Exploratory analysis of accesses to support centers for gender-based violence in Apulia

Data

The dataset employed regards the counts of accesses to gender-based violence support centers in the Apulia region by residence municipality of the women victims of violence during 2022. `R` codes to generate the dataset are in the R script posted here which this report is based on. Observational period are years 2021, 2022, 2023.

Here, we only take into account the violence reports which support centers actually take charge of, at the risk of underestimating the counts of gender-based violence cases. This choice is driven by the need of avoiding duplicated records, since e.g. it may happen that a support center redirects a victim to another support center.

In order to avoid singletons in the spatial structure of the dataset, we removed the Tremiti Islands from the list of municipalities included (0 accesses recorded so far).

Therefore, the municipality-level dataset in scope consists of 256 observations.

We can only take into account the accesses to support centers for which the origin municipality of victims is reported; therefore the total count of accesses in scope is 1477, 1516 and 1822 for the three reference years respectively.

Here, we plot the log-access rate per residence municipality, i.e. the logarithm of the ratio between access counts and female population. Blank areas correspond to municipalities from which zero women accessed support centers (82 municipalities).

Covariates

Our target is explaining the number of accesses to support centers, y, defined at the municipality level, on the basis of a set of candidate known variables. Unfortunately, these data are only available for year 2021. y is modelled with simple Poisson regression.

We have at disposal a number of candidate explanatory variables, which include the distance of a municipality from the closest support center and a set of variables measuring social vulnerability under different dimensions; these latter covariates are provided by the ISTAT.A more detailed description of these covariates is in this excel metadata file.

All covariates are scaled to have null mean and unit variance.

- TEP, i.e. the distance of each municipality from the closest municipality hosting a support center. Distance is measured by road travel time in minutes (acronym TEP stays for Tempo Effettivo di Percorrenza, i.e. Actual Travel Time). Since to the best of our knowledge the list of active support centers changed between 2022 and 2023, we employ the list of centers active until 2022 for 2021-2022 data, and the list of centers active in 2023 for 2023 data.
- AES, the distance from the closest infrastructural pole, always measured in travel time.
- MFI, i.e. the decile of municipality vulnerability index.
- PDI, i.e. the dependency index, i.e. population either ≤ 20 or ≥ 65 years over population in $[20 - 64]$ years.

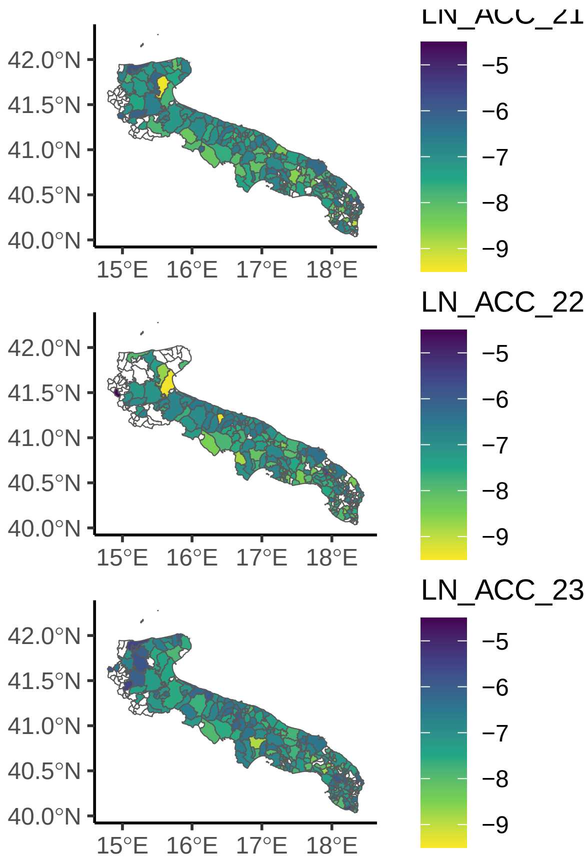


Figure 1: Log-access rate

- ELL, i.e. the proportion of people aged $[25-54]$ with low education.
- ERR, i.e. employment rate among people aged $[20-64]$.
- PGR, i.e. population growth rate with respect to 2011.
- UIS, i.e. the ventile of the density of local units of industry and services (where density is defined as the ratio between the counts of industrial units and population).
- ELI, i.e. the ventile of employees in low productivity local units by sector for industry and services.

First, we visualise the correlations among these explanatory variables:

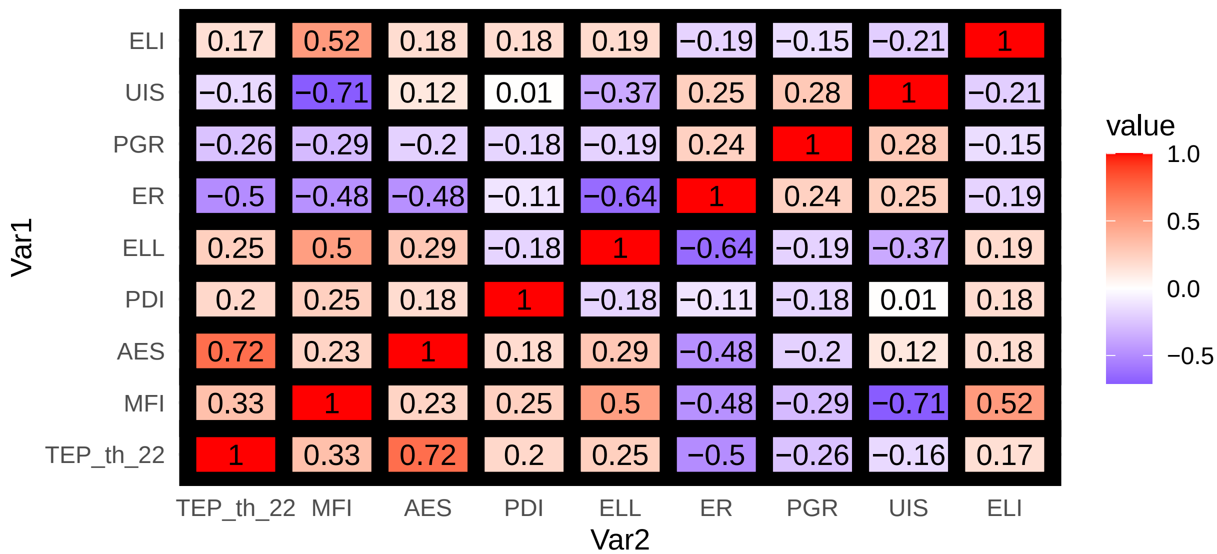


Figure 2: Correlations in explanatory variables

We see the correlation between the two distances is very high (0.72), and so is the correlation between the fragility index decile and the density of productive units.

In the first case, we drop the distance from the nearest infrastructural pole. In the latter we drop `MFI`, which is a combination of all covariates except for `TEP_th`, and is a weakly informative choice.

Nonspatial regression

We regress the counts of accesses y to support centers on the aforementioned explanatory variables. To estimate regression coefficients, all covariates are scaled to zero mean an unit variance.

$$y_{it} \mid \eta_{it} \sim \text{Poisson}(E_{it}\, e^{\eta_{it}}) \quad \text{where} \quad \eta_{it} = X_{it}^{\top}\alpha \tag{1}$$

Where X are the covariate defined earlier, α are covariate effects, and E_{it} is the female population aged ≥ 15 in municipality i and year t.

To gain more insight on the role of all explanatory variables we show the posterior summaries of the full regression model

% latex table generated in R 4.4.1 by xtable 1.8-4 package % Fri Mar 14 11:28:27 2025

- `TEP_th_22`: The distance from the closest support center appears to play an important role. The easiest interpretation is that the physical distance represents a barrier to violence reporting. This is quite

Effect	Mean_2021	Mean_2022	Mean_2023	Sd_2021	Sd_2022	Sd_2023
Intercept	-7.30	-7.47	-7.08	0.04	0.05	0.04
TEP	-0.25	-0.40	-0.18	0.04	0.04	0.03
ELI	-0.04	-0.07	-0.05	0.03	0.03	0.03
PGR	0.04	0.13	-0.04	0.04	0.04	0.04
UIS	-0.04	-0.10	0.12	0.03	0.04	0.03
ELL	-0.21	-0.12	-0.15	0.04	0.04	0.04
PDI	-0.05	-0.07	-0.07	0.04	0.05	0.04
ER	-0.25	-0.07	-0.30	0.05	0.05	0.04

intuitive if we think of the material dynamics of reporting gender-based violence: one could reasonably expect violent men to prevent their partners to come out and report the violence suffered.

- `ELI`: The (ventile of the distribution of the) share of employees in low productivity economic units is a clear indicator of (relative) economic underdevelopment. The most naive interpretation wuld be that in underdeveloped areas reporting gender violence is somewhat harder than in developed ones; however this relationship does not appear to be strong and is indeed negligible for 2021 and 2023 data.
- `PGR`: The association with population growth rate is harder to interpret. This association is most likely influenced by several demographic instrumental variables we are not keeping into account and would indeed deserve a more dedicated focus. Only in 2022 does growth rate appear to have a significant association with AVCs accesses.
- `UIS`: The (ventile of the distribution of the) density of production units has a somewhat ambiguous interpretation. From the one side, it has a strong negative relationship with the social frailty index. It should be therefore considered an indicator of economic development. Nevertheless, for 2022 data the regression coefficient bears the same negative sign as the incidence of low-productivity economic units; for 2023 data the association with AVCs accesses is positive instead. For 2021 data, this association appears not significantly different from zero. *Honestly I have no idea on how to interpret it.*
- `ELL`: The association with the proportion of people with low educational level has negative sign and is high in absolute value. The interpretation seems quite easy: cultural development, in general, would encourage reporting violence.
- `PDI`: The association with population dependency index does not seem significantly different from zero
- `ER`: The association with employment rate is very strong and bears negative sign for 2021 and 2023 data.

Spatial regression

We plot the log-residuals ε of the GLM regression models, defined as $\varepsilon := \ln y_{it} - \ln \hat{y}_{it}$ being $\hat{\eta}_{it}$ the fitted value.

Residuals may exhibit spatial structure. To assess it, we employ the Moran and Geary tests. Since

Please notice that log-residuals only take finite values across the municipalities whose female citizens have reported at least one case of violence in 2022.

Additionally, this set of municipalities may include some singletons, which we remove to assess the value of the Moran and Geary statistics. Thus, for each year we have defined the indexes set `nonzero_con` as the set of municipalities from which at least one case of gender-based violence has been reported, *and* which have at least one neighbouring municipality from which at least one case of gender-based violence was reported as well. For brevity, we only show the standardised I values, which under the null hypothesis should be distributed as $N(0,1)$.

```
unlist(spdep::moran.test(
  resids_glm_21[nonzero_con_21],
```

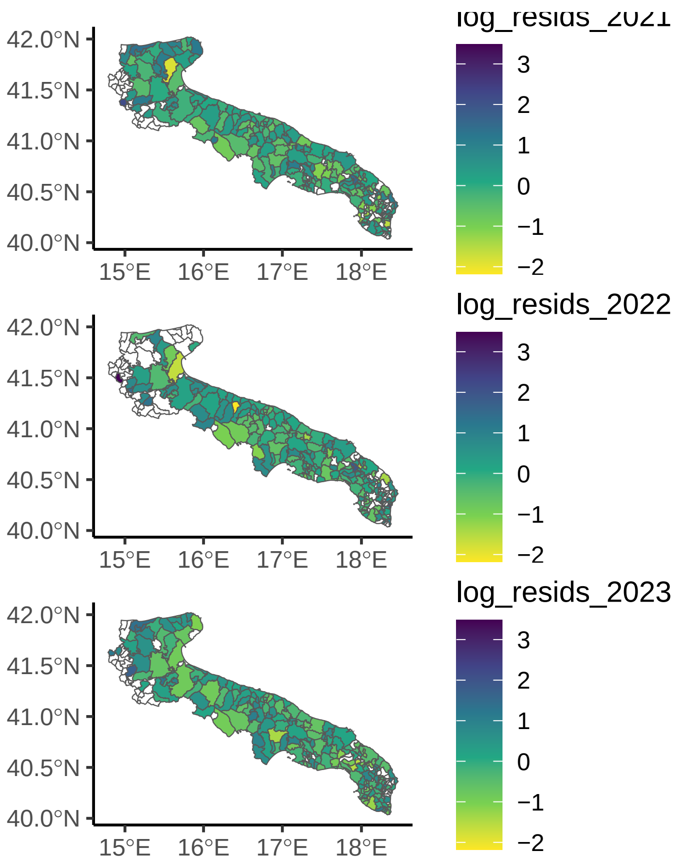


Figure 3: Log-residuals in GLM regression

```r
  listw = spdep::nb2listw(
    nb_con_nonzero_21))[c("statistic", "p.value")])
```

```
## statistic.Moran I statistic standard deviate
##                                   1.73072455
##                                      p.value
##                                   0.04175045
```

```r
unlist(spdep::geary.test(
  resids_glm_21[nonzero_con_21],
  listw = spdep::nb2listw(
    nb_con_nonzero_21))[c("statistic", "p.value")])
```

```
## statistic.Geary C statistic standard deviate
##                                   2.23543917
##                                      p.value
##                                   0.01269427
```

```r
unlist(spdep::moran.test(
  resids_glm_22[nonzero_con_22],
  listw = spdep::nb2listw(
    nb_con_nonzero_22))[c("statistic", "p.value")])
```

```
## statistic.Moran I statistic standard deviate
##                                    2.1943509
##                                      p.value
##                                    0.0141051
```

```r
unlist(spdep::geary.test(
  resids_glm_22[nonzero_con_22],
  listw = spdep::nb2listw(
    nb_con_nonzero_22))[c("statistic", "p.value")])
```

```
## statistic.Geary C statistic standard deviate
##                                  2.644274318
##                                      p.value
##                                  0.004093314
```

```r
unlist(spdep::moran.test(
  resids_glm_23[nonzero_con_23],
  listw = spdep::nb2listw(
    nb_con_nonzero_23))[c("statistic", "p.value")])
```

```
## statistic.Moran I statistic standard deviate
##                                  3.83315e+00
##                                      p.value
##                                  6.32565e-05
```

```r
unlist(spdep::geary.test(
  resids_glm_23[nonzero_con_23],
  listw = spdep::nb2listw(
    nb_con_nonzero_23))[c("statistic", "p.value")])
```

```
## statistic.Geary C statistic standard deviate
##                                  4.190337304
##                                      p.value
##                                  0.000013927
```

In all these three cases, we find evidence for spatial autocorrelation. However, we must stress out this result does not refer to all the regional territory, but only to a subset of all municipalities.

Based on the autocorrelation evidence, though it has only been assessed for a subset of all municipalities, we try implementing some simple spatial models by adding a conditionally autoregressive latent effect, say z, to the linear predictor

$$\eta_{it} = X_{it}^{\top}\alpha + z_{it} \tag{2}$$

We test a total of four models, all of which have a prior distribution depending on the spatial structure of the underlying graph, in this case the Apulia region.

In the following, the area-specific latent field is denoted as $z_i = (z_{i1}^{\top}\, z_{i2}^{\top}\, z_{i3}^{\top})^{\top}$

We describe the spatial structure starting from municipalities neighbourhood, and introduce the neighbourhood matrix W, whose generic element w_{ij} takes value 1 if municipalities i and j are neighbours and 0 otherwise. For each $i \in [1, n]$, $d_i := \sum_{j=1}^{n} w_{ij}$ is the number of neigbours of i-th municipality. Plase notice we have have $n = 256$.

For all models, we define Λ as the precision parameter of the latent effect, and assign it a Wishart prior.

Spatial models are computed by approximating the marginal posteriors of interest via the Integrated Nested Laplace Approximation (INLA), adopting the novel Variational Bayes Approach (Van Niekerk et al. 2023).

Priors for spatial effects have been defined using the `INLAMSM` `R` package (Palmí-Perales, Gómez-Rubio, and Martinez-Beneito 2021).

ICAR model The Intrinsic CAR model is the simplest formulation among spatial autoregressive models. The conditional distribution of each value $z_i \mid z_{-i}$ is:

$$z_i \mid z_{-i} \sim N\left(\sum_{j=1}^{n} \frac{w_{ij}}{d_i} z_j \,,\, \frac{1}{d_i}\Lambda^{-1}\right) \tag{3}$$

Since the joint distribution of z is improper, a sum-to-zero constraint is required for identifiability.

PCAR model The intrinsic autoregressive model is relatively simple to interpret and to implement, while also requiring the minimum number of additional parameter (either the scale or the precision).

The drawback, however, is that we implicitly assume a deterministic spatial autocorrelation coefficient equal to 1. When the autocorrelation is weak, setting an ICAR prior may be a form of misspecification.

A generalisation of this model is the PCAR (proper CAR), which introduces an autocorrelation parameter ρ:

$$z_i \mid z_{-i} \sim N\left(\sum_{j=1}^{n} \rho\frac{w_{ij}}{d_i} z_j \,,\, \frac{}{d_i}\Lambda^{-1}\right) \tag{4}$$

We show the posterior summary for the autocorrelation coefficient.

```
##       mean         sd quant0.025  quant0.25   quant0.5  quant0.75 quant0.975
## 0.92063945 0.04801464 0.79596553 0.89862720 0.93188514 0.95486721 0.97978353
```

The credible interval for ρ is quite pushed towards unity, denoting the model estimates a strong spatial autocorrelation.

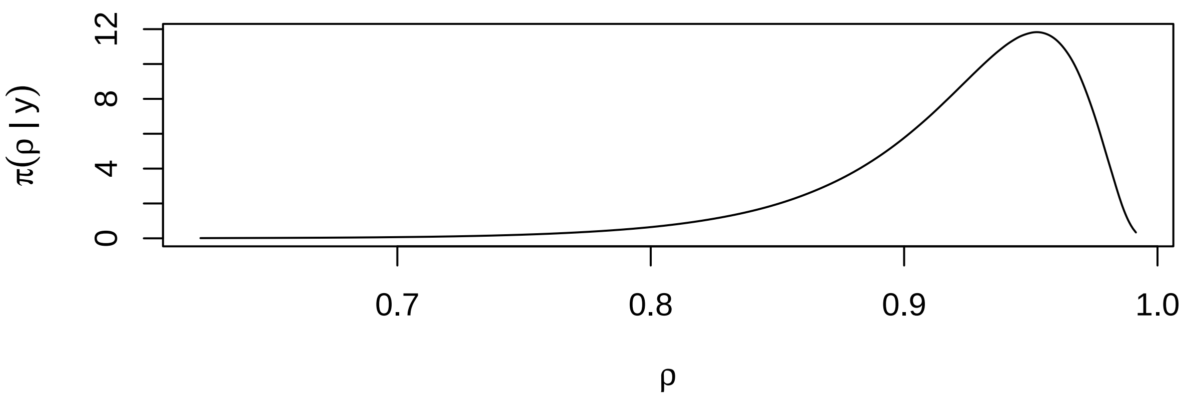


Figure 4: Posterior marginal of the PCAR autocorrelation parameter

Leroux model As an alternative to take into account both structured and unstructured latent effects, we also test the Leroux autoregressive model (Leroux, Lei, and Breslow 2000). In this case, the local prior for z_i is

$$z_i \mid z_{-i} \sim N\left(\sum_{j=1}^{n} \frac{\xi w_{ij}}{1-\xi+\xi d_i} z_j \, , \Lambda^{-1} \frac{1}{1-\xi+\xi d_i}\right) \tag{5}$$

Where $\xi \in [0,1]$ is the mixing parameter. A more interesting representation of the Leroux model is the joint prior

$$z \mid \Lambda, \xi \sim N(0, [\Lambda \otimes (\xi L + (1-\xi)I)]^{-1})$$

where $L := D - W$ is the graph Laplacian matrix, W is the neighbourhood matrix and D is the corresponding degree matrix. We can clearly see how the mixing parameter allocates variability between two precision components, i.e. the Laplacian matrix for the spatial part and the the identity matrix for the noise.

The drawback of this model is the scarce interpretability with respect to more sophisticated ones like the BYM, but the multivariate framework complicates the definition of the BYM model (and it does not seem to be possible to reparametrise it in such a way to have a sparse precision, hence computations are unfeasible) and hampers the use of PC priors, which would have indeed been a useful tool to prevent overfitting.

BYM model Another popular model to control for both spatial autocorrelation and random noise is the Besag, York and Mollié model. We employ a simplified formulation, with a unique mixing parameter for all three years. Under this model the latent effect is defined as:

$$z = \sqrt{\phi} u M + \sqrt{1-\phi} v M \tag{6}$$

Where: - $u \sim N(0, I_p \otimes L_{\text{scaled}})$ is an independent multivariate ICAR process whose precision matrix is scaled in order that the geometric mean of marginal variances (i.e. the diagonal entries of R^+) is one - $v \sim N(0, I_p \otimes I_n)$ is a Standard Normal variable - M is a positive definite matrix such that $M'M = \Lambda^{-1}$. It is not necessarily the Cholesky factor of the scale parameter; in fact, a convenient but not unique way to define it may be $M = D^{-\frac{1}{2}} E^{\top}$ where E and D are the eigenvector and eigenvalues matrices of Λ (Urdangarin et al. 2024).

- $\phi \in [0, 1]$ is the mixing parameter. We assign a Uniform prior on ϕ (but the PC-prior would be a more rigorous choice) and the usual Wishart prior to Λ.

The BYM model intruduces a mixing parameter as the LCAR does, but improves its interpretability by allowing to scale the ICAR and IID components.

We use a rather primitive BYM definition here. It could be improved in many directions: either defining a PC prior on the mixing parameter and employing a more elegant parametrisation allowing for sparse precision (Riebler et al. 2016), or relaxing the naive hypothesis of a unique mixing parameter implementing an M-model.

However, even this rudimentary formulation allows to interpret the mixing parameter more clearly.

```
## Time needed for scaling Laplacian matrix:  2.577  seconds
```

Spatiotemporal model A different perspective than multivariate modelling would be spatiotemporal modelling.

In this case, we would assume the DGP of the linear predictor to be of this form:

$$\eta_{it} = X_{it}^{\top}\alpha_t + z_i + \delta_{it} \tag{7}$$

For brevity, we immediately assume z is a BYM process, hence $z = \sigma\left(\sqrt{(\phi)}u + \sqrt{(1-\phi)}v\right)$, where u is an ICAR field with typical variance $= 1$ and v is a Standard iid Gaussian field.

The term δ_{it} controls for spatio-temporal interaction. To our aims, two spatiotemporal interactions are relevant: - Type I interaction, i.e. interaction between the temporal effect (assumed IID) and v - Type III interaction, i.e. interaction between the temporal effect (assumed IID) and u.

Since the purely temporal effect is negligible we do not bother about defining priors on it different than the default.

We start fitting the type-I model. Type-III apparently has some misspecifications needing to be fixed.

I don't know if this is correct, but a PC-prior is set on δ_{it}, though its definition relies on the assumption that z is fixed.

```
cav_STbym_i1_INLA <- inla(N_ACC ~ 1 +TEP_th + ELI + PGR + UIS + ELL + PDI + ER +
                            f(Area, model = "bym2", graph = W_con,  scale.model = T,
                              hyper = list(prec = list(prior = "pc.prec", param = c(1.5, 0.01))))+
                            f(Year, model = "iid") +
                            f(ID, model = "iid",
                              hyper = list(prec = list(prior = "pc.prec", param = c(1.5, 0.01)))),
                            family = c("poisson", "poisson", "poisson"), offset = log(nn), data =dd_lis
                            num.threads = 1, control.compute =
                              list(internal.opt = F, cpo = T, waic = T),
                            #inla.mode = "classic", control.inla = list(strategy = "laplace", int.strat
                            control.predictor = list(compute = T),
                            verbose = T)
```

Model assessment We briefly compare the three models in scope through the WAIC (Gelman, Hwang, and Vehtari 2014):

% latex table generated in R 4.4.1 by xtable 1.8-4 package % Fri Mar 14 11:31:24 2025

As we can see, adding a spatial model is an improving element, not a waste of complexity.

Model	WAIC	Eff_params
Null	3449.12	76.90
ICAR	2915.03	192.97
PCAR	2915.41	201.24
Leroux	2911.35	200.46
BYM	2889.66	194.48
ST_I	2925.40	195.65

Preliminary findings

Here we show some posterior summaries for α under the BYM model.

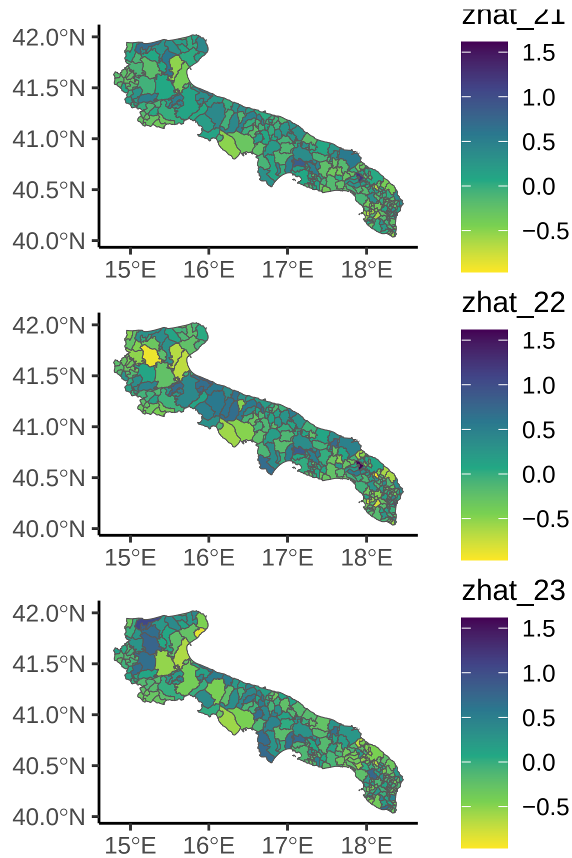
% latex table generated in R 4.4.1 by xtable 1.8-4 package % Fri Mar 14 11:31:25 2025

Effect	Mean_2021	Mean_2022	Mean_2023	Sd_2021	Sd_2022	Sd_2023
Intercept	-7.41	-7.63	-7.20	0.05	0.06	0.05
TEP	-0.23	-0.39	-0.18	0.06	0.08	0.06
ELI	-0.00	-0.03	-0.05	0.05	0.06	0.05
PGR	0.08	0.14	0.03	0.06	0.07	0.06
UIS	-0.01	-0.10	0.10	0.05	0.07	0.06
ELL	-0.27	-0.24	-0.21	0.07	0.08	0.07
PDI	-0.07	-0.06	-0.08	0.06	0.08	0.07
ER	-0.29	-0.14	-0.33	0.08	0.09	0.08

Estimations of α differ slightly from the nonspatial model. For all variables, credibility intervals are wider due to increased uncertainty.

- `TEP_th_22` The effect of the distance from the closest support center remains similar in mean and the interpretation is not altered.
- `ELI`: The effect of the incidence of low-productivity economic units is utterly negligible
- `PGR`: The association with population growth rate can only be considered barely significant for 2022 data
- `UIS`: The association with the density of productive units is negligible for 2021 and 2022 data, and can be considered slightly significant for 2023, bearing positive sign.
- `ELL`: The association with the incidence of low education levels, is even higher in mean than under the nonspatial model. We interpret this result as a strong *potential* impact of education on the chance that gender violence is reported
- `PDI`: The effect of structural dependency index is utterly negligible, as for the GLM.
- `ER`: The effect associated with employment rate is increased for 2021 data, more than doubled for 2022 data, and slightly increased for 2023 data. How to interpret this finding? Employment rate is clearly an indicator of economic development, hence the easiest interpretation is that `-` as it was with `ELI` under the nonspatial model `-` in more developed areas there is a higher chance that gender violence is reported.

We show the expected latent effects



As we can see, the behaviour of 2023 data is quite puzzling, since high values of the latent effect are observed in areas such as the Northern province of Foggia, which was previously characterised by low accesses.

We additionally show the fitted values of the counts of accesses, rounded to the closest integer.

We show the posterior median of the marginal variances of z, i.e. Λ^{-1}:

For more insight on the dependence structure between the three years, posterior summaries for correlations are shown in the following. In terms of point estimates, the only high correlation appears that of 2021-2022. % latex table generated in R 4.4.1 by xtable 1.8-4 package % Fri Mar 14 11:31:28 2025

Years	Mean	Sd	Q0.025	Median	Q0.975
2021-22	0.712	0.090	0.507	0.723	0.856
2021-23	0.296	0.133	0.020	0.302	0.538
2022-23	0.211	0.131	-0.055	0.214	0.457

Lastly, we have a look at the mixing parameter under both the Leroux and BYM models.

```
##      mean        sd quant0.025  quant0.25   quant0.5  quant0.75 quant0.975
## 0.6956351 0.1249525  0.4187255  0.6151433  0.7096032  0.7899070  0.8956358
```

As we see, the Leroux mixing parameter is not particularly high, but interpreting it properly is hampered by the difficulty in scaling the precision matrix, which is a drawback of non-intrinsic models.

```
##       mean         sd quant0.025  quant0.25   quant0.5  quant0.75 quant0.975
## 0.84441378 0.08847709 0.62287832 0.79804603 0.86288051 0.90992533 0.96157778
```

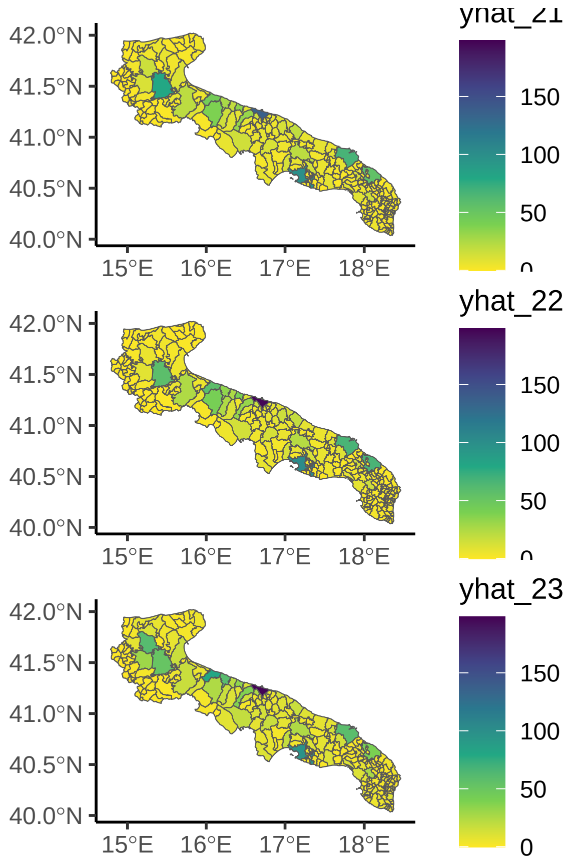


Figure 5: Fitted values using the Leroux model

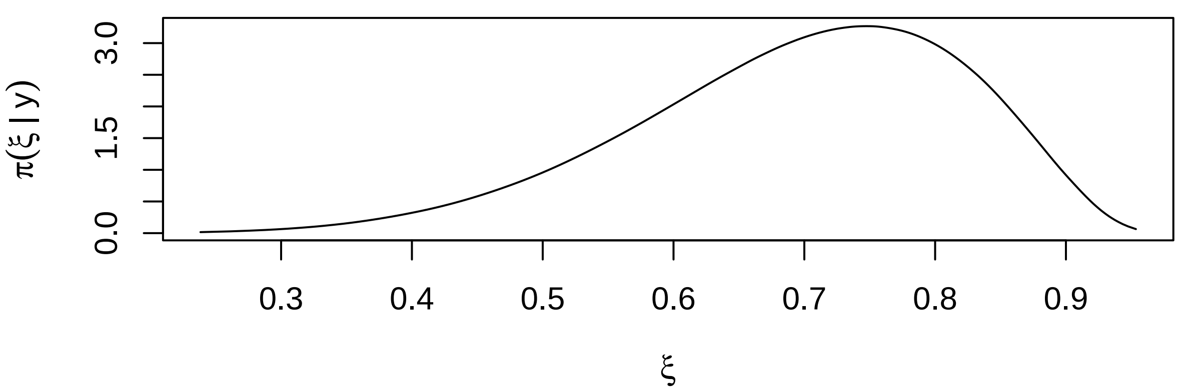


Figure 6: Posterior marginal of the LCAR mixing parameter

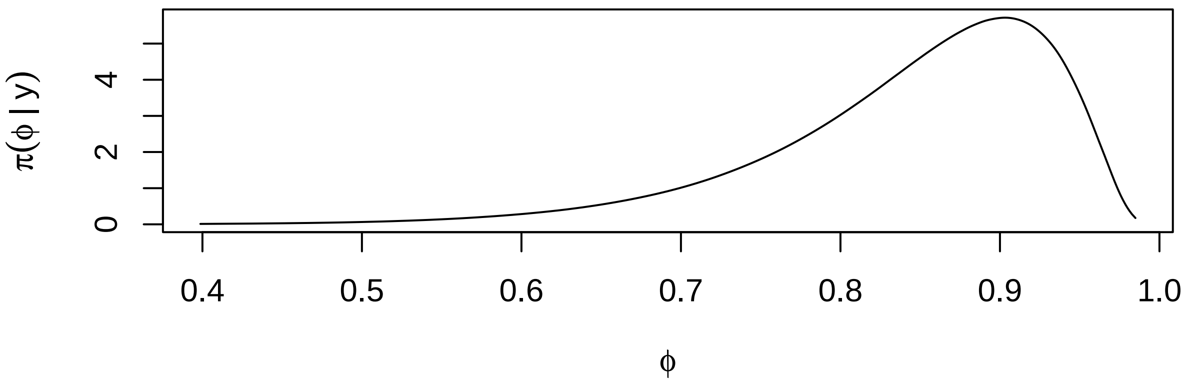


Figure 7: Posterior marginal of the BYM mixing parameter

Further development: pogit model for underreporting

The observed counts of accesses to AVCs clearly imply a severe under-reporting of gender-based violence, regarding the vast majority of incidents.

If this does not provide a valid reason to model the under-reporting process explicitly - as there are not any data to assess the spatial distribution of true VAW incindents counts at *this* level of granularity, it is still needful to consider that some risk factors with a strong negative association have same-sign effects on accesses counts.

More specifically, employment rate and low-education incidence have a very high negative correlation, as it is reasonable to expect. Notwithstanding this, both appear to have a negative impact on access counts. A possible explanation for this finding is that these two variables impact on two different processes, which we do not observe, but whose interaction yields to the observed response variable, namely accesses counts.

We therefore attempt at implementing a Poisson - logistic model to explicitly account for under-reporting (Stoner, Economou, and Silva 2019). Given the scarce information at our disposal, we can only guess which variable enter either of the two predictor components, i.e. the VAW reporting and the actual occurrence Specifically, we assign distance from closest AVC and low education incidence to the former component and all other variables plus the spatial effect to the latter.

The model is thus defined:

$$y_{i,t} \sim \text{Poisson}(E_{it}\, \lambda_{it}\, \pi_{it}) \quad \forall i \in [1, 256], t \in \{2021, 2022, 2023\} \tag{8}$$

Where $E_{i,t}$ is the offset, λ_{it} is the expected count of incidents π_{it} is the expected reporting rate. We will denote α as the $k+1$ covariate effects entering λ_{it} and β the $m+1$ covariate effects entering π_{it} for two integers p and m.

We would thus have, for each year t and municipality i;

$$\lambda_{it} = e^{\alpha_{0t} + \alpha_{1t} X_{i1,t} + \ldots + \alpha_{pt} X_{ik,t} + z_{it}}$$

And

$$\pi_{it} = \left[1 + e^{-(\beta_{0t} + \beta_{1t} X_{k+1,t} + \ldots + \beta_{mt} X_{i(k+m),t})}\right]^{-1}$$

Hence the predictor is not linear anymore.

We approximate the predictor through first-order Taylor expansion, which is made possible by the `inlabru` `R` package (Lindgren et al. 2024). To define the nonlinear predictor component, we follow (Wøllo 2022).

Here, we show the `R` code to achieve this. For computational reasons, we only employ the BYM model with a Uniform prior on the mixing parameter. The R function shown here can also be used for spatiotemporal models.

```
logexpit <- function(x, beta){
  pred <- beta[[1]] +
    rowSums(do.call(cbind, lapply(c(1:length(x)), function(n){
      beta[[n+1]]*x[[n]]
      })))
  return(-log(1+exp(-pred)))
}

cmp_spatial <- function(spatial_expr, st_expr = NULL, t_expr = NULL) {

  spatial <- function(id, ...) INLA::f(id, ...)
  spatiotemporal <- function(id, ...) INLA::f(id, ...)
  temporal <- function(id, ...) INLA::f(id, ...)

  f2 <- substitute(spatial_expr)
  f3 <- substitute(t_expr)
  f4 <- substitute(st_expr)
  f1 <- bquote(
    ~ 0 +
      alpha_0_2021(Intercept_2021) +
      alpha_0_2022(Intercept_2022) +
      alpha_0_2023(Intercept_2023) +
      beta_0_2021(main = Intercept_2021,  model = "linear",
                  mean.linear = -2.2,
                  prec.linear = 1e+2) +
      beta_0_2022(main = Intercept_2022,  model = "linear",
                  mean.linear = -2.2,
                  prec.linear = 1e+2) +
      beta_0_2023(main = Intercept_2023,  model = "linear",
                  mean.linear = -2.2,
                  prec.linear = 1e+2) +
      myoffset(log(nn), model = "offset") +
      alpha_ELI_2021(ELI_2021) + alpha_ELI_2022(ELI_2022) + alpha_ELI_2023(ELI_2023) +
      alpha_PGR_2021(PGR_2021) + alpha_PGR_2022(PGR_2022) + alpha_PGR_2023(PGR_2023) +
      alpha_UIS_2021(UIS_2021) + alpha_UIS_2022(UIS_2022) + alpha_UIS_2023(UIS_2023) +
      alpha_PDI_2021(PDI_2021) + alpha_PDI_2022(PDI_2022) + alpha_PDI_2023(PDI_2023) +
      alpha_ER_2021(ER_2021) + alpha_ER_2022(ER_2022) + alpha_ER_2023(ER_2023) +
      beta_TEP_2021(main = Intercept_2021, model = "linear",
                    mean.linear = 0,
                    prec.linear = 1e-3) +
      beta_TEP_2022(main = Intercept_2022, model = "linear",
                    mean.linear = 0,
                    prec.linear = 1e-3)  +
      beta_TEP_2023(main = Intercept_2023, model = "linear",
                    mean.linear = 0,
                    prec.linear = 1e-3) +
```

```
      beta_ELL_2021(main = Intercept_2021, model = "linear",
                    mean.linear = 0,
                    prec.linear = 1e-3)  +
      beta_ELL_2022(main = Intercept_2022, model = "linear",
                    mean.linear = 0,
                    prec.linear = 1e-3)  +
      beta_ELL_2023(main = Intercept_2023, model = "linear",
                    mean.linear = 0,
                    prec.linear = 1e-3) )
  ff <- f1[[2]]
  if (!is.null(f2)) ff <- as.call(c(quote(`+`), ff, f2))
  if (!is.null(f3)) ff <- as.call(c(quote(`+`), ff, f3))
  if (!is.null(f4)) ff <- as.call(c(quote(`+`), ff, f4))
  final_formula <- as.formula(bquote(~ .(ff)))
  return(final_formula)
}

cmp_BYM <- cmp_spatial(spatial(
  ID, model = inla.MBYM.dense(k = 3, W = W_con,
                              PC = FALSE),
  extraconstr = list(A = A_constr, e = rep(0, 3))))

library(inlabru)
```

```
## Caricamento del pacchetto richiesto: fmesher
```

```
##
## Caricamento pacchetto: 'inlabru'
```

```
## Il seguente oggetto è mascherato da 'package:MASS':
##
##     shrimp
```

```
cav_bru <- function(model_cmp, data = dd_extended){

  terms <- c("0", "alpha_0_2021", "alpha_0_2022", "alpha_0_2023",
             "myoffset",
             "alpha_ELI_2021", "alpha_ELI_2022", "alpha_ELI_2023",
             "alpha_PGR_2021", "alpha_PGR_2022", "alpha_PGR_2023",
             "alpha_UIS_2021", "alpha_UIS_2022", "alpha_UIS_2023",
             "alpha_PDI_2021", "alpha_PDI_2022", "alpha_PDI_2023",
             "alpha_ER_2021", "alpha_ER_2022", "alpha_ER_2023",
             paste0("logexpit(x = list(TEP_th_2021, ELL_2021),",
                    "beta = list(beta_0_2021, beta_TEP_2021, beta_ELL_2021))"),
             paste0("logexpit(x = list(TEP_th_2022, ELL_2022),",
                    "beta = list(beta_0_2022, beta_TEP_2022, beta_ELL_2022))"),
             paste0("logexpit(x = list(TEP_th_2023, ELL_2023),",
                    "beta = list(beta_0_2023, beta_TEP_2023, beta_ELL_2023))"))

  if (any(grepl("spatial", as.character(model_cmp)))) terms <- c(terms, "spatial")
  if (any(grepl("time", as.character(model_cmp)))) terms <- c(terms, "time")
  if (any(grepl("spatiotemporal", as.character(model_cmp)))) terms <- c(terms, "spatiotemporal")

  formula <- as.formula(paste("N_ACC ~", paste(terms, collapse = " + ")))
```

```r
  res <- inlabru::bru(
    components = model_cmp,
    lik = inlabru::like(family = "poisson", formula = formula, data = data),
    options = list(verbose = T, num.threads = 1,
                   control.compute = list(
                     waic = T, cpo = T, dic = T, internal.opt = F),
                   control.predictor = list(compute =  T)))
  return(res)
}
```

```r
cav_MBYM_inlabru <- cav_bru(cmp_BYM)
```

```
## Time needed for scaling Laplacian matrix:  1.945  seconds
```

Here we show some posterior summaries for covariate effects. % latex table generated in R 4.4.1 by xtable 1.8-4 package % Fri Mar 14 11:34:43 2025

Effect	Mean_2021	Mean_2022	Mean_2023	Sd_2021	Sd_2022	Sd_2023
alpha_0	-3.70	-3.93	-3.50	0.10	0.11	0.10
beta_0	-2.20	-2.20	-2.20	0.10	0.10	0.10
alpha_ELI	-0.01	-0.03	-0.05	0.05	0.06	0.05
alpha_PGR	0.08	0.14	0.03	0.06	0.07	0.06
alpha_UIS	-0.01	-0.10	0.10	0.05	0.07	0.06
alpha_PDI	-0.07	-0.06	-0.08	0.06	0.08	0.07
alpha_ER	-0.29	-0.14	-0.33	0.08	0.09	0.08
beta_TEP	-0.25	-0.44	-0.20	0.07	0.09	0.07
beta_ELL	-0.29	-0.27	-0.23	0.08	0.09	0.08

Appendices

R code to fit the LCAR The `R` code for the multivariate Leroux CAR model is shown. It is derived from the code for the PCAR in the texttt{R} package `INLAMSM` (Palmí-Perales, Gómez-Rubio, and Martinez-Beneito 2021).

A Gaussian prior is set on the logit of the mixing parameter for analogy with the default prior used in the univariate version (model `"besagproper"` in `R-INLA`)

```r
inla.rgeneric.MLCAR.model <-
  function (cmd = c("graph", "Q", "mu", "initial", "log.norm.const",
                    "log.prior", "quit"), theta = NULL)
  {
    interpret.theta <- function() {
      alpha <-  1/(1 + exp(-theta[1L]))
      mprec <- sapply(theta[as.integer(2:(k + 1))], function(x) {
        exp(x)
      })
      corre <- sapply(theta[as.integer(-(1:(k + 1)))], function(x) {
        (2 * exp(x))/(1 + exp(x)) - 1
      })
      param <- c(alpha, mprec, corre)
      n <- (k - 1) * k/2
      M <- diag(1, k)
      M[lower.tri(M)] <- param[k + 2:(n + 1)]
      M[upper.tri(M)] <- t(M)[upper.tri(M)]
      st.dev <- 1/sqrt(param[2:(k + 1)])
```

```r
    st.dev.mat <- matrix(st.dev, ncol = 1) %*% matrix(st.dev,
                                                      nrow = 1)
    M <- M * st.dev.mat
    PREC <- solve(M)
    return(list(alpha = alpha, param = param, VACOV = M,
                PREC = PREC))
  }
  graph <- function() {
    PREC <- matrix(1, ncol = k, nrow = k)
    G <- kronecker(PREC, Matrix::Diagonal(nrow(W), 1) +
                     W)
    return(G)
  }
  Q <- function() {
    param <- interpret.theta()
    Lapl <- Matrix::Diagonal(nrow(W), apply(W, 1, sum)) -  W
    #Sigma.u <- MASS::ginv(as.matrix(Lapl))
    #Sigma <- param$alpha * Sigma.u + (1-param$alpha)*diag(1, nrow(W))
    R <- param$alpha*Lapl + (1-param$alpha)*diag(1, nrow(W))
    Q <- kronecker(param$PREC, R)
    return(Q)
  }
  mu <- function() {
    return(numeric(0))
  }
  log.norm.const <- function() {
    val <- numeric(0)
    return(val)
  }
  log.prior <- function() {
    param <- interpret.theta()
    # Uniform prior on \lambda
    # val <- -theta[1L] - 2 * log(1 + exp(-theta[1L]))
    # Normal prior on logit lambda, in analogy with the univariate default case
    val <- log(dnorm(theta[1L], mean = 0, sd = sqrt(1/0.45)))

    val <- val + log(MCMCpack::dwish(
      W = param$PREC, v = k, S = diag(rep(1, k)))) +
      sum(theta[as.integer(2:(k + 1))]) +
      sum(log(2) + theta[-as.integer(1:(k + 1))] -
            2 * log(1 + exp(theta[-as.integer(1:(k + 1))])))
    return(val)
  }
  initial <- function() {
    return(c(0, rep(log(1), k), rep(0, (k * (k - 1)/2))))
  }
  quit <- function() {
    return(invisible())
  }
  if (as.integer(R.version$major) > 3) {
    if (!length(theta))
      theta = initial()
  }
```

```
    else {
      if (is.null(theta)) {
        theta <- initial()
      }
    }
    val <- do.call(match.arg(cmd), args = list())
    return(val)
  }

inla.MLCAR.model <- function(...) INLA::inla.rgeneric.define(inla.rgeneric.MLCAR.model, ...)
```

R code for the BYM A possible improvement on the modelistic side would be to give the multivariate BYM model a parametrisation allowing for a sparse precision matrix(Riebler et al. 2016)

In the univariate framework, the model can be reparametrised as a joint model of the form (z, u), such that $\text{Prec}[(\text{vec}(z)^\top, \text{vec}(u)^\top)^\top]$ is indeed sparse (Riebler et al. 2016). In the multivariate framework, this idea can be easily applied to the joint field (z, uM).

Therefore, for the time being we rely on the classic, dense parametrisation, which is however rather inefficient in computational terms.

Here we only show the R code to fit this model. The uniform prior of ϕ is another rather crude choice and more sophisticated priors ought to be implemented as well.

```
inla.rgeneric.MBYM.dense <-
  function (cmd = c("graph", "Q", "mu", "initial", "log.norm.const",
                    "log.prior", "quit"), theta = NULL){
    envir <- parent.env(environment())
    if(!exists("cache.done", envir=envir)){
      starttime.scale <- Sys.time()
      #' Unscaled Laplacian matrix (marginal precision of u_1, u_2 ... u_k)
      L_unscaled <- Matrix::Diagonal(nrow(W), rowSums(W)) -  W
      L_unscaled_block <- kronecker(diag(1,k), L_unscaled)
      A_constr <- t(pracma::nullspace(as.matrix(L_unscaled_block)))
      scaleQ <- INLA:::inla.scale.model.internal(
        L_unscaled_block, constr = list(A = A_constr, e = rep(0, nrow(A_constr))))
      #' Block Laplacian, i.e. precision of U = I_k \otimes L
      n <- nrow(W)
      L <- scaleQ$Q[c(1:n), c(1:n)]
      Sigma.u <- MASS::ginv(as.matrix(L))
      if(PC == TRUE){
        #' Eigenvalues of the SCALED Laplacian, sufficient for trace and determinant entering the KLD
        L_eigen_scaled <- eigen(scaleQ$Q)$values
        #' PC prior on mixing parameter - definition should be fine.
        log.dpc.phi.bym <- INLA:::inla.pc.bym.phi(eigenvalues = L_eigen_scaled,
                                                  marginal.variances = scaleQ$var,
                                                  rankdef = nrow(A_constr),
                                                  u = 0.5, alpha = 2/3)
        assign("log.dpc.phi.bym", log.dpc.phi.bym, envir = envir)

      }
      endtime.scale <- Sys.time()
      cat("Time needed for scaling Laplacian matrix: ",
          round(difftime(endtime.scale, starttime.scale), 3), " seconds \n")
      assign("L", L, envir = envir)
```

```r
    assign("Sigma.u", Sigma.u, envir = envir)
    assign("cache.done", TRUE, envir = envir)
  }
  interpret.theta <- function() {
    alpha <-  1/(1 + exp(-theta[1L]))
    mprec <- sapply(theta[as.integer(2:(k + 1))], function(x) {
      exp(x)
    })
    corre <- sapply(theta[as.integer(-(1:(k + 1)))], function(x) {
      (2 * exp(x))/(1 + exp(x)) - 1
    })
    param <- c(alpha, mprec, corre)
    n <- (k - 1) * k/2
    M <- diag(1, k)
    M[lower.tri(M)] <- param[k + 2:(n + 1)]
    M[upper.tri(M)] <- t(M)[upper.tri(M)]
    st.dev <- 1/sqrt(param[2:(k + 1)])
    st.dev.mat <- matrix(st.dev, ncol = 1) %*% matrix(st.dev,
                                                       nrow = 1)
    M <- M * st.dev.mat
    PREC <- solve(M)
    return(list(alpha = alpha, param = param, VACOV = M,
                PREC = PREC))
  }
  graph <- function() {
    PREC <- matrix(1, ncol = k, nrow = k)
    G <- kronecker(PREC, Matrix::Diagonal(nrow(W), 1) +
                     W)
    return(G)
  }
  Q <- function() {
    param <- interpret.theta()
    #' Weighted average of ICAR and IID variables: variance is the sum of variances.
    #' Precision here defined as inverse variance. Not
    #' the best way to do it; still using sparse
    #' parametrisation requires a latent effect of length 2*np
    Sigma <- param$alpha * Sigma.u + (1-param$alpha)*diag(1, nrow(W))
    Q <- kronecker(param$PREC, solve(Sigma))
    return(Q)
  }
  mu <- function() {
    return(numeric(0))
  }
  log.norm.const <- function() {
    val <- numeric(0)
    return(val)
  }
  log.prior <- function() {
    param <- interpret.theta()
    if(PC == TRUE){
      #' PC prior implementation
      val <- log.dpc.phi.bym(param$phi)- theta[1L] - 2 * log(1 + exp(-theta[1L]))
    } else {
```

```r
        #' Uniform prior
        val <- -theta[1L] - 2 * log(1 + exp(-theta[1L]))
      }
      #' Whishart prior on precision (inverse scale)
      val <- val + log(MCMCpack::dwish(W = param$PREC, v = k,
                                        S = diag(rep(1, k)))) +
        #' This for the change of variable
        #' (code from INLAMSM)
        sum(theta[as.integer(2:(k +  1))]) +
        sum(log(2) + theta[-as.integer(1:(k + 1))] - 2 * log(1 + exp(theta[-as.integer(1:(k + 1))])))
      return(val)
    }
    initial <- function() {
      if(!exists("init", envir = envir)){
        return(c(0, rep(0, k), rep(0, (k * (k - 1)/2))))
      } else{
        return(init)
      }
    }
    quit <- function() {
      return(invisible())
    }
    if (as.integer(R.version$major) > 3) {
      if (!length(theta))
        theta = initial()
    }
    else {
      if (is.null(theta)) {
        theta <- initial()
      }
    }
    val <- do.call(match.arg(cmd), args = list())
    return(val)
  }
```

Here we provide a preliminary code for the sparse parameterisation, still it needs more analysis and validation.

```r
inla.rgeneric.MBYM.sparse <-
  function (cmd = c("graph", "Q", "mu", "initial", "log.norm.const",
                    "log.prior", "quit"), theta = NULL) {
    envir <- parent.env(environment())
    if(!exists("cache.done", envir=envir)){
      starttime.scale <- Sys.time()
      #' Unscaled Laplacian matrix (marginal precision of u_1, u_2 ... u_k)
      L_unscaled <- Matrix::Diagonal(nrow(W), rowSums(W)) -  W
      L_unscaled_block <- kronecker(diag(1,k), L_unscaled)
      A_constr <- t(pracma::nullspace(as.matrix(L_unscaled_block)))
      scaleQ <- INLA:::inla.scale.model.internal(
        L_unscaled_block, constr = list(A = A_constr, e = rep(0, nrow(A_constr))))
      #' Block Laplacian, i.e. precision of U = I_k \otimes L
      n <- nrow(W)
      L <- scaleQ$Q[c(1:n), c(1:n)]
      if(PC == TRUE){
        #' Eigenvalues of the SCALED Laplacian, sufficient for trace and determinant entering the KLD
```

```r
    L_eigen_scaled <- eigen(scaleQ$Q)$values
    #' PC prior on mixing parameter - definition should be fine.
    log.dpc.phi.bym <- INLA:::inla.pc.bym.phi(eigenvalues = L_eigen_scaled,
                                              marginal.variances = scaleQ$var,
                                              rankdef = nrow(A_constr),
                                              u = 0.5, alpha = 2/3)
    assign("log.dpc.phi.bym", log.dpc.phi.bym, envir = envir)

  }
  endtime.scale <- Sys.time()
  cat("Time needed for scaling Laplacian matrix: ",
      round(difftime(endtime.scale, starttime.scale), 3), " seconds \n")
  assign("L", L, envir = envir)
  assign("cache.done", TRUE, envir = envir)
}
interpret.theta <- function() {
  phi <-  1/(1 + exp(-theta[1L]))
  mprec <- sapply(theta[as.integer(2:(k + 1))], function(x) {
    exp(x)
  })
  corre <- sapply(theta[as.integer(-(1:(k + 1)))], function(x) {
    (2 * exp(x))/(1 + exp(x)) - 1
  })
  param <- c(phi, mprec, corre)
  n <- (k - 1) * k/2
  M <- diag(1, k)
  M[lower.tri(M)] <- param[k + 2:(n + 1)]
  M[upper.tri(M)] <- t(M)[upper.tri(M)]
  st.dev <- 1/sqrt(param[2:(k + 1)])
  st.dev.mat <- matrix(st.dev, ncol = 1) %*% matrix(st.dev,
                                                    nrow = 1)
  M <- M * st.dev.mat
  PREC <- solve(M)
  return(list(phi = phi, param = param, VACOV = M,
              PREC = PREC))
}
graph <- function() {
  BPrec <- matrix(1, ncol = 2*k, nrow = 2*k)
  G <- kronecker(BPrec, Matrix::Diagonal(nrow(W), 1) +
                   W)
  return(G)
}
Q <- function() {
  param <- interpret.theta()
  Q11 <- 1/(1 - param$phi) * kronecker(param$PREC, Matrix::Diagonal(n = nrow(W), x  = 1))
  Q12 <- Q21 <- -sqrt(param$phi)/(1 - param$phi) * kronecker(param$PREC, Matrix::Diagonal(n = nrow(
  Q22 <- kronecker(param$PREC,
                   ((param$phi/(1-param$phi))* Matrix::Diagonal(n = nrow(W), x = 1) + L))
  Q <- rbind(cbind(Q11, Q12), cbind(Q21, Q22))
  return(Q)
}
mu <- function() {
  return(numeric(0))
```

```r
  }
  log.norm.const <- function() {
    val <- numeric(0)
    return(val)
  }

  log.prior <- function() {
    param <- interpret.theta()
    if(PC == TRUE){
      #' PC prior implementation
      val <-  log.dpc.phi.bym(param$phi) - theta[1L] - 2 * log(1 + exp(-theta[1L]))
    }else {
      #' Uniform prior
      val <- -theta[1L] - 2 * log(1 + exp(-theta[1L]))
    }
    #' Whishart prior on precision (inverse scale)
    val <- val + log(MCMCpack::dwish(W = param$PREC, v = k,
                                     S = diag(rep(1, k)))) +
      #' This for the change of variable
      #' (code from INLAMSM)
      sum(theta[as.integer(2:(k +  1))]) +
      sum(log(2) + theta[-as.integer(1:(k + 1))] - 2 * log(1 + exp(theta[-as.integer(1:(k + 1))])))
    return(val)
  }
  initial <- function() {
    if(!exists("init", envir = envir)){
      return(c(0, rep(0, k), rep(0, (k * (k - 1)/2))))
    } else{
      return(init)
    }
  }
  quit <- function() {
    return(invisible())
  }
  if (as.integer(R.version$major) > 3) {
    if (!length(theta))
      theta <- initial()
  }
  else {
    if (is.null(theta)) {
      theta <- initial()
    }
  }
  val <- do.call(match.arg(cmd), args = list())
  return(val)
}
```

Addressing the potential issue of spatial confounding The changes in $E[\alpha|y]$ may suggest we are in presence of confounding bias. We adopt the most parsimonious approach to overcome this situation, i.e. removing spatial trends up to a certain order from all explanatory variables except for the distance from the closest AVC. In this case, interpreting a distance indicator from which spatial structure is removed becomes challenging. For all other covariates we follow the methodology of (Urdangarin et al. 2024).

The fundamental decision regards the number of spatial trends to remove from each covariate. We try at removing as few spatial trends as possible in order to obtain a value of the Moran's standardised index less than 1.645, which corresponds to the 95-th percentile of the standard Normal distribution.

To do so, we remove the last 3 eigenvectors from `ELI` covariate, 13 eigenvectors from `PGR`, 19 from `UIS`, 19 from `ELL`, 6 from `PDI` and 31 from `ER`.

Then we fit the BYM model to the deconfounded data

```
## Time needed for scaling Laplacian matrix:  1.566  seconds
```

Resulting estimates for covariate effects are summarised: % latex table generated in R 4.4.1 by xtable 1.8-4 package % Fri Mar 14 11:35:41 2025

Effect	Mean_2021	Mean_2022	Mean_2023	Sd_2021	Sd_2022	Sd_2023
Intercept	-7.42	-7.62	-7.20	0.05	0.06	0.05
TEP	-0.18	-0.40	-0.13	0.06	0.07	0.06
ELI	0.02	0.01	-0.01	0.05	0.06	0.05
PGR	0.05	0.12	0.03	0.06	0.07	0.06
UIS	0.05	-0.02	0.11	0.05	0.06	0.05
ELL	-0.22	-0.21	-0.19	0.06	0.07	0.06
PDI	-0.02	0.03	-0.04	0.06	0.07	0.06
ER	-0.19	-0.18	-0.21	0.06	0.07	0.06

The most notable change is perhaps the shrinkage in the effect of employment rate in 2021 and 2023. This being said, the interpretation of the posteriors of α is overall consistent with the base MLCAR model.

Gelman, Andrew, Jessica Hwang, and Aki Vehtari. 2014. "Understanding Predictive Information Criteria for Bayesian Models." *Statistics and Computing* 24 (6): 997–1016. https://doi.org/10.1007/S11222-013-9416-2.

Leroux, Brian G., Xingye Lei, and Norman Breslow. 2000. "Estimation of Disease Rates in Small Areas: A New Mixed Model for Spatial Dependence." In *Statistical Models in Epidemiology, the Environment, and Clinical Trials*, edited by M. Elizabeth Halloran and Donald Berry, 179–91. New York, NY: Springer New York. https://doi.org/https://doi.org/10.1007/978-1-4612-1284-3_4.

Lindgren, Finn, Fabian Bachl, Janine Illian, Man Ho Suen, Håvard Rue, and Andrew E Seaton. 2024. "Inlabru: Software for Fitting Latent Gaussian Models with Non-Linear Predictors." *arXiv Preprint arXiv:2407.00791*. https://doi.org/https://doi.org/10.48550/arXiv.2407.00791.

Palmí-Perales, Francisco, Virgilio Gómez-Rubio, and Miguel A. Martinez-Beneito. 2021. "Bayesian Multivariate Spatial Models for Lattice Data with INLA." *Journal of Statistical Software* 98 (2): 1–29. https://doi.org/10.18637/jss.v098.i02.

Riebler, Andrea, Sigrunn H Sørbye, Daniel Simpson, and Håvard Rue. 2016. "An Intuitive Bayesian Spatial Model for Disease Mapping That Accounts for Scaling." *Statistical Methods in Medical Research* 25 (4): 1145–65.

Stoner, Oliver, Theo Economou, and Gabriela Drummond Marques da Silva. 2019. "A Hierarchical Framework for Correcting Under-Reporting in Count Data." *Journal of the American Statistical Association*.

Urdangarin, Arantxa, Tomas Goicoa, T. Kneib, and M. D. Ugarte. 2024. "A Simplified Spatial+ Approach to Mitigate Spatial Confounding in Multivariate Spatial Areal Models." *Spatial Statistics* 59: 100804. https://doi.org/10.1016/j.spasta.2023.100804.

Van Niekerk, Janet, Elias Krainski, Denis Rustand, and Haavard Rue. 2023. "A New Avenue for Bayesian Inference with INLA." *Computational Statistics and Data Analysis* 181. https://doi.org/10.1016/j.csda.2023.107692.

Wøllo, Sara E. 2022. "Correcting for Under-Reporting of Violence Against Women in Italy Using INLA." NTNU Open. https://hdl.handle.net/11250/3026838.