Spatial Regression Modeling Example - TAMU RDC

Corey S. Sparks, Ph.D. - University of Texas at San Antonio

May 9, 2018

This empirical example illustrates the use of R for extracting data from the American Community Survey, and using these data in a spatial regression model.

See what variables are in the ACS Demographic Profile Tables

```
library(tidycensus);  library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------ tidyverse 1

## v ggplot2 2.2.1.9000     v purrr   0.2.4
## v tibble  1.4.2          v dplyr   0.7.4
## v tidyr   0.8.0          v stringr 1.3.0
## v readr   1.1.1          v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------------ tidyverse_conflic
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::vars()   masks ggplot2::vars()
```

```
v15 <- load_variables(2015 , "acs5/profile", cache = TRUE)
#View(v15)
```

Extract from ACS summary file from 2015

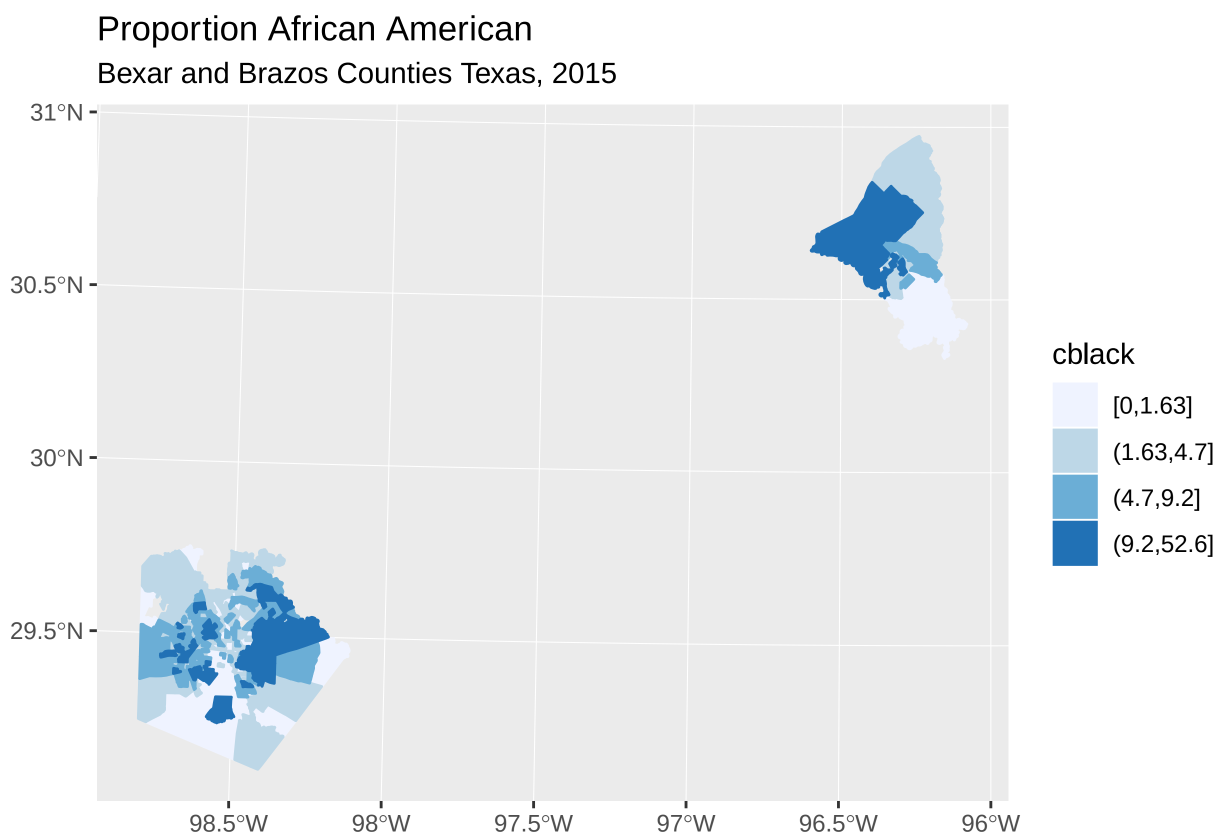
```
sa_acs<-get_acs(geography = "tract", state="TX", county = c("Bexar", "Brazos"), year = 2015,
                variables=c("DP05_0001E", "DP03_0009P", "DP03_0062E", "DP03_0119PE",
                            "DP05_0001E","DP02_0009PE","DP02_0008PE", "DP02_0040E","DP02_0038E",
                            "DP02_0066PE","DP02_0067PE","DP02_0080PE","DP02_0092PE",
                            "DP03_0005PE","DP03_0028PE","DP03_0062E","DP03_0099PE","DP03_0101PE",
                            "DP03_0119PE","DP04_0046PE","DP04_0078PE","DP05_0072PE","DP05_0073PE",
                            "DP05_0066PE", "DP05_0072PE", "DP02_0113PE") ,
                geometry = T, output = "wide")
```

```
## Getting data from the 2011-2015 5-year ACS

## Downloading feature geometry from the Census website.  To cache shapefiles for use in future session

## Using the ACS Data Profile
## Using the ACS Data Profile
```

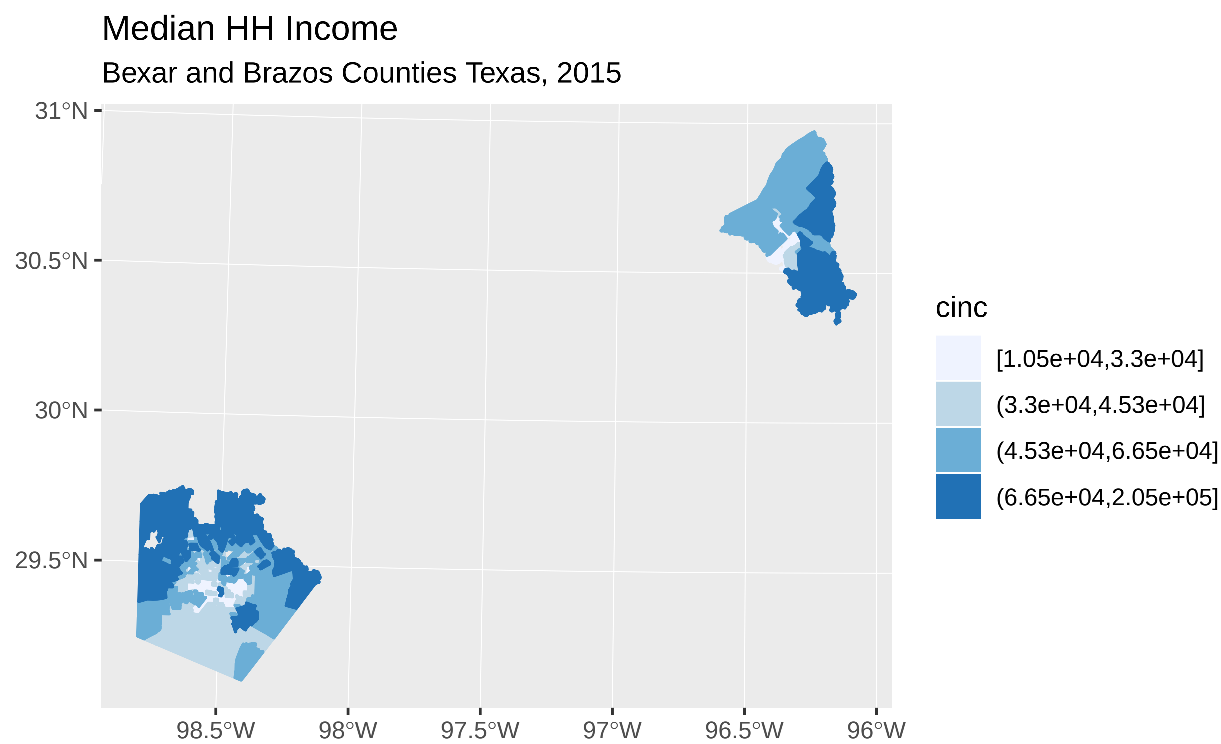
```
sa_acs$county<-substr(sa_acs$GEOID, 1, 5)

sa_acs2<-sa_acs%>%
  mutate(totpop= DP05_0001E, fertrate = DP02_0040E,pwhite=DP05_0072PE,
         pblack=DP05_0073PE , phisp=DP05_0066PE, pfemhh=DP02_0008PE,
         phsormore=DP02_0066PE,punemp=DP03_0009PE, medhhinc=DP03_0062E,
         ppov=DP03_0119PE, pforn=DP02_0092PE,plep=DP02_0113PE) %>%
  na.omit()
```

Some basic mapping of variables

```
sa_acs2 %>%
  mutate(cblack=cut(pblack,breaks = quantile(pblack, na.rm=T),
                    include.lowest = T))%>%
  ggplot(aes(fill = cblack, color = cblack)) +
  geom_sf() +
  ggtitle("Proportion African American",
          subtitle = "Bexar and Brazos Counties Texas, 2015")+
  coord_sf(crs =  102009) +
  scale_fill_brewer(palette = "Blues") +
  scale_color_brewer(palette = "Blues")
```




```
sa_acs2 %>%
  mutate(cinc=cut(medhhinc,breaks = quantile(medhhinc, na.rm=T),
                  include.lowest = T))%>%
  ggplot(aes(fill = cinc, color = cinc)) +
  geom_sf() +
  ggtitle("Median HH Income",
          subtitle = "Bexar and Brazos Counties Texas, 2015")+
  coord_sf(crs =  102009) +
  scale_fill_brewer(palette = "Blues") +
  scale_color_brewer(palette = "Blues")
```



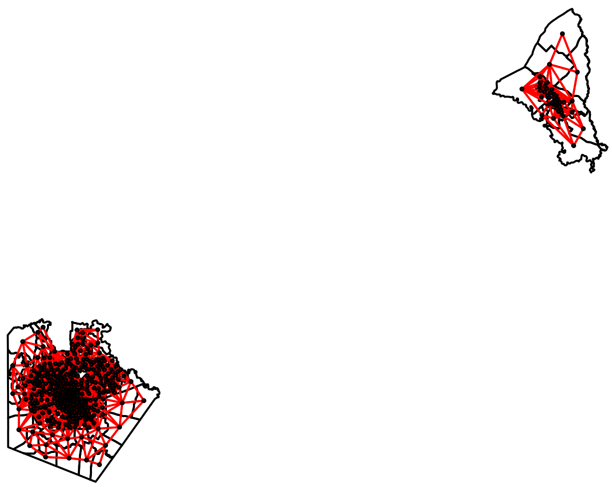
Spatial Representation of the data

```r
library(spdep)
```

```
## Loading required package: sp

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand

## Loading required package: spData

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source'))`
```

```r
sapoly<-as(sa_acs2, "Spatial")
nbs<-poly2nb(sapoly, queen = T)
wts<-nb2listw(nbs, style="W")
plot(sapoly,main="Queen Contiguity Spatial Neighbors" )
plot(nbs, coords=coordinates(sapoly), col=2, add=T, cex=.2 )
```

Queen Contiguity Spatial Neighbors



Basic spatial clustering of data

```
moran.mc(x=sapoly$medhhinc, listw = wts, nsim = 999, na.action = na.omit)
```

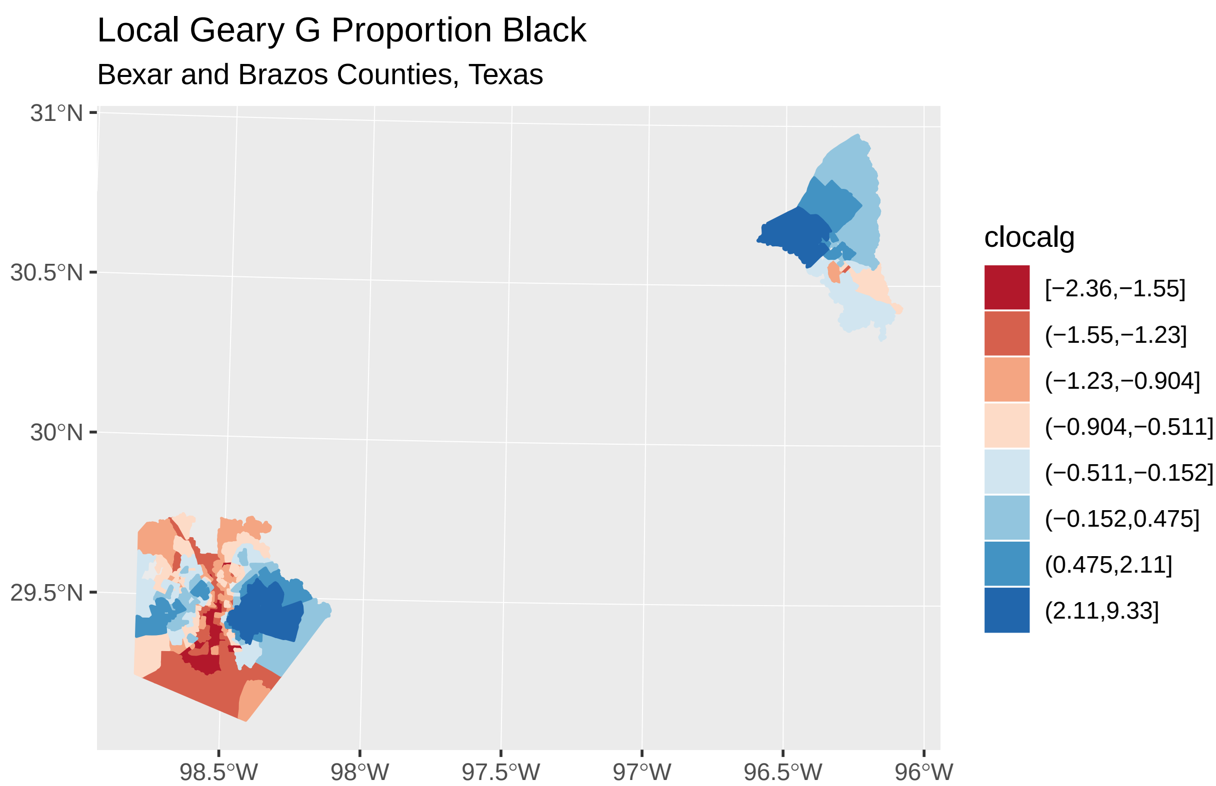
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  sapoly$medhhinc
## weights: wts
## number of simulations + 1: 1000
##
## statistic = 0.63437, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

```
moran.mc(x=sapoly$pblack, listw = wts, nsim = 999, na.action = na.omit)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  sapoly$pblack
## weights: wts
## number of simulations + 1: 1000
##
## statistic = 0.64936, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

```
sa_acs2$local_g<- localG(sapoly$pblack, listw = wts)
```

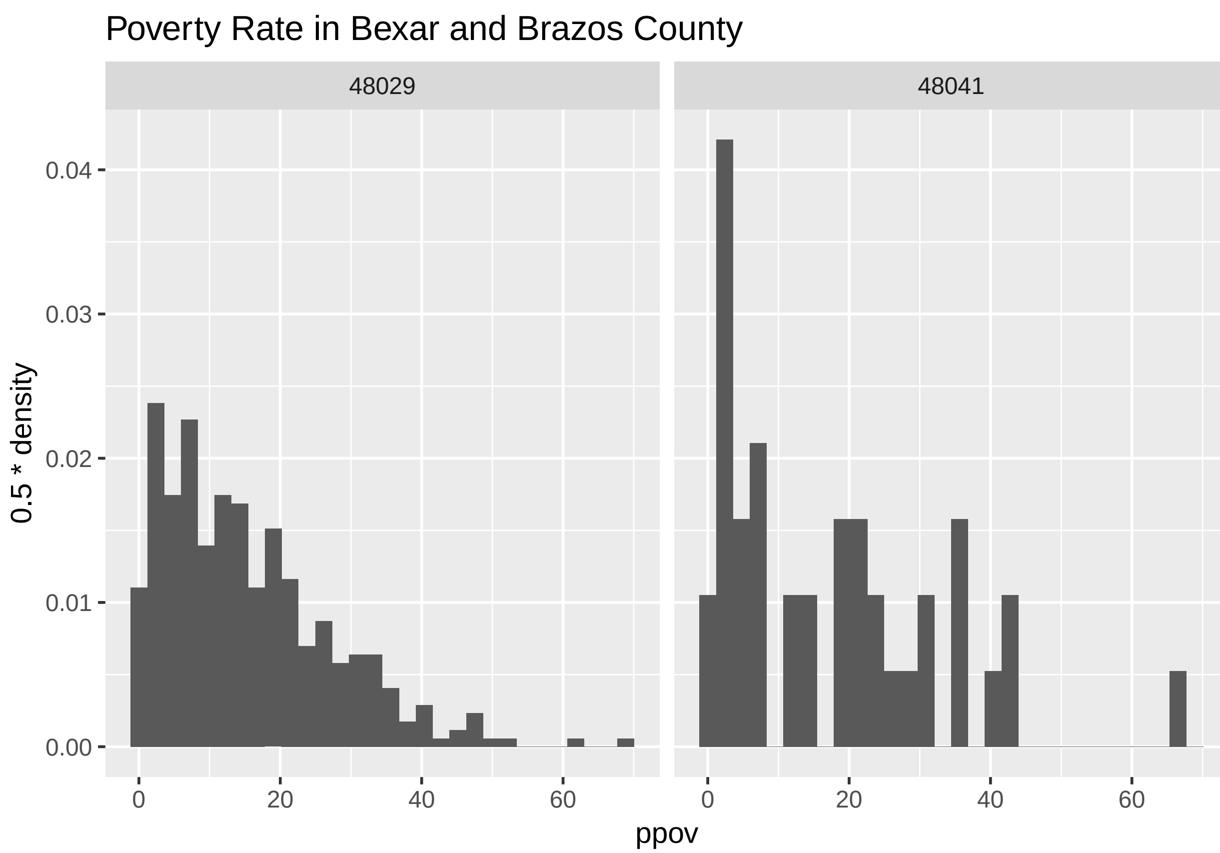
```
sa_acs2 %>%
  mutate(clocalg=cut(local_g, breaks = quantile(local_g, probs = seq(0,1, length.out = 9)),
                     include.lowest = T))%>%
  ggplot(aes(fill = clocalg, color = clocalg)) +
  geom_sf() +
  ggtitle("Local Geary G Proportion Black", subtitle = "Bexar and Brazos Counties, Texas")+
  coord_sf(crs =  102009) +
  scale_fill_brewer(palette = "RdBu") +
  scale_color_brewer(palette = "RdBu")
```



Descriptive analysis

```
sa_acs2%>%
  ggplot()+geom_histogram(aes(x =ppov , y=0.5*..density..))+
  facet_wrap(~county)+
  ggtitle(label = "Poverty Rate in Bexar and Brazos County")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



OLS regression models

```r
fit.1<-lm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc)+county, data=sa_acs2)
summary(fit.1)
```

```
##
## Call:
## lm(formula = phsormore ~ pforn + pblack + ppov + plep + punemp +
##     ppov + log(medhhinc) + county, data = sa_acs2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.9340  -3.4460  -0.0902   3.5301  27.0007
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    55.82949   12.53796   4.453 1.11e-05 ***
## pforn           0.71372    0.06916  10.320  < 2e-16 ***
## pblack          0.03095    0.03624   0.854 0.393632
## ppov           -0.22986    0.04454  -5.161 3.91e-07 ***
## plep           -1.46550    0.06981 -20.992  < 2e-16 ***
## punemp         -0.22391    0.08658  -2.586 0.010059 *
## log(medhhinc)   3.75401    1.09724   3.421 0.000688 ***
## county48041     0.84048    1.01322   0.830 0.407315
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 5.706 on 394 degrees of freedom
## Multiple R-squared:  0.844,  Adjusted R-squared:  0.8412
## F-statistic: 304.4 on 7 and 394 DF,  p-value: < 2.2e-16
```

```r
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:dplyr':
##
##     recode
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
#variance inflation factor for beta's
vif(fit.1)
```

```
##         pforn        pblack          ppov          plep        punemp
##      2.482648      1.142759      3.716357      4.026892      1.776046
## log(medhhinc)        county
##      3.665739      1.135699
```

```r
#Breush-Pagan test for constant variance
bptest(fit.1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit.1
## BP = 44.769, df = 7, p-value = 1.516e-07
```

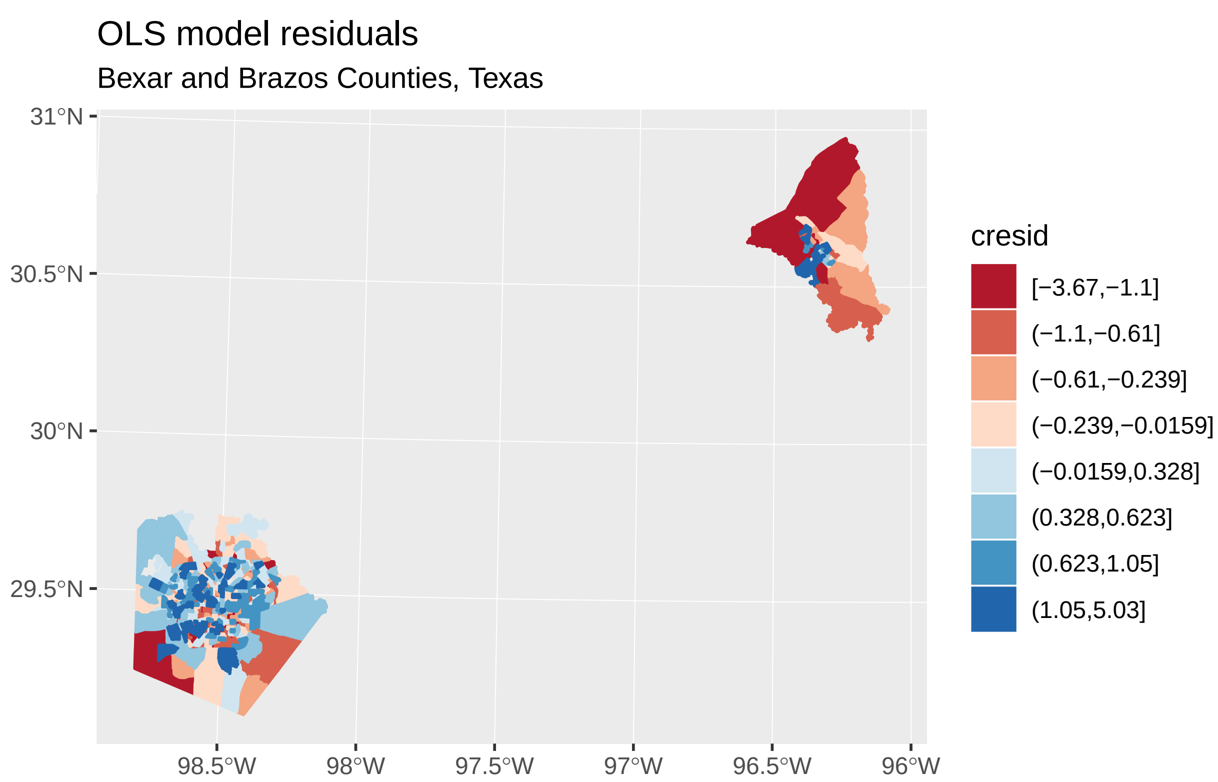
```r
#looks like we have heteroskedasticity

Anova(fit.1, white.adjust=T)
```

```
## Coefficient covariances computed by hccm()
## Analysis of Deviance Table (Type II tests)
##
## Response: phsormore
##                 Df        F    Pr(>F)
## pforn            1  70.4181 8.695e-16 ***
## pblack           1   0.4424 0.5063440
## ppov             1  13.8192 0.0002304 ***
## plep             1 247.5227 < 2.2e-16 ***
```

```
## punemp          1   5.5576 0.0188888 *
## log(medhhinc)   1   8.1796 0.0044619 **
## county          1   0.5522 0.4578764
## Residuals     394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Map of model residuals

```
#extract studentized residuals from the fit, and examine them
sa_acs2$residfit1<-rstudent(fit.1)
sa_acs2 %>%
  mutate(cresid=cut(residfit1,breaks = quantile(residfit1,p=seq(0,1,length.out = 9)),
                    include.lowest = T))%>%
  ggplot(aes(fill = cresid, color = cresid)) +
  geom_sf() +
  ggtitle("OLS model residuals", subtitle = "Bexar and Brazos Counties, Texas")+
  coord_sf(crs =  102009) +
  scale_fill_brewer(palette = "RdBu") +
  scale_color_brewer(palette = "RdBu")
```



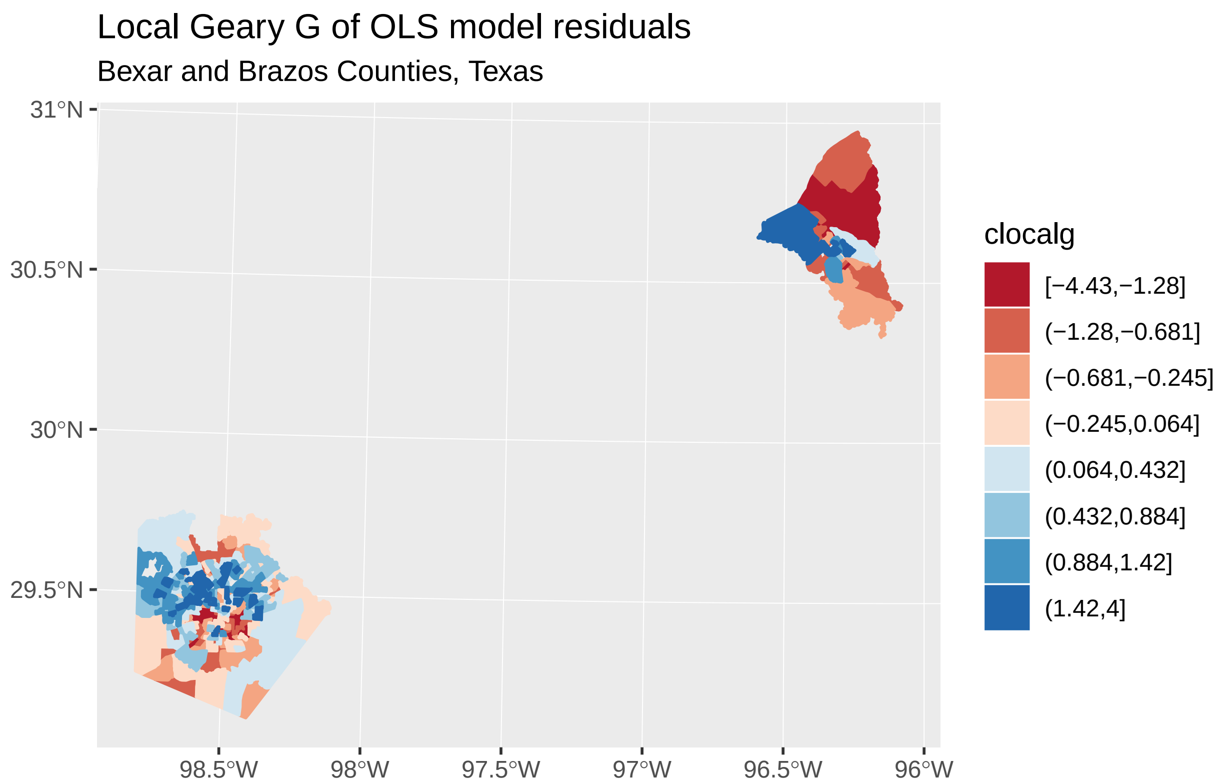
Analysis of residual clustering

```
#test for residual autocorrelation
lm.morantest(fit.1, listw=wts)
```

```
##
##  Global Moran I for regression residuals
```

```
##
## data:
## model: lm(formula = phsormore ~ pforn + pblack + ppov + plep +
## punemp + ppov + log(medhhinc) + county, data = sa_acs2)
## weights: wts
##
## Moran I statistic standard deviate = 6.1087, p-value = 5.022e-10
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation         Variance
##     0.1600961814    -0.0104723871     0.0007796511
```

```
#looks like we have significant autocorrelation in our residuals

#Let's look at the local autocorrelation in our residuals
#get the values of I
sa_acs2$local_G<-as.numeric(localG(sa_acs2$residfit1, wts))


sa_acs2 %>%
  mutate(clocalg=cut(local_G, breaks = quantile(local_G, probs = seq(0,1, length.out = 9)),
                     include.lowest = T))%>%
  ggplot(aes(fill = clocalg, color = clocalg)) +
  geom_sf() +
  ggtitle("Local Geary G of OLS model residuals", subtitle = "Bexar and Brazos Counties, Texas")+
  coord_sf(crs =  102009) +
  scale_fill_brewer(palette = "RdBu") +
  scale_color_brewer(palette = "RdBu")
```



Picking a spatial regression model

```r
#perform a model specification test
lm.LMtests(fit.1, listw=wts, test="all")
```

```
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = phsormore ~ pforn + pblack + ppov + plep +
## punemp + ppov + log(medhhinc) + county, data = sa_acs2)
## weights: wts
##
## LMerr = 31.091, df = 1, p-value = 2.462e-08
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = phsormore ~ pforn + pblack + ppov + plep +
## punemp + ppov + log(medhhinc) + county, data = sa_acs2)
## weights: wts
##
## LMlag = 112.29, df = 1, p-value < 2.2e-16
##
##
##  Lagrange multiplier diagnostics for spatial dependence
```

```
##
## data:
## model: lm(formula = phsormore ~ pforn + pblack + ppov + plep +
## punemp + ppov + log(medhhinc) + county, data = sa_acs2)
## weights: wts
##
## RLMerr = 0.034807, df = 1, p-value = 0.852
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = phsormore ~ pforn + pblack + ppov + plep +
## punemp + ppov + log(medhhinc) + county, data = sa_acs2)
## weights: wts
##
## RLMlag = 81.238, df = 1, p-value < 2.2e-16
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = phsormore ~ pforn + pblack + ppov + plep +
## punemp + ppov + log(medhhinc) + county, data = sa_acs2)
## weights: wts
##
## SARMA = 112.33, df = 2, p-value < 2.2e-16
```

```
#Now we fit the spatial lag model
#The lag mode is fit with lagsarlm() in the spdep library
#we basically specify the same model as in the lm() fit above
#But we need to specify the spatial weight matrix and the type
#of lag model to fit
```

Fit the spatial regression models

```
#Spatial Error model
fit.err<-errorsarlm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc)+county,
                    data=sa_acs2, listw=wts)
#summary(fit.err, Nagelkerke=T)



#Spatial Lag Model
fit.lag<-lagsarlm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc)+county,
                    data=sa_acs2, listw=wts, type="lag")
#summary(fit.lag, Nagelkerke=T)

#Spatial Durbin Lag Model
fit.lag2<-lagsarlm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc)+county,
                     data=sa_acs2, listw=wts, type="mixed")
#summary(fit.lag2, Nagelkerke=T)

#Spatial Durbin Error Model
fit.errdurb<-errorsarlm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc)+county,
                     data=sa_acs2, listw=wts, etype="emixed", method="spam")
```

```
## `validspamobject()` is deprecated. Use `validate_spam()` directly
```

```r
#summary(fit.errdurb, Nagelkerke=T)


#SAC Model
fit.sac<-sacsarlm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc)+county,
                   data=sa_acs2, listw=wts, type="sac", method="MC")
```

```
## Warning in sqrt(diag(fdHess)[-c(1, 2)]): NaNs produced
```

```r
#summary(fit.sac, Nagelkerke=T)

#SMA Model
fit.sma<-spautolm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc)+county,
                   data=sa_acs2, listw=wts, family="SMA")
#summary(fit.sma)

#which says we still have significant autocorrelation in the residuals, even after
#accounting for autocorrelation in the outcome
bptest.sarlm(fit.lag)
```

```
##
##  studentized Breusch-Pagan test
##
## data:
## BP = 31.395, df = 7, p-value = 5.257e-05
```

```r
bptest.sarlm(fit.lag2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:
## BP = 44.621, df = 13, p-value = 2.42e-05
```

```r
AIC(fit.lag)
```

```
## [1] 2430.949
```

```r
AIC(fit.lag2)
```

```
## [1] 2418.411
```

```r
AIC(fit.errdurb)
```

```
## [1] 2418.929
```

```r
AIC(fit.sac)
```

```
## [1] 2433.421
```

```r
AIC(fit.sma)
```

```
## [1] 2513.918
```

```r
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.1. https://CRAN.R-project.org/package=stargazer
```

```r
stargazer(fit.err, fit.lag, fit.lag2,header=FALSE, type='latex',
          ci = T,title = "Spatially autoregressive models",
          column.labels = c("Error Model", "Lag Model", "Durbin Lag Model"))
```

Table 1: Spatially autoregressive models

	Dependent variable:		
	phsormore		
	spatial error	*spatial autoregressive*	
	Error Model	Lag Model	Durbin Lag Model
	(1)	(2)	(3)
pforn	0.308***	0.381***	0.273***
	(0.163, 0.452)	(0.260, 0.502)	(0.134, 0.413)
pblack	0.045	−0.009	0.061
	(−0.055, 0.144)	(−0.069, 0.051)	(−0.045, 0.168)
ppov	−0.135***	−0.180***	−0.176***
	(−0.209, −0.061)	(−0.253, −0.107)	(−0.248, −0.103)
plep	−0.949***	−0.941***	−0.847***
	(−1.101, −0.798)	(−1.077, −0.804)	(−0.998, −0.697)
punemp	−0.172**	−0.188***	−0.166**
	(−0.318, −0.026)	(−0.330, −0.046)	(−0.308, −0.024)
log(medhhinc)	4.524***	1.044	2.855**
	(2.298, 6.751)	(−0.844, 2.933)	(0.671, 5.038)
county48041	3.603	0.091	−0.984
	(−2.368, 9.575)	(−1.583, 1.764)	(−2.778, 0.811)
lag.pforn			0.399***
			(0.162, 0.636)
lag.pblack			−0.100
			(−0.240, 0.040)
lag.ppov			−0.251***
			(−0.426, −0.076)
lag.plep			−0.336**
			(−0.653, −0.018)
lag.punemp			−0.042
			(−0.365, 0.280)
lag.log(medhhinc)			−7.389***
			(−11.214, −3.563)
Constant	43.998***	46.216***	115.959***
	(18.729, 69.268)	(25.675, 66.758)	(74.365, 157.553)
Observations	402	402	402
Log Likelihood	−1,235.543	−1,205.475	−1,193.205
σ^2	24.352	22.750	21.656
Akaike Inf. Crit.	2,491.085	2,430.949	2,418.411
Wald Test (df = 1)	272.941***	154.313***	28.550***
LR Test (df = 1)	61.860***	121.996***	23.967***
Note:			*p<0.1; **p<0.05; ***p<0.01

Spatial Impacts in lag models

```
#impacts for lag models
im.sdm<-impacts(fit.lag,listw=wts, R=100)
summary( im.sdm, zstats=TRUE)
```

```
## Impact measures (lag, exact):
##                     Direct     Indirect       Total
## pforn          0.395573976  0.280072981  0.67564696
## pblack        -0.009442059 -0.006685135 -0.01612719
## ppov          -0.186845084 -0.132289440 -0.31913452
## plep          -0.976673509 -0.691501154 -1.66817466
## punemp        -0.195121755 -0.138149461 -0.33327122
## log(medhhinc)  1.084438999  0.767800921  1.85223992
## county48041    0.094032339  0.066576466  0.16060881
## ========================================================
## Simulation results (asymptotic variance matrix):
## Direct:
##
## Iterations = 1:100
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 100
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean      SD Naive SE Time-series SE
## pforn          0.4002 0.05703 0.005703       0.005703
## pblack        -0.0111 0.02749 0.002749       0.002749
## ppov          -0.1856 0.03422 0.003422       0.003422
## plep          -0.9861 0.06699 0.006699       0.006699
## punemp        -0.1899 0.07068 0.007068       0.007068
## log(medhhinc)  1.0848 0.99701 0.099701       0.099701
## county48041    0.1378 0.83228 0.083228       0.083228
##
## 2. Quantiles for each variable:
##
##                   2.5%      25%      50%       75%    97.5%
## pforn          0.30041  0.36043  0.39398  0.435663  0.50450
## pblack        -0.06344 -0.02981 -0.01081  0.007477  0.04022
## ppov          -0.24351 -0.21066 -0.18706 -0.162260 -0.11512
## plep          -1.12485 -1.02578 -0.97921 -0.940348 -0.87873
## punemp        -0.33871 -0.23996 -0.17896 -0.140834 -0.07854
## log(medhhinc) -0.84441  0.44605  1.04241  1.795226  3.22203
## county48041   -1.58329 -0.30370  0.19115  0.713530  1.65400
##
## ========================================================
## Indirect:
##
## Iterations = 1:100
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 100
##
```

```
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                     Mean      SD Naive SE Time-series SE
## pforn           0.282778 0.04364 0.004364       0.004364
## pblack         -0.008279 0.01994 0.001994       0.001994
## ppov           -0.131284 0.02607 0.002607       0.002607
## plep           -0.698105 0.07566 0.007566       0.007566
## punemp         -0.134091 0.04986 0.004986       0.004986
## log(medhhinc)  0.763351 0.71015 0.071015       0.071015
## county48041    0.095568 0.58733 0.058733       0.058733
##
## 2. Quantiles for each variable:
##
##                    2.5%     25%       50%       75%    97.5%
## pforn           0.20304  0.2497  0.278549  0.313521  0.37308
## pblack         -0.04566 -0.0202 -0.007184  0.005234  0.02684
## ppov           -0.17598 -0.1470 -0.130235 -0.118292 -0.07827
## plep           -0.85070 -0.7432 -0.697626 -0.646432 -0.56322
## punemp         -0.23082 -0.1696 -0.133535 -0.096227 -0.05575
## log(medhhinc) -0.59400  0.3129  0.740958  1.263147  2.29366
## county48041   -1.09292 -0.2333  0.122693  0.503534  1.10233
##
## ===========================================================
## Total:
##
## Iterations = 1:100
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 100
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                    Mean      SD Naive SE Time-series SE
## pforn           0.68298 0.09354 0.009354       0.009354
## pblack         -0.01938 0.04735 0.004735       0.004735
## ppov           -0.31690 0.05758 0.005758       0.005758
## plep           -1.68421 0.11378 0.011378       0.007872
## punemp         -0.32396 0.11891 0.011891       0.011891
## log(medhhinc)  1.84817 1.70296 0.170296       0.170296
## county48041    0.23333 1.41753 0.141753       0.141753
##
## 2. Quantiles for each variable:
##
##                   2.5%     25%     50%      75%    97.5%
## pforn           0.5047  0.6111  0.6789  0.74952  0.86997
## pblack         -0.1090 -0.0488 -0.0183  0.01271  0.06762
## ppov           -0.4141 -0.3557 -0.3198 -0.28043 -0.19621
## plep           -1.9223 -1.7478 -1.6708 -1.61157 -1.46506
## punemp         -0.5526 -0.4044 -0.3128 -0.23651 -0.13657
## log(medhhinc) -1.4419  0.7591  1.7794  3.07390  5.51568
## county48041   -2.6530 -0.5494  0.3090  1.21779  2.70795
##
```

```
## ========================================================
## Simulated z-values:
##                    Direct   Indirect       Total
## pforn           7.0172197  6.4797357   7.3012070
## pblack         -0.4038482 -0.4152134  -0.4093015
## ppov           -5.4234289 -5.0348702  -5.5037453
## plep          -14.7211399 -9.2269679 -14.8020886
## punemp         -2.6862452 -2.6891085  -2.7244943
## log(medhhinc)   1.0880675  1.0749208   1.0852698
## county48041     0.1655277  0.1627152   0.1646058
##
## Simulated p-values:
##               Direct     Indirect   Total
## pforn         2.2633e-12 9.1883e-11 2.8511e-13
## pblack        0.686324   0.6779857  0.68232
## ppov          5.8466e-08 4.7817e-07 3.7181e-08
## plep          < 2.22e-16 < 2.22e-16 < 2.22e-16
## punemp        0.007226   0.0071643  0.00644
## log(medhhinc) 0.276565   0.2824102  0.27780
## county48041   0.868529   0.8707426  0.86925
```

Spatial Regimes, split data by unemployment rate

```
sa_acs2$unemp_cut<-cut(sa_acs2$punemp, breaks = quantile(sa_acs2$punemp,
                                                          p=c(0, .33, .66, 1)))
table(sa_acs2$unemp_cut)
```

(0,5.4] (5.4,8.6] (8.6,33.8] 134 131 136

```
fit.low<-lm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc),
            data=sa_acs2, subset=unemp_cut=="(0,5.4]")

fit.med<-lm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc),
            data=sa_acs2, subset=unemp_cut=="(5.4,8.6]")

fit.hi<-lm(phsormore~pforn+pblack+ppov+plep +punemp+ppov+log(medhhinc),
           data=sa_acs2, subset=unemp_cut=="(8.6,33.8]")

stargazer(fit.low, fit.med, fit.hi,header=FALSE, type='latex',
          ci = T,title = "Spatial regime models",
          column.labels = c("Low Unemployment", "Moderate Unemployment", "High Unemployment"))
```

Table 2: Spatial regime models

	Dependent variable:		
	phsormore		
	Low Unemployment	Moderate Unemployment	High Unemployment
	(1)	(2)	(3)
pforn	0.401^{***}	0.580^{***}	0.977^{***}
	(0.213, 0.589)	(0.376, 0.785)	(0.724, 1.231)
pblack	−0.081	0.156^{**}	−0.017
	(−0.224, 0.063)	(0.026, 0.286)	(−0.123, 0.088)
ppov	-0.246^{***}	-0.264^{***}	−0.086
	(−0.396, −0.097)	(−0.410, −0.119)	(−0.241, 0.070)
plep	-0.929^{***}	-1.710^{***}	-1.664^{***}
	(−1.153, −0.705)	(−1.911, −1.509)	(−1.900, −1.429)
punemp	−0.052	−0.253	-0.353^{*}
	(−0.681, 0.576)	(−1.232, 0.727)	(−0.722, 0.016)
log(medhhinc)	4.518^{***}	0.539	8.531^{***}
	(1.645, 7.392)	(−2.656, 3.734)	(3.620, 13.442)
Constant	46.225^{***}	95.580^{***}	4.212
	(12.770, 79.680)	(57.760, 133.399)	(−50.485, 58.908)
Observations	134	131	136
R^2	0.758	0.874	0.805
Adjusted R^2	0.746	0.868	0.795
Residual Std. Error	4.213 (df = 127)	4.860 (df = 124)	6.659 (df = 129)
F Statistic	66.208^{***} (df = 6; 127)	143.524^{***} (df = 6; 124)	88.520^{***} (df = 6; 129)
Note:			*p<0.1; **p<0.05; ***p<0.01

```r
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 15063)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] stargazer_5.2.1    lmtest_0.9-36      zoo_1.8-1
##  [4] car_3.0-0          carData_3.0-1      spdep_0.7-7
##  [7] spData_0.2.8.3     Matrix_1.2-14      sp_1.2-7
## [10] bindrcpp_0.2.2     forcats_0.3.0      stringr_1.3.0
## [13] dplyr_0.7.4        purrr_0.2.4        readr_1.1.1
## [16] tidyr_0.8.0        tibble_1.4.2       ggplot2_2.2.1.9000
## [19] tidyverse_1.2.1    tidycensus_0.4.6
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-137       sf_0.6-2           lubridate_1.7.4
##  [4] gmodels_2.16.2     RColorBrewer_1.1-2 httr_1.3.1
##  [7] rprojroot_1.3-2    tools_3.5.0        backports_1.1.2
## [10] rgdal_1.2-20       R6_2.2.2           DBI_1.0.0
## [13] lazyeval_0.2.1     colorspace_1.3-2   withr_2.1.2
## [16] mnormt_1.5-5       curl_3.2           compiler_3.5.0
## [19] cli_1.0.0          rvest_0.3.2        expm_0.999-2
## [22] xml2_1.2.0         labeling_0.3       scales_0.5.0.9000
## [25] classInt_0.2-3     psych_1.8.4        rappdirs_0.3.1
## [28] digest_0.6.15      foreign_0.8-70     rmarkdown_1.9
## [31] rio_0.5.10         pkgconfig_2.0.1    htmltools_0.3.6
## [34] rlang_0.2.0.9001   readxl_1.1.0       rstudioapi_0.7
## [37] bindr_0.1.1        jsonlite_1.5       gtools_3.5.0
## [40] magrittr_1.5       dotCall64_0.9-5.2  Rcpp_0.12.16
## [43] munsell_0.4.3      RPostgreSQL_0.6-2  abind_1.4-5
## [46] stringi_1.2.2      yaml_2.1.19        MASS_7.3-50
## [49] plyr_1.8.4         grid_3.5.0         maptools_0.9-2
## [52] parallel_3.5.0     gdata_2.18.0       crayon_1.3.4
## [55] udunits2_0.13      deldir_0.1-15      lattice_0.20-35
## [58] haven_1.1.1        splines_3.5.0      hms_0.4.2
## [61] knitr_1.20         pillar_1.2.2       uuid_0.1-2
## [64] boot_1.3-20        reshape2_1.4.3     LearnBayes_2.15.1
## [67] glue_1.2.0         evaluate_0.10.1    data.table_1.11.0
## [70] modelr_0.1.1       spam_2.1-4         cellranger_1.1.0
## [73] gtable_0.2.0       assertthat_0.2.0   openxlsx_4.0.17
```

```
## [76] broom_0.4.4        e1071_1.6-8        coda_0.19-1
## [79] class_7.3-14       tigris_0.7         units_0.5-1
```