a numeric vector of 1000*(Bk - 0.63)^2 where Bk is the proportion of blacks</td>
</tr>
<tr>
<td>LSTAT</td>
<td>a numeric vector of percentage values of lower status population</td>
</tr>
</tbody>
</table>

### 3. Linear Regression Modeling of Boston Housing Data Using Ordinary Least Square

We are interested in studying the relationship between housing price (MEDV; response or dependent variable) and its driving factors (explanatory or independent variables) in Boston. Assume we have identified a set of driving factors (refer to the following table), including crime (CRIM), proportion of residential land (ZN), adjacency to Charles River (CHAS), number of rooms per dwelling (RM), distance to employment centers (DIS), proportion of lower status population (LSTAT).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEDV</td>
<td>Median values of owner-occupied housing in USD 1000</td>
</tr>
<tr>
<td>CRIM</td>
<td>Per capita crime</td>
</tr>
<tr>
<td>ZN</td>
<td>Proportions of residential land zoned for lots over 25000 sq. ft per town</td>
</tr>
<tr>
<td>CHAS</td>
<td>A factor with levels 1 if tract borders Charles River; 0 otherwise</td>
</tr>
<tr>
<td>RM</td>
<td>Average numbers of rooms per dwelling</td>
</tr>
<tr>
<td>DIS</td>
<td>Weighted distances to five Boston employment centres</td>
</tr>
<tr>
<td>LSTAT</td>
<td>Percentage values of lower status population</td>
</tr>
</tbody>
</table>

To conduct a linear regression model, we define the following formula:

$$ MEDV \sim CRIM + ZN + CHAS + RM + DIS + LSTAT $$

Then, we could apply linear regression using the following command:

```r
> bostlm1<-lm(MEDV~ CRIM + ZN + CHAS + RM + DIS + LSTAT,data=boston.c)
> summary(bostlm1)
```
To get the log-likelihood of the regression model, try the following command:

```r
> logLik(bostlm1)
```

Your results will be:

'log Lik.' -1550.049 (df=8)

Now, let’s apply another OLS linear regression modeling by log-tranforming the data. Try the following command:

```r
> bostlm2<-lm(log(MEDV)~ CRIM + ZN + CHAS + RM + log(DIS) +
log(LSTAT),data=boston.c)
> summary(bostlm2)
```

To obtain the likelihood of the model, try:

```r
> logLik(bostlm2)
```
The log-likelihood for the revised model is:
'log Lik.' 91.9578 (df=8)

Given the regression results of a linear model, we need to examine the residuals of our fitting and associated spatial autocorrelation. We could plot the residuals vs. response variable to visually evaluate the quality of our fitting.

```r
> plot(boston.c$MEDV, residuals(bostlm1))
> plot(log(boston.c$MEDV), residuals(bostlm2))
```

We could also inspect visually the spatial distribution of our model residuals:

```r
>bos<-boston.c
>coordinates(bos)<-~LAT+LON
>bos$resid<-resid(bostlm1)
bos$resid2<-resid(bostlm2)
> spplot(bos,'resid', col.regions=bpy.colors(30))
> spplot(bos,'resid2',col.regions=bpy.colors(30))
```
Note that besides longitude/latitude coordinates, you could also assign X/Y coordinates for the spatial point pattern (bos).

```r
> bos <- boston.c
> bos$x <- boston.utm[,1]
> bos$y <- boston.utm[,2]
> coordinates(bos) <- ~ x+y
```

To evaluate the spatial autocorrelation structure of the residuals, we could conduct spatial autocorrelation analysis using Moran’s I or Geary’s C.

```r
> w <- nb2listw(boston.soi)
> lm.morantest(bostlm1, w)
> lm.morantest(bostlm2, w)
```

**Global Moran’s I for regression residuals**

data:  
model: lm(formula = MEDV ~ CRIM + ZN + CHAS + RM + DIS + LSTAT, data = boston.c)  
weights: w

Moran I statistic standard deviate = 15.163, p-value < 2.2e-16  
alternative hypothesis: greater  
sample estimates:

<table>
<thead>
<tr>
<th>Observed Moran's I</th>
<th>Expectation</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4702336086</td>
<td>-0.0082755787</td>
<td>0.000957827</td>
</tr>
</tbody>
</table>

**Global Moran’s I for regression residuals**

data:  
model: lm(formula = log(MEDV) ~ CRIM + ZN + CHAS + RM + log(DIS) + log(LSTAT), data = boston.c)  
weights: w

Moran I statistic standard deviate = 15.1202, p-value < 2.2e-16  
alternative hypothesis: greater  
sample estimates:

<table>
<thead>
<tr>
<th>Observed Moran's I</th>
<th>Expectation</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4686311596</td>
<td>-0.0093555200</td>
<td>0.000951049</td>
</tr>
</tbody>
</table>

Or, as we have learned, semivariogram analysis can be used to examine the spatial dependency structure in the model residuals.

```r
> library(gstat)
> bosResid.vgm <- variogram(bos$resid~1, bos)
> bosResid2.vgm <- variogram(bos$resid2~1, bos)
> plot(bosResid.vgm)
> plot(bosResid2.vgm)
```
Go to Question 1

4. Spatial Regression Modeling Using Generalized Least Square

Generalize Least Square (GLS) allows for the incorporation of spatial correlation structure directly into your regression model. In R, a function gls() in the library nlme is available for GLS-based linear regression.

```
gls(model, data, correlation, weights, subset, method, na.action, control, verbose)
```

```
## S3 method for class 'gls':
update((object, model., ..., evaluate = TRUE))
```

Basically, what we need to do is to specify the spatial correlation structure (i.e., variance-covariance matrix; for the parameter correlation) that we want to incorporate in GLS. We have learned that there are a series of covariance models that we could use, for example, spherical, exponential…

To conduct GLS in R, you need to install the nlme package first. Then load it to your R environment.

```
> library(nlme)
```

For the first linear regression model in Section 3, we could apply GLS using the following command:

```
> bos.gls<-gls(MEDV~ CRIM + ZN + CHAS + RM + DIS + LSTAT,data=bos,corr=corSpher(c(0.02,0.2) , form=~LAT+LON,nugget=T))
> summary(bos.gls)
```
For the second linear regression model in Section 3, we could apply GLS using the following command:

```r
>bos.gls2<-gls(log(MEDV)~ CRIM + ZN + CHAS + RM + log(DIS) + log(LSTAT),data=bos,corr=corSpher(c(0.02,0.2) , form=~LAT+LON,nugget=T))
> summary(bos.gls2)
```

```r
> bos.gls2<-gls(log(MEDV)~ CRIM + ZN + CHAS + RM + log(DIS) + log(LSTAT),data=bos,corr=corSpher(c(0.02,0.2) , form=~LAT+LON,nugget=T))
> summary(bos.gls2)
```

Go to Question 2
5. Spatial Regression Modeling Using Simultaneous Autoregression
In this section, we learn how to conduct spatial regression modeling with correlated errors. We focus on two types of models that we have learned: spatial lag modeling and spatial error modeling, both belonging to the category of spatial simultaneous autoregression.

5.1. Spatial Lag Modeling
For spatial lag modeling, we take into account the spatial autocorrelated structure in dependent variables. In R, the function lagsarlm() can be used to conduct spatial lag modeling. Lagsarlm is based on maximum likelihood estimation of spatial simultaneous autoregressive lag.

\[
lagsarlm(formula, data = list(), listw, 
  na.action, type="lag", method="eigen", quiet=NULL, 
  zero.policy=NULL, interval=NULL, tol.solve=1.0e-10, trs=NULL, 
  control=list())
\]

For our Boston Housing data, we could use the following command to conduct spatial lag modeling:

```r
> bostsar1<-lagsarlm(log(MEDV)~ CRIM + ZN + CHAS + RM + log(DIS) + 
  log(LSTAT),data=boston.c,w)
> summary(bostsar1)
```

5.2. Spatial Error Modeling
In R, function errorsarlm() supports spatial error modeling, which allows for taking into account the spatial autoregressive structure in the residuals of our model.

Try the following command to apply spatial error modeling to the Boston Housing data:

```r
> bostsar2<-errorsarlm(log(MEDV)~ CRIM + ZN + CHAS + RM + log(DIS) + 
  log(LSTAT),data=boston.c,w)
```
> summary(bostsar2)

> summary(bostsar2)

Call: errorarlm(formula = log(MEDV) ~ CRIM + ZN + CHAS + RM + log(DIS) + log(LSTAT), data = boston.c, listw = w)

Residuals:
         Min       1Q   Median       3Q      Max
-0.69758268 -0.06812901 0.00087692 0.08602011 0.38396541

Type: error

Coefficients: (asymptotic standard errors)

     Estimate Std. Error  z value Pr(>|z|)    
(Intercept) 3.25453323 0.14296478 22.7646 < 2.2e-16
CRIM -0.00610646 0.00101452 -6.0191 1.754e-09
ZN  0.00078028 0.00053549  1.4571 0.1451
CHAS1 -0.02841273 0.03040200 -0.9346 0.3500
RM  0.09470663 0.01410326  6.7152 1.877e-11
log(DIS) -0.01319936 0.04418358 -0.2987 0.7651
log(LSTAT) -0.33290832 0.02251756 -14.7840 < 2.2e-16

Lambda: 0.73342, LR test value: 245.8, p-value: < 2.2e-16
Asymptotic standard error: 0.030367
z-value: 24.152, p-value: < 2.2e-16
Wald statistic: 563.32, p-value: < 2.2e-16

Log likelihood: 214.8572 for error model
ML residual variance (sigma squared): 0.020863, (sigma: 0.14444)
Number of observations: 506
Number of parameters estimated: 9
AIC: -411.71, (AIC for lm: -167.92)

Go to Question 3

6. Geographically Weighted Regression

In this section, we learn to conduct geographically weighted regression (GWR). There is an R package named “spgwr” that supports GWR for the exploration of spatial nonstationarity in your data. Also, in ArcGIS, GWR toolbox is available.

In R, we first need to install the spgwr package. Then load the package into your R environment

```r
>library(spgwr)
```

To conduct GWR on Boston housing data, we will next need to select the bandwidth for GWR.

```r
>bost.bw <- gwr.sel(log(MEDV) ~ CRIM + ZN + CHAS + RM + log(DIS) + log(LSTAT),data=boston.c,coords=cbind(boston.c$LAT,boston.c$LON))
```
Once bandwidth is chosen, we conduct GWR using the following command:

```r
> bostgwr<-gwr(log(MEDV)~ CRIM + ZN + CHAS + RM + log(DIS) + log(LSTAT), data=boston.c, coords=cbind(boston.c$LAT,boston.c$LON), bandwidth=bost.bw, hatmatrix=TRUE)
> bostgwr
```

Go to Question 4
Questions:

**Question 1:** For the Boston Housing Data, identify a new set of factors to conduct linear regression using OLS (as shown in Section 3). Interpret the results of each linear regression model (at least 1 paragraph). Then compare the results of the two models and give your justification on why a model is better than the other one (at least 1 paragraph).

**Question 2:** Based on the same set of factors that you identify in Question 1, conduct spatial regression modeling using generalized least square (GLS) as you have learned in Section 4. Interpret the results of each spatial regression model (at least 1 paragraph). Then compare the results of the two GLS models with those in Question 1 (at least 1 paragraph).

**Question 3:** Based on the same set of factors that you identify in Question 1, conduct spatial regression modeling using simultaneous autoregression (SAR) as you have learned in Section 5. Interpret the results of each spatial regression model (at least 1 paragraph). Then compare the results of the two SAR models with those in Question 1 and 2 (at least 1 paragraph).

**Question 4:** What is Geographically Weighted Regression? Based on the results of GWR and other linear regression models for the new set of factors that you have identified, compare GWR with linear regression models.

**Note that** you need to provide the R commands (code) that you use together with the results. Otherwise, you will receive ZERO credit.