Individual Research Project - Geographically Weighted Regression

Jonathan Schierbaum

3/02/2021

Data description

Our dataset contains major building code violation rates at the census tract level in the City of Philadelphia. The file also contains demographic and socioeconomic neighborhood characteristics from the 2012-2016 American Community Survey.

The record layout is below.

Variables we'll use

- usarea: Number of major building code violations per area in square miles 2015-2017
- lmhhinc: Log median household income 2012-2016
- lpop: Log total population 2012-2016

- punemp: Percent of civilian labor force that are unemployed 2012-2016
- lmhval: Log median housing value 2012-2016
- pvac: Percent of housing units that are vacant 2012-2016
- ph70: Percent of housing units built before 1970 2012-2016
- phnew: Percent of housing units built 2014 and after 2012-2016
- phisp: Percent Hispanic 2012-2016
- pnhblk: Percent non-Hispanic black 2012-2016

Variables we won't

- OBJECTID: ID
- STATEFP10: State FIPS code
- COUNTYFP10: County FIPS code
- TRACTCE10: Tract FIPS code
- GEOID10: Complete FIPS tract ID
- totpop: Total population 2012-2016
- mhhinc: Median household income 2012-2016
- mrent: Median monthly rent 2012-2016
- mhval: Median housing value 2012-2016
- pnhwhite: Percent non-Hispanic white 2012-2016
- pnhasian: Percent non-Hispanic Asian 2012-2016
- pcol: Percent with college degree 2012-2016
- ppa: Percent of adults on public assistance 2012-2016
- ppov: Percent below poverty line 2012-2016
- popd: Population density 2012-2016

Objective: We want to examine the relationship between neighborhood characteristics and major building code violation rates.

```
phil    <- st_read("phil_tracts.shp")
```

```
## Reading layer `phil_tracts' from data source `D:\OneDrive\R\phil_tracts.shp' using driver `ESRI Shap
```

```
## Simple feature collection with 376 features and 25 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 476461.9 ymin: 4413070 xmax: 503772 ymax: 4443067
## projected CRS:  UTM_Zone_18_Northern_Hemisphere
```

```
phil.sp <- as(phil, "Spatial")

# note that units are in meters.
proj4string(phil.sp)
```

```
## [1] "+proj=utm +zone=18 +ellps=GRS80 +units=m +no_defs"
```

```
# Our response variable and 9 predictor variables
formula <- usarea ~ lmhhinc + lpop + pnhblk +
                    punemp + pvac  + ph70 +
                    lmhval + phnew + phisp
```

Let's run a basic **Ordinary Least Squares (OLS)** regression on the number of major building code violations per area in square miles (usarea)

```
phil.ols <- phil.sp


fit.ols <-glm(formula, data = phil.ols)

summary(fit.ols)
```

```
##
## Call:
## glm(formula = formula, data = phil.ols)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -177.24   -34.81    -9.91    23.03   670.17
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  534.491    164.270   3.254  0.00124 **
## lmhhinc        2.462     12.176   0.202  0.83990
## lpop          -1.344      6.338  -0.212  0.83216
## pnhblk        21.158     18.077   1.170  0.24260
## punemp        -5.097     63.645  -0.080  0.93622
## pvac         371.699     58.427   6.362 5.96e-10 ***
## ph70         -79.691     35.535  -2.243  0.02552 *
## lmhval       -45.668     10.458  -4.367 1.64e-05 ***
## phnew         17.958    319.042   0.056  0.95514
## phisp        -56.308     30.695  -1.834  0.06741 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4829.927)
##
##     Null deviance: 2938287  on 375  degrees of freedom
## Residual deviance: 1767753  on 366  degrees of freedom
## AIC: 4268.4
```

```
##
## Number of Fisher Scoring iterations: 2
```

We find:

- the percent of vacant units is positively associated with the number of major building code violations per square mile.
- the percent of units built before 1970 and log median housing value are negatively associated with the number of major building code violations per square mile.

But:

- This model assumes spatial homogeneity in these relationships. . . is this an appropriate assumption?

Let's check for any spatial patterning in the residuals!

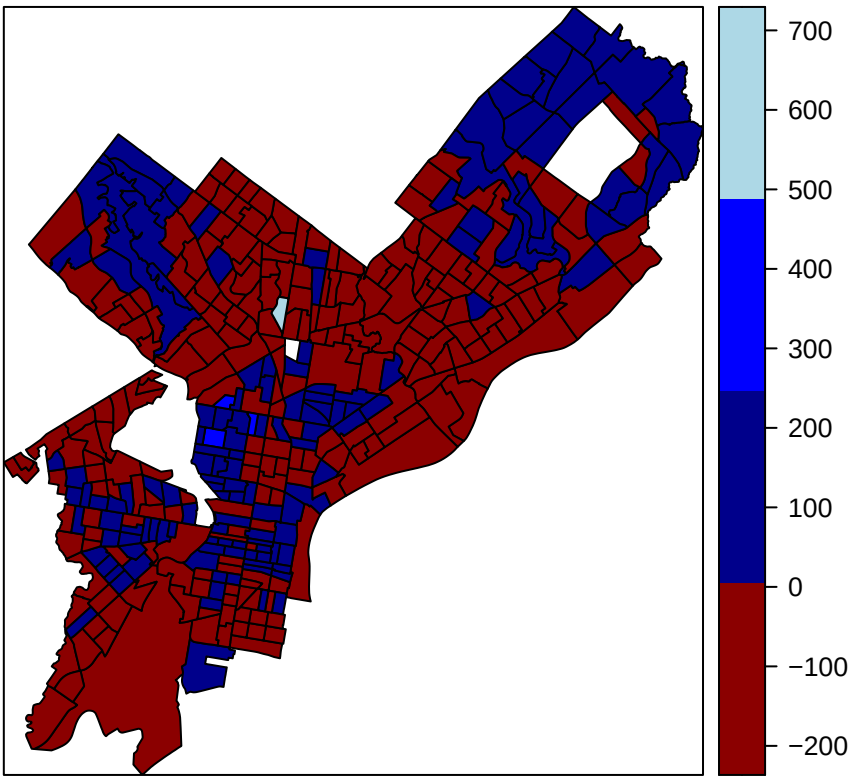
```
phil.ols$resids <- rep(0 , nrow(phil.ols))

resids <-residuals(fit.ols)
names(resids) <- NULL

phil.ols$resids <- resids

colors <- c("dark red", "dark blue", "blue", "light blue")

spplot(phil.ols, zcol = "resids", cuts=3, col.regions=colors, cex=1)
```



Big chunks of red, big chunks of blue. The 4 tracts with the largest residuals are close in proximity.

Let's explore further with **Geographically Weighted Regression**

There are two major decisions to make when running a GWR:

(1) the kernel density function assigning individual weights

(2) the bandwidth h of the kernel function, which determines the degree of distance decay

Choosing a kernel density function:

- many to choose from, but recommend you start with two.
- Use Gaussian weighting function when the influence of neighboring features becomes smoothly and gradually less important and there is a distance after which that influence is always present regardless of how far away the surrounding features are.
- Use Bisquare weighting function to specify a distance after which features will have no impact on the regression results. Features outside of the neighborhood specified are assigned zero and do not impact the local regression for the target feature.
- when comparing a Bisquare weighting scheme to a Gaussian weighting scheme with the same neighborhood specifications, weights will decrease more quickly with Bisquare.

Selecting the bandwidth h of the kernel function:

- R does this by leave one out cross validation (LOOCV), where R chooses the best h that minimizes the sum of squared errors at all locations.
- You can also select this manually, if you want.

```
phil.gwr <- phil.sp

gwr.b1 <-gwr.sel(formula, phil.gwr)
```

```
## Bandwidth: 14224.76 CV score: 1845565
## Bandwidth: 22993.17 CV score: 1860162
## Bandwidth: 8805.59 CV score: 1818722
## Bandwidth: 5456.356 CV score: 1750890
## Bandwidth: 3386.415 CV score: 1610987
## Bandwidth: 2107.121 CV score: 1497008
## Bandwidth: 1316.474 CV score: 1412725
## Bandwidth: 827.8277 CV score: 1787166
## Bandwidth: 1618.475 CV score: 1446078
## Bandwidth: 1129.828 CV score: 1448167
## Bandwidth: 1377.629 CV score: 1414671
## Bandwidth: 1298.316 CV score: 1413143
## Bandwidth: 1324.036 CV score: 1412698
## Bandwidth: 1322.62 CV score: 1412697
## Bandwidth: 1322.714 CV score: 1412696
## Bandwidth: 1322.708 CV score: 1412696
## Bandwidth: 1322.708 CV score: 1412696
## Bandwidth: 1322.708 CV score: 1412696
## Bandwidth: 1322.708 CV score: 1412696
```

```
gwr.b1
```

```
## [1] 1322.708
```

```
# This represents your value h.
# The weighting function will search, and include all observations within this radius.
# This is the distance in meters, because our data are projected in a system measured in meters,
```

```
# change the weighting function (default is Guassian)

gwr.b2 <-gwr.sel(formula, gweight = gwr.bisquare, data = phil.gwr)
```

```
## Bandwidth: 14224.76 CV score: 1797387
## Bandwidth: 22993.17 CV score: 1840913
## Bandwidth: 8805.59 CV score: 1634064
## Bandwidth: 5456.356 CV score: 1578492
## Bandwidth: 1760.637 CV score: NA
## Bandwidth: 4044.717 CV score: NA
## Bandwidth: 6735.649 CV score: 1585951
## Bandwidth: 5535.278 CV score: 1581610
## Bandwidth: 4917.158 CV score: NA
## Bandwidth: 5250.4 CV score: 1569337
## Bandwidth: 5123.113 CV score: 1563084
## Bandwidth: 5044.445 CV score: NA
## Bandwidth: 5171.732 CV score: 1565614
## Bandwidth: 5093.064 CV score: 1561374
## Bandwidth: 5074.493 CV score: NA
## Bandwidth: 5104.542 CV score: 1562042
## Bandwidth: 5085.971 CV score: NA
## Bandwidth: 5097.448 CV score: 1561631
## Bandwidth: 5090.355 CV score: NA
## Bandwidth: 5094.739 CV score: 1561473
## Bandwidth: 5092.029 CV score: NA
## Bandwidth: 5093.704 CV score: 1561412
## Bandwidth: 5092.669 CV score: NA
## Bandwidth: 5093.309 CV score: 1561389
## Bandwidth: 5092.913 CV score: 1561365
## Bandwidth: 5092.82 CV score: NA
## Bandwidth: 5092.971 CV score: 1561369
## Bandwidth: 5092.878 CV score: NA
## Bandwidth: 5092.935 CV score: 1561367
## Bandwidth: 5092.9 CV score: 1561365
## Bandwidth: 5092.891 CV score: NA
## Bandwidth: 5092.905 CV score: 1561365
## Bandwidth: 5092.897 CV score: NA
## Bandwidth: 5092.902 CV score: 1561365
## Bandwidth: 5092.899 CV score: 1561365
## Bandwidth: 5092.898 CV score: 1561364
## Bandwidth: 5092.897 CV score: NA
## Bandwidth: 5092.898 CV score: 1561365
## Bandwidth: 5092.898 CV score: 1561364
## Bandwidth: 5092.898 CV score: NA
## Bandwidth: 5092.898 CV score: 1561364
```

```
gwr.b2
```

```
## [1] 5092.898
```

To adapt, or not to adapt?

- The GWR models we ran above yielded a fixed distance to search for neighbors to include in the local regression.
- If the sample points are reasonably regularly spaced in the study area, then a kernel with a fixed

bandwidth is a suitable choice for modelling.

- If the sample points are clustered in the study area, it is generally desirable to allow the kernel to accommodate this irregularity.
- The bandwidth can adapt as needed, allowing neighborhoods to be smaller where features are dense and larger where features are sparse.

```
# adaptive kernel
gwr.b3 <-gwr.sel(formula, data = phil.gwr, adapt = TRUE)
```

```
## Adaptive q: 0.381966 CV score: 1785033
## Adaptive q: 0.618034 CV score: 1814714
## Adaptive q: 0.236068 CV score: 1733969
## Adaptive q: 0.145898 CV score: 1650907
## Adaptive q: 0.09016994 CV score: 1565518
## Adaptive q: 0.05572809 CV score: 1471615
## Adaptive q: 0.03444185 CV score: 1401939
## Adaptive q: 0.02128624 CV score: 1384273
## Adaptive q: 0.01588534 CV score: 1431365
## Adaptive q: 0.02662582 CV score: 1380970
## Adaptive q: 0.02518866 CV score: 1380056
## Adaptive q: 0.02491844 CV score: 1380031
## Adaptive q: 0.02487775 CV score: 1380032
## Adaptive q: 0.02495914 CV score: 1380032
## Adaptive q: 0.02491844 CV score: 1380031
```

```
gwr.b3
```

```
## [1] 0.02491844
```

- The weighting function and include this fraction of observations in a model for each tract.
- Instead of a specific distance, the bandwidth is the number of nearest neighbors.

Build the GWR models

```
gwr.fit1 <-gwr(formula, data = phil.gwr, bandwidth = gwr.b1, se.fit=T, hatmatrix=T)

gwr.fit2 <-gwr(formula, data = phil.gwr, bandwidth = gwr.b2, gweight = gwr.bisquare, se.fit=T, hatmatri

gwr.fit3 <-gwr(formula, data = phil.gwr, adapt = gwr.b3, se.fit=T, hatmatrix=T)
```

Investigate bandwidth behavior

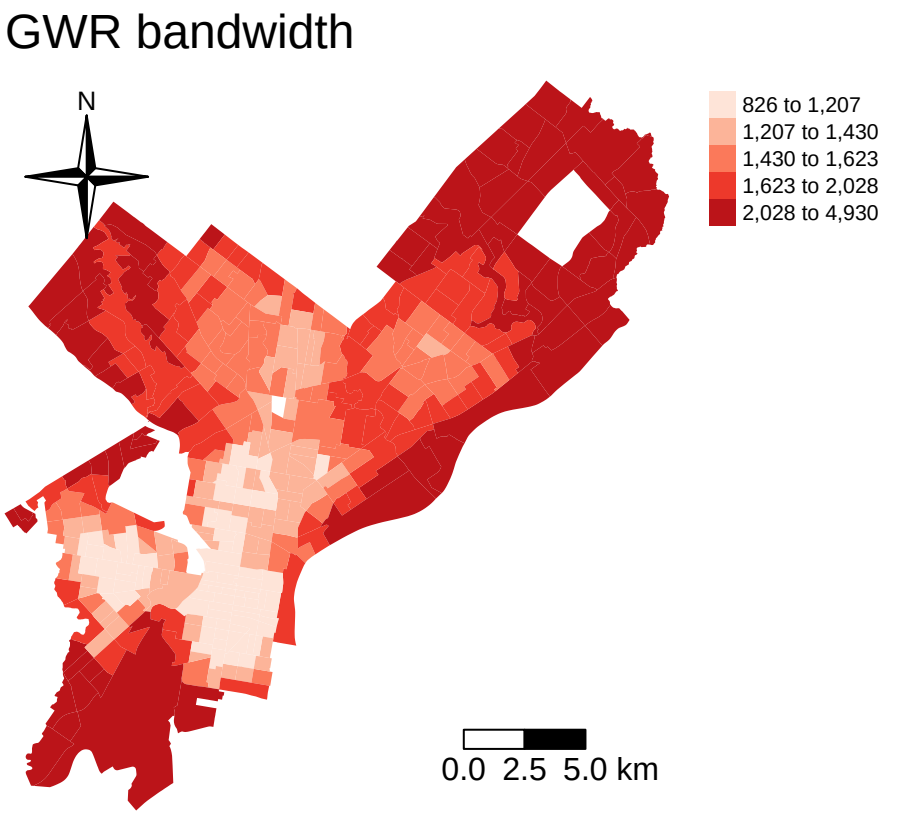
```
summary(gwr.fit3$bandwidth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   825.8  1256.4  1500.6  1669.6  1867.9  4929.9
```

```
phil$bwadapt <- gwr.fit3$bandwidth

phil.sp2 <- as(phil, "Spatial")

tm_shape(phil, unit = "km") +
  tm_polygons(col = "bwadapt", style = "quantile", palette = "Reds",
              border.alpha = 0, title = "") +
  tm_scale_bar(breaks = c(0, 2.5, 5), text.size = 1, position = c("right", "bottom")) +
  tm_compass(type = "4star", position = c("left", "top")) +
  tm_layout(main.title = "GWR bandwidth", frame = FALSE, legend.outside = TRUE)
```



Investigate regression results

```
# don't use summary!
gwr.fit3
```

```
## Call:
## gwr(formula = formula, data = phil.gwr, adapt = gwr.b3, hatmatrix = T,
##     se.fit = T)
## Kernel function: gwr.Gauss
## Adaptive quantile: 0.02491844 (about 9 of 376 data points)
## Summary of GWR coefficient estimates at data points:
##                       Min.     1st Qu.      Median     3rd Qu.        Max.
## X.Intercept. -1413.25718     2.04814   150.67770   593.38119  2856.09861
## lmhhinc        -77.30238    -6.62505     2.08877    20.59832   121.03243
## lpop           -71.53993     0.32328     6.55222    19.42020    93.59455
## pnhblk        -139.33868    -0.35274    39.43998   102.07286   462.87992
## punemp        -592.27650  -109.64202    -3.93096    63.56270   623.38186
## pvac         -1410.12965    11.95427   193.34738   350.39251  1047.77143
## ph70          -975.65611  -190.62161   -67.38336   -13.17506   137.47857
## lmhval        -185.48730   -73.39044   -36.70912    -7.56967    48.91389
## phnew        -2570.54553  -577.37945    29.21937   654.40082  4045.23829
## phisp         -182.91660   -29.72723    -7.23980    65.71058   771.29484
##                Global
## X.Intercept. 534.4908
## lmhhinc        2.4616
## lpop          -1.3441
## pnhblk        21.1576
```

```
## punemp        -5.0966
## pvac         371.6993
## ph70         -79.6910
## lmhval       -45.6676
## phnew         17.9575
## phisp        -56.3076
## Number of data points: 376
## Effective number of parameters (residual: 2traceS - traceS'S): 177.8408
## Effective degrees of freedom (residual: 2traceS - traceS'S): 198.1592
## Sigma (residual: 2traceS - traceS'S): 54.21695
## Effective number of parameters (model: traceS): 135.2358
## Effective degrees of freedom (model: traceS): 240.7642
## Sigma (model: traceS): 49.18654
## Sigma (ML): 39.35938
## AICc (GWR p. 61, eq 2.33; p. 96, eq. 4.21): 4258.02
## AIC (GWR p. 96, eq. 4.22): 3964.174
## Residual sum of squares: 582484.4
## Quasi-global R2: 0.8017605
```

```
# notice the global coefficients match our OLS regression
summary(fit.ols)
```

```
##
## Call:
## glm(formula = formula, data = phil.ols)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -177.24   -34.81    -9.91    23.03   670.17
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  534.491    164.270   3.254  0.00124 **
## lmhhinc        2.462     12.176   0.202  0.83990
## lpop          -1.344      6.338  -0.212  0.83216
## pnhblk        21.158     18.077   1.170  0.24260
## punemp        -5.097     63.645  -0.080  0.93622
## pvac         371.699     58.427   6.362 5.96e-10 ***
## ph70         -79.691     35.535  -2.243  0.02552 *
## lmhval       -45.668     10.458  -4.367 1.64e-05 ***
## phnew         17.958    319.042   0.056  0.95514
## phisp        -56.308     30.695  -1.834  0.06741 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4829.927)
##
##     Null deviance: 2938287  on 375  degrees of freedom
## Residual deviance: 1767753  on 366  degrees of freedom
## AIC: 4268.4
##
## Number of Fisher Scoring iterations: 2
```

```
# important results are found within SDF
names(gwr.fit3$SDF)
```

```
##  [1] "sum.w"               "X.Intercept."        "lmhhinc"
##  [4] "lpop"                "pnhblk"              "punemp"
##  [7] "pvac"                "ph70"                "lmhval"
## [10] "phnew"               "phisp"               "X.Intercept._se"
## [13] "lmhhinc_se"          "lpop_se"             "pnhblk_se"
## [16] "punemp_se"           "pvac_se"             "ph70_se"
## [19] "lmhval_se"           "phnew_se"            "phisp_se"
## [22] "gwr.e"               "pred"                "pred.se"
## [25] "localR2"             "X.Intercept._se_EDF" "lmhhinc_se_EDF"
## [28] "lpop_se_EDF"         "pnhblk_se_EDF"       "punemp_se_EDF"
## [31] "pvac_se_EDF"         "ph70_se_EDF"         "lmhval_se_EDF"
## [34] "phnew_se_EDF"        "phisp_se_EDF"        "pred.se.1"
```

```
# the regression coefficients
names(gwr.fit3$SDF[2:11])
```

```
##  [1] "X.Intercept." "lmhhinc"      "lpop"         "pnhblk"       "punemp"
##  [6] "pvac"         "ph70"         "lmhval"       "phnew"        "phisp"
```

```
# the coefficient standard errors.
names(gwr.fit3$SDF[12:21])
```

```
##  [1] "X.Intercept._se" "lmhhinc_se"      "lpop_se"         "pnhblk_se"
##  [5] "punemp_se"       "pvac_se"         "ph70_se"         "lmhval_se"
##  [9] "phnew_se"        "phisp_se"
```

```
results <- as.data.frame(gwr.fit3$SDF[,2:11])

names(results) <- c("X.Int.coef","lmhhinc.coef","lpop.coef","pnhblk.coef","punemp.coef","pvac.coef","ph

phil <- cbind(phil, results)
```

But is the spatial variability in the coefficients significant?

- The package spgwr has a battery of tests comparing OLS and GWR models.
- The null in these tests is the OLS.
- A statistically significant test statistic indicates that the GWR provides a statistically significant improvement over an OLS in terms of its ability to match observed values.

```
# 4 tests compare overall model fit
BFC02.gwr.test(gwr.fit3)
```

```
##
##  Brunsdon, Fotheringham & Charlton (2002, pp. 91-2) ANOVA
##
## data:  gwr.fit3
## F = 3.0349, df1 = 366.00, df2 = 198.16, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## SS OLS residuals SS GWR residuals
##        1767753.4         582484.4
```

```
BFC99.gwr.test(gwr.fit3)
```

```
##
##  Brunsdon, Fotheringham & Charlton (1999) ANOVA
##
```

```
## data:  gwr.fit3
## F = 2.4024, df1 = 333.76, df2 = 260.43, p-value = 2.142e-13
## alternative hypothesis: greater
## sample estimates:
## SS GWR improvement   SS GWR residuals
##          1185269.0           582484.4
```

```
LMZ.F1GWR.test(gwr.fit3)
```

```
##
##  Leung et al. (2000) F(1) test
##
## data:  gwr.fit3
## F = 0.6086, df1 = 260.43, df2 = 366.00, p-value = 1.108e-05
## alternative hypothesis: less
## sample estimates:
## SS OLS residuals SS GWR residuals
##        1767753.4         582484.4
```

```
LMZ.F2GWR.test(gwr.fit3)
```

```
##
##  Leung et al. (2000) F(2) test
##
## data:  gwr.fit3
## F = 1.4621, df1 = 233.86, df2 = 366.00, p-value = 0.0005763
## alternative hypothesis: greater
## sample estimates:
##   SS OLS residuals SS GWR improvement
##            1767753            1185269
```

```
# All show that the GWR shows significant improvement in explanatory power over an OLS.


# examines spatial variation in individual coefficients.
LMZ.F3GWR.test(gwr.fit3)
```

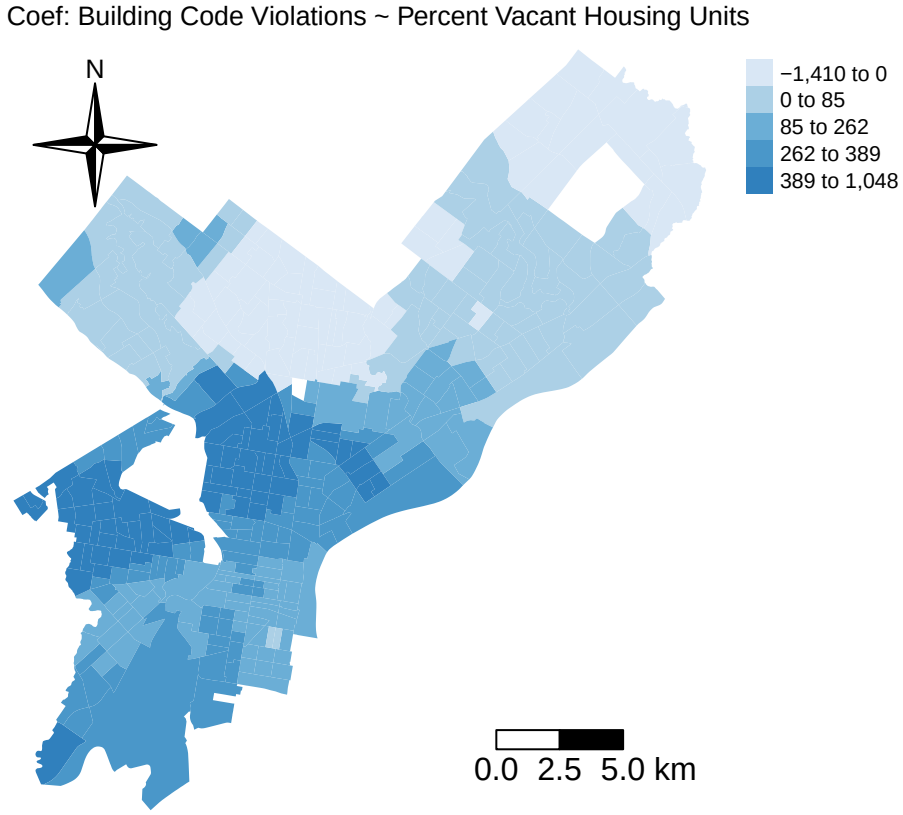
```
##
## Leung et al. (2000) F(3) test
##
##             F statistic Numerator d.f. Denominator d.f.     Pr(>)
## (Intercept)     0.94419      122.59061           260.43 0.6369477
## lmhhinc         0.57486      105.11265           260.43 0.9993521
## lpop            0.83970      117.68318           260.43 0.8595358
## pnhblk          1.43484       89.33826           260.43 0.0151795 *
## punemp          0.92026      119.99577           260.43 0.6949096
## pvac            2.94794      106.14539           260.43 1.055e-12 ***
## ph70            1.69122      100.67109           260.43 0.0004901 ***
## lmhval          0.91483      116.24838           260.43 0.7054314
## phnew           0.22495       11.59110           260.43 0.9966467
## phisp           0.35611       40.49489           260.43 0.9999100
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The last test shows that the variables percent homes vacant, percent black, and percent of homes built before 1970's indicate statistically significant spatial heterogeneity in its GWR coefficients.

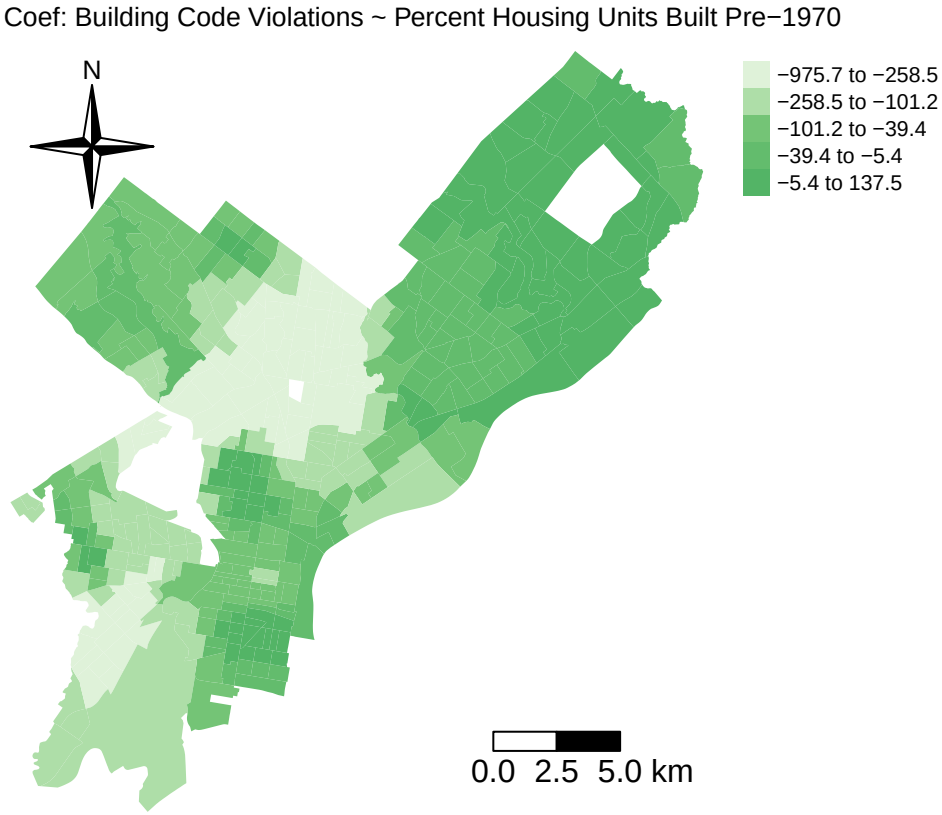
These results indicate that there is spatial heterogeneity in the relationships between our covariates and major build code violations.

Let's look at the spatial distribution of the GWR coefficients for these variables to see the range of variation between the local coefficients.

```
tm_shape(phil, unit = "km") +
  tm_polygons(col = "pvac.coef", style = "quantile", palette = "Blues", midpoint = NA,
              border.alpha = 0, title = "") +
  tm_scale_bar(breaks = c(0, 2.5, 5), text.size = 1, position = c("right", "bottom")) +
  tm_compass(type = "4star", position = c("left", "top")) +
  tm_layout(main.title = "Coef: Building Code Violations ~ Percent Vacant Housing Units", main.title.si
```




```
tm_shape(phil, unit = "km") +
  tm_polygons(col = "ph70.coef", style = "quantile", palette = "Greens", midpoint = NA,
              border.alpha = 0, title = "") +
  tm_scale_bar(breaks = c(0, 2.5, 5), text.size = 1, position = c("right", "bottom")) +
  tm_compass(type = "4star", position = c("left", "top")) +
  tm_layout(main.title = "Coef: Building Code Violations ~ Percent Housing Units Built Pre-1970", main.
```




```
tm_shape(phil, unit = "km") +
  tm_polygons(col = "pnhblk.coef", style = "quantile", palette = "Oranges", midpoint = NA,
              border.alpha = 0, title = "") +
  tm_scale_bar(breaks = c(0, 2.5, 5), text.size = 1, position = c("right", "bottom")) +
  tm_compass(type = "4star", position = c("left", "top")) +
  tm_layout(main.title = "Coef: Building Code Violations ~ Percent Black", main.title.size = 0.90, frame
```

Coef: Building Code Violations ~ Percent Black