Lab: Spatial Regression Modeling

This lab is designed to provide the intuition behind spatial autoregressive models, specifically the **spatial lag** and **spatial error** models.

The **data** are derived from several different sources:

- Zillow Inc. real estate estimates
  - median neighborhood home price per square foot (`price`)
- Satellite remote sensing observations
  - normalized difference vegetation index (`ndvi`)
  - land surface temperature (`lst`)
  - open space fraction (`open_space_fraction`)
  - tree canopy cover (`tcc`)
- American Community Survey
  - median number of rooms (`median_number_rooms`)
  - median age of home (`median_age_home`)
  - median age of residents (`median_age`)
  - proportion of residents with bachelors degree (`attained_bachelors`)
  - population density in thousands of people per square km (`popden`)
  - median household income in thousands of dollars (`mhhi_family`)
  - proportion of residents that identify as white (`white`)

The original motiviation for this analysis was to identify the economic effects of *environmental attributes* (NDVI, LST, TCC, open space) on *home values* in Zillow neighborhoods. The full study included all major metropolitan areas in the United States, but this abbreviated activity will focus on a single city - Houston, Texas - in order to simplify computation.

Load packages

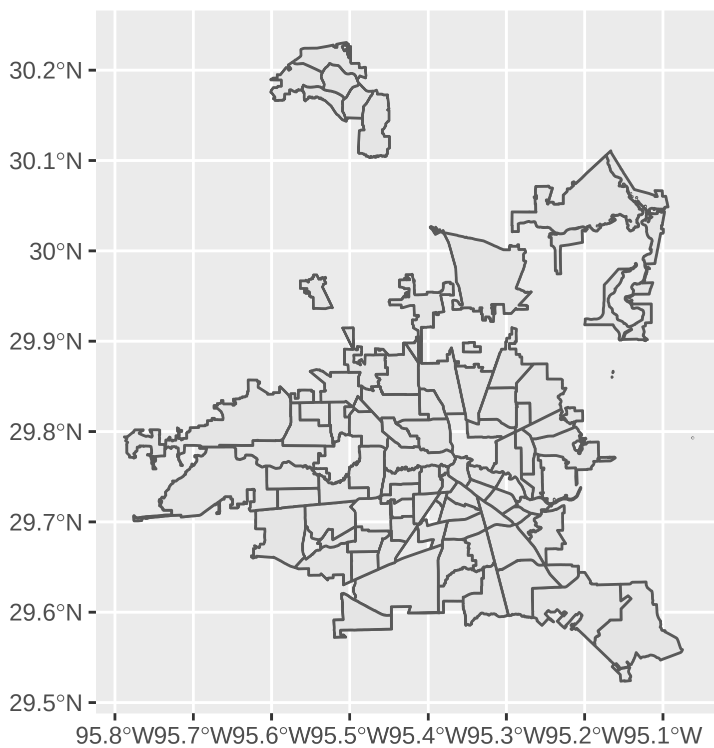
```
# regular suite of tidyverse packages
library(tidyverse)
library(broom)
library(knitr)
library(patchwork) #organize plots in a grid

#visualize spatial data
library(RColorBrewer) #custom color palettes

#wrangle and model spatial data
library(sf)
library(spatialreg)
library(spdep)
```

Load the Data

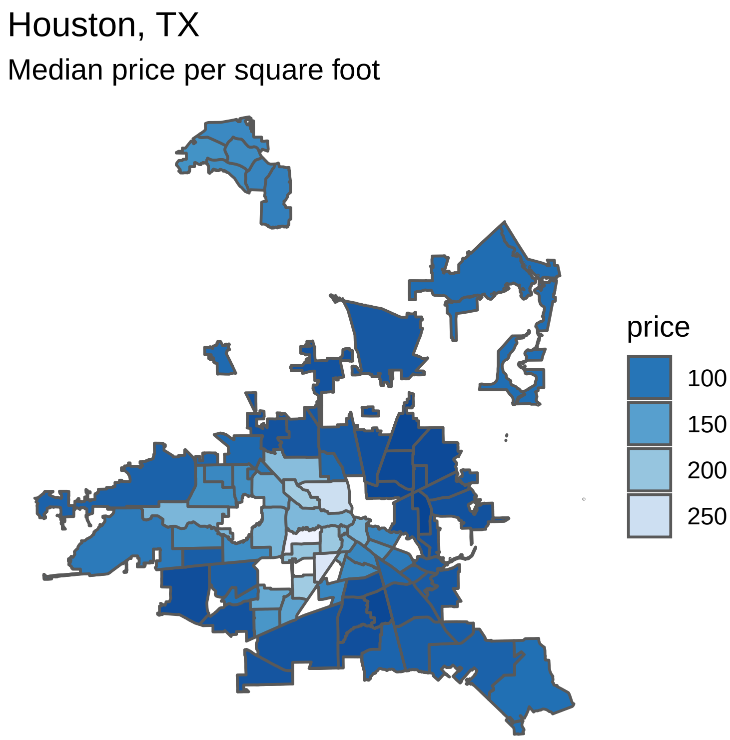
```r
# read the shapefile
shape <- read_sf(dsn = "data", layer = "Zillow_Houston")

# convert RegionID to numeric before we join anddrop some columns that we don't need
shape <- shape %>%
  mutate(RegionID = as.character(RegionID),
         RegionID = as.numeric(RegionID)) %>%
  dplyr::select(-fid, -State, -County, -City, -Name, -layer, -path, -population)

# load the rest of the data
big_df <- read_csv("data/full_dataset.csv")

# merge keeping only those in both data sets
merged <- inner_join(shape, big_df, by = "RegionID")
```

Visualize the data

```r
ggplot(data = merged) +
  geom_sf()
```

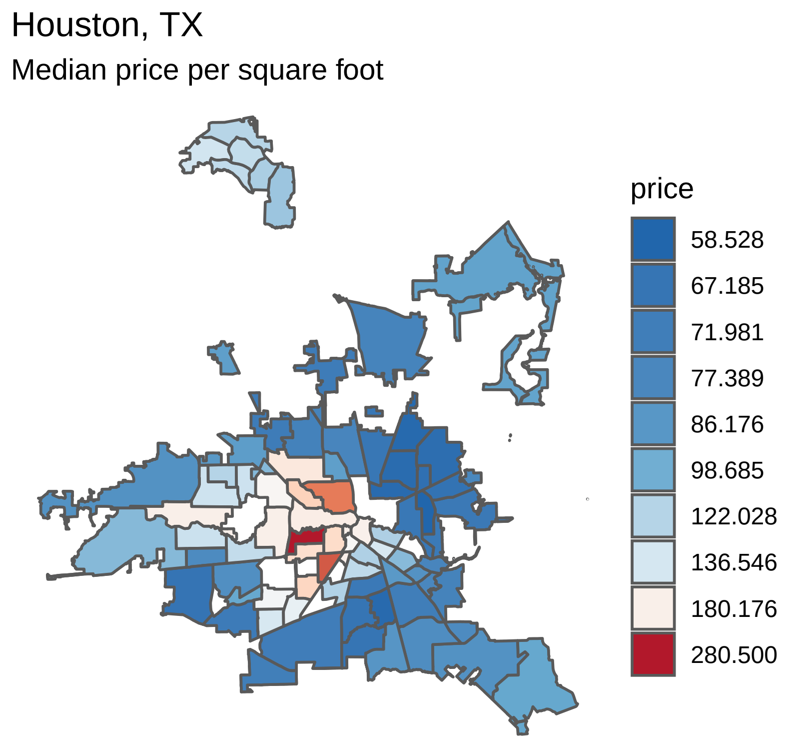



```r
# plot median home price per square foot add formatting
ggplot(data = merged, aes(fill = price)) +
  geom_sf() +
  labs(title = "Houston, TX",
       subtitle = "Median price per square foot") +
  theme_void() +
  scale_fill_distiller(guide = "legend")
```



It is a bit difficult to discern differences at the lower price levels. Let's plot again using quantile breaks in the data. We'll also use a better color palette.

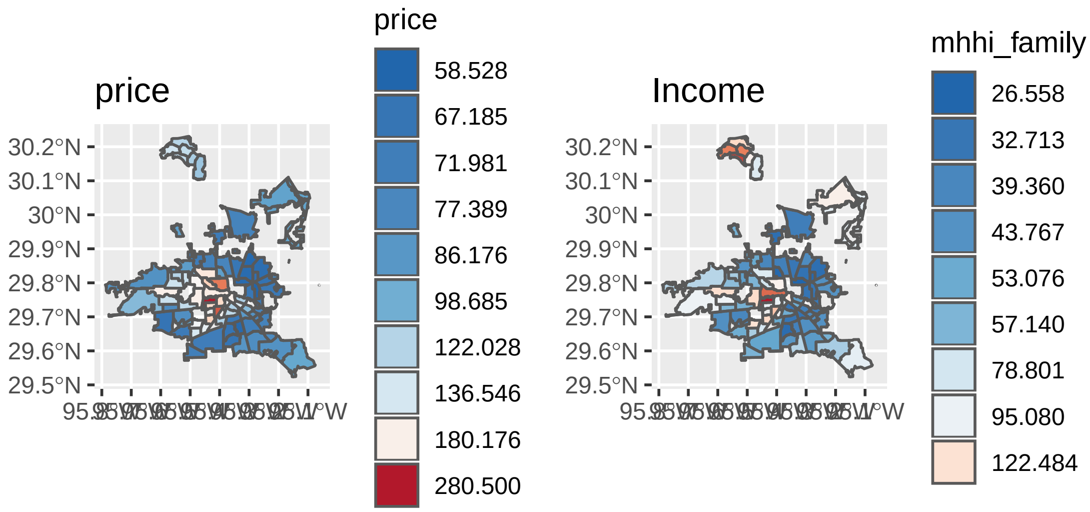
```r
# determine number of quantiles
grps <- 10

# compute the quantiles
brks <- quantile(merged$price, 0:(grps-1)/(grps-1), na.rm=TRUE, names = FALSE)
brks <- round(brks, 3)

# plot with color scale adjusted for quantiles
ggplot(data = merged, aes(fill = price)) +
  geom_sf() +
  labs(title = "Houston, TX",
       subtitle = "Median price per square foot") +
  theme_void() +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)
```



Exercise 1: Make a plot for each of the following variables:

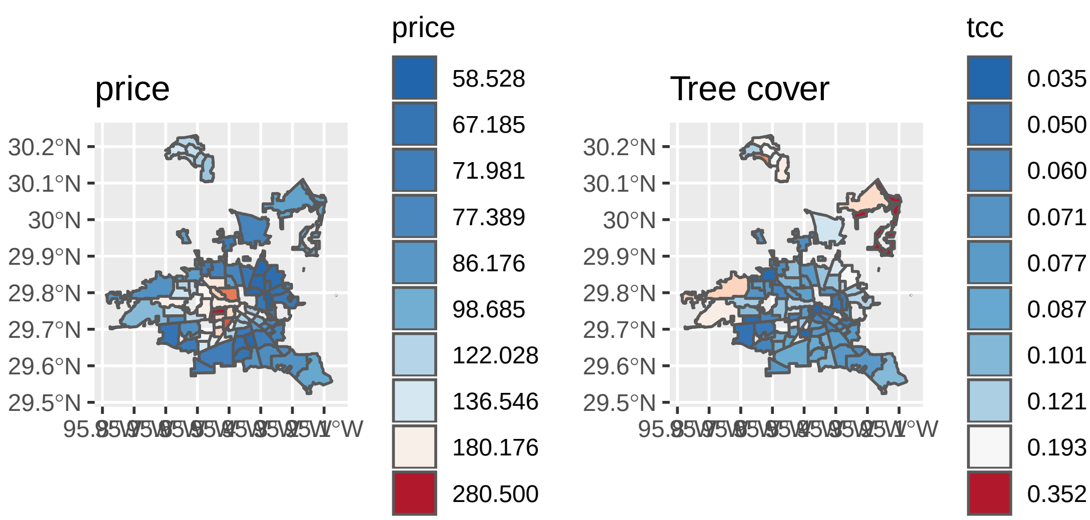
- median household income ('mhhi_family'),
- tree canopy cover ('tcc')
- land surface temperature ('lst')
- population density ('popden')

```
# determine number of quantiles
grps <- 10

# compute the quantiles
brks <- quantile(merged$price, 0:(grps-1)/(grps-1), na.rm=TRUE, names = FALSE)
brks <- round(brks, 3)

# plot with color scale adjusted for quantiles
p1 <- ggplot(data = merged, aes(fill = price)) +
  geom_sf() +
  labs(title = "price") +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)

# compute the quantiles
brks <- quantile(merged$mhhi_family, 0:(grps-1)/(grps-1), na.rm=TRUE, names = FALSE)
brks <- round(brks, 3)

# plot with color scale adjusted for quantiles
p2 <- ggplot(data = merged, aes(fill = mhhi_family)) +
  geom_sf() +
  labs(title = "Income") +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)

p1 + p2
```



Price and income appear to be moderately correlated.

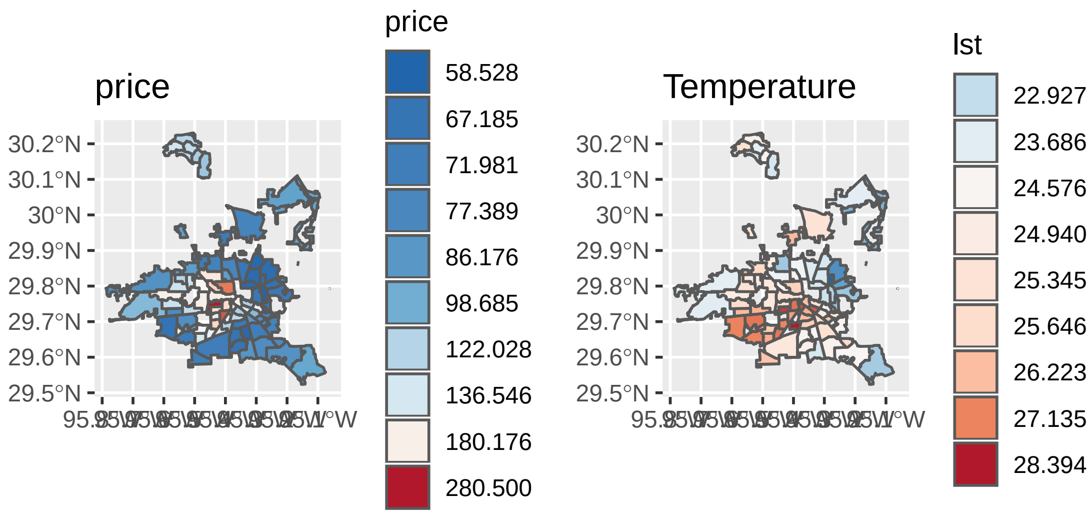
```
# compute the quantiles
brks <- quantile(merged$tcc, 0:(grps-1)/(grps-1), na.rm=TRUE, names = FALSE)
brks <- round(brks, 3)

# plot with color scale adjusted for quantiles
p3 <- ggplot(data = merged, aes(fill = tcc)) +
  geom_sf() +
  labs(title = "Tree cover") +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)

p1 + p3
```



Price and tree cover do not appear to be correlated.

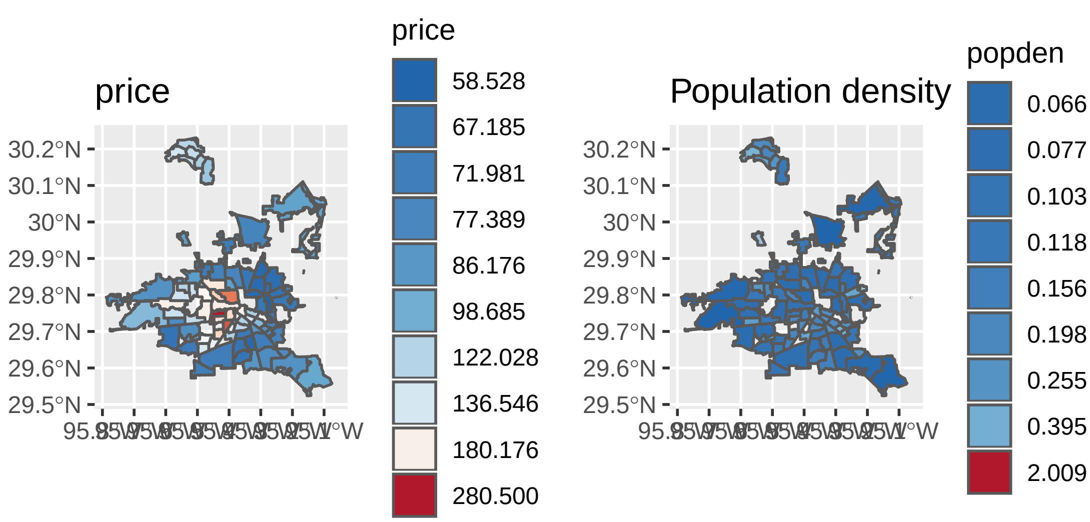
```
  # compute the quantiles
brks <- quantile(merged$lst, 0:(grps-1)/(grps-1), na.rm=TRUE, names = FALSE)
brks <- round(brks, 3)

# plot with color scale adjusted for quantiles
p4 <- ggplot(data = merged, aes(fill = lst)) +
```

```r
  geom_sf() +
  labs(title = "Temperature") +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)

p1 + p4
```



Temperature and price appear to be mildly correlated.

```r
  # compute the quantiles
brks <- quantile(merged$popden, 0:(grps-1)/(grps-1), na.rm=TRUE, names = FALSE)
brks <- round(brks, 3)

# plot with color scale adjusted for quantiles
p5 <- ggplot(data = merged, aes(fill = popden)) +
  geom_sf() +
  labs(title = "Population density") +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)

p1 + p5
```



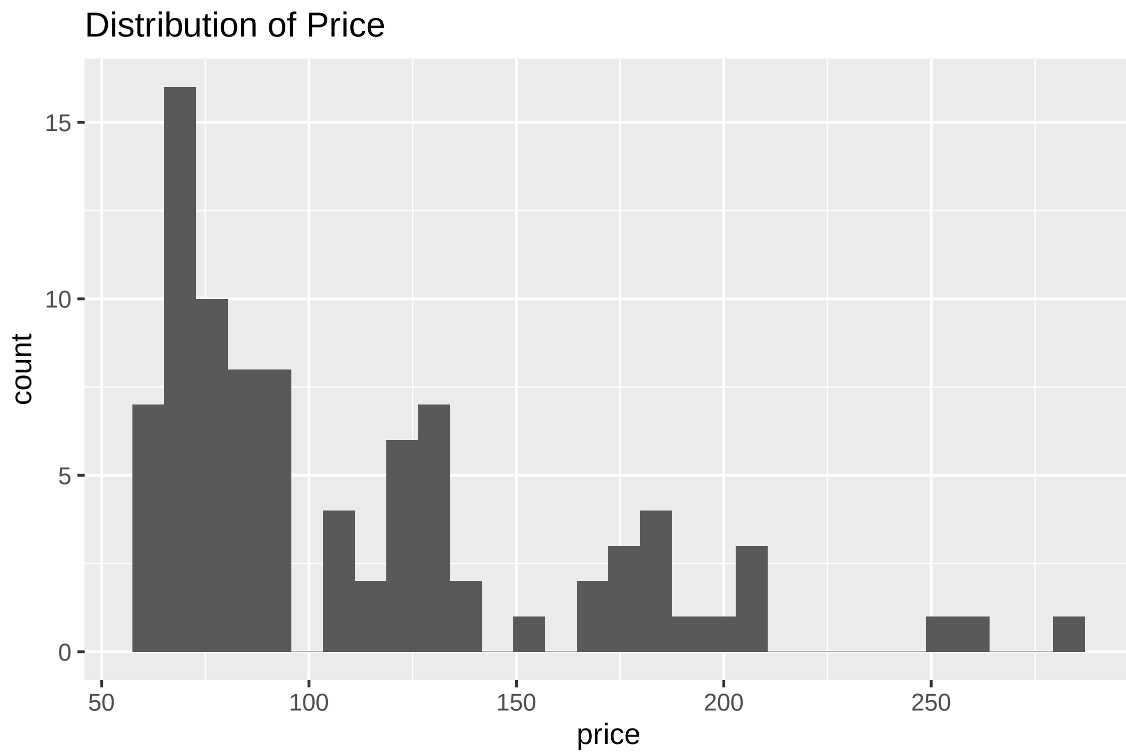
Price and population density do not appear to be correlated.

Compare these plots to the modified plot above for median price per square foot. Does there appear to be correlation between any of the variables and the median home price per square foot? Briefly explain your response.

Build a simple model

Your task is to model the median home price per square foot as a function of the other variables in the dataset. Let's check the distribution of the response variable ('price').

```
ggplot(data = merged, aes(x = price)) +
  geom_histogram() +
  labs(title = "Distribution of Price")
```



Next, let's fit a regression model where the response variable is `price` and the predictors are socio-demographic and environmental variables.

```
# ordinary least-squares model
m1 <- lm(price ~ median_number_rooms
         + median_age_home
         + median_age
         + attained_bachelors
         + mhhi_family
         + popden
         + white
         + ndvi
         + tcc
         + lst
         + open_space_fraction,
         data = merged)
```

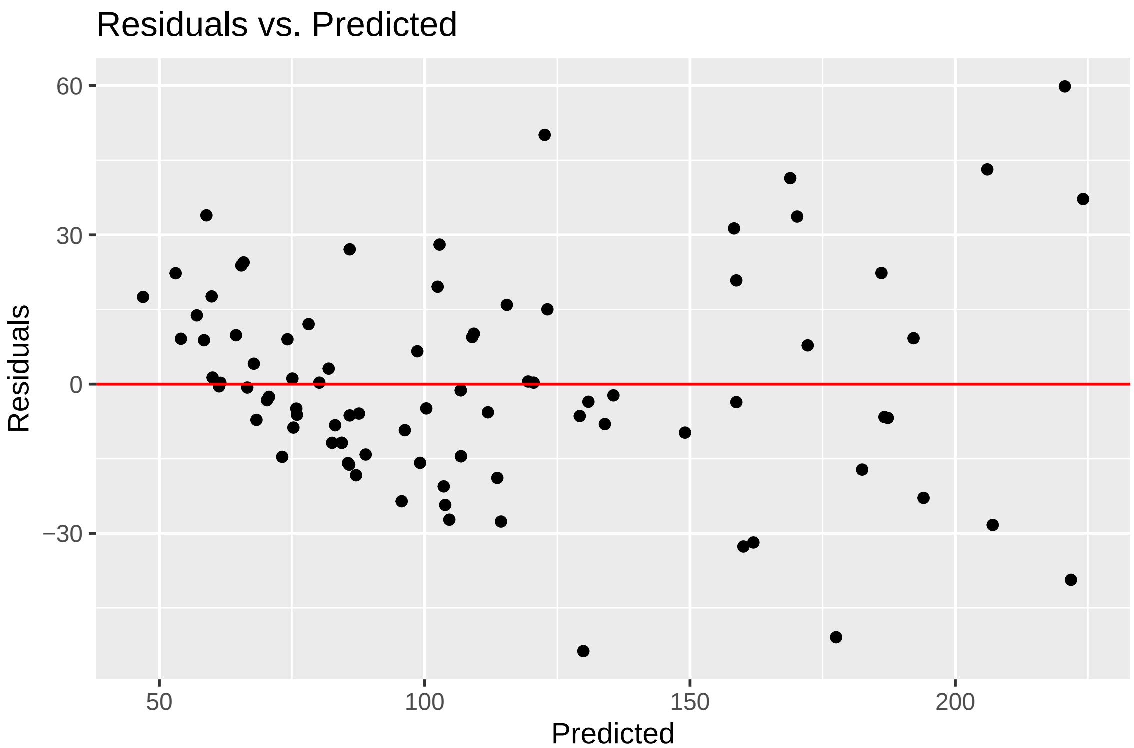
```r
tidy(m1) %>%
  kable(format = "markdown", digits = 4)
```

term	estimate	std.error	statistic	p.value
(Intercept)	112.8154	81.0687	1.3916	0.1681
median_number_rooms	-18.0424	5.0494	-3.5732	0.0006
median_age_home	1.8705	0.2948	6.3451	0.0000
median_age	-2.3231	0.8554	-2.7160	0.0082
attained_bachelors	259.1321	47.0443	5.5083	0.0000
mhhi_family	1.1001	0.1633	6.7347	0.0000
popden	11.4345	10.2518	1.1154	0.2682
white	-11.2170	18.3585	-0.6110	0.5430
ndvi	-112.1834	113.6854	-0.9868	0.3269
tcc	-50.6300	88.1282	-0.5745	0.5673
lst	0.8838	2.5429	0.3475	0.7291
open_space_fraction	50.8945	64.4888	0.7892	0.4325

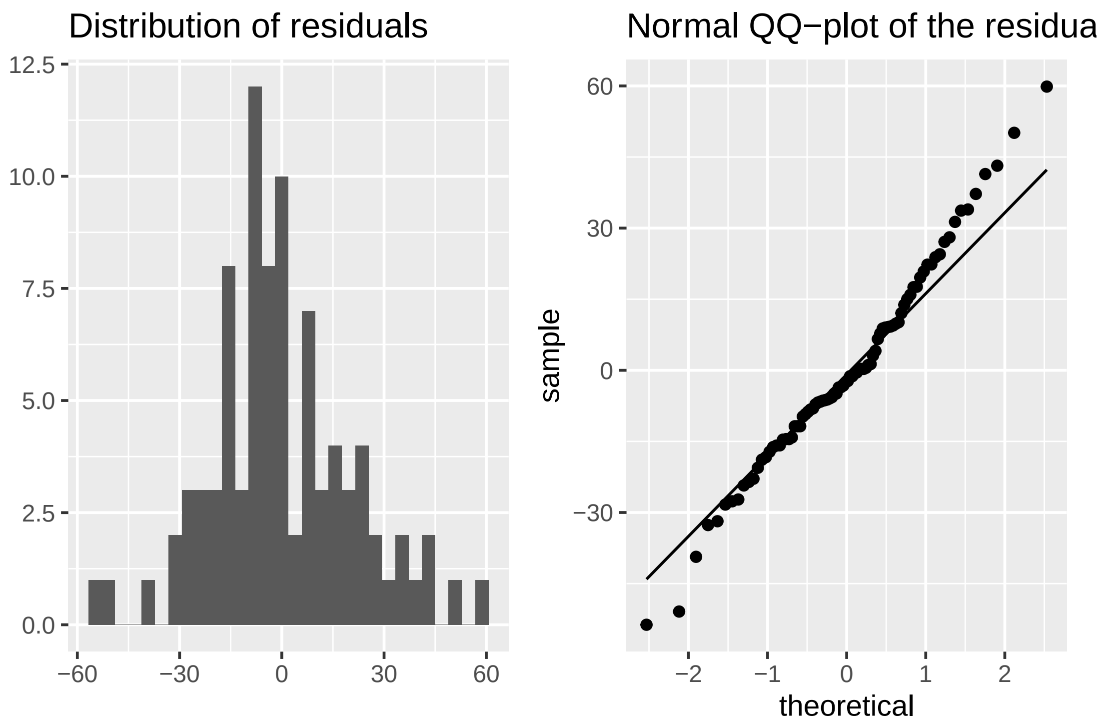
Below are some of the residual plots we need to check the model assumptions.

```r
# add model residuals to the data
merged <- merged %>%
  mutate(resid = resid(m1),
         pred = predict(m1))
```

```r
ggplot(data = merged, aes(x = pred, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residuals vs. Predicted",
       x = "Predicted",
       y = "Residuals")
```



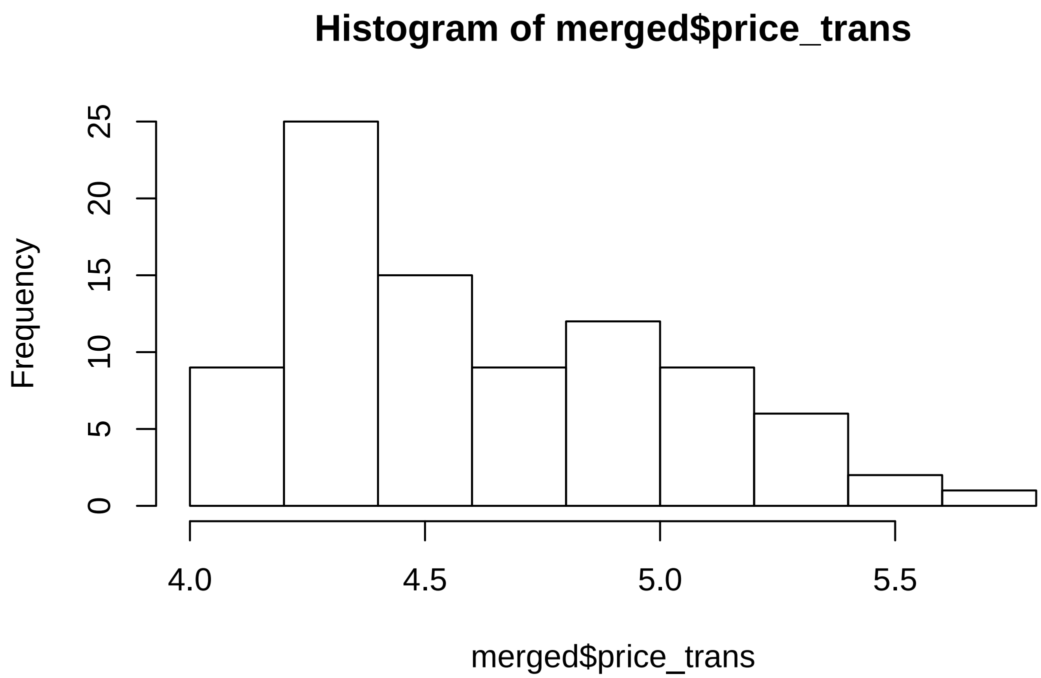

```r
p1 <- ggplot(data = merged, aes(x = resid)) +
  geom_histogram() +
  labs(title = "Distribution of residuals",
       x = "", y = "")

p2 <- ggplot(data = merged, aes(sample = resid)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Normal QQ-plot of the residuals")

#arrange plots using patchwork package
p1 + p2
```



Exercise 2: Which assumption(s) appear to be violated based on the plots of the residuals? How can we transform the response variable `price` to address the violation in assumption(s)? Show your code below to create a new variable called `price_trans` that is the transformed version of the response variable `price`.

There is evidence that the response variable is skewed (not normally distributed). We can see from the histogram that the response variable is right-skewed. Furthermore, the plot of residuals vs. predicted has a "fan" shape, which is evidence that the response variable is not normally distributed. We can log-transform the response variable to make it closer to normally-distributed.

```r
# log-transform the response variable
merged$price_trans <- log(merged$price)

# plot histogram of the new response variable
hist(merged$price_trans)
```



Exercise 3: Refit the previous model with the transformed response variable, `price_trans`, created in Exercise 2. Show your code and model output.

```
# ordinary least-squares model
m1 <- lm(price_trans ~ median_number_rooms
         + median_age_home
         + median_age
         + attained_bachelors
         + mhhi_family
         + popden
         + white
         + ndvi
         + tcc
         + lst
         + open_space_fraction,
         data = merged)
```

```
tidy(m1) %>%
  kable(format = "markdown", digits = 4)
```

term	estimate	std.error	statistic	p.value
(Intercept)	4.3373	0.6066	7.1502	0.0000
median_number_rooms	-0.1189	0.0378	-3.1465	0.0024
median_age_home	0.0130	0.0022	5.9154	0.0000
median_age	-0.0189	0.0064	-2.9499	0.0042
attained_bachelors	2.3168	0.3520	6.5818	0.0000
mhhi_family	0.0068	0.0012	5.5469	0.0000
popden	0.1182	0.0767	1.5412	0.1274
white	0.0718	0.1374	0.5230	0.6025
ndvi	-0.6715	0.8506	-0.7894	0.4323
tcc	-0.4081	0.6594	-0.6189	0.5378
lst	0.0139	0.0190	0.7281	0.4688

term	estimate	std.error	statistic	p.value
open_space_fraction	0.3169	0.4825	0.6568	0.5133

Exercise 4: Interpret the output from the ordinary least squares model created in the previous exercise. Which variables are statistically significant? What is their estimated effect on the response variable?

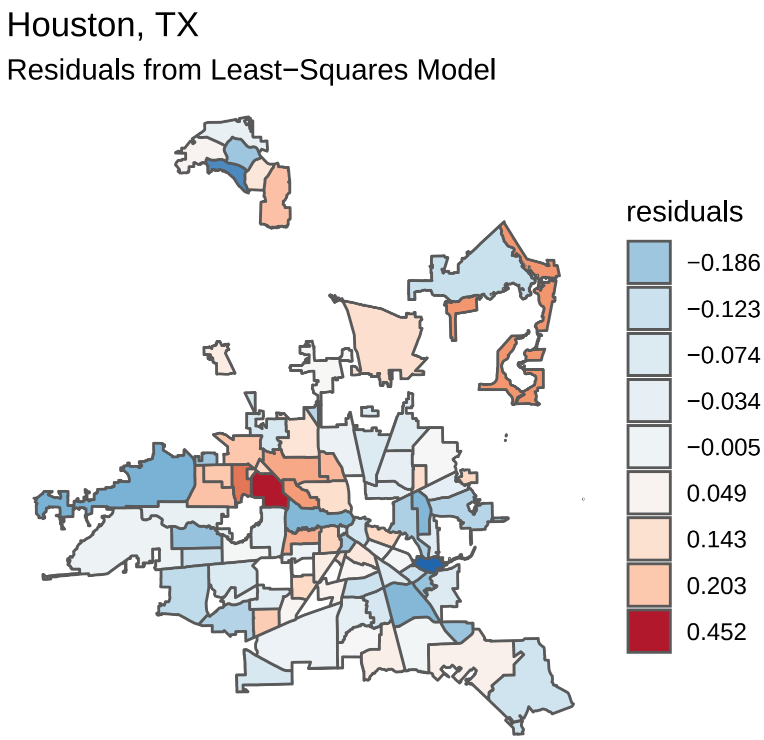
The statistically significant variables with a positive effect on price are median home age, bachelors degree, and median household income. The statistically significant variables with a negative effect on price are median number of rooms and median age.

Exercise 5: Add a new column called `residuals` to the `merged` dataset that contains the residuals from the model in Exercise 3.

```
# add model residuals to the data
merged <- merged %>%
  mutate(residuals = resid(m1),
         pred = predict(m1))
```

Next, let's make an assessment about the independence assumption by looking at the residuals distributed in space. If the residuals appear to be randomly distributed, then there is no spatial autocorrelation. If the errors are **not** randomly distrubuted in space, then we need to test for spatial autocorrelation. ‘

```
# plot the residuals on the map broken down by quantiles
grps <- 10
brks <- quantile(merged$residuals, 0:(grps-1)/(grps-1), na.rm=TRUE,
                 names = FALSE)
brks <- round(brks, 3)

ggplot(data = merged, aes(fill = residuals)) +
  geom_sf() +
  labs(title = "Houston, TX",
       subtitle = "Residuals from Least-Squares Model") +
  theme_void() +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)
```



Exercise 6: If there was no spatial correlation, i.e. the residuals were randomly distributed in space, what would you expect the map to look like? Based on this, do you think the model residuals are randomly distributed in space? What might be a mechanism for this phenomenon? (In other words, why might the median home price of one neighborhood affect the median home price of an adjacent neighborhood?)

If there was no spatial autocorrelation, then I would expect the red and blue polygons to be randomly distributed. Since the red and blue polygons do not appear to be randomly distributed (i.e., red is next to red, and blue is next to blue), then we can hypothesize that there is spatial autocorrelation.

A possible mechanism for the spatial autocorrelation is that wealthy neighborhoods are desirable. Homeowners would rather live near wealthy homeownevers compared to non-wealthy homeowners. Therefore, wealth itself begets adjacent wealth simply because homeowners are willing to pay a premium to be proximate to wealthy homeowners.

As we saw in the lecture, Moran's I test is a robust way to test for spatial autocorrelation. We can use the `spdep` package to calculate Moran's I for our model residuals. Once again, ideally there will be no spatial autocorrelation, i.e. a Moran's I value close to zero.

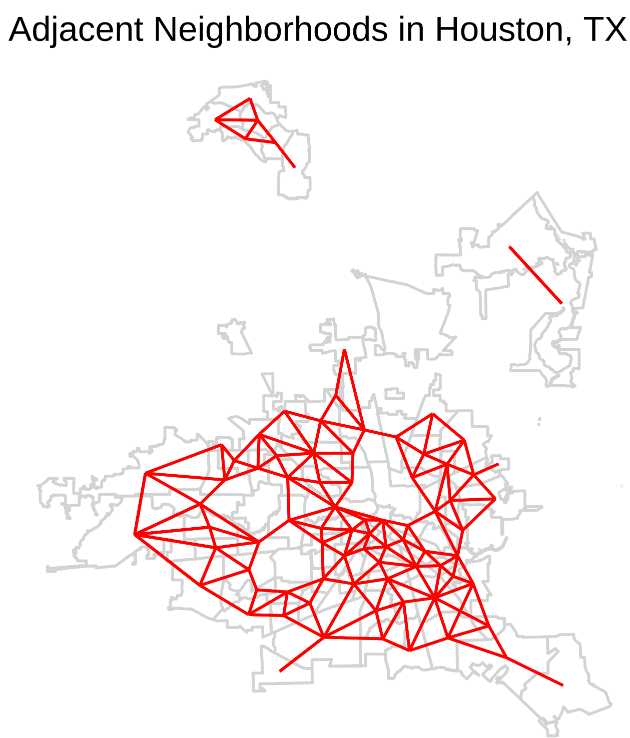
First, generate the neighborhood list object. The neighborhood list object determines which observations are adjacent to other observations.

```
# make a neighbor list using the sdep package
nb <- poly2nb(merged)
nb
```

```
## Neighbour list object:
## Number of regions: 88
## Number of nonzero links: 384
## Percentage nonzero weights: 4.958678
## Average number of links: 4.363636
## 2 regions with no links:
## 4 60
```

```
#make a data frame for neighbors
merged_sp <- as(merged, "Spatial")

nb_lines <- nb %>%
  nb2lines(coords = coordinates(merged_sp)) %>%
  as("sf") %>%
  st_set_crs(st_crs(merged))
```

```
# plot neighbors
ggplot(data = merged) +
  geom_sf(fill = "white", color = "lightgrey") +
  geom_sf(data = nb_lines, col = "red") +
  labs(title = "Adjacent Neighborhoods in Houston, TX") +
  theme_void()
```



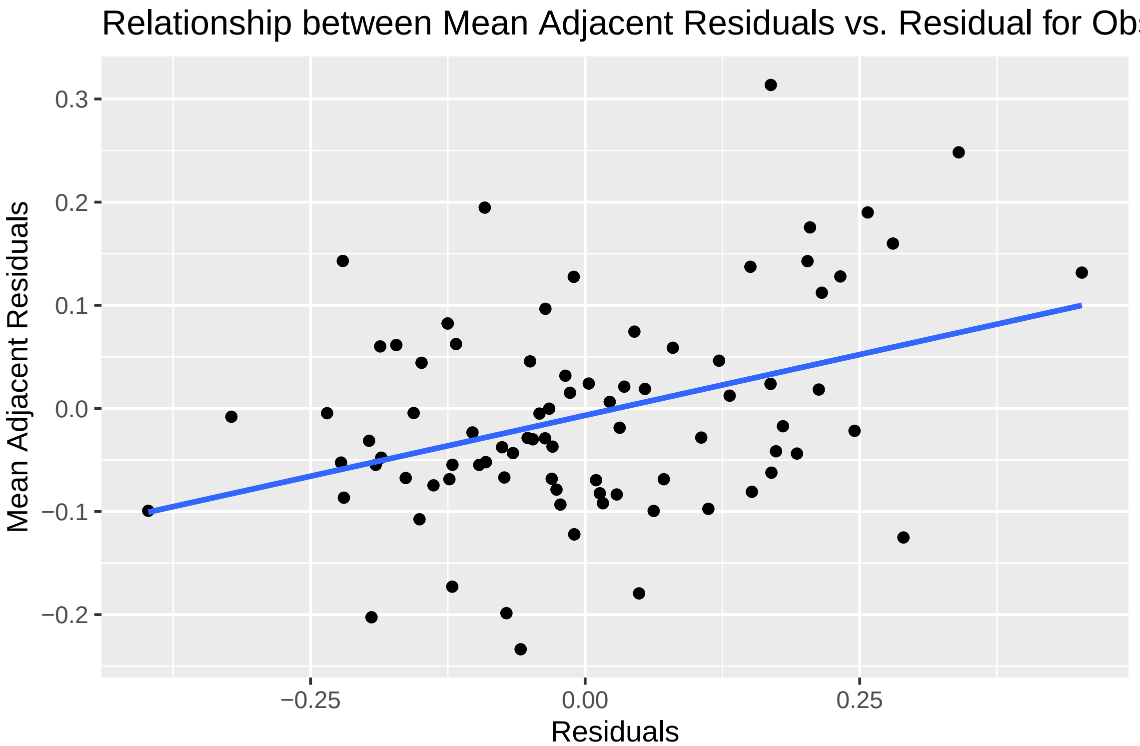
That's a lot of adjacent neighborhoods!

We already have an idea of whether or not the errors (model residuals) are correlated in space. Let's make one more plot to help us understand this correlation (or lack thereof).

```
# calculate the average neighborhing residual for each observation
resnb_calc <- sapply(nb, function(x) mean(merged$residuals[x]))

# add average neighboring residuals to merged data frame
merged <- merged %>%
  mutate(resnb = resnb_calc)
```

```
# plot the average neighboring residuals vs. residuals.
ggplot(data = merged, aes(x = residuals, y = resnb)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)+
  labs(x = "Residuals",
```

```
       y = "Mean Adjacent Residuals",
       title = "Relationship between Mean Adjacent Residuals vs. Residual for Observation i")
```



Now that we've built some intuition for spatial autocorrelation, let's calculate Moran's I. If the observed Moran's I is statistically greater than the null hypothesized value 0, then there is sufficient evidence to conclude that there is spatial autocorrelation.

```
# calculate weights matrix
ww <- nb2listw(nb, style = 'B', zero.policy = T) # binary weights matrix

# monte carlo Moran's test
moran.mc(merged$residuals, ww, 1000, zero.policy = T)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  merged$residuals
## weights: ww
## number of simulations + 1: 1001
##
## statistic = 0.28096, observed rank = 1000, p-value = 0.000999
## alternative hypothesis: greater
```

> Exercise 7: What is the test statisic? What is the p-value? Does Moran's I provide evidence for or against there being significnat spatial autocorrelation? Briefly explain your reasoning.

The test statistic is 0.28. The p-value is 0.001. Therefore, we conclude that Moran's I test provides evidence for significant spatial autocorrelation. Our accepted p-value cutoff is 0.05. Since the p-value is less than 0.05, we can conclude that the test statistic is significantly greater than the null hypothesis, i.e. we reject the null hypothesis.

Spatial regression models

We will introduce two different types of spatial regression models: the **spatial lag model** and the **spatial error model**. Both models are similar in that they both add a term to the right-hand side of equation that includes the spatial weights matrix W.

Consider a simple linear regression model:

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + \epsilon$$

where y is the response variable, x_1, x_2, etc. are the predictor variables, β_1, β_2, etc. are estimated coefficients, and ϵ is an uncorrelated error term.

The **spatial lag model** adds a term that is a product of W and the **response variable**. The **spatial lag model** would be:

$$y = \rho W y + \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + \epsilon$$

where W is the spatial weights matrix and ρ is an estimated coefficient.

The **spatial error model**, on the other hand, incorporates W into the **error term**:

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + \lambda W u + \epsilon$$

where λ is an estimated coefficient and u is a correlated spatial error term.

Let's try both models on our data and see if they address the issue of spatial autocorrelation.

Spatial lag model

```
m1_sp_lag <- lagsarlm(price_trans ~ median_number_rooms
         + median_age_home
         + median_age
         + attained_bachelors
         + mhhi_family
         + popden
         + white
         + ndvi
         + tcc
         + lst
         + open_space_fraction,
         data = merged,
         listw = ww,
         zero.policy = T)
summary(m1_sp_lag)
```

```
##
## Call:spatialreg::lagsarlm(formula = formula, data = data, listw = listw,
##     na.action = na.action, Durbin = Durbin, type = type, method = method,
##     quiet = quiet, zero.policy = zero.policy, interval = interval,
##     tol.solve = tol.solve, trs = trs, control = control)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.397930 -0.112939 -0.019878  0.116928  0.457808
```

```
##
## Type: lag
## Regions with no neighbours included:
##  4 60
## Coefficients: (asymptotic standard errors)
##                      Estimate Std. Error z value  Pr(>|z|)
## (Intercept)          4.4006353  0.5815320  7.5673 3.819e-14
## median_number_rooms -0.1226759  0.0361490 -3.3936 0.0006898
## median_age_home      0.0133203  0.0021427  6.2167 5.078e-10
## median_age          -0.0191651  0.0059746 -3.2078 0.0013378
## attained_bachelors   2.3633396  0.3438358  6.8735 6.267e-12
## mhhi_family          0.0068454  0.0011435  5.9864 2.146e-09
## popden               0.1108414  0.0732390  1.5134 0.1301728
## white                0.0676941  0.1278426  0.5295 0.5964508
## ndvi                -0.6719505  0.7897325 -0.8509 0.3948480
## tcc                 -0.4468291  0.6183008 -0.7227 0.4698809
## lst                  0.0130150  0.0177627  0.7327 0.4637318
## open_space_fraction  0.2927821  0.4516331  0.6483 0.5168075
##
## Rho: -0.001136, LR test value: 0.17598, p-value: 0.67485
## Asymptotic standard error: 0.0026917
##     z-value: -0.42205, p-value: 0.67299
## Wald statistic: 0.17813, p-value: 0.67299
##
## Log likelihood: 38.61698 for lag model
## ML residual variance (sigma squared): 0.024342, (sigma: 0.15602)
## Number of observations: 88
## Number of parameters estimated: 14
## AIC: -49.234, (AIC for lm: -51.058)
## LM test for residual autocorrelation
## test value: 15.852, p-value: 6.8493e-05
```

Moran's I of the spatial lag model:

```r
merged <- merged %>%
  mutate(residuals_lag = residuals(m1_sp_lag))
```

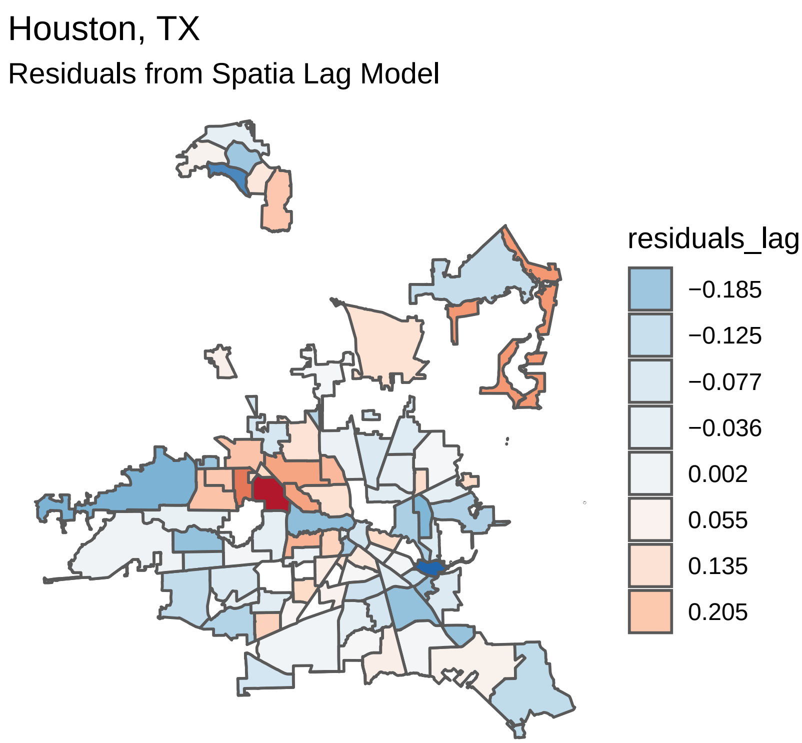
```r
moran.mc(merged$residuals_lag, ww, 1000, zero.policy = T)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  merged$residuals_lag
## weights: ww
## number of simulations + 1: 1001
##
## statistic = 0.28003, observed rank = 1001, p-value = 0.000999
## alternative hypothesis: greater
```

Plot the residuals:

```r
brks <- quantile(merged$residuals_lag, 0:(grps-1)/(grps-1), na.rm = TRUE,
                 names = FALSE)
brks <- round(brks, 3)
```

```r
ggplot(data = merged, aes(fill = residuals_lag)) +
  geom_sf() +
  labs(title = "Houston, TX",
       subtitle = "Residuals from Spatia Lag Model") +
  theme_void() +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)
```



Exercise 8: What can you conclude about the spatial autocorrelation of the original model compared to the lag model? Use your observations from the residuals plot and Moran's test to explain your reasoning.

Moran's I from the original model is 0.28. Moran's I from the lag model is also 0.28. Both of these statistics are significantly different than the null hypothesis. Therefore, we conclude that the spatial lag model does not address issues of spatial autocorrelation in the model residuals. However, it is important to note that we are measuring spatial autocorrelation in the model residuals (error). So, we expect that the spatial error model will have a much greater impact on the model error than the spatial lag model.

Spatial error model

```r
m1_sp_err = errorsarlm(price_trans ~ median_number_rooms
         + median_age_home
         + median_age
         + attained_bachelors
         + mhhi_family
         + popden
         + white
         + ndvi
         + tcc
         + lst
```

```r
         + open_space_fraction,
         data = merged,
         listw = ww,
         zero.policy = T)
summary(m1_sp_err)
```

```
##
## Call:spatialreg::errorsarlm(formula = formula, data = data, listw = listw,
##     na.action = na.action, Durbin = Durbin, etype = etype, method = method,
##     quiet = quiet, zero.policy = zero.policy, interval = interval,
##     tol.solve = tol.solve, trs = trs, control = control)
##
## Residuals:
##         Min          1Q      Median          3Q         Max
## -0.26792276 -0.09900193 -0.00020953  0.09316427  0.27844950
##
## Type: error
## Regions with no neighbours included:
##  4 60
## Coefficients: (asymptotic standard errors)
##                       Estimate Std. Error z value  Pr(>|z|)
## (Intercept)          4.0083232  0.6053790  6.6212 3.563e-11
## median_number_rooms -0.0621514  0.0335217 -1.8541  0.063730
## median_age_home      0.0113140  0.0019128  5.9149 3.320e-09
## median_age          -0.0110353  0.0049547 -2.2273  0.025930
## attained_bachelors   1.5814769  0.3055849  5.1752 2.276e-07
## mhhi_family          0.0045743  0.0010140  4.5110 6.451e-06
## popden               0.1715575  0.0595641  2.8802  0.003974
## white                0.2080597  0.1150580  1.8083  0.070559
## ndvi                -0.7402011  0.7348953 -1.0072  0.313829
## tcc                  0.0496272  0.6065570  0.0818  0.934792
## lst                  0.0112056  0.0191367  0.5856  0.558174
## open_space_fraction  0.3794105  0.3637519  1.0430  0.296926
##
## Lambda: 0.14901, LR test value: 20.089, p-value: 7.3902e-06
## Asymptotic standard error: 0.01494
##     z-value: 9.9734, p-value: < 2.22e-16
## Wald statistic: 99.468, p-value: < 2.22e-16
##
## Log likelihood: 48.57373 for error model
## ML residual variance (sigma squared): 0.01685, (sigma: 0.12981)
## Number of observations: 88
## Number of parameters estimated: 14
## AIC: -69.147, (AIC for lm: -51.058)
```

Moran's I of the spatial error model:

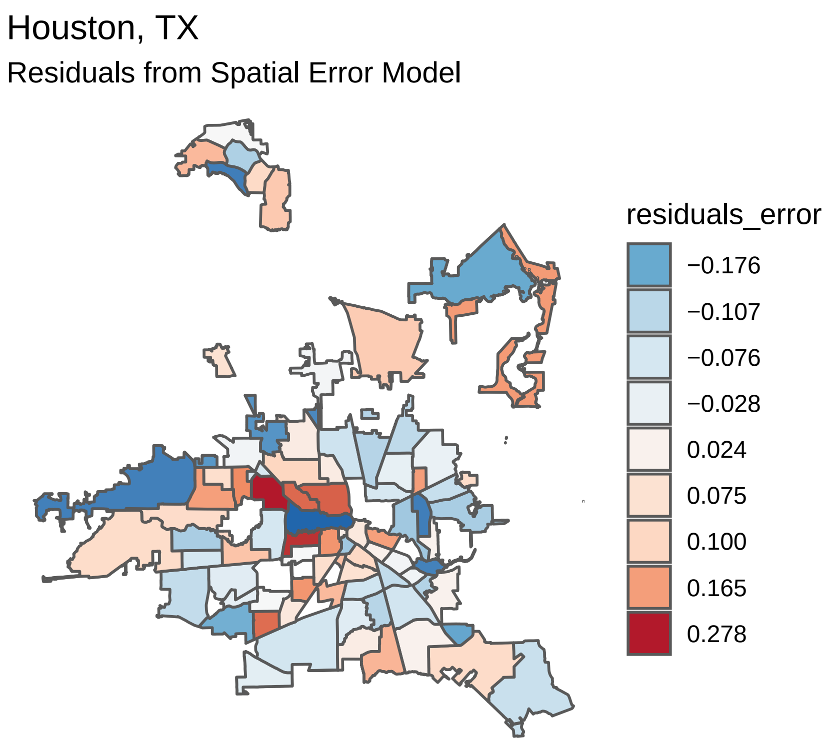
```r
merged <- merged %>%
  mutate(residuals_error = residuals(m1_sp_err))

moran.mc(merged$residuals_error, ww, 1000, zero.policy = T)
```

```
##
##  Monte-Carlo simulation of Moran I
##
```

```
## data:  merged$residuals_error
## weights: ww
## number of simulations + 1: 1001
##
## statistic = -0.05304, observed rank = 311, p-value = 0.6893
## alternative hypothesis: greater
```

Plot the residuals:

```
brks <- quantile(merged$residuals_error, 0:(grps-1)/(grps-1), na.rm=TRUE,
                 names = FALSE)
brks <- round(brks, 3)

ggplot(data = merged, aes(fill = residuals_error)) +
  geom_sf() +
  labs(title = "Houston, TX",
       subtitle = "Residuals from Spatial Error Model") +
  theme_void() +
  scale_fill_distiller(palette = 'RdBu', guide = "legend", breaks = brks)
```



> Exercise 9: Let's compare the three different models. - How does the spatial autocorrelation of the spatial error model compare to that of the original model and the spatial lag model? Use your observations from the residuals plot and Moran's test to explain your reasoning.

Moran's I tests from the first 2 models are significantly different from the null hypothesis. This suggests that the ordinary least squares model (1) and the spatial lag model (2) have statistically significant spatial autocorrelation in the residuals. The spatial error model (3), on the other hand, has a Moran's I statistic that is NOT statistically significant from the null hypothesis. Therefore, the spatial error model has addressed the issue of spatial autocorrelation in the error term.

> Exercise 10: Briefly describe how the coefficients of the predictor variables differ across the three models. How are the coefficients similar? How do the coefficients differ? Did anything surprise you?

The statistically significant effects have the same signs across all three models EXCEPT the error model has population density as a statistically significant positive effect, whereas this effect is not significant in the other models.

I am surprised that the environmental attributes were not statistically significant in any of the models.

> Exercise 11: Which model would you choose to explain variation `price` in the median house price in Houston, TX? Briefly explain your choice.

I would choose the spatial error model because it is the only model that has proved to addresses spatial autocorrelation.

> Exercise 12: There is a model in `spdep` that combines the spatial lag and spatial error models. It looks like this:
>
> $$y = \rho W y + X\beta + \lambda W u + \epsilon$$
>
> Implement this model using the function `sacsarlm`. You can use the code for the `lagsarlm` and `errorsarlm` models as a guide for the syntax. Comment on the coefficient estimates and their significance. Would you use this model versus the one you chose in the previous exercise? Briefly explain why or why not.

```
m1_sp_lag_err = sacsarlm(price_trans ~ median_number_rooms
         + median_age_home
         + median_age
         + attained_bachelors
         + mhhi_family
         + popden
         + white
         + ndvi
         + tcc
         + lst
         + open_space_fraction,
         data = merged,
         listw = ww,
         zero.policy = T)
summary(m1_sp_lag_err)
```

```
##
## Call:spatialreg::sacsarlm(formula = formula, data = data, listw = listw,
##     listw2 = listw2, na.action = na.action, Durbin = Durbin,
##     type = type, method = method, quiet = quiet, zero.policy = zero.policy,
##     tol.solve = tol.solve, llprof = llprof, interval1 = interval1,
##     interval2 = interval2, trs1 = trs1, trs2 = trs2, control = control)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.230891 -0.091531  0.016194  0.109917  0.257203
##
## Type: sac
## Coefficients: (asymptotic standard errors)
##                        Estimate  Std. Error z value  Pr(>|z|)
## (Intercept)          4.13884254  0.60766113  6.8111 9.685e-12
## median_number_rooms -0.05523255  0.03285312 -1.6812  0.092725
## median_age_home      0.01090567  0.00187558  5.8146 6.079e-09
## median_age          -0.00872364  0.00478737 -1.8222  0.068422
## attained_bachelors   1.46389507  0.29946784  4.8883 1.017e-06
```

```
## mhhi_family          0.00426998  0.00098158  4.3501 1.361e-05
## popden               0.15232426  0.05877226  2.5918  0.009548
## white                0.25805254  0.11234134  2.2970  0.021616
## ndvi                -0.84358706  0.71471352 -1.1803  0.237875
## tcc                  0.05278346  0.59261908  0.0891  0.929028
## lst                  0.00468988  0.01913256  0.2451  0.806359
## open_space_fraction  0.42937643  0.34771145  1.2349  0.216881
##
## Rho: -0.0068858
## Asymptotic standard error: 0.0038176
##     z-value: -1.8037, p-value: 0.071274
## Lambda: 0.16639
## Asymptotic standard error: 0.0096671
##     z-value: 17.212, p-value: < 2.22e-16
##
## LR test value: 22.31, p-value: 1.4306e-05
##
## Log likelihood: 49.68385 for sac model
## ML residual variance (sigma squared): 0.015506, (sigma: 0.12452)
## Number of observations: 88
## Number of parameters estimated: 15
## AIC: -69.368, (AIC for lm: -51.058)
```

The coefficients of the fourth model have the same signs and significance of the third model (spatial error) EXCEPT that the proportion of White residents is significant and positive in the fourth model, whereas it was not significant in the others. The fourth model has the most significant terms of all the models.

I would choose this model over the other models because it has the highest log-likelihood.