Recalculate IMD

Reka Solymosi

2023-05-16

## Preparing the data

For this demo we will need data about an outcome - in this example Burglary - at Local Super Output Area (LSOA) level. We can get this from the data.police.uk website. Here we have one year’s worth of data from March 2022 to March 2023 for Cleveland police.

# read in monthly ASB counts from Cleveland Police from March 2022 to March 2023   
# downloaded from data.police.uk  
nm <- list.files(path="data/police\_uk\_data")  
open\_data <- do.call(rbind, lapply(paste0("data/police\_uk\_data/",nm), function(x) read.csv(file=x)))  
# filter only ASB  
asb\_data <- open\_data %>% dplyr::filter(Crime.type == "Anti-social behaviour") %>% clean\_names()  
# filter only Burglary  
burglary\_data <- open\_data %>% dplyr::filter(Crime.type == "Burglary") %>% clean\_names()  
  
# filter only Violence and sexual offences  
violence\_data <- open\_data %>% dplyr::filter(Crime.type == "Violence and sexual offences") %>% clean\_names()  
  
# remove rest of crime data  
rm(open\_data)

We will also need the Index of Multiple Deprivation 2019 data. We will download “File 5” which contains the scores for the IMD and the individual domains - this is so we can reproduce the score with the transformed scores, and the supplementary “File 9” which contains the transformed domain scores, which are used to calculate the overall deprivation score, and what we will use to re-calculate this overall score without the crime indicator.

# read in 2019 Index of Multiple Deprivation (IMD) data  
# available to download from gov.uk  
imd\_data <- read\_excel("data/imd\_data/File\_5\_-\_IoD2019\_Scores.xlsx",   
 sheet = "IoD2019 Scores") %>% clean\_names()  
# also import "file 9" which contains the transformed domain scores  
transformed\_imd <- read\_excel("data/imd\_data/File\_9\_-\_IoD2019\_Transformed\_Scores.xlsx",   
 sheet = "IoD2019 Transformed Scores") %>%   
 clean\_names()

Finally we need a geography with LSOAs for Cleveland area. Here we use the 2011 Census boundaries file available from the ONS Geography portal. We first download all LSOAs for England and Wales, and then use the police force area boundary from the data.police.uk website in order to select only those areas that fall within the Cleveland police jurisdiction.

Although 2021 Census boundaries are available, the 2019 IMD was created using the 2011 boundaries therefore for analysis using the IMD2019 we must use these geographies.

# read in all LSOA for England and Wales (downloaded from ONS geography portal)  
lsoa\_boundaries <- st\_read("data/LSOA\_2011\_Boundaries\_Super\_Generalised\_Clipped\_BSC\_EW\_V4\_-6793269404754981576.geojson")

## Reading layer `LSOA\_2011\_Boundaries\_Super\_Generalised\_Clipped\_BSC\_EW\_V4\_-6793269404754981576' from data source `/Users/user/Dropbox (The University of Manchester)/IMD\_without\_crime\_indicator/recalculating\_imd/data/LSOA\_2011\_Boundaries\_Super\_Generalised\_Clipped\_BSC\_EW\_V4\_-6793269404754981576.geojson'   
## using driver `GeoJSON'  
## Simple feature collection with 34753 features and 9 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: 82678 ymin: 5343 xmax: 655604.7 ymax: 657534.1  
## Projected CRS: OSGB36 / British National Grid

# We can get force boundaries from the data.police.uk site   
cleveland\_force\_boundary <- st\_read("data/cleveland.kml")

## Reading layer `Layer #0' from data source   
## `/Users/user/Dropbox (The University of Manchester)/IMD\_without\_crime\_indicator/recalculating\_imd/data/cleveland.kml'   
## using driver `KML'  
## Simple feature collection with 1 feature and 2 fields  
## Geometry type: MULTIPOLYGON  
## Dimension: XYZ  
## Bounding box: xmin: -1.452617 ymin: 54.46415 xmax: -0.788388 ymax: 54.72717  
## z\_range: zmin: 0 zmax: 0  
## Geodetic CRS: WGS 84

# Let's make sure they are the same projection (British National Grid)  
cleveland\_force\_boundary <- st\_transform(cleveland\_force\_boundary, 27700)  
# st\_crs(cleveland\_force\_boundary) == st\_crs(lsoa\_boundaries)  
  
# add small buffer around the PFA, since the boundary is not generalised whereas the LSOAs are  
cleveland\_force\_boundary <- st\_buffer(cleveland\_force\_boundary, 300)  
  
# intersection - identify all LSOAs contained in PFA  
cleveland\_lsoa\_contains <- st\_contains(cleveland\_force\_boundary, lsoa\_boundaries)  
# subsetting - select only these LSOAs  
cleveland\_lsoa\_contains <- lsoa\_boundaries[unlist(cleveland\_lsoa\_contains),]

Great, now we have all our data we can join these together. To join the IMD data sets we can simply use the matching LSOA code columns

# join the imd file with the overall IMD score  
cleveland\_w\_imd <- left\_join(cleveland\_lsoa\_contains, imd\_data %>%   
 select(lsoa\_code\_2011, lsoa\_name\_2011, local\_authority\_district\_code\_2019,   
 local\_authority\_district\_name\_2019, index\_of\_multiple\_deprivation\_imd\_score),   
 by = c("LSOA11CD" = "lsoa\_code\_2011"))  
# then join the imd file with the transformed scores  
cleveland\_w\_imd <- left\_join(cleveland\_w\_imd, transformed\_imd %>%   
 select(-c(lsoa\_name\_2011, local\_authority\_district\_code\_2019,   
 local\_authority\_district\_name\_2019)), by = c("LSOA11CD" = "lsoa\_code\_2011"))

Finally count number of ASB incidents per LSOA and join this as well.

# count number of ASB incidents per LSOA  
asb\_by\_lsoa <- asb\_data %>%   
 group\_by(lsoa\_code) %>%   
 summarise(count\_of\_asb = n())  
  
# join to existing dataframe  
cleveland\_w\_imd <- left\_join(cleveland\_w\_imd, asb\_by\_lsoa, by = c("LSOA11CD" = "lsoa\_code"))

Or if we want to look at burglary then the number of burglary incidents per LSOA and join this as well.

# count number of burglary incidents per LSOA  
burg\_by\_lsoa <- burglary\_data %>%   
 group\_by(lsoa\_code) %>%   
 summarise(count\_of\_burglary = n())  
  
# join to existing dataframe  
cleveland\_w\_imd <- left\_join(cleveland\_w\_imd, burg\_by\_lsoa, by = c("LSOA11CD" = "lsoa\_code"))

Or if we want to look at burglary then the number of Violence and sexual offences incidents per LSOA and join this as well.

# count number of burglary incidents per LSOA  
violence\_by\_lsoa <- violence\_data %>%   
 group\_by(lsoa\_code) %>%   
 summarise(count\_of\_violence = n())  
  
# join to existing dataframe  
cleveland\_w\_imd <- left\_join(cleveland\_w\_imd, violence\_by\_lsoa, by = c("LSOA11CD" = "lsoa\_code"))

## Calculating the IMD score

Now that we have all our variables in one dataset, we can calculate the IMD score from its individual component. Let’s start by demonstrating the proof of concept, still including the crime indicator. So in this case, we want to create a weighted sum of the transformed scores of the indicators, in order to re-create the IMD score, following the instructions in the technical documentation for the IMD.

The total IMD score is made up of the 7 key indicators plus weighting:

* Income Score - exponentially transformed - 22.5%
* Employment Score - exponentially transformed - 22.5%
* Education, Skills and Training Score - exponentially transformed - 13.5%
* Health Deprivation and Disability Score - exponentially transformed - 13.5%
* Crime Score - exponentially transformed - 9.3%
* Barriers to Housing and Services Score - exponentially transformed - 9.3%
* Living Environment Score - exponentially transformed - 9.3%

Let’s use the exponentially tranformed scores (released in table 9) to recreate the IMD Score (available from table 5):

cleveland\_w\_imd$manual\_imd <- cleveland\_w\_imd$income\_score\_exponentially\_transformed\* 0.225+  
 cleveland\_w\_imd$employment\_score\_exponentially\_transformed\*0.225+   
 cleveland\_w\_imd$education\_score\_exponentially\_transformed\*0.135 +   
 cleveland\_w\_imd$health\_score\_exponentially\_transformed\*0.135 +   
 cleveland\_w\_imd$crime\_score\_exponentially\_transformed\*0.093 +  
 cleveland\_w\_imd$barriers\_score\_exponentially\_transformed\* 0.093 +  
 cleveland\_w\_imd$living\_environment\_score\_exponentially\_transformed\*0.093

We can now check whether this manually calculated IMD matches with the IMD score presented in table 5. First, we will notice that in table 5 these scores are rounded to 3 decimal places, so we do that first. Then we can compare the two vectors:

cleveland\_w\_imd$manual\_imd <- round(cleveland\_w\_imd$manual\_imd, 3)  
identical(cleveland\_w\_imd$manual\_imd, cleveland\_w\_imd$index\_of\_multiple\_deprivation\_imd\_score)

## [1] FALSE

**NOTE:** So in 34 cases, the match is not identical. It appears to be an issue of rounding, because they are always off by 0.001. I have emailed the relevant team to ask about why this is happening as I couldn’t find anything about rounding in the technical documentation of the IMD.

If we wish to recalculate the IMD without the crime indicator, we simply remove this, and adjust the weights of the other indicators, so they add up to 1. There is detail on this in *Appendix B* of the IMD Research report (Ministry of Housing, Communities and Local Government, 2019b). To do this we simply re-weight the other 6 domain weights in the same ratio to ensure that the total weight is still 1.00 (i.e. divide them all by 1-0.093). Thus the 22.5% weight increases to 24.81%, the 13.5% to to 14.88% and the 9.3% to 10.25%.

# Make the IMD without the crime indicator  
# Same weights but to sum up to 1 need to divide by 1-0.093  
  
cleveland\_w\_imd$no\_crime\_imd <- (cleveland\_w\_imd$income\_score\_exponentially\_transformed\* 0.225+  
 cleveland\_w\_imd$employment\_score\_exponentially\_transformed\*0.225+   
 cleveland\_w\_imd$education\_score\_exponentially\_transformed\*0.135 +   
 cleveland\_w\_imd$health\_score\_exponentially\_transformed\*0.135 +   
 cleveland\_w\_imd$barriers\_score\_exponentially\_transformed\* 0.093 +  
 cleveland\_w\_imd$living\_environment\_score\_exponentially\_transformed\*0.093)/(1-0.093)

## Brief example illustrating the recalculated IMD

Let’s focus on our example research question now. Is there an association between area-level deprivation and number of ASB incidents in Cleveland. If we use our re-calculated IMD score, we can avoid the issues mentioned above.

Let’s use here a simple spatial error model, which accounts here for the possible spatial autocorrelation in our data.

# Calculate the weights  
#We coerce the sf object into a new sp object  
cleveland\_sp <- as(cleveland\_w\_imd, "Spatial")  
# queen contiguity neighbours list  
nb\_queen <- poly2nb(cleveland\_sp, row.names=cleveland\_sp$OBJECTID)  
# create row standardised weights matrix  
wm\_queen\_rs <- nb2mat(nb\_queen, style='W')  
# create a list of weights for neighbouringness (using row standardised weights)  
lw\_queen <- nb2listw(nb\_queen, style='W')  
# # spatial error model  
# error\_mod <- errorsarlm(count\_of\_asb ~ no\_crime\_imd, data=cleveland\_sp, lw\_queen)  
# summary(error\_mod)  
  
# spatial error model for burglary  
error\_mod <- errorsarlm(count\_of\_burglary ~ no\_crime\_imd, data=cleveland\_sp, lw\_queen)  
summary(error\_mod)

##   
## Call:  
## errorsarlm(formula = count\_of\_burglary ~ no\_crime\_imd, data = cleveland\_sp,   
## listw = lw\_queen)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -25.0907 -5.6124 -1.8771 2.6960 85.1269   
##   
## Type: error   
## Coefficients: (asymptotic standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 5.194003 1.982845 2.6195 0.008807  
## no\_crime\_imd 0.287560 0.037294 7.7107 1.243e-14  
##   
## Lambda: 0.6039, LR test value: 73.348, p-value: < 2.22e-16  
## Asymptotic standard error: 0.056063  
## z-value: 10.772, p-value: < 2.22e-16  
## Wald statistic: 116.03, p-value: < 2.22e-16  
##   
## Log likelihood: -1344.591 for error model  
## ML residual variance (sigma squared): 128.35, (sigma: 11.329)  
## Number of observations: 346   
## Number of parameters estimated: 4   
## AIC: 2697.2, (AIC for lm: 2768.5)

How would the results differ if we hadn’t recalculated the IMD without crime, but instead just used the aggregate IMD score?

# # spatial error model  
# error\_mod\_all\_imd <- errorsarlm(count\_of\_asb ~ index\_of\_multiple\_deprivation\_imd\_score, data=cleveland\_sp, lw\_queen)  
# summary(error\_mod\_all\_imd)  
  
# spatial error model for burglary  
error\_mod\_all\_imd <- errorsarlm(count\_of\_burglary ~ index\_of\_multiple\_deprivation\_imd\_score, data=cleveland\_sp, lw\_queen)  
summary(error\_mod\_all\_imd)

##   
## Call:  
## errorsarlm(formula = count\_of\_burglary ~ index\_of\_multiple\_deprivation\_imd\_score,   
## data = cleveland\_sp, listw = lw\_queen)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -24.9450 -5.4787 -1.7970 2.7133 85.5044   
##   
## Type: error   
## Coefficients: (asymptotic standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 4.772495 1.943207 2.4560 0.01405  
## index\_of\_multiple\_deprivation\_imd\_score 0.305904 0.038052 8.0391 8.882e-16  
##   
## Lambda: 0.58866, LR test value: 67.516, p-value: 2.2204e-16  
## Asymptotic standard error: 0.057302  
## z-value: 10.273, p-value: < 2.22e-16  
## Wald statistic: 105.53, p-value: < 2.22e-16  
##   
## Log likelihood: -1343.177 for error model  
## ML residual variance (sigma squared): 127.9, (sigma: 11.309)  
## Number of observations: 346   
## Number of parameters estimated: 4   
## AIC: 2694.4, (AIC for lm: 2759.9)

Our conclusions don’t change dramatically when deploying the revised measure. However, we observe the model is slightly worse (larger AIC) when deploying the revised IMD measure and the coefficient for the effect of IMD on ASB is slightly smaller. So for each increase in IMD score, there are 0.31 more burglary incidents in the LSOA, compared with 0.29 more if we look at the IMD without the crime indicator as a predictor. This is in line with what we would expect, as including the thing itself in the model will make the correlation stronger artificially. But it does not seem to make big differences to the conclusions we would draw from these data - that is that there is a positive association between deprivation score and burglary.

## Violence

# spatial error model for burglary  
error\_mod <- errorsarlm(count\_of\_violence ~ no\_crime\_imd, data=cleveland\_sp, lw\_queen)  
summary(error\_mod)

##   
## Call:  
## errorsarlm(formula = count\_of\_violence ~ no\_crime\_imd, data = cleveland\_sp,   
## listw = lw\_queen)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -134.4099 -27.9277 -8.9451 10.8201 927.2520   
##   
## Type: error   
## Coefficients: (asymptotic standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 10.14244 11.21823 0.9041 0.3659  
## no\_crime\_imd 2.78595 0.25031 11.1301 <2e-16  
##   
## Lambda: 0.42951, LR test value: 31.996, p-value: 1.5447e-08  
## Asymptotic standard error: 0.068507  
## z-value: 6.2695, p-value: 3.6225e-10  
## Wald statistic: 39.306, p-value: 3.6225e-10  
##   
## Log likelihood: -2059.108 for error model  
## ML residual variance (sigma squared): 6804.3, (sigma: 82.488)  
## Number of observations: 352   
## Number of parameters estimated: 4   
## AIC: 4126.2, (AIC for lm: 4156.2)

and for the whole IMD:

# spatial error model for burglary  
error\_mod\_all\_imd <- errorsarlm(count\_of\_violence ~ index\_of\_multiple\_deprivation\_imd\_score, data=cleveland\_sp, lw\_queen)  
summary(error\_mod\_all\_imd)

##   
## Call:  
## errorsarlm(formula = count\_of\_violence ~ index\_of\_multiple\_deprivation\_imd\_score,   
## data = cleveland\_sp, listw = lw\_queen)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -132.4164 -27.5521 -9.1451 10.2661 931.1520   
##   
## Type: error   
## Coefficients: (asymptotic standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 8.05012 11.07966 0.7266 0.4675  
## index\_of\_multiple\_deprivation\_imd\_score 2.88537 0.25332 11.3904 <2e-16  
##   
## Lambda: 0.41363, LR test value: 29.12, p-value: 6.8045e-08  
## Asymptotic standard error: 0.069496  
## z-value: 5.9519, p-value: 2.6503e-09  
## Wald statistic: 35.425, p-value: 2.6503e-09  
##   
## Log likelihood: -2058.057 for error model  
## ML residual variance (sigma squared): 6783.3, (sigma: 82.361)  
## Number of observations: 352   
## Number of parameters estimated: 4   
## AIC: 4124.1, (AIC for lm: 4151.2)