WorkBook

Zack W Almquist (University of Minnesota)

November 9, 2016

Preliminaries

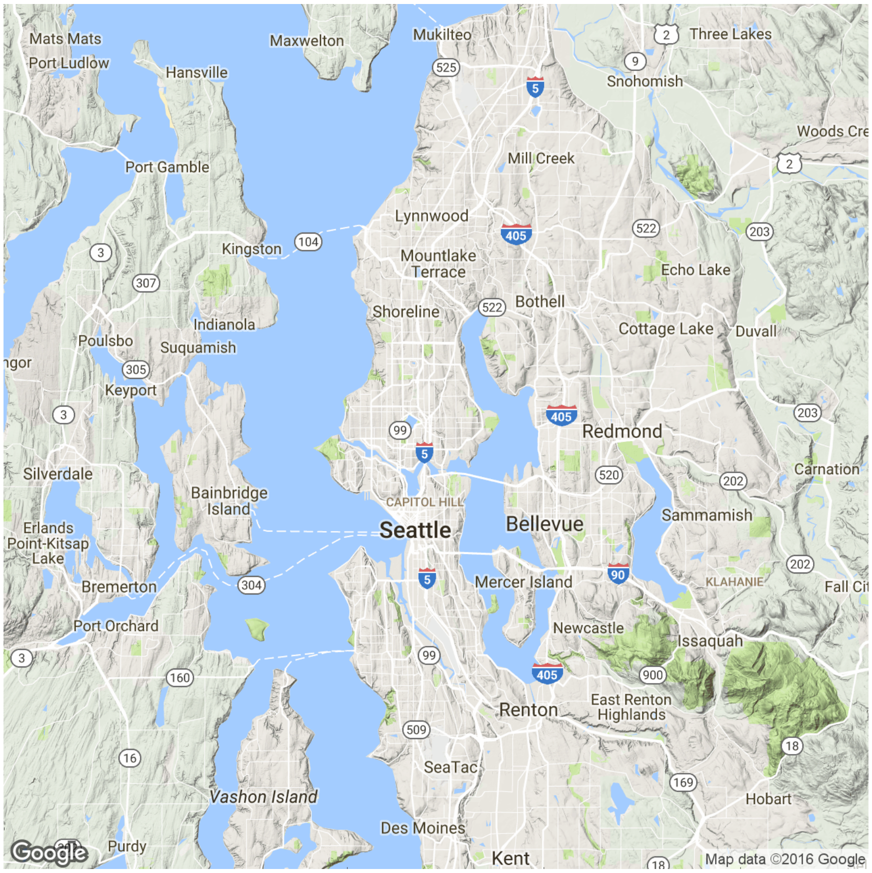
In the following component of this workshop we will attempt to explore and use the packages and R files to accomplish tasks. The first thing we are going to do, however, is to explore the 'ggmap' and 'openmaps' R packages so that we can bring into our GIS tools the ability plot street and aerial maps.

ggmap package

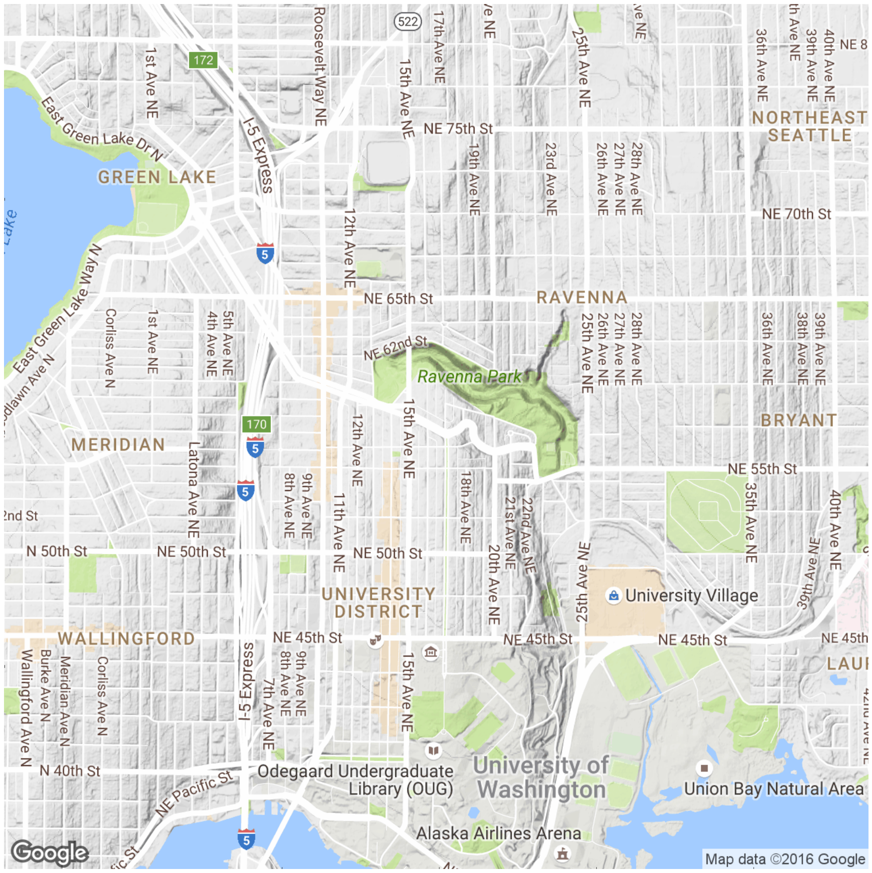
```
install.packages("ggmap")
```

```
library(ggplot2)
library(ggmap)
library(caTools)
library(sp)
library(rgdal)
library(rgeos)
library(spdep)
library(dplyr)
library(plyr)
library(readr)
library(UScensus2010)
library(UScensus2010cdp)
```

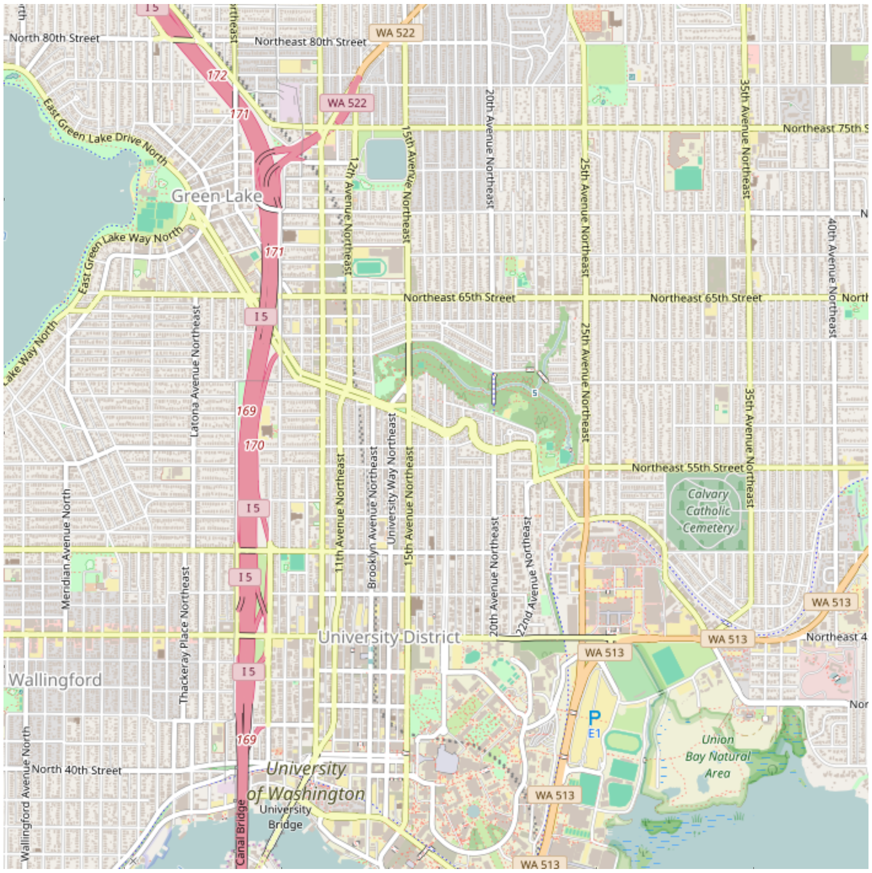
```
qmap(location = "University of Washington, Seattle, WA")
```




```r
qmap(location = "University of Washington, Seattle, WA", zoom = 14)
```




```r
qmap(location = "University of Washington, Seattle, WA", zoom = 14,
    source = "osm")
```



Overlay

Let's overlay Zillow's neighborhoods onto google map graphics.

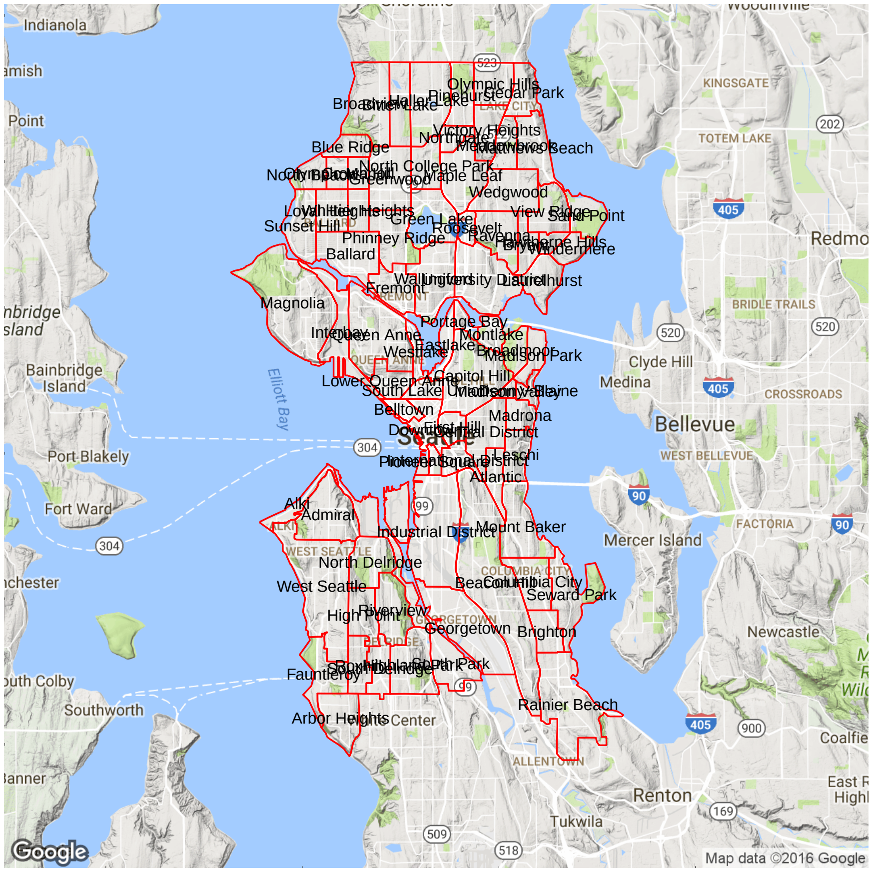
```
seatle_map <- qmap(location = "Seattle, WA", zoom = 11)
base_address <- "http://www.zillow.com/static/shp/"
state <- "WA"
city <- "Seattle"
seattle_nb <- readZillow_NB(base_address, state, city)
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/var/folders/y5/f2xwbx9d46z7m05mmqwr27_r0000gn/T//RtmpTPCode/file138688c31a29", layer: "Zil
## with 299 features
## It has 5 fields
```

```
zillowurl <- "http://www.zillowstatic.com/vstatic/70a941d/static/logos/Zillow_Logo_HoodsProvided_RightA
zillow_logo <- read.gif(zillowurl, flip = TRUE)
```

```
seattle_nb <- spTransform(seattle_nb, CRS("+proj=longlat +datum=WGS84"))
Neighborhoods <- fortify(seattle_nb)
LabelData <- data.frame(coordinates(seattle_nb), NAME = seattle_nb$NAME,
    stringsAsFactors = FALSE)
```

```r
seatle_map + geom_polygon(aes(x = long, y = lat, group = group),
    fill = "grey", size = 0.3, color = "red", data = Neighborhoods,
    alpha = 0) + geom_text(aes(x = X1, y = X2, label = NAME),
    size = 2, data = LabelData)
```



NEIGHBORHOODS PROVIDED BY Zillow

```r
seatle_map2 <- qmap(location = "Seattle, WA", zoom = 14)

seatle_map2 + geom_polygon(aes(x = long, y = lat, group = group),
    fill = "grey", size = 0.3, color = "red", data = Neighborhoods,
    alpha = 0) + geom_text(aes(x = X1, y = X2, label = NAME),
    size = 4, data = LabelData)
```

NEIGHBORHOODS PROVIDED BY Zillow

Geo Code

Use the above code and google to geocode your favorite locations into the map. For example add your favorite coffeeshop, park or walk. Be creative!

```
## Your Code Here
```

Add Census Data

Go back to the ACS tutorial and look over the code and possible variables. Find one that you are interested in for Seattle. Download the data at the tract level and plot it ontop of the ggmap object.

```
## Your Code Here
```

Spatial Statistics

To explore some spatial statistics models we are going to combine ACS data with AirBnB through http://http://insideairbnb.com/. Seattle on 04 January, 2016.

Code for ACS

```
key<-"Your Key Here"
acsvariables<-c(
                "B06010_023E", #    Born in other state in the United States:
                "B07013_002E", #    Householder lived in owner-occupied housing units
                "B07013_003E", #    Householder lived in renter-occupied housing units
                "B17001F_002E") #   Income in the past 12 months below poverty level:
                                # SEX BY AGE (SOME OTHER RACE ALONE)

WA_ACS<-CensusAPI2010Spatial(acsvariables,state.fips=c("53"),
level="tract",key,summaryfile="ACS")
```

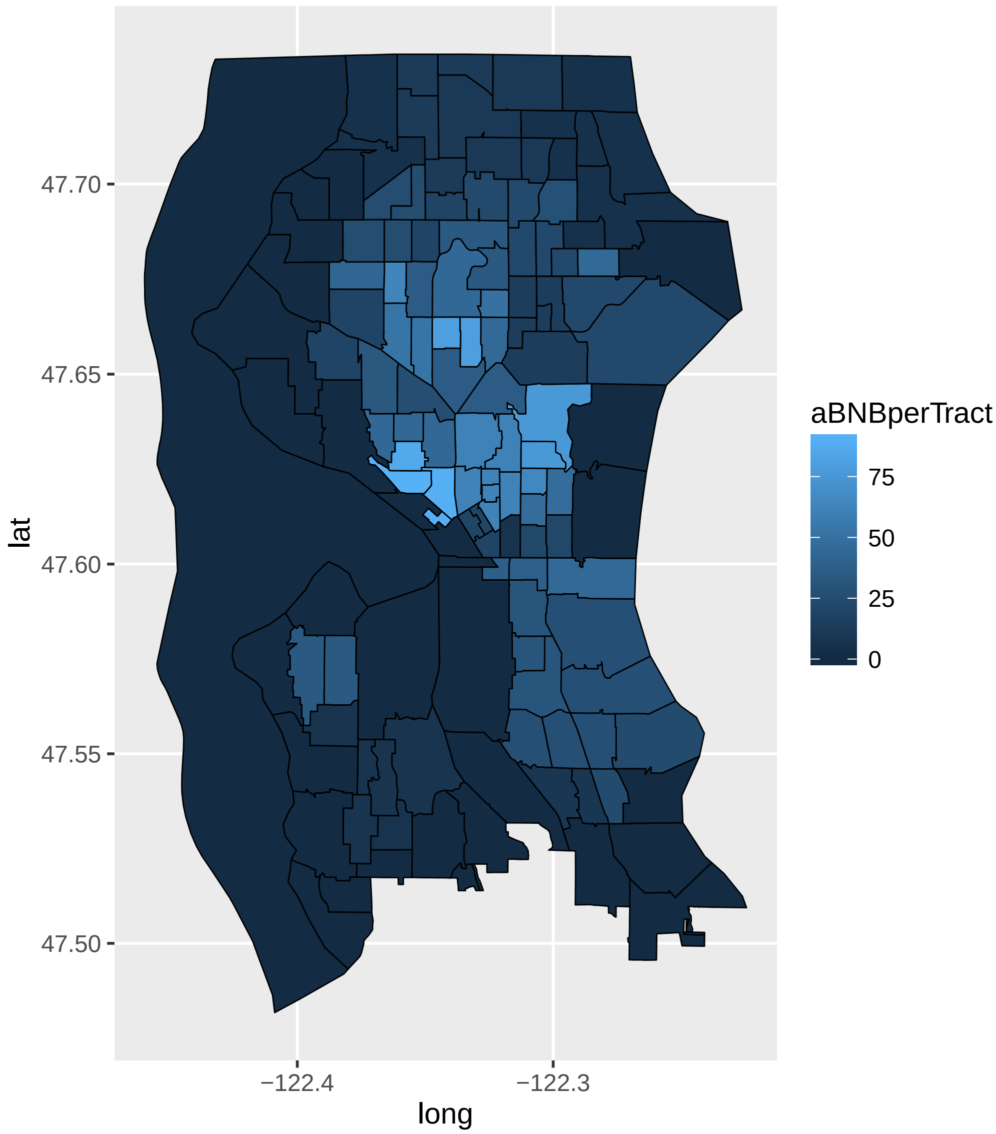
Airbnb and ACS data

```
load("data/airbnb.rda")
load("data/wa_acs.rda")

airbnb_sp <- SpatialPointsDataFrame(cbind(airBNB_seattle$longitude,
    airBNB_seattle$latitude), data = as.data.frame(airBNB_seattle),
    proj4string = CRS(proj4string(WA_ACS)))

airbnbPerTract <- sapply(1:length(WA_ACS@polygons), function(x) {
    temp <- over(airbnb_sp, WA_ACS[x, ])
    sum(!is.na(temp[, 1]))
})

WA_ACS$aBNBperTract <- airbnbPerTract

## Generate Seattle Only Tracts
data(washington.cdp10)
seattle <- washington.cdp10[washington.cdp10$name == "Seattle",
    ]
seattle_tract <- gIntersection(WA_ACS, seattle, byid = TRUE,
    drop_not_poly = TRUE)
data <- over(seattle_tract, WA_ACS)
rownames(data) <- sapply(slot(seattle_tract, "polygons"), function(i) slot(i,
    "ID"))
seattle_tract <- SpatialPolygonsDataFrame(seattle_tract, data = data)
```

```
load("data/seattle_tract.rda")
seattle_tract@data$id <- rownames(seattle_tract@data)
seattle_tract.points <- fortify(seattle_tract)
seattle_tract.df <- join(seattle_tract.points, seattle_tract@data,
    by = "id")
```

ggplot of the data

```r
ggplot() + geom_polygon(data = seattle_tract.df, aes(x = long,
    y = lat, group = group, fill = aBNBperTract), color = "black",
    size = 0.25) + coord_map()
```



Simple analysis

```r
seattle_proj <- spTransform(seattle_tract, CRS("+proj=merc +zone=10s +ellps=WGS84 +datum=WGS84"))
seattle_proj$area <- areaPoly(seattle_proj)
seattle_proj$den <- seattle_proj$P0010001/seattle_proj$area

summary(lm1 <- lm(aBNBperTract ~ den, data = seattle_proj))
```

```
##
## Call:
```

```
## lm(formula = aBNBperTract ~ den, data = seattle_proj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.213 -18.109  -4.487  11.864  66.733
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.266      2.287   9.734   <2e-16 ***
## den          677.167    492.826   1.374    0.172
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.77 on 134 degrees of freedom
## Multiple R-squared:  0.01389,    Adjusted R-squared:  0.006535
## F-statistic: 1.888 on 1 and 134 DF,  p-value: 0.1717
```

```
summary(lm2 <- lm(aBNBperTract ~ den, weights = I(1/area), data = seattle_proj))
```

```
##
## Call:
## lm(formula = aBNBperTract ~ den, data = seattle_proj, weights = I(1/area))
##
## Weighted Residuals:
##       Min        1Q    Median        3Q       Max
## -0.037397 -0.012065 -0.007266  0.001722  0.089263
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   33.189      2.672  12.420  < 2e-16 ***
## den         -795.090    170.847  -4.654 7.73e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01785 on 134 degrees of freedom
## Multiple R-squared:  0.1391, Adjusted R-squared:  0.1327
## F-statistic: 21.66 on 1 and 134 DF,  p-value: 7.735e-06
```

```
summary(lm3 <- lm(aBNBperTract ~ den + B06010_023E, weights = I(1/area),
    data = seattle_proj))
```

```
##
## Call:
## lm(formula = aBNBperTract ~ den + B06010_023E, data = seattle_proj,
##     weights = I(1/area))
##
## Weighted Residuals:
##       Min        1Q    Median        3Q       Max
## -0.023984 -0.007337 -0.001999  0.002535  0.068768
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.66494    3.29716  -1.112    0.268
## den         -577.90226  113.89198  -5.074 1.28e-06 ***
## B06010_023E    0.02209    0.00167  13.226  < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01177 on 133 degrees of freedom
## Multiple R-squared:  0.6282, Adjusted R-squared:  0.6226
## F-statistic: 112.4 on 2 and 133 DF,  p-value: < 2.2e-16
```

```
sapply(list(lm1, lm2, lm3), AIC)
```

```
## [1] 1251.777 1336.123 1223.947
```

```
nb <- poly2nb(seattle_proj)
any(card(nb) == 0)
```

```
## [1] FALSE
```

```
nbw <- nb2listw(nb, style = "B", zero.policy = TRUE)  # B is the basic binary coding
lm.morantest(lm3, nbw, zero.policy = TRUE)
```

```
##
##  Global Moran I for regression residuals
##
## data:
## model: lm(formula = aBNBperTract ~ den + B06010_023E,
## data = seattle_proj, weights = I(1/area))
## weights: nbw
##
## Moran I statistic standard deviate = 6.5246, p-value
## = 3.409e-11
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation         Variance
##      0.311565912     -0.009369102      0.002419508
```

```
lm.morantest(lm2, nbw, zero.policy = TRUE)
```

```
##
##  Global Moran I for regression residuals
##
## data:
## model: lm(formula = aBNBperTract ~ den, data =
## seattle_proj, weights = I(1/area))
## weights: nbw
##
## Moran I statistic standard deviate = 7.5719, p-value
## = 1.84e-14
```

```
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation         Variance
##      0.368986944     -0.006656632      0.002461201
```

```
summary(alm1 <- spautolm(aBNBperTract ~ den, data = seattle_proj,
    listw = nbw, family = "SAR"))
```

```
##
## Call:
## spautolm(formula = aBNBperTract ~ den, data = seattle_proj, listw = nbw,
##     family = "SAR")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -37.4267  -8.7805  -3.1991   9.0669  63.9861
##
## Coefficients:
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  17.5317     4.2986  4.0784 4.534e-05
## den         484.8094   371.3753  1.3054    0.1917
##
## Lambda: 0.12469 LR test value: 68.423 p-value: < 2.22e-16
## Numerical Hessian standard error of lambda: 0.010572
##
## Log likelihood: -588.6771
## ML residual variance (sigma squared): 295.48, (sigma: 17.189)
## Number of observations: 136
## Number of parameters estimated: 4
## AIC: 1185.4
```

```
summary(alm2 <- spautolm(aBNBperTract ~ den + B06010_023E, data = seattle_proj,
    listw = nbw, family = "SAR"))
```

```
##
## Call:
## spautolm(formula = aBNBperTract ~ den + B06010_023E, data = seattle_proj,
##     listw = nbw, family = "SAR")
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -23.60939  -6.04768  -0.78393   4.75274  49.45593
##
## Coefficients:
##                Estimate  Std. Error z value Pr(>|z|)
## (Intercept)  -3.5355138   3.7881932 -0.9333   0.3507
## den         -54.6718535 284.5890064 -0.1921   0.8477
## B06010_023E   0.0163567   0.0015954 10.2526   <2e-16
##
## Lambda: 0.12171 LR test value: 53.092 p-value: 3.183e-13
## Numerical Hessian standard error of lambda: 0.011555
##
## Log likelihood: -549.7776
```

```
## ML residual variance (sigma squared): 168.2, (sigma: 12.969)
## Number of observations: 136
## Number of parameters estimated: 5
## AIC: 1109.6
```

```r
sapply(list(alm1, alm2), AIC)
```

```
## [1] 1185.354 1109.555
```

```r
LR.sarlm(alm1, alm2)  #http://rpackages.ianhowson.com/rforge/spdep/man/LR.sarlm.html
```

```
##
##  Likelihood ratio for spatial linear models
##
## data:
## Likelihood ratio = -77.799, df = 1, p-value < 2.2e-16
## sample estimates:
## Log likelihood of alm1 Log likelihood of alm2
##              -588.6771              -549.7776
```