R: Spatial autocorrelation

Roger Bivand

Thursday, 5 September 2019, 14:15-14:30

Spatial autocorrelation

```
library(sf)
```

```
## Linking to GEOS 3.7.2, GDAL 3.0.1, PROJ 6.2.0
```

```
lux <- st_read("../data/lux_tmerc.gpkg")
```

```
## Reading layer `lux_tmerc' from data source `/home/rsb/presentations/ectqg19-workshop/data/lux_tmerc.g
## Simple feature collection with 102 features and 16 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 48930.89 ymin: 57015.29 xmax: 106113.8 ymax: 138759.2
## epsg (SRID):    NA
## proj4string:    +proj=tmerc +lat_0=49.8333333333333 +lon_0=6.16666666666667 +k=1 +x_0=80000 +y_0=100
```

Contiguity Queen neighbours

```
library(spdep)
```

```
## Loading required package: sp
```

```
## Loading required package: spData
```

```
nb_cont <- poly2nb(lux, row.names=as.character(lux$LAU2))
lw_B <- nb2listw(nb_cont, style="B")
lw_W <- nb2listw(nb_cont) # default style="W"
```

Global spatial autocorrelation

Basic global Moran's I test under randomisation (analytical permutation, differs from test under normality by a kurtosis term in the variance), first with binary weights

```
moran.test(lux$light_level, listw=lw_B, randomisation=TRUE, alternative="two.sided")
```

```
##
##  Moran I test under randomisation
##
## data:  lux$light_level
## weights: lw_B
##
## Moran I statistic standard deviate = 4.7986, p-value = 1.598e-06
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.25764008       -0.00990099        0.00310848
```

Next with row standardised weights

```r
(mi <- moran.test(lux$light_level, listw=lw_W, randomisation=TRUE, alternative="two.sided"))
```

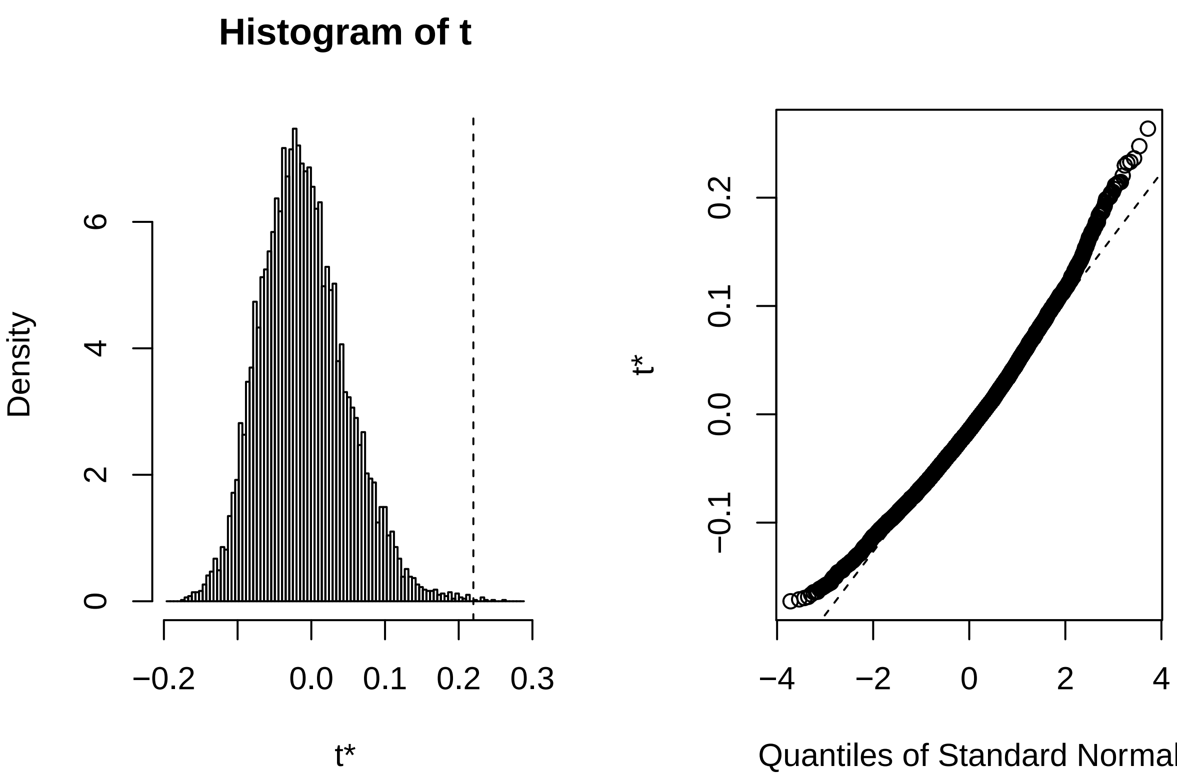
```
##
##  Moran I test under randomisation
##
## data:  lux$light_level
## weights: lw_W
##
## Moran I statistic standard deviate = 3.9672, p-value = 7.273e-05
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.219888588      -0.009900990       0.003355057
```

The permutation test gives very similar outcomes to the standard test under randomisation

```r
set.seed(1)
perm_boot <- moran.mc(lux$light_level, listw=lw_W, nsim=9999, return_boot=TRUE)
c(mean=mean(perm_boot$t), var=var(perm_boot$t))
```

```
##        mean         var
## -0.01020404  0.00340162
```

```r
plot(perm_boot)
```



Under normality, the outcome is also similar despite kurtosis playing a role

```r
moran(lux$light_level, listw=lw_W, S0=Szero(lw_W), n=nrow(lux))$K
```

```
## [1] 15.23457
```

```r
moran.test(lux$light_level, listw=lw_W, randomisation=FALSE, alternative="two.sided")
```

```
##
##  Moran I test under normality
##
## data:  lux$light_level
## weights: lw_W
##
## Moran I statistic standard deviate = 3.7084, p-value = 0.0002086
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.219888588      -0.009900990       0.003839686
```

The test for regression residuals for a null model (intercept only) is the same as the basic test under normality

```r
OLS0 <- lm(light_level ~ 1, lux)
lm.morantest(OLS0, listw=lw_W, alternative="two.sided")
```

```
##
##  Global Moran I for regression residuals
##
## data:
## model: lm(formula = light_level ~ 1, data = lux)
## weights: lw_W
##
## Moran I statistic standard deviate = 3.7084, p-value = 0.0002086
## alternative hypothesis: two.sided
## sample estimates:
## Observed Moran I      Expectation         Variance
##      0.219888588     -0.009900990      0.003839686
```

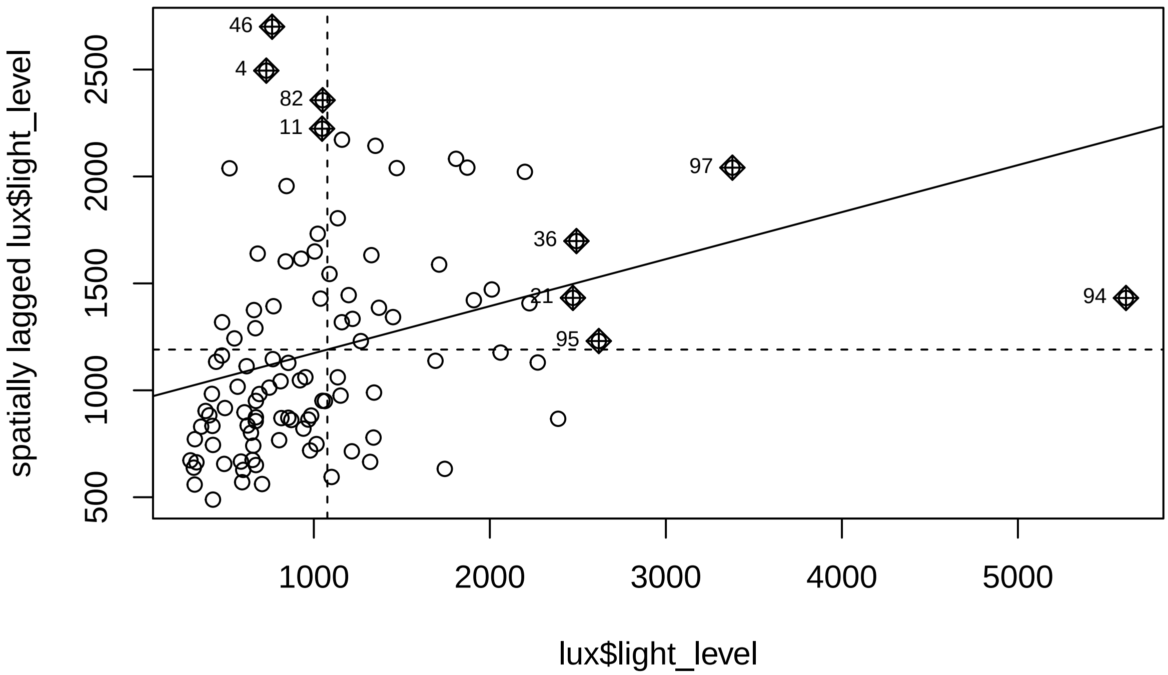
All the tests so far depend on assumptions, so calculating the exact variance may give different results

```r
lm.morantest.exact(OLS0, listw=lw_W, alternative="two.sided")
```

```
##
##  Global Moran I statistic with exact p-value
##
## data:
## model:lm(formula = light_level ~ 1, data = lux)
## weights: lw_W
##
## Exact standard deviate = 3.3871, p-value = 0.0007063
## alternative hypothesis: two.sided
## sample estimates:
## [1] 0.2198886
```

The Moran scatterplot shows the by observation relationship between the observed values and their spatial lags. The marked points are observations exerting stronger influence on the linear Moran relationship

```r
moran.plot(lux$light_level, listw=lw_W)
```



But maybe this autocorrelation is rather driven by a missing covariate in the mean model. The basic test is of a mean model including only the intercept, the mean of the variable being tested. If we include a relevant covariate such here as population density with a similar spatial footprint to the variable of interest, conclusions may change

```r
OLS <- lm(light_level ~  pop_den, lux)
summary(OLS)
```

```
##
## Call:
## lm(formula = light_level ~ pop_den, data = lux)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1225.7  -418.8  -145.5   185.5  3149.2
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 871.3159    84.0395  10.368  < 2e-16 ***
## pop_den       0.7087     0.1669   4.246 4.89e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 693.7 on 100 degrees of freedom
## Multiple R-squared:  0.1527, Adjusted R-squared:  0.1443
## F-statistic: 18.03 on 1 and 100 DF,  p-value: 4.887e-05
```

As we can see, there is much less autocorrelation in the residuals of an updated mean model

```r
lm.morantest(OLS, listw=lw_W, alternative="two.sided")
```

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

```
##
## data:
## model: lm(formula = light_level ~ pop_den, data = lux)
## weights: lw_W
##
## Moran I statistic standard deviate = 2.2291, p-value = 0.02581
## alternative hypothesis: two.sided
## sample estimates:
## Observed Moran I      Expectation         Variance
##      0.122316539     -0.014379855      0.003760716
```

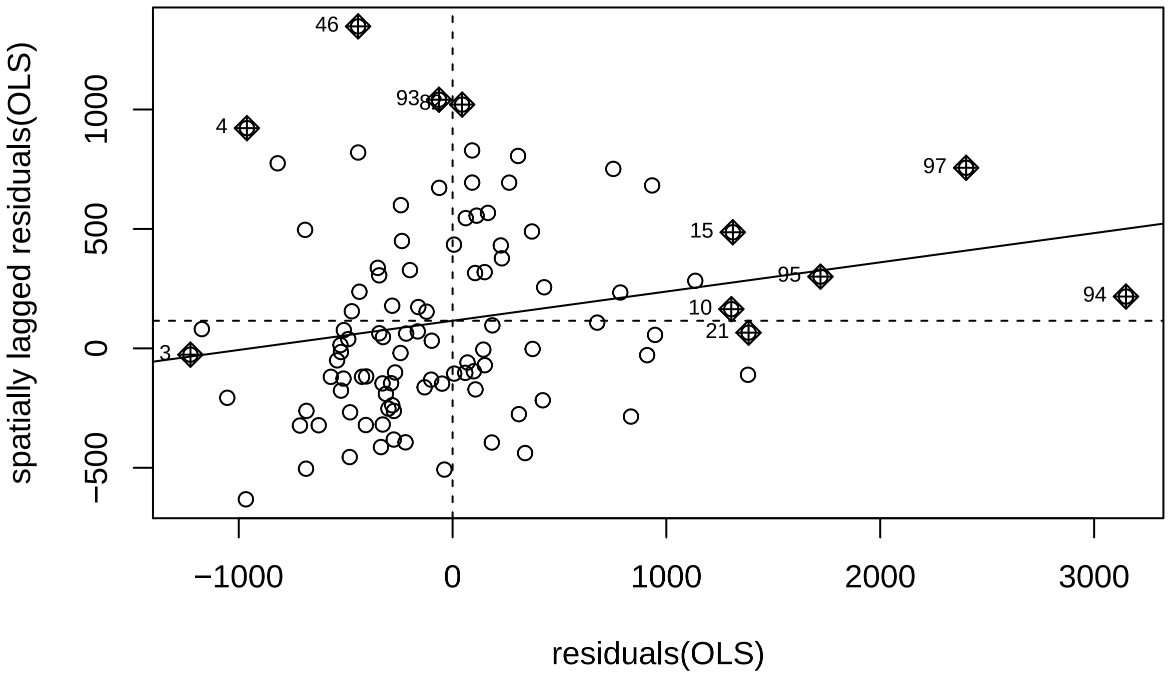
And using the exact test, there is little spatial autocorrelation left

```
(mie <- lm.morantest.exact(OLS, listw=lw_W, alternative="two.sided"))
```

```
##
##  Global Moran I statistic with exact p-value
##
## data:
## model:lm(formula = light_level ~ pop_den, data = lux)
## weights: lw_W
##
## Exact standard deviate = 2.1172, p-value = 0.03425
## alternative hypothesis: two.sided
## sample estimates:
## [1] 0.1223165
```

We can also see that the slope of the Moran relationship is much flatter

```
moran.plot(residuals(OLS), listw=lw_W)
```



Local autocorrelation

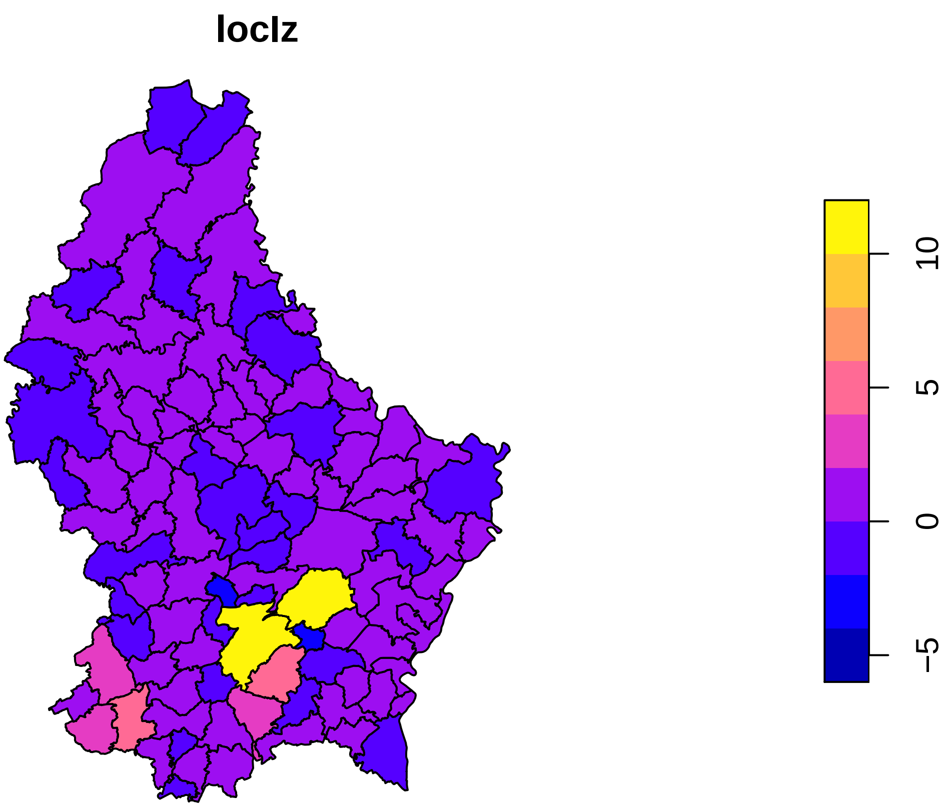
There are a number of functions in R to calculate local Moran's I_i, inclusing the standard measure with variances calculated under randomisation. The sum of the local Moran's I_i values divided by the sum of the weights is the same as the value of global Moran's I

```
locm <- localmoran(lux$light_level, listw=lw_W, alternative="two.sided")
all.equal(sum(locm[,1])/Szero(lw_W), mi$estimate[1], check.attributes=FALSE)
```

```
## [1] TRUE
```

Plots will show indicator standard deviates; not that most "hot-spot"/"cluster" maps forget that probability values must be adjusted for the false discovery rate due to multiple comparison. For this reason **spdep** does not offer such potentially misleading functionalities

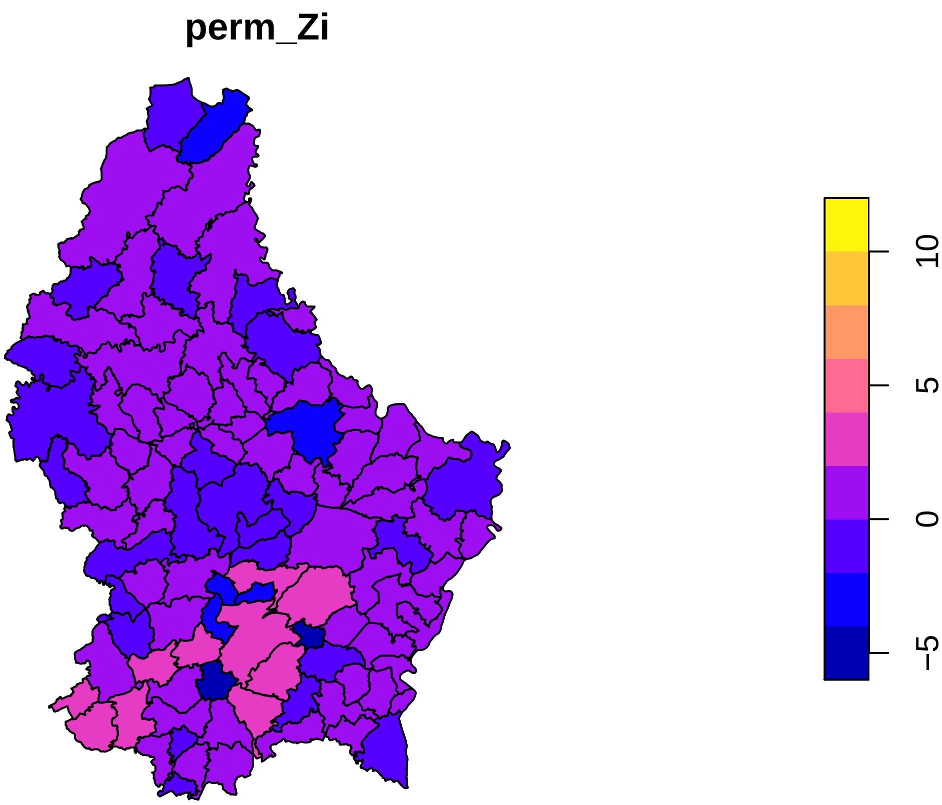
```
lux$locIz <- locm[,4]
plot(lux[,"locIz"], breaks=seq(-6, 12, 2))
```



It is possible to bootstrap this measure by sampling from the complete set of observations omitting the observation of interest, but this is not (yet) provided as there are better alternatives (saddlepoint approximation and exact computations offering possibly richer mean models *and* the removal of global spatial autocorrelation which pollutes local measures); here is a roll-your-own bootstrap

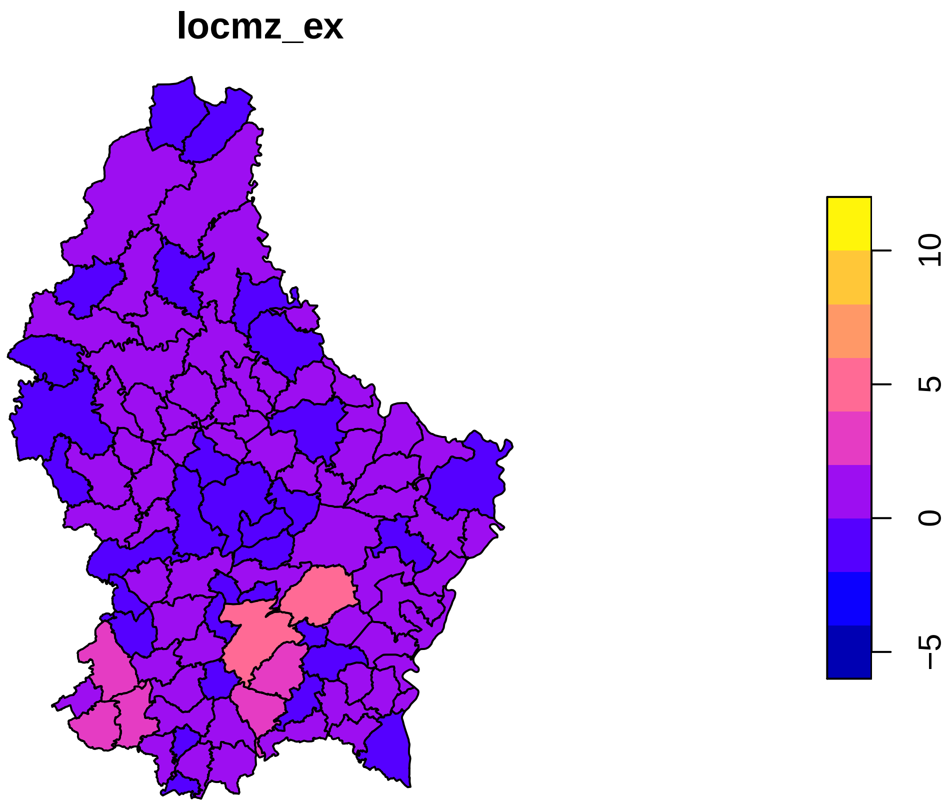
```
x <- lux$light_level
lw <- lw_W
xx <- mean(x)
z <- x - xx
s2 <- sum(z^2)/length(x)
crd <- card(lw$neighbours)
nsim <- 999
res_p <- numeric(nsim)
```

```
mns <- sds <- numeric(length(x))
set.seed(1)
for (i in seq(along=x)) {
  wtsi <- lw$weights[[i]]
  zi <- z[i]
  z_i <- z[-i]
  crdi <- crd[i]
  if (crdi > 0) {
    for (j in 1:nsim) {
      sz_i <- sample(z_i, size=crdi)
      lz_i <- sum(sz_i*wtsi)
      res_p[j] <- (zi/s2)*lz_i
    }
    mns[i] <- mean(res_p)
    sds[i] <- sd(res_p)
  } else {
    mns[i] <- as.numeric(NA)
    sds[i] <- as.numeric(NA)
  }
}
lux$perm_Zi <- (locm[,1] - mns)/sds
plot(lux[, "perm_Zi"], breaks=seq(-6, 12, 2))
```



The exact measures are more comptationally intensive, but are well supported in theory

```
locm_ex <- localmoran.exact(OLS0, nb=nb_cont, style="W", alternative="two.sided")
lux$locmz_ex <- as.data.frame(locm_ex)[,4]
plot(lux[,"locmz_ex"], breaks=seq(-6, 12, 2))
```

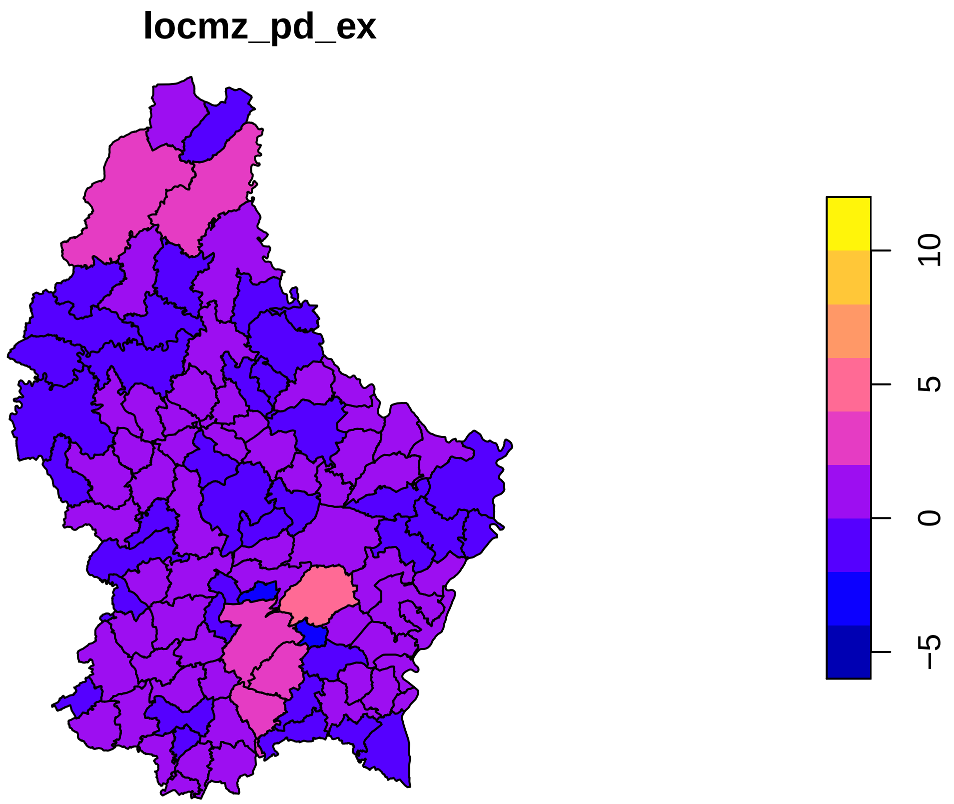


It is better to use the exact measures for the upgraded mean model including the population density; moderate or small global residual autocorrelation may mask local spatial association, even though the summation relationship stil holds

```
locm_pop_den_ex <- as.data.frame(localmoran.exact(OLS, nb=nb_cont, style="W", alternative="two.sided"))
all.equal(sum(locm_pop_den_ex[,1])/Szero(lw_W), mie$estimate[1], check.attributes=FALSE)
```

```
## [1] TRUE
```

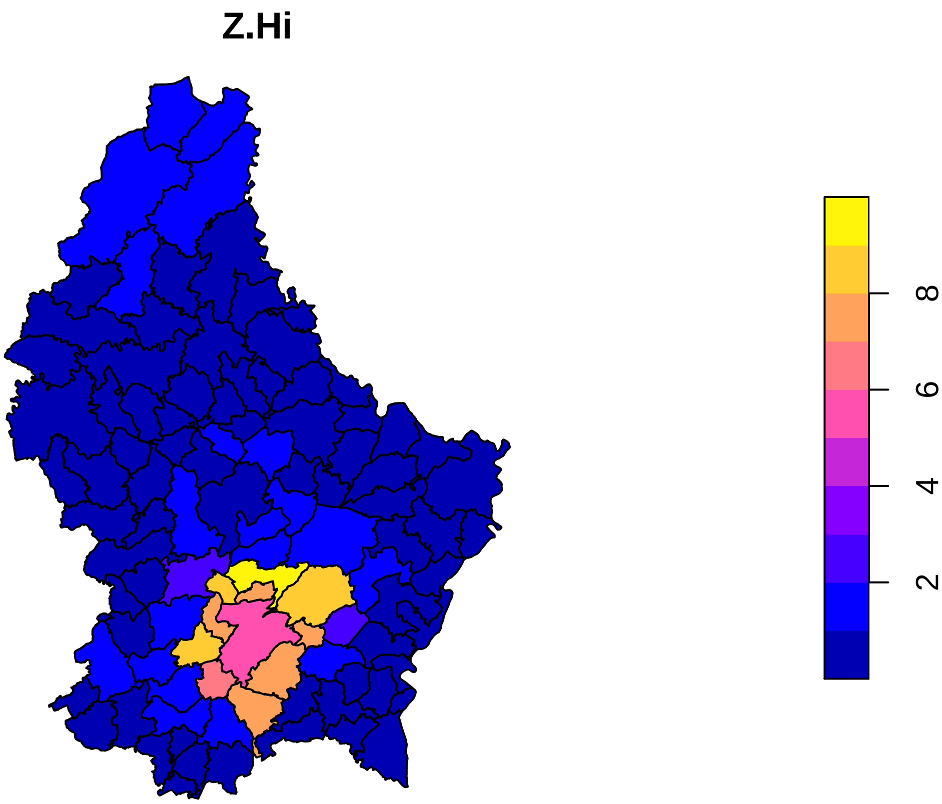
```
lux$locmz_pd_ex <- locm_pop_den_ex[,4]
plot(lux[,"locmz_pd_ex"], breaks=seq(-6, 12, 2))
```



Spatial heterogeneity statistic

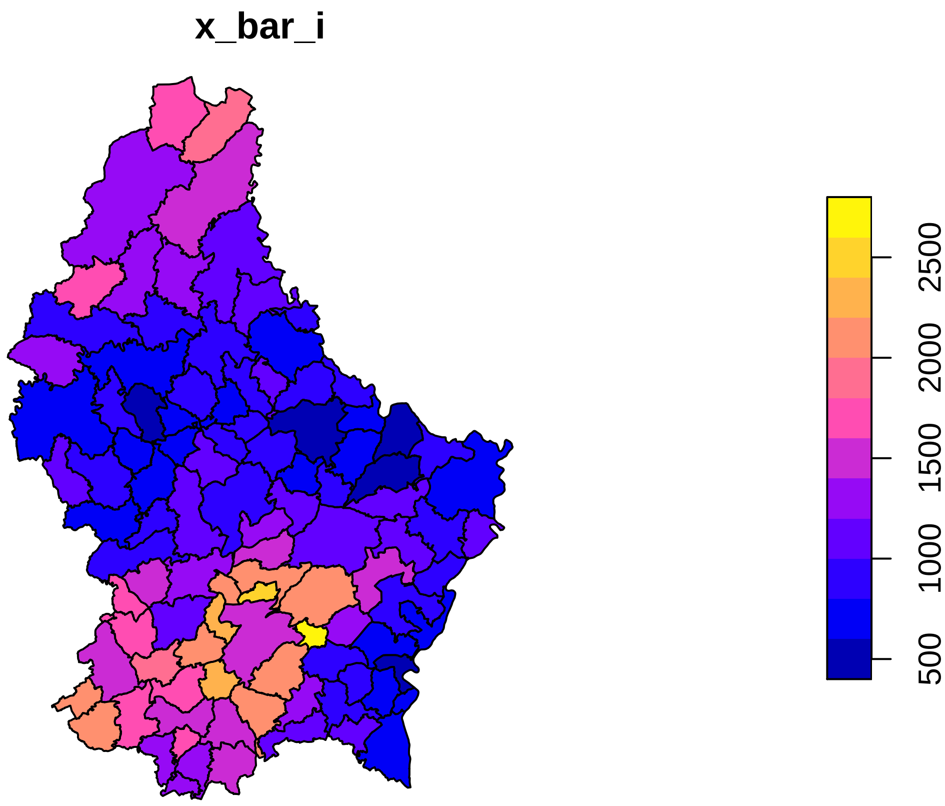
The latest addition to local indicators of spatial mis-specification is LOSH, directed at showing spatial heteroskedasticity (local variations in value variability); here the standard deviate of the measure

```
LOSH <- LOSH.cs(lux$light_level, listw=lw_W)
lux$Z.Hi <- LOSH[,4]
plot(lux[,"Z.Hi"])
```



These also offer a spatially smoothed view of the variable of interest

```
lux$x_bar_i <- LOSH[,5]
plot(lux[,"x_bar_i"])
```


x_bar_i
500
1000
1500
2000
2500