Creating synthetic instrumental variables from spatial filters

Bruno Santos & Antonio Paez

2025-07-03

This Rmarkdown file is part of the **CommuteCA** package. This package was created in conjunction with the office of the *Research Data Center* at *McMaster University*, the *Sherman Centre for Digital Scholarship* and the *Mobilizing Justice*[1].

The **CommuteCA** R package was created to develop standardized methods for transport analysis in research, particularly for analysis using the *2021 Census of Population* from Statistics Canada. We focused our efforts on the *Commuting Reference Guide*, which provides valuable variables and information on commuting for the Canadian population aged 15 and older living in private households.

Section objectives

After identifying disadvantaged populations in terms of job accessibility, we will now advance our analysis by developing regression models to explore the relationship between job accessibility and individual outcomes. These models examine the connection between a dependent variable (an outcome) and predictor variables (job accessibility indicators and other socioeconomic information). They estimate the probability of the dependent variable, conditional on a linear combination of the predictor variables.

In social science, it is well understood that "correlation does not imply causality" [Cunningham, 2021]. Therefore, it is important to evaluate the actual impact of a predictor variable (the cause) on the outcome (the effect). This requires studying the process that determines the cause-and-effect relationship between variables, a process known as *causal inference.*

Including causal inference in our modeling approach means we must account for the possible (and indeed likely) presence of *endogeneity.* Endogeneity occurs when the effect of an independent variable on a dependent variable cannot be interpreted causally, often due to omitted variables that lead to biased (i.e., inconsistent) estimates [Antonakis et al., 2010]. There are various ways to address endogeneity. In our case, we propose using an instrumental variable.

In this section, we will create synthetic instrumental variables obtained after applying a spatial filtering to the spatial availability variables created in the previous sections, in a methodology created by Gallo and Páez [2013].

[1] For this demonstration, we will use the case of the city of Toronto. If you want to use the other options, the original data from the Census in an RDC office or the test data for all locations in Canada, *update* the address in the chunk.

Suggested Readings

- Le Gallo, J., & Páez, A. (2013). Using synthetic variables in instrumental variable estimation of spatial series models. *Environment and Planning A*, *45*(9), 2227-2242.
- Cunningham, S. (2021). *Causal inference: The mixtape.* Yale university press.

Instrumental variables

A fundamental assumption in regression analysis is the absence of correlation between the explanatory variables and the error terms [Le Gallo and Páez, 2013]. Endogeneity occurs when an independent variable correlates with the regression error term. To predict outcomes from the 2021 Census of Population, we propose to address endogeneity by applying an instrumental variable technique. When an endogenous variable is included in a regression model, the consistency of the estimators is compromised, affecting the results and hindering the ability to establish a cause-effect relationship between the explanatory and dependent variables.

An instrumental variable Z must satisfy the following criteria ([Cunningham, 2021, Luz and Portugal, 2022]):

1. Z must be highly correlated with and have a causal effect on the endogenous variable D, or share a common cause with D.

2. Z must affect the dependent variable Y only through D, with no direct effect of Z on Y.

3. Z must not be correlated with the regression residuals of D on Y.

A good instrument for the proposed models must be highly correlated with and have a causal effect (or share a common cause) on the accessibility variable (the endogenous variable); affect employment outcomes only through accessibility; and not share common causes with employment outcomes. In simpler terms, the instrument must be correlated with employment outcomes only through accessibility.

Examples of instrumental variables used in studies analyzing the relationship between employment outcomes and job accessibility include Euclidean distance from the spatial unit centroid to the nearest major road or employment subcenter ([Delmelle et al., 2021, Jin and Paulsen, 2018]), share of no-vehicle households ([Hu, 2017]), transit accessibility and car unavailability ([Johnson et al., 2017]), distance to rivers ([Duarte et al., 2023]), and population density ([Bastiaanssen et al., 2021]).

However, two important challenges arise in the search for relevant instruments:

1. The likelihood that the instruments and error terms are correlated increases as the correlation between the endogenous variable and the instruments becomes stronger, which can render the instruments invalid.

2. Instruments that are only weakly correlated with the endogenous variable may be ineffective and perform poorly.

In response to these challenges, Le Gallo and Páez ([2013]) developed an alternative, cutting-edge approach that relies on synthetic instrumental variables when the model involves spatial data. The researchers demonstrated that synthetic variables can be generated using spatial filtering, a technique that removes spatial residual autocorrelation to improve regression analysis. They propose the use of an eigenvector-based technique, in which transforms a projection of the contiguity matrix into latent map patterns through eigenvector structural analysis. These latent patterns, included in the spatial structure of the system, are then used to generate synthetic variables, which serve as instruments in IV estimation. Since they are derived from the spatial structure of the system, these synthetic instruments are exogenous, thereby fulfilling the core requirement of IV estimation.

We have chosen to use this technique because, given our aim to create a methodology that can be applied to any Canadian city, there are no universally applicable candidates for instrumentation that could be obtained in every context.

Spatial Filtering

According to Le Gallo and Páez [2013], spatial filtering is a method to deal with, or filter from the residuals, spatial autocorrelation present during regression analysis. Spatial autocorrelation is the degree of similarity among neighboring observation. Usually, the definition of the neighboring (ie, spatial structure) is made by the use of spatial weight matrix W, with individual values represented by:

$$w_{ij} = \begin{cases} w_{ij} > 0, & \text{if i and j are neighbors,} \\ w_{ij} = 0, & \text{otherwise.} \end{cases}$$

Neighborhood can be defined on the basis of contiguity, distance, or length of shared edge, among other criteria. So, having a weighted matrix that represents the spatial structure, the steps to generate a filter can be defined as follows:

1. Build a spatial weights matrix to capture the distances between all pairs of spatial units (in our case, Dissemination Areas, or DAs) under analysis.

2. Select a distance threshold to identify neighbors for each DA. This process usually results in a binary spatial contiguity matrix. There is no clear theoretical guidance on the optimal distance threshold for constructing the binary contiguity matrix.
3. Compute all eigenvectors associated with the contiguity matrix. These eigenvectors are orthogonal to one another and each represents a portion of the variance in the spatial contiguity matrix. Each independent variable will have its own set of associated eigenvectors.
4. Construct the spatial filter as a linear combination of a subset of these eigenvectors. The spatial filter for each independent variable is typically obtained by regressing the independent variable on the subset of eigenvectors with p-values below a specified threshold (for instance, p 0.05), and then using the predicted values of the independent variable as the synthetic instrument.

The pseudo-code to build the spatial filter is as follows:

- Initialize an index value $i = 1$, an empty vector for the spatial filter $S = 0$, and a constant vector $X = 1$.
- In a for-loop and for every $i <= n$, with n being number of eigenvectors, select the eigenvector E_i obtained from the weighted matrix as a candidate for inclusion in the spatial filter, and estimate the following model using Ordinary Least Squares (OLS), where θ and β are vectors of coefficients, and ϵ is a vector of error terms.

$$Y = \theta X + \beta E_i + \epsilon$$

- If coefficient i is significant at a pre-determined level (eg, $p - value <= 0.05$), then synthesize the eigenvector and the existing filter: $S = S + \beta E_i$.
- The spatial filter S is the synthetic instrumental variable.

In summary, a spatial filter will be created to generate a synthetic counterpart that accurately reproduces a spatial random variable. To explore the potential of using eigenvectors to build synthetic instrumental variables, we will not only construct a spatial filter based on the contiguity matrix but also create additional filters based on commuting duration. Specifically, we will use a travel time matrix - since commute duration also depends on the spatial structure and represents an explicit space-time relationship. Additionally, we will develop a spatial filter based on commuting time impedance values that were derived after applying impedance functions (calculated from spatial accessibility) to the travel time matrix.

Let's code!

Load the packages:

```r
library(CommuteCA)
library(dplyr)# A Grammar of Data Manipulation
library(fitdistrplus) # Help to Fit of a Parametric Distribution to Non-Censored or Censored Data
library(scales) # Scale data column-wise in a computationally efficient way
library(here) # enable easy file referencing in project-oriented workflows
library(ggplot2) # Create Elegant Data Visualizations Using the Grammar of Graphics
library(RColorBrewer) # color schemes for maps (and other graphics) designed
library(sf) # support for simple features, a standardized way to encode spatial vector data
library(tmap) # thematic maps
library(tidyr) # tidying data
library(matlib) # To get the eigenvectors
library(Hmisc) # To considerate weights when calculate the median
library(sf) # support for simple features, a standardized way to encode spatial vector data
library(spdep) # Construct neighbours list from polygon list
```

Data

The dataset used in this demonstration is test data produced to replicate the variables available in the original Census of Population for the City of Toronto. The test data contains 52,650 rows and 31 columns. As in the original census data, each row refers to a respondent and each column refers to a variable[2]. The creation of test data was necessary because the surveys provided by Statistics Canada are confidential and cannot be accessed outside of a Research Data Center.

If you want to work with the original Census dataset, the process for obtaining the spatial filters will be the same as for the test data, except that you will have to update the address of the file in the chunk called *load-census-data*.

For this R markdown, we'll use the following variables[3]:

Table 1: Census variables used in this section.

Variable	**Description**
PRCDDA	Refers to the dissemination area (DA) of current residence.
PCD	Census division of current residence.
CompW1	Weight for the households and dwellings universes.
PWDA	Place of work dissemination area.
PWCD	Place of work census division.

[2] You can check out more information about the Census on the Dictionary website.

[3] The explanation of each variable can be found in the *2021 Census of Population's website*.

PWDUR	Commuting duration, it refers to the length of time, in minutes, usually required by a person to travel to their place of work.
PwMode	Main mode of commuting' refers to the main mode of transportation a person uses to travel to their place of work.

We will use the job spatial availability measures. The job spatial availability file has the following variables:

Table 2: Job spatial availability variables.

Variable	**Description**
PRCDDA	Refers to the DA defined as the origin.
PwMode	Transportation mode used to calculate the travel time.
SA_im	Job spatial availability for the DA.

To build the distance matrix, we will use the dissemination areas spatial file, previously downloaded using the `cancensus` package. Finally, we will explore the production of other spatial filters using the travel time matrix with the impedance values, created and exported after modelling job accessibilities. We are interested in the following variables of the travel time matrix:

Table 3: Travel time table variables used in this section.

Variable	**Description**
from_id	Refers to the DA defined as the origin.
to_id	Refers to the DA defined as the destination.
travel_time	Estimated travel time in minutes from origin to destination.
PwMode	Transportation mode used to calculate the travel time.
f	The impedance value obtained for the travel time in `travel_time` column.

Read and preprocess the files

- Census

Reading census data and creating a R data frame[4]:

[4] For this demonstration, we will use the case of the city of Toronto. If you want to use the other options, the original data from the Census in an RDC office or the test data for all locations in Canada, *update* the address in the chunk.

```
data("census_test_toronto")
census <- census_test_toronto
```

If the original Census dataset available in the RDC is not in .csv (comma-separated values) format but is instead provided in other formats such as SPSS, SAS, or SAS Data, you can use the foreign package (a built-in R library) to import it:

```
# library(foreign)
# foreign::read.dta(files_address) # For Stata
# foreign::read.spss(files_address) # For SPSS
# There are many other options! You can search for this library in the 'Packages' window and explore ad
```

NOTE: If the code above did not run correctly, you probably are experiencing a file address error. Try to identify the correct address and update the chunk named `census-file-address` to continue.

We will filter the data frame by census division[5]. The chunk below shows how to make this procedure:

```
code <- 3520 # census division of Toronto

# Filtering by census division
 census_filtered <- census %>%
                   filter(PCD == code) # Only select respondents who live in  Toronto

census <- census %>%
                   filter(PCD == code) # Only select respondents who live in  Toronto
```

Select only the variables that we will use to build the spatial filters:

```
census_filtered <- census_filtered %>%
          dplyr::select("PRCDDA",
                        "PCD",
                        "CompW1",
                        "PwMode",
                        "PWDUR",
                        "PWDA",
                        "PWCD")
```

According to the census code book, the variable ‘PwMode’ has the following possible values:

- -3: Not applicable.
- 1: Car, truck or van - as a driver.
- 2: Car, truck or van - as a passenger.
- 3: Bus.
- 4: Subway or elevated rail.

[5] As said before, we will perform our analysis for the city of Toronto. If you want to select a specific area to work, uncomment the code above that applies to your case and select the apropriate code of your interest unit. Please, check the dictionary to have more informations about the provinces code, census divisions, and census metropolitan areas

- 5: Light rail, streetcar or commuter train.
- 6: Passenger ferry.
- 7: Walked.
- 8: Bicycle.
- 9: Motorcycle, scooter or moped.
- 10: Other method.

We'll rename the travel modes to facilitate the readability of the data. Additionally, we'll remove from our analysis travel modes signed as 'Other methods':

```
census_filtered <- census_filtered  %>%
                  filter(PwMode < 10) %>%
                  mutate(PwMode = case_when(PwMode > 0 & PwMode <= 2 ~ "Car/motor",
                                            PwMode == 9 ~ "Car/motor",
                           PwMode >= 3 & PwMode <= 6  ~ "Transit",
                           PwMode == 7  ~ "Walk",
                           PwMode == 8  ~ "Bike"),

         PwMode = factor(PwMode, levels = c("Bike", "Walk", "Car/motor", "Transit"))) %>%
  filter(PwMode %in% c("Bike", "Walk", "Car/motor", "Transit"))

census <- census  %>%
                  mutate(PwMode = case_when(PwMode > 0 & PwMode <= 2 ~ "Car/motor",
                                            PwMode == 9 ~ "Car/motor",
                           PwMode >= 3 & PwMode <= 6  ~ "Transit",
                           PwMode == 7  ~ "Walk",
                           PwMode == 8  ~ "Bike"),

         PwMode = factor(PwMode, levels = c("Bike", "Walk", "Car/motor", "Transit")))
```

Creating a Data Frame with workers who live and work in the city under study:

```
census_workers <- census_filtered %>%
  filter(PCD == code  & PWCD == code)
```

Create a new Data Frame with all possible combinations of DA code and transportation modes:

```
census_expanded <- expand.grid(PRCDDA = unique(census_filtered$PRCDDA),
                               PwMode = unique(census_filtered$PwMode))
```

- Accessibility measures

Setting the folder with the accessibility measures:

```
# Folder with the accessibility files
measures_folder <- paste0(here::here(),"/data-raw/output/PCD3520/DA/accessibility-measures/")

# Read file job spatial availability by mode
SA <- read.csv(paste0(measures_folder, "SA_mode_original_RDC.csv"))
```

Preparing the accessibility file:

```
SA_mode <- census_expanded %>%
  left_join(SA, by = c("PRCDDA","PwMode")) %>%
  mutate(SA_im = replace_na(SA_im, 0))
```

- Travel time matrix

Reading the travel time table with the impedance measures:

```
# Folder
ttm_folder <- paste0(here(),"/data-raw/output/PCD3520/DA/travel_times/") # Update it if necessary

# Read file
ttm_f <- read.csv(paste0(ttm_folder, "ttm_f.csv"))
```

- Spatial files

Reading the dissemination areas spatial file:

```
# Folder
directory_spatial_files <- paste0(here(),"/data-raw/output/PCD3520/spatial-files/") # Update it if nece

# Read file
da_file <- st_read(paste0(directory_spatial_files, "dissemination_areas.shp"))
```

```
## Reading layer `dissemination_areas' from data source
##   `C:\Bruno\CommuteCA\data-raw\output\PCD3520\spatial-files\dissemination_areas.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 3743 features and 6 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -79.63931 ymin: 43.58095 xmax: -79.11542 ymax: 43.85547
## Geodetic CRS:  WGS 84
```

Reading the census_divisions geometries:

```
census_divisions <-   st_read(paste0(directory_spatial_files, "census_divisions.shp"))
```

```
## Reading layer `census_divisions' from data source
##   `C:\Bruno\CommuteCA\data-raw\output\PCD3520\spatial-files\census_divisions.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 1 feature and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -79.63931 ymin: 43.58095 xmax: -79.11543 ymax: 43.85547
## Geodetic CRS:  WGS 84
```

Creating folder to save files

Creating a directory to export the figures and tables:

```
diretorio_export_figures <- paste0(here(),"/data-raw/output/PCD3520/DA/spatial-filter-figures/") # Upda

if(!dir.exists(diretorio_export_figures)){
  dir.create(diretorio_export_figures, recursive = TRUE)}

diretorio_export_tables <- paste0(here(),"/data-raw/output/PCD3520/DA/spatial-filter-measures/") # Upda

if(!dir.exists(diretorio_export_tables)){
  dir.create(diretorio_export_tables, recursive = TRUE)}
```

Spatial filters

Spatial filter based on the contiguity

The first spatial filter we will create is based on contiguity. This means that the weighted matrix will only have values of 0 or 1, where 0 indicates that the DAs are not neighbors and 1 indicates that they are neighbors.

First, we will create the contiguity matrix:

```
# Contiguity matrix
mtx_distance_binary <- spdep::nb2mat(spdep::poly2nb(da_file), style = "B", zero.policy = TRUE)

# Changing the diagonal column to 1
diag(mtx_distance_binary) <- 1

# Renaming the columns and row index so we can identify each DA
rownames(mtx_distance_binary) <- da_file$DAUID
colnames(mtx_distance_binary) <- da_file$DAUID
```

Calculating the contiguity matrix eigenvectors:

```
ev_dist <- eigen(mtx_distance_binary)

# Creating an Eigenvectors Data Frame
eigv_df <- as.data.frame(Re(ev_dist$vectors))
```

```
colnames(eigv_df) <- paste0("EV", seq_len(ncol(eigv_df)))

# Adding the DA code to the Eigenvectors table
weight_matrix_eigv_dist <- cbind(eigv_df, PRCDDA = as.numeric(rownames(mtx_distance_binary)))
```

```
modes <- unique(census_workers$PwMode)

mode_results_dist <- list()

for (mode in modes) {
  cat("Creating the spatial filter (distance) for the transportation mode:", mode, "\n")

  # Creating a df with the accessibility values
  y <- SA_mode %>%
    filter(PwMode == mode) %>%
    left_join(weight_matrix_eigv_dist, by = c("PRCDDA" = "PRCDDA")) %>%
    mutate(x = 1, Sf_dist = 0)

  # Creating the spatial filter (Sf_dist)
  for (ev_name in names(y)[startsWith(names(y), "EV")]) {
    formula <- as.formula(paste("SA_im ~ x + Sf_dist +", ev_name))
    model <- lm(formula, data = y)
    coefs <- summary(model)$coefficients

    if (ev_name %in% rownames(coefs)) {
      p_value <- coefs[ev_name, "Pr(>|t|)"]
      B_value <- coefs[ev_name, "Estimate"]

      if (!is.na(p_value) && p_value < 0.05) {
        y$Sf_dist <- y$Sf_dist + B_value * y[[ev_name]]
      }
    }
  }

 mode_results_dist[[mode]] <- y
}
```

```
## Creating the spatial filter (distance) for the transportation mode: Transit
## Creating the spatial filter (distance) for the transportation mode: Car/motor
## Creating the spatial filter (distance) for the transportation mode: Walk
## Creating the spatial filter (distance) for the transportation mode: Bike
```

Creating a data frame with all spatial filters and accessibility:

```
spatial_filter_dist <- as.data.frame(bind_rows(mode_results_dist, .id = "PwMode")) %>%
  dplyr::select(PRCDDA, PwMode, SA_im, Sf_dist)
```

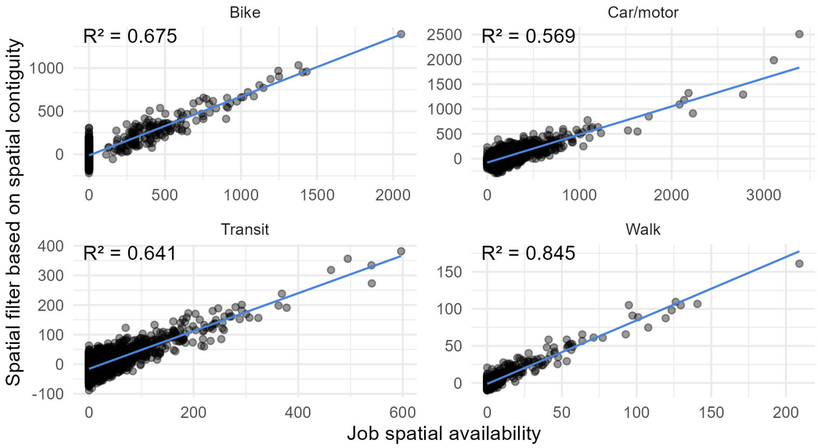
Pearson correlation between job spatial availability and spatial filter by transportation mode:

```
correlations <- spatial_filter_dist %>%
  group_by(PwMode) %>%
  summarise(correlation = cor(SA_im, Sf_dist, use = "complete.obs"))

correlations
```

```
## # A tibble: 4 x 2
##   PwMode    correlation
##   <chr>           <dbl>
## 1 Bike            0.822
## 2 Car/motor       0.754
## 3 Transit         0.800
## 4 Walk            0.919
```

Creating a scatter plot figure:

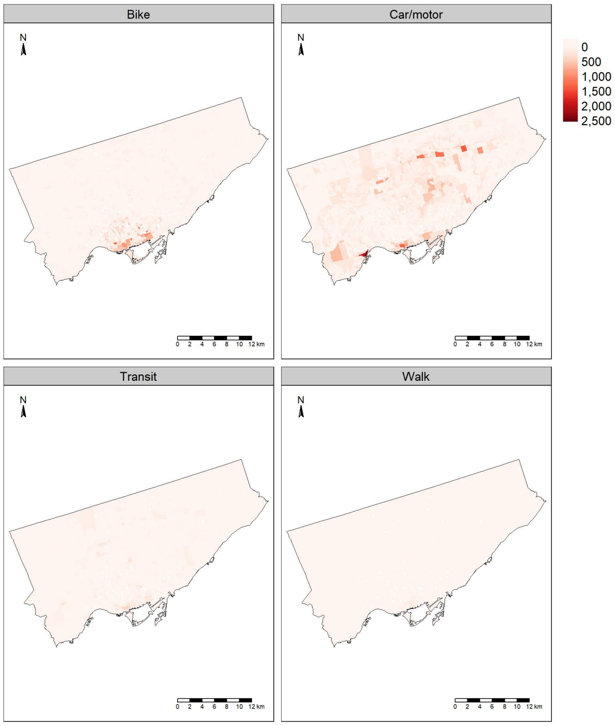
```
r2_labels <- spatial_filter_dist %>%
  group_by(PwMode) %>%
  summarise(r2 = summary(lm(Sf_dist ~ SA_im))$r.squared) %>%
  mutate(label = paste0("R² = ", round(r2, 3)))

spatial_filter_dist_fig <- ggplot(spatial_filter_dist, aes(x = SA_im, y = Sf_dist)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "lm", se = FALSE, color = "#4b84db", linewidth = 0.55) +
  geom_text(data = r2_labels,
            aes(x = -Inf, y = Inf, label = label),
            hjust = -0.1, vjust = 1.1, inherit.aes = FALSE) +
  facet_wrap(~ PwMode, scales = "free") +
  theme_minimal() +
  labs(x = "Job spatial availability", y = "Spatial filter based on spatial contiguity") +
theme_minimal()

# Saving figure
ggsave(file = paste0(diretorio_export_figures,"/spatial_filter_dist_scatterplot.jpg"),
       plot = spatial_filter_dist_fig,
       width = 16,
       height = 9,
       units = "cm",
       dpi = 300)

knitr::include_graphics(paste0(diretorio_export_figures,"/spatial_filter_dist_scatterplot.jpg"))
```



Visualizing the spatial filter in form of maps:

```
spatial_filter_contiguity_spatial <- census_expanded %>%
  filter(!is.na(PwMode)) %>%
  left_join(spatial_filter_dist, by = c("PRCDDA" = "PRCDDA",
                                        "PwMode" = "PwMode")) %>%
  left_join(da_file %>% mutate(DAUID = as.numeric(DAUID)), by = c("PRCDDA" = "DAUID"))

spatial_filter_contiguity_spatial <- st_as_sf(spatial_filter_contiguity_spatial)

spatial_filter_contiguity_fig <- tm_shape(spatial_filter_contiguity_spatial) +
   tm_polygons("Sf_dist",
              style = "cont",
              palette = "Reds",
              title = " ",
              border.col = NULL) +
   tm_facets(by = "PwMode") +
   tm_scale_bar(position = c("right", "bottom")) +
   tm_compass(position = c("left", "top"), size = 1.0) +
   tm_shape(census_divisions) +
   tm_borders("black", lwd=0.5)

tmap_save(spatial_filter_contiguity_fig,
          paste0(diretorio_export_figures,"contiguity_spatial_filter.jpg"),
          width = 20,
          height = 18,
          units = "cm",
          dpi = 300)

spatial_filter_contiguity_fig2 <- tm_shape(spatial_filter_contiguity_spatial) +
   tm_polygons("Sf_dist",
              style = "cont",
               palette = "Reds",
               title = " ",
               border.col = NULL) +
   tm_facets(by = "PwMode", free.scales = TRUE) +
  tm_scale_bar(position = c("right", "bottom")) +
   tm_compass(position = c("left", "top"), size = 1.0) +
   tm_shape(census_divisions) +
   tm_borders("black", lwd=0.5)

tmap_save(spatial_filter_contiguity_fig2,
          paste0(diretorio_export_figures,"contiguity_spatial_filter_2.jpg"),
          width = 20,
          height = 18,
          units = "cm",
          dpi = 300)
```

```
knitr::include_graphics(paste0(diretorio_export_figures,"/contiguity_spatial_filter.jpg"))
```



Spatial filter based on the commute duration

After creating the traditional spatial filter based on a contiguity matrix, we will now explore the potential of building a spatial filter based on a time matrix. First, we will create a binary matrix, where the values will be 0 if the commute duration is higher than the median duration, and 1 otherwise.

Next, we will explore the possibility of building a spatial filter using the impedance values themselves. To do so, we will normalize the impedance values so they range from 0 to 1.

Median commute time

Obtaining the median commute duration for each transportation mode:

```
dur_medians <- census_workers %>%
  filter(PwMode %in% c("Bike", "Walk", "Car/motor", "Transit")) %>%
  group_by(PwMode) %>%
  dplyr::summarise("Median" =  Hmisc::wtd.quantile(PWDUR, weights = CompW1, probs = 0.5),
            .groups = "drop")

dur_medians
```

```
## # A tibble: 4 x 2
##   PwMode    Median
##   <fct>      <dbl>
## 1 Bike          16
## 2 Walk          34
## 3 Car/motor     41
## 4 Transit       51
```

Creating the spatial filter for based on the median commute duration:

```
mode_results <- list()

for (mode in modes) {
  cat("Creating the spatial filter (median duration based) for the transportation mode:", mode, "\n")

  # Mode weight matrix
  weight_matrix <- ttm_f %>%
    filter(PwMode == mode) %>%
    mutate(threshold =
             case_when(travel_time <= as.numeric(dur_medians$Median[dur_medians$PwMode == mode]) ~ 1,
                       TRUE ~ 0)) %>%
    dplyr::select(from_id, to_id, threshold) %>%
    pivot_wider(names_from = to_id, values_from = threshold, values_fill = 0) %>%
    as.data.frame()

  # Compute eigenvectors
  mat <- as.matrix(weight_matrix[, -1])
  ev <- eigen(mat)
  eigv_df <- as.data.frame(Re(ev$vectors))
  colnames(eigv_df) <- paste0("EV", seq_len(ncol(eigv_df)))

  weight_matrix_eigv <- cbind(from_id = weight_matrix$from_id, eigv_df)

  # Creating a df with the accessibility values
  y <- SA_mode %>%
    filter(PwMode == mode) %>%
    left_join(weight_matrix_eigv, by = c("PRCDDA" = "from_id")) %>%
    mutate(x = 1, Sf = 0)

  # Creating the spatial filter (Sf)
  for (ev_name in names(y)[startsWith(names(y), "EV")]) {
    formula <- as.formula(paste("SA_im ~ x + Sf +", ev_name))
    model <- lm(formula, data = y)
    coefs <- summary(model)$coefficients

    if (ev_name %in% rownames(coefs)) {
      p_value <- coefs[ev_name, "Pr(>|t|)"]
      B_value <- coefs[ev_name, "Estimate"]

      if (!is.na(p_value) && p_value < 0.05) {
        y$Sf <- y$Sf + B_value * y[[ev_name]]
      }
    }
  }

 mode_results[[mode]] <- y
}
```

```
## Creating the spatial filter (median duration based) for the transportation mode: Transit
## Creating the spatial filter (median duration based) for the transportation mode: Car/motor
## Creating the spatial filter (median duration based) for the transportation mode: Walk
## Creating the spatial filter (median duration based) for the transportation mode: Bike
```

Creating a data frame with all spatial filters and job accessibility:

```
spatial_filter_dur_medians <- as.data.frame(bind_rows(mode_results, .id = "PwMode")) %>%
  dplyr::select(PRCDDA, PwMode, SA_im, Sf) %>%
  rename(Sf_median_dur = Sf)
```

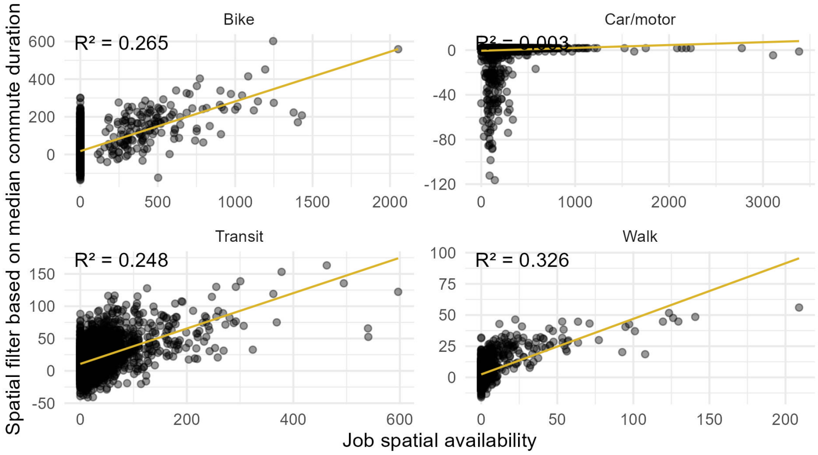
Pearson correlation between job spatial availability and spatial filter by transportation mode:

```
correlations <- spatial_filter_dur_medians %>%
  group_by(PwMode) %>%
  summarise(correlation = cor(SA_im, Sf_median_dur, use = "complete.obs"))

correlations
```

```
## # A tibble: 4 x 2
##   PwMode    correlation
##   <chr>           <dbl>
## 1 Bike           0.515
## 2 Car/motor      0.0505
## 3 Transit        0.498
## 4 Walk           0.571
```

Creating a scatter plot figure:

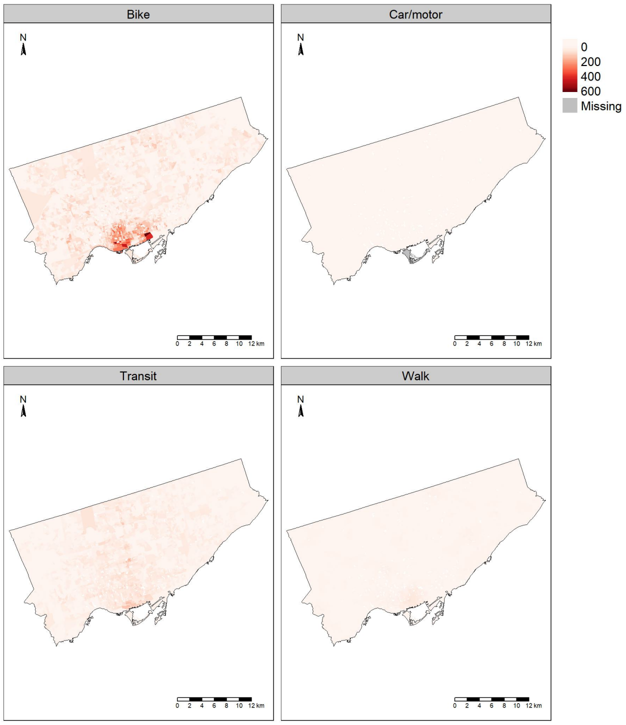
```
r2_labels <- spatial_filter_dur_medians %>%
  group_by(PwMode) %>%
  summarise(r2 = summary(lm(Sf_median_dur ~ SA_im))$r.squared) %>%
  mutate(label = paste0("R² = ", round(r2, 3)))

spatial_filter_dur_medians_fig <- ggplot(spatial_filter_dur_medians, aes(x = SA_im, y = Sf_median_dur))
  geom_point(alpha = 0.4) +
  geom_smooth(method = "lm", se = FALSE, color = "#dbb52c", linewidth = 0.55) +
  geom_text(data = r2_labels,
            aes(x = -Inf, y = Inf, label = label),
            hjust = -0.1, vjust = 1.1, inherit.aes = FALSE) +
  facet_wrap(~ PwMode, scales = "free") +
  theme_minimal() +
  labs(x = "Job spatial availability", y = "Spatial filter based on median commute duration") +
theme_minimal()

# Saving figure
ggsave(file = paste0(diretorio_export_figures,"/spatial_filter_dur_median_scatterplot.jpg"),
       plot = spatial_filter_dur_medians_fig,
       width = 16,
       height = 9,
       units = "cm",
       dpi = 300)

knitr::include_graphics(paste0(diretorio_export_figures,"/spatial_filter_dur_median_scatterplot.jpg"))
```



Visualizing the spatial filter in form of maps:

```
spatial_filter_dur_medians_spatial <- census_expanded %>%
  filter(!is.na(PwMode)) %>%
  left_join(spatial_filter_dur_medians, by = c("PRCDDA" = "PRCDDA",
                                               "PwMode" = "PwMode")) %>%
  left_join(da_file %>% mutate(DAUID = as.numeric(DAUID)), by = c("PRCDDA" = "DAUID"))

spatial_filter_dur_medians_spatial <- st_as_sf(spatial_filter_dur_medians_spatial)

spatial_filter_dur_medians_fig <- tm_shape(spatial_filter_dur_medians_spatial) +
   tm_polygons("Sf_median_dur",
              style = "cont",
              palette = "Reds",
              title = " ",
              border.col = NULL) +
   tm_facets(by = "PwMode") +
   tm_scale_bar(position = c("right", "bottom")) +
   tm_compass(position = c("left", "top"), size = 1.0) +
   tm_shape(census_divisions) +
   tm_borders("black", lwd=0.5)

tmap_save(spatial_filter_dur_medians_fig,
          paste0(diretorio_export_figures,"median_duration_spatial_filter.jpg"),
          width = 20,
          height = 18,
          units = "cm",
          dpi = 300)

spatial_filter_dur_medians_fig_2 <- tm_shape(spatial_filter_dur_medians_spatial) +
   tm_polygons("Sf_median_dur",
              style = "cont",
              palette = "Reds",
              title = " ",
              border.col = NULL) +
   tm_facets(by = "PwMode", free.scales = TRUE) +
  tm_scale_bar(position = c("right", "bottom")) +
   tm_compass(position = c("left", "top"), size = 1.0) +
   tm_shape(census_divisions) +
   tm_borders("black", lwd=0.5)

tmap_save(spatial_filter_dur_medians_fig_2,
          paste0(diretorio_export_figures,"median_duration_spatial_filter_2.jpg"),
          width = 20,
          height = 18,
          units = "cm",
          dpi = 300)
```

```
knitr::include_graphics(paste0(diretorio_export_figures,"/median_duration_spatial_filter.jpg"))
```



Commute duration impedance values

Creating a spatial filter based on the commute duration impedance values:

```
mode_results <- list()
```

```
for (mode in modes) {
  cat("Creating the spatial filter (duration impedance based) for the transportation mode:", mode, "\n"

  # Mode weight matrix
  weight_matrix <- ttm_f %>%
    filter(PwMode == mode) %>%
    dplyr::select(from_id, to_id, f) %>%
    pivot_wider(names_from = to_id, values_from = f, values_fill = 0) %>%
    as.data.frame()

  # Compute eigenvectors
  mat <- as.matrix(weight_matrix[, -1])
  mat_normalized <- (mat - min(mat)) / (max(mat) - min(mat))
  ev <- eigen(mat_normalized)
  eigv_df <- as.data.frame(Re(ev$vectors))
  colnames(eigv_df) <- paste0("EV", seq_len(ncol(eigv_df)))

  weight_matrix_eigv <- cbind(from_id = weight_matrix$from_id, eigv_df)

  # Creating a df with the accessibility values
  y <- SA_mode %>%
    filter(PwMode == mode) %>%
    left_join(weight_matrix_eigv, by = c("PRCDDA" = "from_id")) %>%
    mutate(x = 1, Sf_imp_dur = 0)

  # Creating the spatial filter (Sf)
  for (ev_name in names(y)[startsWith(names(y), "EV")]) {
    formula <- as.formula(paste("SA_im ~ x + Sf_imp_dur +", ev_name))
    model <- lm(formula, data = y)
    coefs <- summary(model)$coefficients

    if (ev_name %in% rownames(coefs)) {
      p_value <- coefs[ev_name, "Pr(>|t|)"]
      B_value <- coefs[ev_name, "Estimate"]

      if (!is.na(p_value) && p_value < 0.05) {
        y$Sf_imp_dur <- y$Sf_imp_dur + B_value * y[[ev_name]]
      }
    }
  }

 mode_results[[mode]] <- y
}
```

```
## Creating the spatial filter (duration impedance based) for the transportation mode: Transit
## Creating the spatial filter (duration impedance based) for the transportation mode: Car/motor
## Creating the spatial filter (duration impedance based) for the transportation mode: Walk
## Creating the spatial filter (duration impedance based) for the transportation mode: Bike
```

Creating a data frame with all spatial filters and accessibilities:

```
spatial_filter_imp_dur <- as.data.frame(bind_rows(mode_results, .id = "PwMode")) %>%
  dplyr::select(PRCDDA, PwMode, SA_im, Sf_imp_dur)
```

Pearson correlation between job spatial availability and spatial filter by transportation mode:

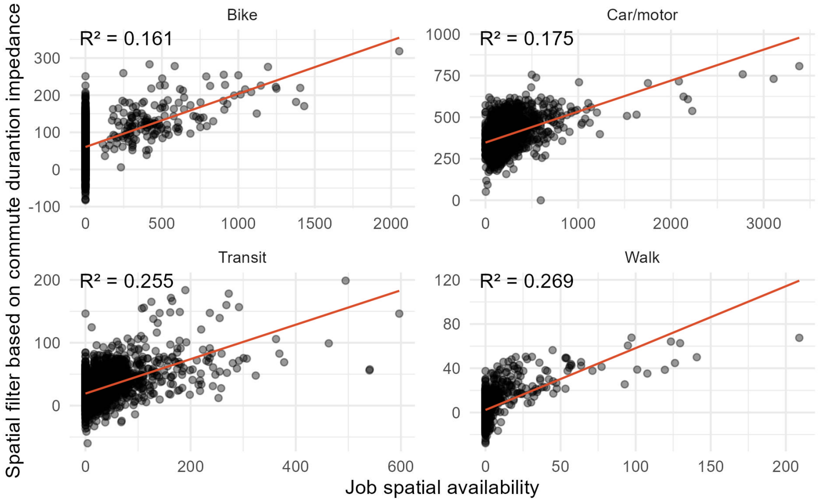
```
correlations <- spatial_filter_imp_dur %>%
  group_by(PwMode) %>%
  summarise(correlation = cor(SA_im, Sf_imp_dur, use = "complete.obs"))

correlations
```

```
## # A tibble: 4 x 2
##   PwMode    correlation
##   <chr>           <dbl>
## 1 Bike            0.401
## 2 Car/motor       0.419
## 3 Transit         0.505
## 4 Walk            0.518
```

Creating a scatter plot figure:

```
r2_labels <- spatial_filter_imp_dur %>%
  group_by(PwMode) %>%
  summarise(r2 = summary(lm(Sf_imp_dur ~ SA_im))$r.squared) %>%
  mutate(label = paste0("R² = ", round(r2, 3)))

spatial_filter_imp_dur_fig <- ggplot(spatial_filter_imp_dur, aes(x = SA_im, y = Sf_imp_dur)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "lm", se = FALSE, color = "#db4f2c", linewidth = 0.55) +
  geom_text(data = r2_labels,
            aes(x = -Inf, y = Inf, label = label),
            hjust = -0.1, vjust = 1.1, inherit.aes = FALSE) +
  facet_wrap(~ PwMode, scales = "free") +
  theme_minimal() +
  labs(x = "Job spatial availability", y = "Spatial filter based on commute durantion impedance") +
theme_minimal()

# Saving figure
```

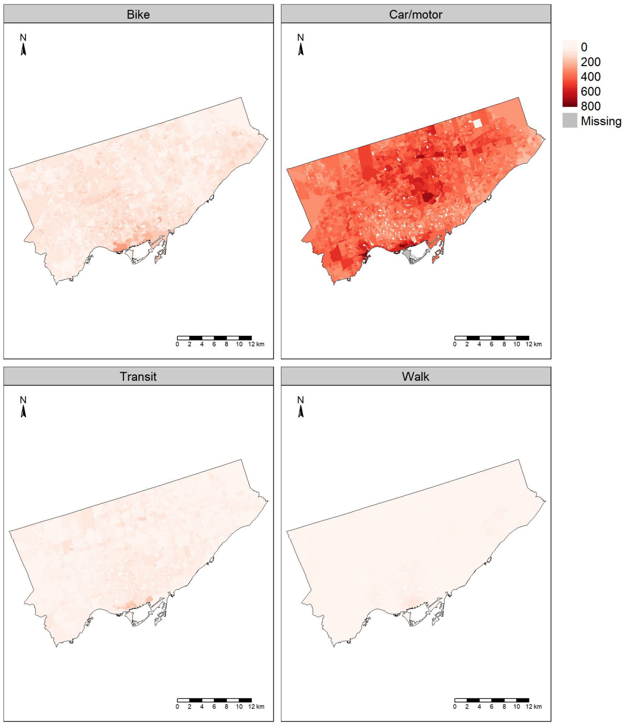
```
ggsave(file = paste0(diretorio_export_figures,"/spatial_filter_imp_dur_scatterplot.jpg"),
       plot = spatial_filter_imp_dur_fig,
       width = 16,
       height = 10,
       units = "cm",
       dpi = 300)

knitr::include_graphics(paste0(diretorio_export_figures,"/spatial_filter_imp_dur_scatterplot.jpg"))
```



Visualizing the spatial filter in form of maps:

```
spatial_filter_imp_dur_spatial <- census_expanded %>%
  filter(!is.na(PwMode)) %>%
  left_join(spatial_filter_imp_dur, by = c("PRCDDA" = "PRCDDA",
                                          "PwMode" = "PwMode")) %>%
  left_join(da_file %>% mutate(DAUID = as.numeric(DAUID)), by = c("PRCDDA" = "DAUID"))

spatial_filter_imp_dur_spatial <- st_as_sf(spatial_filter_imp_dur_spatial)

spatial_filter_imp_dur_fig <- tm_shape(spatial_filter_imp_dur_spatial) +
   tm_polygons("Sf_imp_dur",
              style = "cont",
              palette = "Reds",
              title = " ",
              border.col = NULL) +
   tm_facets(by = "PwMode") +
   tm_scale_bar(position = c("right", "bottom")) +
   tm_compass(position = c("left", "top"), size = 1.0) +
   tm_shape(census_divisions) +
   tm_borders("black", lwd=0.5)
```

```
tmap_save(spatial_filter_imp_dur_fig,
          paste0(diretorio_export_figures,"impedance_spatial_filter.jpg"),
          width = 20,
          height = 18,
          units = "cm",
          dpi = 300)

spatial_filter_imp_dur_fig_2 <- tm_shape(spatial_filter_imp_dur_spatial) +
   tm_polygons("Sf_imp_dur",
              style = "cont",
              palette = "Reds",
              title = " ",
              border.col = NULL) +
   tm_facets(by = "PwMode", free.scales = TRUE) +
  tm_scale_bar(position = c("right", "bottom")) +
   tm_compass(position = c("left", "top"), size = 1.0) +
   tm_shape(census_divisions) +
   tm_borders("black", lwd=0.5)

tmap_save(spatial_filter_imp_dur_fig_2,
          paste0(diretorio_export_figures,"impedance_spatial_filter_2.jpg"),
          width = 20,
          height = 18,
          units = "cm",
          dpi = 300)

knitr::include_graphics(paste0(diretorio_export_figures,"/impedance_spatial_filter.jpg"))
```



Data export

Creating a unique file with all spatial filters (contiguity, median commute duration, and impedance values):

```
spatial_filter <- spatial_filter_dist %>%
  left_join(spatial_filter_dur_medians[,c(1,2,4)], by = c("PRCDDA" = "PRCDDA", "PwMode" = "PwMode")) %>
  left_join(spatial_filter_imp_dur[,c(1,2,4)], by = c("PRCDDA" = "PRCDDA", "PwMode" = "PwMode"))
```

Exporting the spatial filters (original file):

```
write.csv(spatial_filter, paste0(diretorio_export_tables, "spatial_filter_original.csv"), row.names=FALS
```

Confidentiality vetting

If you are interested in manipulating the spatial filters outside a RDC Office, the following codes will create a release version that it is in accordance to the confidentiality vetting rules.

Confidentiality vetting is the process of reviewing the results to be released by the Research Data Center to ensure that confidentiality risks for StatCan respondents are minimized.

The following rules apply to our results:

- *Statistics must not be released for identifiable areas with less than 40 persons* ($\sum CompW1$ 40). Statistics must not be released for identifiable areas with less than 40 persons. This condition is usually met with geography levels used at the RDCs. The population threshold can be applied to the (long form) weighted population estimate.

The block above checks the total population of each DA:

```
vetting_da_mode <- census %>%
  group_by(PRCDDA, PwMode) %>%
  dplyr::summarize(Weighted_pop_mode = sum(CompW1)) %>%
         dplyr::select(PRCDDA, PwMode, Weighted_pop_mode)
```

Creating new files with the DA population:

```
spatial_filter_support <- spatial_filter %>%
  left_join(vetting_da_mode, by = c('PRCDDA' = 'PRCDDA','PwMode' = 'PwMode'))
```

Apply the rules to avoid confidentiality risks for the accessibility tables:

```
spatial_filter_support <- spatial_filter_support %>%
  filter(Weighted_pop_mode >= 40)

spatial_filter_release <- spatial_filter_support %>%
  filter(Weighted_pop_mode >= 40) %>%
  dplyr::select(-Weighted_pop_mode)
```

To finalize the methodology of this R markdown, we will export the processed data.

```
write.csv(spatial_filter_support, paste0(diretorio_export_tables, "spatial_filter_support.csv"), row.nam

write.csv(spatial_filter_release, paste0(diretorio_export_tables, "spatial_filter_release.csv"), row.nam
```

References

John Antonakis, Samuel Bendahan, Philippe Jacquart, and Rafael Lalive. On making causal claims: A review and recommendations. *The Leadership Quarterly*, 21(6):1086–1120, 12 2010. DOI: 10.1016/j.leaqua.2010.10.010. URL https://linkinghub.elsevier.com/retrieve/pii/S1048984310001475.

Jeroen Bastiaanssen, Daniel Johnson, and Karen Lucas. Does better job accessibility help people gain employment? the role of public transport in great britain. *Urban Studies*, 59(2), 05 2021. DOI: 10.1177/00420980211012635. URL https://journals.sagepub.com/doi/10.1177/00420980211012635.

Scott Cunningham. *7 Instrumental Variables*. Yale University Press, 01 2021. URL https://mixtape.scunning.com/07-instrumental_variables.

Elizabeth Delmelle, Isabelle Nilsson, and Providence Adu. Poverty suburbanization, job accessibility, and employment outcomes. *Social Inclusion*, 9(2):166–178, 05 2021. DOI: 10.17645/si.v9i2.3735. URL https://www.cogitatiopress.com/socialinclusion/article/view/3735.

Leandro Batista Duarte, Raul da Mota Silveira Neto, and Diego Firmino Costa da Silva. The relevance of job accessibility to labour market outcomes: Evidence for the são paulo metropolitan region. *Urban Studies*, 60(16), 04 2023. DOI: https://doi.org/10.1177/00420980231165481.

Lingqian Hu. Job accessibility and employment outcomes: which income groups benefit the most? *Transportation*, 44(6):1421–1443, 11 2017. DOI: 10.1007/s11116-016-9708-4. URL https://doi.org/10.1007/s11116-016-9708-4.

Jangik Jin and Kurt Paulsen. Does accessibility matter? understanding the effect of job accessibility on labour market outcomes. *Urban Studies*, 55(1):91–115, 01 2018. DOI: 10.1177/0042098016684099. URL https://doi.org/10.1177/0042098016684099. Publisher: SAGE Publications Ltd.

Daniel Johnson, Marco Ercolani, and Peter Mackie. Econometric analysis of the link between public transport accessibility and employment. *Transport Policy*, 60:1–9, 11 2017. DOI: 10.1016/j.tranpol.2017.08.001. URL https://www.sciencedirect.com/science/article/pii/S0967070X16303353.

Julie Le Gallo and Antonio Páez. Using synthetic variables in instrumental variable estimation of spatial series models. *Environment and Planning A: Economy and Space*, 45(9):2227–2242, 09 2013. DOI: 10.1068/a45443. URL https://journals.sagepub.com/doi/10.1068/a45443.

Gregorio Luz and Licinio da Silva Portugal. Understanding transport-related social exclusion through the lens of capabilities approach. *Transport Reviews*, 42(4):503–525, 07 2022. DOI: 10.1080/01441647.2021.2005183. URL https://www.sciencedirect.com/org/science/article/abs/pii/S0144164722004317.