Assignment One: The MAUP and Multilevel Modelling

Contents

1 Demonstrating the MAUP

1.1 Background

Areal units in zoning systems amalgamate into objects that constitute the basic units for the observation and analysis of spatial phenomena (Openshaw 2015). Yet, no gold standard for guiding the spatial aggregation process exists, with the validity of zonal objects subject to the arbitrary and modifiable decision-making of quantitative geographers. Problematically, the analysis of socioeconomic data involving areal units is encumbered by the modifiable areal unit problem (MAUP): "the sensitivity of analytical results to the definition of units for which data are collected." According to the literature, the MAUP constrains the reliability of analyses for aggregated spatial data, as findings have shown varying results with the scale of aggregation and configuration of zoning systems (Avery and Clark 2015).

In practice, the MAUP is condensed into two issues of scale and zoning sensitivity which this paper will attempt to demonstrate in Section 1.2. The first issue, described as the *scale problem*, is the variation in findings when data for zonal units are progressively aggregated. This has been demonstrated empirically by Avery and Clark (2015) who found that whilst correlation coefficients did not increase monotonically with aggregation[1], a general increase in data aggregation corresponds to an increase in correlation coefficients.

The second issue, the *zoning problem*, pertains to the variation in findings when alternative combinations of zonal units are analysed with the scale or number of units held constant (Openshaw 2015). Zoning sensitivity in multivariate analysis has been demonstrated empirically in Fotheringham and Wong (2015) who simulated the aggregation of 871 block groups into 218 zones in 150 different iterations. They highlight the severity of the zoning problem by demonstrating the possibility of concluding with one iteration of zones no association between the percentage of blue-collar workers and mean family income, with another iteration finding a unit increase in blue-collar worker percentages as reducing mean family income by $20,000. In all, ignoring scale and zoning sensitivity in model calibration can lead to inferential conclusions that a researcher's areal data is applicable to the constituents who form the zones under study - the ecological fallacy problem (Openshaw 2015).

[1] At higher levels of aggregation, there is smaller adjacency of zonal units meaning groupings are more heterogenous leading to lower correlation coefficients.

1.2 Demonstrating the MAUP

1.2.1 Data

To demonstrate the scaling and zoning sensitivities of the MAUP, we calculate the bivariate strength of association between two open data variables. For our first variable, *crime_count*, we submit a HTTP request using the POST verb to send a custom polygon for retrieving all street-level crimes occuring in 2012.

```
# download geojson
u <- "http://statistics.data.gov.uk/boundaries/E08000012.json"
# store in temporary directory
downloader::download(url = u, destfile = "/tmp/lpool.geojson")
lpool <- readOGR(dsn = "/tmp/lpool.geojson", layer = "OGRGeoJSON")
# access coords slot
lpool <- lpool@polygons[[1]]@Polygons[[1]]@coords
# build lat/lon + date string to send with postrequest
curl.string <- paste0('poly=',paste0(sprintf('%s,%s',lpool[,2], lpool[,1])
                                     , collapse = ':'))

# build dates list for loop
dates = c("2012-01", "2012-02", "2012-03", "2012-04", "2012-05", "2012-06",
          "2012-07", "2012-08", "2012-09", "2012-10", "2012-11", "2012-12")

document <- lapply(dates, function(month) {
  # format acceptable packet for http request
  curl.string <- list(poly=c(curl.string), date=c(month))
  # post custom polygon to police api
  r <- httr::POST("https://data.police.uk/api/crimes-street/all-crime",
                  body = curl.string, encode="multipart", verbose())
  json <- content(r, "text", encoding = "ISO-8859-1")
  # return as data.frame
  jsonlite::fromJSON(txt=json)
})
```

```
# cast lat/lon columns to numeric data type
document$lat <- as.numeric(document$lat)
document$lon <- as.numeric(document$lon)

# convert data.frame to shapefile in rgdal
coordinates(document) <- ~lon+lat
proj4string(document) <- CRS("+init=epsg:4326") # WGS 84

# subsetting operations
document <- document[document$category == "violent-crime",]

# write once
writeOGR(document, "/Users/samcomber/Documents/spatial_analysis/shp/crimes",
         "crimes", driver = "ESRI Shapefile")

crimes <- readOGR(dsn = "/Users/samcomber/Documents/spatial_analysis/shp/crimes",
                  layer = "crimes")
crimes <- crimes[sample(nrow(crimes), 500), ]
```

Regarding our second variable, *tweet_count*, we aggregate geo-referenced tweets containing timestamps relating to Twitter postings within the municipality of Liverpool for 2012[2].

```
# read twitter shapefile
tweets <- readOGR(dsn = "/Users/samcomber/Documents/spatial_analysis/shp/tweets",
                  layer = "tweets_mersey")

# sample 500 tweets
tweets <- tweets[sample(nrow(tweets), 500), ]
```

To demonstrate the MAUP, we aggregate a count of each crime and tweet into separate variables for each region in the shapefile, before computing correlations between the two vectors of values in the dataframe. We investigate the scaling problem using an iterative process that increments the number of regions in a hex-binned lattice from 10 to 100 (stepping each iteration by 10 regions). To demonstrate the zoning problem we constrain cardinality - i.e. the number of regions - to 30, and calculate correlation coefficients using 10 different permutations of 30 regions. To build random regions, we use the multiple aggregatons feature in IMAGE studio (**???**) which uses a depth-first search algorithm for a given contiguity matrix to create aggregated spatial regions. We then dissolve regions with the `unionSpatialPolygons` using the generated `ZoneID` for aggregations.

1.2.2 Scale problem

```
# directory loop
folders <- dir("/Users/samcomber/Documents/spatial_analysis/shp/img_stud/run1/")

# we have folders + file names and are in run1 directory
setwd("/Users/samcomber/Documents/spatial_analysis/shp/img_stud/run1/")

# create master dataframe to hold correlation coefficents
cors.scale.master <- data.frame()

# loop over aggregation folder - i.e. 10 regions, 20, 30
for (folder in folders) {
  setwd(paste0("/Users/samcomber/Documents/spatial_analysis/shp/img_stud/run1","/"
               ,folder))

  # reread hex shapefile everytime we move folder
  hex <- readOGR(dsn = "/Users/samcomber/Documents/spatial_analysis/shp/hex",
                 layer = "hex_1_clip")

  out <- data.frame()
  files <- list.files(pattern = "*.zon")

  # pick first iteration from each scale of aggregation
  img <- read.csv(files[1])

  # join regions to hex shapefile on area id
  hex@data$id <- as.numeric(hex@data$id)
  hex@data <- left_join(hex@data, img, by = c("id"= "AreaID"))
  hex@data <- hex@data[order(hex@data$id),]

  # dissolve by area id
```

[2] This dataset was data mined by Guy Lansley from UCL, and processed by Dani Arribas-Bel.

```r
  tes <- unionSpatialPolygons(hex, hex@data$ZoneID)

  # convert spatial polygons to spatialpolygonsdataframe
  oldw <- getOption("warn")

  # silence irrelevant warnings
  options(warn = -1)
  df <- data.frame(id = getSpPPolygonsIDSlots(tes))
  row.names(df) <- getSpPPolygonsIDSlots(tes)
  options(warn = oldw)

  spdf <- SpatialPolygonsDataFrame(tes, data = df)

  # points in polygon spatial join
  pts.crimes <- point.in.poly(crimes, spdf)
  pts.tweets <- point.in.poly(tweets, spdf)

  # aggregate crimes/tweets data
  pts.agg.crimes <- pts.crimes@data %>% group_by(id.1) %>% summarise(num = n())
  %>% arrange(id.1)
  pts.agg.tweets <- pts.tweets@data %>% group_by(id) %>% summarise(num = n())
  %>% arrange(id)

  # cast ids as factors for left_join
  pts.agg.crimes$id.1 <- as.integer(pts.agg.crimes$id.1)
  pts.agg.tweets$id <- as.integer(pts.agg.tweets$id)

  # join summed crimes/tweets back to data.frame
  hex@data <- left_join(hex@data, pts.agg.tweets, by = c("ZoneID" = "id"))
  hex@data <- left_join(hex@data, pts.agg.crimes, by = c("ZoneID" = "id.1"))

  # correlate unique zone id sums
  uniques <- hex@data[!duplicated(hex@data$num.x), ]
  cor.tc <- cor(uniques$num.x, uniques$num.y, use="complete.obs")

  # rbind to master frame
  cors.scale.master <- rbind(cors.scale.master, cor.tc)
}
```

From Table 1, we demonstrate the scaling problem. Generally, we observe rising correlations with increasing scale. Had the MAUP not existed, we would observe little systematic variation in the correlation between violent crime and tweets - we observe a difference of 0.36 between the upper and lower coefficients. Interestingly, we observe a correlation coefficient of 0.61 at 80 regions which lends evidence to Avery and Clark (2015) findings that the low adjacency of zonal units at higher aggregations leads to lower correlation.

1.2.3 Zoning problem

```r
# we choose 20 regions to illustrate zoning problem
setwd("/Users/samcomber/Documents/spatial_analysis/shp/img_stud/run1/z070")

# get all 10 iterations of 20 regions
files.20 <- list.files(pattern = "*.zon")
```

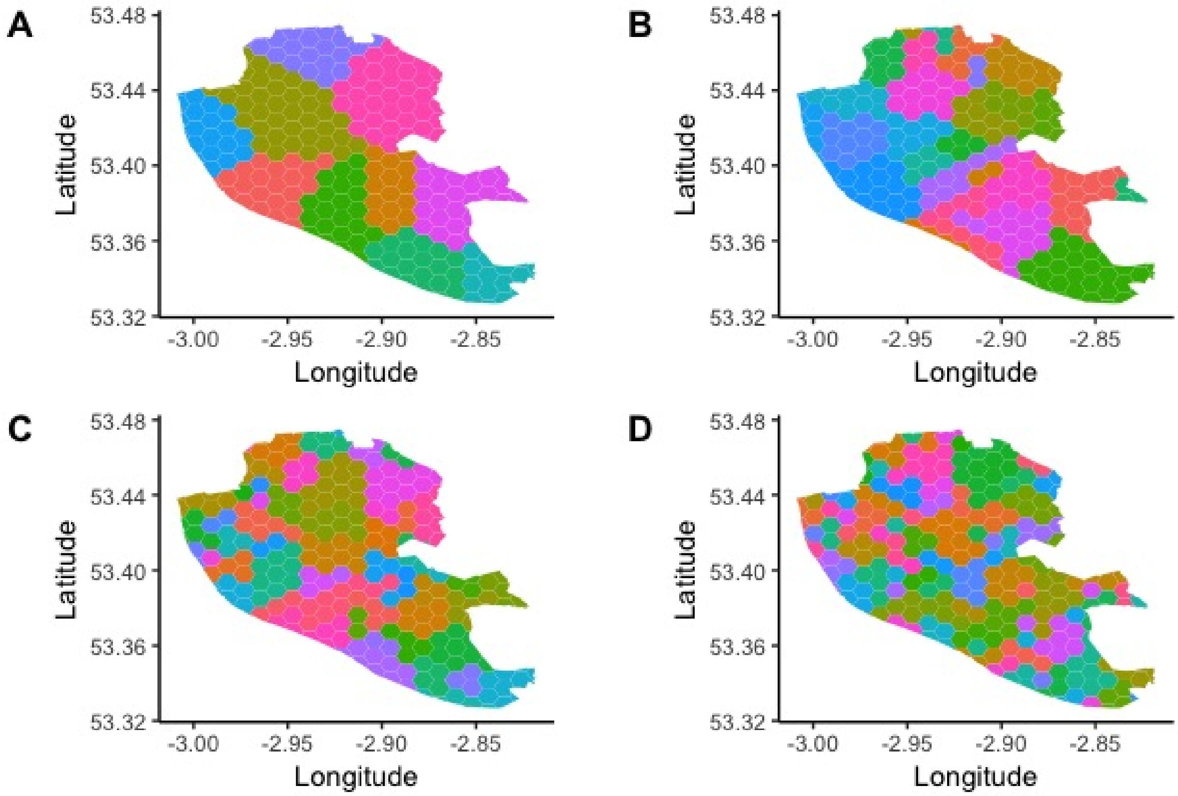


Figure 1: Scale aggregations. A shows 10 regions, B shows 30 regions, C shows 60 regions and D shows 100 regions.

Table 1: Correlations between crime and tweet counts.

Number of regions	*Correlation coeff.*
10	0.84
20	0.88
30	0.88
40	0.82
50	0.73
60	0.89
70	0.89
80	0.61
90	0.97
100	0.63

```r
cors.zone.master
# build master data.frame
cors.zone.master <- data.frame()
for (file in files.20) {
  hex <- readOGR(dsn = "/Users/samcomber/Documents/spatial_analysis/shp/hex",
                 layer = "hex_1_clip")

  # read file to df
  zon <- read.csv(file)

  # join regions to hex shapefile on area id
  hex@data$id <- as.numeric(hex@data$id)
  hex@data <- left_join(hex@data, zon, by = c("id"= "AreaID"))
  hex@data <- hex@data[order(hex@data$id),]

  # dissolve by area id
  tes <- unionSpatialPolygons(hex, hex@data$ZoneID)

  # convert spatial polygons to spatialpolygonsdataframe
  oldw <- getOption("warn")

  # silence irrelevant warnings
  options(warn = -1)
  df <- data.frame(id = getSpPPolygonsIDSlots(tes))
  row.names(df) <- getSpPPolygonsIDSlots(tes)
  options(warn = oldw)

  spdf <- SpatialPolygonsDataFrame(tes, data = df)

  # points in polygon spatial join
  pts.crimes <- point.in.poly(crimes, spdf)
  pts.tweets <- point.in.poly(tweets, spdf)

  # aggregate crimes/tweets data
  pts.agg.crimes <- pts.crimes@data %>% group_by(id.1) %>% summarise(num = n())
  %>% arrange(id.1)
  pts.agg.tweets <- pts.tweets@data %>% group_by(id) %>% summarise(num = n())
  %>% arrange(id)

  # cast ids as factors for left_join
  pts.agg.crimes$id.1 <- as.integer(pts.agg.crimes$id.1)
  pts.agg.tweets$id <- as.integer(pts.agg.tweets$id)

  # join summed crimes/tweets back to data.frame
  hex@data <- left_join(hex@data, pts.agg.tweets, by = c("ZoneID" = "id"))
  hex@data <- left_join(hex@data, pts.agg.crimes, by = c("ZoneID" = "id.1"))

  # correlate unique zone id sums
  uniques <- hex@data[!duplicated(hex@data$num.x), ]
  cor.tc <- cor(uniques$num.x, uniques$num.y, use="complete.obs")

  # rbind to master frame
  cors.zone.master <- rbind(cors.zone.master, cor.tc)
```

```
}
```

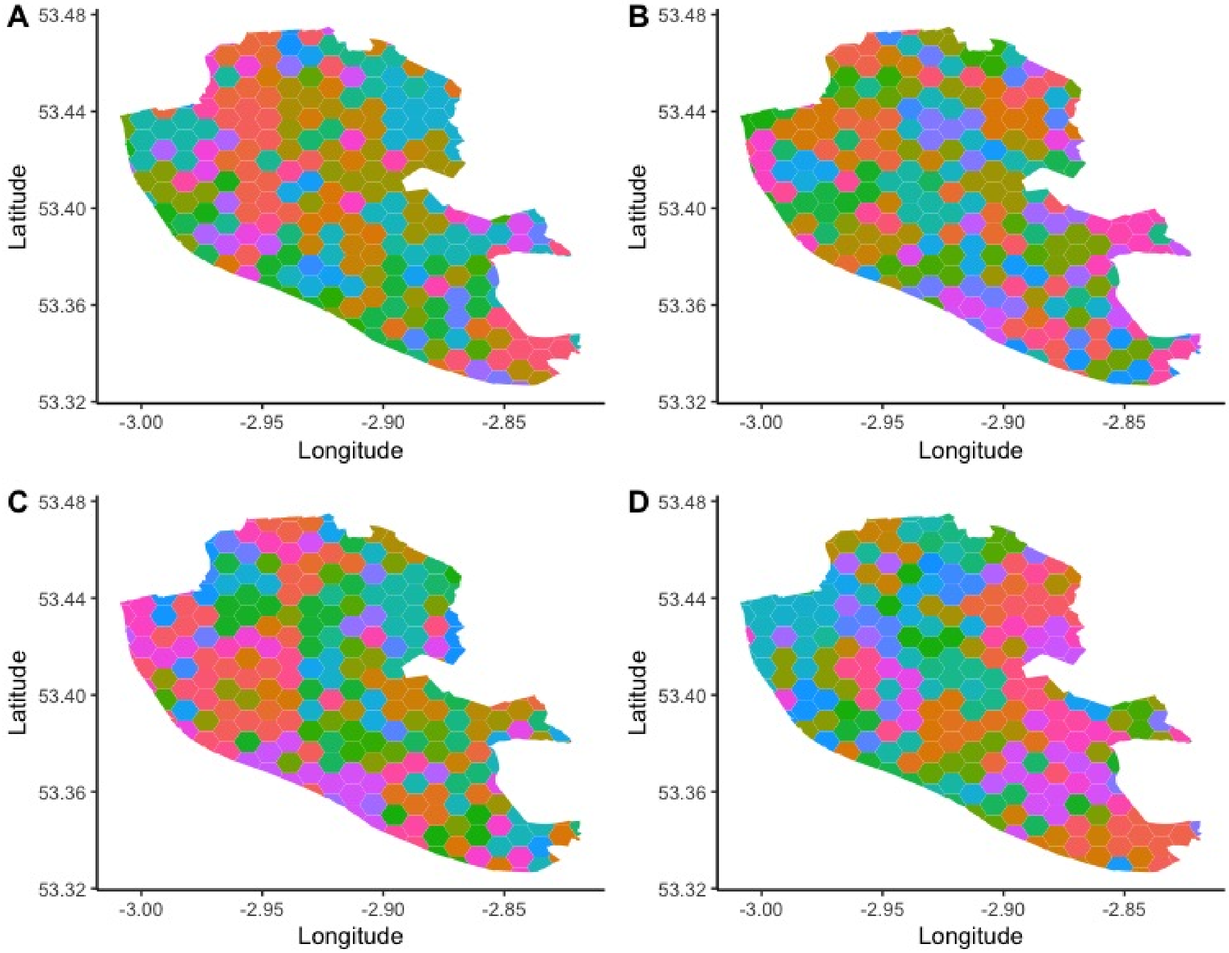


Figure 2: Zone aggregations. Four iterations displayed A-D for 70 regions.

From Table 2, we demonstrate the zoning problem: with 10 aggregation iterations for 70 regions we observe positive correlation between violent crime and tweets ranging from 0.42 (moderate assocation) to 0.96 (very strong association).

2 Multilevel Modelling

Having demonstrated the MAUP, we implement a two-level multilevel model that accounts for spatial heterogeneity effects between the scale geographies. Firstly, we aggregate the median price paid per property for each LSOA:

We then tidy the data, before log transforming median housing price per LSOA to relax the assumption of normality:

```
setwd("/Users/samcomber/documents/spatial_analysis/shp/spatial_join")
# LSOA and MSOA geographies joined by location in QGIS as
# R lacks library for polygon.in.polygon joins
```

Table 2: Correlation coefficients for 10 iterations of 70 regions.

Iteration	*Correlation coeff.*
1	0.89
2	0.66
3	0.61
4	0.89
5	0.42
6	0.96
7	0.75
8	0.92
9	0.82
10	0.84

```
lsoa.join <- readOGR(dsn = ".", layer = "lsoa_join", stringsAsFactors = FALSE)
# take only dataframe
lsoa.join <- lsoa.join@data
# cast as integer
lsoa.join$md_2015 <- as.numeric(lsoa.join$md_2015)
# log housing prices
lsoa.join$md_2015 <- log(lsoa.join$md_2015)
# take only complete cases
lsoa.join <- lsoa.join[complete.cases(lsoa.join),]
```

2.1 Null Model

Generally, our goal of estimation is the partial-pooling estimates of the median logged housing prices among all LSOAs in MSOA j. In our null model of no predictors, we approximate the multilevel estimate for MSOA j as a weighted average of the mean of properties sold in the MSOA - an unpooled estimate $\hat{y}_j$ - and the mean across all MSOAs - the completely pooled estimate, $\hat{y}_{\text{all}}$. As the weighted average uses information available about individual LSOAs, averages from MSOAs with a small number of constituent LSOAs carry less weighting, and so the multilevel estimate is pulled closer to Liverpool's overall average[3] (Gelman 2017). To allow this 'soft constraint' on α_j, partial-pooling estimates are derived by assigning a probability distribution to the α_j's which pulls estimates of α_j some way towards their mean level μ_α as below:

$$\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2) \text{ for } j = 1, ..., J,$$

where the mean μ_α and standard deviation σ_α derived from the data. To begin decomposing how median log housing prices vary by LSOA and MSOA geographies, we specify the following null multilevel model:

$$y_{ij} = \alpha_{0,j} + \epsilon_{ij}$$

$$\alpha_{0j} = \alpha_0 + \mu_{0j}$$

where in the first level, y_{ij} is the median logged housing price for the i^{th} LSOA in MSOA j, α_{0j} is the MSOA-level mean of y_{ij} (varying intercept) for the j^{th} MSOA, and ϵ_{ij} is the residual error term assumed

[3]In this way, the relative weights are determined by the sample size in the group and the variation within and between groups.

i.i.d; where in the second level, α_0 is the overall intercept, and finally μ_{0j} is the random error component for the deviation of the intercept of the j^{th} group from the overall intercept.

```
# ----------- NULL MODEL ------------

# (1|MSOA11CD) allows intercept (coefficient of 1 is the constant
# term in the regression) to vary by MSOA
model.null <- lmer(md_2015 ~ 1 + (1|MSOA11CD), data = lsoa.join)

# get variance
vc <- VarCorr(model.null)
print(vc,comp=c("Variance"))
```

```
##  Groups   Name        Variance
##  MSOA11CD (Intercept) 0.18341
##  Residual             0.12261
```

```
# calculate variance partition coefficient
print(paste0(round(0.18341 / (0.18341 + 0.12261), 3),"%"))
```

```
## [1] "0.599%"
```

Overall, we estimate mean attainment (across MSOAs) as *. The mean for the j^{th} MSOA is estimated as $30.60 + \mu_{0,j}$, where $\mu_{0,j}$ is the MSOA-level residual[4].

Having observed a Variance Partition Coefficient (VPC) of `0.599`, we approximate ~60% of the variation in median log housing prices as explained by inequalities between MSOAs, and ~40% within. This infers the presence of spatial heterogeneity between MSOAs, which justifies the use of a multilevel model to simultaneously capture outcomes at LSOA and MSOA geographies.

Following this, we estimate

```
# store MSOA-level residuals
u0 <- ranef(model.null, condVar = TRUE)
u0se <- sqrt(attr(u0[[1]], "postVar")[1, , ])
```

```
# catepillar graph to visualise mean of each MSOA random effect
msoa.rand.eff <- REsim(model.null)
p <- plotREsim(msoa.rand.eff)
```

2.2 Random-slopes model

To rationalise a varying intercept and slope model, we assess the variability between median housing prices and the percentage of neighbourhood income deprivation.

As Figure 3 demonstrates variance between median logged housing prices and percentage of neighbourhood income deprivation, we are able to motivate the use of a random slope multilevel model[5]. As observed, given random intercept models assume parallel grouping lines, they perform poorly at fitting the data. For this reason, we allow coefficients of explanatory variables to vary for each MSOA, which can be understood as interactions between group-level indicators and an individual-level predictor (Gelman 2017).

[4]For MSOA ** DO EXAMPLE HERE

[5]-2 log likelihood test also concluded **.

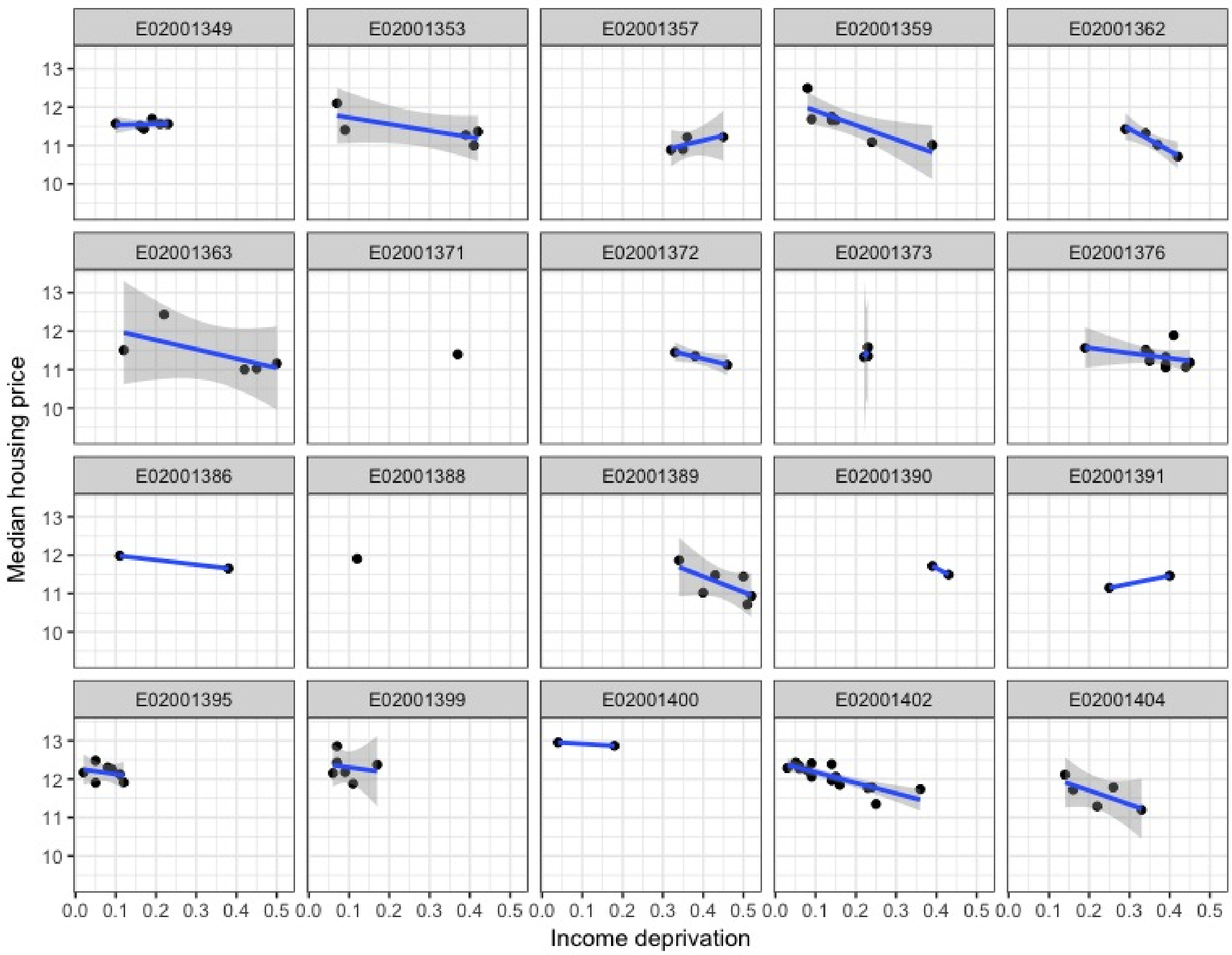


Figure 3: Variability in income deprivation by median log housing price for 20 sampled MSOAs.

This is achieved by adding a random term to the coefficient of x_k which relaxes the constraint of β_k as fixed across each MSOA, allowing the coefficient to vary randomly across groups. We specify our random-slope model as follows:

$$y_{ij} = \alpha_{ij} + \beta_1 x_{1,ij} + \beta_{2j} x_{2,ij} + \epsilon_{0,ij}$$

$$\alpha_{0j} = \alpha_0 + \mu_{0j}$$

$$\beta_{2j} = \beta + 1 + \mu_{1j}$$

which can be re-expressed into a single level by substituting formulae for α_{0j} and β_{2j} into y_{ij} as:

$$y_{ij} = \alpha_0 + \beta_1 x_{ij} + \mu_{0j} + \mu_{2j} x_{ij} + \epsilon_{ij}$$

Conceptually, this equation is similar to the null model, but the $\mu_{2j} x_{ij}$ term has been added to reflect the interaction between the j^{th} MSOA and predictor x. The random effects μ_{0j} and μ_{2j} are assumed:

$$\begin{bmatrix} \mu_{0j} \\ \mu_{2j} \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} \sigma^2_{\mu 0} & \\ \sigma_{\mu 0 \mu 2} & \sigma^2_{\mu 2} \end{bmatrix})$$

meaning the slope of regression line for MSOA j is $\beta_2 + \mu_{ij}$, which allows the coefficient to vary by group. We define our random-slopes model in R as follows:

```
model.var.slope <- lmer(md_2015 ~ lr_crime
                        + (1 + lr_income | MSOA11CD), data=lsoa.join, REML = FALSE)

model.var.slope.fit <- lmer(md_2015 ~ lr_crime
                        + (1 | MSOA11CD), data=lsoa.join,  REML = FALSE)

anova(model.var.slope, model.var.slope.fit)
```

```
## Data: lsoa.join
## Models:
## model.var.slope.fit: md_2015 ~ lr_crime + (1 | MSOA11CD)
## model.var.slope: md_2015 ~ lr_crime + (1 + lr_income | MSOA11CD)
##                     Df    AIC    BIC   logLik deviance Chisq Chi Df
## model.var.slope.fit  4 267.23 281.68 -129.616   259.23
## model.var.slope      6 203.76 225.44  -95.881   191.76 67.47      2
##                     Pr(>Chisq)
## model.var.slope.fit
## model.var.slope      2.233e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Firstly, we use a likelihood ratio test derived from an ANOVA table to estimate whether the income deprivation effect varies across MSOAs. As the deduction of the log-likelihood values computes a likelihood ratio value of `67.47` on 2 degrees of freedom, we can infer the income deprivation effect varies across MSOAs, justifying the use of a random-slopes model.

```r
summary(model.var.slope)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: md_2015 ~ lr_crime + (1 + lr_income | MSOA11CD)
##    Data: lsoa.join
##
##      AIC      BIC   logLik deviance df.resid
##    203.8    225.4    -95.9    191.8      268
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5302 -0.6339 -0.0657  0.5694  3.2254
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  MSOA11CD (Intercept) 0.85051  0.9222
##           lr_income   5.99199  2.4479   -0.98
##  Residual             0.07373  0.2715
## Number of obs: 274, groups:  MSOA11CD, 50
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 11.35562    0.04852  234.04
## lr_crime    -0.11878    0.03943   -3.01
##
## Correlation of Fixed Effects:
##          (Intr)
## lr_crime -0.636
```

To begin interpreting the intercept and slope residuals, we graphically project the plot of income deprivation slopes by intercept ($\hat{\mu}_{1j}$ by $\hat{\mu}_{0j}$) in Figure 4. Intuitively, this plot conforms with conventional wisdom: MSOAs with higher-than-average median log housing prices also have below-average slopes for income deprivation as shown in the bottom-right quadrant. Interestingly, in the bottom-left quadrant, we observe an MSOA *.

```r
# data object containing random slope and intercepts
inc.random <- ranef(model.var.slope, condVar = TRUE)

plot(inc.random[[1]], xlab = "Intercept (u0j)", ylab = "Slope of lr_income (u1j)")
abline(h = 0, col = "red")
abline(v = 0, col = "red")
```

To produce the equation for the fitted regression line of MSOA j, one can solve: $\hat{y}_{ij} = (11.36 + \hat{\mu}_{0j}) + (-0.12 + \hat{\mu_{2j}})\text{lr\_income}_{ij}$ where `11.36` and `-0.12` are derived from * and the values of $\hat{\mu}_{0j}$ and $\hat{\mu}_{2j}$ can be derived from the pairwise residual plot in Figure 4.

```r
# create caterpillar graph
msoa.rand <- REsim(model.var.slope)
plotREsim(msoa.rand)
```

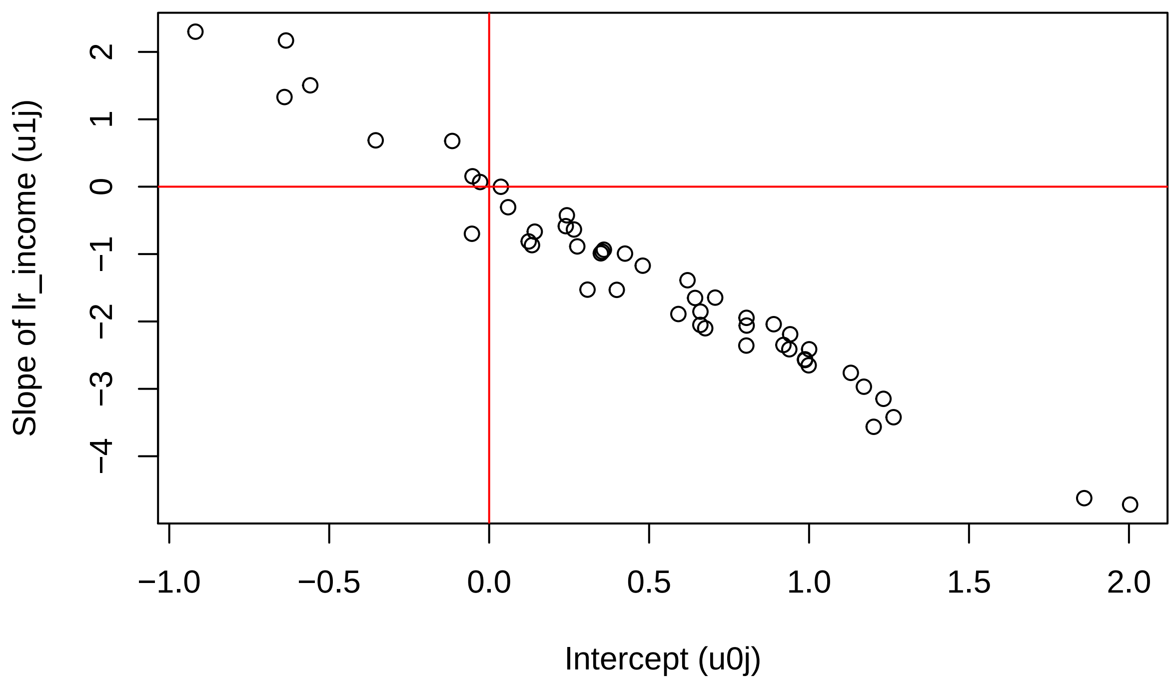
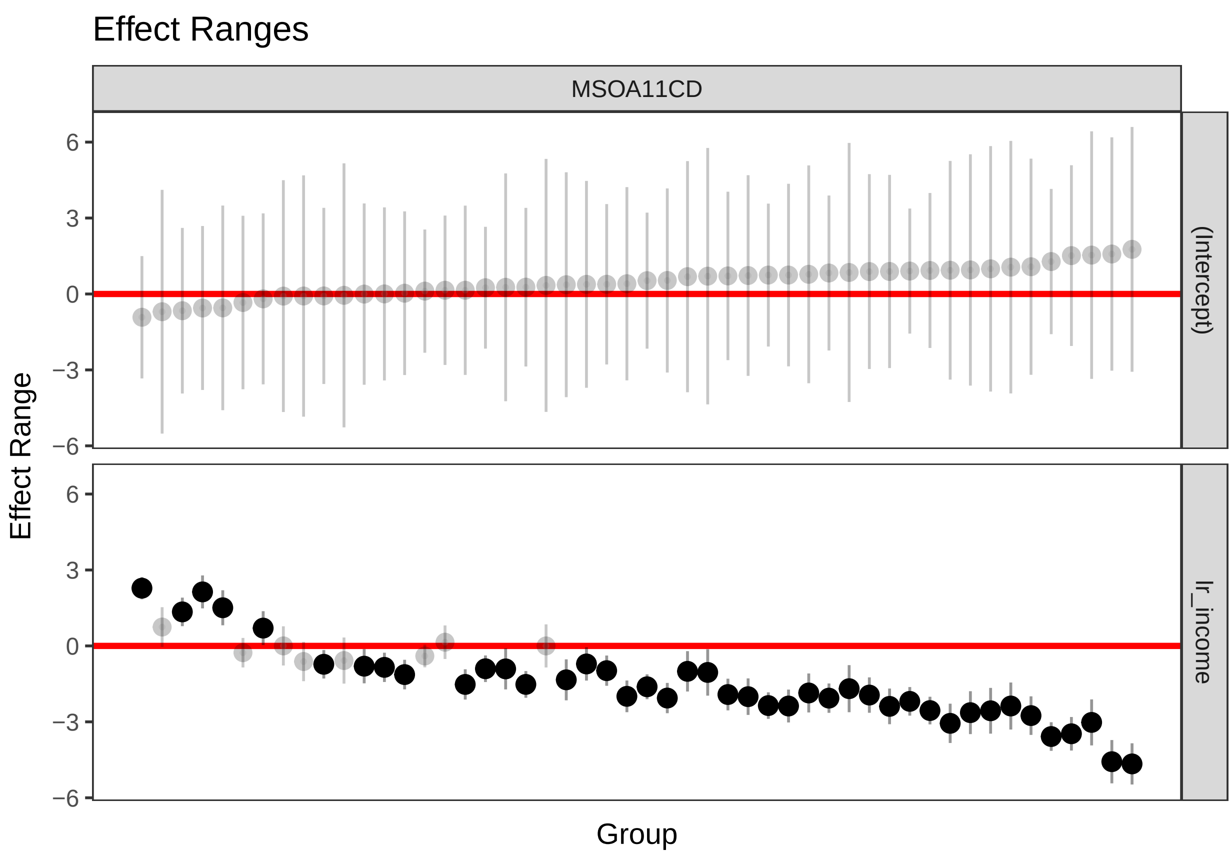


Figure 4: Income deprivation slopes versus intercept.



2.3 Space and conclusions

With our interest in space, we conclude this paper by displaying the spatial distribution of random slopes in Figure 5 using `spplot` for the presentation layer.

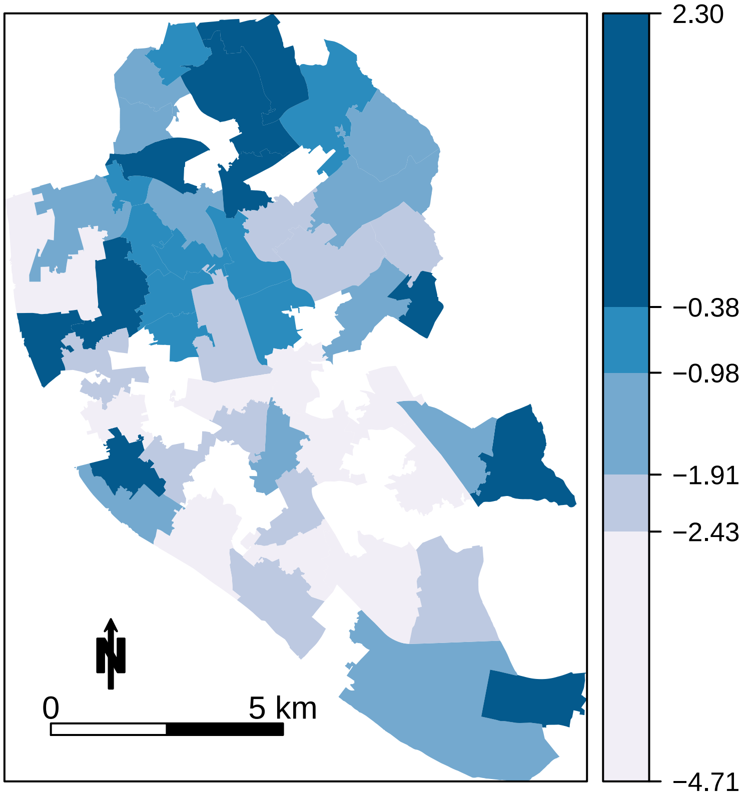


Figure 5: Spatial patterning of varying slopes at MSOA scale.

3 Bibliography

- t
- t

4 Appendix

Avery and Clark. 2015. "R: A Language and Environment for Statistical Computing." Journal Article. http://www.R-project.org.

Fotheringham and Wong. 2015. "R: A Language and Environment for Statistical Computing." Journal Article. http://www.R-project.org.

Gelman. 2017. "R: A Language and Environment for Statistical Computing." Journal Article. http://www.R-project.org.

Openshaw. 2015. "R: A Language and Environment for Statistical Computing." Journal Article. http://www.R-project.org.