Spatial Data Analysis

Tobias Rüttenauer

2024-04-04

Table of contents

Introduction

This course materials are designed for a 1-day workshop on spatial data analysis. Rüttenauer (2024) provides a handbook chapter accompanying these workshop materials.

In recent years, more and more spatial data has become available, providing the possibility to combine otherwise unrelated data, such as social, economic, and environmental data. This also opens up the possibility of analysing spatial patterns and processes (e.g., spillover effects or diffusion).

Many social science research questions are spatially dependent such as voting outcomes, housing prices, labour markets, protest behaviour, or migration decisions. Observing an event in one region or neighbourhood increases the likelihood that we observe similar processes in proximate areas. As Tobler's first law of geography puts it: "Everything is related to everything else, but near things are more related than distant things". This dependence can stem from spatial contagion, spatial spillovers, or common confounders. Therefore, basic assumptions of standard regression models are violated when analysing spatial data. However, more importantly, spatial processes are interesting for their own sake. Spatial regression models can detect spatial dependence and explicitly model spatial relations, identifying spatial clustering, spillovers or diffusion processes.

The main objective of the course is the theoretical understanding and practical application of spatial regression models. This course will first give an overview on how to perform common spatial operations using spatial information, such as aggregating spatial units, calculating distances, merging spatial data as well as visualizing them. The course will further focus on the analysis of geographic data and the application of spatial regression techniques to model and analyze spatial processes, and furthermore, the course addresses several methods for defining spatial relationships, detecting and diagnosing spatial dependence and autocorrelation. Finally, we will discuss various spatial regression techniques to model processes, clarify the assumptions of these models, and show how they differ in their applications and interpretations.

The field has developed very quickly over the past few years, and *R* now provides a rich set of packages for various spatial data operations. For a more in-depth introduction into spatial data analysis in *R*, have a look into the materials references below.

The material introduces the use of geographical information to connect and analyze different spatial data sources very briefly. This introduction is limited to the fundamentals of using geographical information in *R*. Stefan Jünger & Anne-Kathrin Stroppe have provided a

comprehensive GESIS workshop on geospatial techniques in R. The focus of this workshop will be on techniques for spatial data analysis, such as spatial regression models.

Schedule

Time	Session
09:00 – 09:30	Refresher on R for spatial data
09:30 – 11:00	Spatial Relationships (W) and Spatial Dependence
11:15 – 12:00	Practical exercise
Lunch break	
13:00 – 14:30	Spatial Regression Models (SLX, Error, lagged DV)
14:45 – 16:00	Interpreting Results: Spatial Impacts
16:15 – 18:00	Practical exercise

Some useful packages

By now, *R* provides a lot of functionalities for GIS applications and spatial econometrics, and further extensions. There are lots of packages providing a huge variety of spatial functionalities and methods (see e.g. R. Bivand, Millo, and Piras 2021). Important packages for fundamental spatial operations are:

- Spatial data workhorses: sf (Pebesma 2018) and terra
- Visualization: mapview (Appelhans et al. 2021) and tmap (Tennekes 2018)
- Spatial weights and other relations: spdep (R. S. Bivand and Rudel 2018)
- Spatial interpolation and kriging: gstat (Gräler, Pebesma, and Heuvelink 2016)
- Spatial regression models: spatialreg and sphet (R. Bivand and Piras 2015)
- The packages have constantly developed over the past years, and older packages such as rgdal, rgeos, and sp are currently retiring (Blog post)

Further Readings

- Great up-to-date introduction to spatial R: Lovelace, Nowosad, and Muenchow (2019), updated version available online
- Great open-science book on Spatial Data Science Pebesma and Bivand (2023)

- Comprehensive introduction to spatial econometrics: LeSage and Pace (2009)
- Relative intuitive introduction to spatial econometrics: Ward and Gleditsch (2008)
- Article-length introductions to spatial econometrics: Elhorst (2012), Halleck Vega and Elhorst (2015), LeSage (2014), Rüttenauer (2024), and Rüttenauer (2022)

Course materials

- I highly recommend the great Introduction to Geospatial Techniques for Social Scientists in R including, see Stefan Jünger & Anne-Kathrin Stroppe's GESIS workshop materials. Nice materials on GIS, spatial operations and spatial data visualisation!
- For those looking for a more in-depth introduction, I highly recommend Roger Bivand's course on Spatial Data Analysis: Youtube recordings, Course Materials
- I've learned most of what I know about spatial econometrics from Scott J. Cook and his workshop on Spatial Econometrics at the Essex Summer school.

1 Refresher

Required packages

```
pkgs <- c("sf", "gstat", "mapview", "nngeo", "rnaturalearth", "dplyr",
          "nomisr", "osmdata", "tidyr", "texreg")
lapply(pkgs, require, character.only = TRUE)
```

Session info

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default


locale:
[1] LC_COLLATE=English_United Kingdom.utf8
[2] LC_CTYPE=English_United Kingdom.utf8
[3] LC_MONETARY=English_United Kingdom.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.utf8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] texreg_1.38.6       tidyr_1.3.0         osmdata_0.2.3
```

```
 [4] nomisr_0.4.7        dplyr_1.1.2         rnaturalearth_0.3.3
 [7] nngeo_0.4.7         mapview_2.11.0      gstat_2.1-1
[10] sf_1.0-13

loaded via a namespace (and not attached):
 [1] xfun_0.39           raster_3.6-23       htmlwidgets_1.6.2   lattice_0.21-8
 [5] vctrs_0.6.3         tools_4.3.1         crosstalk_1.2.0     generics_0.1.3
 [9] stats4_4.3.1        tibble_3.2.1        proxy_0.4-27        spacetime_1.3-0
[13] fansi_1.0.4         xts_0.13.1          pkgconfig_2.0.3     KernSmooth_2.23-
21
[17] satellite_1.0.4     data.table_1.14.8   leaflet_2.1.2       webshot_0.5.5
[21] lifecycle_1.0.3     compiler_4.3.1      FNN_1.1.3.2         rsdmx_0.6-2
[25] munsell_0.5.0       terra_1.7-39        codetools_0.2-19    snakecase_0.11.0
[29] htmltools_0.5.5     class_7.3-22        pillar_1.9.0        classInt_0.4-9
[33] tidyselect_1.2.0    digest_0.6.32       purrr_1.0.1         fastmap_1.1.1
[37] grid_4.3.1          colorspace_2.1-0    cli_3.6.1           magrittr_2.0.3
[41] base64enc_0.1-3     XML_3.99-0.14       utf8_1.2.3          leafem_0.2.0
[45] e1071_1.7-13        scales_1.2.1        sp_1.6-1            rmarkdown_2.23
[49] httr_1.4.6          zoo_1.8-12          png_0.1-8           evaluate_0.21
[53] knitr_1.43          rlang_1.1.1         Rcpp_1.0.10         glue_1.6.2
[57] DBI_1.1.3           rstudioapi_0.14     jsonlite_1.8.5      R6_2.5.1
[61] plyr_1.8.8          intervals_0.15.4    units_0.8-2
```

1.1 Packages

Please make sure that you have installed the following packages:

```
pks <- c("dplyr",
"gstat",
"mapview",
"nngeo",
"nomisr",
"osmdata",
"rnaturalearth",
"sf",
"spatialreg",
"spdep",
"texreg",
"tidyr",
"tmap",
"viridisLite")
```

The most important package is sf: Simple Features for R. users are strongly encouraged to install the sf binary packages from CRAN. If that does not work, please have a look at the installation instructions. It requires software packages GEOS, GDAL and PROJ.

1.2 Coordinates

In general, spatial data is structured like conventional/tidy data (e.g. data.frames, matrices), but has one additional dimension: every observation is linked to some sort of geo-spatial information. Most common types of spatial information are:

- Points (one coordinate pair)
- Lines (two coordinate pairs)
- Polygons (at least three coordinate pairs)
- Regular grids (one coordinate pair for centroid + raster / grid size)

1.2.1 Coordinate reference system (CRS)

In its raw form, a pair of coordinates consists of two numerical values. For instance, the pair `c(51.752595, -1.262801)` describes the location of Nuffield College in Oxford (one point). The fist number represents the latitude (north-south direction), the second number is the longitude (west-east direction), both are in decimal degrees.

However, we need to specify a reference point for latitudes and longitudes (in the Figure above: equator and Greenwich). For instance, the pair of coordinates above comes from Google Maps which returns GPS coordinates in 'WGS 84' (EPSG:4326).

```
# Coordinate pairs of two locations
coords1 <- c(51.752595, -1.262801)
coords2 <- c(51.753237, -1.253904)
coords <- rbind(coords1, coords2)

# Conventional data frame
nuffield.df <- data.frame(name = c("Nuffield College", "Radcliffe Camera"),
                          address = c("New Road", "Radcliffe Sq"),
                          lat = coords[,1], lon = coords[,2])

head(nuffield.df)
```

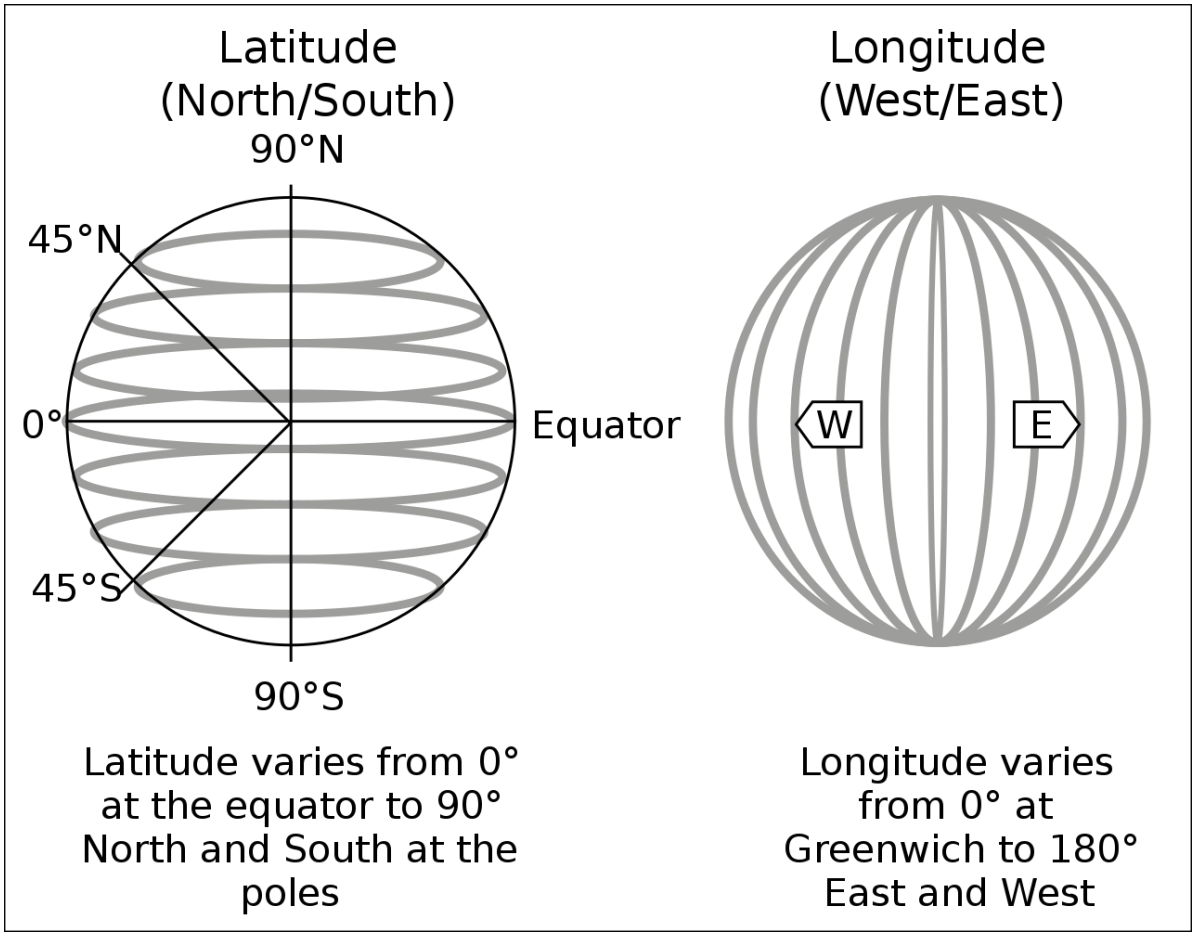
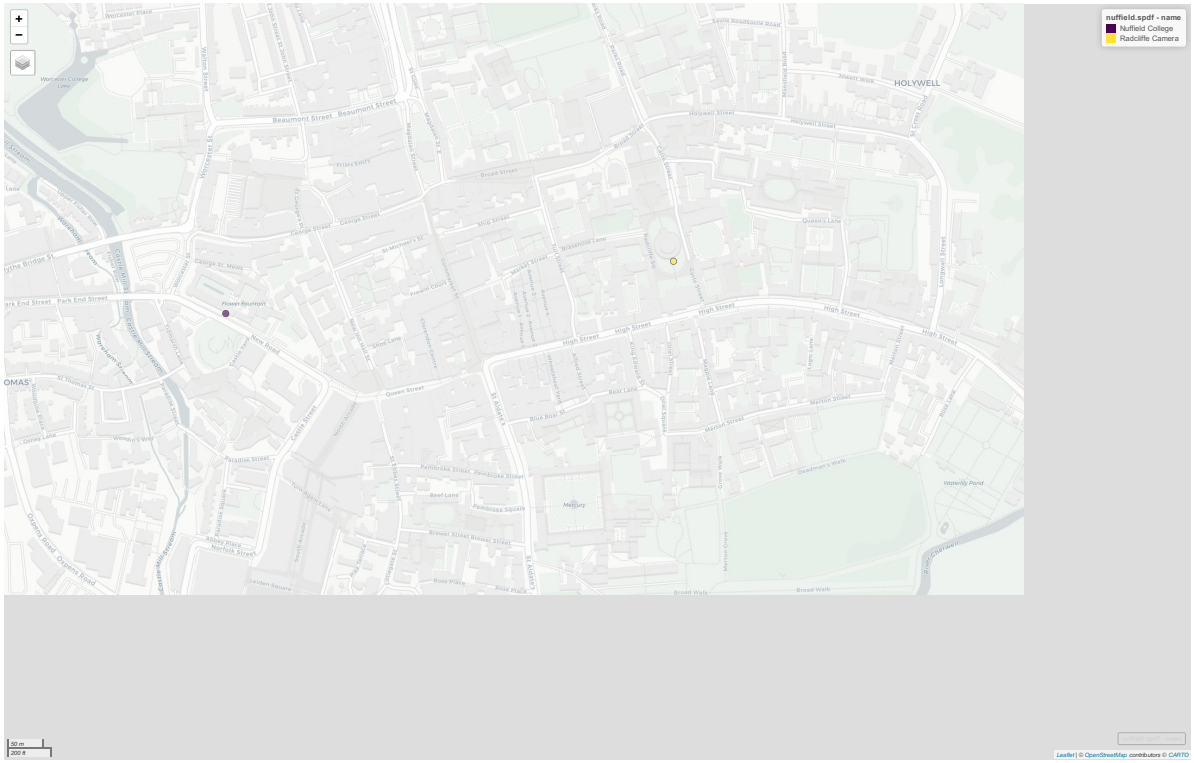


Figure 1.1: Figure: Latitude and longitude, Source: Wikipedia

```
                    name      address      lat       lon
coords1 Nuffield College     New Road 51.75259 -1.262801
coords2 Radcliffe Camera Radcliffe Sq 51.75324 -1.253904
```

```r
# Combine to spatial data frame
nuffield.spdf <- st_as_sf(nuffield.df,
                          coords = c("lon", "lat"), # Order is important
                          crs = 4326) # EPSG number of CRS

# Map
mapview(nuffield.spdf, zcol = "name")
```



1.2.2 Projected CRS

However, different data providers use different CRS. For instance, spatial data in the UK usually uses 'OSGB 1936 / British National Grid' (EPSG:27700). Here, coordinates are in meters, and projected onto a planar 2D space.

There are a lot of different CRS projections, and different national statistics offices provide data in different projections. Data providers usually specify which reference system they use.

This is important as using the correct reference system and projection is crucial for plotting and manipulating spatial data.

If you do not know the correct CRS, try starting with a standards CRS like EPSG:4326 if you have decimal degree like coordinates. If it looks like projected coordinates, try searching for the country or region in CRS libraries like https://epsg.io/. However, you must check if the projected coordinates match their real location, e.g. using `mapview()`.

1.2.3 Why different projections?

By now, (most) people agree that the earth is not flat. So, to plot data on a 2D planar surface and to perform certain operations on a planar world, we need to make some re-projections. Depending on where we are, different re-projections of our data (globe in this case) might work better than others.

```
world <- ne_countries(scale = "medium", returnclass = "sf")
class(world)
```

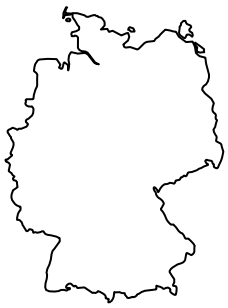
```
[1] "sf"         "data.frame"
```

```
st_crs(world)
```

```
Coordinate Reference System:
  User input: +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
  wkt:
BOUNDCRS[
    SOURCECRS[
        GEOGCRS["unknown",
            DATUM["World Geodetic System 1984",
                ELLIPSOID["WGS 84",6378137,298.257223563,
                    LENGTHUNIT["metre",1]],
                ID["EPSG",6326]],
            PRIMEM["Greenwich",0,
                ANGLEUNIT["degree",0.0174532925199433],
                ID["EPSG",8901]],
            CS[ellipsoidal,2],
                AXIS["longitude",east,
                    ORDER[1],
                    ANGLEUNIT["degree",0.0174532925199433,
                        ID["EPSG",9122]]],
```

```
            AXIS["latitude",north,
                ORDER[2],
                ANGLEUNIT["degree",0.0174532925199433,
                    ID["EPSG",9122]]]],
TARGETCRS[
    GEOGCRS["WGS 84",
        DATUM["World Geodetic System 1984",
            ELLIPSOID["WGS 84",6378137,298.257223563,
                LENGTHUNIT["metre",1]]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        CS[ellipsoidal,2],
            AXIS["latitude",north,
                ORDER[1],
                ANGLEUNIT["degree",0.0174532925199433]],
            AXIS["longitude",east,
                ORDER[2],
                ANGLEUNIT["degree",0.0174532925199433]],
        ID["EPSG",4326]]],
ABRIDGEDTRANSFORMATION["Transformation from unknown to WGS84",
    METHOD["Geocentric translations (geog2D domain)",
        ID["EPSG",9603]],
    PARAMETER["X-axis translation",0,
        ID["EPSG",8605]],
    PARAMETER["Y-axis translation",0,
        ID["EPSG",8606]],
    PARAMETER["Z-axis translation",0,
        ID["EPSG",8607]]]]
```

```r
# Extract a country and plot in current CRS (WGS84)
ger.spdf <- world[world$name == "Germany", ]
plot(st_geometry(ger.spdf))
```



```r
# Now, let's transform Germany into a CRS optimized for Iceland
ger_rep.spdf <- st_transform(ger.spdf, crs = 5325)
plot(st_geometry(ger_rep.spdf))
```

Depending on the angle, a 2D projection of the earth looks different. It is important to choose a suitable projection for the available spatial data. For more information on CRS and re-projection, see e.g. Lovelace, Nowosad, and Muenchow (2019) or Stefan Jünger & Anne-Kathrin Stroppe's GESIS workshop materials.

1.3 Importing some real world data

`sf` imports many of the most common spatial data files, like geojson, gpkg, or shp.

1.3.1 London shapefile (polygon)

Let's get some administrative boundaries for London from the London Datastore. We use the `sf` package and its funtion `st_read()` to import the data.

```
# Create subdir (all data withh be stored in "_data")
dn <- "_data"
ifelse(dir.exists(dn), "Exists", dir.create(dn))
```

```
[1] "Exists"
```

```r
# Download zip file and unzip
tmpf <- tempfile()
boundary.link <- "https://data.london.gov.uk/download/statistical-gis-boundary-files-londo
download.file(boundary.link, tmpf)
unzip(zipfile = tmpf, exdir = paste0(dn))
unlink(tmpf)

# This is a shapefile
# We only need the MSOA layer for now
msoa.spdf <- st_read(dsn = paste0(dn, "/statistical-gis-boundaries-london/ESRI"),
                     layer = "MSOA_2011_London_gen_MHW" # Note: no file ending
                     )
```

```
Reading layer `MSOA_2011_London_gen_MHW' from data source
  `C:\work\Lehre\Geodata_Spatial_Regression_short\_data\statistical-gis-
boundaries-london\ESRI'
  using driver `ESRI Shapefile'
Simple feature collection with 983 features and 12 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 503574.2 ymin: 155850.8 xmax: 561956.7 ymax: 200933.6
Projected CRS: OSGB36 / British National Grid
```

The object `msoa.spdf` is our spatial data.frame. It looks essentially like a conventional data.frame, but has some additional attributes and geo-graphical information stored with it. Most importantly, notice the column `geometry`, which contains a list of polygons. In most cases, we have one polygon for each line / observation.

```r
head(msoa.spdf)
```

```
Simple feature collection with 6 features and 12 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 530966.7 ymin: 180510.7 xmax: 551943.8 ymax: 191139
Projected CRS: OSGB36 / British National Grid
   MSOA11CD                 MSOA11NM   LAD11CD              LAD11NM   RGN11CD
1 E02000001       City of London 001 E09000001       City of London E12000007
2 E02000002 Barking and Dagenham 001 E09000002 Barking and Dagenham E12000007
3 E02000003 Barking and Dagenham 002 E09000002 Barking and Dagenham E12000007
4 E02000004 Barking and Dagenham 003 E09000002 Barking and Dagenham E12000007
5 E02000005 Barking and Dagenham 004 E09000002 Barking and Dagenham E12000007
```

```
6 E02000007 Barking and Dagenham 006 E09000002 Barking and Dagenham E12000007
  RGN11NM USUALRES HHOLDRES COMESTRES POPDEN HHOLDS AVHHOLDSZ
1  London     7375     7187       188   25.5   4385       1.6
2  London     6775     6724        51   31.3   2713       2.5
3  London    10045    10033        12   46.9   3834       2.6
4  London     6182     5937       245   24.8   2318       2.6
5  London     8562     8562         0   72.1   3183       2.7
6  London     8791     8672       119   50.6   3441       2.5
                        geometry
1 MULTIPOLYGON (((531667.6 18...
2 MULTIPOLYGON (((548881.6 19...
3 MULTIPOLYGON (((549102.4 18...
4 MULTIPOLYGON (((551550 1873...
5 MULTIPOLYGON (((549099.6 18...
6 MULTIPOLYGON (((549819.9 18...
```

Shapefiles are still among the most common formats to store and transmit spatial data, despite them being inefficient (file size and file number).

However, `sf` reads everything spatial, such as `geo.json`, which usually is more efficient, but less common (but we're getting there).

```
# Download file
ulez.link <- "https://data.london.gov.uk/download/ultra_low_emissions_zone/936d71d8-c5fc-4
download.file(ulez.link, paste0(dn, "/ulez.json"))

# Read geo.json
st_layers(paste0(dn, "/ulez.json"))
```

```
Driver: GeoJSON
Available layers:
                  layer_name geometry_type features fields
1 CentralUltraLowEmissionZone Multi Polygon        1      4
                        crs_name
1 OSGB36 / British National Grid
```

```
ulez.spdf <- st_read(dsn = paste0(dn, "/ulez.json")) # here dsn is simply the file
```

```
Reading layer `CentralUltraLowEmissionZone' from data source
  `C:\work\Lehre\Geodata_Spatial_Regression_short\_data\ulez.json'
  using driver `GeoJSON'
```

```
Simple feature collection with 1 feature and 4 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 527271.5 ymin: 178041.5 xmax: 533866.3 ymax: 183133.4
Projected CRS: OSGB36 / British National Grid
```

```
head(ulez.spdf)
```

```
Simple feature collection with 1 feature and 4 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 527271.5 ymin: 178041.5 xmax: 533866.3 ymax: 183133.4
Projected CRS: OSGB36 / British National Grid
  fid OBJECTID BOUNDARY Shape_Area                        geometry
1   1        1 CSS Area   21.37557 MULTIPOLYGON (((531562.7 18...
```

Again, this looks like a conventional `data.frame` but has the additional column `geometry` containing the coordinates of each observation. `st_geometry()` returns only the geographic object and `st_drop_geometry()` only the `data.frame` without the coordinates. We can plot the object using `mapview()`.

```
mapview(msoa.spdf[, "POPDEN"])
```

1.3.2 Census API (admin units)

Now that we have some boundaries and shapes of spatial units in London, we can start looking for different data sources to populate the geometries.

A good source for demographic data is for instance the 2011 census. Below we use the nomis API to retrieve population data for London, See the Vignette for more information (Guest users are limited to 25,000 rows per query). Below is a wrapper to avoid some errors with sex and urban-rural cross-tabulation in some of the data.

```
### For larger request, register and set key
# Sys.setenv(NOMIS_API_KEY = "XXX")
# nomis_api_key(check_env = TRUE)

x <- nomis_data_info()

# Get London ids
london_ids <- msoa.spdf$MSOA11CD

### Get key statistics ids
```

```r
# select requires tables (https://www.nomisweb.co.uk/sources/census_2011_ks)
# Let's get KS201EW (ethnic group), KS205EW (passport held), and KS402EW (housing tenure)

# Get internal ids
stats <- c("KS201EW", "KS402EW", "KS205EW")
oo <- which(grepl(paste(stats, collapse = "|"), x$name.value))
ksids <- x$id[oo]
ksids # This are the internal ids
```

```
[1] "NM_608_1" "NM_612_1" "NM_619_1"
```

```r
### look at meta information
q <- nomis_overview(ksids[1])
head(q)
```

```
# A tibble: 6 x 2
  name           value
  <chr>          <list>
1 analyses       <named list [1]>
2 analysisname   <chr [1]>
3 analysisnumber <int [1]>
4 contact        <named list [4]>
5 contenttypes   <named list [1]>
6 coverage       <chr [1]>
```

```r
a <- nomis_get_metadata(id = ksids[1], concept = "GEOGRAPHY", type = "type")
a # TYPE297 is MSOA level
```

```
# A tibble: 24 x 3
   id      label.en                                              description.en
   <chr>   <chr>                                                 <chr>
 1 TYPE265 NHS area teams                                        NHS area teams
 2 TYPE266 clinical commissioning groups                         clinical commissi~
 3 TYPE267 built-up areas including subdivisions                 built-up areas in~
 4 TYPE269 built-up areas                                        built-up areas
 5 TYPE273 national assembly for wales electoral regions 2010    national assembly~
 6 TYPE274 postcode areas                                        postcode areas
 7 TYPE275 postcode districts                                    postcode districts
 8 TYPE276 postcode sectors                                      postcode sectors
```

```
 9 TYPE277 national assembly for wales constituencies 2010    national assembly~
10 TYPE279 parishes 2011                                      parishes 2011
# i 14 more rows
```

```
b <- nomis_get_metadata(id = ksids[1], concept = "MEASURES", type = "TYPE297")
b # 20100 is the measure of absolute numbers
```

```
# A tibble: 2 x 3
  id    label.en description.en
  <chr> <chr>    <chr>
1 20100 value    value
2 20301 percent  percent
```

```
### Query data in loop over the required statistics
for(i in ksids){

  # Determin if data is divided by sex or urban-rural
  nd <- nomis_get_metadata(id = i)
  if("RURAL_URBAN" %in% nd$conceptref){
    UR <- TRUE
  }else{
    UR <- FALSE
  }
  if("C_SEX" %in% nd$conceptref){
    SEX <- TRUE
  }else{
    SEX <- FALSE
  }

  # make data request
  if(UR == TRUE){
    if(SEX == TRUE){
      tmp_en <- nomis_get_data(id = i, time = "2011",
                               geography = london_ids, # replace with "TYPE297" for all MS
                               measures = 20100, RURAL_URBAN = 0, C_SEX = 0)
    }else{
      tmp_en <- nomis_get_data(id = i, time = "2011",
                               geography = london_ids, # replace with "TYPE297" for all MS
                               measures = 20100, RURAL_URBAN = 0)
    }
  }else{
```

```
  if(SEX == TRUE){
    tmp_en <- nomis_get_data(id = i, time = "2011",
                             geography = london_ids, # replace with "TYPE297" for all MS
                             measures = 20100, C_SEX = 0)
  }else{
    tmp_en <- nomis_get_data(id = i, time = "2011",
                             geography = london_ids, # replace with "TYPE297" for all MS
                             measures = 20100)
  }

}

# Append (in case of different regions)
ks_tmp <- tmp_en

# Make lower case names
names(ks_tmp) <- tolower(names(ks_tmp))
names(ks_tmp)[names(ks_tmp) == "geography_code"] <- "msoa11"
names(ks_tmp)[names(ks_tmp) == "geography_name"] <- "name"

# replace weird cell codes
onlynum <- which(grepl("^[[:digit:]]+$", ks_tmp$cell_code))
if(length(onlynum) != 0){
  code <- substr(ks_tmp$cell_code[-onlynum][1], 1, 7)
  if(is.na(code)){
    code <- i
  }
  ks_tmp$cell_code[onlynum] <- paste0(code, "_", ks_tmp$cell_code[onlynum])
}

# save codebook
ks_cb <- unique(ks_tmp[, c("date", "cell_type", "cell", "cell_code", "cell_name")])

### Reshape
ks_res <- tidyr::pivot_wider(ks_tmp, id_cols = c("msoa11", "name"),
                             names_from = "cell_code",
                             values_from = "obs_value")

### Merge
if(i == ksids[1]){
  census_keystat.df <- ks_res
```

```
    census_keystat_cb.df <- ks_cb
  }else{
    census_keystat.df <- merge(census_keystat.df, ks_res, by = c("msoa11", "name"), all =
    census_keystat_cb.df <- rbind(census_keystat_cb.df, ks_cb)
  }

}


# Descriptions are saved in the codebook
head(census_keystat_cb.df)
```

```
# A tibble: 6 x 5
   date cell_type     cell cell_code   cell_name
  <dbl> <chr>        <dbl> <chr>       <chr>
1  2011 Ethnic Group     0 KS201EW0001 All usual residents
2  2011 Ethnic Group   100 KS201EW_100 White
3  2011 Ethnic Group     1 KS201EW0002 White: English/Welsh/Scottish/Northern I~
4  2011 Ethnic Group     2 KS201EW0003 White: Irish
5  2011 Ethnic Group     3 KS201EW0004 White: Gypsy or Irish Traveller
6  2011 Ethnic Group     4 KS201EW0005 White: Other White
```

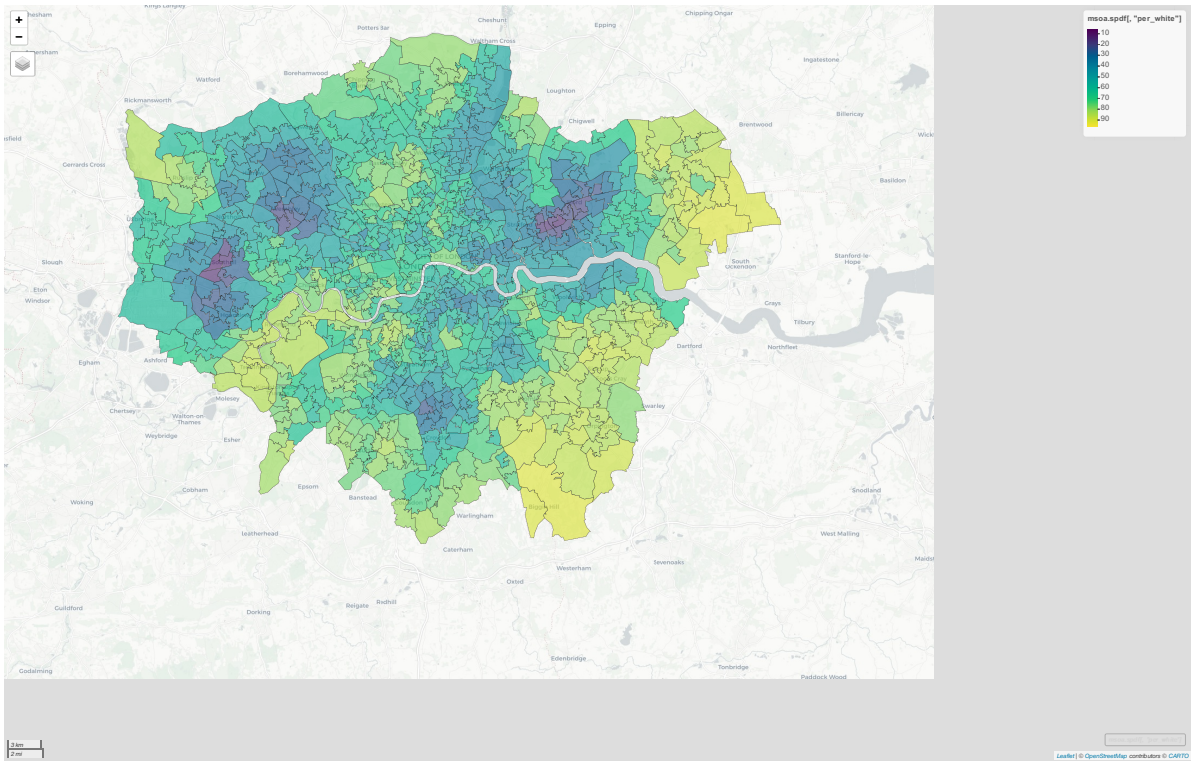
```
save(census_keystat_cb.df, file = "_data/Census_codebook.RData")
```

Now, we have one file containing the geometries of MSOAs and one file with the census information on ethnic groups. Obviously, we can easily merge them together using the MSOA identifiers.

```
msoa.spdf <- merge(msoa.spdf, census_keystat.df,
                   by.x = "MSOA11CD", by.y = "msoa11", all.x = TRUE)
```

And we can, for instance, plot the spatial distribution of ethnic groups.

```
msoa.spdf$per_white <- msoa.spdf$KS201EW_100 / msoa.spdf$KS201EW0001 * 100
msoa.spdf$per_mixed <- msoa.spdf$KS201EW_200 / msoa.spdf$KS201EW0001 * 100
msoa.spdf$per_asian <- msoa.spdf$KS201EW_300 / msoa.spdf$KS201EW0001 * 100
msoa.spdf$per_black <- msoa.spdf$KS201EW_400 / msoa.spdf$KS201EW0001 * 100
msoa.spdf$per_other <- msoa.spdf$KS201EW_500 / msoa.spdf$KS201EW0001 * 100

mapview(msoa.spdf[, "per_white"])
```



If you're interested in more data sources, see for instance APIs for social scientists: A collaborative review by Paul C. Bauer, Camille Landesvatter, Lion Behrens. It's a collection of several APIs for social sciences.

1.3.3 Gridded data

So far, we have queried data on administrative units. However, often data comes on other spatial scales. For instance, we might be interested in the amount of air pollution, which is provided on a regular grid across the UK from Defra.

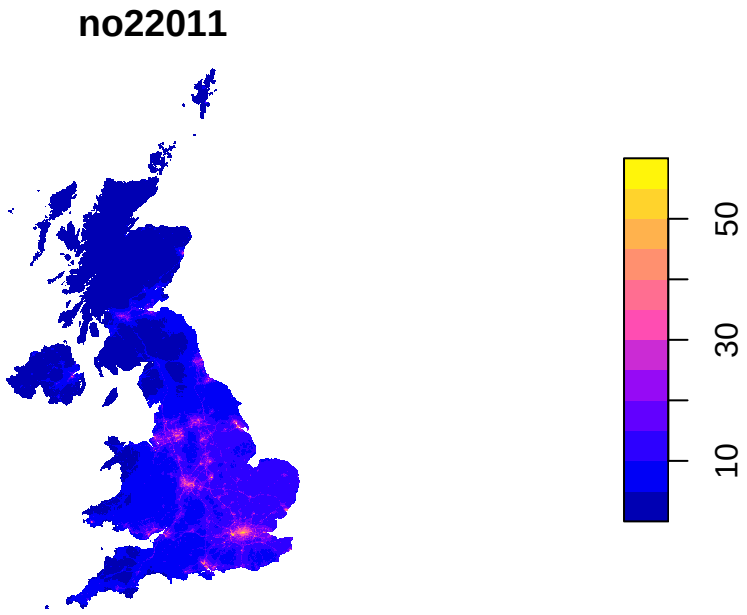
```
# Download
pol.link <- "https://uk-air.defra.gov.uk/datastore/pcm/mapno22011.csv"
download.file(pol.link, paste0(dn, "/mapno22011.csv"))
pol.df <- read.csv(paste0(dn, "/mapno22011.csv"), skip = 5, header = T, sep = ",",
                   stringsAsFactors = F, na.strings = "MISSING")

head(pol.df)
```

```
ukgridcode      x       y no22011
```

```
1      54291 460500 1221500      NA
2      54292 461500 1221500      NA
3      54294 463500 1221500      NA
4      54979 458500 1220500      NA
5      54980 459500 1220500      NA
6      54981 460500 1220500      NA
```

The data comes as point data with x and y as coordinates. We have to transform this into spatial data first. We first setup a spatial points object with `st_as_sf`. Subsequently, we transform the point coordinates into a regular grid. We use a buffer method `st_buffer` with "diameter", and only one segment per quadrant (`nQuadSegs`). This gives us a 1x1km regular grid.

```
# Build spatial object
pol.spdf <- st_as_sf(pol.df, coords = c("x", "y"),
                     crs = 27700)

# we transform the point coordinates into a regular grid with "diameter" 500m
pol.spdf <- st_buffer(pol.spdf, dist = 500, nQuadSegs  = 1,
                      endCapStyle = 'SQUARE')

# Plot NO2
plot(pol.spdf[, "no22011"], border = NA)
```



1.3.4 OpenStreetMap (points)

Another interesting data source is the OpenStreetMap API, which provides information about the geographical location of a serious of different indicators. Robin Lovelace provides a nice introduction to the osmdata API. Available features can be found on OSM wiki.

First we create a bounding box of where we want to query data. `st_bbox()` can be used to get bounding boxes of an existing spatial object (needs `CRS = 4326`). An alternative would be to use `opq(bbox = 'greater london uk')`.

```
# bounding box of where we want to query data
q <- opq(bbox = st_bbox(st_transform(msoa.spdf, 4326)))
```

And we want to get data for all pubs and bars which are within this bounding box.

```
# First build the query of location of pubs in London
osmq <- add_osm_feature(q, key = "amenity", value = "pub")

# And then query the data
pubs.osm <- osmdata_sf(osmq)
```

Right now there are some results in polygons, some in points, and they overlap. Often, data from OSM needs some manual cleaning. Sometimes the same features are represented by different spatial objects (e.g. points + polygons).

```
# Make unique points / polygons
pubs.osm <- unique_osmdata(pubs.osm)

# Get points and polygons (there are barley any pubs as polygons, so we ignore them)
pubs.points <- pubs.osm$osm_points
pubs.polys <- pubs.osm$osm_multipolygons

# # Drop OSM file
# rm(pubs.osm); gc()

# Reduce to point object only
pubs.spdf <- pubs.points

# Reduce to a few variables
pubs.spdf <- pubs.spdf[, c("osm_id", "name", "addr:postcode", "diet:vegan")]
```

Again, we can inspect the results with `mapview`.

```
mapview(st_geometry(pubs.spdf))
```

Note that OSM is solely based on contribution by users, and the **quality of OSM data varies**. Usually data quality is better in larger cities, and better for more stable features (such as hospitals, train stations, highways) rahter than pubs or restaurants which regularly appear and disappear. However, data from London Datastore would indicate more pubs than what we find with OSM.

1.3.5 Save

We will store the created data to use them again in the next session.

```
save(msoa.spdf, file = "_data/msoa_spatial.RData")
save(ulez.spdf, file = "_data/ulez_spatial.RData")
save(pol.spdf, file = "_data/pollution_spatial.RData")
save(pubs.spdf, file = "_data/pubs_spatial.RData")
```

1.4 Data Manipulation

Required packages

```
pkgs <- c("sf", "gstat", "mapview", "nngeo", "rnaturalearth", "dplyr",
          "nomisr", "osmdata", "tidyr", "texreg", "downlit", "xml2")
lapply(pkgs, require, character.only = TRUE)
```

Having data with geo-spatial information allows to perform a variety of methods to manipulate and link different data sources. Commonly used methods include 1) subsetting, 2) point-in-polygon operations, 3) distance measures, 4) intersections or buffer methods.

The online Vignettes of the sf package provide a comprehensive overview of the multiple ways of spatial manipulations.

1.4.0.1 Check if data is on common projection

```
st_crs(msoa.spdf) == st_crs(pol.spdf)
```

```
[1] FALSE
```

```
st_crs(msoa.spdf) == st_crs(pubs.spdf)
```

```
[1] FALSE
```

```
st_crs(msoa.spdf) == st_crs(ulez.spdf)
```

```
[1] FALSE
```

The spatial data files are on different projections. Before we can do any spatial operations with them, we have to transform them into a common projection.

```
# MSOA in different crs --> transform
pol.spdf <- st_transform(pol.spdf, crs = st_crs(msoa.spdf))
pubs.spdf <- st_transform(pubs.spdf, crs = st_crs(msoa.spdf))
ulez.spdf <- st_transform(ulez.spdf, crs = st_crs(msoa.spdf))
```

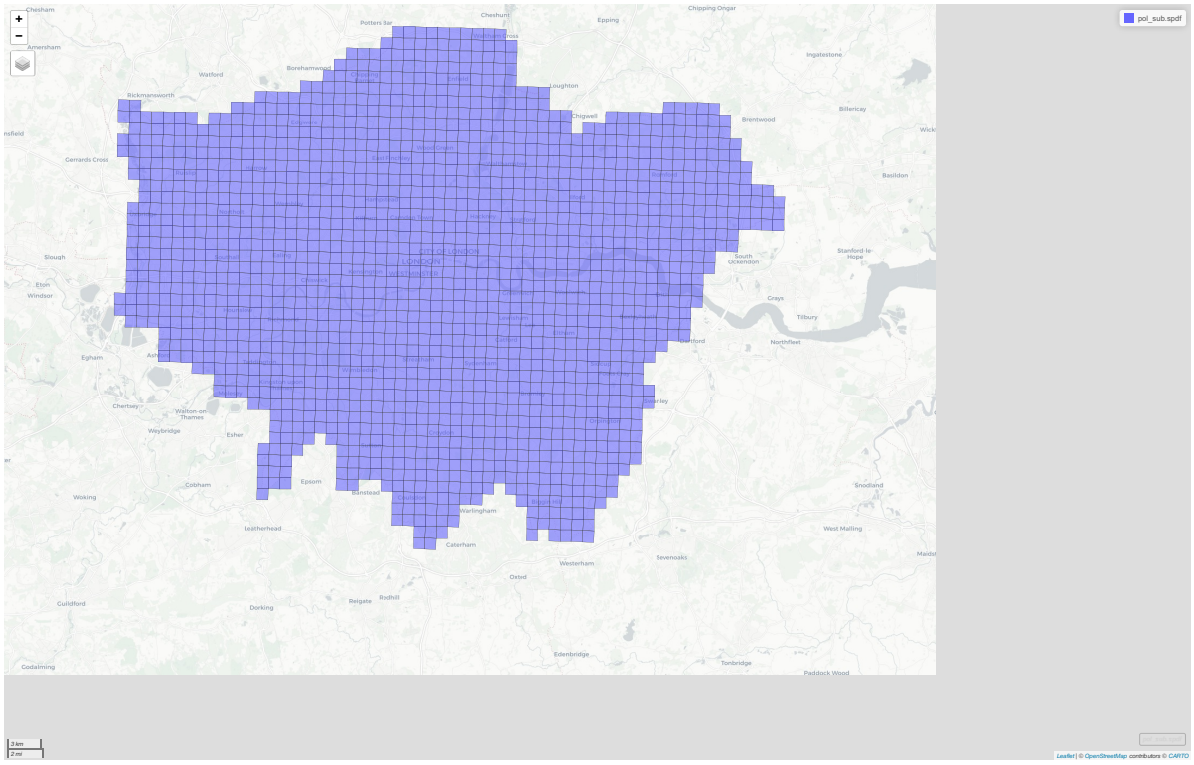
```
# Check if all geometries are valid, and make valid if needed
msoa.spdf <- st_make_valid(msoa.spdf)
```

The `st_make_valid()` can help if the spatial geometries have some problems such as holes or points that don't match exactly.

1.4.1 Subsetting

We can subset spatial data in a similar way as we subset conventional data.frames or matrices. For instance, below we simply reduce the pollution grid across the UK to observations in London only.

```
# Subset to pollution estimates in London
pol_sub.spdf <- pol.spdf[msoa.spdf, ] # or:
pol_sub.spdf <- st_filter(pol.spdf, msoa.spdf)
mapview(pol_sub.spdf)
```



Or we can reverse the above and exclude all intersecting units by specifying `st_disjoint` as

alternative spatial operation using the `op =` option (note the empty space for column selection). `st_filter()` with the `.predicate` option does the same job. See the sf Vignette for more operations.

```
# Subset pubs to pubs not in the ulez area
sub2.spdf <- pubs.spdf[ulez.spdf, , op = st_disjoint] # or:
sub2.spdf <- st_filter(pubs.spdf, ulez.spdf, .predicate = st_disjoint)
mapview(sub2.spdf)
```

We can easily create indicators of whether an MSOA is within ulez or not.

```
msoa.spdf$ulez <- 0

# intersecting lsoas
within <- msoa.spdf[ulez.spdf,]

# use their ids to create binary indicator
msoa.spdf$ulez[which(msoa.spdf$MSOA11CD %in% within$MSOA11CD)] <- 1
table(msoa.spdf$ulez)
```

```
  0   1
955  28
```

1.4.2 Point in polygon

We are interested in the number of pubs in each MSOA. So, we count the number of points in each polygon.

```
# Assign MSOA to each point
pubs_msoa.join <- st_join(pubs.spdf, msoa.spdf, join = st_within)

# Count N by MSOA code (drop geometry to speed up)
pubs_msoa.join <- dplyr::count(st_drop_geometry(pubs_msoa.join),
                               MSOA11CD = pubs_msoa.join$MSOA11CD,
                               name = "pubs_count")
sum(pubs_msoa.join$pubs_count)
```

```
[1] 1601
```

```
# Merge and replace NAs with zero (no matches, no pubs)
msoa.spdf <- merge(msoa.spdf, pubs_msoa.join,
                   by = "MSOA11CD", all.x = TRUE)
msoa.spdf$pubs_count[is.na(msoa.spdf$pubs_count)] <- 0
```

1.4.3 Distance measures

We might be interested in the distance to the nearest pub. Here, we use the package `nngeo` to find k nearest neighbours with the respective distance.

```
# Use geometric centroid of each MSOA
cent.sp <- st_centroid(msoa.spdf[, "MSOA11CD"])
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```
# Get K nearest neighbour with distance
knb.dist <- st_nn(cent.sp,
                  pubs.spdf,
```

```
                    k = 1,              # number of nearest neighbours
                    returnDist = TRUE, # we also want the distance
                    progress = FALSE)
```

projected points

```
msoa.spdf$dist_pubs <- unlist(knb.dist$dist)
summary(msoa.spdf$dist_pubs)
```

```
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  9.079  305.149  565.018  701.961  948.047 3735.478
```

1.4.4 Intersections + Buffers

We may also want the average pollution within 1 km radius around each MSOA centroid. Note that it is usually better to use a ego-centric method where you calculate the average within a distance rather than using the characteristic of the intersecting cells only (B. A. Lee et al. 2008; Mohai and Saha 2007).

Therefore, we first create a buffer with `st_buffer()` around each midpoint and subsequently use `st_intersetion()` to calculate the overlap.

```
# Create buffer (1km radius)
cent.buf <- st_buffer(cent.sp,
                      dist = 1000) # dist in meters
mapview(cent.buf)
```


```
# Add area of each buffer (in this constant)
cent.buf$area <- as.numeric(st_area(cent.buf))

# Calculate intersection of pollution grid and buffer
int.df <- st_intersection(cent.buf, pol.spdf)
```

```
Warning: attribute variables are assumed to be spatially constant throughout
all geometries
```

```
int.df$int_area <- as.numeric(st_area(int.df)) # area of intersection

# Area of intersection as share of buffer
int.df$area_per <- int.df$int_area / int.df$area
```

And we use the percent overalp areas as the weights to calculate a weighted mean.

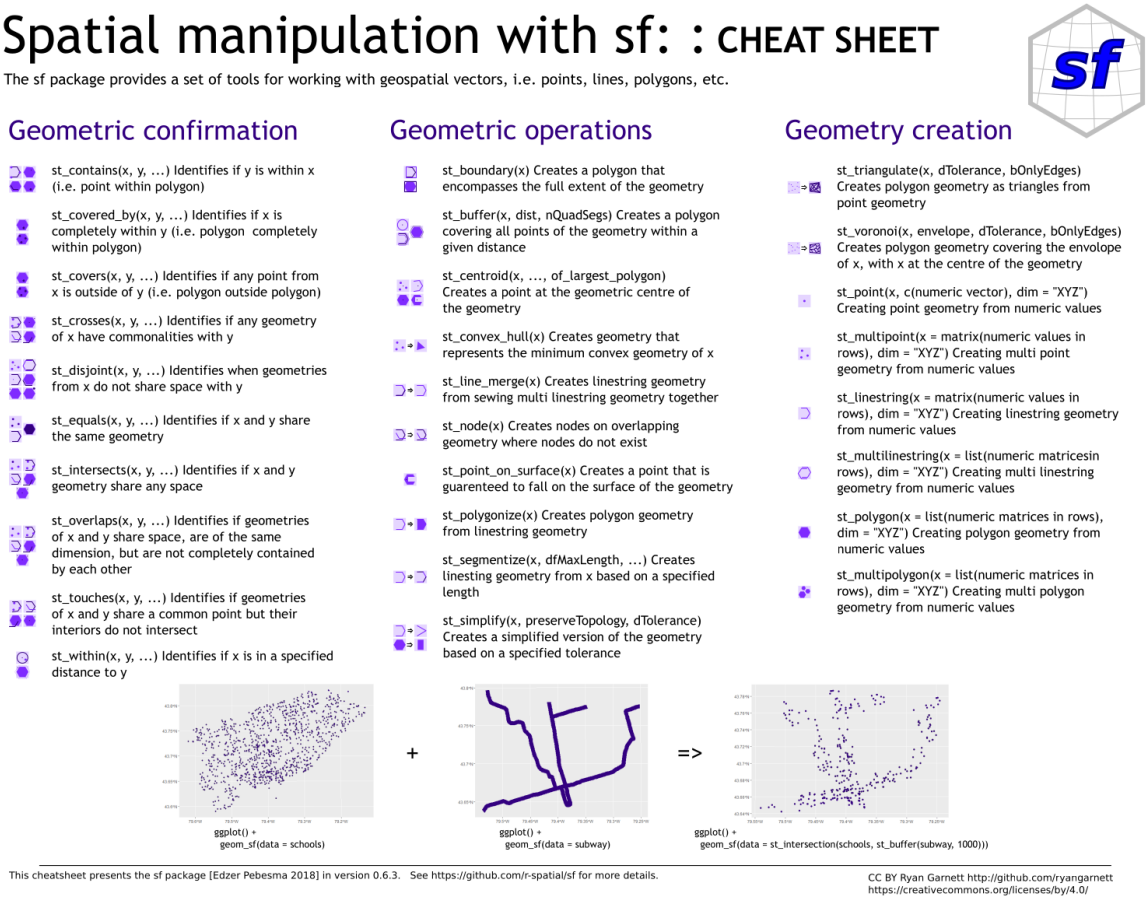
```
# Aggregate as weighted mean
int.df <- st_drop_geometry(int.df)
```

```r
int.df$no2_weighted <- int.df$no22011 * int.df$area_per
int.df <- aggregate(list(no2 = int.df[, "no2_weighted"]),
                    by = list(MSOA11CD = int.df$MSOA11CD),
                    sum)

# Merge back to spatial data.frame
msoa.spdf <- merge(msoa.spdf, int.df, by = "MSOA11CD", all.x = TRUE)

mapview(msoa.spdf[, "no2"])
```

Note: for buffer related methods, it often makes sense to use population weighted centroids instead of geographic centroids (see here for MSOA population weighted centroids). However, often this information is not available.

1.4.5 and more

There are more spatial operation possible using sf. Have a look at the sf Cheatsheet.



1.5 Data visualisation

For mapping

```
pkgs <- c("tmap", "tmaptools", "viridisLite",
          "ggplot2", "ggthemes", "rmapshaper")
lapply(pkgs, require, character.only = TRUE)
```

A large advantage of spatial data is that different data sources can be connected and combined. Another nice advantage is: you can create very nice maps. And it's quite easy to do! Stefan Jünger & Anne-Kathrin Stroppe provide more comprehensive materials on mapping in their GESIS workshop on geospatial techniques in R.

Many packages and functions can be used to plot maps of spatial data. For instance, ggplot as a function to plot spatial data using `geom_sf()`. I am personally a fan of `tmap`, which makes

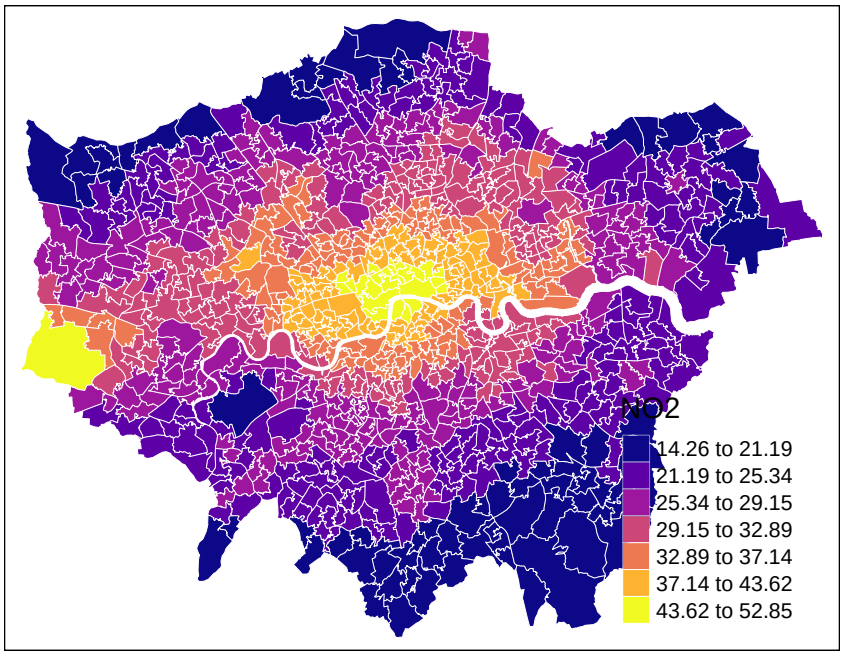
many steps easier (but sometimes is less flexible).

A great tool for choosing coulour is for instance Colorbrewer. `viridisLite` provides another great resource to chose colours.

1.5.1 Tmaps

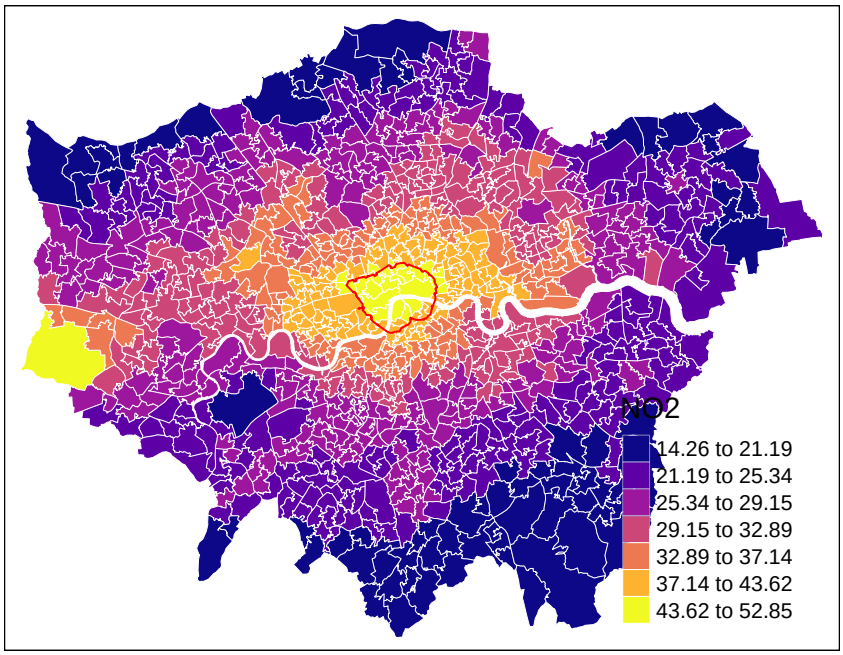
For instance, lets plot the NO2 estimates using tmap + `tm_fill()` (there are lots of alternatives like `tm_shape`, `tm_points()`, `tm_dots()`).

```
# Define colours
cols <- viridis(n = 7, direction = 1, option = "C")

mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "no2",
          style = "fisher", # algorithm to def cut points
          n = 7, # Number of requested cut points
          palette = cols, # colours
          alpha = 1, # transparency
          title = "NO2",
          legend.hist = FALSE # histogram next to map?
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5)

mp1
```



Tmap allows to easily combine different objects by defining a new object via `tm_shape()`.

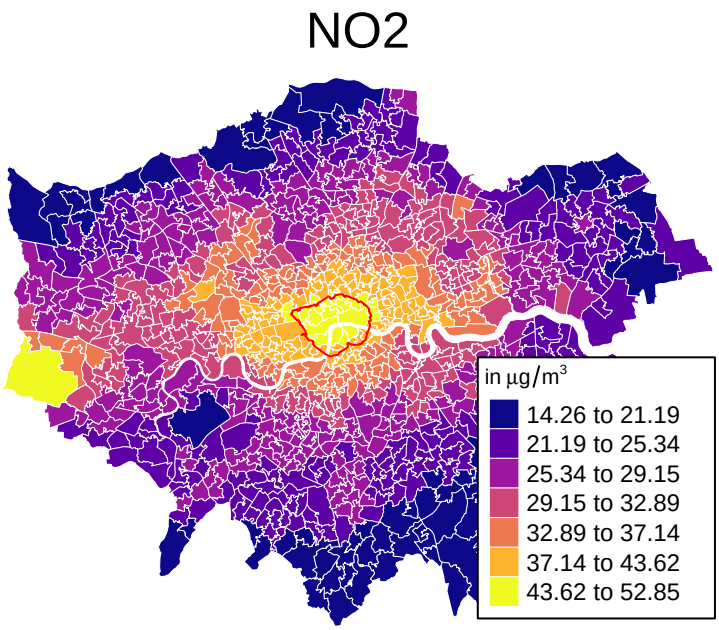
```
# Define colours
cols <- viridis(n = 7, direction = 1, option = "C")

mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "no2",
          style = "fisher", # algorithm to def cut points
          n = 7, # Number of requested cut points
          palette = cols, # colours
          alpha = 1, # transparency
          title = "NO2",
          legend.hist = FALSE # histogram next to map?
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_shape(ulez.spdf) +
  tm_borders(col = "red", lwd = 1, alpha = 1)

mp1
```



And it is easy to change the layout.

```
# Define colours
cols <- viridis(n = 7, direction = 1, option = "C")

mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "no2",
          style = "fisher", # algorithm to def cut points
          n = 7, # Number of requested cut points
          palette = cols, # colours
          alpha = 1, # transparency
          title = expression('in'~mu*'g'/m^{3}),
          legend.hist = FALSE # histogram next to map?
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_shape(ulez.spdf) +
  tm_borders(col = "red", lwd = 1, alpha = 1) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("right", "bottom"),
            legend.outside = FALSE,
            main.title = "NO2",
```

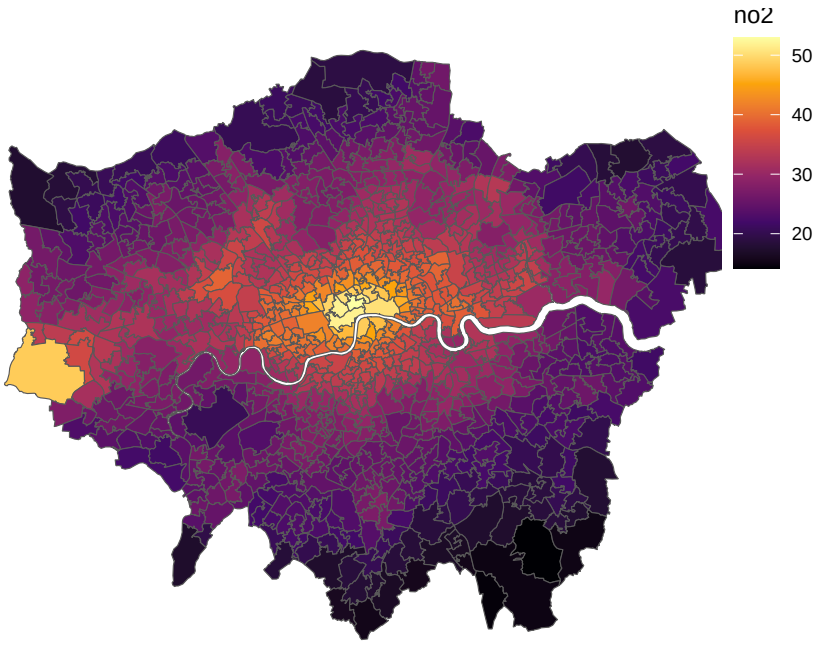
```
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8)

mp1
```



1.5.2 ggplot

For those of you have rather stick with the basic ggplot package, we can also use ggplot for spatial maps.

```
gp <- ggplot(msoa.spdf)+
    geom_sf(aes(fill = no2))+
    scale_fill_viridis_c(option = "B")+
    coord_sf(datum = NA)+
    theme_map()+
    theme(legend.position = c(.9, .6))
gp
```

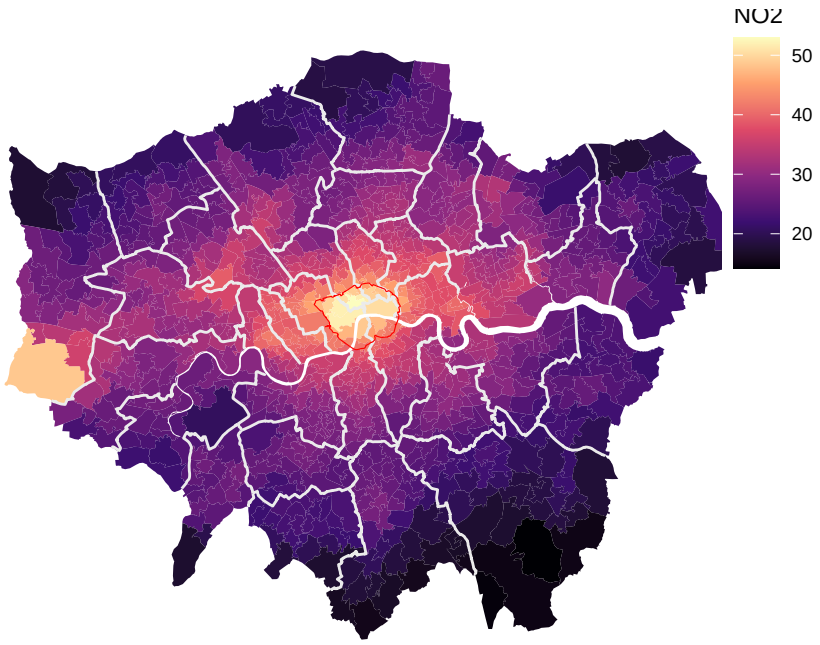



```
# Get some larger scale boundaries
borough.spdf <- st_read(dsn = paste0("_data", "/statistical-gis-boundaries-london/ESRI"),
                        layer = "London_Borough_Excluding_MHW" # Note: no file ending
                        )
```

```
Reading layer `London_Borough_Excluding_MHW' from data source
  `C:\work\Lehre\Geodata_Spatial_Regression_short\_data\statistical-gis-
boundaries-london\ESRI'
  using driver `ESRI Shapefile'
Simple feature collection with 33 features and 7 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 503568.2 ymin: 155850.8 xmax: 561957.5 ymax: 200933.9
Projected CRS: OSGB36 / British National Grid
```

```
# transform to only inner lines
borough_inner <- ms_innerlines(borough.spdf)

# Plot with inner lines
gp <- ggplot(msoa.spdf)+
    geom_sf(aes(fill = no2), color = NA)+
```

```
    scale_fill_viridis_c(option = "A")+
    geom_sf(data = borough_inner, color = "gray92")+
    geom_sf(data = ulez.spdf, color = "red", fill = NA)+
    coord_sf(datum = NA)+
    theme_map()+
    labs(fill = "NO2")+
    theme(legend.position = c(.9, .6))
gp
```



1.6 Exercise

1) What is the difference between a spatial "sf" object and a conventional "data.frame"? What's the purpose of the function `st_drop_geometry()`?

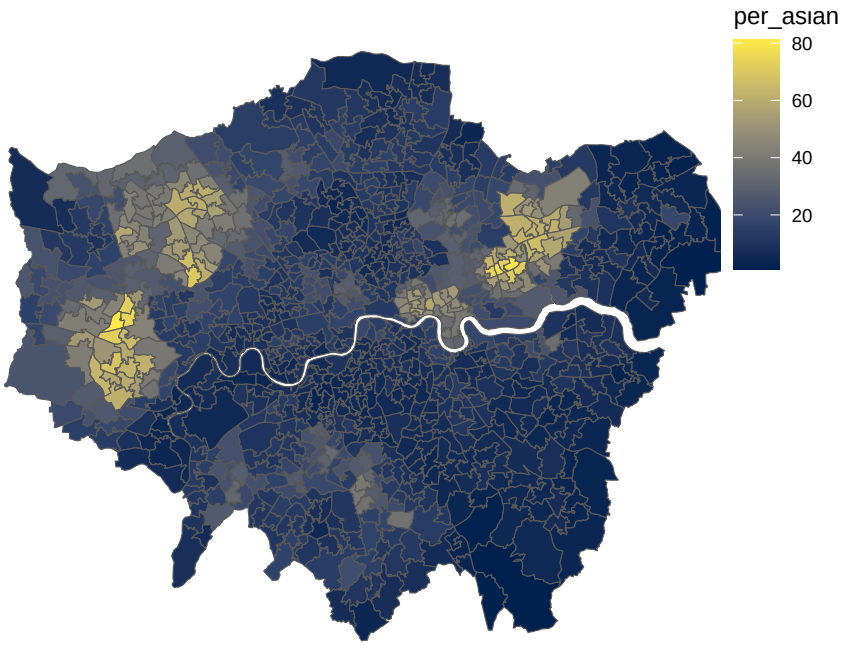
It's the same. A spatial "sf" object just has an additional column containing the spatial coordinates.

2) Using msoa.spdf, please create a spatial data frame that contains only the MSOA areas that are within the ulez zone.

```
sub4.spdf <- msoa.spdf[ulez.spdf, ]
```

3) Please create a map for London (or only the msoa-ulez subset) which shows the share of Asian residents (or any other ethnic group).

```
gp <- ggplot(msoa.spdf)+
    geom_sf(aes(fill = per_asian))+
    scale_fill_viridis_c(option = "E")+
    coord_sf(datum = NA)+
    theme_map()+
    theme(legend.position = c(.9, .6))
gp
```



4) Please calculate the distance of each MSOA to the London city centre

a) use google maps to get lon and lat,
b) use `st_as_sf()` to create the spatial point
c) use `st_distance()` to calculate the distance

```
### Distance to city center
# Define centre
centre <- st_as_sf(data.frame(lon = -0.128120855701165,
                              lat = 51.50725909644806),
                   coords = c("lon", "lat"),
```

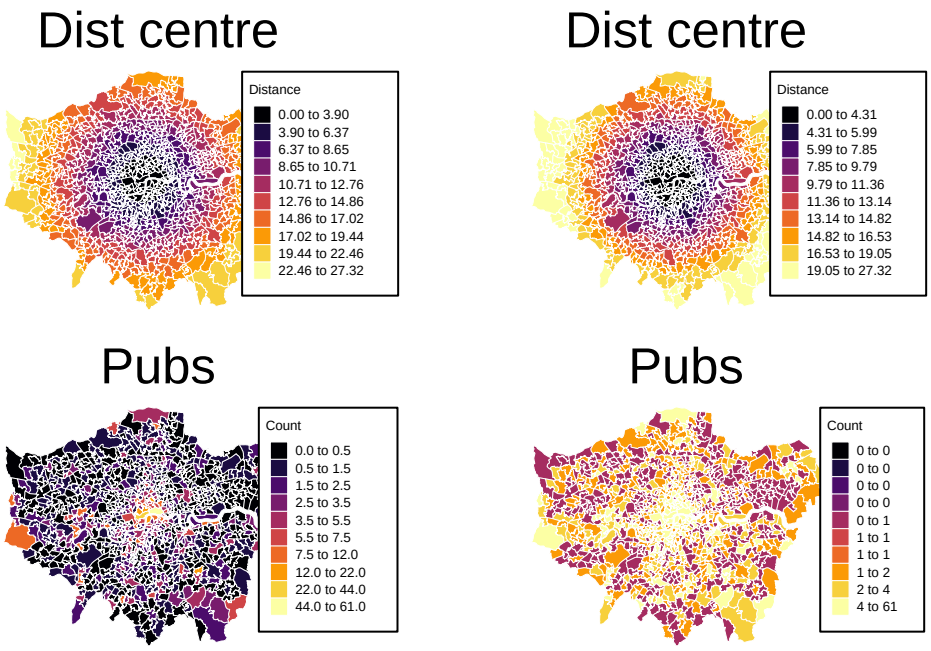
```
                    crs = 4326)
# Reproject
centre <- st_transform(centre, crs = st_crs(msoa.spdf))
# Calculate distance
msoa.spdf$dist_centre <- as.numeric(st_distance(msoa.spdf, centre)) / 1000
# hist(msoa.spdf$dist_centre)
```

5) Can you create a plot with the distance to the city centre and pub counts next to each other?

```
# Define colours
cols <- viridis(n = 10, direction = 1, option = "B")
cols2 <- viridis(n = 10, direction = 1, option = "E")


mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "dist_centre",
          style = "fisher", # algorithm to def cut points
          n = 10, # Number of requested cut points
          palette = cols, # colours
          alpha = 1, # transparency
          title = "Distance",
          legend.hist = FALSE # histogram next to map?
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("right", "bottom"),
            legend.outside = FALSE,
            main.title = "Dist centre",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8)


mp2 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "dist_centre",
          style = "quantile", # algorithm to def cut points
          n = 10, # Number of requested cut points
          palette = cols, # colours
```

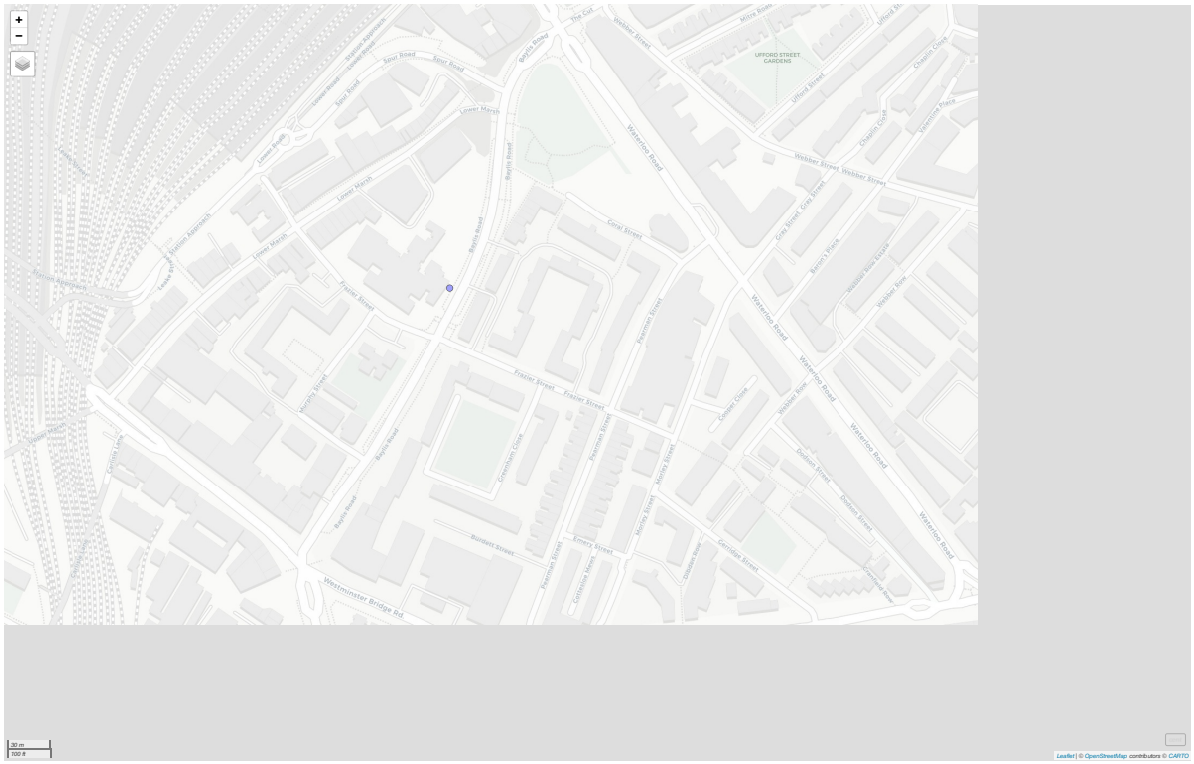
```
          alpha = 1, # transparency
          title = "Distance",
          legend.hist = FALSE # histogram next to map?
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("right", "bottom"),
            legend.outside = FALSE,
            main.title = "Dist centre",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8)


mp3 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "pubs_count",
          style = "fisher", # algorithm to def cut points
          n = 10, # Number of requested cut points
          palette = cols, # colours
          alpha = 1, # transparency
          title = "Count",
          legend.hist = FALSE # histogram next to map?
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("right", "bottom"),
            legend.outside = FALSE,
            main.title = "Pubs",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8)


mp4 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "pubs_count",
          style = "quantile", # algorithm to def cut points
          n = 10, # Number of requested cut points
```

```
          palette = cols, # colours
          alpha = 1, # transparency
          title = "Count",
          legend.hist = FALSE # histogram next to map?
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("right", "bottom"),
            legend.outside = FALSE,
            main.title = "Pubs",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8)


tmap_arrange(mp1, mp2, mp3, mp4, ncol = 2, nrow = 2)
```



1.6.1 Geogrphic cter

```
# Make one single schape
london <- st_union(msoa.spdf)

# Calculate center
cent <- st_centroid(london)

mapview(cent)
```



2 Spatial Relationships

Required packages

```
pkgs <- c("sf", "mapview", "spdep", "spatialreg", "tmap", "viridisLite") # note: load spde
lapply(pkgs, require, character.only = TRUE)
```

Session info

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default


locale:
[1] LC_COLLATE=English_United Kingdom.utf8
[2] LC_CTYPE=English_United Kingdom.utf8
[3] LC_MONETARY=English_United Kingdom.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.utf8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] viridisLite_0.4.2 tmap_3.3-3        spatialreg_1.2-9  Matrix_1.5-4.1
[5] spdep_1.2-8       spData_2.3.0      mapview_2.11.0    sf_1.0-13
```

```
loaded via a namespace (and not attached):
 [1] xfun_0.39          raster_3.6-23      htmlwidgets_1.6.2  lattice_0.21-8
 [5] vctrs_0.6.3        tools_4.3.1        crosstalk_1.2.0    LearnBayes_2.15.1
 [9] generics_0.1.3     parallel_4.3.1     sandwich_3.0-2     stats4_4.3.1
[13] tibble_3.2.1       proxy_0.4-27       fansi_1.0.4        pkgconfig_2.0.3
[17] KernSmooth_2.23-21 satellite_1.0.4    RColorBrewer_1.1-3 leaflet_2.1.2
[21] webshot_0.5.5      lifecycle_1.0.3    compiler_4.3.1     deldir_1.0-9
[25] munsell_0.5.0      terra_1.7-39       leafsync_0.1.0     codetools_0.2-
19
[29] stars_0.6-1        htmltools_0.5.5    class_7.3-22       pillar_1.9.0
[33] MASS_7.3-60        classInt_0.4-9     lwgeom_0.2-13      wk_0.7.3
[37] abind_1.4-5        boot_1.3-28.1      multcomp_1.4-25    nlme_3.1-162
[41] tidyselect_1.2.0   digest_0.6.32      mvtnorm_1.2-2      dplyr_1.1.2
[45] splines_4.3.1      fastmap_1.1.1      grid_4.3.1         colorspace_2.1-
0
[49] expm_0.999-7       cli_3.6.1          magrittr_2.0.3     base64enc_0.1-3
[53] dichromat_2.0-0.1  XML_3.99-0.14      survival_3.5-5     utf8_1.2.3
[57] TH.data_1.1-2      leafem_0.2.0       e1071_1.7-13       scales_1.2.1
[61] sp_1.6-1           rmarkdown_2.23     zoo_1.8-12         png_0.1-8
[65] coda_0.19-4        evaluate_0.21      knitr_1.43         tmaptools_3.1-1
[69] s2_1.1.4           rlang_1.1.1        Rcpp_1.0.10        glue_1.6.2
[73] DBI_1.1.3          rstudioapi_0.14    jsonlite_1.8.5     R6_2.5.1
[77] units_0.8-2
```

Reload data from pervious session

```
load("_data/msoa2_spatial.RData")
```

2.1 Spatial interdependence

We can not only use coordinates and geo-spatial information to connect different data sources, we can also explicitly model spatial (inter)dependence in the analysis of our data. In many instance, accounting for spatial dependence might even be necessary to avoid biased point estimates and standard errors, as observations are often not independent and identically distributed.

Tobler's first law of geography has been used extensively (11,584 citation in 2023-06) to describe spatial dependence: 'Everything is related to everything else, but near things are more related than distant things' (Tobler 1970).

> **Note**
>
> Tobler's first law is a bit of story
> And it has been labeled as an excuse to not think too much about the reasons for spatial dependence or auto-correlation. For instance, measurement error, omitted variables, or inappropriate levels of aggregation are among reasons for auto-correlation (Pebesma and Bivand 2023).

We will come back to the reasons of spatial dependence. However, for now, we are interested in some tools to detect and analyse spatial relations.

To analyse spatial relations, we first need to define some sort of connectivity between units (e.g. similar to network analysis). There are some obvious candidates that be used to define these relations here: adjacency and proximity.

2.2 W: Connectivity between units

The connectivity between units is usually represented in a matrix **W**. There is an ongoing debate about the importance of spatial weights for spatial econometrics and about the right way to specify weights matrices (LeSage and Pace 2014; Neumayer and Plümper 2016). The following graph shows some possible options in how to define connectivity between units.

In spatial econometrics, the spatial connectivity (as shown above) is usually represented by a spatial weights matrix **W**:

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}$$

The spatial weights matrix **W** is an $N \times N$ dimensional matrix with elements w_{ij} specifying the relation or connectivity between each pair of units i and j.

Note: The diagonal elements $w_{i,i} = w_{1,1}, w_{2,2}, \dots, w_{n,n}$ of **W** are always zero. No unit is a neighbour of itself. This is not true for spatial multiplier matrices (as we will see later).

2.2.1 Contiguity weights

A very common type of spatial weights. Binary specification, taking the value 1 for neighbouring units (queens: sharing a common edge; rook: sharing a common border), and 0 otherwise.

Contiguity weights $w_{i,j}$, where

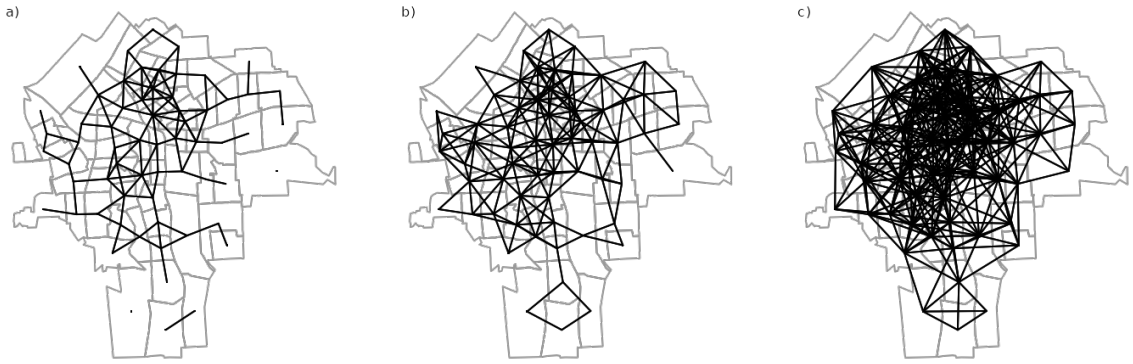


Figure 2.1: Figure: Different measures of connectivity, Source: R. S. Bivand and Rudel (2018)

$$w_{i,j} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ neighbours} \\ 0 & \text{otherwise} \end{cases}$$

A contiguity weights matrix with three units, where unit 1 and unit 3 are neighbours, while unit 2 has no neighbours would look like this:

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

- Sparse matrices
- Problem of 'island': units without neighbours (if I calculate an average of their neigbours, would that be zero, or NA, or a mean?)

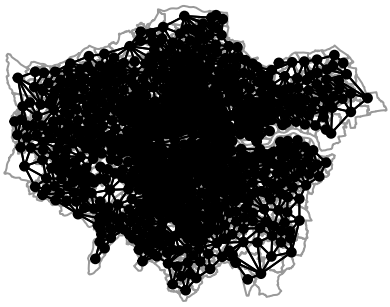
Lets create a contiguity weights matrix (Queens neighbours) for the London MSOAs: we create a neighbours list (`nb`) using `poly2nb()`, which is an efficient way of storing $\mathbf{W}$. A `snap` of 1 meter accounts for potential lacks of accuracy between lines and points.

```
# Contiguity (Queens) neighbours weights
queens.nb <- poly2nb(msoa.spdf,
                     queen = TRUE, # a single shared boundary point meets the contiguity c
                     snap = 1) # we consider points in 1m distance as 'touching'
summary(queens.nb)
```

```
Neighbour list object:
Number of regions: 983
Number of nonzero links: 5648
Percentage nonzero weights: 0.5845042
Average number of links: 5.745677
Link number distribution:

  2   3   4   5   6   7   8   9  10  11  12  13
  9  39 130 264 273 169  66  19   5   6   2   1
9 least connected regions:
160 270 475 490 597 729 755 778 861 with 2 links
1 most connected region:
946 with 13 links
```

```
# Lets plot that
plot(st_geometry(msoa.spdf), border = "grey60")
```

```r
plot(queens.nb, st_centroid(st_geometry(msoa.spdf)),
     add = TRUE, pch = 19, cex = 0.6)
```



```r
# We can also transform this into a matrix W
W <- nb2mat(queens.nb, style = "B")
print(W[1:10, 1:10])
```

```
   [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
1     0    0    0    0    0    0    0    0    0     0
2     0    0    1    0    0    0    0    0    0     0
3     0    1    0    0    1    0    0    0    0     0
4     0    0    0    0    0    1    0    0    0     1
5     0    0    1    0    0    1    1    0    0     0
6     0    0    0    1    1    0    1    0    1     1
7     0    0    0    0    1    1    0    1    1     0
8     0    0    0    0    0    0    1    0    0     0
9     0    0    0    0    0    1    1    0    0     1
10    0    0    0    1    0    1    0    0    1     0
```

Question

Among those first 10 units that you see above, which are the neighbours of unit number 6?
Why is the diagonal of this matrix all zero?

Overall, the matrix W has dimensions $N \times N$, a row and a column for each observation. The value in a cell shows how units i (row number) and j (column number) are related to each other.

```
dim(W)
```

```
[1] 983 983
```

The row and column sums indicate the number of neighbours of each observation.

```
rowSums(W)[1:10]
```

```
 1  2  3  4  5  6  7  8  9 10
11  6  7  5  5  6  6  6  6  5
```

```
colSums(W)[1:10]
```

```
 [1] 11  6  7  5  5  6  6  6  6  5
```

Adjacency or graph-based neighbour's weights matrices are usually symmetric. If unit 1 is a neighbour of unit 55, then unit 55 is also a neighbour of unit 1.

Higher Order Neighbours

Your neighbours have neighbours too, and they are called higher (second) order neighbours. The neighbours of your neighbour's neighbours are third order neighbours.
You can use `nblag()` to calculate higher order neighbour relations.

2.2.2 Distance based weights

Another common type uses the distance d_{ij} between each unit i and j.

- Inverse distance weights $w_{i,j} = \frac{1}{d_{ij}^{\alpha}}$, where α define the strength of the spatial decay.

$$\mathbf{W} = \begin{bmatrix} 0 & \frac{1}{d_{ij}^{\alpha}} & \frac{1}{d_{ij}^{\alpha}} \\ \frac{1}{d_{ij}^{\alpha}} & 0 & \frac{1}{d_{ij}^{\alpha}} \\ \frac{1}{d_{ij}^{\alpha}} & \frac{1}{d_{ij}^{\alpha}} & 0 \end{bmatrix}$$

- Dense matrices
- Specifying thresholds may be useful (to get rid of very small non-zero weights)

For now, we will just specify a neighbours list with a distance threshold of 3km using `dnearneigh()`. An alternative would be k nearest neighbours using `knearneigh()`. We will do the inverse weighting later.

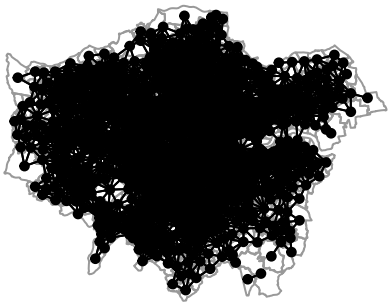
```
# Crease centroids
coords <- st_geometry(st_centroid(msoa.spdf))
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```
# Neighbours within 3km distance
dist_3.nb <- dnearneigh(coords, d1 = 0, d2 = 3000)
summary(dist_3.nb)
```

```
Neighbour list object:
Number of regions: 983
Number of nonzero links: 22086
Percentage nonzero weights: 2.285652
Average number of links: 22.46796
Link number distribution:

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 4  3  7 13 11 14 14 17 26 22 26 30 33 34 46 34 59 43 38 30 25 19 22 15 21 14
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
23 17 17 23 28 19 26 24 29 24 27 25 22 18  8 10 12  5  3  2  1
4 least connected regions:
158 160 463 959 with 1 link
1 most connected region:
545 with 47 links
```

```
# Lets plot that
plot(st_geometry(msoa.spdf), border = "grey60")
plot(dist_3.nb, coords,
     add = TRUE, pch = 19, cex = 0.6)
```



And you can see that the matrix is not so sparse anymore:

```
W2 <- nb2mat(dist_3.nb, style = "B")
W2[1:10, 1:10]
```

```
   [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
1     0    0    0    0    0    0    0    0    0     0
2     0    0    1    0    1    0    0    0    0     0
3     0    1    0    0    1    1    1    0    0     0
4     0    0    0    0    1    1    1    0    1     1
5     0    1    1    1    0    1    1    1    1     1
6     0    0    1    1    1    0    1    1    1     1
7     0    0    1    1    1    1    0    1    1     1
8     0    0    0    0    1    1    1    0    1     0
9     0    0    0    1    1    1    1    1    0     1
10    0    0    0    1    1    1    1    0    1     0
```

2.3 Normalization of W

Normalizing ensures that the parameter space of the spatial multiplier is restricted to $-1 < \rho > 1$, and the multiplier matrix is non-singular (don't worry, more on this later).

The main message: Normalizing your weights matrix is always a good idea. Otherwise, the spatial parameters might blow up – if you can estimate the model at all. It also ensure easy interpretation of spillover effects.

Again, how to normalize a weights matrix is subject of debate (LeSage and Pace 2014; Neumayer and Plümper 2016).

2.3.1 Row-normalization

Row-normalization divides each non-zero weight by the sum of all weights of unit i, which is the sum of the row.

$$\frac{w_{ij}}{\sum_j^n w_{ij}}$$

- With contiguity weights, spatially lagged variables contain mean of this variable among the neighbours of i
- Proportions between units such as distances get lost (can be bad!)
- Can induce asymmetries: $w_{ij} \neq w_{ji}$

For instance, we can use row-normalization for the Queens neighbours created above, and create a neighbours list with spatial weights.

```
queens.lw <- nb2listw(queens.nb,
                      style = "W") # W ist row-normalization
summary(queens.lw)
```

```
Characteristics of weights list object:
Neighbour list object:
Number of regions: 983
Number of nonzero links: 5648
Percentage nonzero weights: 0.5845042
Average number of links: 5.745677
Link number distribution:

  2   3   4   5   6   7   8   9  10  11  12  13
```

```
  9  39 130 264 273 169  66  19   5   6   2   1
9 least connected regions:
160 270 475 490 597 729 755 778 861 with 2 links
1 most connected region:
946 with 13 links

Weights style: W
Weights constants summary:
    n      nn  S0       S1      S2
W 983 966289 983 355.1333 4017.47
```

To see what happened, let's look at our example in matrix format again.

```
# transform into matrix with row-normalization
W_norm <- nb2mat(queens.nb, style = "W")
print(W_norm[1:10, 1:10])
```

```
   [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
1     0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
2     0 0.0000000 0.1666667 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
3     0 0.1428571 0.0000000 0.0000000 0.1428571 0.0000000 0.0000000 0.0000000
4     0 0.0000000 0.0000000 0.0000000 0.0000000 0.2000000 0.0000000 0.0000000
5     0 0.0000000 0.2000000 0.0000000 0.0000000 0.2000000 0.2000000 0.0000000
6     0 0.0000000 0.0000000 0.1666667 0.1666667 0.0000000 0.1666667 0.0000000
7     0 0.0000000 0.0000000 0.0000000 0.1666667 0.1666667 0.0000000 0.1666667
8     0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.1666667 0.0000000
9     0 0.0000000 0.0000000 0.0000000 0.0000000 0.1666667 0.1666667 0.0000000
10    0 0.0000000 0.0000000 0.2000000 0.0000000 0.2000000 0.0000000 0.0000000
        [,9]      [,10]
1  0.0000000 0.0000000
2  0.0000000 0.0000000
3  0.0000000 0.0000000
4  0.0000000 0.2000000
5  0.0000000 0.0000000
6  0.1666667 0.1666667
7  0.1666667 0.0000000
8  0.0000000 0.0000000
9  0.0000000 0.1666667
10 0.2000000 0.0000000
```

Question

Overall, how many neighbours does unit 9 have (including all columns)? How do you know?

```
rowSums(W)[9]
```

We can also use the nb object to see which ones the neighbours are. Here, for instance, neighbours of unit 6:

```
queens.nb[6]
```

```
[[1]]
[1]   4   5   7   9  10 462
```

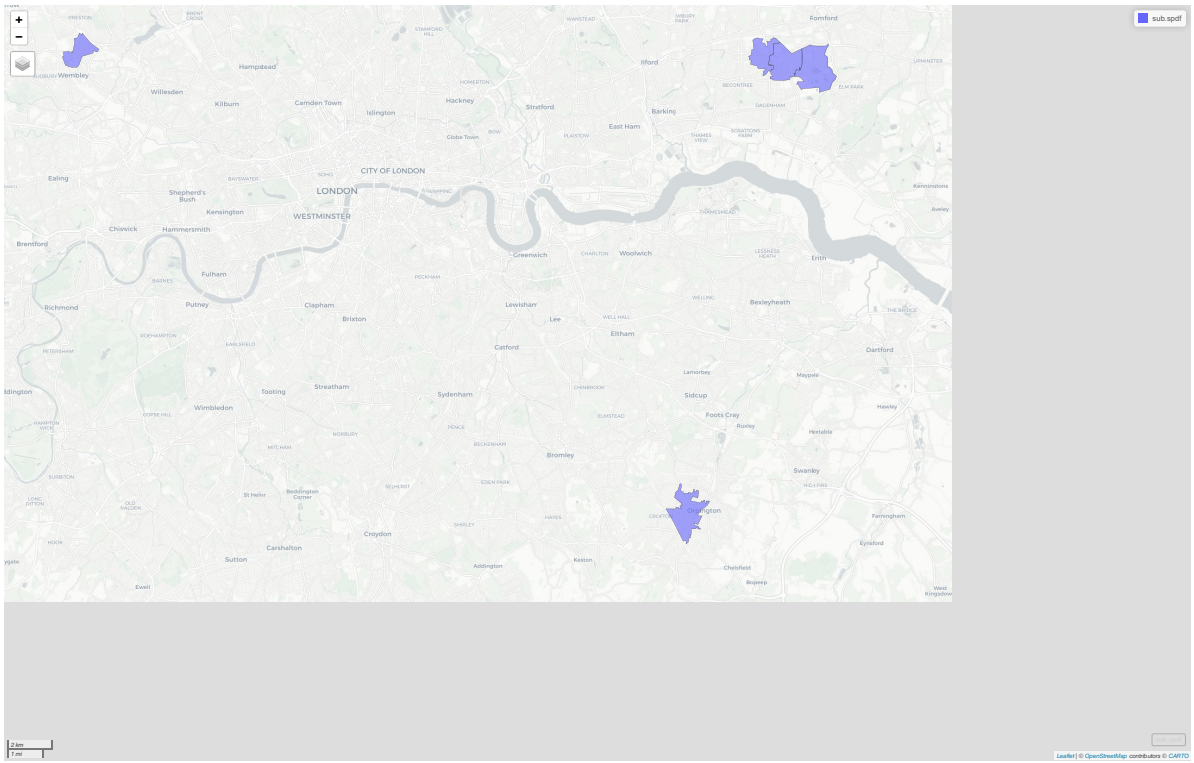
This fits to what we see in the matrix above.

Warning

Note that row-normalization has some undesirable properties when we use some non-contigutiy based neighbour relations, such as distance based neighbours.
The problem: It obscures the proportion due to dividing by a row-specific value.

Let's construct a hypothetical example

```
# Subset of 5 units
sub.spdf <- msoa.spdf[c(4, 5, 6, 102, 150), ]
mapview(sub.spdf)
```



We construct the **inverse-distance weighted 2 nearest neighbors**.

```
# 2 closest neighbours
sub.coords <- st_geometry(st_centroid(sub.spdf))
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```
knn.nb <- knearneigh(sub.coords,
                     k = 2) # number of nearest neighbours
```

```
Warning in knearneigh(sub.coords, k = 2): k greater than one-third of the
number of data points
```

```
knn.nb <- knn2nb(knn.nb)
summary(knn.nb)
```

```
Neighbour list object:
Number of regions: 5
Number of nonzero links: 10
Percentage nonzero weights: 40
Average number of links: 2
Non-symmetric neighbours list
Link number distribution:

2
5
5 least connected regions:
1 2 3 4 5 with 2 links
5 most connected regions:
1 2 3 4 5 with 2 links
```

```r
# listw with inverse-distance based weights
sub.lw <- nb2listwdist(knn.nb,
                       x = sub.coords, # needed for idw
                       type = "idw", # inverse distance weighting
                       alpha = 1, # the decay parameter for distance weighting
                       style = "raw") # without normalization
W_sub <- listw2mat(sub.lw)
formatC(W_sub, format = "f", digits = 6)
```

```
  [,1]       [,2]       [,3]       [,4]       [,5]
1 "0.000000" "0.000414" "0.000723" "0.000000" "0.000000"
2 "0.000414" "0.000000" "0.000962" "0.000000" "0.000000"
3 "0.000723" "0.000962" "0.000000" "0.000000" "0.000000"
4 "0.000000" "0.000033" "0.000032" "0.000000" "0.000000"
5 "0.000049" "0.000000" "0.000049" "0.000000" "0.000000"
```

As you can see, units 1, 2, 3 have relatively proximate neighbours (.e.g inverse distance 0.000962: 3 zeros). Units 4 and 5, in contrast, have only very distant neighbours (e.g. inverse distance 0.000049: 4 zeros).

Now, see what happens when we use row-normalization.

```r
sub.lw <- nb2listwdist(knn.nb,
                       x = sub.coords, # needed for idw
                       type = "idw", # inverse distance weighting
                       alpha = 1, # the decay parameter for distance weighting
```

```
                        style = "W") # for row normalization
W_sub <- listw2mat(sub.lw)
formatC(W_sub, format = "f", digits = 6)
```

```
  [,1]       [,2]       [,3]       [,4]       [,5]
1 "0.000000" "0.364083" "0.635917" "0.000000" "0.000000"
2 "0.300879" "0.000000" "0.699121" "0.000000" "0.000000"
3 "0.429123" "0.570877" "0.000000" "0.000000" "0.000000"
4 "0.000000" "0.507955" "0.492045" "0.000000" "0.000000"
5 "0.499360" "0.000000" "0.500640" "0.000000" "0.000000"
```

All rows sum up to 1, but the strength of the relation is now similar for the distant units 4 and 5, and the proximate units 1, 2, 3.

2.3.2 Maximum eigenvalues normalization

Maximum eigenvalues normalization divides each non-zero weight by the overall maximum eigenvalue λ_{max}. Each element of $\mathbf{W}$ is divided by the same scalar parameter, which preserves the relations.

$$\frac{\mathbf{W}}{\lambda_{max}}$$

- Interpretation may become more complicated
- Keeps proportions of connectivity strengths across units (relevant esp. for distance based $\mathbf{W}$)

We use eigenvalue normalization for the inverse distance neighbours. We use `nb2listwdist()` to create weight inverse distance based weights and normalize in one step.

```
coords <- st_geometry(st_centroid(msoa.spdf))
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```
idw.lw <- nb2listwdist(dist_3.nb,
                       x = coords, # needed for idw
                       type = "idw", # inverse distance weighting
                       alpha = 1, # the decay parameter for distance weighting
                       style = "minmax") # for eigenvalue normalization
```

```
summary(idw.lw)
```

```
Characteristics of weights list object:
Neighbour list object:
Number of regions: 983
Number of nonzero links: 22086
Percentage nonzero weights: 2.285652
Average number of links: 22.46796
Link number distribution:

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 4  3  7 13 11 14 14 17 26 22 26 30 33 34 46 34 59 43 38 30 25 19 22 15 21 14
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
23 17 17 23 28 19 26 24 29 24 27 25 22 18  8 10 12  5  3  2  1
4 least connected regions:
158 160 463 959 with 1 link
1 most connected region:
545 with 47 links

Weights style: minmax
Weights constants summary:
         n     nn       S0       S1       S2
minmax 983 966289 463.6269 23.92505 1117.636
```

Examples from above: See how this keeps the proportions in our example. Instead of transforming values to sum up to 1 in each row, we now have much smaller values for 4 and 5 then we have for the proximate units 1, 2, 3.

```
sub.lw <- nb2listwdist(knn.nb,
                       x = sub.coords, # needed for idw
                       type = "idw", # inverse distance weighting
                       alpha = 1, # the decay parameter for distance weighting
                       style = "minmax") # for eigenvalue normalization
W_sub <- listw2mat(sub.lw)
formatC(W_sub, format = "f", digits = 6)
```

```
  [,1]       [,2]       [,3]       [,4]       [,5]
1 "0.000000" "0.245687" "0.429123" "0.000000" "0.000000"
2 "0.245687" "0.000000" "0.570877" "0.000000" "0.000000"
3 "0.429123" "0.570877" "0.000000" "0.000000" "0.000000"
```

```
4 "0.000000" "0.019663" "0.019047" "0.000000" "0.000000"
5 "0.029099" "0.000000" "0.029174" "0.000000" "0.000000"
```

2.4 Islands / missings

In practice, we often have a problem with islands. If we use contiguity based or distance based neighbour definitions, some units may end up with empty neighbours sets: they just do not touch any other unit and do not have a neighbour within a specific distance. This however creates a problem: what is the value in the neighbouring units?

The `zero.policy` option in `spdep` allows to proceed with empty neighbours sets. However, many further functions may run into problems and return errors. It often makes sense to either drop islands, to choose weights which always have neighbours (e.g. k nearest), or impute empty neighbours sets by using the nearest neighbours.

2.5 Global Autocorrelation

If spatially close observations are more likely to exhibit similar values, we cannot handle observations as if they were independent.

$$\mathrm{E}(\varepsilon_i \varepsilon_j) \neq \mathrm{E}(\varepsilon_i)\mathrm{E}(\varepsilon_j) = 0$$

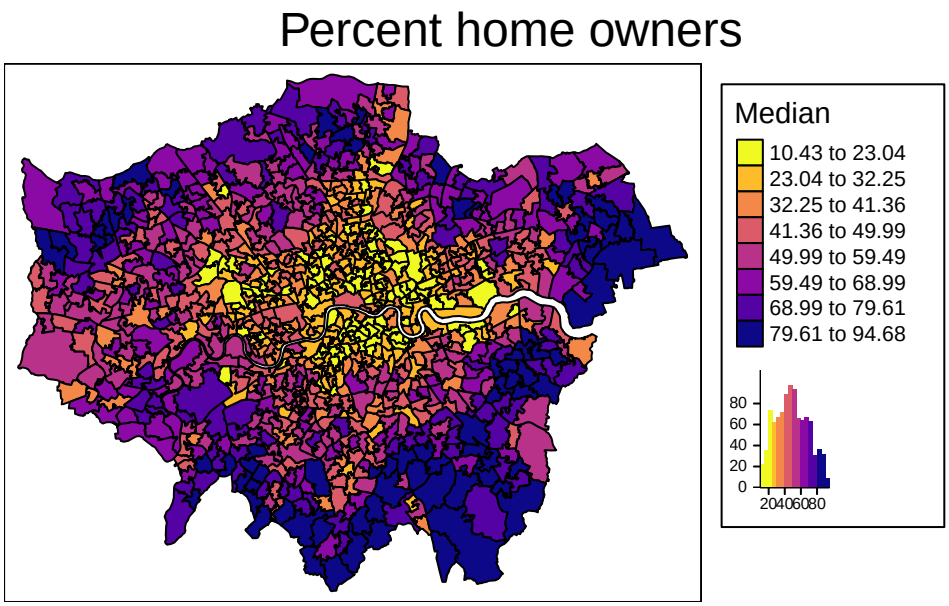
This violates a basic assumption of the conventional OLS model. We will talk more about whether that is good or bad (any guess?).

2.5.1 Visualization

There is one very easy and intuitive way of detecting spatial autocorrelation: Just look at the map. We do so by using `tmap` for plotting the share of home owners.

```
mp1 <- tm_shape(msoa.spdf) +
  tm_fill(col = "per_owner",
          #style = "cont",
          style = "fisher", n = 8,
          title = "Median",
          palette = viridis(n = 8, direction = -1, option = "C"),
          legend.hist = TRUE) +
  tm_borders(col = "black", lwd = 1) +
  tm_layout(legend.frame = TRUE, legend.bg.color = TRUE,
```

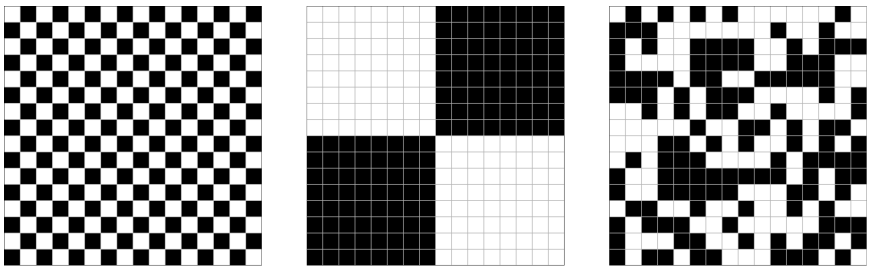
```
            #legend.position = c("right", "bottom"),
            legend.outside = TRUE,
            main.title = "Percent home owners",
            main.title.position = "center",
            title.snap.to.legend = TRUE)

mp1
```



We definitely see some clusters with spatial units having a low share of home owner (e.g. in the city center), and other clusters where home ownership is high (e.g. suburbs in the south and east, such as Bromley or Havering).

However, this is (to some degree) dependent on how we define cutoffs and coloring of the map: the Modifiable Areal Unit Problem (Wong 2009).

> **Question**
>
> Which of the following three checkerboards has no (or the lowest) autocorrelation?



a) b) c)

Would your answer be the same if we would aggregate the data to four larger areas / districts using the average within each of the four districts?

2.5.2 Moran's I

The most common and well known statistic for spatial dependence or autocorrelation is Moran's I, which goes back to Moran (1950) and Cliff and Ord (1972). For more extensive materials on Moran's I see for instance Kelejian and Piras (2017), Chapter 11.

To calculate Moran's I, we first define a neighbours weights matrix W.

Global Moran's I test statistic:

$$\mathbf{I} = \frac{N}{S_0} \frac{\sum_i \sum_j w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_i (y_i - \bar{y})^2}, \text{where } S_0 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}$$

It is often written with deviations z

$$\mathbf{I} = \frac{N}{S_0} \frac{\sum_i \sum_j w_{ij}(z_i)(z_j)}{\sum_i (z_i)^2}, \text{where } S_0 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}$$

Note that in the case of row-standardized weights, $S_0 = N$. The I can be interpreted as: *Relation of the deviation from the mean value between unit i and neighbours of unit i*. Basically, this measures correlation between neighbouring values.

- Negative values: negative autocorrelation
- Around zero: no autocorrelation
- Positive values: positive autocorrelation

To calculate Moran's I, we first need to define the relationship between units. As in the previous example, we define contiguity weights and distance-based weights.

```r
# Contiguity (Queens) neighbours weights
queens.nb <- poly2nb(msoa.spdf,
                     queen = TRUE,
                     snap = 1) # we consider points in 1m distance as 'touching'
queens.lw <- nb2listw(queens.nb,
                      style = "W")

# Neighbours within 3km distance
coords <- st_geometry(st_centroid(msoa.spdf))
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```r
dist_3.nb <- dnearneigh(coords,
                        d1 = 0, d2 = 3000)
idw.lw <- nb2listwdist(dist_3.nb,
                       x = coords, # needed for idw
                       type = "idw", # inverse distance weighting
                       alpha = 1, # the decay parameter for distance weighting
                       style = "minmax") # for eigenvalue normalization
```

Subsequently, we can calculate the average correlation between neighbouring units.

For contiguity weights, we get:

```r
# Global Morans I test of housing values based on contiguity weights
moran.test(msoa.spdf$per_owner, listw = queens.lw, alternative = "two.sided")
```

```
    Moran I test under randomisation

data:  msoa.spdf$per_owner
weights: queens.lw

Moran I statistic standard deviate = 38.161, p-value < 2.2e-16
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation          Variance
      0.728706855      -0.001018330       0.000365663
```

And for inverse distance weighting, we get:

```r
# Global Morans I test of housing values based on idw
moran.test(msoa.spdf$per_owner, listw = idw.lw, alternative = "two.sided")
```

```
    Moran I test under randomisation

data:  msoa.spdf$per_owner
weights: idw.lw

Moran I statistic standard deviate = 65.853, p-value < 2.2e-16
alternative hypothesis: two.sided
sample estimates:
Moran I statistic       Expectation          Variance
     0.6838957350     -0.0010183299      0.0001081719
```

Interpretation: In both cases, we have very strong autocorrelation between neighbouring/closer units (~.7). It barely matters which of the weights matrices we use. This autocorrelation is highly significant. we can thus reject the Null that units are independent of each other (at least at this spatial level and for the share of home owners).

2.5.3 Residual-based Moran's I

We can also use the same Moran's I test to inspect spatial autocorrelation in residuals from an estimated linear model.

Let's start with an intercept only model.

```r
lm0 <- lm(per_owner ~ 1, msoa.spdf)
lm.morantest(lm0, listw = queens.lw, alternative = "two.sided")
```

```
    Global Moran I for regression residuals

data:
model: lm(formula = per_owner ~ 1, data = msoa.spdf)
weights: queens.lw

Moran I statistic standard deviate = 38.177, p-value < 2.2e-16
alternative hypothesis: two.sided
sample estimates:
```

```
Observed Moran I      Expectation         Variance
    0.7287068548    -0.0010183299     0.0003653613
```

This is exactly what we have received in the general case of Moran's I.

Now, lets add some predictors. For instance, the distance to the city centre, and the population density may be strongly related to the home ownership rates and explain parts of the spatial dependence.

```
### Distance to city center
# Define centre
centre <- st_as_sf(data.frame(lon = -0.128120855701165,
                              lat = 51.50725909644806),
                   coords = c("lon", "lat"),
                   crs = 4326)
# Reproject
centre <- st_transform(centre, crs = st_crs(msoa.spdf))
# Calculate distance
msoa.spdf$dist_centre <- as.numeric(st_distance(msoa.spdf, centre)) / 1000
# hist(msoa.spdf$dist_centre)

### Run model with predictors
lm1 <- lm(per_owner ~ dist_centre + POPDEN, msoa.spdf)
lm.morantest(lm1, listw = queens.lw, alternative = "two.sided")
```

```
    Global Moran I for regression residuals

data:
model: lm(formula = per_owner ~ dist_centre + POPDEN, data = msoa.spdf)
weights: queens.lw

Moran I statistic standard deviate = 22.674, p-value < 2.2e-16
alternative hypothesis: two.sided
sample estimates:
Observed Moran I      Expectation         Variance
    0.4298146060    -0.0024065617     0.0003633607
```

There is still considerable auto-correlation in the residuals. However, we have reduce it by a substantial amount with two very simple control variables.

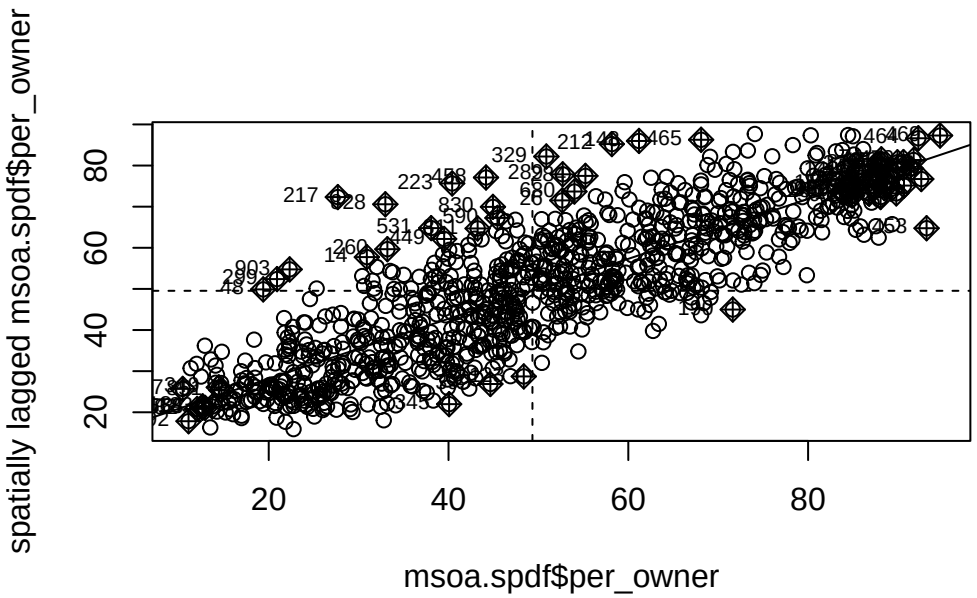
2.6 Local Autocorrelation

The Global Moran's I statistic above summarizes the spatial pattern by a single value. Although this is helpful to get a feeling of the strength of the general spatial association, it is often more helpful to inspect the spatial pattern in more detail.

The most prominent measure is the Local Indicators of Spatial Association (LISA) (Anselin 1995). LISA measures assess the importance and significance of a satistic at different spatial locations. For more information see for instance the GeoData Materials by Luc Anselin.

For instance, we can use the Moran Plot to identify how single (pairs of) units contribute to the overall dependence.

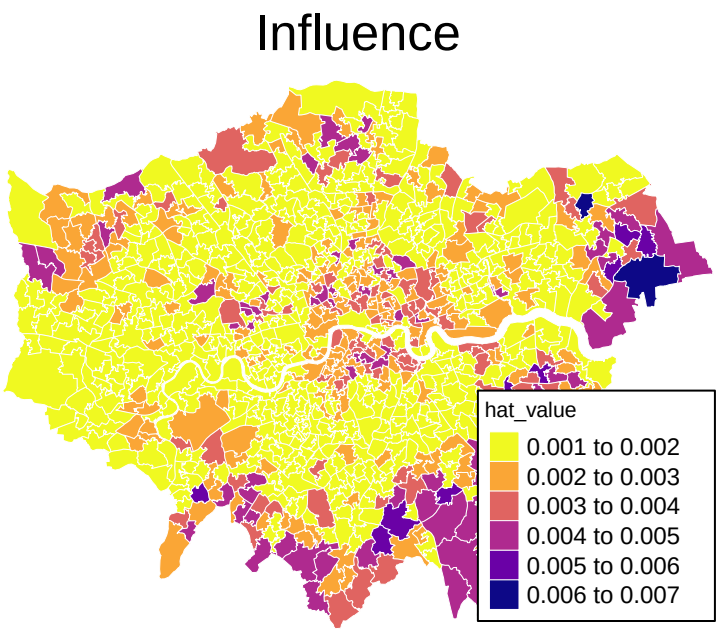
```
mp <- moran.plot(msoa.spdf$per_owner, queens.lw)
```



In the lower left corner, we see units with a low-low share of home ownership: focal and neighbouring units have a low share of home owners. In the top right corner, by contrast, we see high-high units.

And we can plot influence values on the Overall Moran statistic.

```
msoa.spdf$hat_value <- mp$hat
mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = "hat_value",
          palette = viridis(n = 10, direction = -1, option = "C"),
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("right", "bottom"),
            legend.outside = FALSE,
            main.title = "Influence",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8)

mp1
```



2.7 Local Moran's I

Local Moran's I is a local version of the overall Moran's I to identify local clusters and local spatial outliers (Anselin 1995). The Local Moran's I is just a local version which is calculated for each location:

$$\mathbf{I}_i = \frac{z_i \sum_j w_{ij} z_j}{\sum_i (z_i)^2/(n-1)}, \text{where}$$

We use the unfction `localmoran()` to calculate the local test statistic .

```
loci <- localmoran(msoa.spdf$per_owner, listw = queens.lw)
head(loci)
```

```
           Ii           E.Ii      Var.Ii       Z.Ii Pr(z != E(Ii))
1  0.42322928 -1.285364e-04 0.011367934  3.9706976   7.166249e-05
2 -0.12775982 -2.229957e-05 0.003634711 -2.1187688   3.411001e-02
3  0.38111534 -6.569549e-04 0.091630752  1.2611995   2.072370e-01
4  1.02874685 -1.428679e-03 0.279333375  1.9491704   5.127507e-02
5  0.08553291 -2.108521e-04 0.041275789  0.4220412   6.729949e-01
6 -0.24014505 -2.228818e-04 0.036321252 -1.2588964   2.080678e-01
```

It also has an attribute with the Moran plot quadrant of each observation.

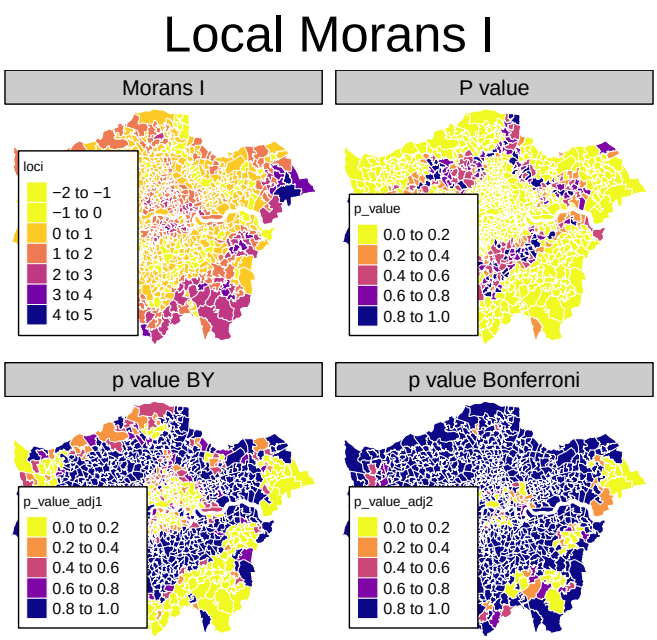
```
head(attr(loci, "quadr"))
```

```
       mean    median     pysal
1   Low-Low   Low-Low   Low-Low
2  Low-High  Low-High  Low-High
3 High-High High-High High-High
4 High-High High-High High-High
5 High-High High-High High-High
6  Low-High  Low-High  Low-High
```

This returns a data.frame with local moran statisic, the expectation of local moran statistic, its variance, and a p value for the satistical significance of each unit. Note that we obviously have a problem of multiple comparisons here and thus may want to correct the significance level, e.g. by Bonferroni adjustment (R. Bivand and Wong 2018).

```r
loci.df <- data.frame(loci)
names(loci.df) <- gsub("\\.", "", names(loci.df))
msoa.spdf$loci <- loci.df$Ii
msoa.spdf$p_value <- loci.df$PrzEIi
msoa.spdf$p_value_adj1 <- p.adjust(loci.df$PrzEIi, "BY")
msoa.spdf$p_value_adj2 <- p.adjust(loci.df$PrzEIi, "bonferroni")
```

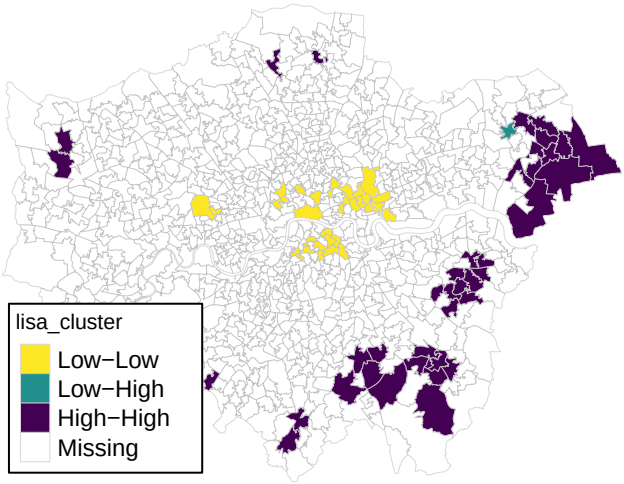
```r
mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = c("loci", "p_value", "p_value_adj1", "p_value_adj2"),
          palette = viridis(n = 10, direction = -1, option = "C"),
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("left", "bottom"),
            legend.outside = FALSE,
            main.title = "Local Morans I",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8,
            panel.labels = c("Morans I",
                             "P value",
                             "p value BY",
                           "p value Bonferroni"))

mp1
```



Something you can often see are so called LISA hotspot maps. They are based on the same idea as the moran plot, and show cluster of high-high and low-low values. We can use the hotspot function to identify the clusters, with a cutoff for singificance and the adjustment for multiple testing.

```
# Calculate clusters
msoa.spdf$lisa_cluster <- hotspot(loci,
                                  "Pr(z != E(Ii))",
                                  cutoff = 0.05,
                                  quadrant.type = "mean",
                                  p.adjust = "BY")

# Map
mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = c("lisa_cluster"),
          palette = viridis(n = 3, direction = -1, option = "D"),
          colorNA = "white") +
  tm_borders(col = "grey70", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("left", "bottom"),
            legend.outside = FALSE,
```

```
            main.title = "Home Ownership \n LISA Clusters p(BY) < 0.05",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8,)

mp1
```



Note that it is not suggested to interpret those cluster as significant in the strict statistical sense. Pebesma and Bivand (2023) suggest to speak of *interesting clusters*. After all, this is an explorative approach. Nevertheless, it can help to identify spatial patterns and clusters.

There are more ways of calculating these hotspot maps and more choices on the cutoffs and calculation of the statistical significance. For more materials see Chapter 15 of Pebesma and Bivand (2023).

2.8 Example

Tate.2021

This study explores the geography of flood exposure and social vulnerability in the conterminous United States based on spatial analysis of fluvial and pluvial flood extent, land cover, and social vulnerability.

Mobile homes and racial minorities are most overrepresented in hotspots compared to elsewhere. The results identify priority locations where interventions can mitigate both physical and social aspects of flood vulnerability.

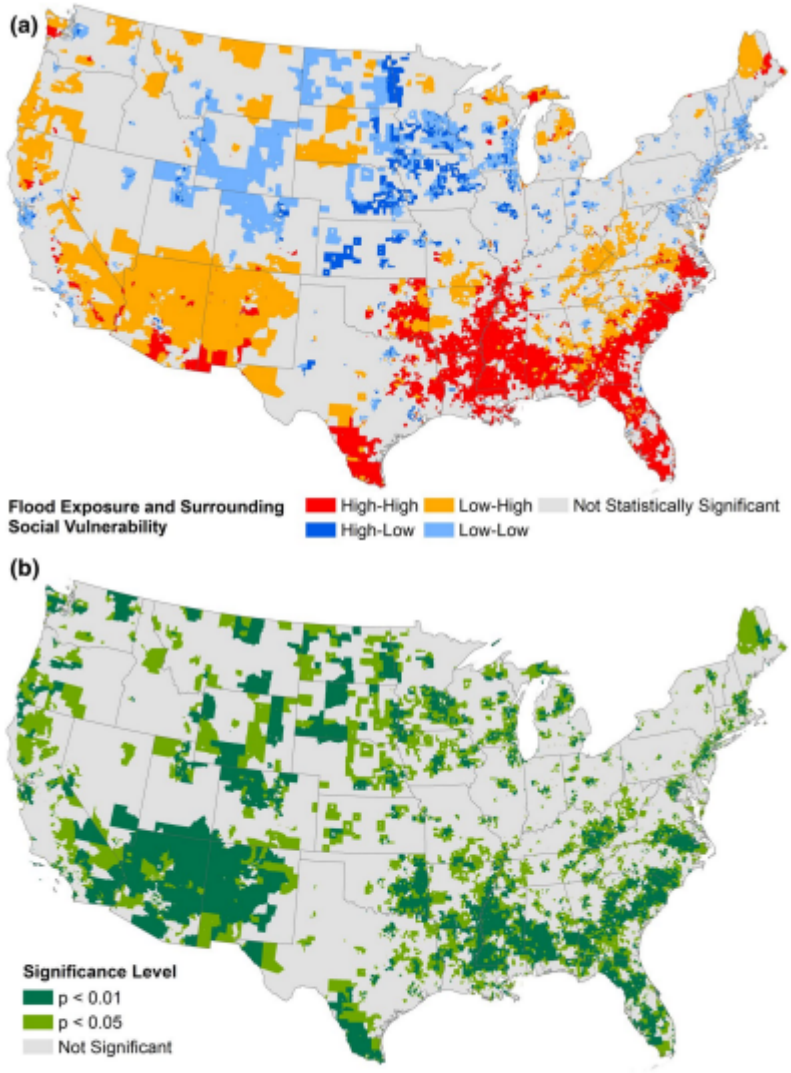


Fig. 4 Bivariate LISA of 100-year flood exposure and surrounding social vulnerability. **a** Cluster map and **b** Cluster significance

3 Exercise I

Required packages

```
pkgs <- c("sf", "mapview", "spdep", "spatialreg", "tmap", "viridisLite") # note: load spde
lapply(pkgs, require, character.only = TRUE)
```

Session info

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default


locale:
[1] LC_COLLATE=English_United Kingdom.utf8
[2] LC_CTYPE=English_United Kingdom.utf8
[3] LC_MONETARY=English_United Kingdom.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.utf8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] viridisLite_0.4.2 tmap_3.3-3        spatialreg_1.2-9  Matrix_1.5-4.1
[5] spdep_1.2-8       spData_2.3.0      mapview_2.11.0    sf_1.0-13
```

```
loaded via a namespace (and not attached):
 [1] xfun_0.39          raster_3.6-23      htmlwidgets_1.6.2  lattice_0.21-8
 [5] vctrs_0.6.3        tools_4.3.1        crosstalk_1.2.0    LearnBayes_2.15.1
 [9] generics_0.1.3     parallel_4.3.1     sandwich_3.0-2     stats4_4.3.1
[13] tibble_3.2.1       proxy_0.4-27       fansi_1.0.4        pkgconfig_2.0.3
[17] KernSmooth_2.23-21 satellite_1.0.4    RColorBrewer_1.1-3 leaflet_2.1.2
[21] webshot_0.5.5      lifecycle_1.0.3    compiler_4.3.1     deldir_1.0-9
[25] munsell_0.5.0      terra_1.7-39       leafsync_0.1.0     codetools_0.2-
19
[29] stars_0.6-1        htmltools_0.5.5    class_7.3-22       pillar_1.9.0
[33] MASS_7.3-60        classInt_0.4-9     lwgeom_0.2-13      wk_0.7.3
[37] abind_1.4-5        boot_1.3-28.1      multcomp_1.4-25    nlme_3.1-162
[41] tidyselect_1.2.0   digest_0.6.32      mvtnorm_1.2-2      dplyr_1.1.2
[45] splines_4.3.1      fastmap_1.1.1      grid_4.3.1         colorspace_2.1-
0
[49] expm_0.999-7       cli_3.6.1          magrittr_2.0.3     base64enc_0.1-3
[53] dichromat_2.0-0.1  XML_3.99-0.14      survival_3.5-5     utf8_1.2.3
[57] TH.data_1.1-2      leafem_0.2.0       e1071_1.7-13       scales_1.2.1
[61] sp_1.6-1           rmarkdown_2.23     zoo_1.8-12         png_0.1-8
[65] coda_0.19-4        evaluate_0.21      knitr_1.43         tmaptools_3.1-1
[69] s2_1.1.4           rlang_1.1.1        Rcpp_1.0.10        glue_1.6.2
[73] DBI_1.1.3          rstudioapi_0.14    jsonlite_1.8.5     R6_2.5.1
[77] units_0.8-2
```

Reload data from pervious session

```
load("_data/msoa2_spatial.RData")
```

3.1 General Exercises

1. Please calculate a neighbours weights matrix of the nearest 10 neighbours (see `spdep::knearneigh()`), and create a listw object using row normalization.

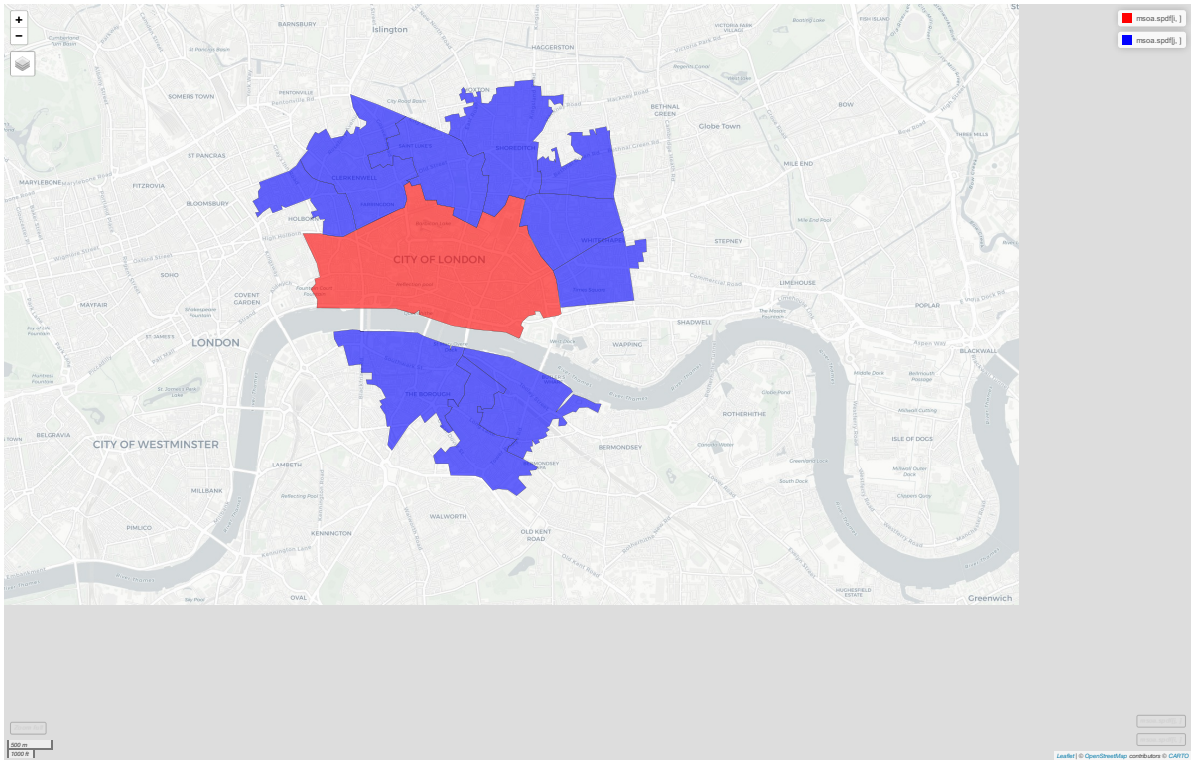
```
coords <- st_centroid(msoa.spdf)
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```
k10.nb <- knearneigh(coords, k = 10)
```

2. Can you create a map containing the City of London (MSOA11CD = "E02000001") and its ten nearest neighbours?

```
i <- which(msoa.spdf$MSOA11CD == "E02000001")

# Extract neigbours
j <- k10.nb$nn[i,]

mapview(list(msoa.spdf[i,], msoa.spdf[j,]), col.regions = c("red", "blue"))
```



3. Chose another characteristics from the data (e.g. ethnic groups or house prices) and calculate global Moran's I for it.

```
# Gen nb object
k10.nb <- knn2nb(k10.nb)

# Gen listw object
```

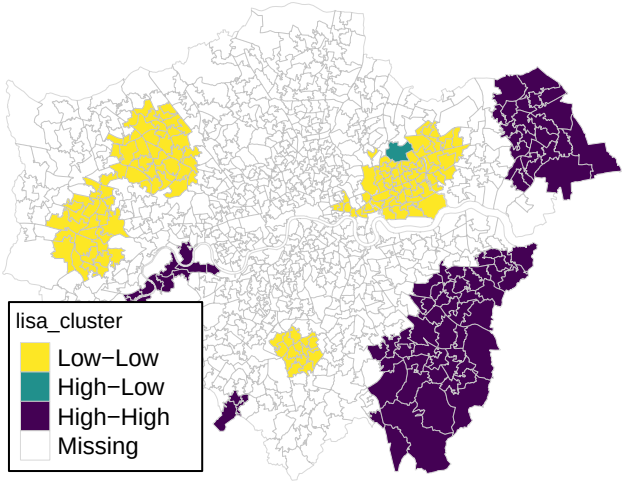
```r
k10.listw <- nb2listw(k10.nb, style = "W")

# MOran test
moran.test(msoa.spdf$per_white, listw = k10.listw)
```

```
    Moran I test under randomisation

data:  msoa.spdf$per_white
weights: k10.listw

Moran I statistic standard deviate = 55.733, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.7623842505     -0.0010183299      0.0001876235
```

4. Produce a LISA cluster map for the characteristic you have chosen.

```r
loci2 <- localmoran(msoa.spdf$per_white, listw = k10.listw)

# Calculate clusters
msoa.spdf$lisa_cluster <- hotspot(loci2,
                                  "Pr(z != E(Ii))",
                                  cutoff = 0.05,
                                  quadrant.type = "mean",
                                  p.adjust = "BY")

# Map
mp1 <-  tm_shape(msoa.spdf) +
  tm_fill(col = c("lisa_cluster"),
          palette = viridis(n = 3, direction = -1, option = "D"),
          colorNA = "white") +
  tm_borders(col = "grey70", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("left", "bottom"),
            legend.outside = FALSE,
            main.title = "Percentage White \n LISA Clusters p(BY) < 0.05",
            main.title.position = "center",
            main.title.size = 1.6,
```

```
            legend.title.size = 0.8,
            legend.text.size = 0.8,)

mp1
```



3.2 Environmental inequality

How would you investigate the following descriptive research question: Are ethnic (and immigrant) minorities in London exposed to higher levels of pollution? Also consider the spatial structure. What's your dependent and what's your independent variable?

1) Define a neigbours weights object of your choice

You can choose any preferred neighbours-weights definition, such as for instance contiguity neighbours or all neighbourhoods within 2.5km.

```
coords <- st_centroid(msoa.spdf)
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```r
# Neighbours within 3km distance
dist_15.nb <- dnearneigh(coords, d1 = 0, d2 = 2500)

summary(dist_15.nb)
```

```
Neighbour list object:
Number of regions: 983
Number of nonzero links: 15266
Percentage nonzero weights: 1.579859
Average number of links: 15.53001
4 regions with no links:
158 463 478 505
Link number distribution:

 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 4  5  9 23 19 26 36 31 53 39 61 63 59 48 42 35 24 31 28 30 27 26 25 19 38 29
26 27 28 29 30 31 32 33 34
32 38 26 16 20 10  8  1  2
5 least connected regions:
160 469 474 597 959 with 1 link
2 most connected regions:
565 567 with 34 links
```

```r
# There are some mpty one. Lets impute with the nearest neighbour
k2.nb <- knearneigh(coords, k = 1)

# Replace zero
nolink_ids <- which(card(dist_15.nb) == 0)
dist_15.nb[card(dist_15.nb) == 0] <- k2.nb$nn[nolink_ids, ]

summary(dist_15.nb)
```

```
Neighbour list object:
Number of regions: 983
Number of nonzero links: 15270
Percentage nonzero weights: 1.580273
Average number of links: 15.53408
Link number distribution:

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

```
 9  9 23 19 26 36 31 53 39 61 63 59 48 42 35 24 31 28 30 27 26 25 19 38 29 32
27 28 29 30 31 32 33 34
38 26 16 20 10  8  1  2
9 least connected regions:
158 160 463 469 474 478 505 597 959 with 1 link
2 most connected regions:
565 567 with 34 links
```

```r
# listw object with row-normalization
dist_15.lw <- nb2listw(dist_15.nb, style = "W")
```

2) Estimate the extent of spatial auto-correlation

The most common way would be to calculate Global Moran's I.

```r
moran.test(msoa.spdf$no2, listw = dist_15.lw)
```

```
    Moran I test under randomisation

data:  msoa.spdf$no2
weights: dist_15.lw

Moran I statistic standard deviate = 65.197, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.891520698      -0.001018330       0.000187411
```

4 Spatial Regression Models

Required packages

```
pkgs <- c("sf", "mapview", "spdep", "spatialreg", "tmap", "viridisLite") # note: load spde
lapply(pkgs, require, character.only = TRUE)
```

Session info

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default


locale:
[1] LC_COLLATE=English_United Kingdom.utf8
[2] LC_CTYPE=English_United Kingdom.utf8
[3] LC_MONETARY=English_United Kingdom.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.utf8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] viridisLite_0.4.2 tmap_3.3-3        spatialreg_1.2-9  Matrix_1.5-4.1
[5] spdep_1.2-8       spData_2.3.0      mapview_2.11.0    sf_1.0-13
```

```
loaded via a namespace (and not attached):
 [1] xfun_0.39          raster_3.6-23      htmlwidgets_1.6.2  lattice_0.21-8
 [5] vctrs_0.6.3        tools_4.3.1        crosstalk_1.2.0    LearnBayes_2.15.1
 [9] generics_0.1.3     parallel_4.3.1     sandwich_3.0-2     stats4_4.3.1
[13] tibble_3.2.1       proxy_0.4-27       fansi_1.0.4        pkgconfig_2.0.3
[17] KernSmooth_2.23-21 satellite_1.0.4    RColorBrewer_1.1-3 leaflet_2.1.2
[21] webshot_0.5.5      lifecycle_1.0.3    compiler_4.3.1     deldir_1.0-9
[25] munsell_0.5.0      terra_1.7-39       leafsync_0.1.0     codetools_0.2-
19
[29] stars_0.6-1        htmltools_0.5.5    class_7.3-22       pillar_1.9.0
[33] MASS_7.3-60        classInt_0.4-9     lwgeom_0.2-13      wk_0.7.3
[37] abind_1.4-5        boot_1.3-28.1      multcomp_1.4-25    nlme_3.1-162
[41] tidyselect_1.2.0   digest_0.6.32      mvtnorm_1.2-2      dplyr_1.1.2
[45] splines_4.3.1      fastmap_1.1.1      grid_4.3.1         colorspace_2.1-
0
[49] expm_0.999-7       cli_3.6.1          magrittr_2.0.3     base64enc_0.1-3
[53] dichromat_2.0-0.1  XML_3.99-0.14      survival_3.5-5     utf8_1.2.3
[57] TH.data_1.1-2      leafem_0.2.0       e1071_1.7-13       scales_1.2.1
[61] sp_1.6-1           rmarkdown_2.23     zoo_1.8-12         png_0.1-8
[65] coda_0.19-4        evaluate_0.21      knitr_1.43         tmaptools_3.1-1
[69] s2_1.1.4           rlang_1.1.1        Rcpp_1.0.10        glue_1.6.2
[73] DBI_1.1.3          rstudioapi_0.14    jsonlite_1.8.5     R6_2.5.1
[77] units_0.8-2
```

Reload data from pervious session

```
load("_data/msoa2_spatial.RData")
```

There are various techniques to model spatial dependence and spatial processes (LeSage and Pace 2009). Here, we will just cover a few of the most common techniques / econometric models. One advantage of the most basic spatial model (SLX) is that this method can easily be incorporated in a variety of other methodologies, such as machine learning approaches.

For more in-depth materials see LeSage and Pace (2009) and Kelejian and Piras (2017). Franzese and Hays (2007), Halleck Vega and Elhorst (2015), LeSage (2014), Rüttenauer (2022), Rüttenauer (2024) and Wimpy, Whitten, and Williams (2021) provide article-length introductions.

4.1 Why do we need spatial regression models

4.1.1 Non-spatial OLS

Let us start with a linear model, where $\mathbf{y}$ is the outcome or dependent variable $(N \times 1)$, $\mathbf{X}$ are various exogenous covariates $(N \times k)$, and $(N \times 1)$ is the error term. We are usually interested in the coefficient vector $(k \times 1)$ and its insecurity estimates.

$$\mathbf{y} = \mathbf{X} +$$

The work-horse for estimating in the social science is the OLS estimator (Wooldridge 2010).

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

! OLS assumptions I

1. $\mathrm{E}(\epsilon_i | \mathbf{X}_i) = 0$: for every value of X, the average / expectation of the error term equals zero – put differently: the error term is independent of X,
2. the observations of the sample are independent and identically distributed (i.i.d),
3. the fourth moments of the variables $\mathbf{X}_i$ and Y_i are positive and definite – put differently: extreme values / outliers are very very rare,
4. $\mathrm{rank}(\mathbf{X}) = K$: the matrix $\mathbf{X}$ has full rank – put differently: no perfect multicollinearity between the covariates,

! OLS assumptions II

5. $\mathrm{Var}(\varepsilon | x) = \sigma^2$: the error terms ε are homoskedastic / have the same variance given any value of the explanatory variable,
6. $\varepsilon \sim \mathcal{N}(0, \sigma^2)$: the error terms ε are normally distributed (conditional on the explanatory variables X_i).

Question

Which of the six assumptions above may be violated by spatial dependence?

4.1.2 Problem of ignoring spatial dependence

Does spatial dependence influence the results / coefficient estimates of non-spatial regression models, or in other words: is ignoring spatial dependence harmful?

I've heard different answers, ranging from "It only affects the standard errors" to "it always introduces bias". As so often, the true (or best?) answer is somewhere in the middle: *it depends* (Betz, Cook, and Hollenbach 2020; Cook, Hays, and Franzese 2020; Pace and LeSage 2010; Rüttenauer 2022).

The easiest way to think of it is analogous to the omit variable bias (Betz, Cook, and Hollenbach 2020; Cook, Hays, and Franzese 2020):

$$plim\ \hat{\beta}_{OLS} = \beta + \gamma \frac{\text{Cov}(\mathbf{x}, \mathbf{z})}{\text{Var}(\mathbf{x})},$$

where z is some omit variable, and γ is the conditional effect of $\mathbf{z}$ on $\mathbf{y}$. Now imagine that the neighbouring values of the dependent variable $\mathbf{Wy}$ are autocorrelated to focal unit which we denote with $\rho > 0$, and that the covariance between the focal unit's exogenous covariates and $\mathbf{Wy}$ is not zero. Then we will have an omitted variable bias due to spatial dependence:

$$plim\ \hat{\beta}_{OLS} = \beta + \rho \frac{\text{Cov}(\mathbf{x}, \mathbf{Wy})}{\text{Var}(\mathbf{x})} \neq \beta,$$

For completeness, the entire bias is a bit more complicated (Pace and LeSage 2010; Rüttenauer 2022) and looks like:

$$plim\ \hat{\beta} = \frac{\sum_{ij}(\mathbf{M}(\delta)\mathbf{M}(\delta)^\top \circ \mathbf{M}(\rho))_{ij}}{\text{tr}(\mathbf{M}(\delta)\mathbf{M}(\delta)^\top)}\beta + \frac{\sum_{ij}(\mathbf{M}(\delta)\mathbf{M}(\delta)^\top \circ \mathbf{M}(\rho)\mathbf{W})_{ij}}{\text{tr}(\mathbf{M}(\delta)\mathbf{M}(\delta)^\top)}\theta,$$

where $\circ$ denotes the Hadamard product, $\mathbf{M}(\delta) = (\mathbf{I}_N - \delta\mathbf{W})^{-1}$, and $\mathbf{M}(\rho) = (\mathbf{I}_N - \rho\mathbf{W})^{-1}$.

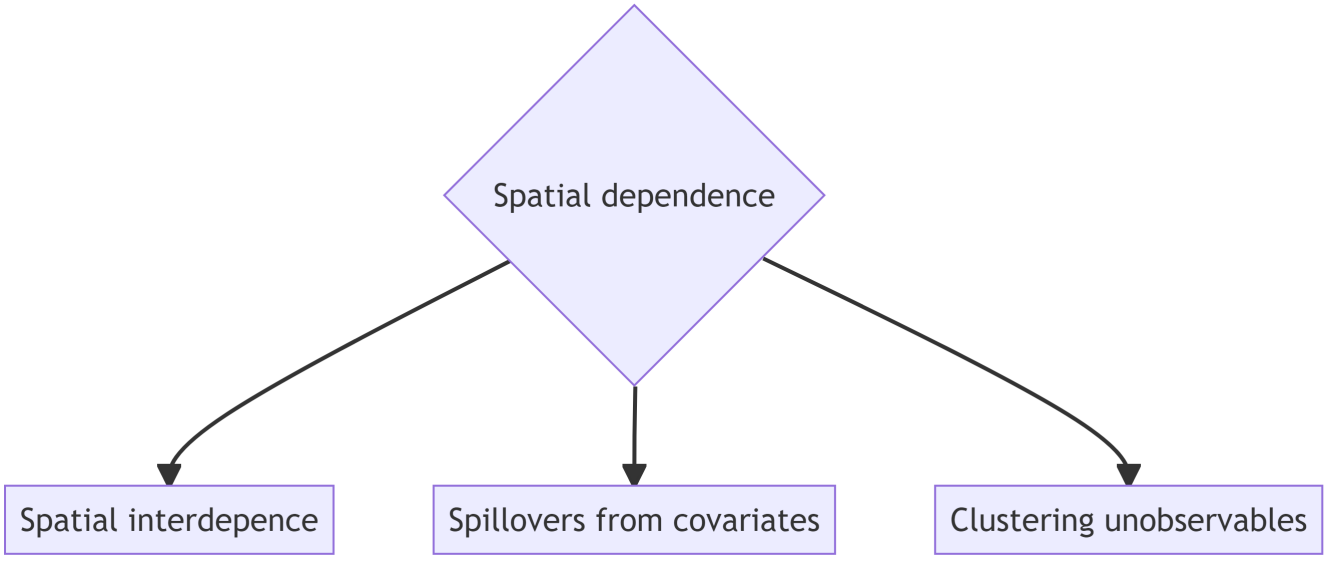
(Don't worry, no need to learn by hard!!)

Essentially, the non-spatial OLS estimator β_{OLS} is biased in the presence of either (Pace and LeSage 2010; Rüttenauer 2022):

- Spatial autocorrelation in the dependent variable ($\rho \neq 0$) and spatial autocorrelation in the covariate ($\delta \neq 0$). This bias increases with ρ, δ, and β.
- Local spatial spillover effects ($\theta \neq 0$) and spatial autocorrelation in the covariate ($\delta \neq 0$). This is analogous to the omitted variable bias resulting from the omission of $\mathbf{Wx}$. It increases with θ and δ, but additionally with ρ if $\theta \neq 0$ and $\delta \neq 0$.
- An omitted variable and $\text{E}(|\mathbf{x}) \neq 0$. This non-spatial omitted variable bias γ is amplified by spatial dependence in the disturbances (λ) and spatial autocorrelation in the dependent variable (ρ), but also increases with positive values of δ if either $\rho \neq 0$ or $\lambda \neq 0$. Obviously, it also increases with γ.

4.2 Spatial Regression Models

Broadly, spatial dependence or clustering in some characteristic can be the result of three different processes:



Strictly speaking, there are some other possibilities too, such as measurement error or the wrong choice on the spatial level. For instance, imagine we have a city-specific characteristic (e.g. public spending) allocated to neighbourhood units. Obviously, this will introduce heavy autocorrelation on the neigbourhood level by construction.

There are three basic ways of incorporating spatial dependence, which then can be further combined. As before, the $N \times N$ spatial weights matrix $\mathbf{W}$ defines the spatial relationship between units.

4.2.1 Spatial Error Model (SEM)

- Clustering on Unobservables

$$\mathbf{y} = \alpha + \mathbf{X} + \mathbf{u},$$
$$\mathbf{u} = \lambda \mathbf{W}\mathbf{u} +$$

λ denotes the strength of the spatial correlation in the errors of the model: *your errors influence my errors*.

- > 0: positive error dependence,
- < 0: negative error dependence,
- $= 0$: traditional OLS model.

λ is defined in the range $[-1, +1]$.

4.2.2 Spatial Autoregressive Model (SAR)

- Interdependence

$$\mathbf{y} = \alpha + \rho \mathbf{W}\mathbf{y} + \mathbf{X} +$$

ρ denotes the strength of the spatial correlation in the dependent variable (spatial autocorrelation): *your outcome influences my outcome*.

- > 0: positive spatial dependence,
- < 0: negative spatial dependence,
- $= 0$: traditional OLS model.

ρ is defined in the range $[-1, +1]$.

4.2.3 Spatially lagged X Model (SLX)

- Spillovers in Covariates

$$\mathbf{y} = \alpha + \mathbf{X} + \mathbf{WX} +$$

θ denotes the strength of the spatial spillover effects from covariate(s) on the dependent variable: *your covariates influence my outcome.*

θ is basically like any other coefficient from a covariate. It is thus not bound to any range.

Moreover, there are models combining two sets of the above specifications.

4.2.4 Spatial Durbin Model (SDM)

- Interdependence
- Spillovers in Covariates

$$\mathbf{y} = \alpha + \rho\mathbf{Wy} + \mathbf{X} + \mathbf{WX} +$$

4.2.5 Spatial Durbin Error Model (SDEM)

- Clustering on Unobservables
- Spillovers in Covariates

$$\begin{aligned}\mathbf{y} &= \alpha + \mathbf{X} + \mathbf{WX} + \mathbf{u},\\ \mathbf{u} &= \lambda\mathbf{Wu} +\end{aligned}$$

4.2.6 Combined Spatial Autocorrelation Model (SAC)

- Clustering on Unobservables
- Interdependence

$$\begin{aligned}\mathbf{y} &= \alpha + \rho\mathbf{Wy} + \mathbf{X} + \mathbf{u},\\ \mathbf{u} &= \lambda\mathbf{Wu} +\end{aligned}$$

4.2.7 General Nesting Spatial Model (GNS)

- Clustering on Unobservables
- Interdependence
- Spillovers in Covariates

$$\mathbf{y} = \alpha + \rho\mathbf{W}\mathbf{y} + \mathbf{X} + \mathbf{W}\mathbf{X} + \mathbf{u},$$
$$\mathbf{u} = \lambda\mathbf{W}\mathbf{u} +$$

Manski's reflection problem

The General Nesting Spatial Model (GNS) is only weakly (or not?) identifiable (Gibbons and Overman 2012).
It's analogous to Manski's reflection problem on neighbourhood effects (Manski 1993): If people in the same group behave similar, this can be because a) imitating behaviour of the group, b) exogenous characteristics of the group influence the behaviour, and c) members of the same group are exposed to the same external circumstances. *We just cannot separate those in observational data.*

Note that all of these models assume different data generating processes (DGP) leading to the spatial pattern. Although there are specifications tests, it is generally not possible to let the data decide which one is the true underlying DGP (Cook, Hays, and Franzese 2020; Rüttenauer 2022). However, there might be theoretical reasons to guide the model specification (Cook, Hays, and Franzese 2020).

Just because SAR is probably the most commonly used model does not make it the best choice. In contrast, various studies (Halleck Vega and Elhorst 2015; Rüttenauer 2022; Wimpy, Whitten, and Williams 2021) highlight the advantages of the relative simple SLX model. Moreover, this specification can basically be incorporated in any other statistical method.

4.2.8 A note on missings

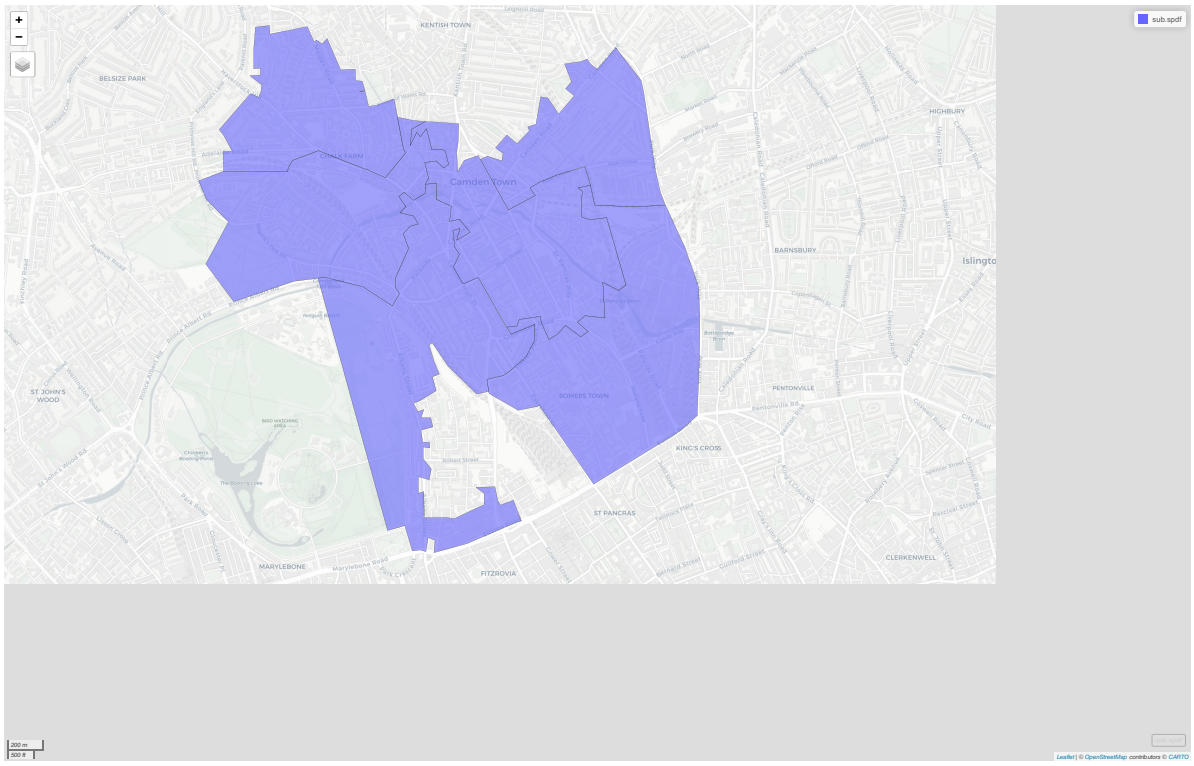
Missing values create a problem in spatial data analysis. For instance, in a local spillover model with an average of 10 neighbours, two initial missing values will lead to 20 missing values in the spatially lagged variable. For global spillover models, one initial missing will 'flow' through the neighbourhood system until the cutoff point (and create an excess amount of missings).

Depending on the data, units with missings can either be dropped and omitted from the initial weights creation, or we need to impute the data first, e.g. using interpolation or Kriging.

4.3 Mini Example

Let's try to make sense of this. We rely on a mini example using a few units in Camden

```
sub.spdf <- msoa.spdf[c(172, 175, 178, 179, 181, 182), ]
mapview(sub.spdf)
```



We then construct queens neighbours, and have a look at the resulting non-normalized matrix $\mathbf{W}$.

```
queens.nb <- poly2nb(sub.spdf, queen = TRUE, snap = 1)
W <- nb2mat(queens.nb, style = "B")
W
```

```
    [,1] [,2] [,3] [,4] [,5] [,6]
172    0    0    1    0    0    0
175    0    0    0    1    0    1
178    1    0    0    1    1    0
179    0    1    1    0    1    1
```

```
181    0    0    1    1    0    1
182    0    1    0    1    1    0
attr(,"call")
nb2mat(neighbours = queens.nb, style = "B")
```

We have selected 6 units. So, $\mathbf{W}$ is a 6×6 matrix. we see that observation 1 has one neighbour: observation 3. Observation 2 has two neighbours: observation 4 and observation 6. The diagonal is zero: no unit is a neighbour of themselves.

No we row-normalize this matrix.

```
queens.lw <- nb2listw(queens.nb,
                      style = "W")
W_rn <- listw2mat(queens.lw)
W_rn
```

```
        [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
172 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000
175 0.0000000 0.0000000 0.0000000 0.5000000 0.0000000 0.5000000
178 0.3333333 0.0000000 0.0000000 0.3333333 0.3333333 0.0000000
179 0.0000000 0.2500000 0.2500000 0.0000000 0.2500000 0.2500000
181 0.0000000 0.0000000 0.3333333 0.3333333 0.0000000 0.3333333
182 0.0000000 0.3333333 0.0000000 0.3333333 0.3333333 0.0000000
```

No every single weight w_{ij} is divided by the total number of neighbours n_i of the focal unit. For observation 1, observation 3 is the only neighbour, thus a weight = 1. For observation two, both neighbours have a weight of 1/2. For observation 3 (with three neighbours) each neighbour got a weight of 1/3.

> Question
>
> What happens if we multiply this matrix $\mathbf{W}$ with a $N \times 1$ vector $\mathbf{y}$ or $\mathbf{x}$?

A short reminder on matrix multiplication.

$$\mathbf{W} * \mathbf{y} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} * \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \end{bmatrix} = \begin{bmatrix} w_{11}y_{11} + w_{12}y_{21} + w_{13}y_{31} \\ w_{21}y_{11} + w_{22}y_{21} + w_{23}y_{31} \\ w_{31}y_{11} + w_{32}y_{21} + w_{33}y_{31} \end{bmatrix}$$

Each line of $\mathbf{W} * \mathbf{y}$ just gives a weighted average of the other y-values y_j in the sample. In case of the row-normalization, each neighbour gets the same weight $\frac{1}{n_i}$. This is simply the mean of y_j of the neighbours in case of a row-normalized contiguity weights matrix.

Note that the *mean* interpretation is only valid with row-normalization. What would we get with inverse-distance based weights?

Let's look at this in our example

```
y <- sub.spdf$med_house_price
x <- sub.spdf$pubs_count

W_rn
```

```
        [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
172 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000
175 0.0000000 0.0000000 0.0000000 0.5000000 0.0000000 0.5000000
178 0.3333333 0.0000000 0.0000000 0.3333333 0.3333333 0.0000000
179 0.0000000 0.2500000 0.2500000 0.0000000 0.2500000 0.2500000
181 0.0000000 0.0000000 0.3333333 0.3333333 0.0000000 0.3333333
182 0.0000000 0.3333333 0.0000000 0.3333333 0.3333333 0.0000000
```

```
y
```

```
[1] 376812.5 414625.0 713125.0 322750.0 495000.0 364000.0
```

```
x
```

```
[1] 1 3 3 1 9 7
```

```
W_rn_y <- W_rn %*% y
W_rn_x <- W_rn %*% x
W_rn_y
```

```
        [,1]
172 713125.0
175 343375.0
178 398187.5
179 496687.5
181 466625.0
182 410791.7
```

```
W_rn_x
```

```
       [,1]
172 3.000000
175 4.000000
178 3.666667
179 5.500000
181 3.666667
182 4.333333
```

Let's check if our interpretation is true

```
W_rn_y[1] == y[3]
```

```
[1] TRUE
```

```
W_rn_y[2] == mean(y[c(4, 6)])
```

```
[1] TRUE
```

```
W_rn_y[4] == mean(y[c(2, 3, 5, 6)])
```

```
[1] TRUE
```

4.4 Real Example

First, we need the a spatial weights matrix.

```
# Contiguity (Queens) neighbours weights
queens.nb <- poly2nb(msoa.spdf,
                     queen = TRUE,
                     snap = 1) # we consider points in 1m distance as 'touching'
queens.lw <- nb2listw(queens.nb,
                      style = "W")
```

We can estimate spatial models using `spatialreg`.

4.4.1 SAR

Lets estimate a spatial SAR model using the `lagsarlm()` with contiguity weights. We use median house value as depended variable, and include population density (`POPDEN`), the air pollution (`no2`), and the share of ethnic minorities (`per_mixed`, `per_asian`, `per_black`, `per_other`).

```
mod_1.sar <- lagsarlm(log(med_house_price) ~ log(no2) + log(POPDEN) +
                        per_mixed + per_asian + per_black + per_other,
                      data = msoa.spdf,
                      listw = queens.lw,
                      Durbin = FALSE) # we could here extend to SDM
summary(mod_1.sar)
```

```
Call:lagsarlm(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other, data = msoa.spdf,
    listw = queens.lw, Durbin = FALSE)

Residuals:
       Min         1Q     Median         3Q        Max
-0.5281789 -0.1220524 -0.0099245  0.0992203  1.0936745

Type: lag
Coefficients: (asymptotic standard errors)
               Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)  3.17383180  0.29041604  10.9286 < 2.2e-16
log(no2)     0.39705423  0.04452880   8.9168 < 2.2e-16
log(POPDEN) -0.05583014  0.01242876  -4.4920 7.055e-06
per_mixed    0.01851577  0.00579832   3.1933  0.001407
per_asian   -0.00228346  0.00045876  -4.9775 6.442e-07
per_black   -0.01263650  0.00100282 -12.6009 < 2.2e-16
per_other   -0.00161419  0.00289082  -0.5584  0.576582

Rho: 0.66976, LR test value: 473.23, p-value: < 2.22e-16
Asymptotic standard error: 0.025311
    z-value: 26.461, p-value: < 2.22e-16
Wald statistic: 700.19, p-value: < 2.22e-16

Log likelihood: 196.7203 for lag model
ML residual variance (sigma squared): 0.035402, (sigma: 0.18815)
Number of observations: 983
```

```
Number of parameters estimated: 9
AIC: -375.44, (AIC for lm: 95.786)
LM test for residual autocorrelation
test value: 8.609, p-value: 0.0033451
```

This looks pretty much like a conventional model output, with some additional information: a highly significant `mod_1.sar$rho` of 0.67 indicates strong positive spatial autocorrelation.

Remember that is the coefficient for the term $\mathbf{y} = \rho\mathbf{W}\mathbf{y}$.... It is bound to be below 1 for positive autocorrelation.

In substantive terms, house prices in the focal unit positively influence house prices in neighbouring units, which again influences house prices among the neighbours of these neighbours, and so on (we'll get back to this).

> **Warning**
>
> The coefficients of covariates in a SAR model are not marginal or partial effects, because of the spillovers and feedback loops in $\mathbf{y}$ (see below)!
> From the coefficient, we can only interpret the direction: there's a positive effect of air pollution and a negative effect of population sensitivity, and so on...

4.4.2 SEM

SEM models can be estimated using `errorsarlm()`.

```
mod_1.sem <- errorsarlm(log(med_house_price) ~ log(no2) + log(POPDEN) +
                          per_mixed + per_asian + per_black + per_other,
                        data = msoa.spdf,
                        listw = queens.lw,
                        Durbin = FALSE) # we could here extend to SDEM
summary(mod_1.sem)
```

```
Call:errorsarlm(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other, data = msoa.spdf,
    listw = queens.lw, Durbin = FALSE)

Residuals:
      Min        1Q    Median        3Q       Max
-0.581785 -0.105218 -0.012758  0.094430  0.913425
```

```
Type: error
Coefficients: (asymptotic standard errors)
               Estimate  Std. Error  z value  Pr(>|z|)
(Intercept) 12.92801104  0.35239139  36.6865 < 2.2e-16
log(no2)     0.15735296  0.10880727   1.4462 0.1481317
log(POPDEN) -0.08316270  0.01254315  -6.6301 3.354e-11
per_mixed   -0.03377962  0.00811054  -4.1649 3.115e-05
per_asian   -0.00413115  0.00096849  -4.2656 1.994e-05
per_black   -0.01653816  0.00126741 -13.0488 < 2.2e-16
per_other   -0.01693012  0.00462999  -3.6566 0.0002556

Lambda: 0.88605, LR test value: 623.55, p-value: < 2.22e-16
Asymptotic standard error: 0.015803
    z-value: 56.068, p-value: < 2.22e-16
Wald statistic: 3143.6, p-value: < 2.22e-16

Log likelihood: 271.8839 for error model
ML residual variance (sigma squared): 0.026911, (sigma: 0.16405)
Number of observations: 983
Number of parameters estimated: 9
AIC: NA (not available for weighted model), (AIC for lm: 95.786)
```

In this case `mod_1.sem$lambda` gives us the spatial parameter. A highly significant lambda of 0.89 indicates that the errors are highly spatially correlated (e.g. due to correlated unobservables). Again, $ = 1 $ would be the maximum.

In spatial error models, we can interpret the coefficients directly, as in a conventional linear model.

4.4.3 SLX

SLX models can either be estimated with `lmSLX()` directly, or by creating $\mathbf{WX}$ manually and plugging it into any available model-fitting function.

```
mod_1.slx <- lmSLX(log(med_house_price) ~ log(no2) + log(POPDEN) +
                     per_mixed + per_asian + per_black + per_other,
                   data = msoa.spdf,
                   listw = queens.lw,
                   Durbin = TRUE) # use a formula to lag only specific covariates
summary(mod_1.slx)
```

```
Call:
lm(formula = formula(paste("y ~ ", paste(colnames(x)[-1], collapse = "+"))),
    data = as.data.frame(x), weights = weights)

Residuals:
     Min       1Q   Median       3Q      Max
-0.50809 -0.16605 -0.01817  0.13055  1.09039

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     10.582440   0.153862  68.779  < 2e-16 ***
log.no2.        -0.440727   0.181063  -2.434  0.01511 *
log.POPDEN.     -0.076840   0.017345  -4.430 1.05e-05 ***
per_mixed       -0.033042   0.011298  -2.925  0.00353 **
per_asian       -0.002381   0.001474  -1.615  0.10655
per_black       -0.016229   0.001801  -9.009  < 2e-16 ***
per_other       -0.020391   0.006564  -3.107  0.00195 **
lag.log.no2.     0.993602   0.199370   4.984 7.38e-07 ***
lag.log.POPDEN.  0.113262   0.028752   3.939 8.76e-05 ***
lag.per_mixed    0.126069   0.014294   8.820  < 2e-16 ***
lag.per_asian   -0.003828   0.001661  -2.305  0.02140 *
lag.per_black   -0.018054   0.002241  -8.056 2.30e-15 ***
lag.per_other    0.048139   0.007971   6.039 2.20e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2262 on 970 degrees of freedom
Multiple R-squared:  0.653, Adjusted R-squared:  0.6487
F-statistic: 152.1 on 12 and 970 DF,  p-value: < 2.2e-16
```

In SLX models, we can simply interpret the coefficients of direct and indirect (spatially lagged) covariates.

For instance, lets look at population density:

Interpretaion SLX

1. A high population density in the focal unit is related to lower house prices (a 1% increase in population density decreses house prices by -0.08%), but
2. A high population density in the neighbouring areas is related to higher house prices (while keeping population density in the focal unit constant). A 1% increase in the

average population density *across the adjacent neighbourhoods* increases house prices in *the focal unit* by 0.11%)

Potential interpretation: areas with a low population density in central regions of the city (high pop density in surrounding neighbourhoods) have higher house prices. We could try testing this interpretation by including the distance to the city center as a control.

Also note how the air pollution coefficient has changed here, with a negative effect in the focal unit and positive one amonf the neighbouring units.

An alternative way of estimating the same model is lagging the covariates first.

```
# Loop through vars and create lagged variables
msoa.spdf$log_POPDEN <- log(msoa.spdf$POPDEN)
msoa.spdf$log_no2 <- log(msoa.spdf$no2)
msoa.spdf$log_med_house_price <- log(msoa.spdf$med_house_price)

vars <- c("log_med_house_price", "log_no2", "log_POPDEN",
          "per_mixed", "per_asian", "per_black", "per_other",
          "per_owner", "per_social", "pubs_count")
for(v in vars){
  msoa.spdf[, paste0("w.", v)] <- lag.listw(queens.lw,
                                            var = st_drop_geometry(msoa.spdf)[, v])
}

# Alternatively:
w_vars <- create_WX(st_drop_geometry(msoa.spdf[, vars]),
                    listw = queens.lw,
                    prefix = "w")

head(w_vars)
```

```
  w.log_med_house_price w.log_no2 w.log_POPDEN w.per_mixed w.per_asian
1              12.98382  3.843750     4.662014    4.748368   23.899916
2              12.28730  3.098960     3.300901    3.978275   19.951593
3              12.21207  3.206338     4.009795    3.997487   20.793559
4              12.18176  3.169934     3.630360    2.759082    7.633439
5              12.11159  3.221203     3.993660    3.930061   12.791140
6              12.08393  3.217865     3.876070    3.419488    8.997514
  w.per_black w.per_other w.per_owner w.per_social w.pubs_count
1    7.879758   3.2080074    25.75738     33.85580    8.5454545
2   10.451828   1.6368986    66.42278     15.75042    0.6666667
```

```
3   12.965863   1.7526693    58.72637     21.38169    0.2857143
4   12.135478   0.6992118    66.52519     19.70500    0.2000000
5   16.108948   1.3817357    53.05539     29.44022    0.4000000
6   15.312652   0.9611710    59.49460     23.81126    0.1666667
```

And subsequently we use those new variables in a linear model.

```
mod_1.lm <- lm (log(med_house_price) ~ log(no2) + log(POPDEN) +
                  per_mixed + per_asian + per_black + per_other +
                  w.log_no2 + w.log_POPDEN +
                  w.per_mixed + w.per_asian + w.per_black + w.per_other,
                data = msoa.spdf)
summary(mod_1.lm)
```

```
Call:
lm(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other + w.log_no2 +
    w.log_POPDEN + w.per_mixed + w.per_asian + w.per_black +
    w.per_other, data = msoa.spdf)

Residuals:
     Min       1Q   Median       3Q      Max
-0.50809 -0.16605 -0.01817  0.13055  1.09039

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.582440   0.153862  68.779  < 2e-16 ***
log(no2)     -0.440727   0.181063  -2.434  0.01511 *
log(POPDEN)  -0.076840   0.017345  -4.430 1.05e-05 ***
per_mixed    -0.033042   0.011298  -2.925  0.00353 **
per_asian    -0.002381   0.001474  -1.615  0.10655
per_black    -0.016229   0.001801  -9.009  < 2e-16 ***
per_other    -0.020391   0.006564  -3.107  0.00195 **
w.log_no2     0.993602   0.199370   4.984 7.38e-07 ***
w.log_POPDEN  0.113262   0.028752   3.939 8.76e-05 ***
w.per_mixed   0.126069   0.014294   8.820  < 2e-16 ***
w.per_asian  -0.003828   0.001661  -2.305  0.02140 *
w.per_black  -0.018054   0.002241  -8.056 2.30e-15 ***
w.per_other   0.048139   0.007971   6.039 2.20e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.2262 on 970 degrees of freedom
Multiple R-squared:  0.653, Adjusted R-squared:  0.6487
F-statistic: 152.1 on 12 and 970 DF,  p-value: < 2.2e-16
```

Looks pretty similar to `lmSLX()` results, and it should! A big advantage of the SLX specification is that we can use the lagged variables in basically all methods which take variables as inputs, such as non-linear models, matching algorithms, and machine learning tools.

Moreover, using the lagged variables gives a high degree of freedom. For instance, we could (not saying that it necessarily makes sense):

- Use different weights matrices for different variables
- Include higher order neighbours using `nblag()` (with an increasing number of orders we go towards a more global model, but we estimate a coefficient for each spillover, instead of estimating just one)
- Use machine learning techniques to determine the best fitting weights specification.

4.4.4 SDEM

SDEM models can be estimated using `errorsarlm()` with the additional option `Durbin = TRUE`.

```
mod_1.sdem <- errorsarlm(log(med_house_price) ~ log(no2) + log(POPDEN) +
                           per_mixed + per_asian + per_black + per_other,
                         data = msoa.spdf,
                         listw = queens.lw,
                         Durbin = TRUE) # we could here extend to SDEM
summary(mod_1.sdem)
```

```
Call:errorsarlm(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other, data = msoa.spdf,
    listw = queens.lw, Durbin = TRUE)

Residuals:
      Min        1Q    Median        3Q       Max
-0.617795 -0.106380 -0.014832  0.095826  0.927446

Type: error
Coefficients: (asymptotic standard errors)
```

```
                  Estimate Std. Error  z value  Pr(>|z|)
(Intercept)     10.4422703  0.3652148  28.5921 < 2.2e-16
log(no2)        -0.2057493  0.1264914  -1.6266 0.1038248
log(POPDEN)     -0.0769743  0.0132094  -5.8272 5.635e-09
per_mixed       -0.0222406  0.0079705  -2.7904 0.0052649
per_asian       -0.0037484  0.0010054  -3.7284 0.0001927
per_black       -0.0179751  0.0012383 -14.5161 < 2.2e-16
per_other       -0.0150218  0.0044895  -3.3460 0.0008199
lag.log(no2)     1.0004491  0.1739833   5.7503 8.911e-09
lag.log(POPDEN) -0.0054241  0.0327802  -0.1655 0.8685763
lag.per_mixed    0.0669699  0.0169349   3.9545 7.668e-05
lag.per_asian   -0.0018566  0.0015957  -1.1635 0.2446368
lag.per_black   -0.0079949  0.0024833  -3.2195 0.0012842
lag.per_other    0.0273378  0.0087430   3.1268 0.0017671

Lambda: 0.76173, LR test value: 455.7, p-value: < 2.22e-16
Asymptotic standard error: 0.024949
    z-value: 30.531, p-value: < 2.22e-16
Wald statistic: 932.15, p-value: < 2.22e-16

Log likelihood: 300.847 for error model
ML residual variance (sigma squared): 0.027504, (sigma: 0.16584)
Number of observations: 983
Number of parameters estimated: 15
AIC: NA (not available for weighted model), (AIC for lm: -117.99)
```

And this SDEM can be interpreted like a combination of SEM and SLX.

First, we still see highly significant auto-correlation in the error term. However, it's lower in magnitude now that we also include the $\mathbf{W}X$ terms.

Second, the coefficients tell a smimilar story as in the SLX (use the same interpretation), but some coefficient magnitudes have become smaller.

4.4.5 SDM

SDM models can be estimated using `lagsarlm()` with the additional option `Durbin = TRUE`.

```
mod_1.sdm <- lagsarlm(log(med_house_price) ~ log(no2) + log(POPDEN) +
                        per_mixed + per_asian + per_black + per_other,
                      data = msoa.spdf,
                      listw = queens.lw,
```

```
                        Durbin = TRUE) # we could here extend to SDM
summary(mod_1.sdm)
```

```
Call:lagsarlm(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other, data = msoa.spdf,
    listw = queens.lw, Durbin = TRUE)

Residuals:
      Min        1Q    Median        3Q       Max
-0.614314 -0.107947 -0.013509  0.092234  0.917398

Type: mixed
Coefficients: (asymptotic standard errors)
                  Estimate Std. Error  z value  Pr(>|z|)
(Intercept)      2.7843426  0.2944721   9.4554 < 2.2e-16
log(no2)        -0.3112762  0.1308101  -2.3796 0.0173312
log(POPDEN)     -0.0802866  0.0125213  -6.4120 1.436e-10
per_mixed       -0.0368998  0.0081596  -4.5223 6.118e-06
per_asian       -0.0033726  0.0010636  -3.1711 0.0015189
per_black       -0.0159770  0.0013006 -12.2848 < 2.2e-16
per_other       -0.0209743  0.0047369  -4.4279 9.516e-06
lag.log(no2)     0.4880923  0.1456778   3.3505 0.0008067
lag.log(POPDEN)  0.0781188  0.0207600   3.7629 0.0001679
lag.per_mixed    0.0640880  0.0104646   6.1243 9.110e-10
lag.per_asian    0.0017665  0.0012101   1.4598 0.1443498
lag.per_black    0.0070487  0.0017938   3.9295 8.511e-05
lag.per_other    0.0284822  0.0057774   4.9299 8.226e-07

Rho: 0.73126, LR test value: 501.83, p-value: < 2.22e-16
Asymptotic standard error: 0.025889
    z-value: 28.246, p-value: < 2.22e-16
Wald statistic: 797.86, p-value: < 2.22e-16

Log likelihood: 323.9111 for mixed model
ML residual variance (sigma squared): 0.026633, (sigma: 0.1632)
Number of observations: 983
Number of parameters estimated: 15
AIC: -617.82, (AIC for lm: -117.99)
LM test for residual autocorrelation
test value: 36.704, p-value: 1.3747e-09
```

And this SDM can be interpreted like a combination of SAR and SLX.

First, there's still substantial auto-correlation in $\mathbf{y}$, and this has become even stronger as compared to SAR.

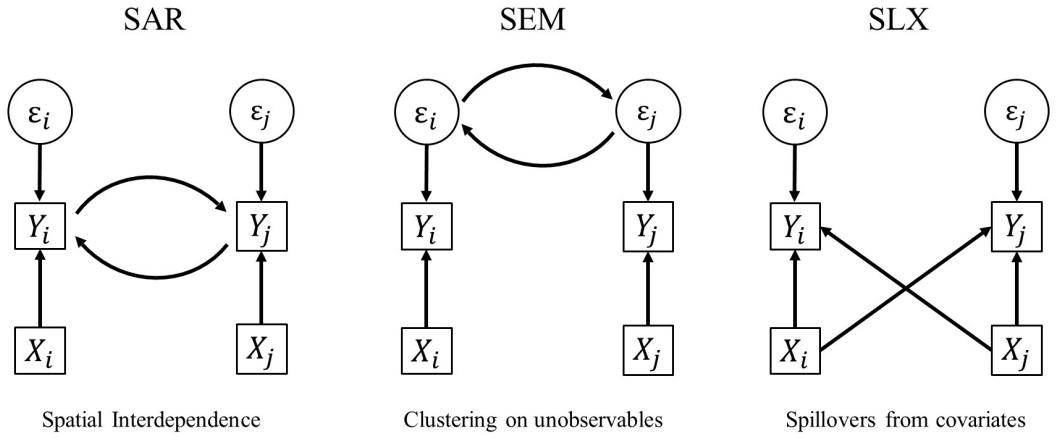
Second, we can interpret the direction of the effect, but we *cannot interpret the coefficient as marginal effects*.

4.5 Estimation

Note that most of the spatial model specifications can not be estimated by Least Squares (LS), as using (constrained) LS estimators for models containing a spatially lagged dependent variable or disturbance leads to inconsistent results (Anselin and Bera 1998; Franzese and Hays 2007). However, an extensive amount of econometric literature discusses different estimation methods based on (quasi-) maximum likelihood (Anselin 1988; L. Lee 2004; Ord 1975) or instrumental variable approaches using generalized methods of moments (Drukker, Egger, and Prucha 2013; Kelejian and Prucha 1998, 2010), in which the endogenous lagged variables can be instrumented by q higher order lags of the exogenous regressors $(\mathbf{X}, \mathbf{WX}, \mathbf{W}^2\mathbf{X}, ..., \mathbf{W}^q\mathbf{X})$ (Kelejian and Prucha 1998).

4.5.1 Simulataneity bias

Remember what is happening when we estimate a spatial auto-regressive model.



Note the circular process here: My X influences my Y, which then influences your Y, which then influences my Y again. We write this as

$$\mathbf{y} = \alpha + \rho\mathbf{W}\mathbf{y} + \mathbf{X} + .$$

If we ignore $\mathbf{X}$ and write the pure auto-regressive term in its reduce form, we get:

$$\mathbf{y} = (\mathbf{I}_n - \rho\mathbf{W})^{-1} \varepsilon,$$

and the spatial lag term is

$$\mathbf{W}\mathbf{y} = \mathbf{W} (\mathbf{I}_n - \rho\mathbf{W})^{-1} \varepsilon.$$

The OLS estimator for the spatial lag term then is

$$\hat{\rho}_{OLS} = \left[\underbrace{(\mathbf{W}\mathbf{y})^\top}_{(1\times n)} \underbrace{(\mathbf{W}\mathbf{y})}_{(n\times 1)} \right]^{-1} \underbrace{(\mathbf{W}\mathbf{y})^\top}_{(1\times n)} \underbrace{\mathbf{y}}_{(n\times 1)}.$$

It can then be shown that the OLS estimators equals

$$\hat{\rho}_{OLS} = \rho + \left[(\mathbf{W}\mathbf{y})^\top (\mathbf{W}\mathbf{y}) \right]^{-1} (\mathbf{W}\mathbf{y})^\top \varepsilon = \rho + \left(\sum_{i=1}^{n} \mathbf{y}_{Li}^2 \right)^{-1} \left(\sum_{i=1}^{n} \mathbf{y}_{Li} \epsilon_i \right),$$

with $\mathbf{y}_{Li}$ defined as the ith element of the spatial lag operator $\mathbf{W}\mathbf{y} = \mathbf{y}_L$. It can further be shown that the second part of the equation $\neq 0$, which demonstrates that OLS gives a biased estimate of ρ (Franzese and Hays 2007; Sarrias 2023).

> **Warning**
>
> Do not estimate spatial lags of the dependent variable in OLS. It will suffer from simultaneity bias.

4.5.2 Instrumental variable

A potential way of estimating spatial lag /SAR models is 2SLS (Kelejian and Prucha 1998).

We start with our standard model

$$\mathbf{y} = \alpha + \rho\mathbf{W}\mathbf{y} + \mathbf{X} + .$$

As we have seen above, there is a problem of simultaneity: the "covariate" $\mathbf{W}\mathbf{y}$ is endogenous. One way of dealing with this endogeneity problem is the Instrumental Variable approach.

So, the question is what are good instruments $\mathbf{H}$ for $\mathbf{Wy}$? As we have specified the mode, we are sure that $\mathbf{X}$ determines $\mathbf{y}$. Thus, it must be true that $\mathbf{WX}$ and $\mathbf{W}^2\mathbf{X}, \dots, \mathbf{W}^l\mathbf{X}$ determines $\mathbf{Wy}$.

Note that $\mathbf{W}^l$ denotes higher orders of $\mathbf{W}$. So $\mathbf{W}^2$ are the second order neighbours (neighbours of neighbours), and $\mathbf{W}^3$ are the third order neighbours (the neighbours of my neighbour's neighbours), and so on…

We will discuss this in more detail later, but note for now that the reduced form of the SAR always contains a series of higher order neighbours.

$$(\mathbf{I}_N - \rho\mathbf{W})^{-1}\beta_k = (\mathbf{I}_N + \rho\mathbf{W} + \rho^2\mathbf{W}^2 + \rho^3\mathbf{W}^3 + ...)\beta_k = (\mathbf{I}_N + \sum_{h=1}^{\infty}\rho^h\mathbf{W}^h)\beta_k.$$

Thus, Kelejian and Prucha (1998) suggested to use a set of lagged covariates as instruments for $\mathbf{WY}$:

$$\mathbf{H} = \mathbf{X}, \mathbf{WX}, \mathbf{W}^2\mathbf{X}, ..., \mathbf{W}^l\mathbf{X},$$

where l is a pre-defined number for the higher order neighbours included. In practice, l is usually restricted to $l = 2$.

This has further been developed by, for instance, using a (truncated) power series as instruments (Kelejian, Prucha, and Yuzefovich 2004):

$$\mathbf{H} = \left[\mathbf{X}, \mathbf{W}\left(\sum_{l=1}^{\infty}\rho^l\mathbf{W}^l\right)\mathbf{X}\right].$$

We can estimate this using the pacakge `spatialreg` with the function `stsls()`,

```
mod_1.sls <- stsls(log(med_house_price) ~ log(no2) + log(POPDEN) +
                     per_mixed + per_asian + per_black + per_other,
                   data = msoa.spdf,
                   listw = queens.lw,
                   robust = TRUE, #  heteroskedasticity robust SEs
                   W2X = TRUE) # Second order neighbours are included as instruments (else
summary(mod_1.sls)
```

```
Call:stsls(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other, data = msoa.spdf,
    listw = queens.lw, robust = TRUE, W2X = TRUE)
```

```
Residuals:
       Min         1Q     Median         3Q        Max
-0.5464924 -0.1238002 -0.0052299  0.0989150  1.0793093

Coefficients:
               Estimate HC0 std. Error z value  Pr(>|z|)
Rho          0.71004211     0.04678235 15.1776 < 2.2e-16
(Intercept)  2.73582523     0.50997823  5.3646 8.113e-08
log(no2)     0.37752751     0.04920257  7.6729 1.688e-14
log(POPDEN) -0.05710992     0.01684036 -3.3913 0.0006957
per_mixed    0.01634307     0.00588488  2.7771 0.0054842
per_asian   -0.00205426     0.00045905 -4.4750 7.640e-06
per_black   -0.01166456     0.00128557 -9.0734 < 2.2e-16
per_other   -0.00280423     0.00332302 -0.8439 0.3987377

Residual variance (sigma squared): 0.035213, (sigma: 0.18765)
```

4.6 Maximum Likelihood

Maximum Likelihood estimation of spatial models is the most common way of estimation. The procedure to estimate SAR models via ML is based on Ord (1975) and Anselin (1988). If you are interested in more details, please look at the more extensive 3-day workshop or have a look into Sarrias (2023).

The package `spatialreg` Pebesma and Bivand (2023) provides a series of functions to calculate the ML estimator for all spatial models we have considered.

Table from Pebesma and Bivand (2023):

model	model name	maximum likelihood estimation function
SEM	spatial error	`errorsarlm(..., Durbin=FALSE)`
SEM	spatial error	`spautolm(..., family="SAR")`
SDEM	spatial Durbin error	`errorsarlm(..., Durbin=TRUE)`
SLM	spatial lag	`lagsarlm(..., Durbin=FALSE)`
SDM	spatial Durbin	`lagsarlm(..., Durbin=TRUE)`
SAC	spatial autoregressive combined	`sacsarlm(..., Durbin=FALSE)`
GNM	general nested	`sacsarlm(..., Durbin=TRUE)`

ML SAR

```r
mod_1.sar <- lagsarlm(log(med_house_price) ~ log(no2) + log(POPDEN) +
                        per_mixed + per_asian + per_black + per_other,
                      data = msoa.spdf,
                      listw = queens.lw,
                      Durbin = FALSE) # we could here extend to SDM
summary(mod_1.sar)
```

```
Call:lagsarlm(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other, data = msoa.spdf,
    listw = queens.lw, Durbin = FALSE)

Residuals:
       Min         1Q     Median         3Q        Max
-0.5281789 -0.1220524 -0.0099245  0.0992203  1.0936745

Type: lag
Coefficients: (asymptotic standard errors)
               Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)  3.17383180  0.29041604  10.9286 < 2.2e-16
log(no2)     0.39705423  0.04452880   8.9168 < 2.2e-16
log(POPDEN) -0.05583014  0.01242876  -4.4920 7.055e-06
per_mixed    0.01851577  0.00579832   3.1933  0.001407
per_asian   -0.00228346  0.00045876  -4.9775 6.442e-07
per_black   -0.01263650  0.00100282 -12.6009 < 2.2e-16
per_other   -0.00161419  0.00289082  -0.5584  0.576582

Rho: 0.66976, LR test value: 473.23, p-value: < 2.22e-16
Asymptotic standard error: 0.025311
    z-value: 26.461, p-value: < 2.22e-16
Wald statistic: 700.19, p-value: < 2.22e-16

Log likelihood: 196.7203 for lag model
ML residual variance (sigma squared): 0.035402, (sigma: 0.18815)
Number of observations: 983
Number of parameters estimated: 9
AIC: -375.44, (AIC for lm: 95.786)
LM test for residual autocorrelation
test value: 8.609, p-value: 0.0033451
```

ML SEM

```r
mod_1.sem <- errorsarlm(log(med_house_price) ~ log(no2) + log(POPDEN) +
                          per_mixed + per_asian + per_black + per_other,
                        data = msoa.spdf,
                        listw = queens.lw,
                        Durbin = FALSE) # we could here extend to SDEM
summary(mod_1.sem)
```

```
Call:errorsarlm(formula = log(med_house_price) ~ log(no2) + log(POPDEN) +
    per_mixed + per_asian + per_black + per_other, data = msoa.spdf,
    listw = queens.lw, Durbin = FALSE)

Residuals:
      Min        1Q    Median        3Q       Max
-0.581785 -0.105218 -0.012758  0.094430  0.913425

Type: error
Coefficients: (asymptotic standard errors)
               Estimate  Std. Error  z value  Pr(>|z|)
(Intercept) 12.92801104  0.35239139  36.6865 < 2.2e-16
log(no2)     0.15735296  0.10880727   1.4462 0.1481317
log(POPDEN) -0.08316270  0.01254315  -6.6301 3.354e-11
per_mixed   -0.03377962  0.00811054  -4.1649 3.115e-05
per_asian   -0.00413115  0.00096849  -4.2656 1.994e-05
per_black   -0.01653816  0.00126741 -13.0488 < 2.2e-16
per_other   -0.01693012  0.00462999  -3.6566 0.0002556

Lambda: 0.88605, LR test value: 623.55, p-value: < 2.22e-16
Asymptotic standard error: 0.015803
    z-value: 56.068, p-value: < 2.22e-16
Wald statistic: 3143.6, p-value: < 2.22e-16

Log likelihood: 271.8839 for error model
ML residual variance (sigma squared): 0.026911, (sigma: 0.16405)
Number of observations: 983
Number of parameters estimated: 9
AIC: NA (not available for weighted model), (AIC for lm: 95.786)
```

5 Spatial Impacts

Required packages

```
pkgs <- c("sf", "mapview", "spdep", "spatialreg", "tmap", "viridisLite") # note: load spde
lapply(pkgs, require, character.only = TRUE)
```

Session info

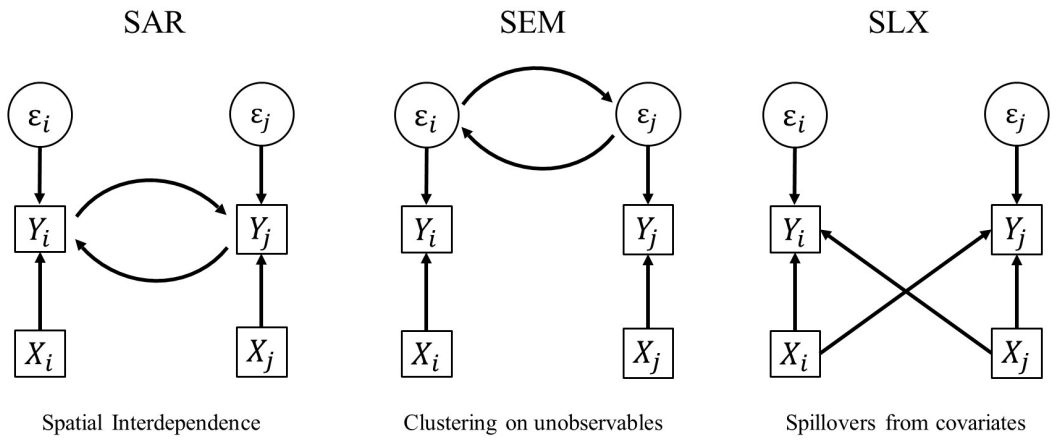
```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default


locale:
[1] LC_COLLATE=English_United Kingdom.utf8
[2] LC_CTYPE=English_United Kingdom.utf8
[3] LC_MONETARY=English_United Kingdom.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.utf8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] viridisLite_0.4.2 tmap_3.3-3        spatialreg_1.2-9  Matrix_1.5-4.1
[5] spdep_1.2-8       spData_2.3.0      mapview_2.11.0    sf_1.0-13
```

```
loaded via a namespace (and not attached):
 [1] xfun_0.39           raster_3.6-23       htmlwidgets_1.6.2   lattice_0.21-8
 [5] vctrs_0.6.3         tools_4.3.1         crosstalk_1.2.0     LearnBayes_2.15.1
 [9] generics_0.1.3      parallel_4.3.1      sandwich_3.0-2      stats4_4.3.1
[13] tibble_3.2.1        proxy_0.4-27        fansi_1.0.4         pkgconfig_2.0.3
[17] KernSmooth_2.23-21 satellite_1.0.4     RColorBrewer_1.1-3  leaflet_2.1.2
[21] webshot_0.5.5       lifecycle_1.0.3     compiler_4.3.1      deldir_1.0-9
[25] munsell_0.5.0       terra_1.7-39        leafsync_0.1.0      codetools_0.2-
19
[29] stars_0.6-1         htmltools_0.5.5     class_7.3-22        pillar_1.9.0
[33] MASS_7.3-60         classInt_0.4-9      lwgeom_0.2-13       wk_0.7.3
[37] abind_1.4-5         boot_1.3-28.1       multcomp_1.4-25     nlme_3.1-162
[41] tidyselect_1.2.0    digest_0.6.32       mvtnorm_1.2-2       dplyr_1.1.2
[45] splines_4.3.1       fastmap_1.1.1       grid_4.3.1          colorspace_2.1-
0
[49] expm_0.999-7        cli_3.6.1           magrittr_2.0.3      base64enc_0.1-3
[53] dichromat_2.0-0.1   XML_3.99-0.14       survival_3.5-5      utf8_1.2.3
[57] TH.data_1.1-2       leafem_0.2.0        e1071_1.7-13        scales_1.2.1
[61] sp_1.6-1            rmarkdown_2.23      zoo_1.8-12          png_0.1-8
[65] coda_0.19-4         evaluate_0.21       knitr_1.43          tmaptools_3.1-1
[69] s2_1.1.4            rlang_1.1.1         Rcpp_1.0.10         glue_1.6.2
[73] DBI_1.1.3           rstudioapi_0.14     jsonlite_1.8.5      R6_2.5.1
[77] units_0.8-2
```

Reload data from pervious session

```
load("_data/msoa2_spatial.RData")
```



5.1 Coefficient estimates $\neq$ 'marginal' effects

> **Warning**
>
> Do not interpret coefficients as marginal effects in SAR, SAC, and SDM!!

At first glance, the specifications presented above seem relatively similar in the way of modelling spatial effects. **Yet, they differ in very important aspects**.

First, models with an endogenous spatial term (SAR, SAC, and SDM) assume a very different spatial dependence structure than models with only exogenous spatial terms as SLX and SDEM specifications. While the first three assume **global** spatial dependence, the second two assume **local** spatial dependence (Anselin 2003; Halleck Vega and Elhorst 2015; LeSage and Pace 2009).

Second, the interpretation of the coefficients differs greatly between models with and without endogenous effects. This becomes apparent when considering the reduced form of the equations above. Exemplary using the SAR model, the reduced form is given by:

$$\begin{aligned} \mathbf{y} - \rho\mathbf{W}\mathbf{y} &= \mathbf{X} + , \\ (\mathbf{I}_N - \rho\mathbf{W})\mathbf{y} &= \mathbf{X} + , \\ \mathbf{y} &= (\mathbf{I}_N - \rho\mathbf{W})^{-1}(\mathbf{X} +), \end{aligned}$$

where $\mathbf{I}_N$ is an $N \times N$ diagonal matrix (diagonal elements equal 1, 0 otherwise). This contains no spatially lagged dependent variable on the right-hand side.

If we want to interpret coefficient, we are usually in marginal or partial effects (the association between a unit change in X and Y). We obtain these effects by looking at the first derivative.

When taking the first derivative of the explanatory variable $\mathbf{x}_k$ from the reduced form in (**??**) to interpret the partial effect of a unit change in variable $\mathbf{x}_k$ on $\mathbf{y}$, we receive

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_k} = \underbrace{(\mathbf{I}_N - \rho\mathbf{W})^{-1}}_{N \times N} \beta_k,$$

for each covariate $k = \{1, 2, ..., K\}$. As can be seen, the partial derivative with respect to $\mathbf{x}_k$ produces an $N \times N$ matrix, thereby representing the partial effect of each unit i onto the focal unit i itself and all other units $j = \{1, 2, ..., i-1, i+1, ..., N\}$.

Note that the diagonal elements of $(\mathbf{I}_N - \rho\mathbf{W})^{-1}$ are not zero anymore (as they are in $\mathbf{W}$). Look at the following minimal example:

$$\tilde{\mathbf{W}} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}, \text{and normalized } \mathbf{W} = \begin{pmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0.33 & 0 & 0.33 & 0 & 0.33 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0.33 & 0 & 0.33 & 0 & 0.33 \\ 0 & 0.5 & 0 & 0.5 & 0 \end{pmatrix}$$

and

$$\rho = 0.6,$$

then

$$\rho\mathbf{W} = \begin{pmatrix} 0 & 0.3 & 0 & 0.3 & 0 \\ 0.2 & 0 & 0.2 & 0 & 0.2 \\ 0 & 0.3 & 0 & 0.3 & 0 \\ 0.2 & 0 & 0.2 & 0 & 0.2 \\ 0 & 0.3 & 0 & 0.3 & 0 \end{pmatrix}.$$

If we want to get the total effect of X on Y we need to add the direct association within i and j and so on…

$$\begin{aligned} \mathbf{I}_N - \rho\mathbf{W} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0.3 & 0 & 0.3 & 0 \\ 0.2 & 0 & 0.2 & 0 & 0.2 \\ 0 & 0.3 & 0 & 0.3 & 0 \\ 0.2 & 0 & 0.2 & 0 & 0.2 \\ 0 & 0.3 & 0 & 0.3 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & -0.3 & 0 & -0.3 & 0 \\ -0.2 & 1 & -0.2 & 0 & -0.2 \\ 0 & 0.3 & 1 & 0.3 & 0 \\ -0.2 & 0 & -0.2 & 1 & -0.2 \\ 0 & -0.3 & 0 & -0.3 & 1 \end{pmatrix}. \end{aligned}$$

And finally we take the inverse of that

$$(\mathbf{I}_N - \rho\mathbf{W})^{-1} = \begin{pmatrix} 1 & -0.3 & 0 & -0.3 & 0 \\ -0.2 & 1 & -0.2 & 0 & -0.2 \\ 0 & 0.3 & 1 & 0.3 & 0 \\ -0.2 & 0 & -0.2 & 1 & -0.2 \\ 0 & -0.3 & 0 & -0.3 & 1 \end{pmatrix}^{-1}$$
$$= \begin{pmatrix} 1.1875 & 0.46875 & 0.1875 & 0.46875 & 0.1875 \\ 0.3125 & 1.28125 & 0.3125 & 0.28125 & 0.3125 \\ 0.1875 & 0.46875 & 1.1875 & 0.46875 & 0.1875 \\ 0.3125 & 0.28125 & 0.3125 & 1.28125 & 0.3125 \\ 0.1875 & 0.46875 & 0.1875 & 0.46875 & 1.1875 \end{pmatrix}.$$

As you can see, $(\mathbf{I}_N - \rho\mathbf{W})^{-1}$ has *diagonal elements* > 1: these are feedback loops. My X influences my Y directly, but my Y then influences my neigbour's Y, which then influences my Y again (also also other neighbour's Ys). Thus the influence of my X on my Y includes a spatial multiplier.

Check yourself:

```
I = diag(5)
rho = 0.6
W = matrix(c(0 , 0.5 , 0 , 0.5 , 0,
             1/3 , 0 , 1/3 , 0 , 1/3,
             0 , 0.5 , 0 , 0.5 , 0,
             1/3 , 0 , 1/3 , 0 , 1/3,
             0 , 0.5 , 0 , 0.5 , 0), ncol = 5, byrow = TRUE)

(IrW = I - rho*W)
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  1.0 -0.3  0.0 -0.3  0.0
[2,] -0.2  1.0 -0.2  0.0 -0.2
[3,]  0.0 -0.3  1.0 -0.3  0.0
[4,] -0.2  0.0 -0.2  1.0 -0.2
[5,]  0.0 -0.3  0.0 -0.3  1.0
```

```
# (I - rho*W)^-1
(M = solve(IrW))
```

```
       [,1]    [,2]   [,3]    [,4]   [,5]
[1,] 1.1875 0.46875 0.1875 0.46875 0.1875
[2,] 0.3125 1.28125 0.3125 0.28125 0.3125
[3,] 0.1875 0.46875 1.1875 0.46875 0.1875
[4,] 0.3125 0.28125 0.3125 1.28125 0.3125
[5,] 0.1875 0.46875 0.1875 0.46875 1.1875
```

The diagonal elements of M indicate how each unit i influences itself (change of x_i on change of y_i), and each off-diagonal elements in column j represents the effect of j on each other unit i (change of x_j on change of y_i).

$$\begin{pmatrix} 1.1875 & 0.46875 & 0.1875 & 0.46875 & 0.1875 \\ 0.3125 & 1.28125 & 0.3125 & 0.28125 & 0.3125 \\ 0.1875 & 0.46875 & 1.1875 & 0.46875 & 0.1875 \\ 0.3125 & 0.28125 & 0.3125 & 1.28125 & 0.3125 \\ 0.1875 & 0.46875 & 0.1875 & 0.46875 & 1.1875 \end{pmatrix}.$$

For instance, W_{12} indicates that unit 2 has an influence of 0.46875 on unit 1. On the other hand, W_{53} indicates that unit 3 has an influence of magnitude 0.1875 on unit 5.

> Question
>
> Why does unit 3 have any effect o unit 5? According to $\mathbf{W}$ those two units are no neighbours $w_{53} = 0$!

5.2 Global and local spillovers

The kind of indirect spillover effects in SAR, SAC, and SDM models differs from the kind of indirect spillover effects in SLX and SDEM models: while the first three specifications represent **global spillover effects**, the latter three represent **local spillover effects** (Anselin 2003; LeSage and Pace 2009; LeSage 2014).

5.2.1 Local spillovers

In case of SLX and SDEM the spatial spillover effects can be interpreted as the effect of a one unit change of $\mathbf{x}_k$ in the spatially weighted neighbouring observations on the dependent variable of the focal unit: the weighted average among neighbours; when using a row-normalised contiguity weights matrix, $\mathbf{W}\mathbf{x}_k$ is the mean value of $\mathbf{x}_k$ in the neighbouring units.

Assume we have $k = 2$ covariates, then

$$\underbrace{\mathbf{W}}_{N\times N}\underbrace{\mathbf{X}}_{N\times 2}\underbrace{\quad}_{2\times 1} = \begin{pmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0.33 & 0 & 0.33 & 0 & 0.33 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0.33 & 0 & 0.33 & 0 & 0.33 \\ 0 & 0.5 & 0 & 0.5 & 0 \end{pmatrix} \begin{pmatrix} 3 & 100 \\ 4 & 140 \\ 1 & 200 \\ 7 & 70 \\ 5 & 250 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$$

$$= \begin{pmatrix} 6 & 105 \\ 3 & 190 \\ 6 & 105 \\ 3 & 190 \\ 6 & 105 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$$

```
X <- cbind(x1 = c(3,4,1,8,5),
           x2 = c(100,140,200,70,270))
(WX <-  W %*% X)
```

```
     x1  x2
[1,]  6 105
[2,]  3 190
[3,]  6 105
[4,]  3 190
[5,]  6 105
```

Thus, only direct neighbours – as defined in $\mathbf{W}$ – contribute to those local spillover effects. The ̂coefficients only estimate how my direct neighbour's $\mathbf{X}$ values influence my own outcome $\mathbf{y}$.

There are no higher order neighbours involved (as long as we do not model them), nor are there any feedback loops due to interdependence.

5.2.2 Global spillovers

In contrast, spillover effects in SAR, SAC, and SDM models do not only include direct neighbours but also neighbours of neighbours (second order neighbours) and further higher-order neighbours. This can be seen by rewriting the inverse $(\mathbf{I}_N - \rho\mathbf{W})^{-1}$ as power series:A power series of $\sum_{k=0}^{\infty} \mathbf{W}^k$ converges to $(\mathbf{I}-\mathbf{W})^{-1}$ if the maximum absolute eigenvalue of $\mathbf{W} < 1$, which is ensured by standardizing $\mathbf{W}$.}

$$(\mathbf{I}_N - \rho\mathbf{W})^{-1}\beta_k = (\mathbf{I}_N + \rho\mathbf{W} + \rho^2\mathbf{W}^2 + \rho^3\mathbf{W}^3 + ...)\beta_k = (\mathbf{I}_N + \sum_{h=1}^{\infty} \rho^h \mathbf{W}^h)\beta_k,$$

where the identity matrix represents the direct effects and the sum represents the first and higher order indirect effects and the above mentioned feedback loops. This implies that a change in one unit i does not only affect the direct neighbours but passes through the whole system towards higher-order neighbours, where the impact declines with distance within the neighbouring system. Global indirect impacts thus are 'multiplied' by influencing direct neighbours as specified in $\mathbf{W}$ and indirect neighbours not connected according to $\mathbf{W}$, with additional feedback loops between those neighbours.

$$\underbrace{(\underbrace{\mathbf{I}_N}_{N\times N} - \underbrace{\rho}_{\hat{=}0.6} \underbrace{\mathbf{W}}_{N\times N})}_{N\times N}{}^{-1}\beta_k = \begin{pmatrix} 1.1875 & 0.46875 & 0.1875 & 0.46875 & 0.1875 \\ 0.3125 & 1.28125 & 0.3125 & 0.28125 & 0.3125 \\ 0.1875 & 0.46875 & 1.1875 & 0.46875 & 0.1875 \\ 0.3125 & 0.28125 & 0.3125 & 1.28125 & 0.3125 \\ 0.1875 & 0.46875 & 0.1875 & 0.46875 & 1.1875 \end{pmatrix} (\beta_1 + \beta_2)$$

.

All diagonal elements of $\text{diag}(\mathbf{W}) = w_{ii} = 0$. However, diagonal elements of higher order neighbours are not zero $\text{diag}(\mathbf{W}^2) = \text{diag}(\mathbf{WW}) \neq 0$.

Intuitively, $\rho\mathbf{W}$ only represents the effects between direct neighbours (and the focal unit is not a neighbour of the focal unit itself), whereas $\rho^2\mathbf{W}^2$ contains the effects of second order neighbours, where the focal unit is a second order neighbour of the focal unit itself. Thus, $(\mathbf{I}_N - \rho\mathbf{W})^{-1}\beta_k$ includes feedback effects from $\rho^2\mathbf{W}^2$ on (they are part of the direct impacts according to the summary measures below). This is way the diagonal above ≥ 1.

In consequence, local and global spillover effects represent two distinct kinds of spatial spillover effects (LeSage 2014). The interpretation of local spillover effects is straightforward: it represents the effect of all neighbours as defined by $\mathbf{W}$ (the average over all neighbours in case of a row-normalised weights matrix).

For instance, the environmental quality in the focal unit itself but also in neighbouring units could influence the attractiveness of a district and its house prices. In this example it seems reasonable to assume that we have local spillover effects: only the environmental quality in directly contiguous units (e.g. in walking distance) is relevant for estimating the house prices.

In contrast, interpreting global spillover effects can be a bit more difficult. Intuitively, the global spillover effects can be seen as a kind of diffusion process. For example, an exogenous event might increase the house prices in one district of a city, thus leading to an adaptation of house prices in neighbouring districts, which then leads to further adaptations in other units

(the neighbours of the neighbours), thereby globally diffusing the effect of the exogenous event due to the endogenous term.

Yet, those processes happen over time. In a cross-sectional framework, the global spillover effects are hard to interpret. Anselin (2003) proposes an interpretation as an equilibrium outcome, where the partial impact represents an estimate of how this long-run equilibrium would change due to a change in $\mathbf{x}_k$ (LeSage 2014).

5.3 Summary impact measures

Note that the derivative in SAR, SAC, and SDM is a $N \times N$ matrix, returning individual effects of each unit on each other unit, differentiated in *direct, indirect, and total impacts.*

$$(\mathbf{I}_N - \rho\mathbf{W})^{-1}\beta = \begin{pmatrix} 1.1875 & 0.46875 & 0.1875 & 0.46875 & 0.1875 \\ 0.3125 & 1.28125 & 0.3125 & 0.28125 & 0.3125 \\ 0.1875 & 0.46875 & 1.1875 & 0.46875 & 0.1875 \\ 0.3125 & 0.28125 & 0.3125 & 1.28125 & 0.3125 \\ 0.1875 & 0.46875 & 0.1875 & 0.46875 & 1.1875 \end{pmatrix} \beta$$

However, the individual effects (how i influences j) mainly vary because of variation in W.

> ⚠ Do not interpret these as "estimated" individual impacts
>
> We estimate two scalar parameters in a SAR model: β for the direct coefficient and *rho* for the auto-regressive parameter.
> All variation in the effects matrix $(\mathbf{I}_N - \rho\mathbf{W})^{-1}$ comes from the relationship in $\mathbf{W}$ which we have given a-priori!

Since reporting the individual partial effects is usually not of interest, LeSage and Pace (2009) proposed to average over these effect matrices. While the average diagonal elements of the effects matrix $(\mathbf{I}_N - \rho\mathbf{W})^{-1}$ represent the so called direct impacts of variable $\mathbf{x}_k$, the average column-sums of the off-diagonal elements represent the so called indirect impacts (or spatial spillover effects).

direct impacts refer to an average effect of a unit change in x_i on y_i, and the indirect (spillover) impacts indicate how a change in x_i, on average, influences all neighbouring units y_j.

Though previous literature (Halleck Vega and Elhorst 2015; LeSage and Pace 2009) has established the notation of direct and indirect impacts, it is important to note that also the direct impacts comprise a spatial 'multiplier' component if we specify an endogenous lagged

depended variable, as a change in $\mathbf{x}_i$ influences $\mathbf{y}_i$, which influences $\mathbf{y}_j$, which in turn influences $\mathbf{y}_i$.

Usually, one should use summary measures to report effects in spatial models (LeSage and Pace 2009). Halleck Vega and Elhorst (2015) provide a nice summary of the impacts for each model:

Model	Direct Impacts	Indirect Impacts	type
OLS/SEM	β_k	–	–
SAR/SAC	Diagonal elements of $(\mathbf{I} - \rho\mathbf{W})^{-1}\beta_k$	Off-diagonal elements of $(\mathbf{I} - \rho\mathbf{W})^{-1}\beta_k$	global
SLX/SDEM	β_k	θ_k	local
SDM	Diagonal elements of $(\mathbf{I} - \rho\mathbf{W})^{-1}[\beta_k + \mathbf{W}\theta_k]$	Off-diagonal elements of $(\mathbf{I} - \rho\mathbf{W})^{-1}[\beta_k + \mathbf{W}\theta_k]$	global

$$(\mathbf{I}_N - \rho\mathbf{W})^{-1}\beta = \begin{pmatrix} 1.1875 & 0.46875 & 0.1875 & 0.46875 & 0.1875 \\ 0.3125 & 1.28125 & 0.3125 & 0.28125 & 0.3125 \\ 0.1875 & 0.46875 & 1.1875 & 0.46875 & 0.1875 \\ 0.3125 & 0.28125 & 0.3125 & 1.28125 & 0.3125 \\ 0.1875 & 0.46875 & 0.1875 & 0.46875 & 1.1875 \end{pmatrix} \beta$$

The different indirect effects / spatial effects mean conceptually different things:

- Global spillover effects: SAR, SAC, SDM
- Local spillover effects: SLX, SDEM

> ⚠ Common ratio between direct and indirect impacts in SAR and SAC
>
> Note that impacts in SAR only estimate one single spatial multiplier coefficient. Thus direct and indirect impacts are bound to a common ratio, say ϕ, across all covariates. if $\beta_1^{direct} = \phi\beta_1^{indirect}$, then $\beta_2^{direct} = \phi\beta_2^{indirect}$, $\beta_k^{direct} = \phi\beta_k^{indirect}$.

We can calculate these impacts using `impacts()` with simulated distributions, e.g. for the SAR model:

```
mod_1.sar.imp <- impacts(mod_1.sar, listw = queens.lw, R = 300)
summary(mod_1.sar.imp, zstats = TRUE, short = TRUE)
```

```
Impact measures (lag, exact):
                 Direct     Indirect        Total
log(no2)     0.447853184  0.754466618  1.202319802
log(POPDEN) -0.062973027 -0.106086209 -0.169059236
per_mixed    0.020884672  0.035182931  0.056067603
per_asian   -0.002575602 -0.004338934 -0.006914536
per_black   -0.014253206 -0.024011369 -0.038264575
per_other   -0.001820705 -0.003067212 -0.004887917
========================================================
Simulation results ( variance matrix):
========================================================
Simulated standard errors
                 Direct     Indirect       Total
log(no2)    0.0493360513 0.0945777116 0.131581964
log(POPDEN) 0.0141980164 0.0259020812 0.039203765
per_mixed   0.0066724654 0.0117588256 0.018210782
per_asian   0.0005243723 0.0009087921 0.001391241
per_black   0.0010254132 0.0023485880 0.002837555
per_other   0.0033282707 0.0056703663 0.008984905

Simulated z-values:
                Direct    Indirect       Total
log(no2)      9.0571100   7.9671535   9.1225055
log(POPDEN)  -4.3718386  -4.0503224  -4.2593668
per_mixed     3.1244348   2.9982320   3.0807776
per_asian    -4.9434649  -4.8056212  -5.0023866
per_black   -13.9619387 -10.2838012 -13.5571549
per_other    -0.4944345  -0.4966615  -0.4965956

Simulated p-values:
            Direct     Indirect   Total
log(no2)    < 2.22e-16 1.5543e-15 < 2.22e-16
log(POPDEN) 1.2320e-05 5.1147e-05 2.0501e-05
per_mixed   0.0017815  0.0027155  0.0020646
per_asian   7.6746e-07 1.5427e-06 5.6625e-07
per_black   < 2.22e-16 < 2.22e-16 < 2.22e-16
per_other   0.6209993  0.6194278  0.6194742
```

```
# Alternative with traces (better for large W)
W <- as(queens.lw, "CsparseMatrix")
trMatc <- trW(W, type = "mult",
```

```
              m = 30) # number of powers
mod_1.sar.imp2 <- impacts(mod_1.sar,
                          tr = trMatc, # trace instead of listw
                          R = 300,
                          Q = 30) # number of power series used for approximation
summary(mod_1.sar.imp2, zstats = TRUE, short = TRUE)
```

```
Impact measures (lag, trace):
                 Direct     Indirect        Total
log(no2)     0.447853101  0.754459497  1.202312598
log(POPDEN) -0.062973015 -0.106085208 -0.169058223
per_mixed    0.020884668  0.035182599  0.056067267
per_asian   -0.002575601 -0.004338893 -0.006914494
per_black   -0.014253203 -0.024011142 -0.038264346
per_other   -0.001820704 -0.003067183 -0.004887888
========================================================
Simulation results ( variance matrix):
========================================================
Simulated standard errors
                Direct     Indirect       Total
log(no2)    0.046635516 0.0952744216 0.129615383
log(POPDEN) 0.014495202 0.0270495306 0.040662858
per_mixed   0.006687957 0.0119307601 0.018399714
per_asian   0.000482245 0.0008561783 0.001296080
per_black   0.001023887 0.0022976623 0.002789672
per_other   0.003024391 0.0053574762 0.008368900

Simulated z-values:
                Direct    Indirect       Total
log(no2)      9.6146416   7.9916373   9.3336328
log(POPDEN)  -4.3877938  -4.0030750  -4.2270334
per_mixed     3.0705063   2.9298510   3.0158492
per_asian    -5.3039484  -5.0675668  -5.3210765
per_black   -13.8320489 -10.4625933 -13.6940653
per_other    -0.5873709  -0.5831022  -0.5855483

Simulated p-values:
            Direct     Indirect   Total
log(no2)    < 2.22e-16 1.3323e-15 < 2.22e-16
log(POPDEN) 1.1451e-05 6.2524e-05 2.3679e-05
per_mixed   0.002137   0.0033912  0.0025626
per_asian   1.1332e-07 4.0293e-07 1.0315e-07
```

```
per_black   < 2.22e-16 < 2.22e-16 < 2.22e-16
per_other   0.556955   0.5598245  0.5581791
```

The indirect effects in SAR, SAC, and SDM refer to global spillover effects. This means a change of x in the focal units flows through the entire system of neighbours (direct nieightbours, neighbours of neighbours, …) influencing 'their y'. One can think of this as diffusion or a change in a long-term equilibrium.

If Log NO2 increases by one unit, this increases the house price in the focal unit by 0.448 units. Overall, a one unit change in log NO2 increases the house prices in the entire neighbourhood system (direct and higher order neighbours) by 0.754.

For SLX models, nothing is gained from computing the impacts, as they equal the coefficients. Again, it's the effects of direct neighbours only.

```
print(impacts(mod_1.slx, listw = queens.lw))
```

```
Impact measures (SlX, glht):
                 Direct     Indirect        Total
log(no2)    -0.440727458  0.993602103  0.552874645
log(POPDEN) -0.076839828  0.113262218  0.036422390
per_mixed   -0.033042221  0.126068686  0.093026466
per_asian   -0.002380698 -0.003828126 -0.006208824
per_black   -0.016229407 -0.018053503 -0.034282910
per_other   -0.020391354  0.048139008  0.027747654
```

5.4 Examples

Boillat, Ceddia, and Bottazzi (2022)

The paper investigates the effects of protected areas and various land tenure regimes on deforestation and possible spillover effects in Bolivia, a global tropical deforestation hotspot.

Table 3
SDM Models (1–5) estimates of ADE, AIE and ATE expressed in terms of elasticity.

	SDM1			SDM2			SDM3		
	ADE	AIE	ATE	ADE	AIE	ATE	ADE	AIE	ATE
TRAVELTIME	−0.15**	0.11	−0.04	−0.12*	0.18	0.06	−0.01	0.40	0.40**
	(0.06)	(0.19)	(0.22)	(0.06)	(0.20)	(0.22)	(0.35)	(0.40)	(0.19)
SLOPE	−0.29***	−0.31***	−0.60***	−0.28***	−0.32***	−0.60***	−0.29	−0.32	−0.61***
	(0.06)	(0.08)	(0.09)	(0.06)	(0.08)	(0.10)	(0.26)	(0.30)	(0.10)
POPDENS12	−0.004	0.09**	0.09**	0.00	0.09***	0.09**	0.00	0.09	0.10***
	(0.005)	(0.04)	(0.04)	(0.01)	(0.04)	(0.03)	(0.08)	(0.09)	(0.03)
WATERACCESS12	0.44***	1.59***	2.03***	0.41**	1.44***	1.85***	0.44	1.19	1.63***
	(0.15)	(0.35)	(0.35)	(0.15)	(0.35)	(0.35)	(1.04)	(1.16)	(0.32)
POVERTY12	−0.15*	−1.08***	−1.22***	−0.13	−1.06***	−1.19***	−0.16	−0.78	−0.93***
	(0.08)	(0.20)	(0.22)	(0.08)	(0.20)	(0.22)	(0.67)	(0.73)	(0.21)
PROTECTEDAREA	−0.13***	−0.21***	−0.34***	−0.12***	−0.20***	−0.32***	−0.12	−0.16	−0.28***
	(0.03)	(0.05)	(0.07)	(0.03)	(0.05)	(0.07)	(0.14)	(0.14)	(0.06)
NOTITLELAND13	−0.11*	0.85***	0.74***	0.17*	0.95***	1.12***	0.21	0.25	0.47
	(0.06)	(0.18)	(0.21)	(0.10)	(0.36)	(0.38)	(0.25)	(0.36)	(0.40)
INDIGENLAND13	−0.15***	0.22*	0.06	−0.01	0.26	0.25	−0.01	−0.01	−0.01
	(0.04)	(0.12)	(0.13)	(0.05)	(0.19)	(0.20)	(0.05)	(0.18)	(0.20)
COMMONLAND13	0.01	0.19**	0.21**	0.12***	0.23	0.35**	0.12**	0.04	0.15
	(0.03)	(0.09)	(0.10)	(0.04)	(0.16)	(0.17)	(0.05)	(0.15)	(0.17)
STATELAND13	−0.20***	0.28*	0.08	0.04	0.38	0.42	0.05	−0.12	−0.07
	(0.06)	(0.15)	(0.17)	(0.09)	(0.28)	(0.30)	(0.13)	(0.35)	(0.34)
PRIVATELAND13				0.22***	0.11	0.33			
				(0.06)	(0.20)	(0.22)			
SMALLPROP13							0.24***	0.01	0.26***
							(0.03)	(0.06)	(0.07)
MEDIUMPROP13							−0.06	−0.13	−0.19**
							(0.11)	(0.16)	(0.09)
BUSINESSLAND13							0.03	−0.02	0.01
							(0.03)	(0.14)	(0.16)

Standard errors in parentheses.
*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Protected areas – which in Bolivia are all based on co-management schemes - also protect forests in adjacent areas, showing an indirect protective spillover effect. Indigenous lands however only have direct forest protection effects.

Fischer et al. (2009)

The focus of this paper is on the role of human capital in explaining labor productivity variation among 198 European regions within a regression framework.

Table 2 Direct, indirect and total impact estimates (t-statistics in parentheses)

Variables	Spatial Durbin model		
	Mean direct impact	Mean indirect impact	Mean total impact
Initial labor productivity	0.6677	0.0683	0.7361
	(27.5716)	(1.8992)	(26.3921)
Human capital	0.1317	−0.1968	−0.0650
	(6.8644)	(−3.7637)	(−1.1847)

Note: t-statistics based on 10,000 sampled raw parameter estimates of the SDM

A ceteris paribus increase in the level of human capital is found to have a significant and positive direct impact. But this positive direct impact is offset by a significant and negative indirect (spillover) impact leading to a total impact that is not significantly different from zero.

The intuition here arises from the notion that it is relative regional advantages in human capital that matter most for labor productivity, so changing human capital across all regions should have little or no total impact on (average) labor productivity levels.

Rüttenauer (2018)

This study investigates the presence of environmental inequality in Germany - the connection between the presence of foreign-minority population and objectively measured industrial pollution.

Table 3
Community-fixed effects estimates. Dependent variable: Industrial air pollution.

	Overall	Urban	Rural	Diff
	M5	M6	M7	(M6-M7)
% Foreigners	0.062***	0.076***	0.035***	0.041*
	(0.009)	(0.016)	(0.007)	(0.017)
W [% Foreigners]	0.163***	0.391***	0.077***	0.315***
	(0.025)	(0.070)	(0.015)	(0.071)
Population	−0.010	−0.011	−0.008	−0.002
	(0.007)	(0.009)	(0.006)	(0.011)
W [Population]	−0.039	−0.124***	−0.023	−0.100**
	(0.023)	(0.030)	(0.017)	(0.034)
% 65 and older	−0.005	−0.001	−0.000	−0.000
	(0.004)	(0.017)	(0.003)	(0.017)
W [% 65 and older]	−0.009	−0.101	0.004	−0.105
	(0.007)	(0.072)	(0.005)	(0.072)
% Vacant housing	0.014**	0.035	0.012**	0.023
	(0.005)	(0.027)	(0.004)	(0.027)
W [% Vacant housing]	0.021*	0.087	0.020**	0.068
	(0.008)	(0.092)	(0.007)	(0.093)
Living space	−0.012**	0.014	−0.008*	0.022
	(0.004)	(0.022)	(0.003)	(0.022)
W [Living space]	−0.036***	−0.161*	−0.022***	−0.139*
	(0.007)	(0.070)	(0.006)	(0.070)
R^2	0.018	0.076	0.005	
Adj. R^2	−0.031	0.067	−0.050	
Num. obs.	93777	9061	84716	

***$p < 0.001$, **$p < 0.01$, *$p < 0.05$. Standardized coefficients. Cluster robust standard errors in parentheses. W is specified as contiguity neighbours weights matrix. The differences and standard errors in column 4 are obtained by an overall fixed-effects model with interaction terms of the urban dummy and all covariates.

Results reveal that the share of minorities within a census cell indeed positively correlates with the exposure to industrial pollution. Furthermore, spatial spillover effects are highly relevant: the characteristics of the neighbouring spatial units matter in predicting the amount of pollution. Especially within urban areas, clusters of high minority neighbourhoods are affected by high levels of environmental pollution.

5.5 Comparing and Selecting Models

As we have seen, a variety of spatial model specifications exist that can be used to account for the spatial structure of the data. Thus, selecting the correct model specification remains a crucial task in applied research.

One way of selecting the model specification is the application of empirical specification tests. In general, there are two different strategies: a specific-to-general or a general-to-specific approach (Florax, Folmer, and Rey 2003; Mur and Angulo 2009).

5.5.1 Specific-to-general

The specific-to-general approach is more common in spatial econometrics. This approach starts with the most basic non-spatial model and tests for possible misspecifications due to omitted

autocorrelation in the error term or the dependent variable.

Anselin et al. (1996) proposed to use Lagrange multiplier (LM) tests for the hypotheses H_0: $\lambda = 0$ and H_0: $\rho = 0$, which are robust against the alternative source of spatial dependence.

5.5.1.1 Lagrange Multiplier Test

We have earlier talked about methods to detect auto-correlation – visualisation and Moran's I. Both methodscan tell us that there is spatial autocorrelation. However, both method do not provide any information on why there is autocorrelation. Possible reasons:

- Interdependence (ρ)
- Clustering on unobservables (λ)
- Spillovers in covariates ()

Lagrange Multiplier test (Anselin et al. 1996):

- (Robust) test for spatial lag dependence LM_ρ^*
- (Robust) test for spatial error dependence LM_λ^*

Robust test for lag dependence: H_0: $\rho = 0$

$$LM_\rho^* = G^{-1}\hat{\sigma}_\epsilon^2\big(\frac{\hat{\epsilon}^\top \mathbf{W}\mathbf{y}}{\hat{\sigma}_\epsilon^2} - \frac{\hat{\epsilon}^\top \mathbf{W}}{\hat{\sigma}_\epsilon^2}\big)^2 \sim \chi^2$$

Robust test for error dependence: H_0: $\lambda = 0$

$$LM_\lambda^* = \frac{\big(\hat{\epsilon}^\top \mathbf{W}/\hat{\sigma}_\epsilon^2 - [T\hat{\sigma}_\epsilon^2(G + T\hat{\sigma}_\epsilon^2)^{-1}\hat{\epsilon}^\top \mathbf{W}\mathbf{y}/\hat{\sigma}_\epsilon^2\big)^2}{T[1 - \frac{\hat{\sigma}_\epsilon^2}{G+\hat{\sigma}_\epsilon^2}]} \sim \chi^2$$

with

$$G = (\mathbf{W}\mathbf{X})^\top(\mathbf{I} - \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top)(\mathbf{W}\mathbf{X})$$
$$T = \text{tr}[(\mathbf{W}^\top + \mathbf{W})\mathbf{W}],$$

where $\text{tr}(\mathbf{A})$ is the sum of the main diagonal of any square matrix $\mathbf{A}$.

5.5.1.2 Problem

The specific-to-general approach based on the robust LM test offers a good performance in distinguishing between SAR, SEM, and non-spatial OLS (Florax, Folmer, and Rey 2003).

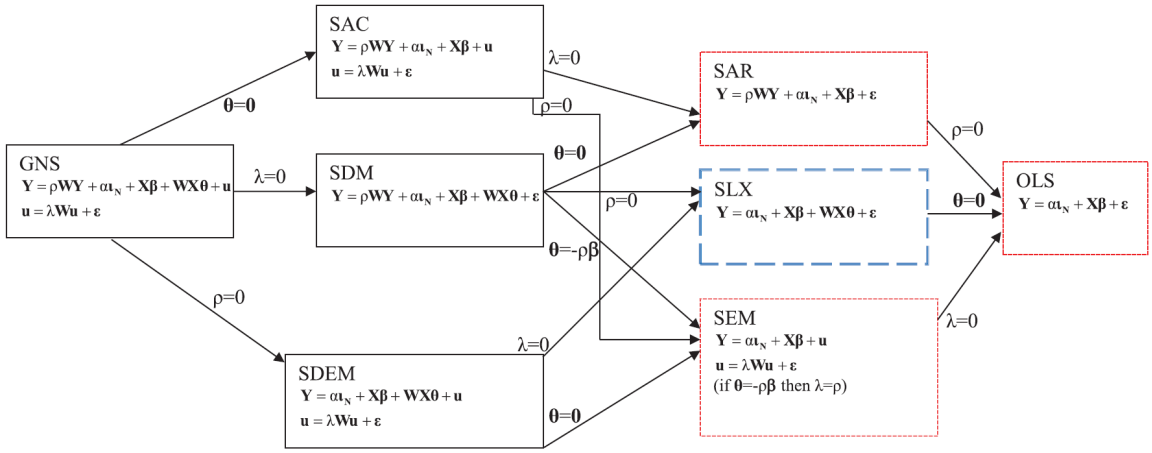
Still, in their original paper, Anselin et al. (1996) already note the declining power of the robust LM_λ test for spatial error dependence with increasing autocorrelation in the dependent variable (indicating some uncertainty under a SAC-like DGP).

Mur and Angulo (2009) demonstrate strong drawbacks of the specific-to-general approach under non-optimal conditions like heteroscedasticity or endogeneity.

Moreover, the test disregard the presence of spatial dependence from local spillover effects (θ is assumed to be zero), as resulting from an SLX-like process. Cook, Hays, and Franzese (2020), for instance, show theoretically that an SLX-like dependence structure leads to the rejection of both hypotheses H_0: $\lambda = 0$ and H_0: $\rho = 0$, though no autocorrelation is present (Elhorst and Halleck Vega 2017; Rüttenauer 2022).

5.5.2 General-to-specific approach

The general-to-specific approach depicts the opposite method of specification search. This approach starts with the most general model and stepwise imposes restrictions on the parameters of this general model.



Note: GNS = general nesting spatial model, SAC = spatial autoregressive combined model, SDM = spatial Durbin model, SDEM = spatial Durbin error model, SAR = spatial autoregressive model, SLX = spatial lag of **X** model, SEM = spatial error model, OLS = ordinary least squares model.

Figure 5.1: Halleck Vega and Elhorst (2015): Nesting of different Spatial Econometric Model Specifications

In theory, we would

1) start with a GNS specification and

2) subsequently restrict the model to simplified specifications based on the significance of parameters in the GNS.

The problem with this strategy is that the GNS is only weakly identified and, thus, is of little help in selecting the correct restrictions (Burridge, Elhorst, and Zigova 2016).

The most intuitive alternative would be to start with one of the two-source models SDM, SDEM, or SAC. This, however, bears the risk of imposing the wrong restriction in the first place (Cook, Hays, and Franzese 2020). Furthermore, Cook, Hays, and Franzese (2020) show that more complicated restrictions are necessary to derive all single-source models from SDEM or SAC specifications.

5.5.3 General advice?

LeSage and Pace (2009), LeSage (2014), Elhorst (2014) argue that there are strong analytical reasons to restrict the model specifications to a subset, as the SDM subsumes the SLX and SAR model, and the SDEM subsumes SLX and SEM.

It is easily observed that SDM reduces to SLX if $\rho = 0$ and to SAR if $\ = 0$, while the SDEM reduces to SLX if $\lambda = 0$ and to SEM if $\ = 0$. Less intuitively, (Anselin 1988) has also shown that the SDM subsumes the SEM. Therefore, we can express the reduced form and rearrange terms:

$$\begin{aligned}
\mathbf{y} &= \mathbf{X} + (\mathbf{I}_N - \lambda\mathbf{W})^{-1} \\
(\mathbf{I}_N - \lambda\mathbf{W})\mathbf{y} &= (\mathbf{I}_N - \lambda\mathbf{W})\mathbf{X} + \\
(\mathbf{I}_N - \lambda\mathbf{W})\mathbf{y} &= \mathbf{X} - \lambda\mathbf{W}\mathbf{X} + \\
\mathbf{y} &= (\mathbf{I}_N - \lambda\mathbf{W})^{-1}(\mathbf{X} + \mathbf{W}\mathbf{X} +).
\end{aligned}$$

Thus, the SEM constitutes a special case of an SDM with the relative simple restriction $\ = -\lambda$, meaning direct and indirect effects are constrained to a common factor (Anselin 1988, 2003).

The fact that SDM subsumes SAR, SLX, and SEM leads to the conclusion that applied research should only consider SDM and SDEM as model specifications (LeSage 2014). Especially in the case of a likely omitted variable bias, (LeSage and Pace 2009, ~68) argue in favour of using the SDM.

Nonetheless, others propose to use the SLX specification as point of departure (Gibbons and Overman 2012; Halleck Vega and Elhorst 2015). First, scholars have argued that SAC and SDM models are only weakly identified in practice (Gibbons and Overman 2012; Pinkse and Slade 2010). Second, the global spillover specification in SAR, SAC, and SDM often seems to be theoretically implausible.

And finally:

5.5.4 Design and Theory

Some argue that the best way of choosing the appropriate model specification is to exclude one or more sources of spatial dependence – autocorrelation in the dependent variable, autocorrelation in the disturbances, or spatial spillover effects of the covariates – by design Gibbons, Overman, and Patacchini (2015).

Natural experiments are probably the best way of making one or more sources of spatial dependence unlikely, thereby restricting the model alternatives to a subset of all available models. However, the opportunities to use natural experiments are restricted in social sciences, making it a favourable but often impractical way of model selection.

Cook, Hays, and Franzese (2020) and Rüttenauer (2022) argue that theoretical considerations should guide the model selection.

1) Rule out some sources of spatial dependence by theory, and thus restrict the specifications to a subset (*Where does the spatial dependence come from?*),
2) Theoretical mechanisms may guide the choice of either global or local spillover effects.

5.5.5 SLX is robust

Rüttenauer (2022) provides Monte Carlo experiments on how different spatial models perform under different scenarios of misspecification. SLX - despite its simplicity - does a very good job even if it is the wrong model for the specific situation. I personally advice for SLX (or SDEM) as the most useful specification if you do not have theoretical assumptions about spatial interdependence in the outcome.

Having a spatially lagged dependent variable comes with an extra layer of complexity. I suggest to only use these models if you have a-priori reasons to believe that it is the correct specification for your study.

Figure 5.2: "I will use spatial lags of X, not spatial lags of Y", J. Wooldridge on twitter

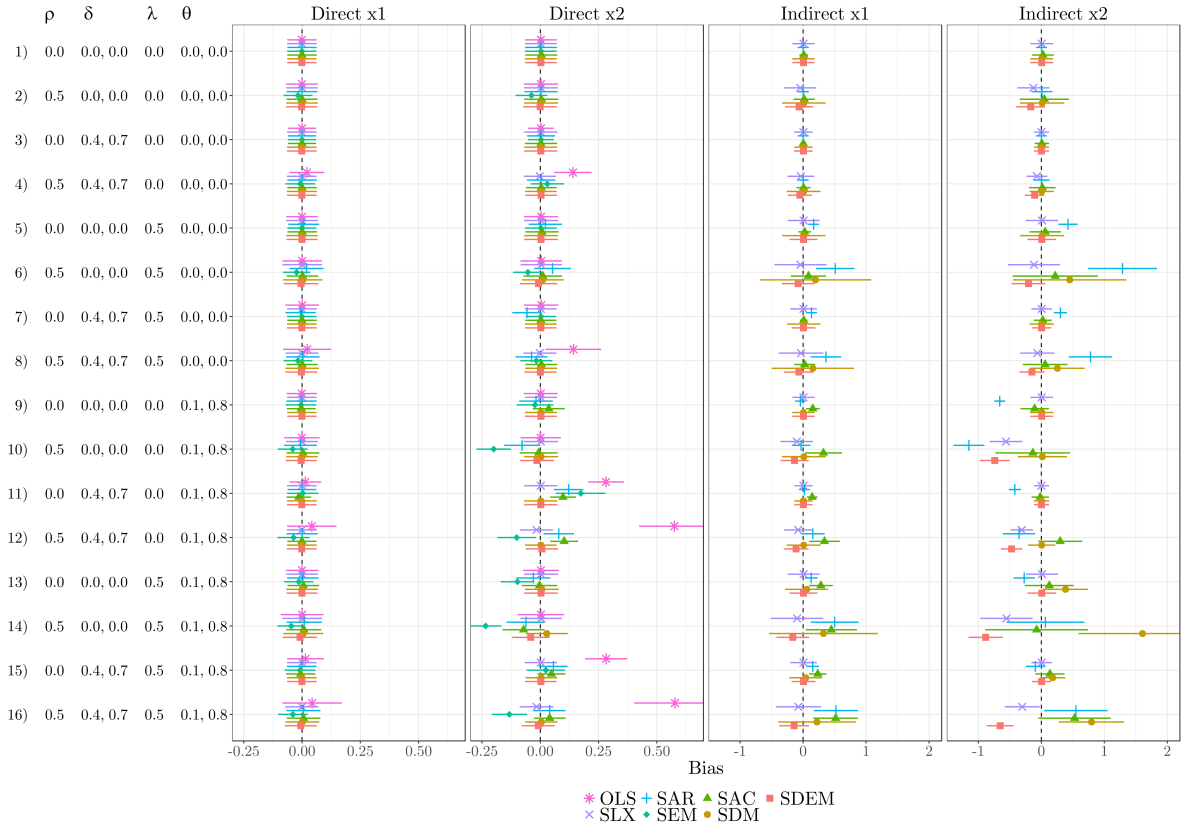


Figure 5.3: Bias of impacts and 95% confidence interval of empirical standard deviation without omv: $= (0.2, 0.5)^\top$, $= (0, 0)^\top$. ρ = autocorrelation in the dependent variable ($\mathbf{Wy}$); = autocorrelation in the covariates ($\mathbf{x}_k = f(\mathbf{Wx}_k)$); λ = autocorrelation in the disturbances ($\mathbf{Wu}$); = spatial spillover effects of covariates ($\mathbf{WX}$); = strength of omv.

6 Exercise II

Required packages

```
pkgs <- c("sf", "mapview", "spdep", "spatialreg", "tmap", "viridisLite") # note: load spde
lapply(pkgs, require, character.only = TRUE)
```

Session info

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default


locale:
[1] LC_COLLATE=English_United Kingdom.utf8
[2] LC_CTYPE=English_United Kingdom.utf8
[3] LC_MONETARY=English_United Kingdom.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.utf8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] viridisLite_0.4.2 tmap_3.3-3        spatialreg_1.2-9  Matrix_1.5-4.1
[5] spdep_1.2-8       spData_2.3.0      mapview_2.11.0    sf_1.0-13
```

```
loaded via a namespace (and not attached):
 [1] xfun_0.39          raster_3.6-23      htmlwidgets_1.6.2  lattice_0.21-8
 [5] vctrs_0.6.3        tools_4.3.1        crosstalk_1.2.0    LearnBayes_2.15.1
 [9] generics_0.1.3     parallel_4.3.1     sandwich_3.0-2     stats4_4.3.1
[13] tibble_3.2.1       proxy_0.4-27       fansi_1.0.4        pkgconfig_2.0.3
[17] KernSmooth_2.23-21 satellite_1.0.4    RColorBrewer_1.1-3 leaflet_2.1.2
[21] webshot_0.5.5      lifecycle_1.0.3    compiler_4.3.1     deldir_1.0-9
[25] munsell_0.5.0      terra_1.7-39       leafsync_0.1.0     codetools_0.2-
19
[29] stars_0.6-1        htmltools_0.5.5    class_7.3-22       pillar_1.9.0
[33] MASS_7.3-60        classInt_0.4-9     lwgeom_0.2-13      wk_0.7.3
[37] abind_1.4-5        boot_1.3-28.1      multcomp_1.4-25    nlme_3.1-162
[41] tidyselect_1.2.0   digest_0.6.32      mvtnorm_1.2-2      dplyr_1.1.2
[45] splines_4.3.1      fastmap_1.1.1      grid_4.3.1         colorspace_2.1-
0
[49] expm_0.999-7       cli_3.6.1          magrittr_2.0.3     base64enc_0.1-3
[53] dichromat_2.0-0.1  XML_3.99-0.14      survival_3.5-5     utf8_1.2.3
[57] TH.data_1.1-2      leafem_0.2.0       e1071_1.7-13       scales_1.2.1
[61] sp_1.6-1           rmarkdown_2.23     zoo_1.8-12         png_0.1-8
[65] coda_0.19-4        evaluate_0.21      knitr_1.43         tmaptools_3.1-1
[69] s2_1.1.4           rlang_1.1.1        Rcpp_1.0.10        glue_1.6.2
[73] DBI_1.1.3          rstudioapi_0.14    jsonlite_1.8.5     R6_2.5.1
[77] units_0.8-2
```

Reload data from pervious session

```
load("_data/msoa2_spatial.RData")
```

6.1 Environmental inequality

How would you investigate the following descriptive research question: Are ethnic (and immigrant) minorities in London exposed to higher levels of pollution? Also consider the spatial structure. What's your dependent and whats your independent variable?

1) Define a neigbours weights object of your choice

Assume a typical neighbourhood would be 2.5km in diameter

```r
coords <- st_centroid(msoa.spdf)
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```r
# Neighbours within 3km distance
dist_15.nb <- dnearneigh(coords, d1 = 0, d2 = 2500)

summary(dist_15.nb)
```

```
Neighbour list object:
Number of regions: 983
Number of nonzero links: 15266
Percentage nonzero weights: 1.579859
Average number of links: 15.53001
4 regions with no links:
158 463 478 505
Link number distribution:

 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 4  5  9 23 19 26 36 31 53 39 61 63 59 48 42 35 24 31 28 30 27 26 25 19 38 29
26 27 28 29 30 31 32 33 34
32 38 26 16 20 10  8  1  2
5 least connected regions:
160 469 474 597 959 with 1 link
2 most connected regions:
565 567 with 34 links
```

```r
# There are some mpty one. Lets impute with the nearest neighbour
k2.nb <- knearneigh(coords, k = 1)

# Replace zero
nolink_ids <- which(card(dist_15.nb) == 0)
dist_15.nb[card(dist_15.nb) == 0] <- k2.nb$nn[nolink_ids, ]

summary(dist_15.nb)
```

```
Neighbour list object:
Number of regions: 983
Number of nonzero links: 15270
```

```
Percentage nonzero weights: 1.580273
Average number of links: 15.53408
Link number distribution:

 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 9  9 23 19 26 36 31 53 39 61 63 59 48 42 35 24 31 28 30 27 26 25 19 38 29 32
27 28 29 30 31 32 33 34
38 26 16 20 10  8  1  2
9 least connected regions:
158 160 463 469 474 478 505 597 959 with 1 link
2 most connected regions:
565 567 with 34 links
```

```r
# listw object with row-normalization
dist_15.lw <- nb2listw(dist_15.nb, style = "W")
```

2) Estimate the extent of spatial auto-correlation

```r
moran.test(msoa.spdf$no2, listw = dist_15.lw)
```

```
    Moran I test under randomisation

data:  msoa.spdf$no2
weights: dist_15.lw

Moran I statistic standard deviate = 65.197, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.891520698      -0.001018330       0.000187411
```

3) Estimate a spatial SAR regression model

a) Estimate a spatial autoregressive SAR model

```r
mod_1.sar <- lagsarlm(log(no2) ~ per_mixed + per_asian + per_black + per_other
                      + per_nonUK_EU + per_nonEU  + log(POPDEN),
                      data = msoa.spdf,
```

```
                      listw = dist_15.lw,
                      Durbin = FALSE) # we could here extend to SDM
summary(mod_1.sar)
```

```
Call:lagsarlm(formula = log(no2) ~ per_mixed + per_asian + per_black +
    per_other + per_nonUK_EU + per_nonEU + log(POPDEN), data = msoa.spdf,
    listw = dist_15.lw, Durbin = FALSE)

Residuals:
       Min         1Q     Median         3Q        Max
-0.2140485 -0.0267085 -0.0021421  0.0238337  0.3505513

Type: lag
Coefficients: (asymptotic standard errors)
                Estimate  Std. Error z value  Pr(>|z|)
(Intercept)  -1.7004e-02  1.8122e-02 -0.9383  0.348110
per_mixed     3.4376e-04  1.4758e-03  0.2329  0.815810
per_asian    -8.5205e-05  1.1494e-04 -0.7413  0.458507
per_black    -4.2754e-04  2.3468e-04 -1.8218  0.068484
per_other     1.9693e-03  7.4939e-04  2.6279  0.008591
per_nonUK_EU  8.9027e-04  3.9638e-04  2.2460  0.024703
per_nonEU     1.8460e-03  3.5159e-04  5.2506 1.516e-07
log(POPDEN)   1.8650e-02  2.7852e-03  6.6963 2.138e-11

Rho: 0.9684, LR test value: 2002.5, p-value: < 2.22e-16
Asymptotic standard error: 0.0063124
    z-value: 153.41, p-value: < 2.22e-16
Wald statistic: 23535, p-value: < 2.22e-16

Log likelihood: 1562.401 for lag model
ML residual variance (sigma squared): 0.0020568, (sigma: 0.045352)
Number of observations: 983
Number of parameters estimated: 10
AIC: -3104.8, (AIC for lm: -1104.3)
LM test for residual autocorrelation
test value: 108.97, p-value: < 2.22e-16
```

b) Have a look into the true multiplier matrix $(\mathbf{I}_N - \rho\mathbf{W})^{-1}\beta_k$

```
W <- listw2mat(dist_15.lw)
I <- diag(dim(W)[1])

rho <- unname(mod_1.sar$rho)

M <- solve(I - rho*W)

M[1:10, 1:10]
```

```
              1           2           3           4           5           6
 [1,] 1.164650997 0.002433319 0.004089559 0.004034508 0.006545994 0.004882511
 [2,] 0.010706605 1.407336301 0.643881932 0.370049927 0.464794934 0.385105188
 [3,] 0.011246286 0.402426207 1.474021599 0.429011868 0.641526285 0.566175019
 [4,] 0.008875918 0.185024963 0.343209495 1.684533322 0.614086824 0.631452476
 [5,] 0.012000989 0.193664556 0.427684190 0.511739020 1.560840834 0.560656534
 [6,] 0.010741524 0.192552594 0.452940016 0.631452476 0.672787841 1.571175245
 [7,] 0.012779708 0.141953871 0.299247377 0.418234186 0.616895800 0.558170218
 [8,] 0.014769006 0.125781189 0.253122442 0.295553039 0.500919513 0.357799398
 [9,] 0.011708131 0.147549264 0.309080773 0.568442619 0.629156269 0.601077237
[10,] 0.009937859 0.152900148 0.306652041 0.727001926 0.553973310 0.679775059
              7          8           9          10
 [1,] 0.005808958 0.00872714 0.005854065 0.003613767
 [2,] 0.283907742 0.32703109 0.324608380 0.244640236
 [3,] 0.374059222 0.41132397 0.424986063 0.306652041
 [4,] 0.418234186 0.38421895 0.625286881 0.581601541
 [5,] 0.514079833 0.54266281 0.576726579 0.369315540
 [6,] 0.558170218 0.46513922 0.661184961 0.543820047
 [7,] 1.475511568 0.58520461 0.614170880 0.463886540
 [8,] 0.450157392 1.46638195 0.474994894 0.272339890
 [9,] 0.558337164 0.56135760 1.581077095 0.517983092
[10,] 0.579858174 0.44255232 0.712226751 1.560083138
```

c) Create an $N \times N$ effects matrix. What is the effect of unit 6 on unit 10?

```
# For beta 1

beta <- mod_1.sar$coefficients

effM <- beta[2] * M

effM[1:10, 1:10]
```

```
                1            2            3            4            5
 [1,] 4.003610e-04 8.364789e-07 1.405829e-06 1.386904e-06 2.250254e-06
 [2,] 3.680507e-06 4.837866e-04 2.213411e-04 1.272085e-04 1.597781e-04
 [3,] 3.866028e-06 1.383382e-04 5.067103e-04 1.474773e-04 2.205314e-04
 [4,] 3.051190e-06 6.360427e-05 1.179819e-04 5.790759e-04 2.110988e-04
 [5,] 4.125465e-06 6.657422e-05 1.470209e-04 1.759156e-04 5.365554e-04
 [6,] 3.692511e-06 6.619197e-05 1.557029e-04 2.170684e-04 2.312779e-04
 [7,] 4.393158e-06 4.879813e-05 1.028694e-04 1.437724e-04 2.120644e-04
 [8,] 5.077000e-06 4.323860e-05 8.701349e-05 1.015994e-04 1.721963e-04
 [9,] 4.024792e-06 5.072160e-05 1.062497e-04 1.954081e-04 2.162790e-04
[10,] 3.416243e-06 5.256102e-05 1.054148e-04 2.499145e-04 1.904341e-04
                6            7            8            9           10
 [1,] 1.678414e-06 1.996890e-06 3.000045e-06 2.012396e-06 1.242270e-06
 [2,] 1.323839e-04 9.759625e-05 1.124204e-04 1.115875e-04 8.409764e-05
 [3,] 1.946286e-04 1.285868e-04 1.413969e-04 1.460934e-04 1.054148e-04
 [4,] 2.170684e-04 1.437724e-04 1.320793e-04 2.149489e-04 1.999316e-04
 [5,] 1.927315e-04 1.767203e-04 1.865460e-04 1.982558e-04 1.269561e-04
 [6,] 5.401079e-04 1.918768e-04 1.598965e-04 2.272892e-04 1.869438e-04
 [7,] 1.918768e-04 5.072225e-04 2.011702e-04 2.111277e-04 1.594658e-04
 [8,] 1.229973e-04 1.547463e-04 5.040841e-04 1.632845e-04 9.361968e-05
 [9,] 2.066266e-04 1.919342e-04 1.929725e-04 5.435118e-04 1.780621e-04
[10,] 2.336798e-04 1.993323e-04 1.521320e-04 2.448354e-04 5.362949e-04
```

```r
# "Effect" of unit 6 on unit 10
effM[10, 6]
```

```
          6
0.0002336798
```

d) Estimate a spatial autoregressive SLX model

```r
mod_1.slx <- lmSLX(log(no2) ~ per_mixed + per_asian + per_black + per_other
                       + per_nonUK_EU + per_nonEU  + log(POPDEN),
                       data = msoa.spdf,
                       listw = dist_15.lw,
                   Durbin = TRUE)
```

e) Calculate and interpret the summary impact measures for SAR and SLX.

```r
mod_1.sar.imp <- impacts(mod_1.sar, listw = dist_15.lw, R = 300)
summary(mod_1.sar.imp)
```

```
Impact measures (lag, exact):
                   Direct     Indirect        Total
per_mixed     0.0004939013  0.010385844  0.010879745
per_asian    -0.0001224192 -0.002574253 -0.002696672
per_black    -0.0006142789 -0.012917166 -0.013531445
per_other     0.0028294759  0.059498722  0.062328198
per_nonUK_EU  0.0012791011  0.026897166  0.028176267
per_nonEU     0.0026523198  0.055773451  0.058425770
log(POPDEN)   0.0267960076  0.563471199  0.590267206
============================================================
Simulation results ( variance matrix):
Direct:

Iterations = 1:300
Thinning interval = 1
Number of chains = 1
Sample size per chain = 300

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                  Mean        SD  Naive SE Time-series SE
per_mixed     0.0006267 0.0021782 1.258e-04      1.258e-04
per_asian    -0.0001126 0.0001635 9.441e-06      9.441e-06
per_black    -0.0006168 0.0003425 1.977e-05      1.977e-05
per_other     0.0028385 0.0010308 5.951e-05      5.461e-05
per_nonUK_EU  0.0012045 0.0005465 3.155e-05      3.155e-05
per_nonEU     0.0026788 0.0005059 2.921e-05      2.921e-05
log(POPDEN)   0.0271287 0.0039427 2.276e-04      2.276e-04

2. Quantiles for each variable:

                  2.5%        25%        50%        75%     97.5%
per_mixed    -0.0033288 -0.0008314  0.0007611  2.142e-03 4.852e-03
per_asian    -0.0004104 -0.0002294 -0.0001095  3.756e-06 2.138e-04
per_black    -0.0012788 -0.0008505 -0.0006017 -3.793e-04 4.158e-05
per_other     0.0008085  0.0021219  0.0028927  3.534e-03 4.739e-03
per_nonUK_EU  0.0002291  0.0008074  0.0011927  1.582e-03 2.249e-03
per_nonEU     0.0017088  0.0023761  0.0026564  3.011e-03 3.728e-03
log(POPDEN)   0.0193241  0.0245104  0.0269354  2.946e-02 3.541e-02

============================================================
Indirect:
```

```
Iterations = 1:300
Thinning interval = 1
Number of chains = 1
Sample size per chain = 300

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                  Mean       SD  Naive SE Time-series SE
per_mixed     0.012133 0.049925 0.0028824      0.0028824
per_asian    -0.002541 0.003758 0.0002170      0.0002170
per_black    -0.013588 0.008733 0.0005042      0.0005042
per_other     0.062066 0.031705 0.0018305      0.0018305
per_nonUK_EU  0.026111 0.013457 0.0007769      0.0007769
per_nonEU     0.059133 0.021725 0.0012543      0.0010592
log(POPDEN)   0.589830 0.162710 0.0093941      0.0093941

2. Quantiles for each variable:

                  2.5%       25%       50%        75%     97.5%
per_mixed    -0.091121 -0.018523  0.015275  4.513e-02 0.1025182
per_asian    -0.009833 -0.004656 -0.002375  7.168e-05 0.0043781
per_black    -0.034631 -0.017951 -0.012726 -7.796e-03 0.0008555
per_other     0.016314  0.045133  0.060168  7.376e-02 0.1125443
per_nonUK_EU  0.004816  0.017386  0.025633  3.299e-02 0.0528933
per_nonEU     0.031068  0.046906  0.055099  6.805e-02 0.1020833
log(POPDEN)   0.383704  0.492660  0.570270  6.558e-01 0.9321412

============================================================
Total:

Iterations = 1:300
Thinning interval = 1
Number of chains = 1
Sample size per chain = 300

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                  Mean       SD  Naive SE Time-series SE
per_mixed     0.012760 0.052016 0.0030032      0.0030032
per_asian    -0.002653 0.003916 0.0002261      0.0002261
```

```
per_black    -0.014205 0.009034 0.0005216      0.0005216
per_other     0.064905 0.032463 0.0018743      0.0018743
per_nonUK_EU  0.027316 0.013920 0.0008037      0.0008037
per_nonEU     0.061812 0.022053 0.0012733      0.0010754
log(POPDEN)   0.616959 0.164249 0.0094829      0.0094829

2. Quantiles for each variable:

                  2.5%       25%       50%        75%     97.5%
per_mixed    -0.095235 -0.019287  0.016059  4.736e-02 0.1067967
per_asian    -0.010196 -0.004932 -0.002476  7.543e-05 0.0046186
per_black    -0.035595 -0.018761 -0.013256 -8.182e-03 0.0008971
per_other     0.017158  0.047454  0.063402  7.716e-02 0.1173944
per_nonUK_EU  0.005058  0.018192  0.026749  3.454e-02 0.0550370
per_nonEU     0.032891  0.049075  0.057995  7.078e-02 0.1056684
log(POPDEN)   0.406268  0.517484  0.596506  6.832e-01 0.9635228
```

For SLX, you can just interpret the coefficients. Impacts will give you the same results.

4) Is SAR the right model choice or would you rather estimate a different model?

f) How do results change once you specify a spatial Durbin model?

g) Please calculate and interpret the impacts for a spatial Durbin model.

6.2 Life Expecatancy in Germany

Below, we read and transform some characteristics of the INKAR data on German counties.

```
load("_data/inkar2.Rdata")
```

Variables are

Variable	Description
"Kennziffer"	ID
"Raumeinheit"	Name
"Aggregat"	Level
"year"	Year
"poluation_density"	Population Density
"median_income"	Median Household income (only for 2020)

Variable	Description
“gdp_in1000EUR”	Gross Domestic Product in 1000 euros
“unemployment_rate”	Unemployment rate
“share_longterm_unemployed”	Share of longterm unemployed (among unemployed)
“share_working_indutry”	Share of employees in undistrial sector
“share_foreigners”	Share of foreign nationals
“share_college”	Share of school-finishers with college degree
“recreational_space”	Recreational space per inhabitant
“car_density”	Density of cars
“life_expectancy”	Life expectancy

And we get the respective county shapes:

```
kreise.spdf <- st_read(dsn = "_data/vg5000_ebenen_1231",
                       layer = "VG5000_KRS")
```

```
Reading layer `VG5000_KRS' from data source
  `C:\work\Lehre\Geodata_Spatial_Regression_short\_data\vg5000_ebenen_1231'
  using driver `ESRI Shapefile'
Simple feature collection with 400 features and 24 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 280353.1 ymin: 5235878 xmax: 921261.6 ymax: 6101302
Projected CRS: ETRS89 / UTM zone 32N
```

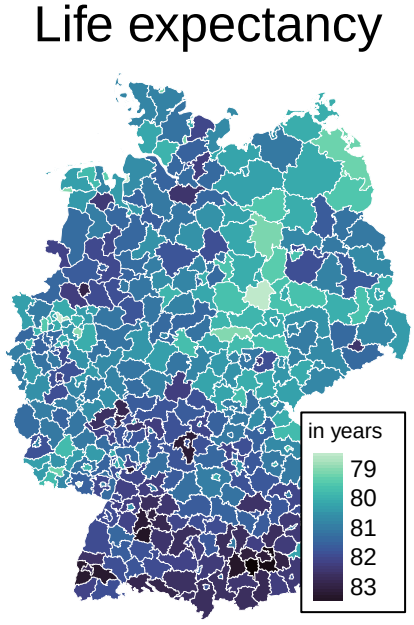
6.2.1 1) Merge data with the shape file (as with conventional data)

```
# Merge
inkar_2020.spdf <- merge(kreise.spdf, inkar.df[inkar.df$year == 2020, ],
                         by.x = "AGS", by.y = "Kennziffer")
```

6.2.2 2) Create a map of life-expectancy

```
cols <- viridis(n = 100, direction = -1, option = "G")

mp1 <-  tm_shape(inkar_2020.spdf) +
```

```
  tm_fill(col = "life_expectancy",
          style = "cont", # algorithm to def cut points
          palette = cols, # colours
          stretch.palette = TRUE,
          title = "in years"
          ) +
  tm_borders(col = "white", lwd = 0.5, alpha = 0.5) +
  tm_layout(frame = FALSE,
            legend.frame = TRUE, legend.bg.color = TRUE,
            legend.position = c("right", "bottom"),
            legend.outside = FALSE,
            main.title = "Life expectancy",
            main.title.position = "center",
            main.title.size = 1.6,
            legend.title.size = 0.8,
            legend.text.size = 0.8)

mp1
```



6.2.3 3) Chose some variables that could predict life expectancy. See for instance the following paper.

6.2.4 4) Generate a neighbours object (e.g. the 10 nearest neighbours).

```
# nb <- poly2nb(kreise.spdf, row.names = "ags", queen = TRUE)
knn <- knearneigh(st_centroid(kreise.spdf), k = 10)
```

```
Warning: st_centroid assumes attributes are constant over geometries
```

```
nb <- knn2nb(knn, row.names = kreise.spdf$ags)
listw <- nb2listw(nb, style = "W")
```

6.2.5 5) Estimate a cross-sectional spatial model for the year 2020 and calculate the impacts.

```
### Use a spatial Durbin Error model

# Spec formula
fm <- life_expectancy ~ median_income + unemployment_rate + share_college + car_density

# Estimate error model with Durbin = TRUE
mod_1.durb <- errorsarlm(fm,
                         data = inkar_2020.spdf,
                         listw = listw,
                         Durbin = TRUE)

summary(mod_1.durb)
```

```
Call:errorsarlm(formula = fm, data = inkar_2020.spdf, listw = listw,
    Durbin = TRUE)

Residuals:
      Min        1Q    Median        3Q       Max
-1.343988 -0.349564  0.013309  0.333105  1.819014

Type: error
```

```
Coefficients: (asymptotic standard errors)
                          Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)            8.4970e+01  1.4366e+00  59.1460 < 2.2e-16
median_income          5.4013e-04  8.2285e-05   6.5642 5.233e-11
unemployment_rate     -3.8970e-01  2.0095e-02 -19.3923 < 2.2e-16
share_college          6.7806e-03  3.2502e-03   2.0862  0.036957
car_density           -3.2042e-03  4.9774e-04  -6.4376 1.214e-10
lag.median_income      4.9282e-04  1.8112e-04   2.7209  0.006510
lag.unemployment_rate -3.4685e-02  4.5454e-02  -0.7631  0.445415
lag.share_college     -1.7066e-03  7.0324e-03  -0.2427  0.808256
lag.car_density       -5.2210e-03  1.7540e-03  -2.9766  0.002915

Lambda: 0.57895, LR test value: 48.146, p-value: 3.9563e-12
Asymptotic standard error: 0.069524
    z-value: 8.3274, p-value: < 2.22e-16
Wald statistic: 69.345, p-value: < 2.22e-16

Log likelihood: -305.6855 for error model
ML residual variance (sigma squared): 0.26001, (sigma: 0.50991)
Number of observations: 400
Number of parameters estimated: 11
AIC: NA (not available for weighted model), (AIC for lm: 679.52)
```

```r
# Calculate impacts (which is unnecessary in this case)
mod_1.durb.imp <- impacts(mod_1.durb, listw = listw, R = 300)
summary(mod_1.durb.imp, zstats = TRUE, short = TRUE)
```

```
Impact measures (SDEM, glht, n):
                          Direct      Indirect        Total
median_income      0.0005401288  0.0004928219  0.001032951
unemployment_rate -0.3896966870 -0.0346850573 -0.424381744
share_college      0.0067806074 -0.0017065824  0.005074025
car_density       -0.0032042421 -0.0052210252 -0.008425267
========================================================
Standard errors:
                         Direct     Indirect        Total
median_income     8.228463e-05 0.0001811231 0.0001813201
unemployment_rate 2.009540e-02 0.0454539507 0.0455753453
share_college     3.250157e-03 0.0070323527 0.0069549479
car_density       4.977409e-04 0.0017540485 0.0018942462
========================================================
```

```
Z-values:
                      Direct   Indirect      Total
median_income       6.564152  2.7209218  5.6968356
unemployment_rate -19.392336 -0.7630812 -9.3116518
share_college       2.086240 -0.2426759  0.7295561
car_density        -6.437570 -2.9765569 -4.4478205

p-values:
                  Direct     Indirect  Total
median_income     5.233e-11  0.0065100 1.2205e-08
unemployment_rate < 2.22e-16 0.4454149 < 2.22e-16
share_college     0.036957   0.8082565 0.46566
car_density       1.214e-10  0.0029151 8.6746e-06
```

6.2.6 6) Calculate the spatial lagged variables for your covariates (e.g. use create_WX(), which needs a non-spatial df as input) .

```r
# Extract covariate names
covars <- attr(terms(fm),"term.labels")

w_vars <- create_WX(st_drop_geometry(inkar_2020.spdf)[, covars],
                    listw = listw,
                    prefix = "w")

inkar_2020.spdf <- cbind(inkar_2020.spdf, w_vars)
```

6.2.7 6) Can you run a spatial machine learning model? (for instance, using randomForest)?

```r
library(randomForest)
```

```
Warning: package 'randomForest' was built under R version 4.3.2
```

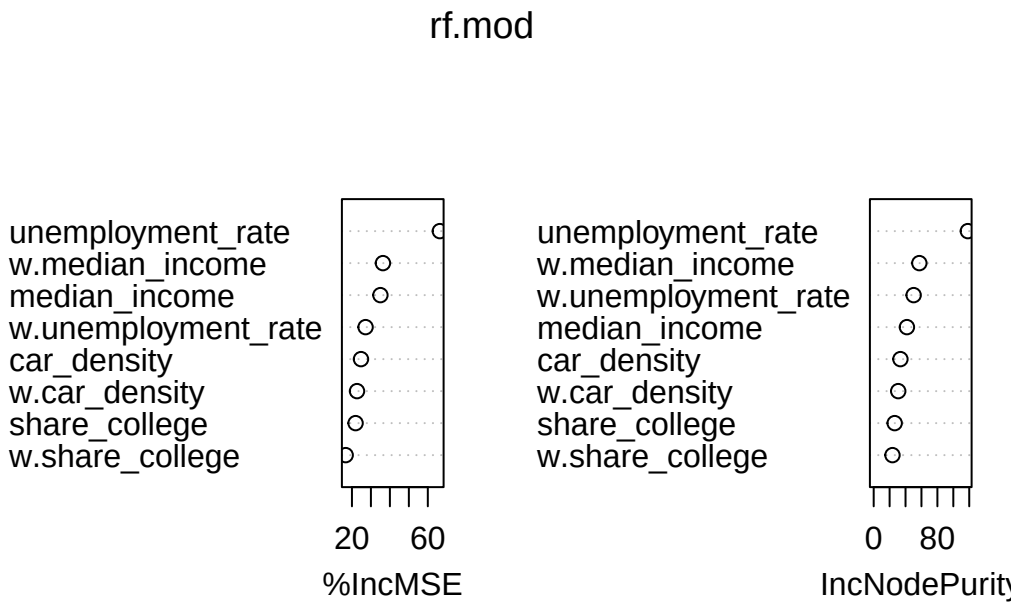
```
randomForest 4.7-1.1
```

```
Type rfNews() to see new features/changes/bug fixes.
```

```r
# Train
rf.mod <- randomForest(life_expectancy ~ median_income + unemployment_rate + share_college
                         w.median_income + w.unemployment_rate + w.share_college + w.car_d
                       data = st_drop_geometry(inkar_2020.spdf),
                       ntree = 1000,
                       importance = TRUE)

# Inspect the mechanics of the model
importance(rf.mod)
```

```
                    %IncMSE IncNodePurity
median_income       35.00535      41.72648
unemployment_rate   66.34426     118.24758
share_college       21.91225      26.35903
car_density         24.75672      33.58266
w.median_income     36.33925      57.35562
w.unemployment_rate 27.17627      50.19094
w.share_college     16.68452      23.58819
w.car_density       22.66050      30.82173
```

```r
varImpPlot(rf.mod)
```



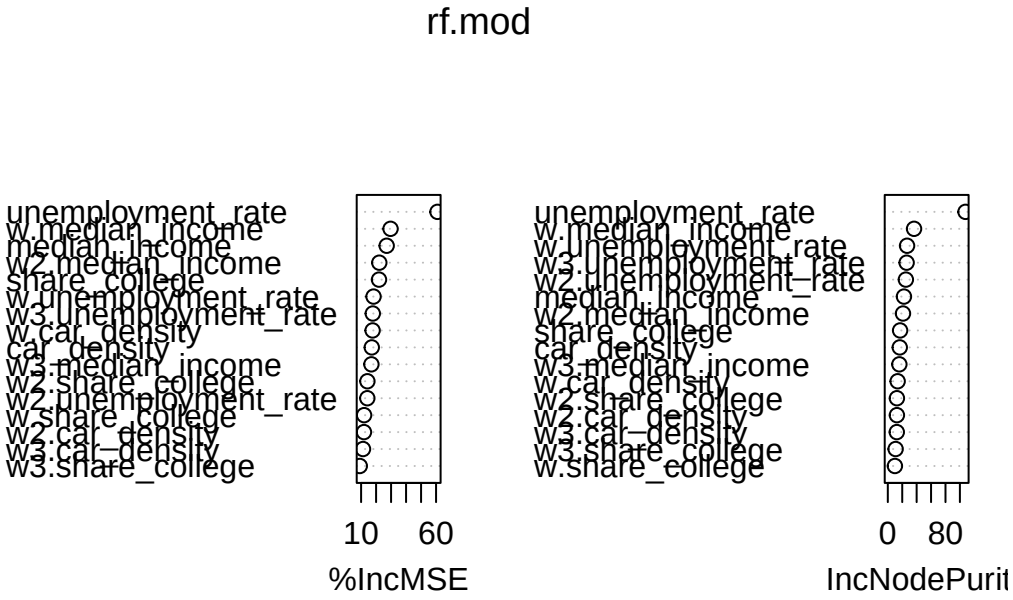
You could even go further and use higher order neighbours (e.g. `nblag(queens.nb, maxlag = 3)`) to check the importance of direct neighbours and the neighbours neighbours and so on …

```
# Create higher order NB object
listw.lag <- nblag(nb, maxlag = 3)


# Create listwise of 1st, 2nd and 3rd order neighbours
listw.lw1 <- nb2listw(listw.lag[[1]], style = "W")
listw.lw2 <- nb2listw(listw.lag[[2]], style = "W")
listw.lw3 <- nb2listw(listw.lag[[3]], style = "W")

# Create lagged X
w_vars2 <- create_WX(st_drop_geometry(inkar_2020.spdf)[, covars],
                     listw = listw.lw2,
                     prefix = "w2")

w_vars3 <- create_WX(st_drop_geometry(inkar_2020.spdf)[, covars],
                     listw = listw.lw3,
                     prefix = "w3")

inkar_2020.spdf <- cbind(inkar_2020.spdf, w_vars2, w_vars3)

# Train
rf.mod <- randomForest(life_expectancy ~ median_income + unemployment_rate + share_college
                         w.median_income + w.unemployment_rate + w.share_college + w.car_d
                         w2.median_income + w2.unemployment_rate + w2.share_college + w2.c
                         w3.median_income + w3.unemployment_rate + w3.share_college + w3.c
                       data = st_drop_geometry(inkar_2020.spdf),
                       ntree = 1000,
                       importance = TRUE)

# Inspect the mechanics of the model
importance(rf.mod)
```

```
                     %IncMSE IncNodePurity
median_income        27.056741     22.381907
unemployment_rate    60.956594    107.776730
share_college        21.940614     17.067775
car_density          17.113650     16.574191
w.median_income      29.638337     36.383690
w.unemployment_rate  18.291600     26.712941
```

```
w.share_college       11.993833      9.829905
w.car_density         17.676351     13.659523
w2.median_income      22.125912     21.053658
w2.unemployment_rate  14.149862     25.031902
w2.share_college      14.196129     12.618829
w2.car_density        11.906223     12.602829
w3.median_income      16.862951     16.008329
w3.unemployment_rate  17.790593     26.111921
w3.share_college       9.131997     10.845357
w3.car_density        11.264291     12.450566
```

```r
varImpPlot(rf.mod)
```



References

Anselin, Luc. 1988. *Spatial Econometrics: Methods and Models*. Studies in Operational Regional Science. Dordrecht: Kluwer.

———. 1995. "Local Indicators of Spatial Association-LISA." *Geographical Analysis* 27 (2): 93–115. https://doi.org/10.1111/j.1538-4632.1995.tb00338.x.

———. 2003. "Spatial Externalities, Spatial Multipliers, and Spatial Econometrics." *International Regional Science Review* 26 (2): 153–66. https://doi.org/10.1177/0160017602250972.

Anselin, Luc, and Anil K. Bera. 1998. "Spatial Dependence in Linear Regression Models with an Introduction to Spatial Econometrics." In *Handbook of Applied Economic Statistics*, edited by Aman Ullah and David E. A. Giles, 237–89. New York: Dekker.

Anselin, Luc, Anil K. Bera, Raymond Florax, and Mann J. Yoon. 1996. "Simple Diagnostic Tests for Spatial Dependence." *Regional Science and Urban Economics* 26 (1): 77–104. https://doi.org/10.1016/0166-0462(95)02111-6.

Appelhans, Tim, Florian Detsch, Chritoph Reudenbach, and Stefan Woellauer. 2021. "Mapview: Interactive Viewing of Spatial Data in R."

Betz, Timm, Scott J. Cook, and Florian M. Hollenbach. 2020. "Spatial Interdependence and Instrumental Variable Models." *Political Science Research and Methods* 8 (4): 646–61. https://doi.org/10.1017/psrm.2018.61.

Bivand, Roger S., and Colin Rudel. 2018. "Rgeos: Interface to Geometry Engine - Open Source ('GEOS')."

Bivand, Roger, Giovanni Millo, and Gianfranco Piras. 2021. "A Review of Software for Spatial Econometrics in R." *Mathematics* 9 (11): 1276. https://doi.org/10.3390/math9111276.

Bivand, Roger, and Gianfranco Piras. 2015. "Comparing Implementations of Estimation Methods for Spatial Econometrics." *Journal of Statistical Software* 63 (18): 1–36. https://doi.org/10.18637/jss.v063.i18.

Bivand, Roger, and David W. S. Wong. 2018. "Comparing Implementations of Global and Local Indicators of Spatial Association." *TEST* 27 (3): 716–48. https://doi.org/10.1007/s11749-018-0599-x.

Boillat, Sébastien, M. Graziano Ceddia, and Patrick Bottazzi. 2022. "The Role of Protected Areas and Land Tenure Regimes on Forest Loss in Bolivia: Accounting for Spatial Spillovers." *Global Environmental Change* 76 (September): 102571. https://doi.org/10.1016/j.gloenvcha.2022.102571.

Burridge, Peter, J. Paul Elhorst, and Katarina Zigova. 2016. "Group Interaction in Research and the Use of General Nesting Spatial Models." In *Spatial Econometrics: Qualitative and Limited Dependent Variables*, edited by Badi H. Baltagi, James P. LeSage, and R.

Kelley Pace, 37:223–58. Advances in Econometrics. Emerald Group Publishing Limited. https://doi.org/10.1108/S0731-905320160000037016.

Cliff, Andrew, and Keith Ord. 1972. "Testing for Spatial Autocorrelation Among Regression Residuals." *Geographical Analysis* 4 (3): 267–84. https://doi.org/10.1111/j.1538-4632.1972.tb00475.x.

Cook, Scott J., Jude C. Hays, and Robert J. Franzese. 2020. "Model Specification and Spatial Interdependence." In *The Sage Handbook of Research Methods in Political Science and International Relations*, edited by Luigi Curini and Robert Franzese, 1st ed, 730–47. Thousand Oaks: SAGE Inc.

Drukker, David M., Peter Egger, and Ingmar R. Prucha. 2013. "On Two-Step Estimation of a Spatial Autoregressive Model with Autoregressive Disturbances and Endogenous Regressors." *Econometric Reviews* 32 (5-6): 686–733. https://doi.org/10.1080/07474938.2013.741020.

Elhorst, J. Paul. 2012. "Dynamic Spatial Panels: Models, Methods, and Inferences." *Journal of Geographical Systems* 14 (1): 5–28. https://doi.org/10.1007/s10109-011-0158-4.

———. 2014. *Spatial Econometrics: From Cross-Sectional Data to Spatial Panels*. SpringerBriefs in Regional Science. Berlin and Heidelberg: Springer. https://doi.org/10.1007/978-3-642-40340-8.

Elhorst, J. Paul, and S. Halleck Vega. 2017. "The SLX Model: Extensions and the Sensitivity of Spatial Spillovers to W." *Papeles de Economía Española* 152: 34–50.

Fischer, Manfred M., Monika Bartkowska, Aleksandra Riedl, Sascha Sardadvar, and Andrea Kunnert. 2009. "The Impact of Human Capital on Regional Labor Productivity in Europe." *Letters in Spatial and Resource Sciences* 2 (2-3): 97–108. https://doi.org/10.1007/s12076-009-0027-7.

Florax, Raymond, Hendrik Folmer, and Sergio J. Rey. 2003. "Specification Searches in Spatial Econometrics: The Relevance of Hendry's Methodology." *Regional Science and Urban Economics* 33 (5): 557–79. https://doi.org/10.1016/S0166-0462(03)00002-4.

Franzese, Robert J., and Jude C. Hays. 2007. "Spatial Econometric Models of Cross-Sectional Interdependence in Political Science Panel and Time-Series-Cross-Section Data." *Political Analysis* 15 (2): 140–64. https://doi.org/10.1093/pan/mpm005.

Gibbons, Steve, and Henry G. Overman. 2012. "Mostly Pointless Spatial Econometrics?" *Journal of Regional Science* 52 (2): 172–91. https://doi.org/10.1111/j.1467-9787.2012.00760.x.

Gibbons, Steve, Henry G. Overman, and Eleonora Patacchini. 2015. "Spatial Methods." In *Handbook of Regional and Urban Economics*, edited by Gilles Duranton, J. Vernon Henderson, and William C. Strange, 5:115–68. Amsterdam: Elsevier. https://doi.org/10.1016/B978-0-444-59517-1.00003-9.

Gräler, Benedikt, Edzer Pebesma, and Gerard Heuvelink. 2016. "Spatio-Temporal Interpolation Using Gstat." *The R Journal* 8 (1): 204–18.

Halleck Vega, Solmaria, and J. Paul Elhorst. 2015. "The SLX Model." *Journal of Regional Science* 55 (3): 339–63. https://doi.org/10.1111/jors.12188.

Kelejian, Harry H., and Gianfranco Piras. 2017. *Spatial Econometrics*. Elsevier. https://doi.org/10.1016/C2016-0-04332-2.

Kelejian, Harry H., and Ingmar R. Prucha. 1998. "A Generalized Spatial Two-Stage Least

Squares Procedure for Estimating a Spatial Autoregressive Model with Autoregressive Disturbances." *The Journal of Real Estate Finance and Economics* 17 (1): 99–121. https://doi.org/10.1023/A:1007707430416.

———. 2010. "Specification and Estimation of Spatial Autoregressive Models with Autoregressive and Heteroskedastic Disturbances." *Journal of Econometrics* 157 (1): 53–67. https://doi.org/10.1016/j.jeconom.2009.10.025.

Kelejian, Harry H., Ingmar R. Prucha, and Yevgeny Yuzefovich. 2004. "Instrumental Variable Estimation of a Spatial Autoregressive Model with Autoregressive Disturbances: Large and Small Sample Results." In *Spatial and Spatiotemporal Econometrics*, edited by James P. LeSage and R. Kelley Pace, 163–98. Advances in Econometrics. Amsterdam and Boston: Elsevier.

Lee, Barrett A., Sean F. Reardon, Glenn Firebaugh, Chad R. Farrell, Stephen A. Matthews, and David O'Sullivan. 2008. "Beyond the Census Tract: Patterns and Determinants of Racial Segregation at Multiple Geographic Scales." *American Sociological Review* 73 (5): 766–91. https://doi.org/10.1177/000312240807300504.

Lee, Lung-fei. 2004. "Asymptotic Distributions of Quasi-Maximum Likelihood Estimators for Spatial Autoregressive Models." *Econometrica* 72 (6): 1899–1925.

LeSage, James P. 2014. "What Regional Scientists Need to Know about Spatial Econometrics." *The Review of Regional Studies* 44 (1): 13–32. https://doi.org/https://dx.doi.org/10.2139/ssrn.2420725.

LeSage, James P., and R. Kelley Pace. 2009. *Introduction to Spatial Econometrics*. Statistics, Textbooks and Monographs. Boca Raton: CRC Press.

———. 2014. "The Biggest Myth in Spatial Econometrics." *Econometrics* 2 (4): 217–49. https://doi.org/10.3390/econometrics2040217.

Lovelace, Robin, Jakub Nowosad, and Jannes Muenchow. 2019. *Geocomputation with R*. 1st ed. Chapman & Hall/CRC the R Series. Boca Raton: Chapman & Hall/CRC.

Manski, Charles F. 1993. "Identification of Endogenous Social Effects: The Reflection Problem." *The Review of Economic Studies* 60 (3): 531–42. https://doi.org/10.2307/2298123.

Mohai, Paul, and Robin Saha. 2007. "Racial Inequality in the Distribution of Hazardous Waste: A National-Level Reassessment." *Social Problems* 54 (3): 343–70. https://doi.org/10.1525/sp.2007.54.3.343.

Moran, P. A. P. 1950. "Notes on Continuous Stochastic Phenomena." *Biometrika* 37 (1/2): 17. https://doi.org/10.2307/2332142.

Mur, Jesús, and Ana Angulo. 2009. "Model Selection Strategies in a Spatial Setting: Some Additional Results." *Regional Science and Urban Economics* 39 (2): 200–213. https://doi.org/10.1016/j.regsciurbeco.2008.05.018.

Neumayer, Eric, and Thomas Plümper. 2016. "W." *Political Science Research and Methods* 4 (01): 175–93. https://doi.org/10.1017/psrm.2014.40.

Ord, John Keith. 1975. "Estimation Methods for Models of Spatial Interaction." *Journal of the American Statistical Association* 70 (349): 120–26. https://doi.org/10.2307/2285387.

Pace, R. Kelley, and James P. LeSage. 2010. "Omitted Variable Biases of OLS and Spatial Lag Models." In *Progress in Spatial Analysis*, edited by Antonio Páez, Julie Gallo, Ron N. Buliung, and Sandy Dall'erba, 17–28. Berlin and Heidelberg: Springer.

Pebesma, Edzer. 2018. “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal* 10 (1): 439. https://doi.org/10.32614/RJ-2018-009.

Pebesma, Edzer, and Roger Bivand. 2023. *Spatial Data Science: With Applications in R*. First. Boca Raton: Chapman and Hall/CRC. https://doi.org/10.1201/9780429459016.

Pinkse, Joris, and Margaret E. Slade. 2010. “The Future of Spatial Econometrics.” *Journal of Regional Science* 50 (1): 103–17. https://doi.org/10.1111/j.1467-9787.2009.00645.x.

Rüttenauer, Tobias. 2018. “Neighbours Matter: A Nation-wide Small-area Assessment of Environmental Inequality in Germany.” *Social Science Research* 70: 198–211. https://doi.org/10.1016/j.ssresearch.2017.11.009.

———. 2022. “Spatial Regression Models: A Systematic Comparison of Different Model Specifications Using Monte Carlo Experiments.” *Sociological Methods & Research* 51 (2): 728–59. https://doi.org/10.1177/0049124119882467.

———. 2024. “Spatial Data Analysis.” arXiv. https://arxiv.org/abs/2402.09895.

Sarrias, Mauricio. 2023. *Intermediate Spatial Econometrics with Applications in R*.

Tennekes, Martijn. 2018. “Tmap : Thematic Maps in R.” *Journal of Statistical Software* 84 (6). https://doi.org/10.18637/jss.v084.i06.

Tobler, Waldo R. 1970. “A Computer Movie Simulating Urban Growth in the Detroit Region.” *Economic Geography* 46: 234–40. https://doi.org/10.2307/143141.

Ward, Michael Don, and Kristian Skrede Gleditsch. 2008. *Spatial Regression Models*. Vol. 155. Quantitative Applications in the Social Sciences. Thousand Oaks: Sage.

Wimpy, Cameron, Guy D. Whitten, and Laron K. Williams. 2021. “X Marks the Spot: Unlocking the Treasure of Spatial-X Models.” *The Journal of Politics* 83 (2): 722–39. https://doi.org/10.1086/710089.

Wong, David. 2009. “The Modifiable Areal Unit Problem (MAUP).” In *The Sage Handbook of Spatial Analysis*, edited by A. Stewart Fotheringham and Peter Rogerson, 105–24. Los Angeles and London: Sage.

Wooldridge, Jeffrey M. 2010. *Econometric Analysis of Cross Section and Panel Data*. Cambridge, Mass.: MIT Press.