Week 10: Effects of Spatial Dependence

```
## Loading required package: sp

## Loading required package: Matrix

## rgdal: version: 1.2-16, (SVN revision 701)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.1.2, released 2016/10/24
##  Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/3.3/Resources/library/rgdal/gdal
##  GDAL binary built with GEOS: FALSE
##  Loaded PROJ.4 runtime: Rel. 4.9.1, 04 March 2015, [PJ_VERSION: 491]
##  Path to PROJ.4 shared files: /Library/Frameworks/R.framework/Versions/3.3/Resources/library/rgdal/pr
##  Linking to sp version: 1.2-5

## Checking rgeos availability: TRUE
```

Modeling areal data

Spatial statistical approach

Simultaneous autoregressive model (SAR)

The SAR specification uses a regression on the values from the other areas to account for the spatial dependence. This means that the error terms e are modelled so that they depend on each other in the following way:

$$e_i = \sum_{i=1}^{m} b_{ij} e_i + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma_i^2)$

or

$$e_1, e_2, \ldots, e_n \sim N(0, (I - B)^{-1} D((I - B)^{-1})^T)$$

If denote $e_i = Y - X^T\beta$, then the SAR model could be expressed as:

$Y - X^T\beta = B(Y - X^T\beta) + \epsilon$ or $(I - B)(Y - X^T\beta) = \epsilon$

where B is a matrix that contains the dependence parameters b_{ij} and I is the identity matrix of the required dimension. It is important to point out that in order for this SAR model to be well defined, the matrix $I - B$ must be non-singular.

A common practice to specify B is $B = \rho W$, where W is the spatial weight matrix and ρ is the `spatial autoregression parameter`.

Frequently employed in the spatial econometric literature with two variants, spatial lag model and spatial error model.

Conditional autoregressive model (CAR)

The CAR specification relies on the conditional distribution of the spatial error terms. In this case, the distribution of e_i conditioning on e_{-i} is given.

$$e_i | e_{-i} \sim N(\sum_j b_{ij} e_j, \tau_i^2)$$

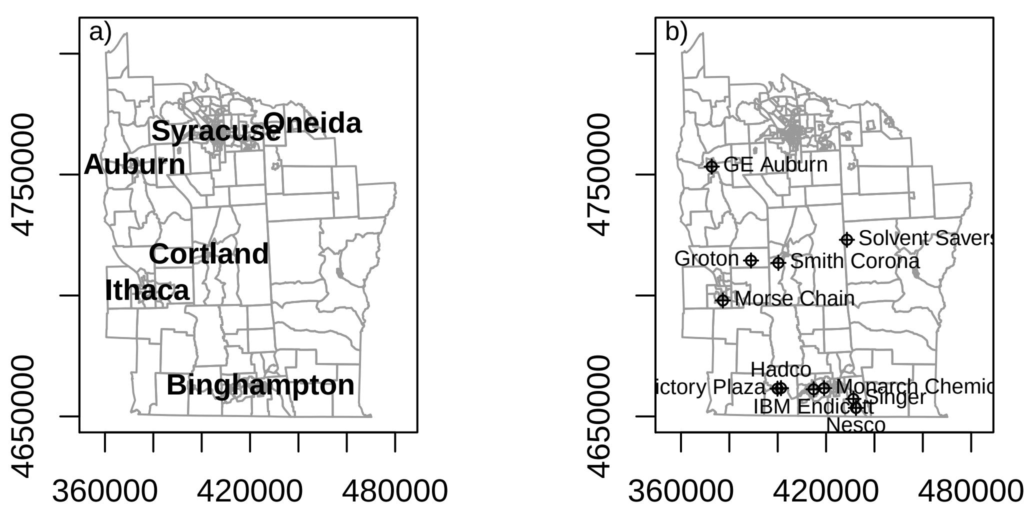
where j is the neighbors of i.

Using Brook's Lemma we can obtain:

$$e_1, e_2, \ldots, e_n \sim N(0, (I - B)^{-1} D)$$

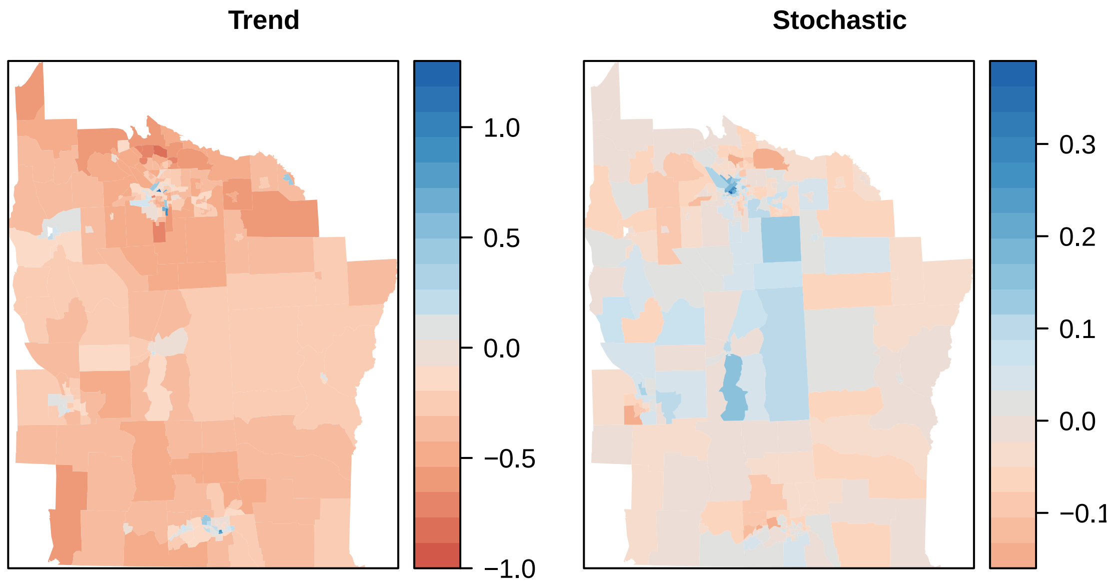
Compared to SAR model, the CAR model is easier to be fitted into a Bayesian hierarchical models.

```
# display the maps

NY8 <- readOGR("Data", "NY8_utm18")
TCE <- readOGR("Data", "TCE")
cities <- readOGR("Data", "NY8cities")

par(mfrow=c(1,2))
plot(NY8, border="grey60", axes=TRUE)
text(coordinates(cities), labels=as.character(cities$names), font=2, cex=0.9)
text(bbox(NY8)[1,1], bbox(NY8)[2,2], labels="a)", cex=0.8)

plot(NY8, border="grey60", axes=TRUE)
points(TCE, pch=1, cex=0.7)
points(TCE, pch=3, cex=0.7)
text(coordinates(TCE), labels=as.character(TCE$name), cex=0.7,
 font=1, pos=c(4,1,4,1,4,4,4,2,3,4,2), offset=0.3)
text(bbox(NY8)[1,1], bbox(NY8)[2,2], labels="b)", cex=0.8)
```



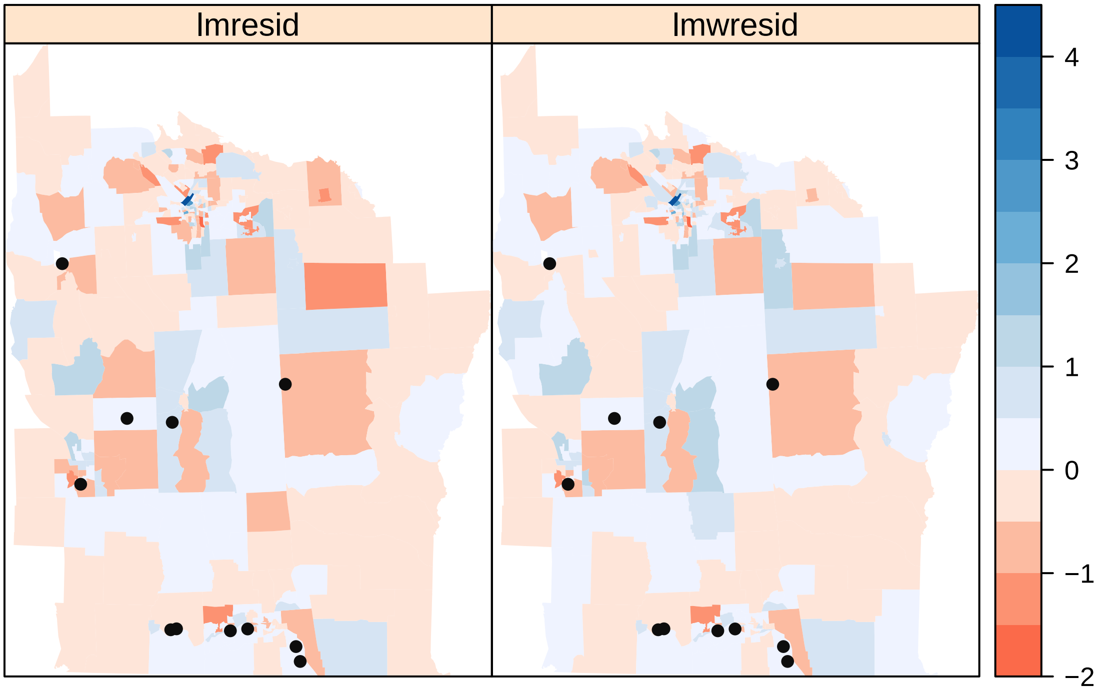

```
# ordinary linear regression

nylm <- lm(Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME, data=NY8)
summary(nylm)
NY8$lmresid <- residuals(nylm)
NY_nb <- read.gal("Data/NY_nb.gal", region.id=row.names(NY8))
NYlistw<-nb2listw(NY_nb, style = "B")

# Moran's I
```

```
lm.morantest(nylm, NYlistw)


# Spatial conditional and simultaneous autoregression

nysar<-spautolm(Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME, data=NY8, listw=NYlistw)

summary(nysar)


nylam1 <- c(nysar$lambda)
nylam2 <- c(LR1.spautolm(nysar)$p.value)
# Notice above that there is still strong spatial autocorrelation, and the proximity to TCE seems not q

# Display the trend and residual component
NY8$sar_trend <- nysar$fit$signal_trend
NY8$sar_stochastic <- nysar$fit$signal_stochastic
rds <- colorRampPalette(brewer.pal(8, "RdBu"))
tr_at <- seq(-1, 1.3, length.out=21)
tr_rds <- rds(sum(tr_at >= 0)*2)[-(1:(sum(tr_at >= 0)-sum(tr_at < 0)))]
tr_pl <- spplot(NY8, c("sar_trend"), at=tr_at, col="transparent", col.regions=tr_rds, main=list(label="'
st_at <- seq(-0.16, 0.39, length.out=21)
st_rds <- rds(sum(st_at >= 0)*2)[-(1:(sum(st_at >= 0)-sum(st_at < 0)))]
st_pl <- spplot(NY8, c("sar_stochastic"), at=st_at, col="transparent", col.regions=st_rds, main=list(lal
plot(tr_pl, split=c(1,1,2,1), more=TRUE)
plot(st_pl, split=c(2,1,2,1), more=FALSE)
```

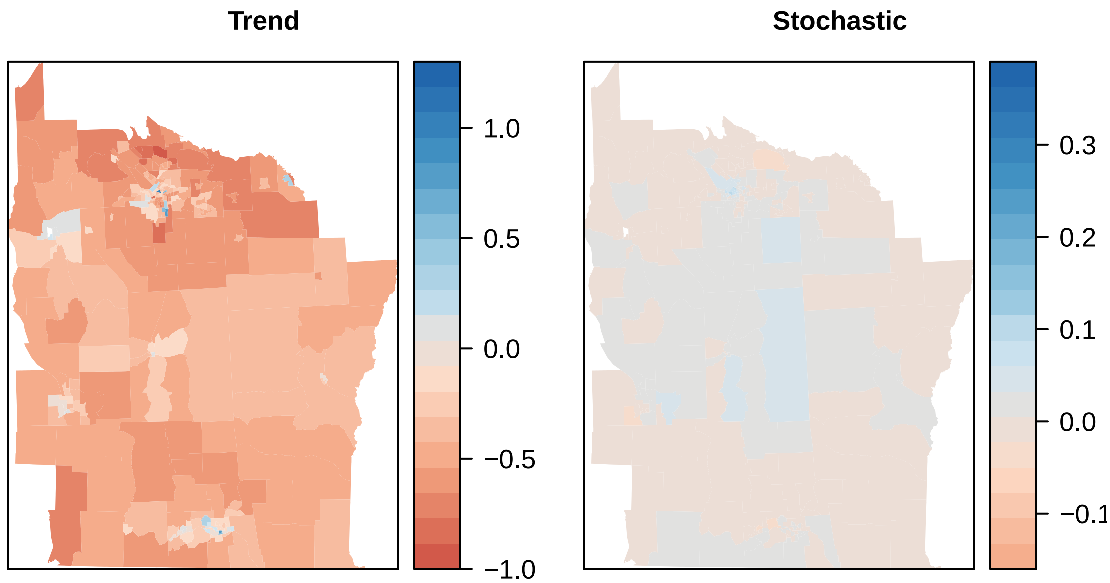



```
# The proximity to a TCE seems not to be significant, after we include
# the population as weights, it becomes significant.

nylmw <- lm(Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME, data=NY8, weights=POP8)
summary(nylmw)
NY8$lmwresid <- residuals(nylmw)

# Display
```

```
gry <- c(rev(brewer.pal(6, "Reds")[1:4]), colorRampPalette(brewer.pal(5, "Blues"))(9))
TCEpts <- list("sp.points", TCE, pch=16, col="grey5")
spplot(NY8, c("lmresid", "lmwresid"), sp.layout=list(TCEpts), col.regions=gry, col="transparent", lwd=0
```




```
# Now check the moran's again

lm.morantest(nylmw, NYlistw)

# Include weights in spautolm

nysarw<-spautolm(Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME , data=NY8, listw=NYlistw, weights=POP8)
summary(nysarw)

NY8$sarw_trend <- nysarw$fit$signal_trend
NY8$sarw_stochastic <- nysarw$fit$signal_stochastic
tr_pl <- spplot(NY8, c("sarw_trend"), at=tr_at, col="transparent", col.regions=tr_rds, main=list(label=
st_pl <- spplot(NY8, c("sarw_stochastic"), at=st_at, col="transparent", col.regions=st_rds, main=list(la
plot(tr_pl, split=c(1,1,2,1), more=TRUE)
plot(st_pl, split=c(2,1,2,1), more=FALSE)
```




```r
# Conditional Autoregressive Models

nycar<-spautolm(Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME , data=NY8, family="CAR",
   listw=NYlistw)
summary(nycar)



nycarw<-spautolm(Z~PEXPOSURE+PCTAGE65P+PCTOWNHOME, data=NY8, family="CAR",
   listw=NYlistw, weights=POP8)
summary(nycarw)
```

Geographically Weighted Regression

Works well for non-stationary cases:

$$Y_i = \beta_0(i) + \beta_1(i)x_1 + \ldots + \beta_p(i)x_p$$

with the estimator:

$\beta(i) = (X^T W(i) X)^{-1} X^T W(i) Y$, where $W(i)$ is a matrix of weights spefici to i-th location such that observations nearer to this location are given greater weight nearer to i are given greater weight than observations further away. Typically, W can be otained from a weighting (or kernel) function.

```r
library(spgwr)

bwG <- gwr.sel(Z ~ PEXPOSURE + PCTAGE65P + PCTOWNHOME, data = NY8, gweight = gwr.Gauss, verbose = FALSE)

gwrG <- gwr(Z ~ PEXPOSURE + PCTAGE65P + PCTOWNHOME, data = NY8, bandwidth = bwG, gweight = gwr.Gauss, ha

# Applied to GLM (generalized linear model) case

gbwG <- ggwr.sel(Cases~PEXPOSURE+PCTAGE65P+PCTOWNHOME+offset(log(POP8)), data=NY8, family="poisson", gw

ggwrG <- ggwr(Cases~PEXPOSURE+PCTAGE65P+PCTOWNHOME+offset(log(POP8)), data=NY8, family="poisson", bandw
```

```r
# GWR local coefficient estimates for the exposure to TCE site covariate

TCEpts <- list("sp.points", TCE, pch=16, col="grey5")
spplot(ggwrG$SDF, "PEXPOSURE", sp.layout=list(TCEpts), col.regions=grey.colors(7, 0.95, 0.55, 2.2), cuts
```