Areal Data Group Project

Kayla, Ingmar, Hanmo

2021-12-06

Contents

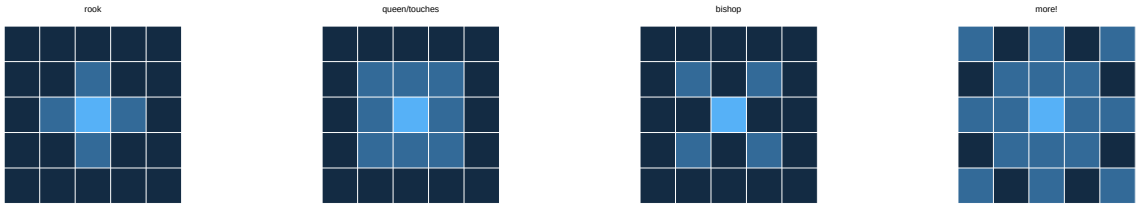
Introduction to Areal Data

Areal data is spatial data which is observed or reported for spatial units which are polygons with defined boarders. Some examples are country level data, state level data, county level data, or zip code level data where the data is aggregated for the defined area. This is often the case for private information (i.e. mean income in a zip code instead of income of each individual with their address), or data the is collected at that scale (i.e. proportion of democratic voters in an electoral unit; Bivand et al., 2013).

Some of the common problems we have with areal data have to do with the selected boundary. For example, gerrymandering is a common issue with votes because changing the boundaries yields different election results. Another issue is the choice of boundaries is often not made for the specific question being asked in the model. Bivand et al., 2013 gives the example of zip codes which are a common unit for demographic data to be reported in, but were designed for postal delivery not demographic data collection. The boundaries not fitting the underlying patterns in the data can cause model misspecification, spatial autocorrelation, and make it difficult to ascertain the number of independent observations in a dataset. Therefore, it is important to check for spatial patterning due to this partitioning to see if the model indicated is appropriate (Bivand et al., 2013).

Neighbors and weights

Because we do not have a continuous surface on which to calculate distance between points to build a variogram we must consider neighbors instead. There are many ways to describe neighbors. Here are a few examples where we considering the neighbors j of the center polygon i where i and j are $1 \dots n_{polygons}$, and the polygons j and colored based on if we consider them to be neighbors or not.



Using these neighbor relationships we can create a `weights matrix` based on understanding what the expected relationships are between neighbors we can build a weight matrix. The most simple example of this is where the $w_{ij} = 0$ if the polygons are not neighbors and $w_{ij} = 1$ if polygons are neighbors.

After we do our analysis we also need to check the residuals to ensure we removed the spatial patterns.

more background on models??

Our dataset

The dataset we are using is published in `spData` and comes from: *Anselin, Luc. 1988. Spatial econometrics: methods and models. Dordrecht: Kluwer Academic, Table 12.1 p. 189.*

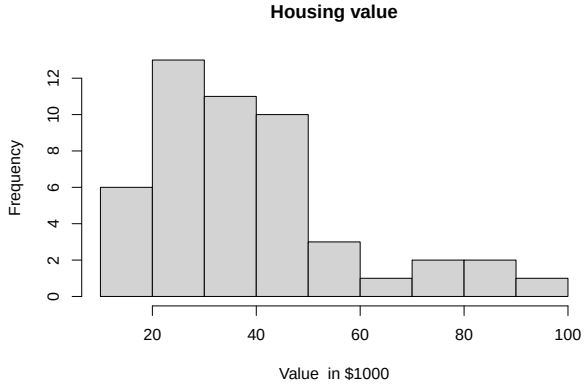
```
## load dataset
columbus <- vect(system.file("shapes/columbus.shp", package="spData")[1])
df.columbus <- as.data.frame(columbus)
```

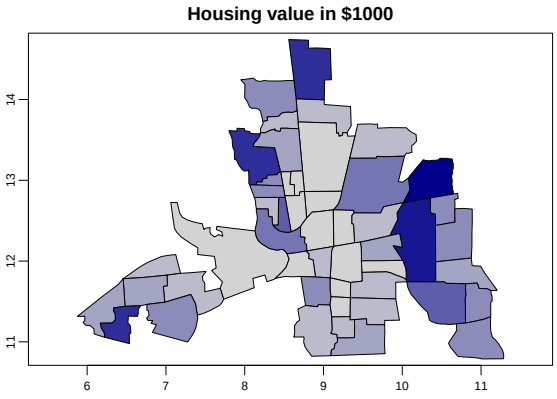
The county data is:

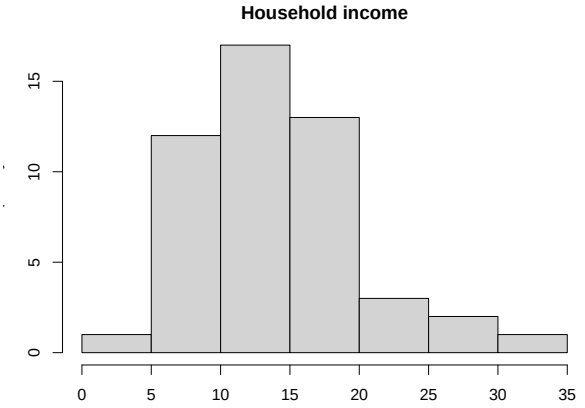
- `HOVAL`$_i$ housing value (in \$1,000) of county $i = 1 \dots 49$

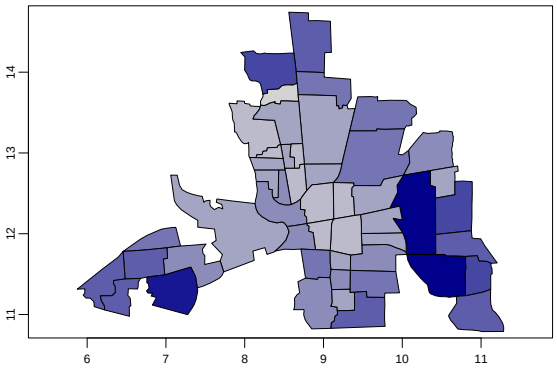
- $\texttt{INC}_i$ household income (in \$1,000) of county $i = 1 \dots 49$
- $\texttt{CRIME}_i$ residential burglaries and vehicle thefts per thousand households in the neighborhood of county $i = 1 \dots 49$
- $\texttt{OPEN}_i$ open space in neighborhood of county $i = 1 \dots 49$
- $\texttt{PLUMB}_i$ percentage housing units without plumbing of county $i = 1 \dots 49$
- $\texttt{CP}_i$ core-periphery (an urban rural proxy) is an indicator variable $I_{CP}(i) = \begin{cases} 1 & \text{if county } i \text{ is in the core} \\ 0 & \text{otherwise} \end{cases}$

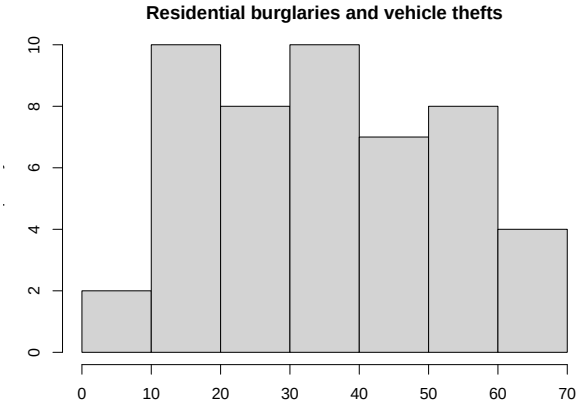
Look at some summaries of those metrics: *Kayla*

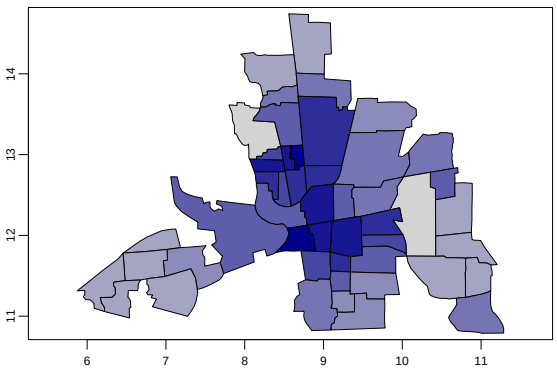


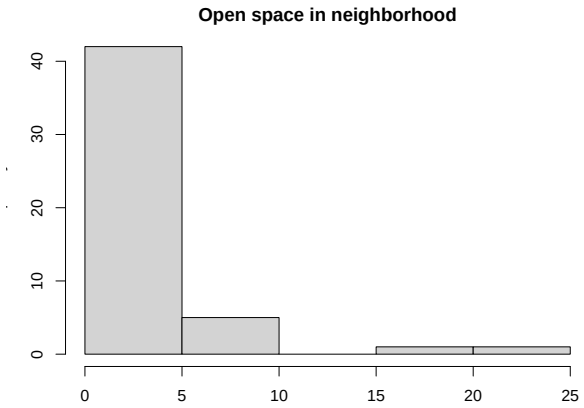


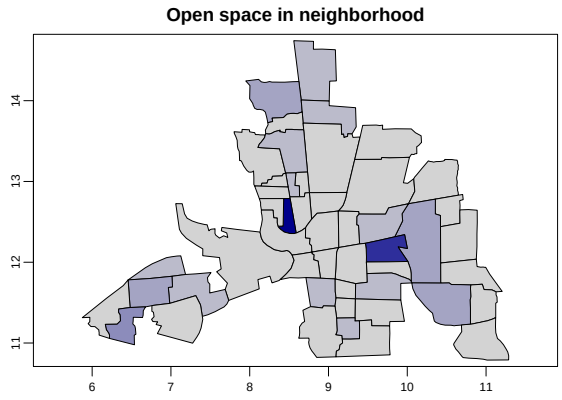


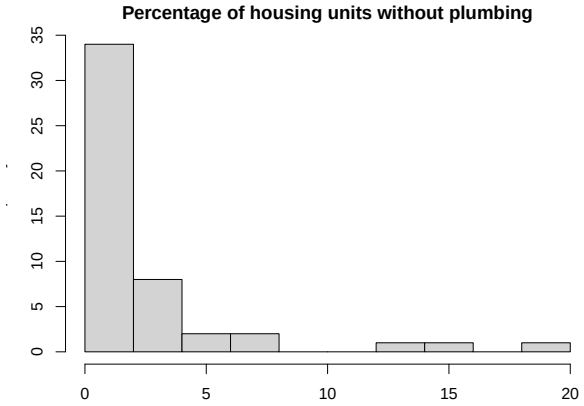


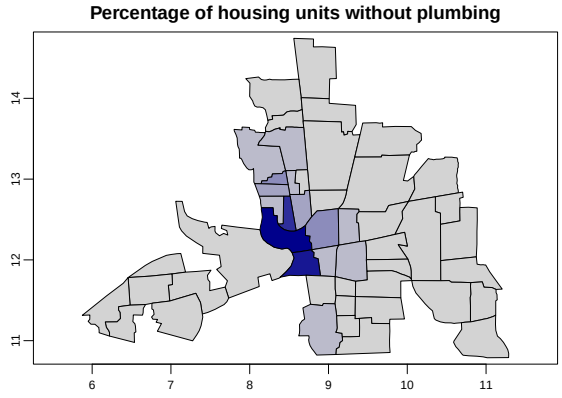


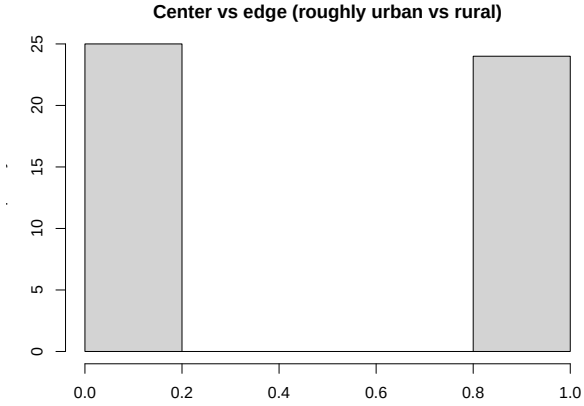


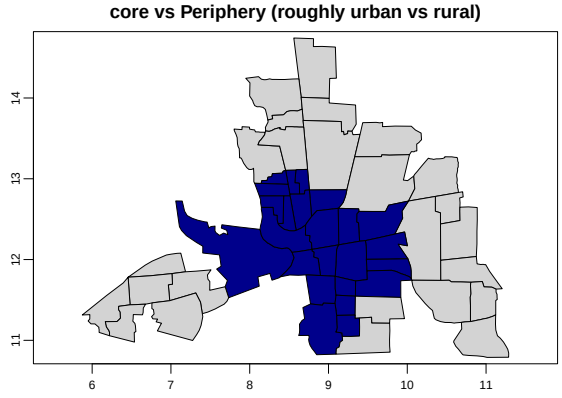






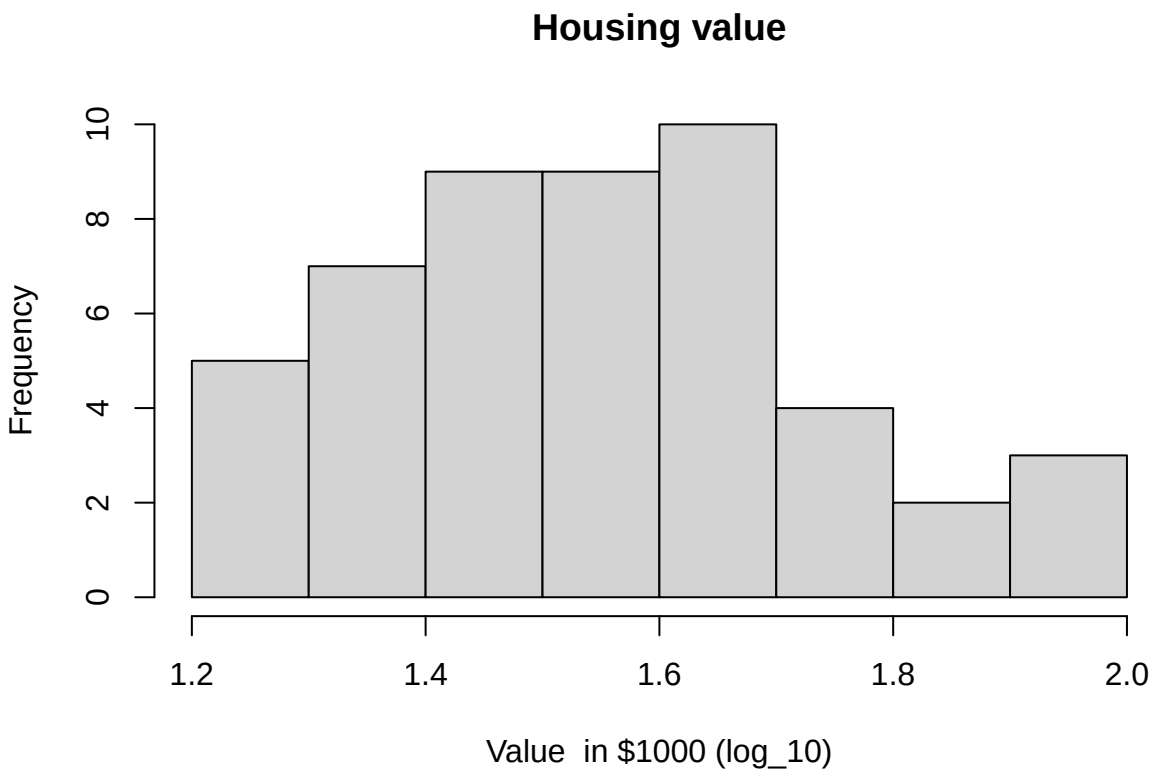






Because we plan to use housing value as our response variable, a logarithm transformation is applied to fix the skewness of the distribution for the variable housing value.

```
# Make histogram
hist(log10(df.columbus$HOVAL),
     main = "Housing value",
     xlab = "Value  in $1000 (log_10)")
```



Exploratory Data Analysis

Measures of spatial association

Neighbors

There are a many options when making the adjacency matrix as outlined above, but for our purposes we are saying that any counties touching each other are neighbors (1s) and any that aren't are not (0s). This is the neighbor schema shown in the queen/touches plot above.

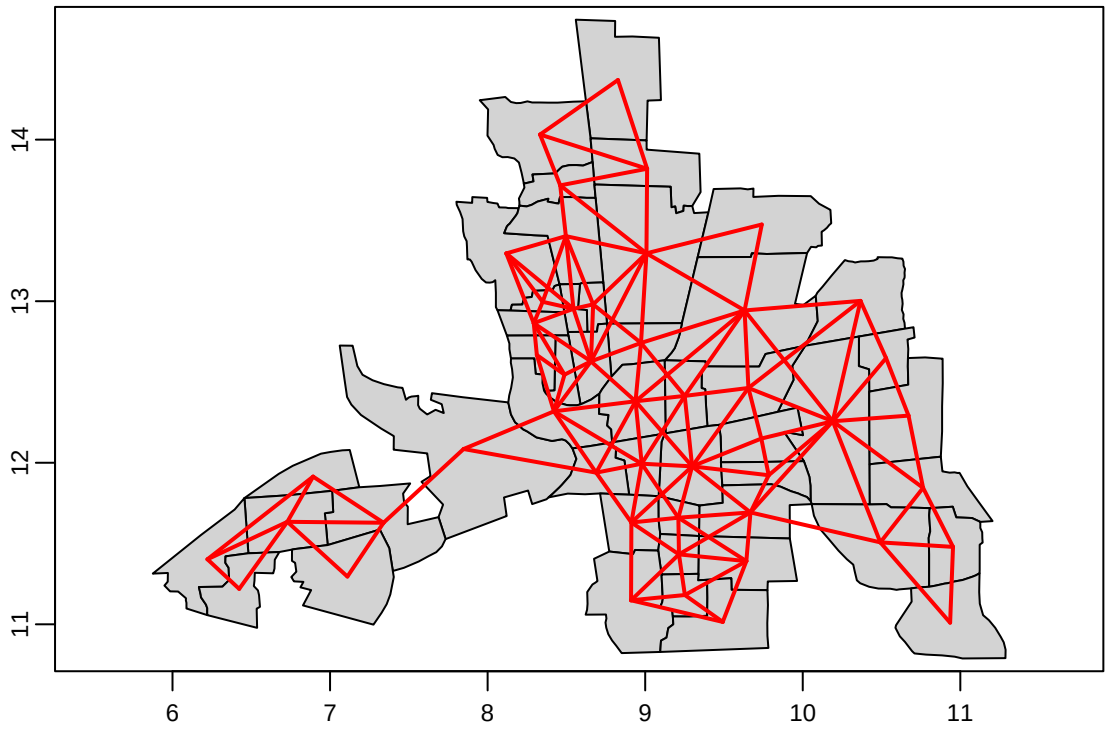
Figure out what the neighbors are, we must decide a few rule. Here we are considering all counties $i = 1 \dots 49$ and $j = 1 \dots 49$ that touch each other to be neighbors and that the relationship is symmetrical (i.e. $w_{ij} = w_{ji}$)

```
xy <- terra::centroids(columbus)
neighbors <- adjacent(columbus, type = "touches", symmetrical=TRUE)
colnames(neighbors) <- c("i", "j")
head(neighbors)
```

```
##      i j
## [1,] 1 2
## [2,] 1 3
## [3,] 2 3
## [4,] 2 4
```

```
## [5,] 3 4
## [6,] 3 5
```

```
plot(columbus, col='lightgray', border='black', lwd=1)
p1 <- xy[neighbors[,1], ]
p2 <- xy[neighbors[,2], ]
lines(p1, p2, col='red', lwd=2)
```



As described above, there are many options when making the adjacency matrix, but for our purposes we are saying that any counties touching each other are neighbors ($w_{ij} = 1$) and any that are not touching each other are not ($w_{ij} = 0$).

Here is what part of the adjacency matrix looks like:

```
adjacent(columbus, "touches", pairs = FALSE)[1:10,1:10]
```

```
##    1 2 3 4 5 6 7 8 9 10
## 1  0 1 1 0 0 0 0 0 0  0
## 2  1 0 1 1 0 0 0 0 0  0
## 3  1 1 0 1 1 0 0 0 0  0
## 4  0 1 1 0 1 0 0 1 0  0
## 5  0 0 1 1 0 1 0 1 1  0
## 6  0 0 0 0 1 0 0 0 1  0
## 7  0 0 0 0 0 0 0 1 0  0
## 8  0 0 0 1 1 0 1 0 0  0
## 9  0 0 0 0 1 1 0 0 0  1
## 10 0 0 0 0 0 0 0 0 1  0
```

From that we can understand that $\text{county}_{i=1}$ is touching $\text{counties}_{j=2,3}$ and not touching $\text{counties}_{j=4...10}$, because the weights of $w_{1,2}, w_{1,3} = 1$ and $w_{1,4...10} = 0$.

Moran's I

The assumptions for this test are (Bivand et al., 2013):

- " the mean model of the data removes systematic spatial patterning from the data"
- the observed spatial autocorrelation is not due to an underlying process in our model
- the chosen weights matrix suits the underlying interactions between the polygons

Therefore the main limitation of this method is that the model variance can be misspecified because it does not meet one of the assumptions above, i.e. the chosen weights matrix needs to suit the underlying interactions between the polygons. Other limitations include that t

Using `terra` to test for spatial autocorrelation of each variable, by county in Columbus, OH.

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(y_i - \overline{y})(y_j - \overline{y})}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

Moran's I is a global measure of spatial autocorrelation with values ranging -1 to 1. Here we are using a neighbor's matrix for any counties that are touching each other (w_{ij}). n is the number of neighborhoods in Columbus, OH which are indexed by i and j (is polygon i a neighbor with j. $\overline{y}$ is the mean value of the variable of interest and $y_{1...n}$ is that value in each polygon.

The adjacency matrix for all counties that touch each other (w_{ij} above, calling ww here):

```
ww <-  adjacent(columbus, "touches", pairs=FALSE)
```

Roughly the expected value for Moran's I is $E(I) = \frac{-1}{n-1}$

```
(ev = -1/(nrow(columbus)-1))
```

```
## [1] -0.02083333
```

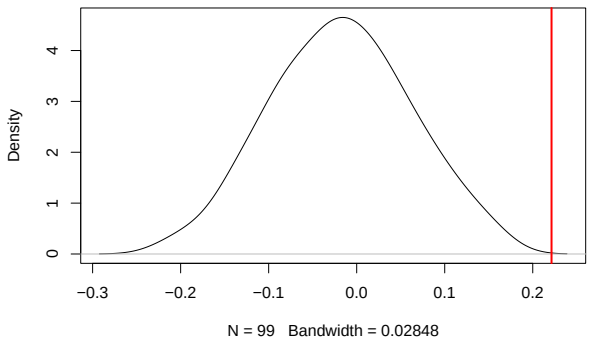
Values significantly ($\alpha = 0.05$) below the expected value indicate negative spatial autocorrelation (a phenomena that generally occurs in random datasets) and above that indicates positive spatial autocorrelation (neighbors are more similar to each other than non-neighbors). The null hypothesis here is that the values are distributed following a random process (I <= -0.0208333), and the alternative hypothesis is that the values are distributed with positive spatial autocorrelation (I > -0.0208333). The p-value is calculated using a Monte Carlo simulation, from which we derive a density plot of the I values from each permutation and calculate the number of times our simulated I value is greater than or equal to the observed value out of all the trials. A monte carlo simulation is the best method for this because it is robust to irregularly shaped polygons.

```
## Moran's I
(ac <- autocor(columbus$HOVAL, ww, "moran"))
```

House value

```
## [1] 0.2213441
```

```
## Monte Carlo sim to test for significance
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$HOVAL), ww, "moran")
})
plot(density(m), main = NA); abline(v=ac, col = "red", lwd = 2) #distribution of values of I using subs
```




```
## p-value
sum(m >= ac) / 100 # number of times I of subset is >= to I of entire dataset / number of trials
```

```
## [1] 0
```

So there is significant (Moran's I = 0.2213441, p < 0.05) spatial autocorrelation in house value, meaning the average value of houses in neighboring neighborhoods are different from the average value of all neighborhoods.

```
## Moran's I
(ac <- autocor(columbus$INC, ww, "moran"))
```

Household income

```
## [1] 0.412344
```

```
## sim
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$INC), ww, "moran")
```

```
})


## p-value
sum(m >= ac) / 100
```

```
## [1] 0
```

There is significant positive (Moran's I = 0.412344, p < 0.05) Spatial autocorrelation in household income.

```
## Moran's I
(ac <- autocor(columbus$CRIME, ww, "moran"))
```

Crime

```
## [1] 0.5154614
```

```
##sim
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$CRIME), ww, "moran")
})

## p-value
sum(m >= ac) / 100
```

```
## [1] 0
```

So again we see significant spatial autocorrelation (Moran's I = 0.5154614, p < 0.05).

```
## Moran's I
(ac <- autocor(columbus$OPEN, ww, "moran"))
```

Open space

```
## [1] -0.03669849
```

```
## sim
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$HOVAL), ww, "moran")
})

## p-value
sum(m >= ac) / 100
```

```
## [1] 0.51
```

There is not spatial autocorrelation with open space (Moran's I = -0.0366985; p-value > 0.05)

```
## Moran's I
(ac <- autocor(columbus$PLUMB, ww, "moran"))
```

Plumbing

```
## [1] 0.4550575
```

```
## sim
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$HOVAL), ww, "moran")
})

## p-value
sum(m >= ac) / 100
```

```
## [1] 0
```

There is significant spatial autocorrelation with plumbing (Moran's I = 0.4550575, p $<$ 0.05).

Geary's C

Geary's C is a measure of local spatial autocorrelation that is roughly inversely related to Moran's I. The values ranging 0 to $>$ 1, with values 0-1 representing positive spatial autocorrelation and values $>$ 1 representing negative spatial autocorrelation.

The neighbor's matrix (w_{ij}) is the same what was used for Moran's I. N is the number of spatial units indexed by i and j; x is the variable of interest; $\bar{x}$ is the mean of x; w_{ij} is a matrix of spatial weights with zeroes on the diagonal (i.e., $w_{ii} = 0$); and W is the sum of all w_{ij}. $\sum_{i \neq j} w_{ij}$ is the sum of that weight matrix with the diagonal equal to 0.

$$C = \frac{(n-1) \sum_i \sum_j w_{ij} (y_i - y_j)^2}{2 \left(\sum_{i \neq j} w_{ij} \right) \sum_i (y_i - \bar{y})^2}$$

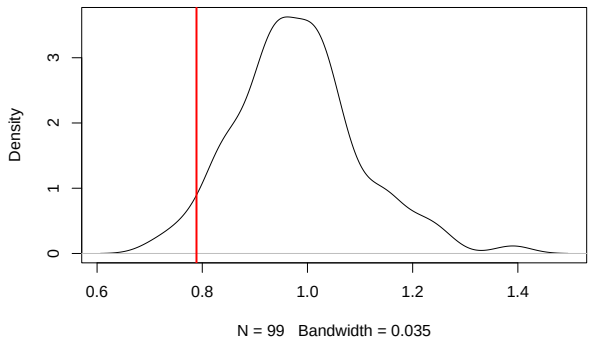
The null hypothesis here is that the values are distributed following a random process or have negative spatial autocorrelation (C >= 1), and the alternative hypothesis is that the values are distributed with positive spatial autocorrelation (C $<$ 1). The p-value is calculated using a Monte Carlo simulation, from which we derive a density plot of the C values from each permutation and calculate the number of times our simulated C value is greater than or equal to the observed value out of all the trials. A monte carlo simulation is the best method for this because it is robust to irregularly shaped polygons.

```
(gearyc <- autocor(columbus$HOVAL, ww, "geary"))
```

House Value

```
## [1] 0.7889937
```

```
## Monte Carlo sim to test for significance
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$HOVAL), ww, "geary")
})
plot(density(m), main = NA); abline(v=gearyc, col = "red", lwd = 2)
```




```
## p-value
sum(m <= gearyc) / 100
```

```
## [1] 0.04
```

No significant spatial autocorrelation (geary's c = 0.7889937, p > 0.05).

```
(gearyc <- autocor(columbus$INC, ww, "geary"))
```

Household Income

```
## [1] 0.7137603
```

```
## Monte Carlo sim to test for significance
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$INC), ww, "geary")
})

## p-value
sum(m <= gearyc) / 100
```

```
## [1] 0
```

Significant spatial autocorrelation (geary's c = 0.7137603, p < 0.05).

```
(gearyc <- autocor(columbus$CRIME, ww, "geary"))
```

Crime

```
## [1] 0.5916113
```

```
## Monte Carlo sim to test for significance
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$CRIME), ww, "geary")
})

## p-value
sum(m <= gearyc) / 100
```

```
## [1] 0
```

Significant spatial autocorrelation (geary's c = 0.5916113, p < 0.05).

```
(gearyc <- autocor(columbus$OPEN, ww, "geary"))
```

Open space

```
## [1] 0.878182
```

```
## Monte Carlo sim to test for significance
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$OPEN), ww, "geary")
})

## p-value
sum(m <= gearyc) / 100
```

```
## [1] 0.28
```

No significant spatial autocorrelation (geary's c = 0.878182, p > 0.05).

```
(gearyc <- autocor(columbus$PLUMB, ww, "geary"))
```

Plumbing

```
## [1] 0.6806864
```

```r
## Monte Carlo sim to test for significance
m <- sapply(1:99, function(i) {
    autocor(sample(columbus$PLUMB), ww, "geary")
})

## p-value
sum(m <= gearyc) / 100
```

```
## [1] 0.05
```

Significant spatial autocorrelation (geary's c = 0.6806864, p < 0.05).

Compare Moran's I and Geary's C

Reinhard Furrer (Furrer and Applied Statistics Group 2021) suggests to take 1-C to compare it to Moran's I more easily.

	Morans_I	signif	Gearys_C	One_minus_Gearys_C	signif.1
House value	0.221	*	0.789	0.211	
Income	0.412	*	0.714	0.286	*
Crime	0.515	*	0.592	0.408	*
Open space	-0.037		0.878	0.122	
Plumbing	0.455	*	0.681	0.319	*

There are differences observed in spatial autocorrelation in the data calculated with Moran's I and Geary's C. For both Moran's I and Geary's C there was not significant spatial autocorrelation in Open space. There was significant positive spatial autocorrelation in household income, crime, and plumbing using Moran's I and Geary's C. Housing value only had positive spatial autocorrelation using Moran's I. However the trends are the same (both find weak to moderate positive spatial correlation)

Spatial regression models

Constant means

We decide to model the `HOVAL` (home value) variable for all of our models. The simplest, yet naive, model is the constant means model, which is essentially an intercept-only model, i.e. the average of the `HOVAL` variable.

$$Y_i = \mu + \varepsilon_i$$

Where Y_i is the value of a home in 1000s of dollars, μ is the mean home value and ϵ_i are the individual deviations from the mean, which we assume to be be i.i.d. distributed.

$$\hat{Y}_i = \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

Where $\hat{Y}_i$ is the estimated value of a home in 1000s of dollars.

We estimate the model using `lm` function (using the `mean()` function yields the same result):

```
# make spatial vector to simple feature
columbus.sf <- sf::st_as_sf(columbus)
zero.means <- lm(HOVAL ~ 1, data=columbus.sf)
summary(zero.means)
```

```
##
## Call:
## lm(formula = HOVAL ~ 1, data = columbus.sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.536 -12.736  -4.936   4.864  57.964
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   38.436      2.638   14.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.47 on 48 degrees of freedom
```

We find the average home value is $38436. Since we will use the log-transformed home value for subsequent models, we repeat the zero means model for the transformed variable for purposes of comparison.

```
# estimate log-transformed model
zero.means.log <- lm(log(HOVAL) ~ 1, data=columbus.sf)
summary(zero.means.log)
```

```
##
## Call:
## lm(formula = log(HOVAL) ~ 1, data = columbus.sf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66751 -0.30582 -0.04076  0.21585  1.01620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.55231    0.06178    57.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4324 on 48 degrees of freedom
```

We find that the model estimates `log(HOVAL)` to be 3.55231, which is $\$\ 3.4894 \times 10^4$ after exponentiating.

Linear Model with Independent Residuals

Clearly, the zero means model is rather simplistic. To improve, we model the home value `HOVAL` as a linear function of its (non-spatial) covariates with i.i.d. errors.

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

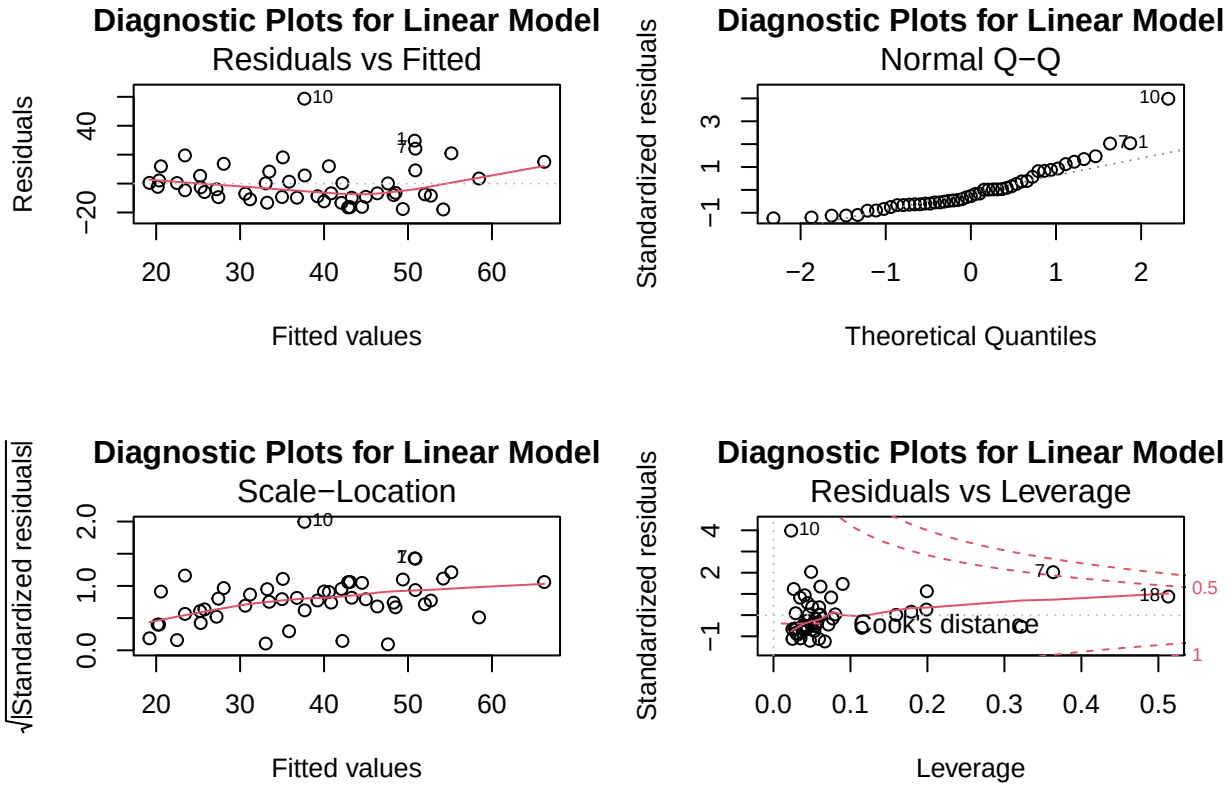
where Y is a vector of home values in 1000s of dollars, X is a matrix of the predictors `INC` (income), `CRIME`, `OPEN` (open space in neighborhood), and `CP` (whether the neighborhood is in the center or periphery). β is a vector of coefficients and ε is a vector of random, normally distributed errors.

We estimate the model with

```
col.lm <- lm(HOVAL~INC+CRIME+OPEN, data=columbus.sf)
summary(col.lm)
```

```
##
## Call:
## lm(formula = HOVAL ~ INC + CRIME + OPEN, data = columbus.sf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.902  -9.296  -3.969   5.608  58.742
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  46.7984    12.9397   3.617 0.000751 ***
## INC           0.4946     0.5316   0.930 0.357130
## CRIME        -0.5024     0.1795  -2.800 0.007509 **
## OPEN          0.7858     0.4677   1.680 0.099839 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.92 on 45 degrees of freedom
## Multiple R-squared:  0.3879, Adjusted R-squared:  0.3471
## F-statistic: 9.506 on 3 and 45 DF,  p-value: 5.588e-05
```

```
par(mfrow=c(2,2))
plot(col.lm, main = "Diagnostic Plots for Linear Model")
```



The diagnostic plots suggests that the assumptions of regression are not satisfied, which is to be expected since we know that most areal data is spatially dependent.

Simultaneous Autoregressive Models (SAR)

SAR error model

To address the shortcomings of the linear model under the assumption of i.i.d. errors, we introduce **Simultaneous Autoregressive Models**. The models solve simultaneously for the regression coefficients and for the autoregressive error structure. In the **Spatial Error Model**, spatial autocorrelation enters in the specification only through the error terms.

To derive the model, we formulate the error as a first-order spatial autoregressive process

$$\varepsilon = \lambda W \varepsilon + u \tag{1}$$

where ε is the error term of a standard regression model, λ is the autoregressive parameter, W is the row-standardised spatial weights matrix W (that is, the weights are standardised such that $\Sigma_j W_{ij} = 1 \quad \text{for all} \quad i$), and u_i a random error term, assumed to be i.i.d. If $|\lambda| < 1$ (to avoid the process exploding) and solving for ε yields

$$\varepsilon = (I - \lambda W)^{-1} u \tag{2}$$

We obtain the spatial error model by inserting ε into the standard regression model

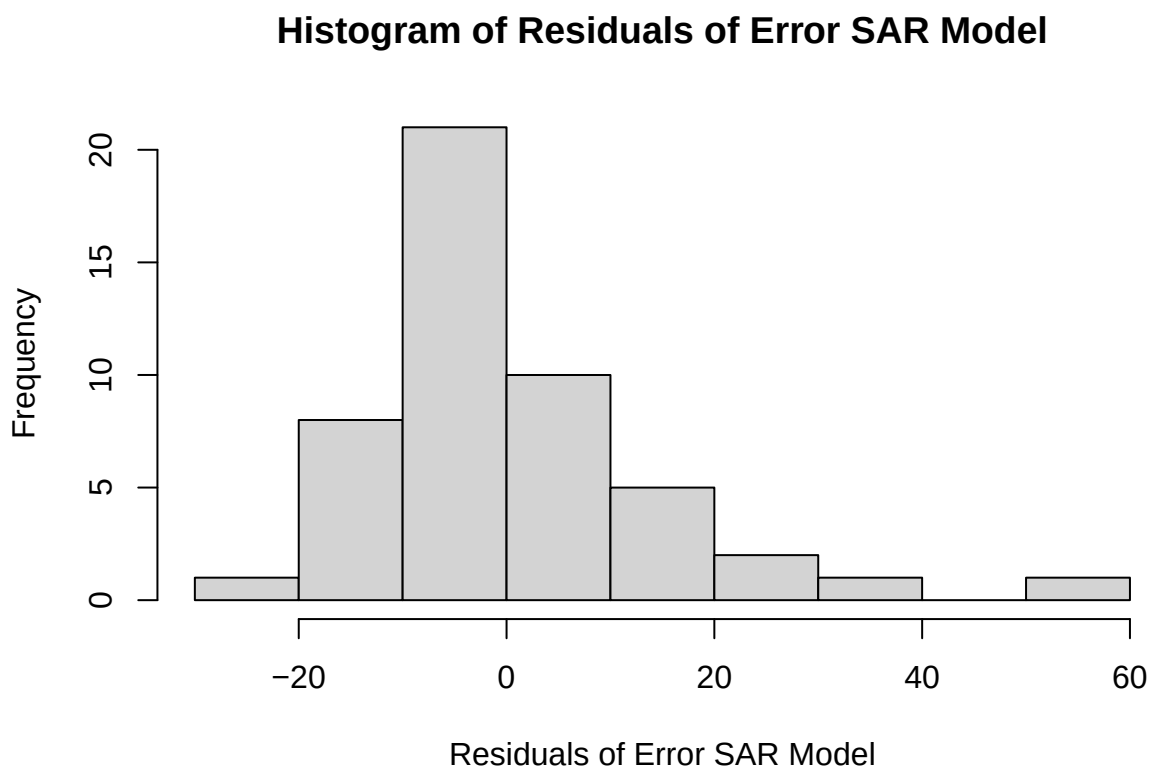
$$Y = X\beta + (I - \lambda W)^{-1}u \tag{3}$$

where Y is a vector of home values in 1000s of dollars, X is a matrix of the covariates `INC`, `CRIME`, `OPEN`, and `CP`. with constant variance $E[uu'] = \sigma^2 I$, which results in the following error variance-covariance matrix

$$E[\varepsilon\varepsilon'] = \sigma^2 (I - \lambda W)^{-1} (I - \lambda W')^{-1} \tag{4}$$

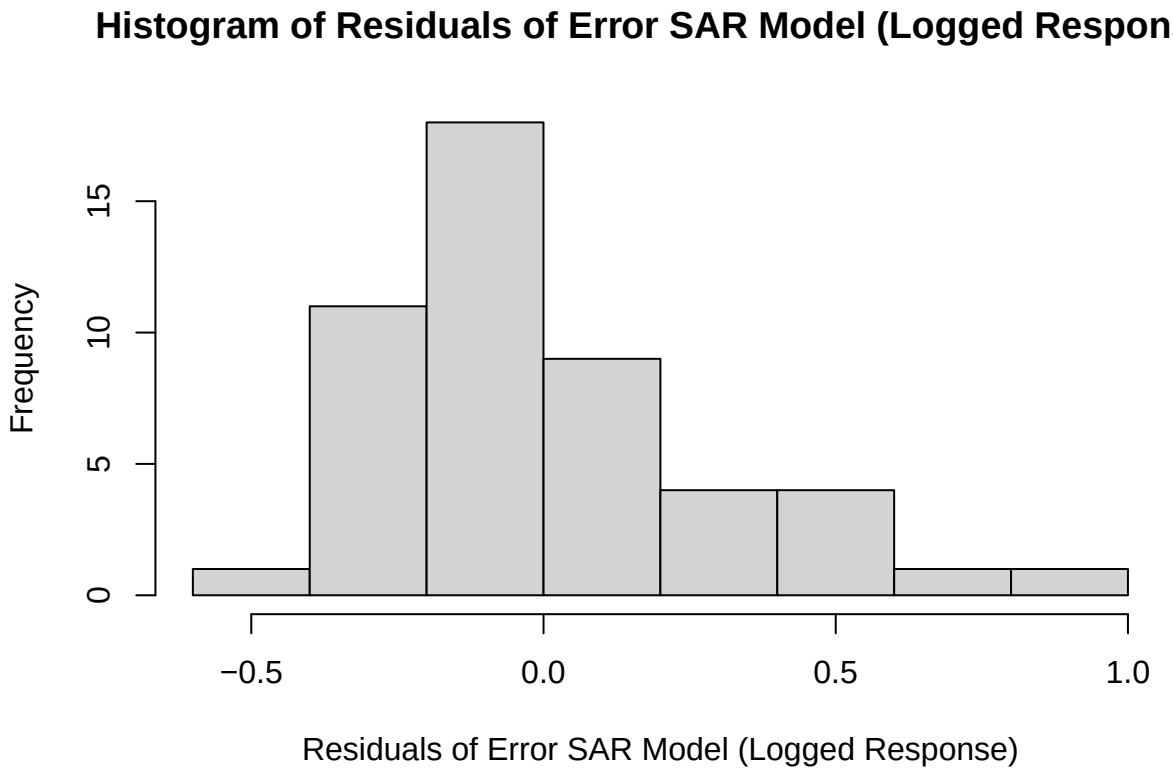
In order to estimate this model, we first create a list of spatial weights for neighbors. Then we estimate the model using the `errorsarlm` function from the `spatialreg` package.

```
# make simple feature to neighborhood
columbus.nb <- poly2nb(columbus.sf)

# make neighborhood to list of weights
lw <- nb2listw(columbus.nb, style="W")

# estimate error SAR model without transformation
col.errW.eig <- errorsarlm(HOVAL~INC+CRIME+OPEN+CP, data=columbus.sf,
 lw, method="eigen", quiet=T)

# look at the residuals
hist(residuals(col.errW.eig), main = "Histogram of Residuals of Error SAR Model", xlab = "Residuals of
```



The residuals are not normally distributed. We therefore, log-transform the response to normalize them.

```r
# estimate error SAR model with log transformation
col.errW.eig.log <- errorsarlm(log(HOVAL)~INC+CRIME+OPEN+CP, data=columbus.sf,
 lw, method="eigen", quiet=T)

hist(residuals(col.errW.eig.log), main = "Histogram of Residuals of Error SAR Model (Logged Response)",
```




```r
# print model summary
summary(col.errW.eig.log, correlation=TRUE)
```

```
##
## Call:
## errorsarlm(formula = log(HOVAL) ~ INC + CRIME + OPEN + CP, data = columbus.sf,
##     listw = lw, method = "eigen", quiet = T)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.503081 -0.190516 -0.051521  0.055445  0.855101
##
## Type: error
## Coefficients: (asymptotic standard errors)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.7832959  0.2525868 14.9782  < 2e-16
## INC          0.0134509  0.0104770  1.2839  0.19919
## CRIME       -0.0099247  0.0042443 -2.3383  0.01937
## OPEN         0.0194551  0.0086313  2.2540  0.02420
```

```
## CP          -0.2520210  0.1435112 -1.7561  0.07907
##
## Lambda: 0.45625, LR test value: 5.3082, p-value: 0.021226
## Asymptotic standard error: 0.15476
##     z-value: 2.9482, p-value: 0.0031967
## Wald statistic: 8.6916, p-value: 0.0031967
##
## Log likelihood: -9.062429 for error model
## ML residual variance (sigma squared): 0.08037, (sigma: 0.2835)
## Number of observations: 49
## Number of parameters estimated: 7
## AIC: 32.125, (AIC for lm: 35.433)
##
##  Correlation of coefficients
##             sigma lambda (Intercept) INC   CRIME OPEN
## lambda      -0.20
## (Intercept)  0.00  0.00
## INC          0.00  0.00  -0.85
## CRIME        0.00  0.00  -0.67        0.38
## OPEN         0.00  0.00   0.04       -0.18  0.03
## CP           0.00  0.00  -0.09        0.20 -0.50 -0.17
```

We find that only `CRIME` and `OPEN` are significant at the $\alpha = 0.05$ level. For every additional major theft (residential burglaries and vehicle theft) per 1000 households, the predicted home value decreases by approximately \$1010 ($e^{0.0099247} = 1.009974$), holding other variables constant. For every additional unit (which is not provided in the description) of open space, the home value is expected to increase by \$1020 ($e^{0.0194551} = 1.019646$), holding other variables constant.

We also check whether there is some spatial dependence within the residuals of the model by performing Moran's I test. The test tests H_0: There is no spatial dependence against H_1: There is spatial dependence. We reject H_0 if the test statistic $< \alpha = 0.05$.

```
# Run Moran's I test for col.errW.eig.log
moran.test(residuals(col.errW.eig.log), lw) # not significant
```

```
##
##  Moran I test under randomisation
##
## data:  residuals(col.errW.eig.log)
## weights: lw
##
## Moran I statistic standard deviate = 0.36942, p-value = 0.3559
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.013061791      -0.020833333       0.008418433
```

With a p-value of 0.3559, we fail to reject H_0. There does not seem to be any spatial dependence present in the residuals of our error model.

Conditional Autoregressive Models

In the next step, we implement the conditional autoregressive (CAR) model on the Columbus data to study the impact of covariates on House values in Columbus, OH, 1980 with spatial information.

The CAR model essentially assumes the spatial estimation is conditional on the value of neighbors. As a typical Bayes model, CAR model assumes prior distribution on the model parameters and applies computationally intensive sampling techniques like Markov Chain Monte Carlo (MCMC) or MCMC with Gibbs sampling to find the fitted parameters.

Since we have one response variable *House value*, the CAR model has only one random effect ϕ_k at each spatial location k. Because *House value* is continuous, Gaussian distribution is preferred for the CAR model and we take logarithm of the *House value* as well to make it normally distributed.

The package *CARBayes* by Duncan Lee is used to apply the CAR model in R. The specific function we use is *S.CARleroux*, where it specifies a CAR model proposed by Brian G. Leroux in 2000. The model expression is as follows.

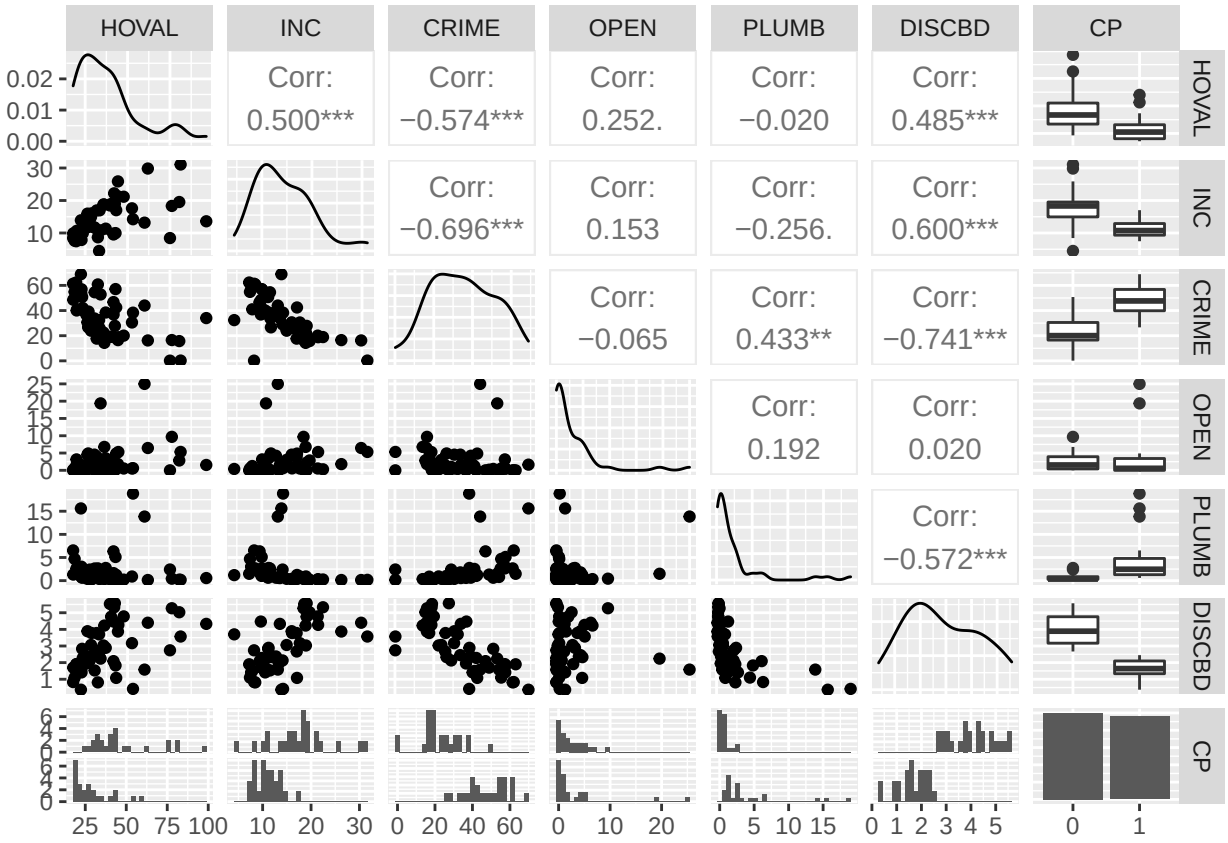
- $\mathbf{W}$ is a 0-1 neighborhood or weight matrix
- $k = 1, \ldots, K$ is the index for a certain areal unit
- ρ is a spatial correlation parameter, with $\rho = 0$ means indenpendence and $\rho = 1$ indicates strong spatial correlation

$$\psi_k = \phi_k$$

$$\phi_k \mid \phi_{-k}, \mathbf{W}, \tau^2, \rho \sim \mathrm{N}\left(\frac{\rho \sum_{i=1}^{K} w_{ki}\phi_i}{\rho \sum_{i=1}^{K} w_{ki} + 1 - \rho}, \frac{\tau^2}{\rho \sum_{i=1}^{K} w_{ki} + 1 - \rho}\right)$$

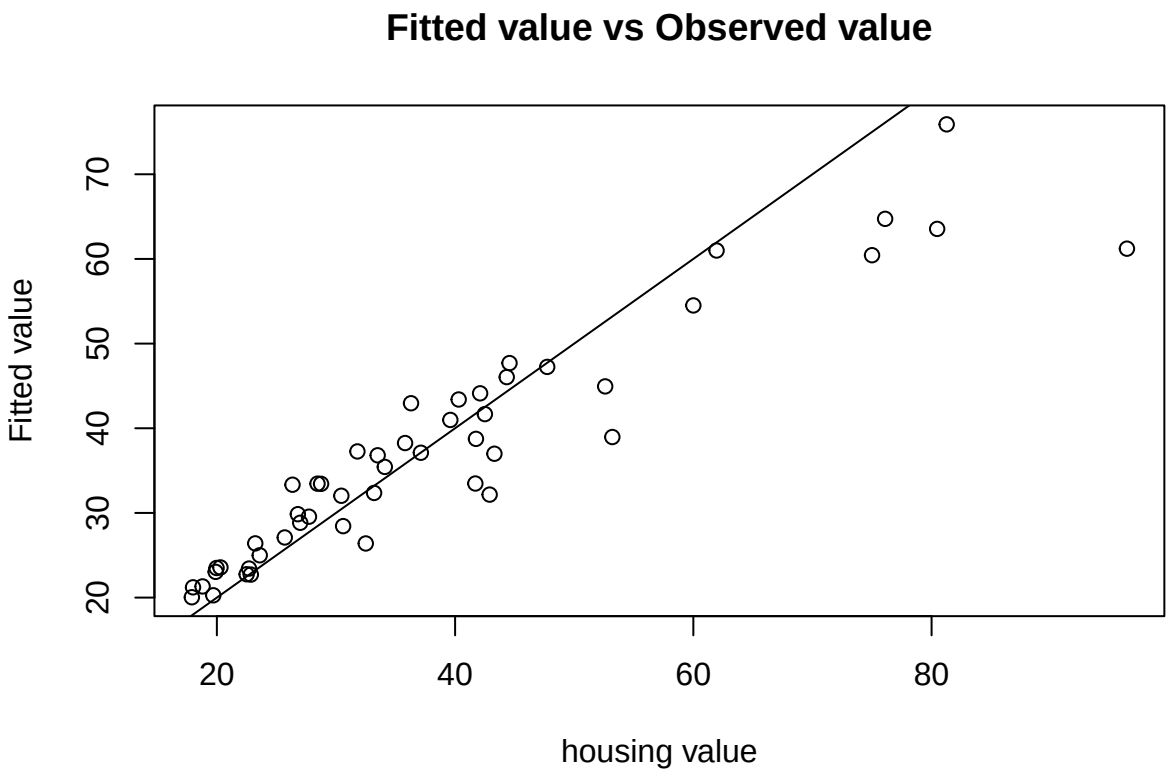
$$\tau^2 \sim \text{Inverse} - \text{Gamma}(a, b)$$

$$\rho \sim \text{ Uniform } (0, 1)$$



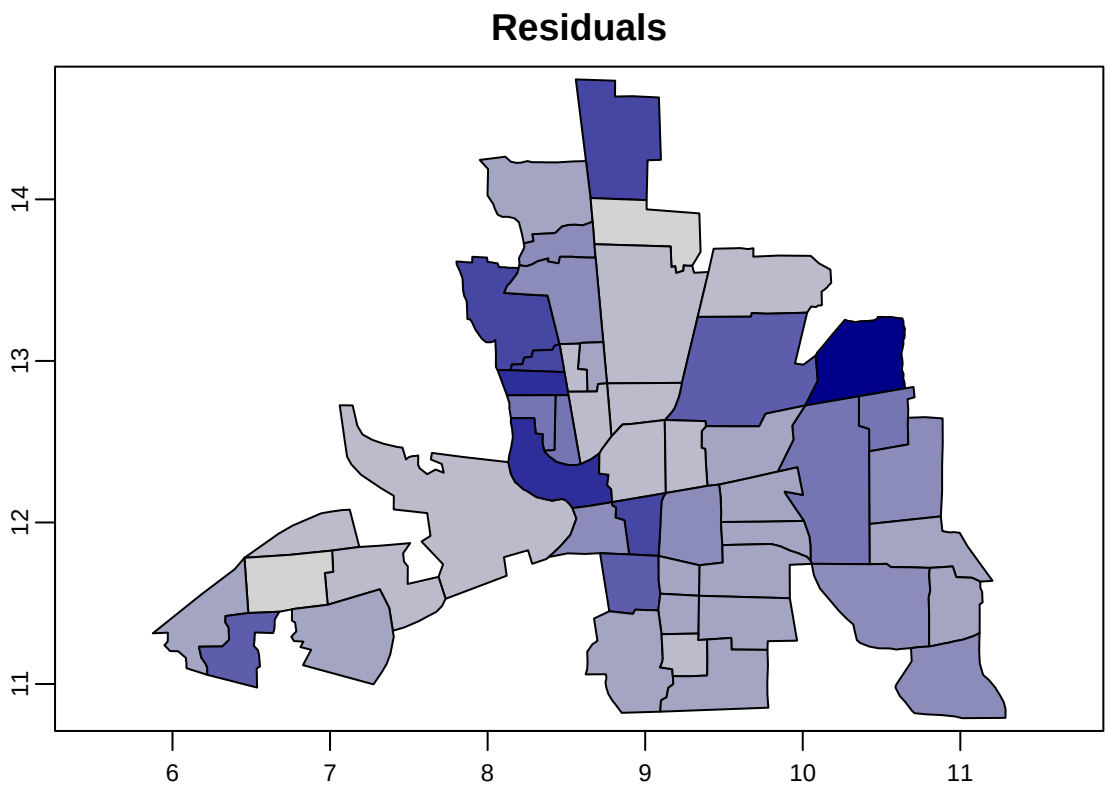
```r
set.seed(2021)
(car_model_gaussian = CARBayes::S.CARleroux(log(HOVAL)~INC+CRIME+OPEN+DISCBD+CP, data = df.columbus_CAR
```

```
##
## #################
## #### Model fitted
## #################
## Likelihood model - Gaussian (identity link function)
## Random effects model - Leroux CAR
## Regression equation - log(HOVAL) ~ INC + CRIME + OPEN + DISCBD + CP
## Number of missing observations - 0
##
## #############
## #### Results
## #############
## Posterior quantities and DIC
##
##              Median    2.5%  97.5% n.effective Geweke.diag
## (Intercept)  3.7227  2.9525 4.4936     11253.8        -0.5
## INC          0.0147 -0.0084 0.0371     10672.1         0.6
## CRIME       -0.0097 -0.0189 0.0000     12029.9         0.7
## OPEN         0.0201  0.0003 0.0400     10658.0        -0.9
## DISCBD      -0.0002 -0.1345 0.1352      9998.4        -0.3
## CP1         -0.1957 -0.5607 0.1764     12099.2         0.5
## nu2          0.0366  0.0032 0.1346       831.7         0.1
## tau2         0.1004  0.0032 0.2920       986.1        -0.1
## rho          0.3919  0.0340 0.8936     10148.1         0.3
##
## DIC =  -26.93797       p.d =  4.499935       LMPL =  -10.04
```



The log likelihood of this model is 17.97 while the training root mean square error (RMSE) is 52.6. From the table of 95% confidence intervals for coefficients, we have

- *CRIME* and *OPEN* are the two variables that is significant since their confidence intervals don't have zero involved.
- Hold other predictors fixed, regions with **less crimes** tend to have higher *House values*.
- Hold other predictors fixed, regions with **higher household incomes** tend to have higher *House values*.
- Hold other predictors fixed, regions with **more open area** tend to have higher *House values*.
- Hold other predictors fixed, regions with **closer distance to CBD** tend to have higher *House values*.
- Hold other predictors fixed, **core** regions tend to have higher *House values*.




```
# Moran I
Moran.I(car_model_gaussian$residuals[,1], ww)$p.value
```

```
## [1] 0.09593303
```

We also tested the Moran's I autocorrelation coefficient for the residuals. It shows there is **no** significant spatial correlation of the residuals since the p-value $p = 0.1 > 0.05$. Therefore, our CAR model fits the areal data nicely and leaves no significant spatial information in the residuals.

Conclusions

From our exploratory analysis and visualizing the data we know the variables of interest, with the exception of open space, have weak to moderate positive spatial autocorrelation, meaning neighboring polygons are more similar to each other than the dataset as a whole. All of the models we tried did not have significant ($\alpha = 0.05$) spatial autocorrelation.

We found that across all of our models average house value significantly increased with open space (units unknown). In some of the models (SAR, DURBIN, CAR) housing value decreased significantly with crime (per 1000 households). In the durbin log model and CAR model the effect of being located in an urban (core) neighborhood was also negative.

The best fit model by log likelihood was the CAR model. There was no significant difference in fit between the SAR lag and Durbin models.

References

Bivand, Roger. 2021. *Spdep: Spatial Dependence: Weighting Schemes, Statistics*. https://github.com/r-spatial/spdep/.

Bivand, Roger S., Edzer Pebesma, and Virgilio Gomez-Rubio. 2013a. *Applied Spatial Data Analysis with R, Second Edition*. Springer, NY. https://asdar-book.org/.

———. 2013b. *Applied Spatial Data Analysis with R, Second Edition*. Springer, NY. https://asdar-book.org/.

Bivand, Roger, Giovanni Millo, and Gianfranco Piras. 2021. "A Review of Software for Spatial Econometrics in R." *Mathematics* 9 (11). https://doi.org/10.3390/math9111276.

Bivand, Roger, Jakub Nowosad, and Robin Lovelace. 2021. *spData: Datasets for Spatial Analysis*. https://nowosad.github.io/spData/.

Bivand, Roger, and David W. S. Wong. 2018. "Comparing Implementations of Global and Local Indicators of Spatial Association." *TEST* 27 (3): 716–48. https://doi.org/10.1007/s11749-018-0599-x.

Furrer, Reinhard, and the Applied Statistics Group. 2021. "Modeling Dependent Data: An Excursion." http://user.math.uzh.ch/furrer/download/sta330/script_sta330.pdf.

Henry, Lionel, and Hadley Wickham. 2020. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

Hijmans, Robert J. 2021. *Terra: Spatial Data Analysis*. https://rspatial.org/terra/.

Hothorn, Torsten, Achim Zeileis, Richard W. Farebrother, and Clint Cummins. 2021. *Lmtest: Testing Linear Regression Models*. https://CRAN.R-project.org/package=lmtest.

Lee, Duncan. 2013. "CARBayes: An r Package for Bayesian Spatial Modeling with Conditional Autoregressive Priors." *Journal of Statistical Software* 55 (13): 1–24.

Müller, Kirill, and Hadley Wickham. 2021. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Paradis, Emmanuel, Simon Blomberg, Ben Bolker, Joseph Brown, Santiago Claramunt, Julien Claude, Hoa Sien Cuong, et al. 2021. *Ape: Analyses of Phylogenetics and Evolution*. http://ape-package.ird.fr/.

Paradis, E., and K. Schliep. 2019. "Ape 5.0: An Environment for Modern Phylogenetics and Evolutionary Analyses in R." *Bioinformatics* 35: 526–28.

Pebesma, Edzer. 2018. "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal* 10 (1): 439–46. https://doi.org/10.32614/RJ-2018-009.

———. 2021. *Sf: Simple Features for r*. https://CRAN.R-project.org/package=sf.

Pebesma, Edzer J., and Roger S. Bivand. 2005. "Classes and Methods for Spatial Data in R." *R News* 5 (2): 9–13. https://CRAN.R-project.org/doc/Rnews/.

Pebesma, Edzer, and Roger Bivand. 2021. *Sp: Classes and Methods for Spatial Data*. https://CRAN.R-project.org/package=sp.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Schloerke, Barret, Di Cook, Joseph Larmarange, Francois Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg, and Jason Crowley. 2021. *GGally: Extension to Ggplot2*. https://CRAN.R-project.org/package=GGally.

Ver Hoef, Jay M, Ephraim M Hanks, and Mevin B Hooten. 2018. "On the Relationship Between Conditional (CAR) and Simultaneous (SAR) Autoregressive Models." *Spatial Statistics* 25: 68–85.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

———. 2019. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2021a. *Forcats: Tools for Working with Categorical Variables (Factors)*. https://CRAN.R-project.org/package=forcats.

———. 2021b. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

———. 2021c. *Tidyverse: Easily Install and Load the Tidyverse*. https://CRAN.R-project.org/package=tidyverse.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software*

4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2021. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Jim Hester. 2021. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Zeileis, Achim, and Gabor Grothendieck. 2005. "Zoo: S3 Infrastructure for Regular and Irregular Time Series." *Journal of Statistical Software* 14 (6): 1–27. https://doi.org/10.18637/jss.v014.i06.

Zeileis, Achim, Gabor Grothendieck, and Jeffrey A. Ryan. 2021. *Zoo: S3 Infrastructure for Regular and Irregular Time Series (z's Ordered Observations)*. https://zoo.R-Forge.R-project.org/.

Zeileis, Achim, and Torsten Hothorn. 2002. "Diagnostic Checking in Regression Relationships." *R News* 2 (3): 7–10. https://CRAN.R-project.org/doc/Rnews/.