Small Area Estimation for Crime Analysis

David Buil-Gil
Department of Criminology, University of Manchester, UK

25/05/2020

Abstract

Victimization surveys provide essential information to study crime and emotions about crime, but their sampling designs only allow analyzing criminological variables at large spatial scales. Crime surveys are designed to allow producing precise direct estimates (i.e., weighted means and totals) for very large areas, but the size of samples in small areas is generally small and direct estimates produced for small geographies will generally be imprecise. Refined model-based small area estimation techniques may be used to increase the reliability of small area estimates produced from victimization surveys. Small area estimation is the term used to describe those methods designed to produce reliable estimates of parameters of interest (and associated measures of reliability) for areas for which only small or zero sample sizes are available. In 2008, the US Panel to Review the Programs of the Bureau of Justice Statistics recommended the use of small area estimation to produce subnational estimates of crime. Since then, small area estimation has been applied to study many variables of interest in criminology. This chapter introduces theory and a step-by-step exemplar study in `R` to show the utility of small area estimation to analyze crime and place. Model-based regional estimates of confidence in policing are produced from European Social Survey data.

Keywords: Confidence in policing, European Social Survey, crime mapping, open data, GIS

Full reference: Buil-Gil, D. (2020). Small area estimation for crime analysis. In E. Groff & C. Haberman (Eds.), *The study of crime and place: A methods handbook*. Temple University Press.

Contact details: David Buil-Gil. G18 Humanities Bridgeford Street Building, Cathie Marsh Institute for Social Research, University of Manchester. E-mail address: *david.builgil@manchester.ac.uk*

ORCID ID: David Buil-Gil: 0000-0002-7549-6317.

Introduction

Foundations of small area estimation

Small area estimation applications for crime analysis

Small area estimation of trust in the police: Step-by-step example in `R`

European Social Survey

The European Social Survey is a biannual cross-national survey designed to measure social attitudes, beliefs and behaviors. It has been conducted since 2001 in more than 35 European countries, and allows for cross-national and cross-sectional comparisons of crime-related issues such as the confidence in police services, worry about crime and crime victimization experience in the last 5 years. The ESS sample is designed to be representative of all individual residents aged 15 or older who live in private households in each participant country, regardless of their nationality, citizenship or language.

Although participant countries are responsible for producing their own national sampling designs, all counties must collow common sampling principles. Namely, respondents must be selected following strict random probability techniques at every stage, sampling frames can be individuals, households or addresses, quota sampling is not allowed, and non-responding units cannot be replaced. Moreover, every country must select at least 1,500 effective respondents (or at least 800 in participant countries with less than 2 million citizens). As a consequence, countries with very different number of residents may select similar sample sizes, and all geographical levels below countries (e.g., regions, counties, cities) are not planned by the original sampling design and record small sample sizes.

Download European Social Survey data

ESS data can be downloaded from their website. But we can also download ESS data direcly into our `R` system using the `essurvey` package developed by Cimentada (2019). This package is designed to facilitate loading ESS survey data into `R`. It allows users to select the countries and years they are interested to analyze and loading them directly in `R` If this is the first time we are using this package, we need to install it by using the `install.packages()` function.

```r
install.packages("essurvey")
```

Once it is installed, we can load the package into our `R` environment using the `library()` function.

```r
library(essurvey)
```

In order to access ESS data in `R`, first we need to create our own personal account in the ESS online portal. ESS users only need to registed once, and then they can have open access to ESS data as many times as they wish. We need to access the ESS website and create a new account with our personal details: https://www.europeansocialsurvey.org/user/new. Filling the online registration form takes less than one minute, and once it is completed we will receive an email to confirm our registration process.

Once we are registered in the ESS platform, we can direcly import all ESS data into `R`. In this exercise we will download and analyze data from the 8th edition of ESS, which was published in 2016. We use the function `set_email()` from `essurvey` to save our email (the email account registered in the ESS platform) as a new environment variable, and then run the `import_rounds()` function to load ESS data from all participant countries. This may take a few seconds.

```r
set_email("your_email@domain.com") # change by your email

ess <- import_rounds(rounds = 8, ess_email = NULL, format = NULL)
```

Now we have loaded the ESS data and we can begin exploring and analyzing it. If we want to see the data, we can use the `View()` function.

Descriptive analyses

The ESS includes various questions that may be of interest for criminologists and crime analysts. For examples, some questions that we may be interested to analyze are:

1.- *"Have you or a member of your household been the victim of a burglary or assault in the last 5 years?"*

2.- *"How safe do you – or would you – feel walking alone in this area after dark?"*

3.- *"Using this card, please tell me on a score of 0-10 how much you personally trust each of the institutions I read out […]": "[…] the legal system" and "[…] the police".*

These measures have previously been used to study victimization, perceived safety, trust in the police, and trust in the legal system (e.g., REFS), but there are many other questions that may also be of interest for criminologists (e.g., racism, discrimination against immigrants, homophobia). We can read the whole ESS questionnaire here: https://www.europeansocialsurvey.org/docs/round8/fieldwork/source/ESS8_source_questionnaires.pdf.

In this exemplar study we will analyze ESS data about trust in police services, following previous research conducted by REFS. The variable name is `trstplc`, and it a Likert scale variable from 0 to 10, where 0 indicates the lowest level of trust, and 10 is the maximum value. We can begin by checking how this measure of trust in the police looks like. We will use the `summary()` function to obtain the summary statistics of this variable.

```
ess <- ess %>%
  mutate(trstplc = as.numeric(trstplc)) # transform variable to numeric

summary(ess$trstplc) # print summary statistics
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   5.000   7.000   6.399   8.000  10.000     320
```

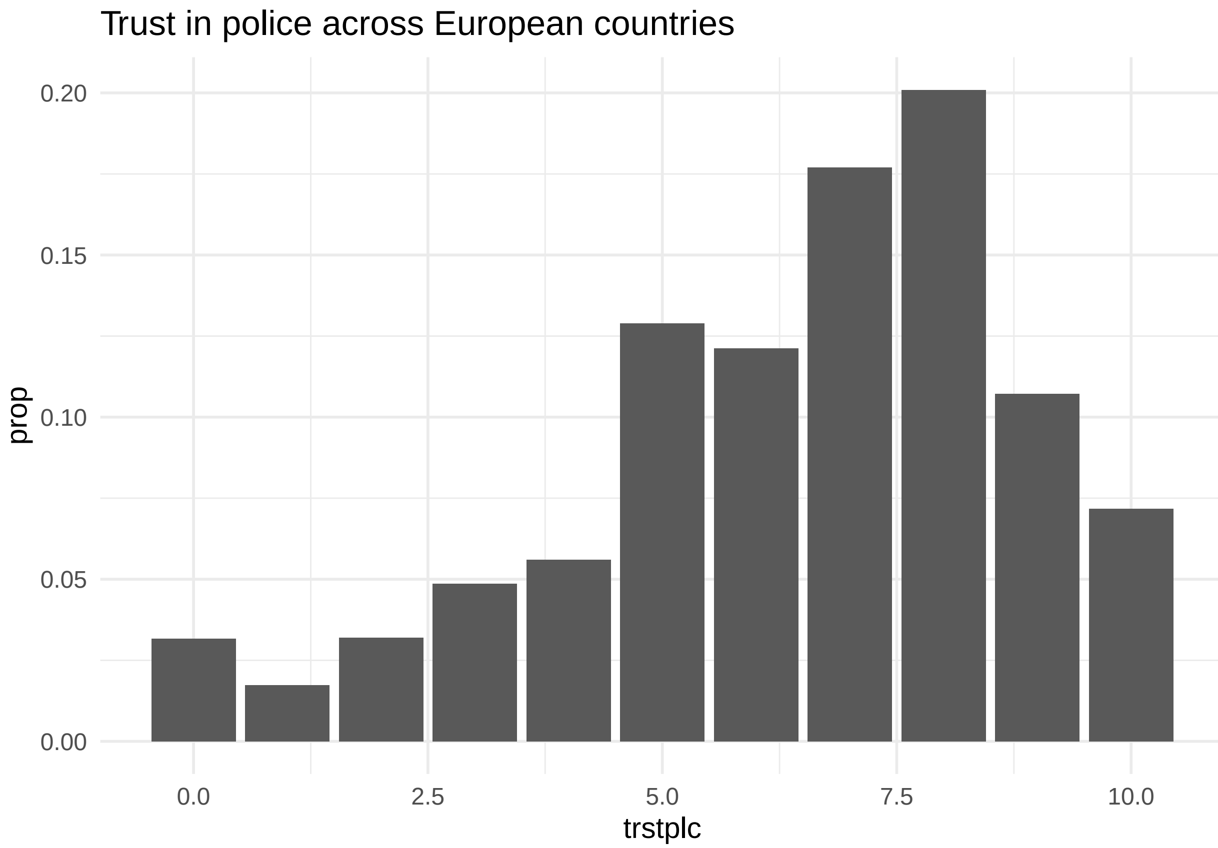
We see that the average score of trust in police in Europe is 6.4, and the median value is 7. We can use the same `summary()` function to compare the values of trust in police services with the citizens' trust in other social institutions, such as the legal system (variable `trstlgl`), politicians (`trtplt`), political parties (`trtprt`), the country's parliament (`trstprl`) or the United Nations (`trstun`). On average, we measures of trust in the police appear to be higher than the Europeans' trust in other key political and legal institutions.

Moreover, we can obtain some more detailed information about the citizens' trust in police services by counting the frequency of respondents that chose each score and creating a bar plot to visualize their distribution. We will use functions from the the packages `dplyr` (Wickham, François, et al., 2020) and `ggplot2` (Wickham, Chang, et al., 2020) for this. More specifically, we use the `group_by()` function from `dplyr` to create groups of respondents based on their score of trust in police, and the functions `summarize()` and `mutate()` from the same package to save the results in two columns showing the number and proportion of respondents in each category. We save this new table in a new dataset called `trust_poli`.

```
trust_poli <- ess %>%
  group_by(trstplc) %>%     # categories based on level of trust
  summarize(n = n()) %>%    # number of respondents per group
  mutate(prop = n / sum(n)) # proportion respondents per group
```

Then, we use the `ggplot()` and `geom_bar()` functions from `ggplot2` to create a bar graph of the number of responses per category. Before plotting this visualization, however, we will run the function `theme_set(theme_minimal())` to set a basic, neat theme for all our plots.

```
theme_set(theme_minimal()) # set white theme for plots

ggplot(data = trust_poli, aes(x = trstplc, y = prop)) + # set variables of interest
  geom_bar(stat="identity") +                           # plot bar graph
  ggtitle("Trust in police across European countries")  # change title
```



We see that few respondents have a low trust in the police, whereas most European citizens seem to trust their police forces quite a lot. This plot, nevertheless, may hide internal heterogeneity between European regions and countries. Based on this bar graph alone, we do not have enough information to be able to know if residents in all participant countries have the similar levels of trust in the police, or whether those respondents with very low or very high confidence in the police concentrate in some countries but not others. We will use small area estimation to produce estimates of trust in the police across European regions.

Since we are particularly interested in analyzing which regions in Europe have more and less trust in police services, and not only what is the level of trust in each area, we will produce regional estimates of the proportion of citizens who have a level of trust above the average in Europe. In other words, our estimates will show a value between 0 and 1 representing which proportion of residents have more trust in the police than the average of European citizens. For instance, a value of 0.6 in a given region would indicate that 60 percent of its residents have more trust in the police than the European average. Thus, we need to recode our variable of interest, and we will use the `mutate()` and `ifelse()` functions from `dplyr` to do so. Those respondents with a score above or equal to the mean will be given a value 1, whereas others will be assigned a value of 0. This will facilitate the interpretation of our results, but future research can explore producing estimates from the original 0-to-10 Likert scale. We will also delete all those respondents who did not answer this question (i.e., *NA*s).

```
ess <- ess %>%
  # if trust is above or equal to mean, 1, 0
  mutate(trstplc = ifelse(trstplc >= mean(trstplc, na.rm = T), 1, 0)) %>%
  filter(!is.na(trstplc)) # delete NAs
```

We can use the `group_by()` and `summarize()` functions seen above to explore how our recoded variable looks like. We can see, for example, that 24721 out of 44067 (i,e., 56.1% of participants) have more trust in the police than the average in Europe.

Exploring spatial data: Coverage and sample sizes

As mentioned above, the ESS sampling design is planned to allow producing reliable direct estimates at the level participant countries, but samples recorded at smaller scales (e.g., regions, cities) may be too small in some areas to allow producing direct estimates of adequate precision. We can check how big ESS sample sizes are in each region (variable `region`) using the functions `filter()`, `group_by()` and `summarize()` from `dplyr` to create a summary table in a new dataframe that we will call `sample_region`. Then, we can use the `summary()` function to print the summary statistics of area sample sizes.

```
sample_region <- ess %>%
  filter(region != 99999) %>% # filter out NAs
  group_by(region) %>%        # categories based on regions
  summarize(n = n())          # calculate sample size

summary(sample_region$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     7.0    60.0   115.0   160.8   209.8  2524.0
```

The average sample size per region is 160.8, which is quite large but may be insufficient to produce reliable direct estimates. Moreover, there are areas with very small sample sizes (the minimum area sample size is 7), where we cannot simply rely of direct estimation techiques to generate estimates of adequate precision.

Moreover, we also need to consider that participant countries can decide whether they want to publish recorded data at the level of NUTS-1, NUTS-2, NUTS-3 or smaller scales. NUTS is the acronym of *Nomenclature of Territorial Units for Statistics*, and it refers to the spatial scales used by the European Union and Eurostat (the statistical office of the European Union) for policy making and statistical reporing purposes. NUTS are basically a way to organize European countries in regions and subregions. In England, for instance NUTS-1 are statistical regions, NUTS-2 are counties (and groups of districts in London), and NUTS-3 are generally unitary authorities (some grouped). Whereas some participant countries publish their data at the level of NUTS-2, others decide to report information for NUTS-1 or NUTS-3 areas. We can check which level of aggregation is published by each participant country in the ESS website: https://www.europeansocialsurvey.org/data/multilevel/guide/essreg.html. We can also run the following lines of code and print this information directly in `R`.

```
ess %>%
  group_by(regunit, cntry) %>% # group by spatial scale and country
  summarize(n = n())           # print sample size per country
```

We see that many counties publish data at the NUTS-2 level, but others participant countries publish their micro-data for NUTS-1 and NUTS-3 areas. We will aggregate data at the NUTS-2 level and produce estimates of confidence in the police at this scale (with the exception of Germany and the UK, who only publish data for NUTS-1).

Converting spatial data into NUTS-2

In order to convert the spatial information provided by all countries into NUTS-2 geographies, we first need to load a lookup table that details which NUTS-3 areas are part of which NUTS-2. I have previously created and saved a lookup table in csv format in an open access Github repository, but similar tables are also available in other formats from the ESS platform: https://www.europeansocialsurvey.org/data/multilevel/guide/bulk.html. In order to load the lookup table in `R`, we can use the `getURL()` functions from `RCurl` [ADD REF] and `read.csv()`.

```
library(RCurl)

url_lookup <- getURL("https://raw.githubusercontent.com/davidbuilgil/SAE_chapter/master/data/NUTS_looku

lookup <- read.csv(text = url_lookup)
```

Now, we can create a new column in the original ESS data that specifies the regions for which we aim to produce small area estimates of trust in the police. We will merge the lookup table with the original ESS data using a `left_join()` function and create a new column called `domain` which shows the NUTS-2 areas (or NUTS-1 in Germany and UK) for which we will produce estimates.

```
ess <- ess%>%
  left_join(lookup, by = c("region" = "nuts3")) %>%          # merge lookup into ESS dataset
  rename(domain = nuts2) %>%                                 # rename NUTS2 variable
  mutate(domain = as.character(domain),                      # convert NUTS2 into character
         domain = ifelse(is.na(domain), region, domain)) %>% # copy NUTS1 data if no NUTS2 information
  filter(!(domain == 99999))                                 # delete NAs
```

Now our data is clean and ready to be used to produce estimates of trust in the police at a regional level.

Producing direct estimates

We will produce direct estimates based on the Horvitz-Thompson estimator (Horvitz & Thompson, 1952), which is one of the most common approaches to produce direct estimates. It makes use of original survey data and survey weights to obtain design-unbiased estimates in each small area, but direct estimates may suffer from high variance and unreliability in those areas with small sample sizes. Moreover, estimates cannot be produced in areas with zero samples. We will produce Horvitz-Thompson estimates of the trust in police for European regions, but it is very likely than many estimators will not show adequate levels of precision. Model-based SAE approaches are needed when direct estimates are not precise enough.

In order to produce small area estimates, we will use the `sae` package [REFS]. We need to install it and load it into your `R` system.

```
library(sae)
```

The Horvitz-Thompson estimator takes into account the population size in each area, and assumes that survey weights adjust our sample to the total population. Thus, we need to know how many people live in each region, and ensure that our weights adjust the sample to the population size. I have previously downloaded the population sizes from Eurostat and uploaded a clean dataset onto Github. Downloading data from sources of official statistics, such as Eurostat, usually mean having to spend some time cleaning the data and selecting those variables that adjust to our research needs. For the purpose of this exercise, I have cleaned the data and uploaded onto an online repository, but later we will also see how to load Eurostat data into our `R` environments.

```
url_pop <- getURL("https://raw.githubusercontent.com/davidbuilgil/SAE_chapter/master/data/population.cs

pop <- read.csv(text = url_pop)
```

```
pop <- pop %>%
  mutate(area = 1:n()) %>%                       # create numeric id value
  rename("domain" = "X.U.FEFF.domain") %>% # rename region column name
  subset(select = c(1, 3, 2)) %>%          # reorder columns
  filter(domain %in% ess$domain)           # filter out areas not present in ESS
```

Now we have almost all information necessary to produce our direct estimates: the variable of interest (variable `trstplc` in the `ess` dataset), the area population size (`pop2014` in `pop` dataset), and spatial information that matches in both datasets. Nevertheless, as introduced above, the Horvitz-Thompson estimator also requires the use of survey weights that adjust our sample to the population size. Given that the weights published by ESS are not designed to let respondents represent a specific number of citizens, but instead they were computed to adjust the sample to the population characteristics, we will need to recalibrate the ESS weights to the population sizes per region. We can do this by running the following lines of code:

```
ess_w_area <- ess %>%
  filter(domain %in% pop$domain) %>%        # filter out areas not present in population dataset
  group_by(domain) %>%                      # create groups by region
  summarise(w_sum = sum(pspwght * pweight)) # sum weights per region

ess <- ess %>%
  filter(domain %in% pop$domain) %>%          # filter out areas not present in population dataset
  left_join(ess_w_area, by = "domain") %>%    # merge sum of weights with ESS units
  left_join(pop, by = "domain") %>%           # merge region population sizes
  mutate(weight = pspwght * pweight,          # compute weights for cross-national analysis
         weight = (weight * pop2016) / w_sum) # recalibrate weights to population sample size
```

After a few steps, we now have all necessary information to produce our direct estimates of confidence in policing. We use the `direct()` function from `sae` to produce Horvitz-Thompson estimates in each region. It will also produce the Coefficient of Variation of each estimate, which will be used to assess the reliability of these direct estimates.

```
dir <- direct(y       = ess$trstplc,
              dom     = ess$area,
              sweight = ess$weight,
              domsize = pop[,2:3],
              replace = FALSE)
```

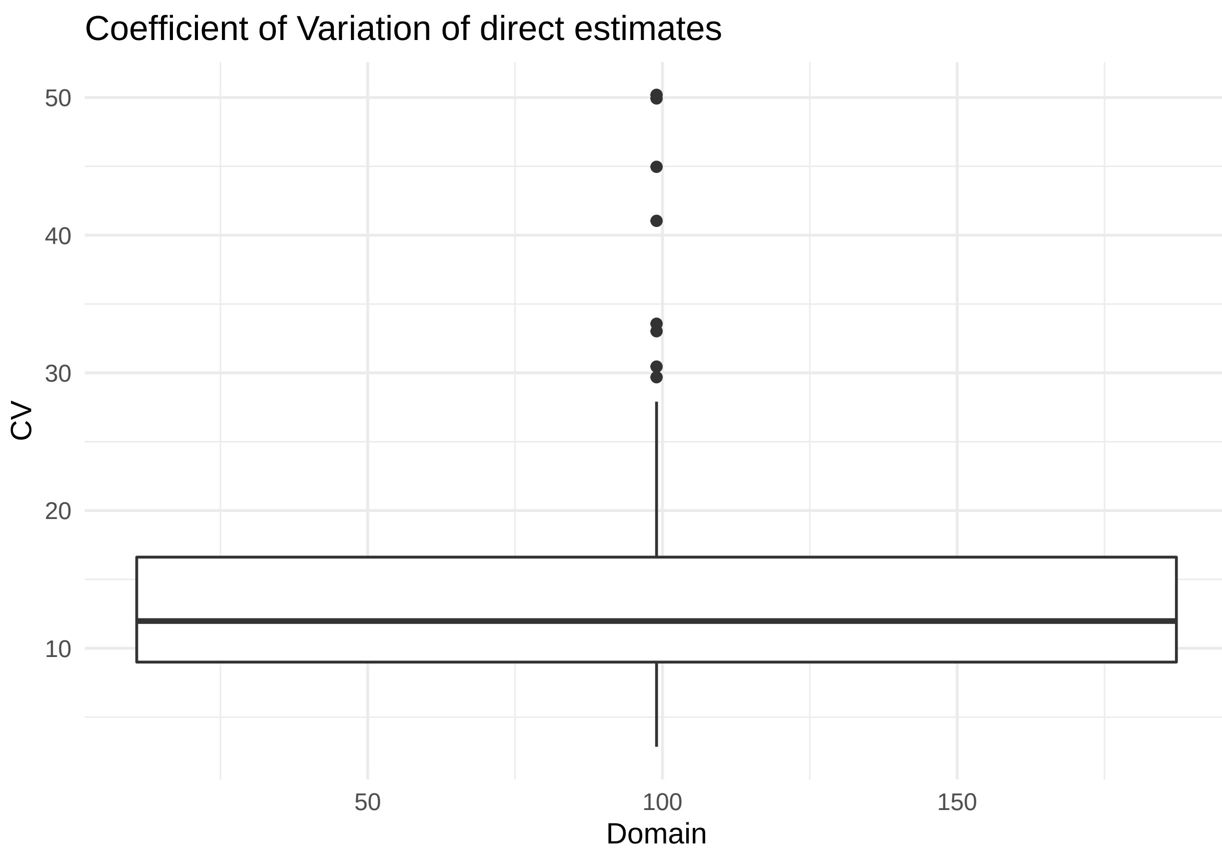
Exploring direct estimates

Once we have produced our direct estimates of trust in the police, we can see how these look like by using some functions introduced above.

```
summary(dir$Direct) # summary statistics of direct estimates
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1783  0.4877  0.5815  0.5838  0.6704  0.9006
```

```
# produce boxplot of coefficients of variation
ggplot(dir, aes(x=Domain, y=CV)) +
  geom_boxplot() +
  ggtitle("Coefficient of Variation of direct estimates")
```



As you can see in the boxplot, the estimates of the majority of regions have a Coefficient of Variation smaller than 20%, which is a very good indicator of reliability of these estimates; but we also have a few regions with Coefficients of Variation larger than 25%. We can improve the accuracy of these estimates by using model-based small area estimation models.

For now, we can merge our direct estimates into the dataset of area-level information by using the `left_join()` function from 'dplyr.

```
pop <- pop %>%
  left_join(dir, by = c("area" = "Domain"))
```

Downloading area-level covariates

In order to fit area-level models of trust in police and produce area-level estimates, we will need area-level covariates that are associated with our variable of interest. ADD SOME LITERATURE ON COVARIATES!!!

We can download various area-level covariates from Eurostat using the **`eurostat`** package (Lahti et al., 2020), which has been created by facilitate downloading data from Eurostat into `R`. Eurostat is a very large data repository that publishes large datasets of social, econonomic and demographic information for European countries and regions. We can use the `search_eurostat()` function to search predefined key words associated with variables of interest for our study. The function will return a list of all datasets including our keywords, and can then explore which of them are more suitable for our study. For example, we may want to know if education levels and crime rates are somehow associated with the regional levels of trust in the police, and thus we can search Eurostat datasets that include the words *"education"* and *"offender"*. I have done this search and found various variables of interest, but you can also try this at home and probably you will also find variables of interest for our models.

```r
library(eurostat)

eurostat_edu   <- search_eurostat("education") # search datasets about education
eurostat_crime <- search_eurostat("offender")  # search datasets about crime
```

Once we know the codes of the datasets we are interested to do, we can use the `get_eurostat()` function to import these into our `R` environment. For example, the dataset `edat_lfs_9918` includes information about the proportion of citizens between 15 and 64 in each NUTS-2 that have a higher education degree. We can download this dataset and see how it looks like:

```r
he <- get_eurostat(id = "edat_lfs_9918")
```

If we open this file (using the `View()` function), we can see that it is includes information about many indicators, years, age groups, spatial scales and divided by sex. All datasets imported from Eurostat provide information abou many different measures, which means that we will need to spend some time wrangling and subsetting these data to make sure we can attach these to our area-level direct estimates to estimate the area-level models needed to produce model-based estimates. For the purpose of this exemplar study, I have previously searched for datasets of interest, downloaded and cleaned their data, and merged all covariates into a unique dataset. We can load this dataset into `R` using the functions provided by `RCurl` package, but you can also spend some time trying to find better, more suitable covariates in the Eurostat website. Moreover, the ESS website also publishes interesting area-level covariates at the different scales: https://www.europeansocialsurvey.org/data/multilevel/guide/bulk.html.

```r
url_covs <- getURL("https://raw.githubusercontent.com/davidbuilgil/SAE_chapter/master/data/covs_short.c

covs <- read.csv(text = url_covs)

pop <- pop %>%
  left_join(covs, by = "domain") # merge covariates with direct estimates
```

rate crimes * 10000

Describe variables like this: - *BLA*: bla bla bla

Once all our covariates are clean and ready to use, we can briefly explore them using the `dplyr` package. For instance, we may want to know the number of missing values in each covariate:

```r
pop %>%
  dplyr::select(fem_p_16, gdp_eurhab_16, robb_r_10, burg_r_10, he_p_16, medage_16) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
##   fem_p_16 gdp_eurhab_16 robb_r_10 burg_r_10 he_p_16 medage_16
## 1        0            13        22        22      10        10
```

Impute missing values

Multiple Imputation using Bootstrap and PMM

```r
library(Hmisc)

fun_input <- aregImpute(~ fem_p_16 + gdp_eurhab_16 + robb_r_10 + burg_r_10 + he_p_16 + medage_16,
                  data = pop, n.impute = 10)
```

```r
imputed <- as.data.frame(impute.transcan(fun_imput,
                                         imputation = 1,
                                         rhsImp = "mean",
                                         data = pop,
                                         list.out = T))
```

```r
pop <- pop %>%
  dplyr::select(domain, area, pop2016, SampSize, Direct, SD, CV) %>%
  cbind(imputed)
```

Fitting area-level models and predicting synthetic estimates

And we will also substract all independent variables from the mean and divide these by two standard deviations, as suggested by Gelman (2008), which will allow us to obtain standardised coefficients not affected by the dimensions of each variable:

```r
model <- lm(Direct ~  fem_p_16  + gdp_eurhab_16 + robb_r_10 +
                      burg_r_10 +  medage_16    + he_p_16,
            data = pop)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.00	0.06	0.0	1.00
Proportion females	-0.24	0.07	-3.2	0.00
GDP per person (€)	0.28	0.09	3.2	0.00
Robbery rate	-0.07	0.07	-1.0	0.33
Burglary rate	-0.01	0.07	-0.1	0.93
Median age	0.18	0.07	2.6	0.01
Proportion HE	0.25	0.08	3.1	0.00

Table 1: Area-level model of trust in the police (standadized coefficients)

Check r squared

```r
summary(model)$r.squared
```

0.35

We can also predict the synthetic estimates from our model. Synthetic estimation is the umbrella term used to describe the group of SAE techniques that produce small area estimates by fitting a regression model with area-level direct estimates as the dependent variable and relevant area-level auxiliary information as covariates and then computing regression-based predictions (i.e., synthetic estimates). Synthetic estimators may be based, for example, on area-level linear models (e.g., Brugal et al., 1999), logistic models (e.g., Hser et al., 1998), multilevel models (e.g., Taylor, 2013; Whitworth, 2012) and spatial models (e.g. Wheeler et al., 2017).

Regression-based synthetic estimates can be produced for all areas regardless of their sample size (also areas with zero sample sizes). However, these are not based on a direct measurement of the variable in each area and suffer from a high risk of producing biased small area estimates (Levy, 1979; Rao. & Molina., 2015).

We use the `predict()` function to product synthetic estimates from our area-level linear model.

```r
synthetic <- predict(model) # predict synthetic estimates
```

```
pop <- pop %>%
  cbind(synthetic)
```

Producing EBLUP estimates

Using the same variables, we also fit our EBLUP (i.e., Empirical Best Linear Unbiased Predictor) model to produce model-based small area estimates.

The area-level EBLUP, which is based on the model developed by Fay III & Herriot (1979), obtains an optimal combination of direct and regression-based synthetic estimates in each small area. The EBLUP combines both estimates in each area and gives more weight to the direct estimate when its sampling variance is small, while more weight is attached to the synthetic estimate when the direct estimate's variance is larger. The EBLUP reduces the variance of direct estimates and the risk of bias of synthetic estimates by producing the optimal combination of these in each area.

We use the `eblupFH()` function from `sae` package.

```
eblup <- eblupFH(formula = pop$Direct    ~ pop$fem_p_16  + pop$gdp_eurhab_16 +
                           pop$robb_r_10 + pop$burg_r_10 + pop$medage_16 +
                           pop$he_p_16,
                 vardir  = pop$SD^2,
                 method  = "REML")
```

```
eblup$fit # print model results
```

We can get the EBLUP model results by using the `summary()` function. describe min, mean and max here

And finally we can merge the model-based small area estimates into our main dataset of area-level information. We use the `cbind()` function to merge the new columns.

```
pop <- pop %>%
  cbind(eblup$eblup) %>% # merge data into main dataset
  rename(eblup = "eblup$eblup") # change name of column
```

Mapping the confidence in police work in Europe

Now we will load a shapefile of combined NUTS regions for all European countries. We will use the `eurostat_geodata_60_2016` shapefile, which is already saved in the `eurostat()` package. This dataset contains spatial information for all NUTS regions across various spatial scales, which will enable us to recode the NUTS-3 data into NUTS-2 codes.

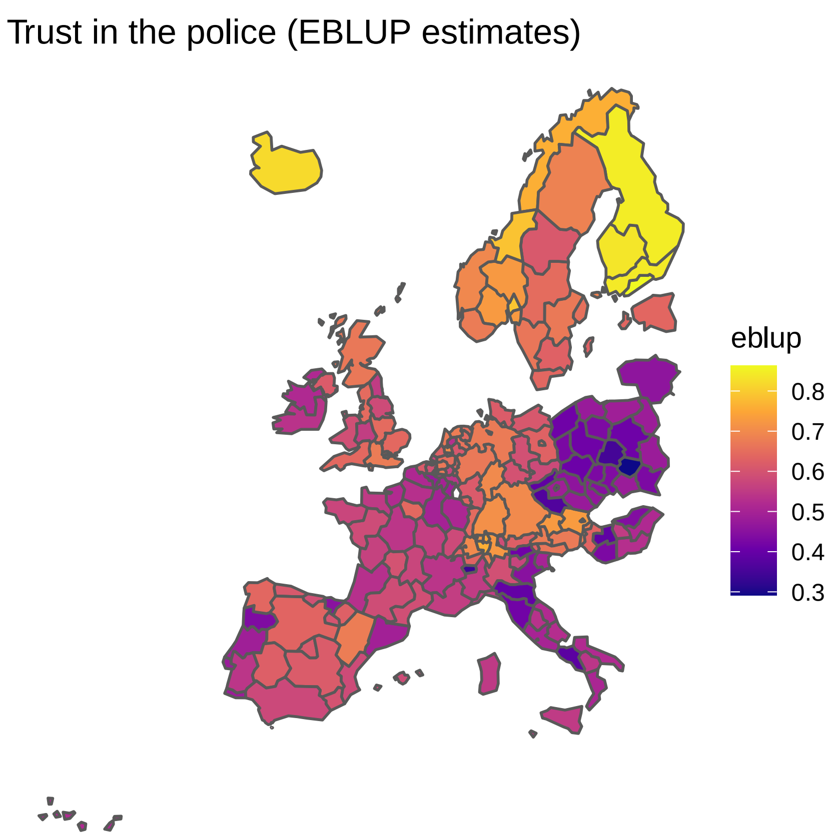
```
library(sf)

# download geojson
nuts <- st_read("https://raw.githubusercontent.com/davidbuilgil/SAE_chapter/master/shapefile/nuts_ess8.

st_crs(nuts) <- 15752 # change CRS to ED79 (EPSG:4668 with transformation)
```

And the last step is to map our small area estimates in Europe. We can use the following codes to prepare our shapefile:

```
geodata <- nuts %>%
  rename("domain" = "NUTS_ID") %>%
  left_join(pop, by = "domain") %>%
  filter(!is.na(Direct))
```

And this to visualise our maps of Direct and EBLUP estimates.

```
ggplot(data = geodata) +
  ggtitle("Trust in the police (EBLUP estimates)") +
  geom_sf(aes(fill = eblup)) +
  theme_void() +
  scale_fill_viridis_c(option = "plasma")
```



Computing the Mean Squared Error of EBLUP estimates

In SAE, each small area estimate needs to be accompanied by its estimated measure of uncertainty, which is frequently defined by the Mean Squared Error (MSE) or the Relative Root Mean Squared Error (RRMSE). The MSE is a measure of the estimate's reliability and refers to the averaged squared error of the estimate. Hence, it represents the squared difference between the estimated value and what is measured. The MSE is always non-negative, and values closer to zero indicate a higher reliability of the small area estimate. The MSE accounts for both the variance of the estimates (i.e., spread of estimates from one sample to another) and their bias (i.e., distance between the averaged estimated value and the true value). The RRMSE is obtained by taking the square root of the MSE (i.e. the Root Mean Squared Error, RMSE) and dividing it by the corresponding small area estimate. The RRMSE is usually presented as a percentage. This allows for direct comparisons between the measures of reliability of estimates obtained from direct and indirect model-based SAE techniques.

The RRMSE can be used to examine which SAE method produces the most reliable estimates and which estimates suffer from inadequate reliability. SAE methods may produce reliable estimates in some areas and unreliable estimates in others. SAE standards tend to establish that "estimates with RRMSEs greater than 25% should be used with caution and estimates with RRMSEs greater than 50% are considered too unreliable for general use" (CDSS, 2015, p. 13).

The measure of uncertainty of direct estimates is defined by their Coefficient of Variation (CV), which is the corresponding measure to the RRMSE for unbiased estimators (Rao. & Molina., 2015).

RRMSEs of model-based estimates can be estimated following analytical and bootstrap procedures. In this exemplar study we will produce the RRMSE of our estimates by following an analytical approach: we use the `mseFH()` function from `sae`.

```
eblup_mse <- mseFH(formula = pop$Direct    ~ pop$fem_p_16  + pop$gdp_eurhab_16 +
                         pop$robb_r_10 + pop$burg_r_10 + pop$medage_16 +
                         pop$he_p_16,
                   vardir  = pop$SD^2,
                   method  = "REML")
```

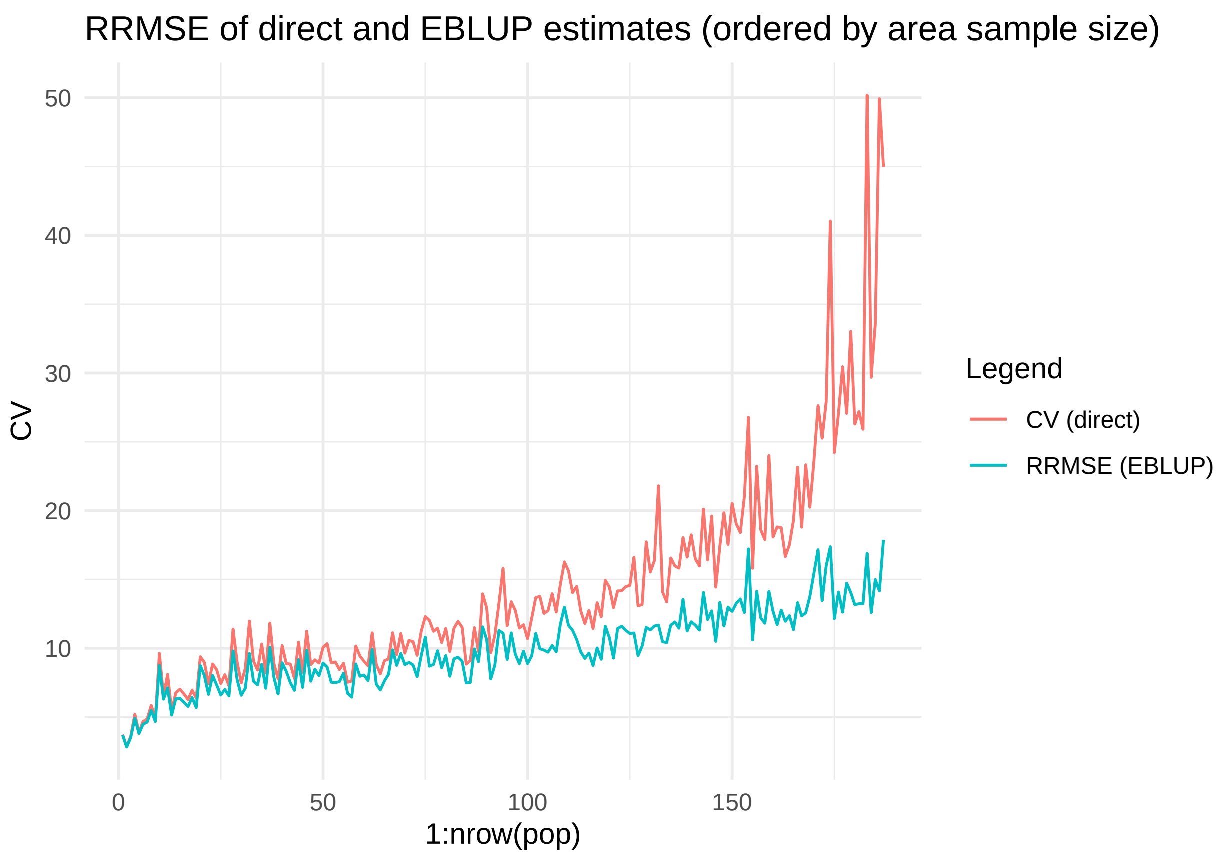
And we will also merge this information into our main dataset:

```
pop <- pop %>%
  cbind(eblup_mse$mse) %>% # merge data
  rename(mse = "eblup_mse$mse") %>% # change name of column
  mutate(rrmse = (sqrt(mse) / eblup) * 100) # compute RRMSE from MSE
```

Plotting the Mean Squared Error of EBLUP estimates

Finally, we will also analyse to what extent our EBLUP small area estimates are more reliable than the original direct estimates. We will plot the RRMSE of EBLUP estimates and the CV of direct estimates using the `ggplot2` package.

```
pop %>%
  arrange(desc(SampSize)) %>%
  ggplot() +
  geom_line(aes(y = CV, x = 1:nrow(pop), color = "darkred")) + # create red line of direct estimates' C
  geom_line(aes(y = rrmse, x = 1:nrow(pop), color="steelblue")) + # create blue line of EBLUP's RRMSE
  scale_color_discrete(name = "Legend", labels = c("CV (direct)", "RRMSE (EBLUP)")) +
  ggtitle("RRMSE of direct and EBLUP estimates (ordered by area sample size)")
```



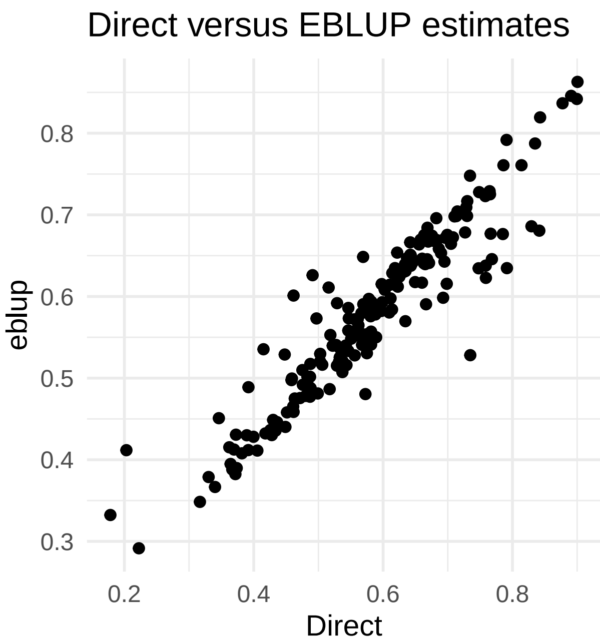
Our small area estimates are more reliable in all areas, and the increased precision is very large is some cases.

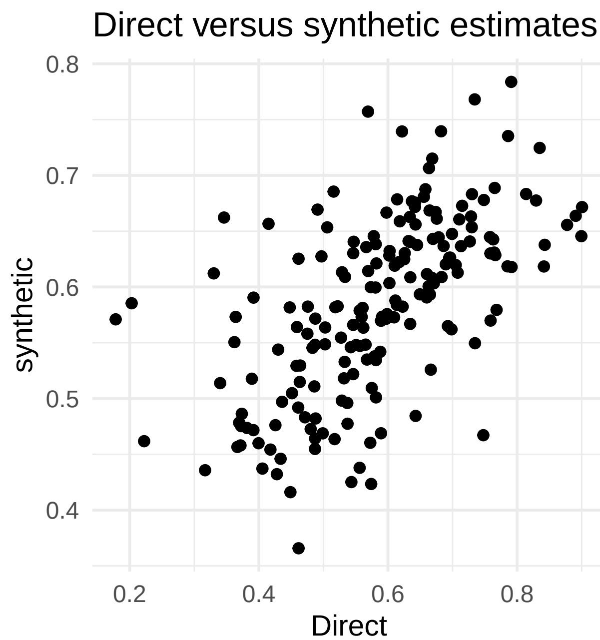
Model diagnostics

Diagnostics of our EBLUP estimates are presented below to examine whether our estimates are biased by the models and to check the model's validity.

We start by producing a scatter plot of direct estimates against the EBLUP estimates. Regarding that direct estimates are design-unbiased, we expect a high linear correlation between direct and model-based estimates.

```
library(gridExtra)

# plot direct estimates against EBLUPs
dir_vs_EBLUP <- ggplot(pop, aes(x = Direct, y = eblup)) +
                geom_point() +
                ggtitle("Direct versus EBLUP estimates") +
                theme(aspect.ratio = 1)

# plot direct estimates against synthetic estimates
dir_vs_synth <- ggplot(pop, aes(x=Direct, y=synthetic)) +
                geom_point() +
                ggtitle("Direct versus synthetic estimates") +
                theme(aspect.ratio = 1)

grid.arrange(dir_vs_EBLUP, dir_vs_synth, ncol=2)
```





dir vs eblup $= 0.93$ (p-value < 0.0001)

dir vs synth $= 0.64$ (p-value < 0.0001)

The scatter plot and the Spearman's rank correlation coefficient show a high linear association between our model-based estimates and the unbiased direct estimates, which shows that our model does not bias our final small area estimates.

We will do the same with the synthetic estimates produced directly from the model, just to see the extent to which model-based synthetic estimates may be biased by the model

The scatter plot shows that many estimates are likely to be affected by bias arising from the model.

We can also calculate the model standardised residuals and present the q-q plots of residuals in order to check their normality.

Final remarks

Author bio

David Buil-Gil is a Research Fellow at the Department of Criminology of the University of Manchester, UK, and a member of the Cathie Marsh Institute for Social Research at this same university. His research interests cover small area estimation applications in criminology, environmental criminology, crime mapping, emotions about crime, crime reporting, new methods for data collection and open data.

Acknowledgments

The author would like to thank Samuel H. Langton and Angelo Moretti for comments that greatly improved the manuscript.

References

Brugal, M. T., Domingo-Salvany, A., Maguire, A., Caylà, J. A., Villalbí, J. R., & Hartnoll, R. (1999). A small area analysis estimating the prevalence of addiction to opioids in barcelona, 1993. *Journal of Epidemiology & Community Health*, *53*(8), 488–494.

CDSS. (2015). *Survey of disability, ageing and carers, 2012. Modelled estimates for small areas, projected 2015.* Commonwealth Department of Social Services, Australian Bureau of Statistics, Release 1.

Cimentada, J. (2019). *Essurvey: Download data from the european social survey on the fly.* https://cran.r-project.org/web/packages/essurvey/essurvey.pdf

Fay III, R. E., & Herriot, R. A. (1979). Estimates of income for small places: An application of james-stein procedures to census data. *Journal of the American Statistical Association*, *74*(366a), 269–277.

Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations. *Statistics in Medicine*, *27*(15), 2865–2873.

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, *47*(260), 663–685.

Hser, Y. I., Prendergast, M., Anglin, M. D., Chen, J. K., & Hsieh, S. C. (1998). A regression analysis estimating the number of drug-using arrestees in 185 us cities. *American Journal of Public Health*, *88*(3), 487–490.

Lahti, L., Huovari, J., Kainu, M., & Biecek, P. (2020). *Eurostat: Tools for eurostat open data.* https://CRAN.R-project.org/package=eurostat

Levy, P. S. (1979). Small area estimation-synthetic and other procedures, 1968-1978. *Synthetic Estimates for Small Areas: Statistical Workshop Papers.*

Rao., J. N. K., & Molina., I. (2015). *Small area estimation. Second edition.* Wiley.

Taylor, J. L. M. (2013). *Small area synthetic estimation of perceptions of alcohol and drug-related anti-social behaviour* [PhD thesis]. University of Portsmouth.

Whitworth, A. (2012). Sustaining evidence-based policing in an era of cuts: Estimating fear of crime at small area level in england. *Crime Prevention and Community Safety*, *14*(1), 48–68.

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., Woo, K., Yutani, H., & Dunnington, D. (2020). *Ggplot2: Create elegant data visualisations using the grammar of graphics.* https://CRAN.R-project.org/package=ggplot2

Wickham, H., François, R., Henry, L., & Müller, K. (2020). *Dplyr: A grammar of data manipulation.* https://CRAN.R-project.org/package=dplyr