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 in 2021-2023. `R` codes to generate the dataset are in the R script posted here which this report is based on.

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, the Tremiti Islands need to be removed 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:

```
dd %>% sf::st_drop_geometry() %>%
  dplyr::group_by(.data$Year) %>%
  dplyr::summarise(Tot_accesses = sum(.data$N_ACC))
```

```
## # A tibble: 3 x 2
##    Year Tot_accesses
##   <dbl>        <dbl>
## 1     1         1477
## 2     2         1516
## 3     3         1822
```

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_th, 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

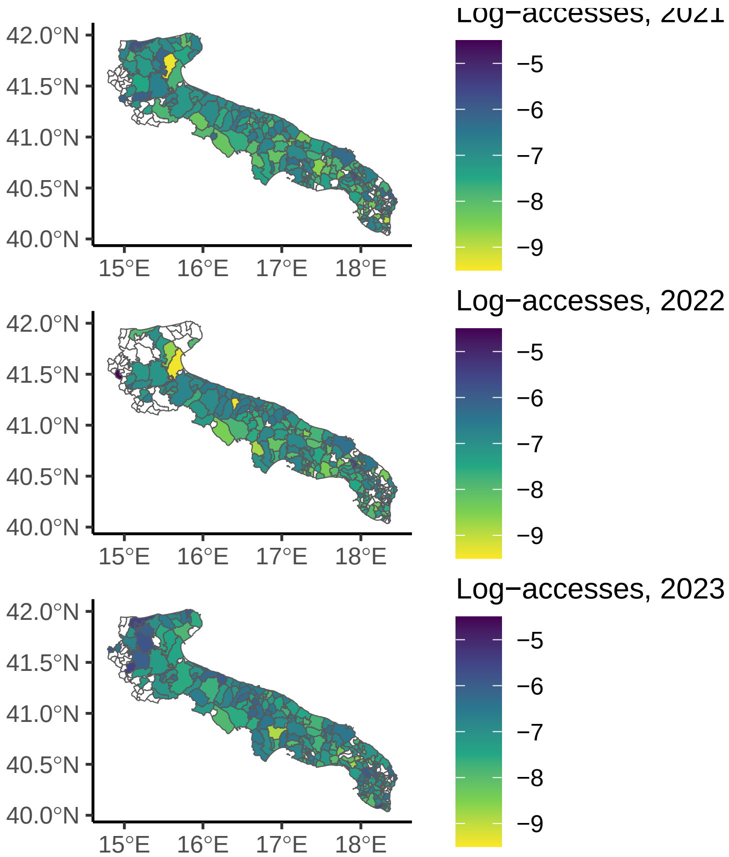


Figure 1: Log-access rate

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.
- 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:

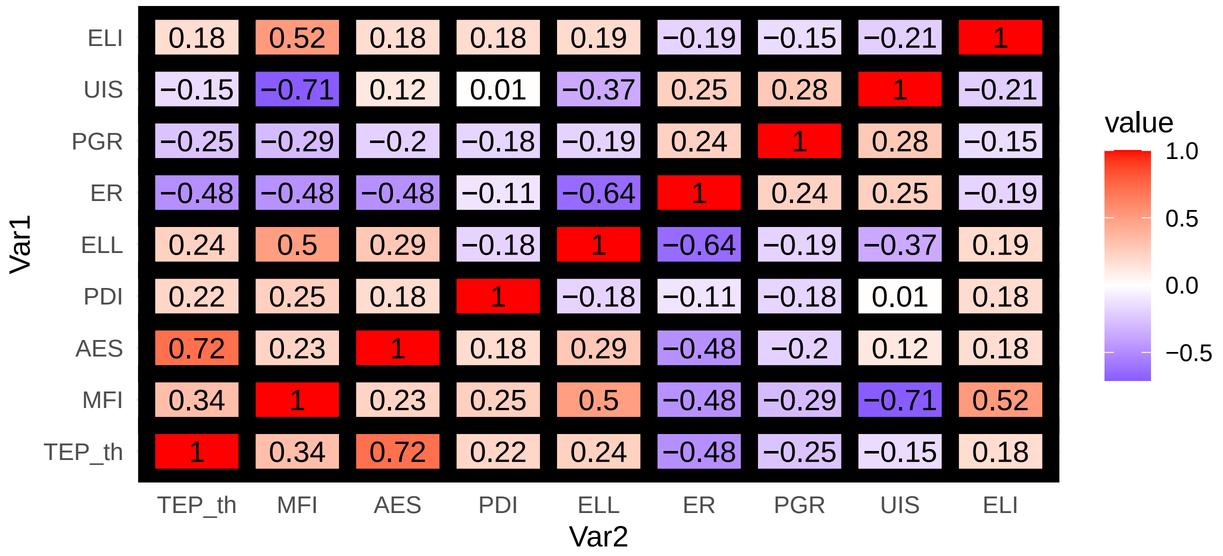


Figure 2: Correlations in explanatory variables

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 regression model

% latex table generated in R 4.4.1 by xtable 1.8-4 package % Wed Apr 30 18:58:40 2025

Effect	Mean	Sd	0.025quant	0.975quant
Int_2021	-7.343	0.032	-7.405	-7.281
Int_2022	-7.315	0.031	-7.376	-7.254
Int_2023	-7.139	0.030	-7.197	-7.080
TEP_th	-0.256	0.020	-0.296	-0.217
ELI	-0.058	0.018	-0.093	-0.023
PGR	0.033	0.023	-0.012	0.079
UIS	0.013	0.019	-0.024	0.050
ELL	-0.161	0.024	-0.209	-0.114
PDI	-0.064	0.024	-0.112	-0.017
ER	-0.221	0.026	-0.272	-0.169

- `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 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

```r
zhat_plot <- function(mod){

  rr <- range(mod$summary.random$ID$mean)
  plot_map <- purrr::map(unique(dd$Year), function (t){
  dd %>% dplyr::mutate(zhat = mod$summary.random$ID$mean) %>%
    dplyr::filter(.data$Year == t) %>%
    ggplot2::ggplot() +
    ggplot2::geom_sf(ggplot2::aes(fill = .data$zhat))+
    ggplot2::labs(title = paste("Year:", t), fill = "zhat") +
    ggplot2::scale_fill_viridis_c(na.value = "white", direction = -1, limits = rr) +
    ggplot2::theme_classic()
})

do.call(gridExtra::grid.arrange, c(plot_map, nrow = 1, ncol = 3))

}
```

Exploratory analysis of residuals 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.

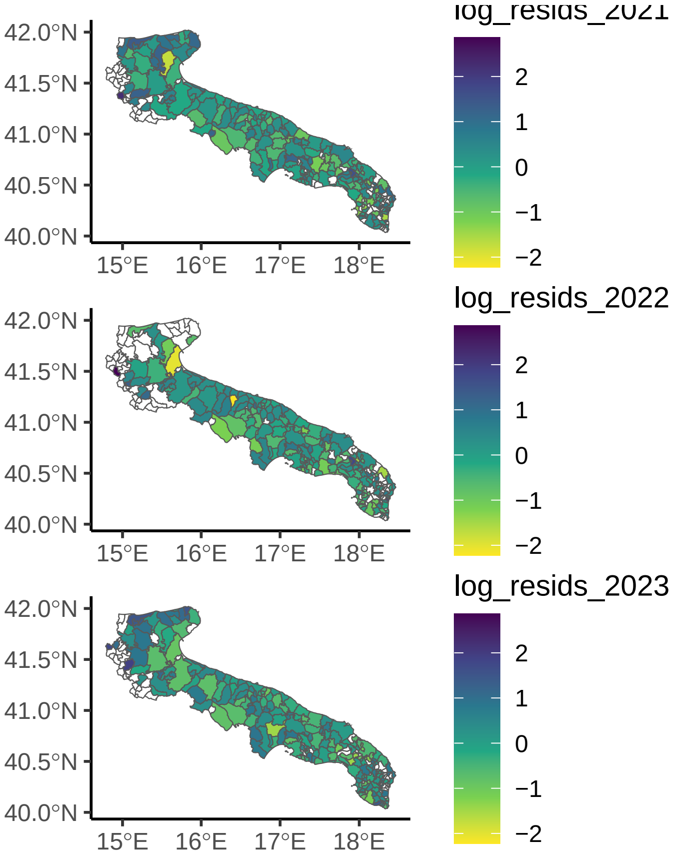


Figure 3: Log-residuals in GLM regression

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 $N(0, 1)$. The Geary's test is also included for completeness.

% latex table generated in R 4.4.1 by xtable 1.8-4 package % Wed Apr 30 18:58:43 2025

Year	Test	Statistic_std	p.value
2021	Moran	2.04	0.02
2021	Geary	2.48	0.01
2022	Moran	1.91	0.03
2022	Geary	2.14	0.02
2023	Moran	5.00	0.00
2023	Geary	5.05	0.00

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 $z_i = (z_{i,2021}^{\top}\, z_{i,2022}^{\top}\, z_{i,2023}^{\top})^{\top}$

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:

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

And the joint prior distribution is:

$$z \mid \Sigma \left(0, \Sigma \otimes (D - W)^{+}\right) \tag{4}$$

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 each year. 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 $\rho = (\rho_{2021}, \rho_{2022}, \rho_{2023})$.

For the sake of notation, z can be conveniently parametrised as $z := UM$, with $M : /M^\top M = \Sigma$, and $U = (U_{2021}^\top, U_{2022}^\top, U_{2023}^\top)^\top$, for $t = 2021, 2022, 2023$ $U_t \sim N(0, (D - \rho_t W)^{-1})$ and $U_t \perp U_s$ for $t \neq s$.

This modelling framework, generalising univiariate autoregressions, goes under the name of M-models (Botella-Rocamora, Martinez-Beneito, and Banerjee 2015). The joint prior for the M-PCAR is:

$z \mid \rho, \Sigma \sim N\left(0, (M^\top \otimes I_n)\mathrm{Bdiag}(D - \rho_t W)(M \otimes I_n)\right)$

$$z \mid \rho, \Sigma \left((M^\top \otimes I_n)(\mathrm{Bdiag}(D - \rho_t W))(M \otimes I_n)\right) \tag{5}$$

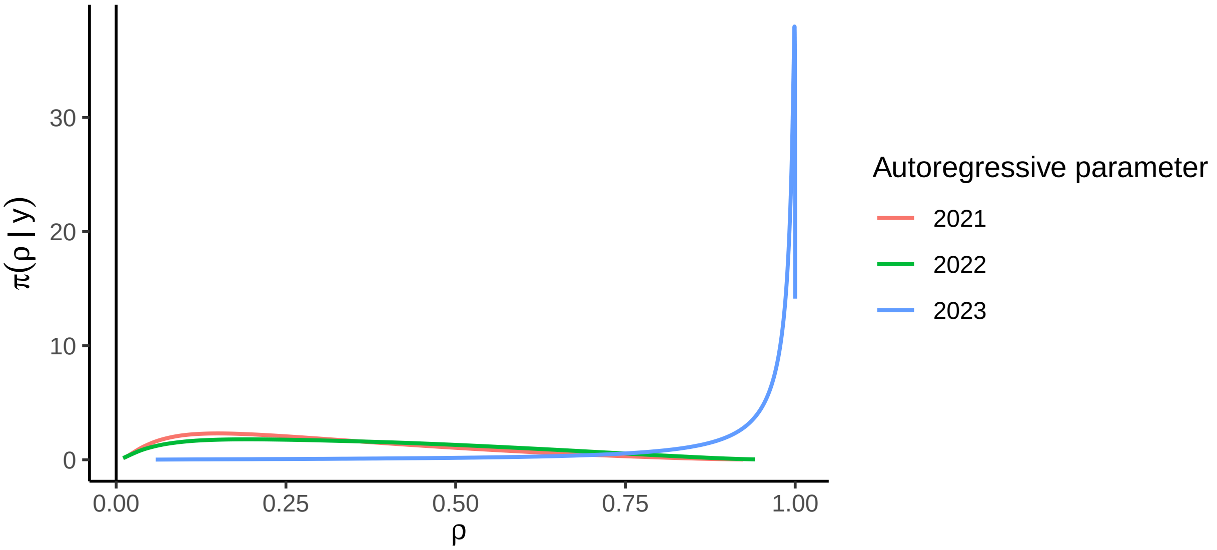
Where $\mathrm{Bdiag}(A_i)$ denotes a block-diagonal matrix with blocks given by A_i for an appropriate set of indices. The R code for implementing M-models is based on the script of (Palmí-Perales, Gómez-Rubio, and Martinez-Beneito 2021), which has been extended to fit also the LCAR and BYM models (vd. infra)

```
inla.rgeneric.Mmodel <-
  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()
      if(Qmod == "BYM"){
        #' 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$Q[c(1:n), c(1:n)]
        Sigma.u <- MASS::ginv(as.matrix(L))
        endtime.scale <- Sys.time()
        cat("Time needed for scaling Laplacian matrix: ",
            round(difftime(endtime.scale, starttime.scale), 3), " seconds \n")
        assign("Sigma.u", Sigma.u, envir = envir)
      }
      assign("cache.done", TRUE, envir = envir)
    }
    interpret.theta <- function() {
      alpha <- alpha.min + (alpha.max - alpha.min)/(1 + exp(-theta[as.integer(1:k)]))
      M <- matrix(theta[-as.integer(1:k)], ncol = k)
      return(list(alpha = alpha, M = M))
    }
    graph <- function() {
      MI <- kronecker(Matrix::Matrix(1, ncol = k, nrow = k),
                      Matrix::Diagonal(nrow(W), 1))
      IW <- Matrix::Diagonal(nrow(W), 1) + W
      BlockIW <- Matrix::bdiag(replicate(k, IW, simplify = FALSE))
      G <- (MI %*% BlockIW) %*% MI
      return(G)
    }
    Q <- function() {
      param <- interpret.theta()
      M.inv <- solve(param$M)
      MI <- kronecker(M.inv, Matrix::Diagonal(nrow(W), 1))
      D <- as.vector(apply(W, 1, sum))
      if(Qmod == "LCAR"){
        print("tempcheck: model = LCAR")
        BlockIW <- Matrix::bdiag(lapply(1:k, function(i) {
          param$alpha[i]*(Matrix::Diagonal(x = D) - W) +
            (1 - param$alpha[i]) * Matrix::Diagonal(nrow(W), 1)
        }))
      } else if (Qmod == "BYM"){
        print("tempcheck: model = BYM")
        BlockIW <- Matrix::bdiag(lapply(1:k, function(i) {
          solve(
            sqrt(param$alpha[i])*Sigma.u+
              sqrt(1-param$alpha[i]*Matrix::Diagonal(nrow(W), 1))
          )
        }))
      } else if(Qmod == "PCAR"){
        print("tempcheck: model = PCAR")
        BlockIW <- Matrix::bdiag(lapply(1:k, function(i) {
          Matrix::Diagonal(x = D) - param$alpha[i] * W
        }))
      }

      Q <- (MI %*% BlockIW) %*% kronecker(t(M.inv), Matrix::Diagonal(nrow(W),  1))

      return(Q)
    }
    mu <- function() {
      return(numeric(0))
    }
    log.norm.const <- function() {
      val <- numeric(0)
      return(val)
    }
    log.prior <- function() {
      param <- interpret.theta()
      val <- sum(-theta[as.integer(1:k)] - 2 * log(1 + exp(-theta[as.integer(1:k)])))
      sigma2 <- 1000
      val = val + log(MCMCpack::dwish(W = crossprod(param$M),
                                      v = k, S = diag(rep(sigma2, k))))
      return(val)
    }
    initial <- function() {
      return(c(rep(0, k), as.vector(diag(rep(1, k)))))
    }
    quit <- function() {
      return(invisible())
    }
    if (as.integer(R.version$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.Mmodel <- function (...)    INLA::inla.rgeneric.define(inla.rgeneric.Mmodel, ...)
```

We show posterior summaries of the autocorrelation parameter. As it can be seen, the spatial component of the first two years is almost randomly distributed, while spatial autocorrelation is very strong for 2023.



% latex table generated in R 4.4.1 by xtable 1.8-4 package % Wed Apr 30 19:00:37 2025

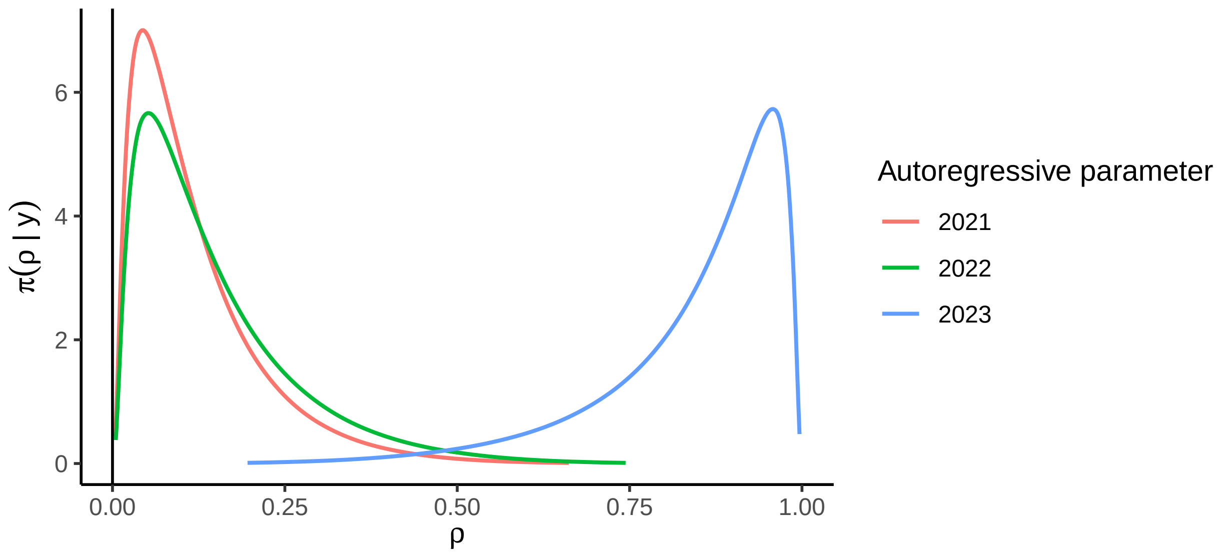
	mean	sd	quant0.025	quant0.5	quant0.975
rho1	0.314	0.192	0.046	0.279	0.748
rho2	0.368	0.210	0.050	0.343	0.802
rho3	0.906	0.147	0.434	0.967	0.999

LCAR model As an alternative to relax the hypothesis of spatial autocorrelation = 1, we also test the Leroux autoregressive model (Leroux, Lei, and Breslow 2000), LCAR hereinafter. In this case, the joint prior for z is

$$z \mid \xi, \Sigma \sim N\left(0, (M^\top \otimes I_n)\text{Bdiag}[\xi_t(D - W) + (1 - \xi_t)I_n](M \otimes I_n)\right) \tag{6}$$

With notation alike equation 5. $\xi \in [0, 1]^3$ is the mixing parameter.

We show posterior summaries of the autocorrelation parameter. As it can be seen, the spatial component of the first two years is almost randomly distributed, while spatial autocorrelation is very strong for 2023.



% latex table generated in R 4.4.1 by xtable 1.8-4 package % Wed Apr 30 19:04:37 2025

	mean	sd	quant0.025	quant0.5	quant0.975
rho1	0.121	0.095	0.015	0.094	0.376
rho2	0.148	0.115	0.018	0.116	0.455
rho3	0.847	0.129	0.498	0.886	0.985

Appendices

```
inla.rgeneric.Mmodel <-
  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()
      if(Qmod == "BYM"){
        #' 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$Q[c(1:n), c(1:n)]
        Sigma.u <- MASS::ginv(as.matrix(L))
        endtime.scale <- Sys.time()
        cat("Time needed for scaling Laplacian matrix: ",
            round(difftime(endtime.scale, starttime.scale), 3), " seconds \n")
        assign("Sigma.u", Sigma.u, envir = envir)
      }
      assign("cache.done", TRUE, envir = envir)
    }
    interpret.theta <- function() {
      alpha <- alpha.min + (alpha.max - alpha.min)/(1 + exp(-theta[as.integer(1:k)]))
      M <- matrix(theta[-as.integer(1:k)], ncol = k)
      return(list(alpha = alpha, M = M))
    }
    graph <- function() {
      MI <- kronecker(Matrix::Matrix(1, ncol = k, nrow = k),
                      Matrix::Diagonal(nrow(W), 1))
      IW <- Matrix::Diagonal(nrow(W), 1) + W
      BlockIW <- Matrix::bdiag(replicate(k, IW, simplify = FALSE))
      G <- (MI %*% BlockIW) %*% MI
      return(G)
    }
    Q <- function() {
      param <- interpret.theta()
      M.inv <- solve(param$M)
      MI <- kronecker(M.inv, Matrix::Diagonal(nrow(W), 1))
      D <- as.vector(apply(W, 1, sum))
      if(Qmod == "LCAR"){
        BlockIW <- Matrix::bdiag(lapply(1:k, function(i) {
          param$alpha[i]*(Matrix::Diagonal(x = D) - W) +
            (1 - param$alpha[i]) * Matrix::Diagonal(nrow(W), 1)
        }))
      } else if (Qmod == "BYM"){
        BlockIW <- Matrix::bdiag(lapply(1:k, function(i) {
          solve(
            sqrt(param$alpha[i])*Sigma.u+
              sqrt(1-param$alpha[i]*Matrix::Diagonal(nrow(W), 1))
          )
        }))
      } else if(Qmod == "PCAR"){
        BlockIW <- Matrix::bdiag(lapply(1:k, function(i) {
          Matrix::Diagonal(x = D) - param$alpha[i] * W
        }))
      }

      Q <- (MI %*% BlockIW) %*% kronecker(t(M.inv), Matrix::Diagonal(nrow(W),  1))



      return(Q)
    }
    mu <- function() {
      return(numeric(0))
    }
    log.norm.const <- function() {
      val <- numeric(0)
      return(val)
    }
    log.prior <- function() {
      param <- interpret.theta()
      val <- sum(-theta[as.integer(1:k)] - 2 * log(1 + exp(-theta[as.integer(1:k)])))
      sigma2 <- 1000
      val = val + log(MCMCpack::dwish(W = crossprod(param$M),
                                      v = k, S = diag(rep(sigma2, k))))
      return(val)
    }
    initial <- function() {
      return(c(rep(0, k), as.vector(diag(rep(1, k)))))
    }
    quit <- function() {
      return(invisible())
    }
    if (as.integer(R.version$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.Mmodel <- function (...)    INLA::inla.rgeneric.define(inla.rgeneric.Mmodel, ...)
```

R code for the M-model

Botella-Rocamora, Paloma, Miguel A Martinez-Beneito, and Sudipto Banerjee. 2015. "A Unifying Modeling Framework for Highly Multivariate Disease Mapping." *Statistics in Medicine* 34 (9): 1548–59.

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.

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.

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.