Table of contents I

Sustainable Development Goal

Survey Limitations.

Introduction to Bayesian Thinking.

Use of SAE Methods

Generalized Variance Function (GVF)

Area Models

Unit Models

Table of contents II

Unit Model Multidimensional Deprivation Index (MDI)

Area Model for Labor Market Statistics

Sustainable Development Goal


OBJETIVOS DE DESARROLLO SOSTENIBLE
1 FIN DE LA POBREZA
2 HAMBRE CERO
3 SALUD Y BIENESTAR
4 EDUCACIÓN DE CALIDAD
5 IGUALDAD DE GÉNERO
6 AGUA LIMPIA Y SANEAMIENTO
7 ENERGÍA ASEQUIBLE Y NO CONTAMINANTE
8 TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO
9 INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA
10 REDUCCIÓN DE LAS DESIGUALDADES
11 CIUDADES Y COMUNIDADES SOSTENIBLES
12 PRODUCCIÓN Y CONSUMO RESPONSABLES
13 ACCIÓN POR EL CLIMA
14 VIDA SUBMARINA
15 VIDA DE ECOSISTEMAS TERRESTRES
16 PAZ, JUSTICIA E INSTITUCIONES SÓLIDAS
17 ALIANZAS PARA LOGRAR LOS OBJETIVOS
OBJETIVOS DE DESARROLLO SOSTENIBLE

Some targets of SDG 2 (Zero Hunger)

By 2030, end hunger and ensure access to all people, particularly the poor and those in vulnerable situations, including infants under 1 year of age, to healthy, nutritious, and sufficient food all year round.

- Prevalence of undernourishment.
- Prevalence of moderate or severe food insecurity in the population, according to the Food Insecurity Experience Scale.

Some targets of SDG 8 (Decent Work)

By 2030, achieve full and productive employment and decent work for all women and men, including youth and persons with disabilities, and equal pay for equal work.

- Unemployment rate, disaggregated by gender, age, and persons with disabilities.

Fundamental Principle of Data Disaggregation

Indicators of the Sustainable Development Goals should be disaggregated, whenever relevant, by income, gender, age, race, ethnicity, migratory status, disability, and geographical location, or other characteristics, in accordance with the Fundamental Principles of Official Statistics.

General Assembly Resolution - 68/261

Survey Limitations.

What is the Coefficient of Variation?

The coefficient of variation is a measure of relative error for an estimator and is defined as:

$$cve\left(\hat{\theta}\right) = \frac{SE\left(\hat{\theta}\right)}{\hat{\theta}}$$

It is often expressed as a percentage, even though it is not bounded to the right, which makes it convenient when discussing the precision of statistics derived from surveys.

Alert Standards in Some Countries (Household Surveys)

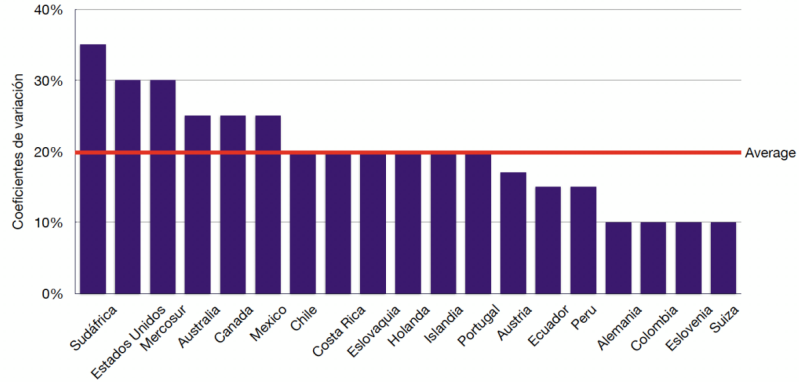


Figure 1: Alerts on Coefficients of Variation

Some Alerts Defined in the Publication

When the threshold of the coefficient of variation is exceeded, some of the following alerts may appear:

- Not published.
- Use with caution.
- Estimates require revisions, are not precise, and should be used with caution.
- Unreliable, less precise.
- Does not meet publication standards.
- With reservation, reference, questionable.
- Very random values, poor estimation.

Study Domains and Subpopulations of Interest

A survey is designed to generate accurate and reliable information within predefined study domains. However, there are subpopulations that the survey did not address in its design but for which greater precision is desired.

- Poverty incidence disaggregated by department or province (known and planned sample size).
- Unemployment rate disaggregated by gender (random but planned sample size).
- Net primary school attendance rate disaggregated by income quintiles (random sample size).

Precision of Estimators

Because a survey is a partial investigation of a finite population, it's important to know that:

- Indicators are not calculated from a survey; they are estimated using survey data.
- It is necessary to calculate the degree of error resulting from the inability to conduct a comprehensive investigation. This error is known as sampling error.
- The precision of an estimator is dependent on the confidence interval.

A narrower interval results in greater precision and, therefore, lower sampling error.

Effective Sample Size

- In household surveys with complex sampling designs, there is no sequence of variables that are independent and identically distributed.
- The sample $y_1, \ldots, y_n$ is not a vector in an n-dimensional space, where each component of the vector can vary independently.
- The final dimension of the vector $(y_1, \ldots, y_n)$ is much smaller than n, due to hierarchical sampling and the relationship between the variable of interest and primary sampling units (PSUs).

Effective Sample Size

The effective sample size is defined as follows:

$$n_{effective} = \frac{n}{Deff}$$

Where Deff is the design effect, which depends on: *1.* The average number of surveys conducted in each PSU. *2.* The correlation between the variable of interest and the same PSUs.

It can be considered that if the effective sample size is not greater than a threshold, then the figure should not be considered for publication.

Degrees of Freedom

In subpopulations, degrees of freedom are not considered fixed but rather variable.

$$df = \sum_{h=1}^{H} v_h \times (n_{Ih} - 1)$$

Note that ν_h is an indicator variable that takes the value one if stratum h contains one or more cases of the subpopulation of interest, and n_{Ih} is the number of primary sampling units (PSUs) in the stratum. In the most general case, degrees of freedom are reduced to the following expression:

```
df = #PSUs - #Stratum
```

Introduction to Bayesian Thinking.

Models of Areas with the **Tom Approach**

And you wake up one day...

- You feel a little strange and weak. You go to the doctor, and they run some tests. One of them comes back positive for a very rare disease that only affects 0.1% of the population.

Not good news.

- You go to the doctor's office and ask how specific the test is. They tell you it's very accurate; it correctly identifies 99% of the people who have the disease.

And you meet Thomas...

Esta es la información que tienes:

- $\mathrm{P(E)} = 0.001$
- $\mathrm{P}(+|\mathrm{E}) = 0.99$
- $\mathrm{P}(-\mathrm{E}) = 0.999$
- $\mathrm{Pr}(+|-\mathrm{E}) = 0.01$

Además, por el teorema de probabilidad total

$$\begin{aligned}\mathrm{P}(+) &= \mathrm{Pr(E)Pr}(+|\mathrm{E}) + \mathrm{Pr}(-\mathrm{E})\mathrm{Pr}(+|-\mathrm{E}) \\ &= 0.001 * 0.99 + 0.999 * 0.01 \\ &= 0.01098\end{aligned}$$

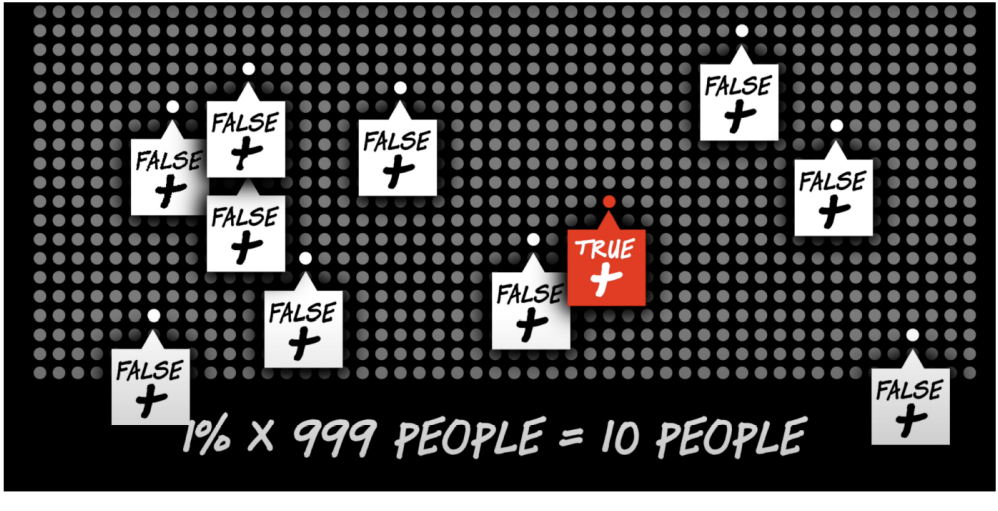
La regla de Bayes afirma lo siguiente:

$$\mathrm{Pr}(\mathrm{E}|+) = \frac{\mathrm{Pr}(+|\mathrm{E}) \times \mathrm{Pr(E)}}{\mathrm{Pr}(+)}$$

Por lo tanto:

$$\mathrm{Pr}(\mathrm{E}|+) = 0.09 \approx 9\%$$

How does it work?



How does it work?

And you seek a second opinion

- This time the doctor orders you to retake the same test... and you test positive for that disease again.
- **And you wonder again:** *What is the probability that I have this disease?*

This time, you've updated your information about $Pr(E)$ because you've tested positive on a test

$$Pr(E) = 0.09 \; And \; Pr(-E) = 0.91$$

Therefore:

$$Pr(E \mid ++) = 0.997 \approx 91\%$$

Elements of Bayes' Rule

In terms of inference for θ, it's necessary to find the distribution of the parameters conditioned on the observation of the data. To achieve this, you need to define the joint distribution of the variable of interest with the parameter vector.

$$p(\theta, Y) = p(\theta)p(Y \mid \theta)$$

- The distribution $p(\theta)$ is known as the prior distribution.
- The term $p(Y \mid \theta)$ is the sampling distribution, likelihood, or data distribution.
- The distribution of the parameter vector conditioned on the observed data is given by

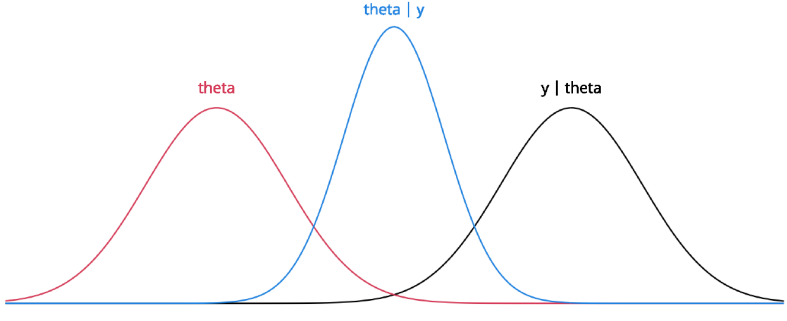
$$p(\theta \mid Y) = \frac{p(\theta, Y)}{p(Y)} = \frac{p(\theta)p(Y \mid \theta)}{p(Y)}$$

Bayes' Rule

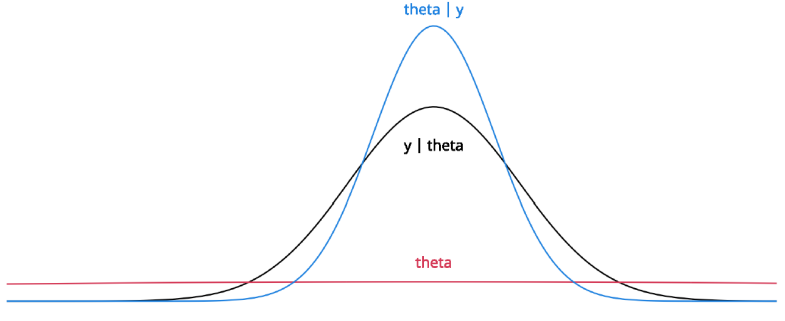
- The term $p(\theta \mid Y)$ is known as the ***posterior*** distribution.
- The denominator does not depend on the parameter vector, and considering the observed data as fixed, it corresponds to a constant and can be omitted. Therefore,

$$p(\theta \mid Y) \propto p(Y \mid \theta)p(\theta)$$

Informative Prior Distribution for θ



Non-Informative Prior Distribution for θ



Poisson Area Model

Suppose $Y = \{Y_1, \ldots, Y_n\}$ is a random sample of variables with a Poisson distribution with parameter θ. The joint distribution function or likelihood function is given by

$$p(Y \mid \theta) = \prod_{i=1}^{n} \frac{e^{-\theta}\theta^{y_i}}{y_i!} I_{\{0,1,\ldots\}}(y_i)$$

$$= \frac{e^{-n\theta}\theta^{\sum_{i=1}^{n} y_i}}{\prod_{i=1}^{n} y_i!} I_{\{0,1,\ldots\}^n}(y_1, \ldots, y_n)$$

where $\{0, 1, ...\}^n$ denotes the Cartesian product n times over the set $\{0, 1, ...\}$.

The parameter θ is restricted to the space $\Theta = (0, \infty)$.

Distribución previa para θ

- La distribución previa del parámetro θ dada por

$$p(\theta \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta} I_{(0,\infty)}(\theta).$$

- La distribución posterior del parámetro θ está dada por

$$\theta \mid Y \sim Gamma\left(\sum_{i=1}^{n} y_i + \alpha, n + \beta\right)$$

Prior Distribution for θ

- The prior distribution for the parameter θ is given by

$$p(\theta \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta} I_{(0,\infty)}(\theta).$$

- The posterior distribution for the parameter θ is given by

Estimation Process in **STAN**

Let Y be the count of surveyed people living below the poverty line, expressed as a rate of (X) per 100 inhabitants, by administrative division of the country.

```
dataPois <- readRDS("www/00_Intro_bayes/Poisson/dataPoisson.rds")
```

Table 1: People Count

dam2	n
05002	2
05031	1
05034	1
05045	2
05079	1
05088	6
05093	1
05120	2
05129	1
05142	1

Histogram with People Count

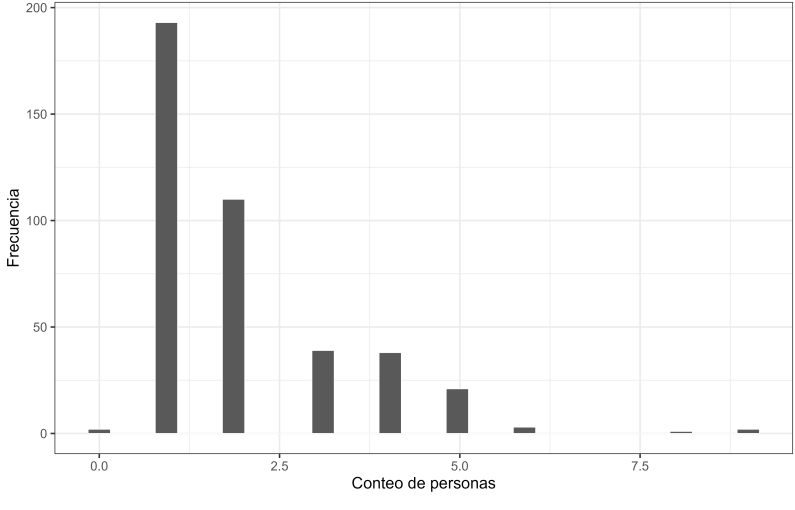


Figure 2: People count by administrative division

Model Written in STAN Code

```
data {
  int<lower=0> n;       // Number of geographic areas
  int<lower=0> y[n];    // Counts per area
  real<lower=0> alpha;
  real<lower=0> beta;
}
parameters {
  real<lower=0> theta;
}
model {
  y ~ poisson(theta);
  theta ~ gamma(alpha, beta);
}
generated quantities {
    real ypred[n];                    // Vector of length n
    for(ii in 1:n){
    ypred[ii] = poisson_rng(theta);
    }
}
```

Preparing Data for STAN Code

- Organizing data for STAN

```
sample_data <- list(n = nrow(dataPois), y = dataPois$n,
                    alpha = 0.001, beta = 0.001)
```

- Running the STAN code

```
stan_pois <- "www/00_Intro_bayes/Poisson/03_Poisson.stan"
model_poisson <-
  stan(
    file = stan_pois, data = sample_data,
    warmup = 500,
    iter = 1000,
    verbose = FALSE,   cores = 4
  )
saveRDS(model_poisson,
          "www/00_Intro_bayes/Poisson/model_poisson.rds")
```

Results of the Estimation of Parameter θ

```
tabla_posi <- summary(model_poisson,
                      pars =c("theta"))$summary
tabla_posi%>%tba()
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
theta	2.03	0.0023	0.0685	1.901	1.982	2.032	2.077	2.166	851.6	1.002

Convergence of Chains for Parameter θ

```
posterior_theta <- as.array(model_poisson, pars = "theta")
p1 <- (mcmc_dens_chains(posterior_theta) +
    mcmc_areas(posterior_theta) ) / mcmc_trace(posterior_theta)
```

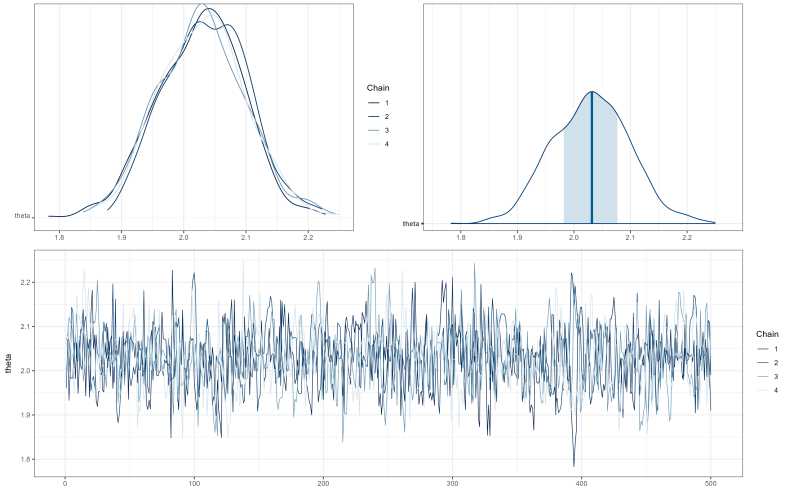


Figure 9: Chains for theta

Posterior Predictive Check

```
y_pred_B<-as.array(model_poisson,pars ="ypred") %>%
  as_draws_matrix()

rowsrandom<-sample(nrow(y_pred_B),100)

y_pred2<-y_pred_B[rowsrandom,]

p1<- ppc_dens_overlay(y =as.numeric(dataPois$n*100), y_pred2*100)
```

Posterior Predictive Check

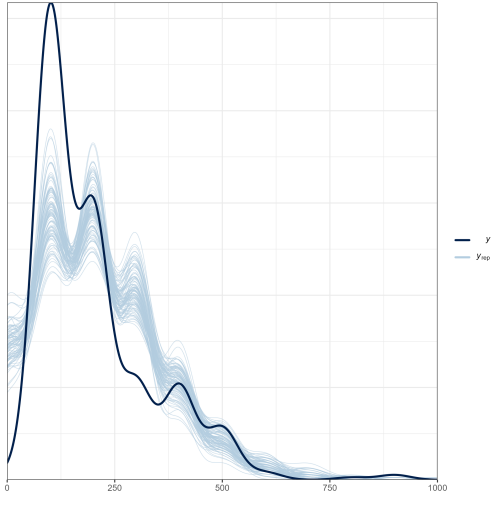


Figure 4: Chains for theta

Unit Model: Normal with Unknown Mean and Variance

- In the normal model, a set of independent and identically distributed variables $Y_1, \cdots, Y_n \sim N(\theta, \sigma^2)$ is considered.
- When both the mean and the variance of the distribution are unknown, different approaches are proposed to assign prior distributions to θ and σ^2 based on the problem context.

Prior Distributions for θ and σ^2

Three possible assumptions about prior distributions for θ and σ^2 are described, considering independence and informativeness.

- Assume that the prior distribution $p(\theta)$ is independent of the prior distribution $p(\sigma^2)$ and that both distributions are informative.
- Assume that the prior distribution $p(\theta)$ is independent of the prior distribution $p(\sigma^2)$ and that both distributions are non-informative.
- Assume that the prior distribution for θ depends on σ^2 and write it as $p(\theta \mid \sigma^2)$, while the prior distribution for σ^2 does not depend on θ and can be written as $p(\sigma^2)$.

The prior distribution is set for the parameter θ as $\theta \sim Normal(0, 10000)$ and for the parameter σ^2 as $\sigma^2 \sim IG(0.0001, 0.0001)$.

Definition of the Normal Model

- The goal of the model is to estimate the average income of people, represented as $\bar{Y}_d = \frac{\sum_{U_d} y_{di}}{N_d}$.
- A way to estimate $\bar{Y}$ using $\hat{y}_{di}$, which is the expected value of y_{di} under a probability measure induced by the model, is shown.
- Finally, the estimate of $\hat{\bar{Y}}_d = \frac{\sum_{U_d} \hat{y}_{di}}{N_d}$ is presented.

Estimation Process

- To estimate the average income of people, i.e.,

$$\bar{Y}_d = \frac{\sum_{U_d} y_{di}}{N_d}$$

where y_{di} is the income of each person. Note that

$$\bar{Y}_d = \frac{\sum_{s_d} y_{di} + \sum_{s_d^c} y_{di}}{N_d}$$

Now, the estimator of $\bar{Y}$ is given by:

$$\hat{\bar{Y}}_d = \frac{\sum_{s_d} y_{di} + \sum_{s_d^c} \hat{y}_{di}}{N_d}$$

Estimation Process

Now, it is possible to assume that $\hat{y}_{di}$ is the conditional expectation given the modeling, that is

$$\hat{y}_{di} = E_{\mathcal{M}}\left(y_{di} \mid x_d, \beta\right)$$

,

where $\mathcal{M}$ refers to the probability measure induced by the modeling. Finally, it is found that

$$\hat{\bar{Y}}_d = \frac{\sum_{U_d} \hat{y}_{di}}{N_d}$$

Estimation Process in **STAN**

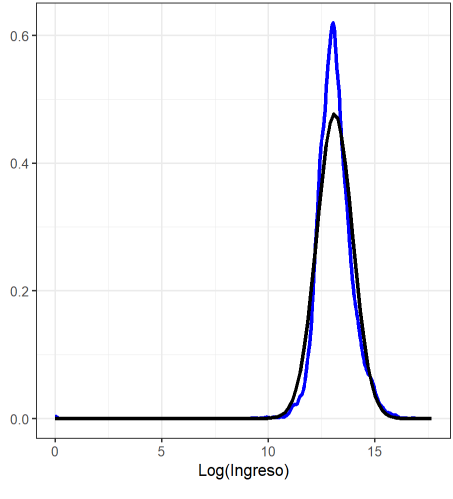
Let Y be the logarithm of income for an administrative division of the country.

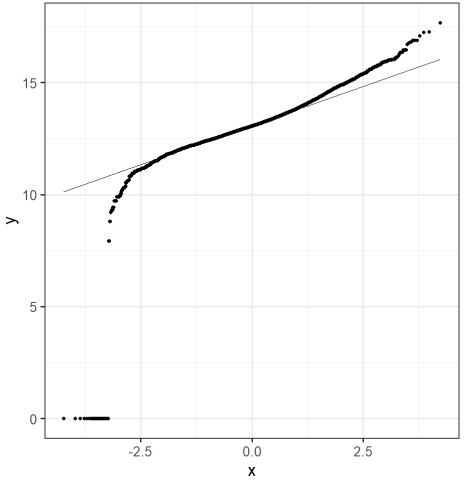
```
dataNormal <- readRDS("www/00_Intro_bayes/Normal/01_dataNormal.rds")
tba(dataNormal %>% head(10), cap = "Logarithm of income" )
```

Table 2: Logarithm of income

dam_ee	logIngreso
08	14.37
08	14.37
08	14.37
08	12.02
08	12.02
08	12.02
08	14.05
08	14.05
08	14.05
08	14.05

Graphical Analysis of Logarithm of Income





Model Written in STAN Code

```
data {
  int<lower=0> n;
  real y[n];
}
parameters {
  real sigma;
  real theta;
}
transformed parameters {
  real sigma2;
  sigma2 = pow(sigma, 2);
}
```

```
model {
  y ~ normal(theta, sigma);
  theta ~ normal(0, 1000);
  sigma2 ~ inv_gamma(0.001, 0.001);
}
generated quantities {
    real ypred[n];
    for(kk in 1:n){
    ypred[kk] = normal_rng(theta,sigma);
}
}
```

Preparing Data for the STAN Code

- Organizing data for STAN

```
sample_data <- list(n = nrow(dataNormal),
                    y = dataNormal$logIngreso)
```

- Running STAN from R using the **rstan** library

```
NormalMeanVar  <- "www/00_Intro_bayes/Normal/03_NormalMeanVar.stan"
model_NormalMedia <- stan(
  file = NormalMeanVar,
  data = sample_data,
  warmup = 500,
  iter = 1000,
  verbose = FALSE, cores = 4
)
saveRDS(model_NormalMedia,
        "www/00_Intro_bayes/Normal/model_NormalMedia2.rds")
```

Results of the Estimation of the Parameters θ and σ^2 are:

```
tabla_Nor2 <- summary(model_NormalMedia,
        pars = c("theta", "sigma2", "sigma"))$summary
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
theta	13.1147	1e-04	0.0039	13.1068	13.1120	13.1148	13.1172	13.1224	1221	0.9996
sigma2	0.6986	1e-04	0.0047	0.6894	0.6955	0.6986	0.7016	0.7080	1931	1.0001
sigma	0.8358	1e-04	0.0028	0.8303	0.8340	0.8358	0.8376	0.8415	1931	1.0001

Convergence of Chains for Parameter θ

```
posterior_theta <- as.array(model_NormalMedia, pars = "theta")
(mcmc_dens_chains(posterior_theta) +
    mcmc_areas(posterior_theta) ) /
  mcmc_trace(posterior_theta)
```

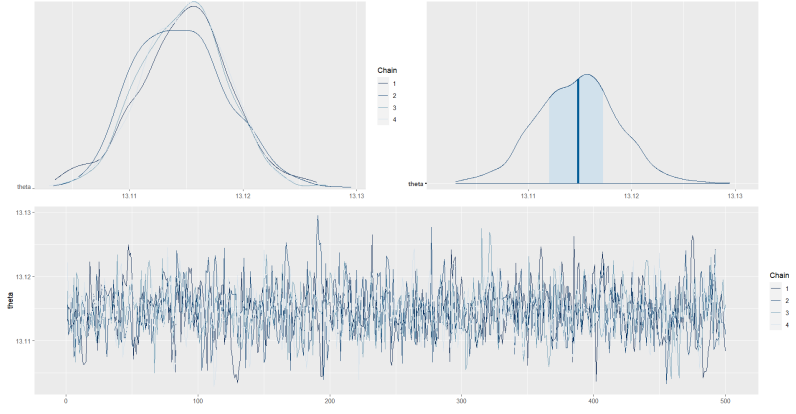


Figure 5: Chains for theta

Convergence of Chains for Parameter σ^2

```
posterior_sigma2 <- as.array(model_NormalMedia, pars = "sigma2")
(mcmc_dens_chains(posterior_sigma2) +
    mcmc_areas(posterior_sigma2) ) /
  mcmc_trace(posterior_sigma2)
```

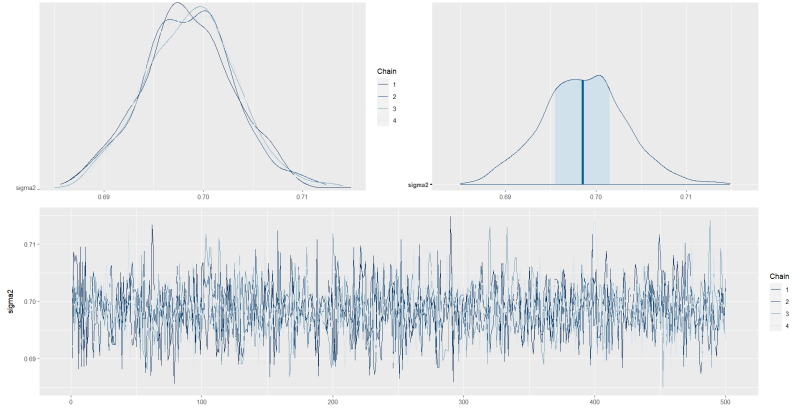


Figure 6: Chains for sigma2

Posterior Predictive Check of Income

```
y_pred_B <- as.array(model_NormalMedia, pars = "ypred") %>%
  as_draws_matrix()

rowsrandom <- sample(nrow(y_pred_B), 100)

y_pred2 <- y_pred_B[rowsrandom,]

ppc_dens_overlay(
  y = as.numeric(exp(dataNormal$logIngreso) - 1), y_pred2) +
  xlim(0, 5000000)
```

Posterior Predictive Check of Income

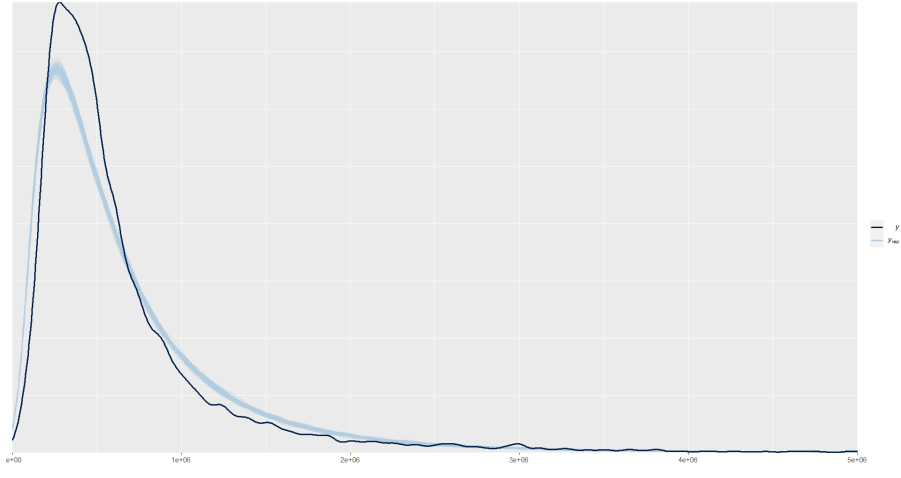


Figure 7: PPC for income

Linear Models.

Linear regression is the basic technique in econometric analysis. Through this technique, we aim to determine linear dependence relationships between a dependent variable, or endogenous variable, and one or more explanatory variables, or exogenous variables.

Bayesian Linear Models

First, note that the particular interest lies in the distribution of the vector of n random variables $Y = (Y_1, \dots, Y_n)'$ conditioned on the matrix of auxiliary variables X and indexed by the vector of parameters of interest $\beta = (\beta_1, \dots, \beta_q)'$ given by $p(Y \mid \beta, X$.

The basic and classical model assumes that the likelihood for the variables of interest is:

$$Y \mid \theta, \sigma^2, X \sim \text{Normal}_n(X\beta, \sigma^2 I_n)$$

where I_n denotes the identity matrix of order $n \times n$. Of course, the normal model is not the only one that can be postulated as the likelihood for the data.

Independent Parameters

Assuming that the parameters are independent a priori, meaning that the joint prior distribution is given by:

$$p(\beta, \sigma^2) = p(\beta)p(\sigma^2)$$

Naturally, the prior distribution for the parameter vector β is normal. However, this time, the variance-covariance matrix will not depend on the other parameter σ^2. So, you have:

$$\beta \sim \text{Normal}_q(b, B)$$

Similarly, the parameter σ^2 does not depend on β, and you can assign it the following prior distribution:

$$\sigma^2 \sim \text{Inverse-Gamma}\left(\frac{n_0}{2}, \frac{n_0\sigma_0^2}{2}\right)$$

Posterior Distribution

The joint posterior distribution of β and σ^2 can be written as:

$$
\begin{aligned}
p(\beta, \sigma^2 \mid Y, X) &\propto p(Y \mid \beta, \sigma^2) p(\beta) p(\sigma^2) \\
&\propto (\sigma^2)^{-n/2} \exp\left\{ -\frac{1}{2\sigma^2} \left(Q(\beta) + S_e^2 \right) \right\} \\
&\times \exp\left\{ -\frac{1}{2} (\beta - b)' B^{-1} (\beta - b) \right\} (\sigma^2)^{-n_0/2 - 1} \exp\left\{ -\frac{n_0 \sigma_0^2}{2\sigma^2} \right\} \\
&= (\sigma^2)^{-\frac{n+n_0}{2} - 1} \exp\left\{ -\frac{1}{2\sigma^2} \left[Q(\beta) + S_e^2 + n_0 \sigma_0^2 \right] \right\} \\
&\times \exp\left\{ -\frac{1}{2} (\beta - b)' B^{-1} (\beta - b) \right\} \qquad (1)
\end{aligned}
$$

Posterior Distribution of β

The posterior distribution of the parameter β conditioned on σ^2, Y, X is:

$$\beta \mid \sigma^2, Y, X \sim \text{Normal}_q(b_q, B_q)$$

Where:

$$B_q = \left(B^{-1} + \frac{1}{\sigma^2} X'X \right)^{-1}$$

$$b_q = B_q \left(B^{-1} b + \frac{1}{\sigma^2} X'Y \right)$$

Posterior Distribution of σ^2

The posterior distribution of the parameter σ^2 conditioned on β, Y, X is:

$$\sigma^2 \mid \beta, Y, X \sim \text{Inverse-Gamma}\left(\frac{n_1}{2}, \frac{n_1\sigma_\beta^2}{2}\right)$$

Where:

$$\begin{aligned} n_1 &= n + n_0 \\ n_1\sigma_\beta^2 &= Q(\beta) + S_e^2 + n_0\sigma_0^2 \\ Q(\beta) &= (\beta - \hat{\beta})'(X'X)(\beta - \hat{\beta}) \\ S_e^2 &= (y - X\hat{\beta})'(y - X\hat{\beta}) \end{aligned}$$

Here, σ_0^2 is a prior estimate of the parameter of interest σ^2.

Linear Model in STAN Code

```
data {
  int<lower=0> n;   // Number of observations
  int<lower=0> K;   // Number of predictors
  matrix[n, K] x;   // Matrix of predictors
  vector[n] y;      // Response vector
}
parameters {
  vector[K] beta;       // Coefficients
  real<lower=0> sigma2; // Error variance
}
model {
  to_vector(beta) ~ normal(0, 1000); // Prior for coefficients
  sigma2 ~ inv_gamma(0.0001, 0.0001); // Prior for error variance
  y ~ normal(x * beta, sqrt(sigma2));  // Likelihood
}
generated quantities {
    real ypred[n];  // Vector of length n for predictions
    ypred = normal_rng(x * beta, sqrt(sigma2));
}
```

Estimation Process in STAN

Let's assume Y represents the logarithm of average income by administrative division of the country.

Table 3: Logarithm of Income

dam2	logingreso	luces_nocturnas	cubrimiento_cultivo	cubrimiento_urbano
05001	13.34	46.0570	2.0996	29.9636
05002	12.41	2.3771	1.3245	0.5746
05031	12.40	0.8686	0.1123	0.2894
05034	12.62	2.9262	1.5261	0.4212
05045	12.81	5.9329	0.5239	1.2359
05079	12.21	16.1735	2.5634	3.6992
05088	13.30	36.3252	8.6041	15.4245
05093	12.54	1.1088	2.4737	0.7746
05120	11.83	0.4323	1.8486	0.1544
05129	13.59	16.9501	0.9336	3.5665

Variable Associations

Table 4: Correlation with Logarithm of Income

Covariable	logingreso
luces_nocturnas	0.6312
cubrimiento_cultivo	0.2220
cubrimiento_urbano	0.4625
modificacion_humana	0.6765
accesibilidad_hospitales	-0.3857
accesibilidad_hosp_caminado	-0.4175

Preparing the STAN Code

Organizing Data for STAN

```
fitLm2 <- "www/00_Intro_bayes/Modelo/03_ModeloLm.stan"

Xdat <- model.matrix(
  logingreso ~ luces_nocturnas +
    cubrimiento_cultivo + cubrimiento_urbano +
    modificacion_humana, data = datalm)

sample_data <- list(n = nrow(datalm),
                    K = ncol(Xdat),
                    x = as.matrix(Xdat),
                    y = datalm$logingreso)
```

Preparing Data for the STAN Code

This code section is preparing and organizing data for a Bayesian linear regression model in STAN. The data contains information about income and related variables. The model will estimate the coefficients and other parameters to describe the relationship between income and the given variables. The results are saved in an RDS file for later analysis and interpretation.

```
model_fitLm2 <-
  stan(
    file = fitLm2,
    data = sample_data,
    warmup = 500,
    iter = 1000,
    verbose = FALSE,
    cores = 4
  )

saveRDS(model_fitLm2,
          "www/00_Intro_bayes/Modelo/model_fitLm2.rds")
```

Results of the Parameter Estimation for β and σ^2

```
tabla_coef <- summary(model_fitLm2,
        pars = c("beta", "sigma2"))$summary
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_e
beta[1]	11.9030	0.0030	0.0757	11.7515	11.8538	11.9020	11.9559	12.0454	645.
beta[2]	0.0129	0.0001	0.0040	0.0054	0.0102	0.0129	0.0157	0.0209	851.
beta[3]	-0.0008	0.0000	0.0015	-0.0037	-0.0017	-0.0008	0.0002	0.0021	2522.
beta[4]	-0.0097	0.0001	0.0051	-0.0197	-0.0130	-0.0097	-0.0061	0.0004	1245.
beta[5]	1.6917	0.0104	0.2630	1.1960	1.5089	1.6849	1.8688	2.2178	638.
sigma2	0.1032	0.0002	0.0069	0.0909	0.0981	0.1028	0.1075	0.1178	964.

Convergence of Chains for Parameter θ

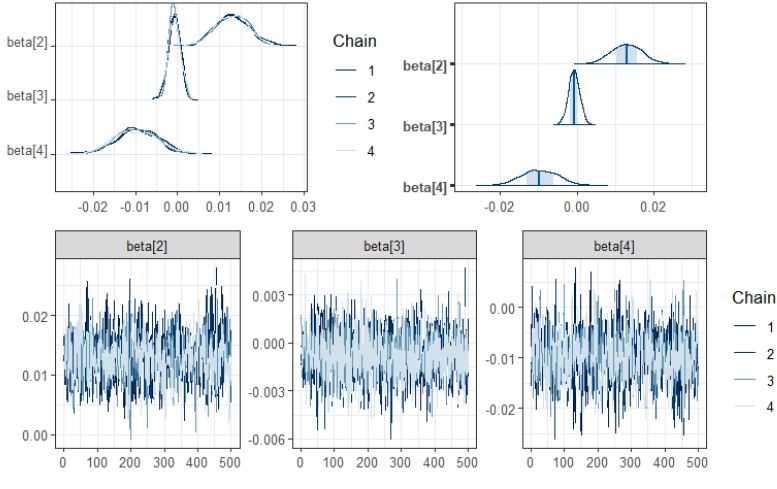


Figure 8: Chains for beta

Convergence of Chains for Parameter σ^2

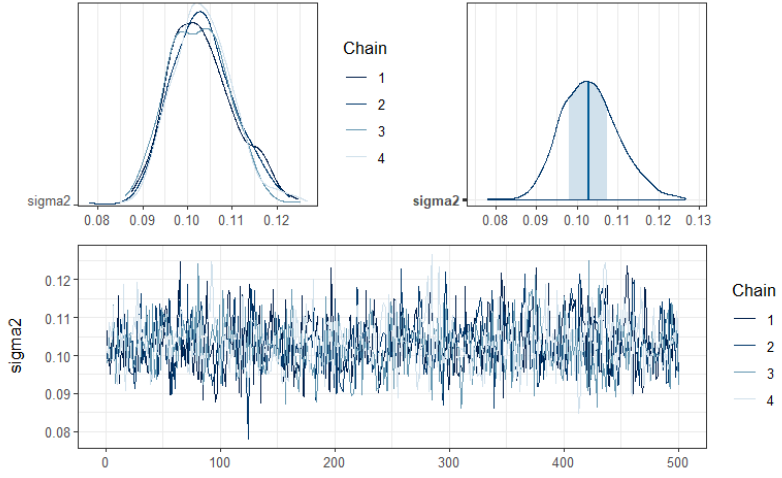


Figure 9: Chains for sigma2

Posterior Predictive Check for Income

```
y_pred_B <- as.array(model_fitLm2, pars = "ypred") %>%
  as_draws_matrix()

rowsrandom <- sample(nrow(y_pred_B), 100)

log_pred2 <- y_pred_B[rowsrandom,]
y_pred2 <- exp(log_pred2)-1

ppc_dens_overlay(
  y = datalm$logingreso, log_pred2)/
ppc_dens_overlay(
  y = as.numeric(exp(datalm$logingreso) - 1), y_pred2) +
  xlim(0, 800000)
```

Posterior Predictive Check for Income

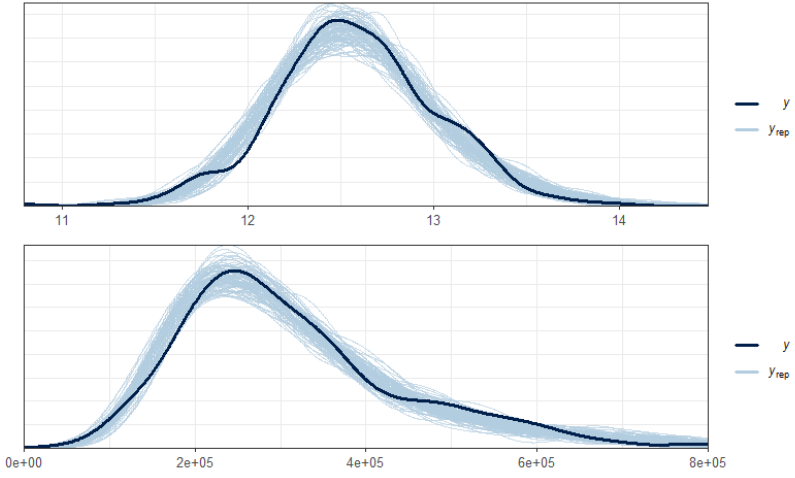


Figure 10: Posterior Predictive Check for Income

Use of SAE Methods

Justification

- Direct estimators, based solely on observed sampling units for each small area, are not reliable enough.
- Small sample size or even no observed units (lack of information).
- The coefficient of variation (CV) is too high for the target indicator at the area level.

Increase in the Coefficient of Variation

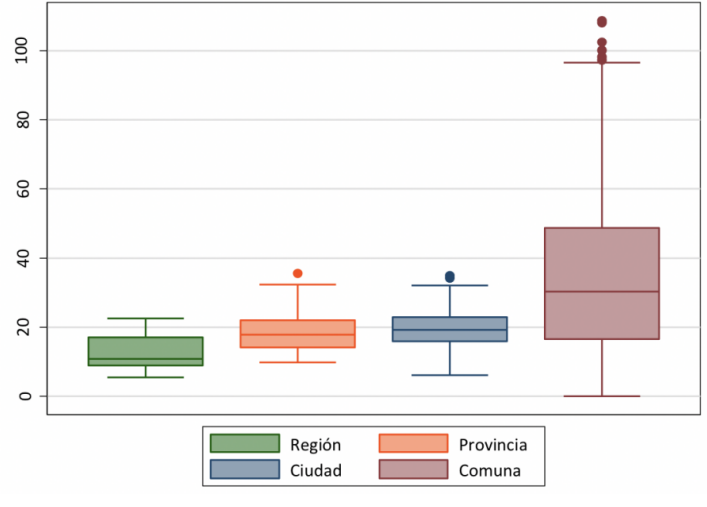


Figure 11: Distribution of coefficients of variation in Chile

Justification

When direct estimators are not reliable for some domains of interest, there are two options:

1. Oversampling: increasing the sample size in the domains of interest (increased costs).
2. Applying statistical techniques that allow reliable estimations in those domains, Small Area Estimation (SAE) methods.

What Is a Small Area?

- Most national surveys are planned to provide reliable estimates at the national and regional levels, but precision decreases at lower levels.
- A small area is a domain for which the specific sample size is not large enough to obtain reliable estimates.
- Typically, small areas are unplanned domains, and their expected sample size is random and larger as the area's population size increases.

What Is a Small Area?

The subpopulation of interest can be a geographic area or specific socioeconomic subgroups.

- Geographic: provinces, labor market areas, municipalities, census tracts, to measure, for example, the unemployment rate at the municipal level.
- Subgroups specific domains: age $\times$ gender $\times$ race within the geographic scope of an area, to measure, for example, the unemployment rate by specific gender or age in urban areas.

Some Methods

- SAE estimators are divided into two main types depending on how models are applied to the data within small areas: area level and unit level.
- Small area estimators are based on area-level calculations if the models link the target variable y with specific area-specific auxiliary variables x.# Use of SAE Methods

Justification

- Direct estimators, based solely on observed sampling units for each small area, are not reliable enough.
- Small sample size or even no observed units (lack of information).
- The coefficient of variation (CV) is too high for the target indicator at the area level.

Increase in the Coefficient of Variation

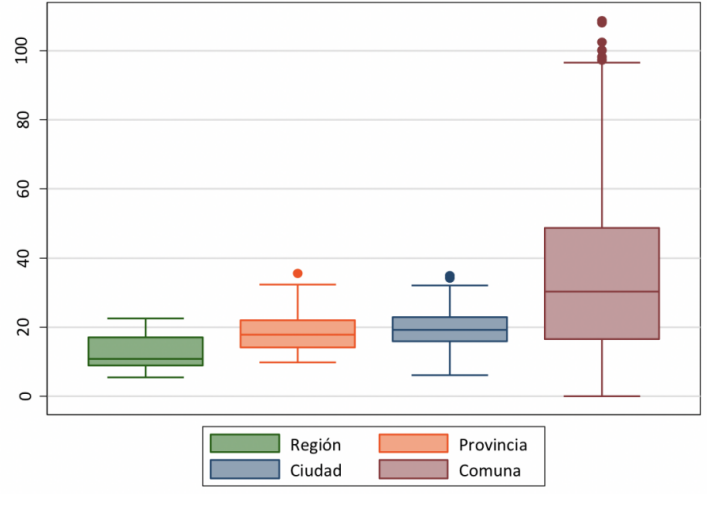


Figure 12: Distribution of coefficients of variation in Chile

Justification

When direct estimators are not reliable for some domains of interest, there are two options:

1. Oversampling: increasing the sample size in the domains of interest (increased costs).
2. Applying statistical techniques that allow reliable estimations in those domains, Small Area Estimation (SAE) methods.

What Is a Small Area?

- Most national surveys are planned to provide reliable estimates at the national and regional levels, but precision decreases at lower levels.
- A small area is a domain for which the specific sample size is not large enough to obtain reliable estimates.
- Typically, small areas are unplanned domains, and their expected sample size is random and larger as the area's population size increases.

What Is a Small Area?

The subpopulation of interest can be a geographic area or specific socioeconomic subgroups.

- Geographic: provinces, labor market areas, municipalities, census tracts, to measure, for example, the unemployment rate at the municipal level.
- Subgroups specific domains: age $\times$ gender $\times$ race within the geographic scope of an area, to measure, for example, the unemployment rate by specific gender or age in urban areas.

Some Methods

- SAE estimators are divided into two main types depending on how models are applied to the data within small areas: area level and unit level.
- Small area estimators are based on area-level calculations if the models link the target variable y with specific area-specific auxiliary variables x.

Some Methods

- They are called unit-level models if individual values for specific unit-specific auxiliary variables are linked.
- Small area estimators are calculated at the area level if unit-level data is not available.
- They can also be calculated if unit-level data is available by summarizing them at the appropriate area level.

Estimation Process

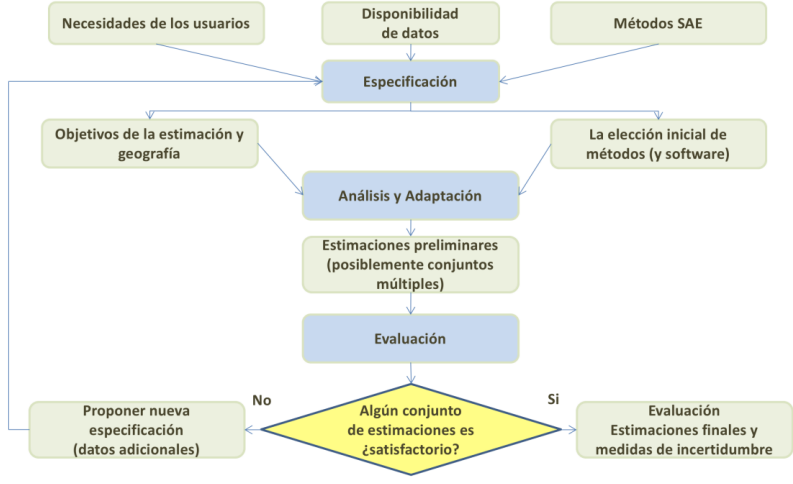


Figure 13: Producing Statistics with SAE

Considerations

- All SAE methods require small area-level auxiliary data from which they borrow strength.
- The effectiveness of SAE methods depends on the degree of association between the target variable and the auxiliary data.
- The search for good auxiliary variables is critical, including imaginative construction of such variables.
- Auxiliary data must be measured consistently across small areas but can include large sample estimates with known sampling error.

Challenges

- Increasing non-response rates.
- Rising costs, less funding.
- Increasing demand for estimates for small domains such as race, ethnicity, or poverty.
- Increasing demand for small area estimates.
- Increasing complexity in survey contents and therefore response burden.
- Increasing demand for secondary analyses, public use, and restricted-use data files.

Generalized Variance Function (GVF)

What is the importance of the Generalized Variance Function?

- The variance of the direct estimator is a crucial input in the area model.
- It is not possible to calculate the variance of the direct estimator at the domain level.
- In domains with very small sample sizes, variance estimates can be unreliable.
- The utility of a variance-smoothing model is suggested.
- The purpose of smoothing is to remove noise and volatility in variance estimates to obtain a more accurate signal of the process.

The Generalized Variance Function

Hidiroglou (2019) states that: $E_{\mathcal{MP}}\left(\hat{\theta}_d^{dir}\right) = x_d^T\beta$ and $V_{\mathcal{MP}}\left(\hat{\theta}_d^{dir}\right) = \sigma_u^2 + \tilde{\sigma}_d^2$, where the subscript $\mathcal{MP}$ refers to the double inference that must be taken into account in this type of adjustment.

- $\mathcal{M}$ refers to the probability measure induced by modeling and the inclusion of auxiliary covariates (x_d).
- $\mathcal{P}$ refers to the probability measure induced by the complex sampling design that induces direct estimates.

Estimation of Sampling Variance

The GVF involves fitting a log-linear model to the estimated direct variance. Starting from the fact that an unbiased estimator of σ^2, denoted by $\hat{\sigma}^2$, is available, we have:

$$E_{\mathcal{MP}}\left(\hat{\sigma}_d^2\right) = E_{\mathcal{M}}\left(E_{\mathcal{P}}\left(\hat{\sigma}_d^2\right)\right) = E_{\mathcal{M}}\left(\sigma_d^2\right) = \tilde{\sigma}_d^2$$

The above equality can be interpreted as an unbiased and simple estimator of $\tilde{\sigma}_d^2$ can be $\hat{\sigma}_d^2$.

Smoothing Models

Rivest and Belmonte (2000) propose smoothing models to estimate direct variances. These models are defined as follows:

$$\log\left(\hat{\sigma}_d^2\right) = z_d^T \alpha + \varepsilon_d$$

Where z_d is a vector of explanatory covariates that are functions of x_d, α is a vector of parameters that need to be estimated, and ε_d are random errors with mean zero and constant variance, which are assumed to be identically distributed conditional on z_d.

Smooth Estimation

- The smoothed estimation of the sampling variance is given by:

$$\tilde{\sigma}_d^2 = E_{\mathcal{MP}}\left(\sigma_d^2\right) = \exp\left(z_d^T \alpha\right) \times \Delta$$

Where $E_{\mathcal{MP}}\left(\varepsilon_d\right) = \Delta$.

- Using the method of moments, we have the following unbiased estimator for Δ:

$$\hat{\Delta} = \frac{\sum_{d=1}^{D} \hat{\sigma}_d^2}{\sum_{d=1}^{D} \exp\left(z_d^T \alpha\right)}$$

Parameter Estimation

- The estimation of the regression parameter coefficients is given by the following expression:

$$\hat{\alpha} = \left(\sum_{d=1}^{D} z_d z_d^T \right)^{-1} \sum_{d=1}^{D} z_d \log\left(\hat{\sigma}_d^2\right)$$

- And the smoothed estimator of the sampling variance is defined as:

$$\hat{\tilde{\sigma}}_d^2 = \exp\left(z_d^T \hat{\alpha}\right) \hat{\Delta}$$

Data: Great Integrated Household Survey (GEIH) of Colombia

The 2018 Great Integrated Household Survey (GEIH) in Colombia used a complex sampling design that included stratification of the population into urban and rural areas, along with cluster sampling. The selected sample was significant, allowing the collection of data in a representative manner throughout the country. In total, 98,000 Primary Sampling Units (UPMs) were used to obtain reliable statistics at the National, Geographic Regions, Major Cities, and Urban/Rural Areas, Socioeconomic Strata.

Data Set

Table 5: GEIH Colombia

dam	dam2	wkx	upm	estrato	pobreza
05	05360	127.2	010126005360	051	0
05	05360	127.2	010126005360	051	0
05	05360	127.2	010126005360	051	0
05	05360	127.2	010126005360	051	0
05	05360	127.2	010126005360	051	0
05	05360	125.7	010126005360	051	1
05	05360	125.7	010126005360	051	1
05	05360	125.7	010126005360	051	1
05	05360	125.7	010126005360	051	1
05	05360	115.9	010126005360	051	1

Sampling Design

To define the sampling design from a survey database, we use the `survey` and `srvyr` libraries.

```
library(survey)
library(srvyr)
options(survey.lonely.psu = "adjust")
encuesta <- readRDS("www/01_FGV/encuesta.rds")
design <-
  as_survey_design(
    ids = upm,
    weights = wkx,
    strata = estrato,
    nest = TRUE,
    .data = encuesta
  )
```

Direct Estimates by Domain

For the direct estimation of the proportion, we use the `direct.supr` function, available in the `0Source_FH.R` file. This function performs estimations and quality criteria in a complex survey sample with stratified and clustered design.

Selected Domains

- At least 50 observations per domain.
- Design effect (Deff) greater than 1.
- At least 3 degrees of freedom.

Table 6: Selected Domain Counts

Flag	n
Excluir	59
Incluir	379

FGV for Colombia's GEIH

For this process, the transformation $\log(\hat{\sigma}_d^2)$ is performed, and columns for municipality identifiers (`dam2`), direct estimation (`pobreza`), the number of people in the domain (`nd`), and estimated variance (`vardir`) are selected.

Table 7: Data Set for FGV

dam2	pobreza	nd	vardir	ln_sigma2
05001	0.1597	27432	0.0000	-10.012
05002	0.4049	257	0.0032	-5.737
05031	0.3817	199	0.0042	-5.463
05034	0.4731	223	0.0018	-6.335
05045	0.2876	480	0.0064	-5.045

Graphical Analysis

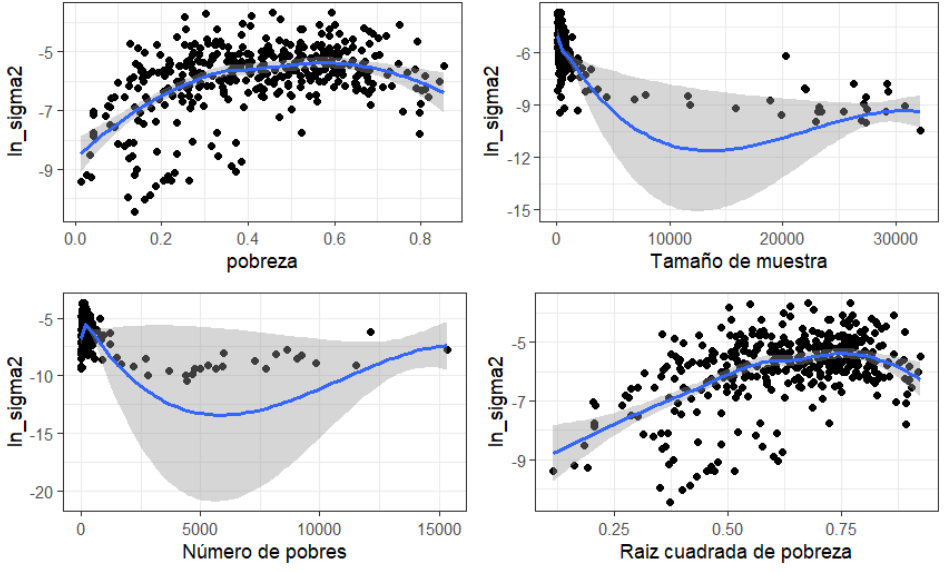


Figure 14: Scatterplots

Model for Variance

The model defined for the dataset is as follows:

$$\log(\hat{\sigma}^2) = \hat{\theta}_{dir} + n_d^2 + \sqrt{\hat{\theta}_{dir}}$$

The result of the model is shown below:

Table 8: Model Summary

Characteristic	**Beta**	**95% CI**	**p-value**
pobreza	-12	-14, -9.5	<0.001
I(nd^2)	0.00	0.00, 0.00	<0.001
I(sqrt(pobreza))	16	14, 19	<0.001
R^2	0.608		
Adjusted R^2	0.604		

Estimation of Δ and Prediction

To obtain the value of the constant Δ from the model estimation, you can use the following code:

```
delta.hat = sum(baseFGV$vardir) /
  sum(exp(fitted.values(mod_fgv)))
```

Finally, you have the smoothed variance:

```
hat.sigma <-
  data.frame(
    dam2 = baseFGV$dam2,
    hat_var = delta.hat * exp(fitted.values(mod_fgv)))
```

Results Validation

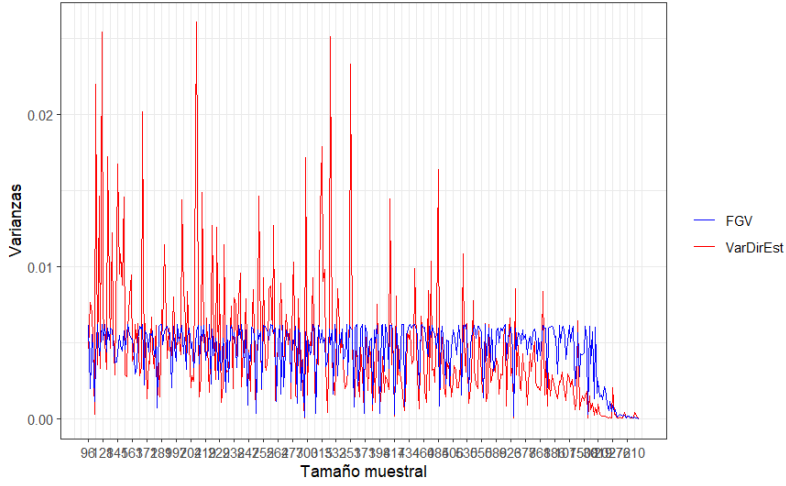


Figure 15: FGV and Direct Variance, by Sample Size

Area Models

Fay Herriot Model

- The Fay Herriot Model, proposed by Fay and Herriot in 1979, is widely used in small area estimation. This statistical approach is applied when individual-level information is limited, but data are available at the area level and auxiliary information related to this data is available.
- The model establishes a relationship between area indicators, θ_d, which vary based on a covariate vector x_d. It is formulated as $\theta_d = x_d^T \beta + u_d$, where u_d is a random effect specific to each area.

Fay Herriot Model

- The real values of the indicators θ_d are not directly observable, so we use the direct estimator $\hat{\theta}_d^{DIR}$ to estimate them, which introduces sampling error. In other words:

$$\hat{\theta}_d^{DIR} = \theta + e_d$$

- The model is adjusted to account for the sampling error e_d, and the variances $\sigma^2_{e_d}$ are estimated from the survey's microdata. This adjustment is expressed as:

$$\hat{\theta}_d^{DIR} = x_d^T \beta + u_d + e_d$$

Here, $\hat{\theta}_d^{DIR}$ represents the direct estimate of the indicator in small area d, x_d stands for area-specific auxiliary covariates, β is the vector of coefficients to be estimated, u_d is a random effect specific to each area, and e_d represents the sampling error.

Fay Herriot Model

The Best Linear Unbiased Predictor (BLUP) under the Fay Herriot model is calculated as $\tilde{\theta}_d^{FH}$. It is based on the use of γ_d to appropriately weight the direct estimator and auxiliary information, allowing for a more precise estimation of the indicators in small areas. The equation for this is as follows:

$$\tilde{\theta}_d^{FH} = x_d^T \tilde{\beta} + \tilde{u}_d$$

,

here $\tilde{u}_d = \gamma_d \left(\hat{\theta}_d^{DIR} - x_d^T \tilde{\beta} \right)$ and $\gamma_d = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_{e_d}^2}$.

Area Model for Poverty Estimation

Let P_d be the probability of finding a person in a state of poverty in the d-th domain of the population. The direct estimator of P_d can be written as:

$$\hat{P}_d^{DIR} = P_d + e_d$$

Now, P_d can be modeled as follows:

$$P_d = x_d^T \beta + u_d$$

Area Model for Poverty Estimation

Rewriting $\hat{P}_d^{DIR}$ in terms of the two equations above, we have:

$$\hat{P}_d^{DIR} = x_d^T\beta + u_d + e_d$$

Now, we can assume that:

- $\hat{P}_d^{DIR} \sim N(x_d^T\beta, \sigma_u^2 + \sigma_{e_d}^2)$,
- $\hat{P}_d^{DIR} \mid u_d \sim N(x_d^T\beta + u_d, \sigma_{e_d}^2)$ and
- $u_d \sim N(0, \sigma_u^2)$

Prior Distributions

The prior distributions for β and σ_u^2 are as follows:

$$\beta_p \sim N(0, 10000)$$

$$\sigma_u^2 \sim Inverse - Gamma(0.0001, 0.0001)$$

Therefore, the Bayesian estimator for P_d is given as $\tilde{P}_d = E\left(P_d \mid \hat{P}_d^{DIR}\right)$

Procedure for Estimating Poverty in Colombian Municipalities

The available covariables are shown in the following table, which has been obtained previously.

Table 9: Available Covariables

dam	dam2	area1	sexo2	edad2	edad3	edad4	edad5
05	05001	0.9832	0.5299	0.2671	0.2201	0.2355	0.1060
05	05002	0.3953	0.4807	0.2229	0.1977	0.2497	0.1281
05	05004	0.3279	0.4576	0.2376	0.2075	0.2316	0.1218
05	05021	0.5770	0.5020	0.2191	0.1946	0.2357	0.1274
05	05030	0.4859	0.5063	0.2571	0.2047	0.2507	0.0997

FH Model: STAN Routine

```
data {
  int<lower=0> N1; // number of data items
  int<lower=0> N2; // number of data items for prediction
  int<lower=0> p;  // number of predictors
  matrix[N1, p] X; // predictor matrix
  matrix[N2, p] Xs; // predictor matrix
  vector[N1] y;    // predictor matrix
  vector[N1] sigma_e; // known variances
}

parameters {
  vector[p] beta;       // coefficients for predictors
  real<lower=0> sigma2_u;
  vector[N1] u;
}
```

FH Model: STAN Routine

```
transformed parameters{
  vector[N1] theta;
  vector[N1] thetaSyn;
  vector[N1] thetaFH;
  vector[N1] gammaj;
  real<lower=0> sigma_u;
  thetaSyn = X * beta;
  theta = thetaSyn + u;
  sigma_u = sqrt(sigma2_u);
  gammaj =  to_vector(sigma_u ./ (sigma_u + sigma_e));
  thetaFH = (gammaj) .* y + (1-gammaj).*thetaSyn;
}
```

FH Model: STAN Routine

```
model {
  // likelihood
  y ~ normal(theta, sigma_e);
  // priors
  beta ~ normal(0, 100);
  u ~ normal(0, sigma_u);
  sigma2_u ~ inv_gamma(0.0001, 0.0001);
}

generated quantities{
  vector[N2] y_pred;
  for(j in 1:N2) {
    y_pred[j] = normal_rng(Xs[j] * beta, sigma_u);
  }
}
```

Preparing Inputs for STAN

- Defining the area model.

```
formula_mod  <- formula(
  ~ gender2 + year_est2 + year_est3 +
    year_est4 + age2 + age3  +  age4  + age5 + ethnicity1 +
    ethnicity2 + unemployment_rate + nighttime_lights +
    crop_coverage + albedo
)
```

Preparing Inputs for STAN

- Split the database into observed and unobserved domains.

```
# Observed domains.
data_dir <- base_FH %>% filter(!is.na(poverty))

Xdat <- model.matrix(formula_mod, data = data_dir)

# Unobserved domains.
data_syn <-
  base_FH %>% anti_join(data_dir %>% select(dam2))

Xs <- model.matrix(formula_mod, data = data_syn)
```

Preparing Inputs for STAN

- Creating a parameter list for STAN.

```
sample_data <- list(
  N1 = nrow(Xdat),   # Observed.
  N2 = nrow(Xs),   # Unobserved.
  p  = ncol(Xdat),       # Number of predictors.
  X  = as.matrix(Xdat),  # Observed Covariates.
  Xs = as.matrix(Xs),    # Unobserved Covariates
  y  = as.numeric(data_dir$poverty), # Direct estimation
  sigma_e = sqrt(data_dir$hat_var)   # Estimation error
)
```

Compiling the Model in STAN

Here's how to compile the STAN code from R.

```
library(rstan)
fit_FH_normal <- "www/02_FH_Nornal/17FH_normal.stan"
options(mc.cores = parallel::detectCores())
model_FH_normal <- stan(
  file = fit_FH_normal,
  data = sample_data,
  verbose = FALSE,
  warmup = 500,
  iter = 1000,
  cores = 4
)
saveRDS(object = model_FH_normal,
        file = "www/02_FH_Nornal/model_FH_normal.rds")
```

Results of the Model for Observed Domains

Using the `ppc_dens_overlay()` function to plot a comparison between the empirical distribution of the observed poverty variable in the data and the simulated posterior predictive distributions for the same variable.

```
y_pred_B <- as.array(model_FH_normal,
                     pars = "theta") %>%
  as_draws_matrix()

rowsrandom <- sample(nrow(y_pred_B), 100)

y_pred2 <- y_pred_B[rowsrandom,]

ppc_dens_overlay(y = as.numeric(data_dir$poverty),
                 y_pred2)
```

Posterior Predictive Check

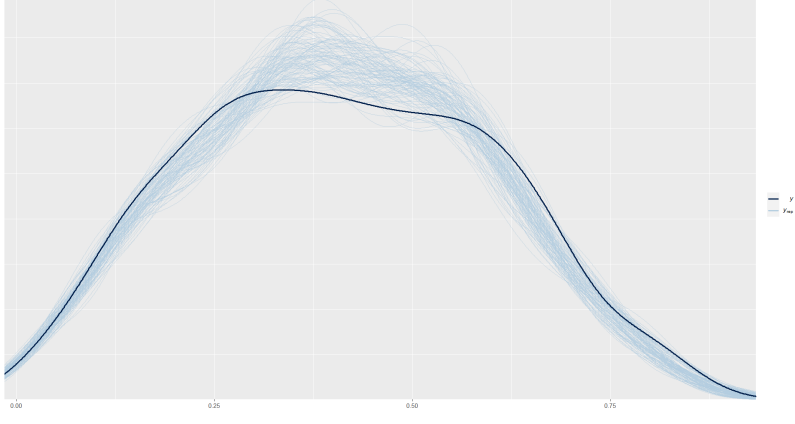


Figure 16: PPC for poverty

Validation of Chains Convergence σ^2

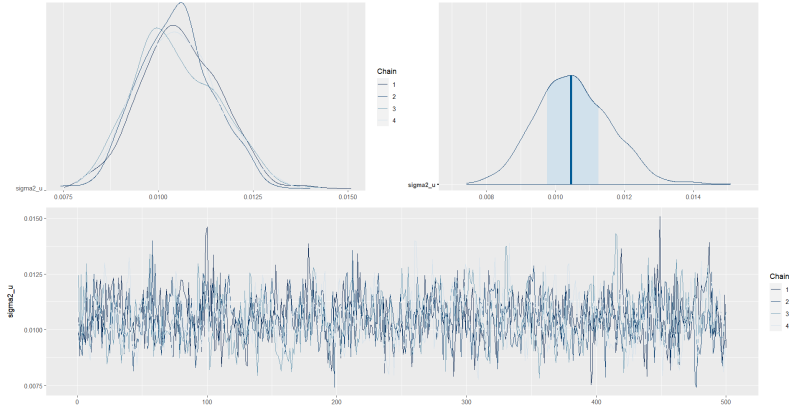


Figure 17: Chain Convergence

Comparison of Estimates

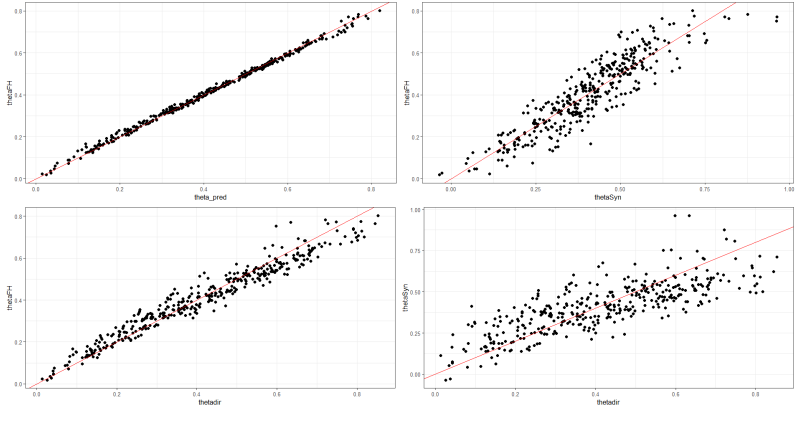


Figure 18: Comparison between the model equations and the gamma weights

Benchmarking Process

- Extract the total number of people by DAM2 from the census.

dam	dam2	total_pp	dam_pp
05	05001	2372330	44164417
05	05002	17599	44164417
05	05004	2159	44164417
05	05021	3839	44164417
05	05030	26821	44164417
05	05031	20265	44164417
05	05034	38144	44164417
05	05036	5027	44164417
05	05038	10500	44164417
05	05040	14502	44164417

Direct Estimation

Obtain direct estimates by DAM or the level of aggregation for which the survey is representative.

```
directoDam <- diseno %>%
   group_by(Aggregated = "National") %>%
  summarise(
    theta_dir = survey_mean(poverty, vartype = c("ci"))
    )
```

Agregado	theta_dir	theta_dir_low	theta_dir_upp
Nacional	0.2986	0.2935	0.3038

Calculation of Weights

After organizing the above information, the weights for benchmarking are calculated.

```
estimationsPre <-
  readRDS("www/02_FH_Nornal/05_tabla_estimacionesPre.rds")
temp <- estimationsPre %>%
  inner_join(N_dam_pp) %>%
  mutate(theta_dir = directoDam$theta_dir)
R_dam2 <- temp %>%
  summarise(
    R_dam_RB = unique(theta_dir) /
      sum((total_pp  / dam_pp) * theta_pred))
```

R_dam_RB
1.016

Estimation with the area model after Benchmarking

```
weights <- temp %>%
  mutate(W_i = total_pp / dam_pp) %>%
  select(dam2, W_i)

estimationsBench <- estimationsPre %>%
  mutate(R_dam_RB = R_dam2$R_dam_RB) %>%
  mutate(theta_pred_RBench = R_dam_RB * theta_pred) %>%
  select(dam, dam2, theta_pred, theta_pred_RBench)
```

dam	dam2	W_i	theta_pred	theta_pred_RBench
05	05001	0.0537	0.1593	0.1618
05	05002	0.0004	0.4130	0.4194
05	05031	0.0005	0.4121	0.4185

Validation of the Results

This code combines the model estimates with benchmarking weights with observed and synthetic values and then summarizes the combined estimates to compare them with the direct estimation obtained earlier.

```
temp <- estimationsBench %>%
  left_join(estimationsPre) %>%
  summarise(
    thetaSyn = sum(W_i * thetaSyn),
    thetaFH = sum(W_i * theta_pred),
    theta_RBench = sum(W_i * theta_pred_RBench)
  ) %>%
  mutate(
    theta_dir = directoDam$theta_dir,
    theta_dir_low = directoDam$theta_dir_low,
    theta_dir_upp = directoDam$theta_dir_upp
  )
```

Validation Result

Table 10: Comparison of Estimations

theta_dir_low	theta_dir_upp	Metodo	Estimacion
0.2935	0.3038	thetaSyn	0.2955
0.2935	0.3038	thetaFH	0.2941
0.2935	0.3038	theta_RBench	0.2986
0.2935	0.3038	theta_dir	0.2986

Result of the Area Model Estimation

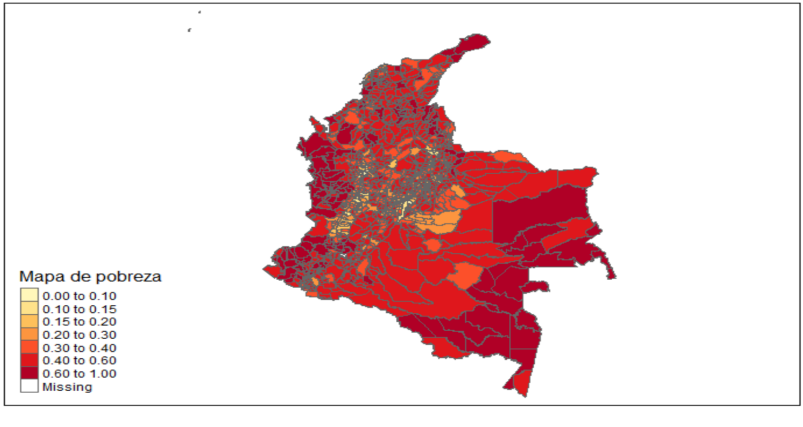


Figure 19: Poverty Map

Area Model: Arcsine Transformation

- In the Fay-Herriot model, the linear combination of covariates can generate values that are outside the acceptable range for a proportion.
- To address this, an arcsine transformation is applied to the estimators: $\hat{z}_d = \arcsin\left(\sqrt{\hat{\theta}_d}\right)$.
- The variance of the arcsine transformation is related to the design effect factor (DEFF) and the effective sample size:

$$Var\left(\hat{z}_d\right) = \frac{\widehat{DEFF}_d}{4 \times n_d} = \frac{1}{4 \times n_{d,efective}}$$

.

Specification of the Fay-Herriot Model

- The Fay-Herriot model is defined with a latent variable Z_d that follows a normal distribution.
- The mean of Z_d (μ_d) is related to the covariates through $x_d^T\beta + u_d$.
- The relationship between the latent variable θ_d and the direct estimator is established as $\theta_d = (\sin(\mu_d))^2$.

This can be simplified as:

$$\begin{aligned} Z_d \mid \mu_d, \sigma_d^2 &\sim N(\mu_d, \sigma_d^2) \\ \mu_d &= x_d^T\beta + u_d \\ \theta_d &= (\sin(\mu_d))^2 \end{aligned}$$

Prior Distributions

Prior distributions are specified for the model parameters:

- $\beta \sim N\left(0, 1000\right)$
- $\sigma_u^2 \sim Inverse - Gamma\left(0.0001, 0.0001\right).$

Area Model: STAN Routine

The code is similar to the previous one, with the following variations:

```
transformed parameters{
  vector[N1] theta;
  vector[N1] lp;
  real<lower=0> sigma_u;
  lp = X * beta + u;
  sigma_u = sqrt(sigma2_u);
  for(k in 1:N1){
    theta[k] = pow(sin(lp[k]), 2);
  }
}
```

```
model {
  // likelihood
  y ~ normal(lp, sigma_e);
  // priors
  beta ~ normal(0, 100);
  u ~ normal(0, sigma_u);
  sigma2_u ~ inv_gamma(0.0001, 0.0001);
}
```

Estimation Procedure

For the previously prepared dataset, you need to select and transform the columns of interest.

```
statelevel_predictors_df <-
  readRDS("www/03_FH_Arcsin/statelevel_predictors.rds")
base_FH <- readRDS("www/03_FH_Arcsin/base_FH_2018.rds") %>%
  transmute(
    dam2,                              # domain IDs
    pobreza,
    T_pobreza = asin(sqrt(pobreza)),   # creating zd
    n_effec = n_eff_FGV,               # effective sample size
    varhat = 1/(4*n_effec)             # variance for zd
    )
```

Preparing Inputs for STAN

Selecting the covariates, which correspond to the previously selected ones.

```
base_FH <- full_join(base_FH,
           statelevel_predictors_df, by = "dam2" )

names_cov <- c(
  "sexo2" , "anoest2" , "anoest3",   "anoest4",
  "edad2" , "edad3" , "edad4" , "edad5" , "etnia1",
  "etnia2" ,  "tasa_desocupacion" , "luces_nocturnas" ,
  "cubrimiento_cultivo" , "alfabeta"
)
```

Dividing the Dataset

The estimation and prediction process is done separately within STAN

- Observed domains.

```
data_dir <- base_FH %>% filter(!is.na(T_pobreza))
Xdat <- cbind(inter = 1, data_dir[, names_cov])
```

- Unobserved domains.

```
data_syn <-
  base_FH %>% anti_join(data_dir %>% select(dam2))
Xs <- cbind(inter = 1, data_syn[, names_cov])
```

Parameter List for STAN

STAN's processing engine is based on C++, which is why the arguments to run the code need to be entered as a list.

```
sample_data <- list(
  N1 = nrow(Xdat),        # Observed.
  N2 = nrow(Xs),          # Unobserved.
  p  = ncol(Xdat),        # Number of regressors.
  X  = as.matrix(Xdat),   # Observed Covariates.
  Xs = as.matrix(Xs),     # Unobserved Covariates.
  y  = as.numeric(data_dir$T_pobreza),
  sigma_e = sqrt(data_dir$varhat)
)
```

Compiling the Model in STAN

```
fit_FH_arcoseno <-
  "www/03_FH_Arcsin/15FH_arcsin_normal.stan"

model_FH_arcoseno <- stan(
  file = fit_FH_arcoseno,
  data = sample_data,
  verbose = FALSE,
  warmup = 500,
  iter = 1000,
  cores = 4
)
saveRDS(model_FH_arcoseno,
        "www/03_FH_Arcsin/model_FH_arcoseno.rds")
```

Results for Observed Domains

Similar to the Fay Herriot model, we perform the graph with posterior predictive checking.

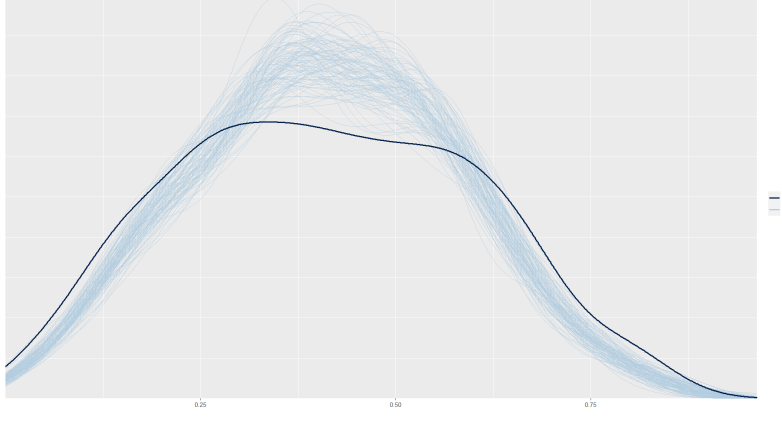


Figure 20: PPC for the Arcsine Area Model

Graphical Analysis of Chain Convergence for σ_u^2

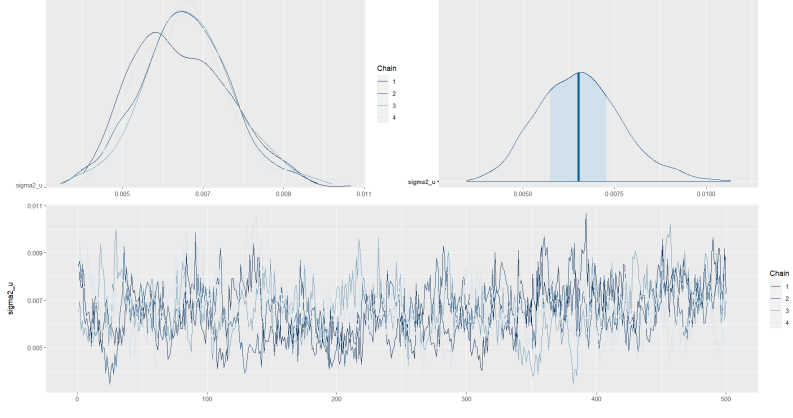


Figure 21: Chain Trace

Poverty Map with Arcsine Transformation

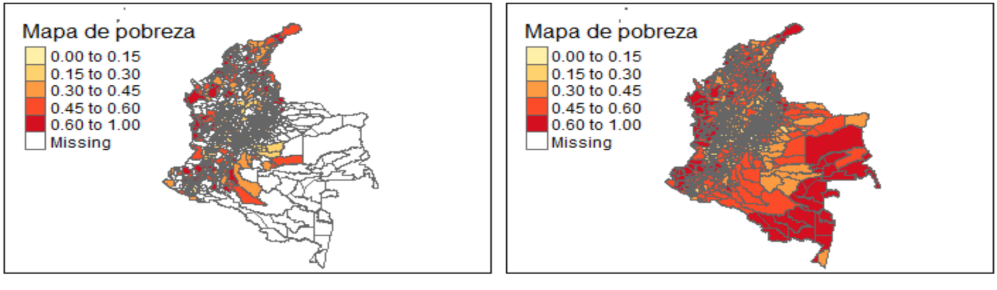


Figure 22: Poverty Map with Arcsine Transformation

Map of Poverty Coefficients of Variation

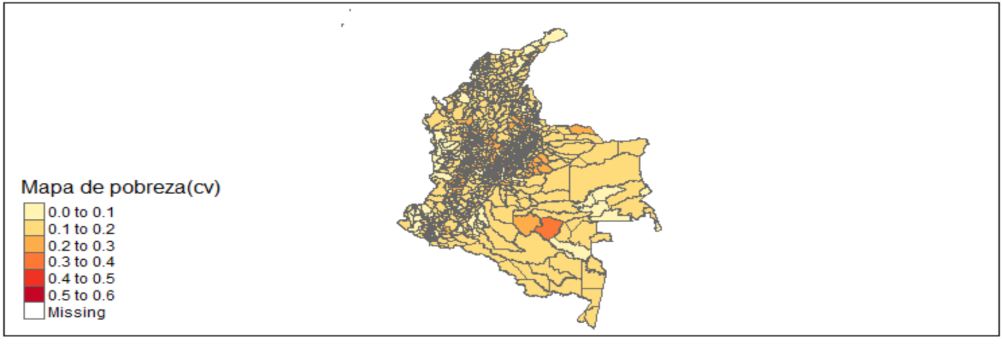


Figure 23: Map of Coefficients of Variation

Area Models with Beta Response Variable

The beta-logistic model was initially introduced in the context of an Empirical Best Prediction (EBP) approach by Jiang and Lahiri in 2006. It was used to estimate domain means in finite populations.

- The area beta-logistic model is defined through the following expression:

$$\hat{p}_d \mid P_d \sim beta(a_d, b_d)$$

.

- The link function is related to the model parameters:

$$logit(P_d) \mid \beta, \sigma_u^2 \sim N(x_d^T \beta, \sigma_u^2)$$

.

Parameter Estimation

- The parameters a_d and b_d are estimated as follows:

$$\begin{aligned} a_d &= P_d \times \phi_d \\ b_d &= (1 - P_d) \times \phi_d \end{aligned}$$

Where $\phi_d = \frac{n_d}{\widehat{DEFF}_d} - 1 = n_{d,efectivo} - 1$.

- Prior distributions are specified for the model parameters:

$$\beta_k \sim N(0, 10000)$$

$$\sigma_u^2 \sim Inverse - Gamma(0.0001, 0.0001)$$

.

Area Model: STAN Routine

In this code block, we can see the transformation performed on the input parameters.

```
transformed parameters{
  vector[N1] LP;
  real<lower=0> sigma_u;
  vector[N1] theta;
  LP = X * beta + u;
  sigma_u = sqrt(sigma2_u);
  for (i in 1:N1) {
    theta[i] = inv_logit(LP[i]);
  }
}
```

FH Model: STAN Routine

```
model {
  // model calculations
  vector[N1] a;
  vector[N1] b;

  for (i in 1:N1) {
    a[i] = theta[i] * phi[i];
    b[i] = (1 - theta[i]) * phi[i];
  }

  // priors
  beta ~ normal(0, 100);
  u ~ normal(0, sigma_u);
  sigma2_u ~ inv_gamma(0.0001, 0.0001);

  // likelihood
  y ~ beta(a, b);
}
```

Estimation Procedure

Similar to the previous models, we use the base that was prepared in advance.

```
base_FH <-
readRDS("www/04_FH_Beta_y_Binomial/base_FH_2018.rds") %>%
  select(dam2, pobreza, n_eff_FGV)

base_FH <- full_join(base_FH,
             statelevel_predictors_df, by = "dam2")
```

The covariates are the same as those used in the previous models.

Splitting the Dataset

The estimation and prediction processes are done separately within STAN.

- Observed domains.

```
data_dir <- base_FH %>% filter(!is.na(T_pobreza))
Xdat <- cbind(inter = 1, data_dir[,names_cov])
```

- Unobserved domains.

```
data_syn <-
  base_FH %>% anti_join(data_dir %>% select(dam2))
Xs <-  cbind(inter = 1, data_syn[,names_cov])
```

List of Parameters for STAN

```
sample_data <- list(
  N1 = nrow(Xdat),   # Observed.
  N2 = nrow(Xs),   # Unobserved.
  p  = ncol(Xdat),       # Number of predictors.
  X  = as.matrix(Xdat),  # Observed covariates.
  Xs = as.matrix(Xs),    # Unobserved covariates
  y  = as.numeric(data_dir$pobreza),
  phi = data_dir$n_eff_FGV - 1
)
```

Compiling the Model in STAN

```
fit_FH_beta_logitic <-
  "www/04_FH_Beta_y_Binomial/16FH_beta_logitc.stan"

model_FH_beta_logitic <- stan(
  file = fit_FH_beta_logitic,
  data = sample_data,
  verbose = FALSE,
  warmup = 500,
  iter = 1000,
  cores = 4
)
saveRDS(model_FH_beta_logitic,
file = "www/04_FH_Beta_y_Binomial/model_FH_beta.rds")
```

Results of the Model for Observed Domains

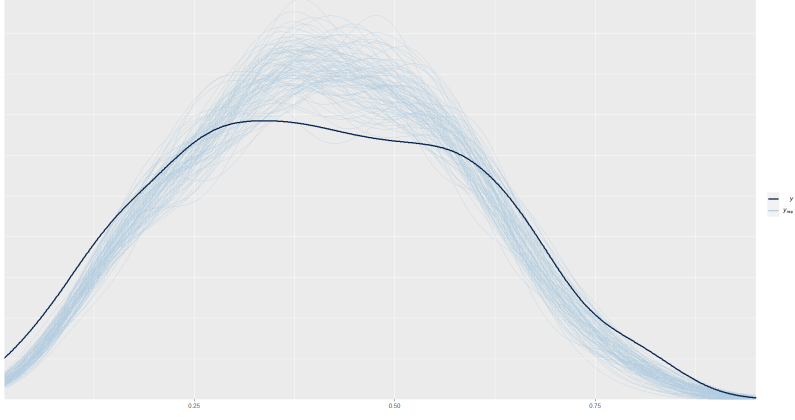


Figure 24: PPC Beta Response Area Model

Graphical Analysis of Chain Convergence for σ_u^2

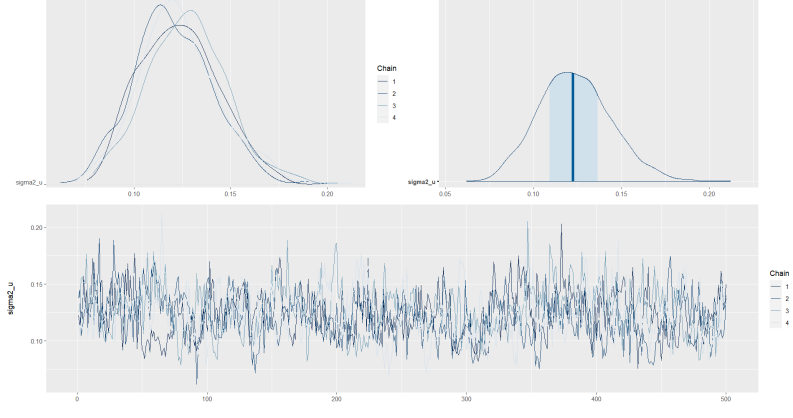


Figure 25: Chain Trace

Poverty Map with Beta Response Area Model

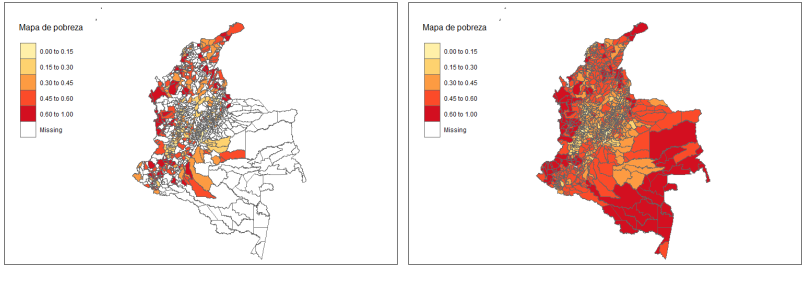


Figure 26: Poverty Map with Beta Response Area Model

Map of Coefficients of Variation for Poverty

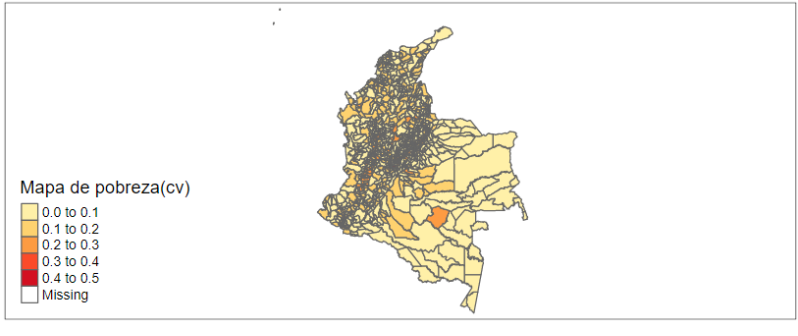


Figure 27: Coefficient of Variation Map

Area Models with Binomial Response Variable

- The Fay-Herriot area model can be substituted by a Generalized Linear Mixed Model (GLMM) when the observed data is inherently discrete, such as counts of individuals or households with certain characteristics.
- In a GLMM, a binomial distribution is assumed for the data Y_d with a success probability θ_d. A logistic model is used for θ_d with normally distributed errors on the logit scale.

Model Formulation

The model is formulated as follows:

$$\begin{aligned} Y_d \mid \theta_d, n_d &\sim Binomial(n_d, \theta_d) \\ logit\,(\theta_d) &= \log\left(\frac{\theta_d}{1-\theta_d}\right) = x_d^T\beta + u_d \end{aligned}$$

Where u_d follows a normal distribution with mean zero and variance σ_u^2, and n_d represents the sample size for area d.

Considerations for the Model

For complex samples, two problems arise:

- The values of Y_d are not integers and are affected by survey weights.
- The sample variance in the binomial distribution is not accurate.

Proposal by Carolina Franco

- An **effective sample size** $\tilde{n}_d$ and an **effective sample size of successes** $\tilde{Y}_d$ are introduced to address these issues and maintain the direct estimation of poverty and its corresponding variance.
- Given this, it is possible to assume that

$$\tilde{n}_d \sim \frac{\breve{\theta}_d \left(1 - \breve{\theta}_d\right)}{\widehat{Var}\left(\hat{\theta}_d\right)}$$

  where $\breve{\theta}_d$ is a preliminary prediction based on the model for the population proportion, $\hat{\theta}_i$ is the direct estimation, and $\widehat{Var}(\hat{\theta}_d)$ is the estimation of the sampling variance.
- Then, it is assumed that $\tilde{n}_d$ is proportional to the adjusted variance, and $\tilde{Y}_d = \tilde{n}_d \times \hat{\theta}_d$.

Prior Distributions

Prior distributions for the parameters β and σ_u^2 are specified as follows:

$$\beta \sim N(0, 10000)$$

$$\sigma_u^2 \sim Inverse - Gamma(0.0001, 0.0001)$$

Area Model: STAN Routine

In this code block, we see the transformation applied to the input parameters.

```
transformed parameters {
   vector[N1] LP;
   vector[N1] theta;
   real<lower=0> sigma_u;

   sigma_u = sqrt(sigma2_u);
   LP = X * beta + u;
   theta = inv_logit(LP);
}
```

FH Model: STAN Routine

```
model {
  to_vector(beta) ~ normal(0, 10000);
  u ~ normal(0, sigma_u);
  sigma2_u ~ cauchy(0, 1000);
  for(ii in 1:N1){
    y_effect[ii] ~ binomial(n_effec[ii], theta[ii]);
  }
}

generated quantities {
  real ypred[N2];
  vector[N2] thetaLP;
  vector[N2] LP_pred;
  LP_pred = Xs * beta;
  thetaLP = inv_logit(LP_pred);
}
```

Estimation Procedure

Reading the database with direct estimates.

```
base_FH <-
readRDS("www/04_FH_Beta_y_Binomial/base_FH_2018.rds") %>%
  select(dam2, pobreza, n_eff_FGV)

base_FH <- full_join(base_FH,
             statelevel_predictors_df, by = "dam2")
```

The covariates are the same as those used in previous models.

Data Splitting

The estimation and prediction process is performed separately within STAN.

- Observed domains.

```
data_dir <- base_FH %>% filter(!is.na(T_pobreza))
Xdat <- cbind(inter = 1,data_dir[,names_cov])
```

- Unobserved domains.

```
data_syn <-
  base_FH %>% anti_join(data_dir %>% select(dam2))
Xs <-  cbind(inter = 1,data_syn[,names_cov])
```

Obtaining Additional Parameters

- Effective sample size $\tilde{n}_d$

```
n_effec = round(data_dir$n_eff_FGV)
```

- Effective number of successful samples $\tilde{Y}_d$

```
y_effect = round((data_dir$pobreza) * n_effec)
```

List of Parameters for STAN

```
sample_data <- list(
  N1 = nrow(Xdat),   # Observed.
  N2 = nrow(Xs),     # Unobserved.
  p = ncol(Xdat),    # Number of predictors.
  X = as.matrix(Xdat),  # Observed Covariates.
  Xs = as.matrix(Xs),    # Unobserved Covariates.
  n_effec = n_effec,
  y_effect = y_effect  # Direct estimation.
)
```

Compiling the Model in STAN

```
fit_FH_binomial <- "www/04_FH_Beta_y_Binomial/14FH_binomial.stan"

model_FH_Binomial <- stan(
  file = fit_FH_binomial,
  data = sample_data,
  verbose = FALSE,
  warmup = 500,
  iter = 1000,
  cores = 4
)

saveRDS(model_FH_Binomial,
file = "www/04_FH_Beta_y_Binomial/model_FH_Binomial.rds")
```

Results of the Model for Observed Domains

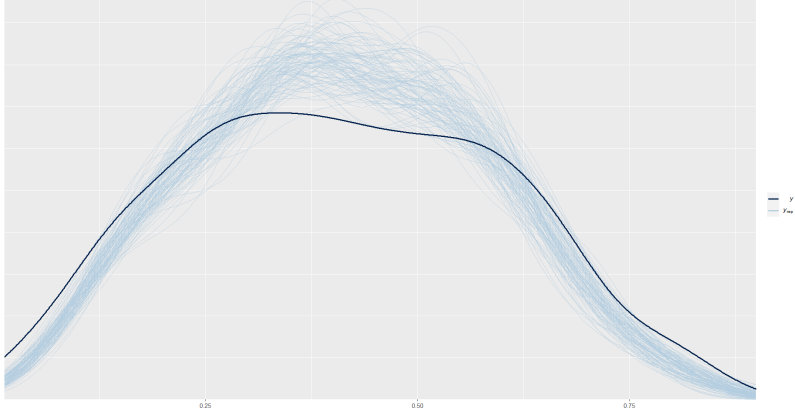


Figure 28: PPC for the Binomial Area Model

Graphic Analysis of the Convergence of σ_u^2 Chains

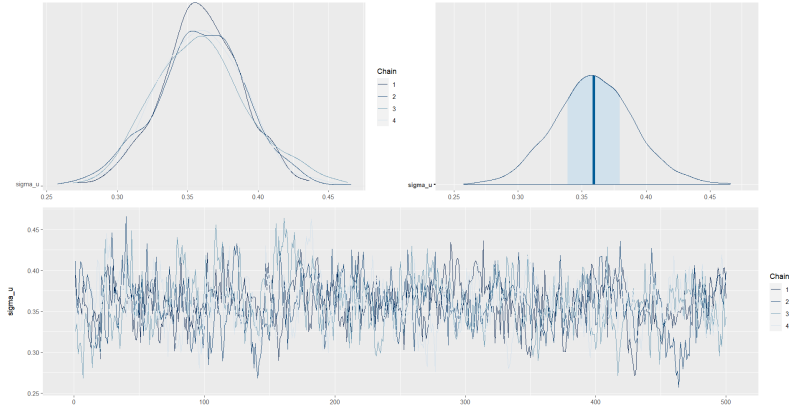


Figure 29: Chains Trace

Poverty Map with Binomial Area Response Model

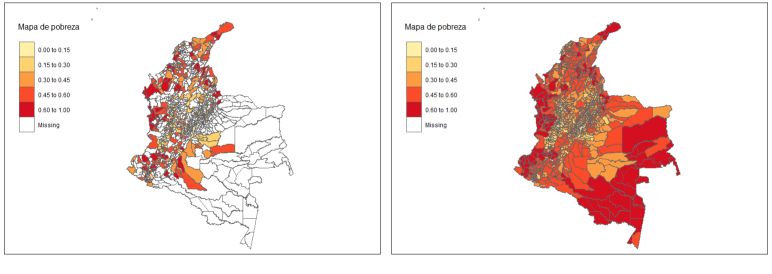


Figure 30: Poverty Map with Binomial Area Response Model

Coefficient of Variation Map for Poverty

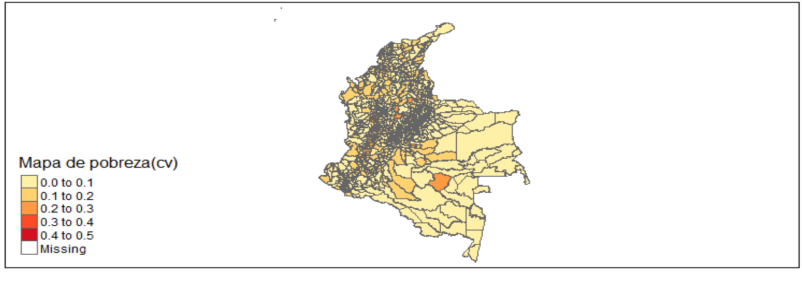


Figure 31: Coefficient of Variation Map

Unit Models

Unit Model for Mean Income Estimation

- This methodology, known as "pseudo-EBP," is a nested error model that incorporates survey expansion factors. This model is based on the concept of best empirical predictor, incorporating information from population census microdata.
- Unlike other nested error models by Battese, Harter, and Fuller (BHF), it does not require prior knowledge or estimation of the model's residual variance. This makes the methodology more accessible and practical.

Higher Level of Disaggregation in Estimates

- Under certain conditions, these models allow for a higher level of disaggregation in estimates. This means that we can generate estimates at the municipal, provincial, or communal level, broken down by various characteristics, such as ethnic self-identification, age group, gender, disability, and others, if individual-level covariates are available.

Unit Model Estimation Method

To estimate the mean income of individuals, i.e.,

$$\bar{Y}_d = \frac{\sum_{U_d} y_{di}}{N_d}$$

where y_{di} is the income of each person. Note that,

$$\bar{Y}_d = \frac{\sum_{s_d} y_{di} + \sum_{s_d^c} y_{di}}{N_d}$$

Unit Model Prediction

The estimator for $\bar{Y}$ is given by:

$$\hat{\bar{Y}}_d = \frac{\sum_{s_d} y_{di} + \sum_{s_d^c} \hat{y}_{di}}{N_d}$$

where

$$\hat{y}_{di} = E_{\mathcal{M}}\left(y_{di} \mid x_d, \beta\right)$$

,

where $\mathcal{M}$ refers to the probability measure induced by the model. Thus, it holds that:

$$\hat{\bar{Y}}_d = \frac{\sum_{U_d} \hat{y}_{di}}{N_d}$$

Unit Model Definition

- We are applying a Bayesian model to predict the mean income in unobserved areas. This is based on the assumption that the mean incomes Y_{di} follow a normal distribution with a mean μ_{di} and variance σ_e^2.
- The mean μ_{di} is related to individual characteristics X through a set of parameters β, along with a domain-specific effect u_d and an estimation error term e_{di}.
- The model:

$$\begin{aligned} Y_{di} &\sim N\left(\mu_{di}, \sigma_e^2\right) \\ \mu_{di} &= x_{di}^T \beta + u_d + e_{di} \end{aligned}$$

Unit Model Definition

- Both u_d and e_{di} follow normal distributions with means of zero and variances σ_u^2 and σ_e^2, respectively.
- We have set non-informative prior distributions for the parameters β_k and σ_y^2. This means that we assume we have little prior information about these parameters and therefore do not assign specific prior distributions to them.

$$\begin{aligned} \beta_k &\sim N(0, 1000) \\ \sigma_y^2 &\sim Inverse - Gamma(0.0001, 0.0001) \end{aligned}$$

Reading Libraries and R Functions

- *plot_interaction*: This function creates a line plot to study the interaction between variables. If there is an overlap of lines, it is recommended to include the interaction in the model.
- *Aux_Agregado*: This function allows obtaining estimates at different levels of aggregation, which becomes relevant when a repetitive process is performed.

```
library(rstan)
library(rstanarm)
source("www/05_Mod_Ingreso/01_funtions.R")
```

These functions are specifically designed for this process.

Standardized Household Surveys

The original database is recoded as follows:

- Years of education (**anoest**) are recoded as:
  - 1 → No education
  - 2 → 1 - 6 years
  - 3 → 7 - 12 years
  - 4 → More than 12 years
  - 98 → Not applicable
  - 99 → DK/NA (Don't know/No answer)
- **Sex** is recoded as:
  - 1 → Male
  - 2 → Female
- Ethnic self-recognition (**etnia**) is recoded as:
  - 1 → Indigenous
  - 2 → Afro-descendant
  - 3 → Other

Standardized Household Surveys

- **Age** is recoded as:
  - 1 $\rightarrow$ 0 - 14
  - 2 $\rightarrow$ 15 - 29
  - 3 $\rightarrow$ 30 - 44
  - 4 $\rightarrow$ 45 - 64
  - 5 $\rightarrow$ 65 and older
- Urban/rural area (**área**) is recoded as:
  - 0 $\rightarrow$ Rural
  - 1 $\rightarrow$ Urban
- $logingreso = \log(ingreso)$

Survey Dataset

```
encuesta_mrp <-
  readRDS("www/05_Mod_Ingreso/encuesta_estan.rds")
```

Table 11: Standardized Survey

dam2	area	logingreso	sexo	anoest	edad	etnia
05360	1	13.27	1	3	3	2
05360	1	13.27	2	2	3	3
05360	1	13.27	1	2	1	2
05360	1	13.27	1	98	1	2
05360	1	13.27	1	98	1	2
05360	1	12.42	1	4	3	2
05360	1	12.42	2	3	3	2
05360	1	12.42	2	2	1	2
05360	1	12.42	1	1	1	2
05360	1	11.99	1	2	4	2

Smoothed Income Histogram

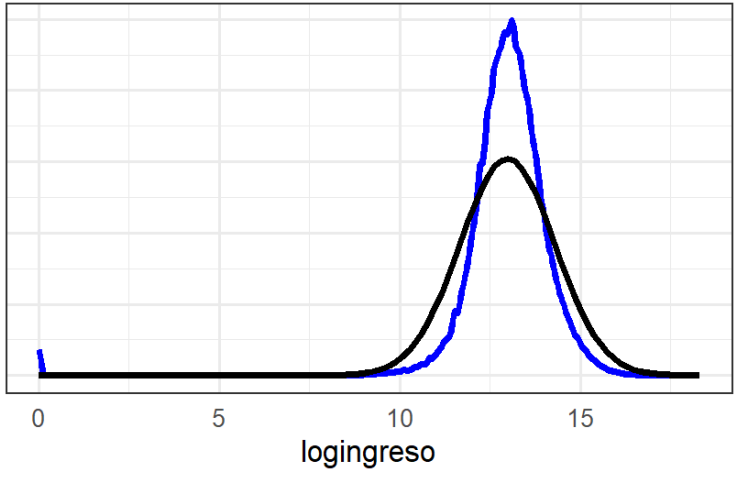


Figure 32: Black line: normal distribution, Blue line: smoothed income

Creating the Aggregated Survey Database

The result of aggregating the database is shown below:

```
byAgrega <- c("dam", "dam2",  "área", "sexo",
              "anoest", "edad",   "etnia")

encuesta_df_agg <-
  encuesta_mrp %>%
  group_by_at(all_of(byAgrega)) %>%
  summarise(n = n(),
            logingreso = mean(logingreso),
            .groups = "drop")
```

Aggregated Survey

Computational processes are optimized when working with aggregated surveys.

Table 12: Aggregated Survey

dam2	area	sexo	anoest	edad	etnia	n	logingreso
47001	1	2	3	2	3	2636	12.72
11001	1	1	3	2	3	2616	13.25
47001	1	1	3	2	3	2550	12.82
23001	1	2	3	2	3	2530	12.77
11001	1	2	3	2	3	2441	13.16

Next, we add the covariates.

```
encuesta_df_agg <-
  inner_join(encuesta_df_agg, statelevel_predictors_df)
```

Defining the Multilevel Model

After organizing the survey, we can move on to defining the model.

```
fit <- stan_lmer(
  logingreso ~                    # Log of Mean Income (Y)
    (1 | dam2) +                  # Random Effect (ud)
    edad +                        # Fixed Effect (Covariates X)
    sexo  + tasa_desocupacion +
    luces_nocturnas + cubrimiento_cultivo +
    cubrimiento_urbano ,
  weights = n,                    # Number of observations.
  data = encuesta_df_agg,         # Aggregated survey
  verbose = TRUE,                 # Show progress
  chains = 4,                     # Number of chains.
  iter = 1000 )                   # Number of chain realizations
saveRDS(fit, file = "Data/fit_ingresos.rds")
```

Convergence Check

```
library(posterior)
library(bayesplot)
p1 <-
  (mcmc_dens_chains(fit, pars = "sigma") +
     mcmc_areas(fit, pars = "sigma")) /
  mcmc_trace(fit, pars = "sigma")
ggsave(p1,
  plot = "www/05_Mod_Ingreso/04_Fig_sigma_ing.png" )
```

Chains for σ^2

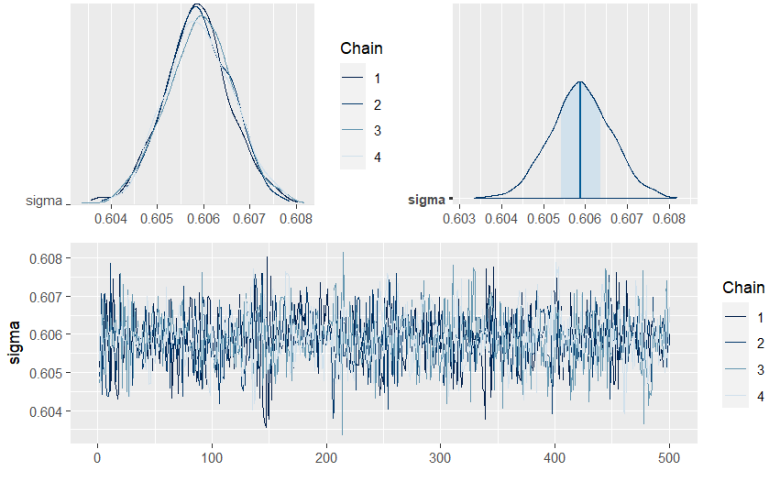


Figure 33: Chains Trace

Posterior Distribution of Coefficients

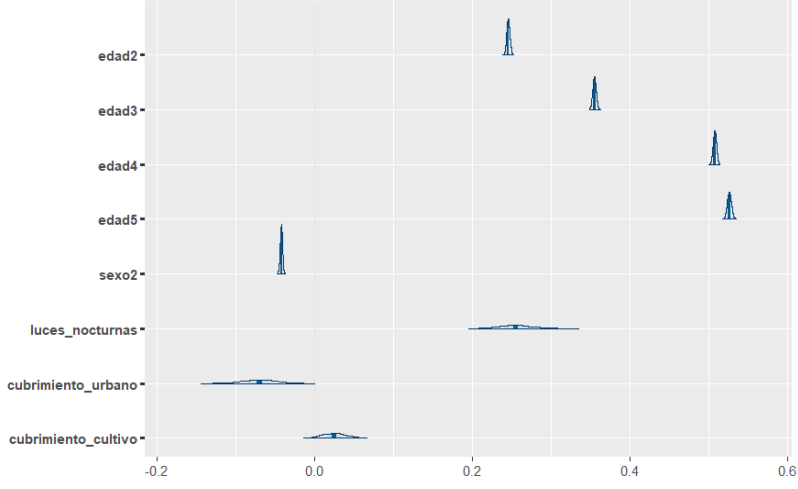


Figure 34: Posterior Distribution for Betas

Model Results in the Survey

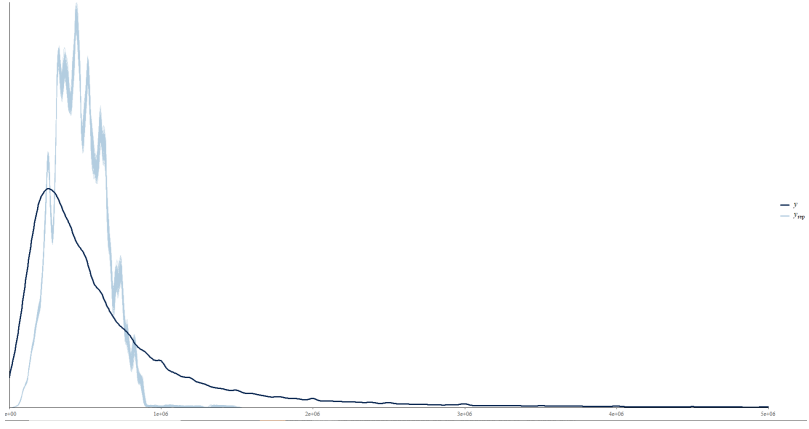


Figure 35: PPC for Income

Model Results in the Survey

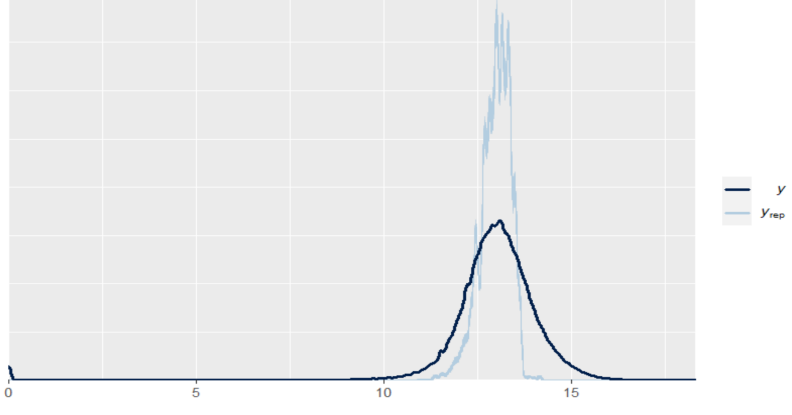


Figure 36: PPC for Log Income

Income Prediction with the Unit Model

The prediction process starts with the reading of the previously standardized aggregated census, which is then joined with the covariate dataset, resulting in the following table.

```
poststrat_df <-
  readRDS("www/05_Mod_Ingreso/censo_dam2.rds") %>%
   left_join(statelevel_predictors_df)
```

Table 13: Aggregated Census and Covariates

dam2	area	sexo	edad	etnia	anoest	n
05001	0	1	1	1	2	1
05001	0	1	1	1	3	2
05001	0	1	1	1	98	1
05001	0	1	1	2	1	5
05001	0	1	1	2	2	20

Posterior Distribution

To obtain a posterior distribution for each observation, you can use the `posterior_epred` function as follows:

```
epred_mat <- posterior_epred(
  fit, newdata = poststrat_df,
  type = "response")
```

Income in Terms of Poverty Lines

To express the estimate of average income in terms of poverty lines, you can use the following code:

```
lp <- encuesta_mrp %>% distinct(área,lp,li))
```

Table 14: Poverty Lines

area	lp	li
1	296845	147169
0	200760	127346

```
lp <- inner_join(poststrat_df,lp,by = "área") %>%
  select(lp)

epred_mat <- (exp(epred_mat)-1)/lp$lp
```

Estimation of National Average Income

The process reduces to matrix operations, which are organized in the `Aux_Agregado` function:

```
mrp_estimate_Ingresolp <-
  Aux_Agregado(poststrat = poststrat_df,
               epredmat = epred_mat,
               byMap = NULL)
```

Table 15: Estimation of National Average Income

Nacional	mrp_estimate	mrp_estimate_se
Nacional	1.931	0.0843

Estimation of Average Income by Administrative Division

In a similar way, it is possible to obtain results for administrative divisions:

```
mrp_estimate_dam2 <-
  Aux_Agregado(poststrat = poststrat_df,
               epredmat = epred_mat,
               byMap = "dam2")
```

Table 16: Estimation by Administrative Division.

dam2	mrp_estimate	mrp_estimate_se
05001	2.604	0.1615
05002	1.028	0.0432
05004	1.066	0.3687
05021	1.089	0.3753
05030	1.575	0.5411

Map of Average Income with the Unit Model

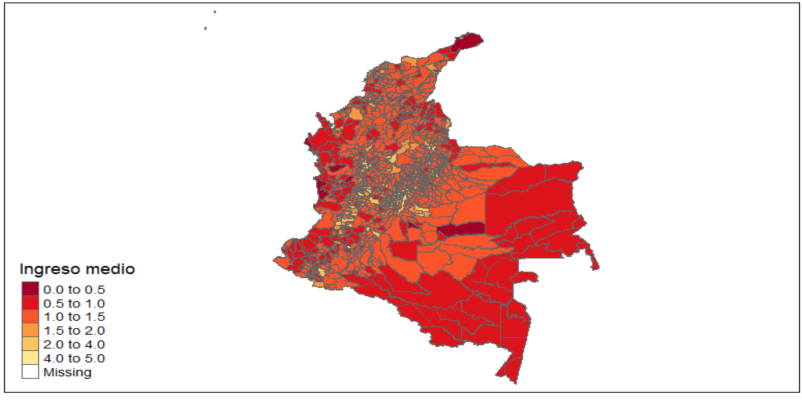


Figure 37: Average Income Map with the Unit Model

Poverty Estimation from Income

Let:

$$y_{ji} = \begin{cases} 1 & \text{if income}_{ji} \leq \text{lp} \\ 0 & \text{otherwise} \end{cases}$$

Where income_{ji} represents the income of the i-th person in the j-th post-stratum, and lp is a threshold, in particular, the poverty line.

```
epred_mat_pobreza_lp <- (exp(epred_mat) - 1) <= lp$lp
```

Poverty Estimation

The process is simplified by applying the previous function.

```
(mrp_estimate_Ingresolp <-
  Aux_Agregado(poststrat = poststrat_df,
               epredmat = epred_mat_pobreza_lp,
               byMap = NULL)
)
```

Table 17: Poverty Estimation

Nacional	mrp_estimate	mrp_estimate_se
Nacional	0.1805	0.0542

Poverty Estimation by dam2

Similarly, it is possible to obtain results for administrative divisions.

```
mrp_estimate_dam2 <-
  Aux_Agregado(poststrat = poststrat_df,
               epredmat = epred_mat,
               byMap = "dam2")
```

Table 18: Estimation by administrative divisions.

dam2	mrp_estimate	mrp_estimate_se
05001	0.0000	0.0000
05002	0.5176	0.0845
05004	0.5230	0.3123
05021	0.5038	0.3061
05030	0.2005	0.2373

Poverty Map by the Unit Model

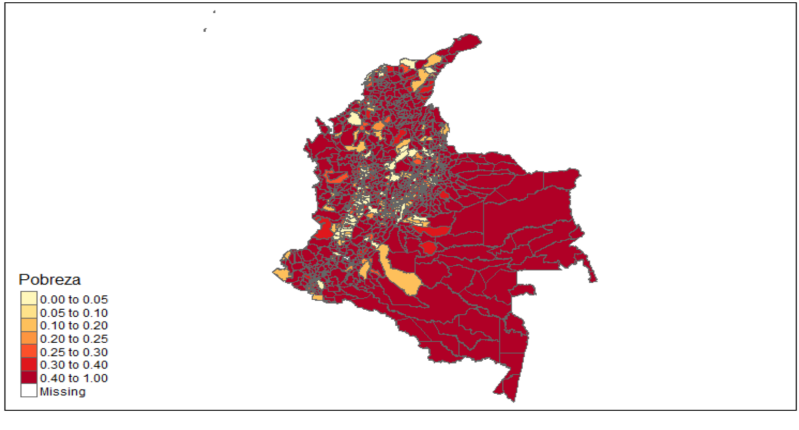


Figure 38: Poverty Map from Mean Income

Unit Model for Poverty Estimation

- Logistic regression is used when the dependent variable is binary since it allows estimating the probability of the studied event.
- To obtain probability estimates, a logarithmic transformation known as the *logit* is performed.
- The logit is calculated as the logarithm of the probability of success divided by the probability of failure:

$$\ln\left(\frac{\theta}{1-\theta}\right)$$

where θ is the probability of success.

Unit Model with Binary Response

- A random effects logistic regression model is employed to relate the expectation θ_{ji} of this variable to available covariates x_{ji} and the random effect u_d.
- The model is expressed as:

$$\ln\left(\frac{\theta_{ji}}{1-\theta_{ji}}\right) = x_{ji}^T\beta + u_d$$

.

- The coefficients β are the fixed effects of the variables on the probabilities, and u_d are random effects.

Prior Distributions

The prior distributions are non-informative and are assumed as follows:

$$\begin{aligned} \beta_k &\sim N(0, 1000) \\ \sigma_y^2 &\sim Inverse - Gamma(0.0001, 0.0001) \end{aligned}$$

Estimation Process

- Estimate the proportion of people below the poverty line:

$$P_d = \frac{\sum_{U_d} y_{di}}{N_d}$$

.

- The estimator is calculated as:

$$\hat{P} = \frac{\sum_{s_d} y_{di} + \sum_{s_d^c} \hat{y}_{di}}{N_d}$$

.

Where $\hat{y}_{di}$ is the expected value of y_{di} under the model.

Estimation in R

The process starts with the definition of poverty using the poverty line defined by CEPAL as follows:

```
encuesta_mrp %<>% mutate(
pobreza = ifelse(ingreso < lp,1,0))
```

Creating a Base with Aggregated Survey Data

Similar to the income model, we now count the number of people below the poverty line aggregated by certain variables.

```
encuesta_df_agg <-
  encuesta_mrp %>%                          # Survey Data
  group_by_at(all_of(byAgrega)) %>%
  summarise(n = n(),                        # Number of observations
  # Count of people with similar characteristics.
       pobreza = sum(pobreza),
       no_pobreza = n-pobreza,
      .groups = "drop") %>%
  arrange(desc(pobreza))                  # Sort the dataset.
```

Aggregated Table

The result of aggregating the database is shown below:

Table 19: Count of people in poverty

dam2	area	sexo	anoest	edad	etnia	pobreza	no_pobreza
47001	1	2	3	2	3	1048	1588
27001	1	2	3	2	2	993	831
20001	1	2	3	2	3	953	1258
23001	1	2	3	2	3	909	1621
47001	1	1	3	2	3	870	1680

Now, the covariates are incorporated.

```
encuesta_df_agg %<>%
  inner_join(statelevel_predictors_df)
```

Unit Model in STAN

After arranging the survey, we can proceed with the definition of the model.

```
fit <- stan_glmer(
  cbind(pobreza, no_pobreza) ~
    (1 | dam2) +              # Random effect (ud)
    edad +                    # Fixed effect (X variables)
    sexo  + tasa_desocupacion +
    luces_nocturnas + cubrimiento_cultivo +
    cubrimiento_urbano ,
    data = encuesta_df_agg, # Aggregated survey
    verbose = TRUE,          # Show progress
    chains = 4,              # Number of chains
    iter = 100, cores = 4,
    family = binomial(link = "logit")
)
saveRDS(fit, file = "www/06_Mod_Pobreza/fit_pobreza.rds")
```

Posterior Distribution of Coefficients

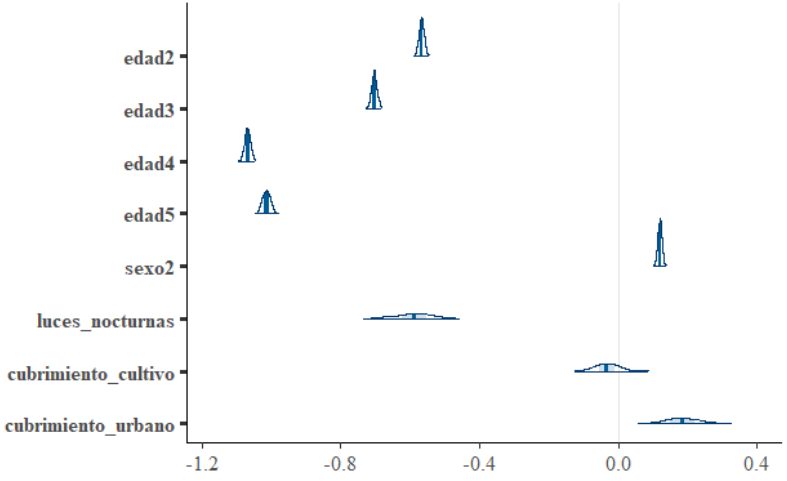


Figure 39: Posterior distribution of coefficients

Trace of Chains for Coefficients

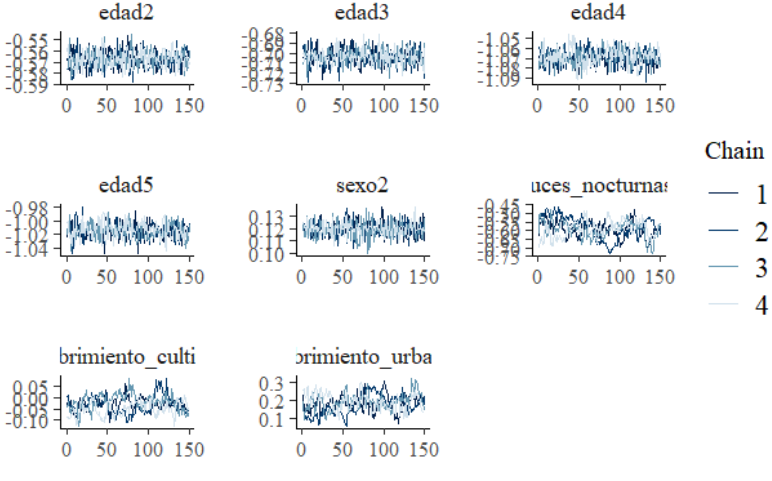


Figure 40: Chains of coefficients

Model Results in the Survey

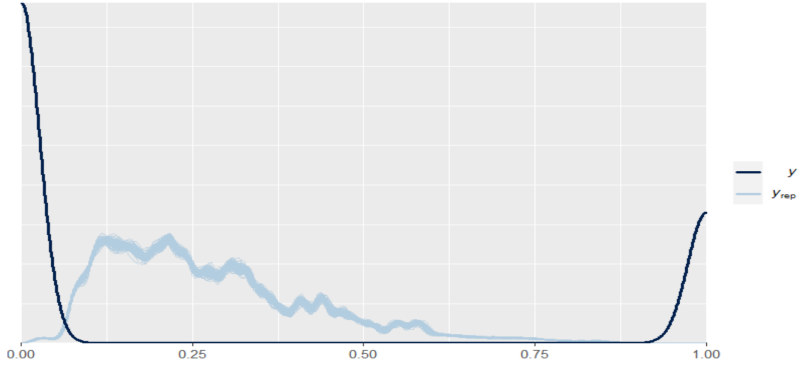


Figure 41: PPC for poverty

Poverty Prediction with the Unit Model

The prediction process begins with the reading of the previously standardized aggregated census, then it is joined with the covariate database, resulting in the following table.

```
poststrat_df <-
  readRDS("www/06_Mod_Pobreza/censo_dam2.rds") %>%
   left_join(statelevel_predictors_df)
```

Table 20: Aggregated Census and Covariates

dam2	area	sexo	edad	etnia	anoest	n
05001	0	1	1	1	2	1
05001	0	1	1	1	3	2
05001	0	1	1	1	98	1
05001	0	1	1	2	1	5
05001	0	1	1	2	2	20

Posterior Distribution

To obtain a posterior distribution for each observation, you can use the *posterior_epred* function as follows:

```
epred_mat <- posterior_epred(
  fit, newdata = poststrat_df,
  type = "response")
```

Estimating the Poverty Rate

Similar to the income model, you can use the *Aux_Agregado* function to obtain poverty rate estimates.

```
(mrp_estimate_Ingresolp <-
  Aux_Agregado(poststrat = poststrat_df,
               epredmat = epred_mat,
               byMap = NULL)
) %>% tba()
```

Table 21: Poverty Rate Estimation

Nacional	mrp_estimate	mrp_estimate_se
Nacional	0.2849	0.0286

Estimating the Poverty Rate by dam2

Similarly, you can obtain results for the administrative divisions of the country.

```
mrp_estimate_dam2 <-
  Aux_Agregado(poststrat = poststrat_df,
               epredmat = epred_mat,
               byMap = "dam2")
```

dam2	mrp_estimate	mrp_estimate_se
05001	0.1532	0.0021
05002	0.4189	0.0300
05004	0.4747	0.1545
05021	0.4638	0.1539
05030	0.2981	0.1337

Estimated Poverty Map with the Unit Model

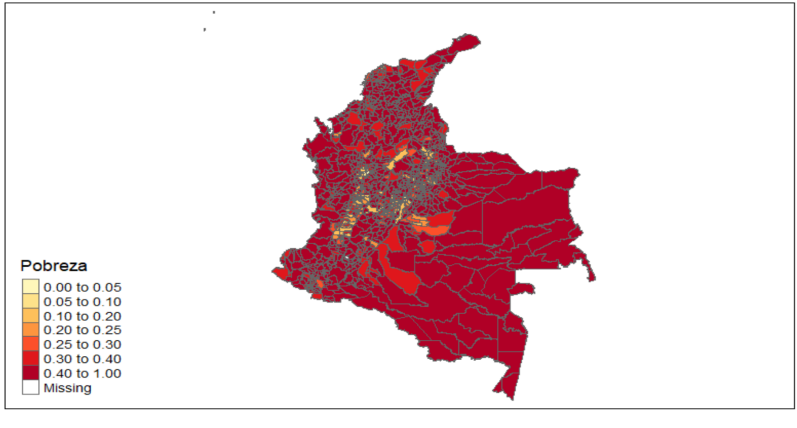


Figure 42: Poverty Map with Unit Model

Unit Model Multidimensional Deprivation Index (MDI)

Introducción

- Poverty is a crucial topic on the national and international agenda, as evidenced by the first goal of the 2030 Sustainable Development Agenda.
- Traditionally, poverty has been measured in a one-dimensional way, based on income and expenses.
- Addressing poverty from a multidimensional perspective allows us to capture a broader range of factors that affect the quality of life.

Multidimensional Poverty Index (MPI)

- The MPI is a measure that assesses poverty by considering multiple dimensions of well-being.
- It is calculated using weights and thresholds based on different indicators of quality of life.
- The MPI is a variant of the FGT methodology (Foster, Greer, and Thorbecke, 1984) used to measure one-dimensional poverty.

MPI Equation

- It is expressed as an average of censored deprivation scores, as detailed in the following equations:

$$MPI = \frac{1}{N} \sum_{i=1}^{N} c_i(z)$$

Where:

- N is the number of individuals or households in the population.
- $c_i(z)$ is the censored deprivation score of observation i.

Calculation of $c_i(z)$

The way to obtain $c_i(z)$ is given by the following equation:

- If $q_i \geq z$, then c_i will be equal to q_i.
- If $q_i < z$, then c_i will be equal to 0.

Where: $q_i = \sum_{k=1}^{K} w_k \cdot y_i^k$, where K is the number of dimensions or indicators of deprivation, w_k is the weight associated with dimension k, and y_i^k is a binary variable.

Components of the MPI:

1. **Headcount Ratio (H):**
   - Measures the proportion of people deprived in at least one dimension of poverty.
   - It is calculated as the number of people deprived in at least one dimension over the total population.

$$H = \frac{1}{N} \sum_{i=1}^{N} I\left(q_i \geq z\right) = \frac{N\left(z\right)}{N}$$

where $N\left(z\right) = \sum_{i=1}^{N} I\left(q_i \geq z\right)$

2. **Intensity of Deprivation (A):**
   - Measures the average intensity of deprivation among the deprived people.
   - It is calculated as the average of deprivation scores for those individuals who are deprived in at least one dimension.

$$A = \sum_{i=1}^{N} \frac{c_i\left(z\right)}{N\left(z\right)}$$

Calculation of the MPI from H and A

- The MPI is obtained by multiplying the values of H and A.
- Mathematically, it is expressed as the average of censored deprivation scores.

$$MPI = \frac{N(z)}{N} \times \sum_{i=1}^{N} \frac{c_i(z)}{N(z)} = \frac{1}{N} \sum_{i=1}^{N} c_i(z)$$

Unit Model for MPI

- In many applications, the variable of interest in small areas is binary, meaning y_{dj} takes values of 0 or 1, representing the absence or presence of a specific characteristic.
- The estimation objective in each domain $d = 1, \cdots, D$ is the proportion $\theta_d = \frac{1}{N_d} \sum_{i=1}^{N_d} y_{di}$ of the population that exhibits this characteristic.
- The logit of θ_{di} is defined as

$$\ln\left(\frac{\theta_{di}}{1-\theta_{di}}\right) = \eta_{di} = x_{di}^T \beta + u_d$$

where β is a vector of fixed effect parameters, and u_d is a specific random effect for the area in domain d with $u_d \sim N\left(0, \sigma_u^2\right)$.

Unit Model for MPI

- The u_d are independent, and $y_{di} \mid u_d \sim Bernoulli(\theta_{di})$ with $E(y_{di} \mid u_d) = \theta_{di}$ and $Var(y_{di} \mid u_d) = \sigma^2_{di} = \theta_{di}(1 - \theta_{di})$.
- x^T_{di} represents a vector of $p \times 1$ values of p auxiliary variables.
- Thus, θ_{di} can be expressed as:

$$\theta_{di} = \frac{\exp(x^T_{di}\beta + u_d)}{1 + \exp(x^T_{di}\beta + u_d)}$$

The model is estimated for each dimension.

Prior Distributions

As is traditional, non-informative prior distributions are used:

$$\begin{aligned} \beta_k &\sim N(0, 1000) \\ \sigma_y^2 &\sim Inverse - Gamma(0.0001, 0.0001) \end{aligned}$$

Estimation of MPI Using Unit Models

- Estimate the proportion of people who have the k-th deprivation, that is, $P_d = \frac{\sum_{U_d} c_{di}(z)}{N_d}$.
- The estimator of P is calculated as:

$$\hat{P}_d = \frac{\sum_{s_d} c_{di}(z) + \sum_{s_d^c} \hat{c}_{di}(z)}{N_d}$$

where $\hat{c}_{di}(z)$ is defined as:

- If $\hat{q}_{di} \geq z$, then c_{di} is equal to $\hat{q}_{di}$
- If $\hat{q}_{di} < z$, then c_{di} is equal to 0

Estimation of q_{di}

The estimation of $\hat{q}$ is given by

$$\hat{q}_{di} = \sum_{k=1}^{K} w_k \cdot \hat{y}_{di}^k$$

where

$$\hat{y}_{di}^k = E_{\mathcal{M}} \left(y_{di}^k \mid x_d, \beta \right)$$

Thus, the estimator of P is obtained for each domain d.

Estimation of θ_{di}^{k}

The estimation of θ_{di}^{k} reflects the probability that a specific unit i in domain d obtains the value 1 in dimension k. To carry out this estimation, we follow the following procedure:

$$\bar{Y}_{d}^{k} = \theta_{d}^{k} = \frac{1}{N_d} \sum_{i=1}^{N_d} y_{di}^{k}$$

Here, y_{di}^{k} can take values of 0 or 1, representing the absence (or presence) of a specific feature.

Estimation of θ_{di}^k

Divide the sum into two parts: s_d, representing the units observed in a sample, and s_d^c, which are the unobserved units. Therefore,

$$\bar{Y}_d^k = \theta_d^k = \frac{1}{N_d}\left(\sum_{s_d} y_{di}^k + \sum_{s_d^c} y_{di}^k\right)$$

Estimation of θ_{di}^{k}

Through a unit model, it is possible to predict y_{di}^{k} for the unobserved units. In this way, the estimator of θ_{d}^{k} is expressed as:

$$\hat{\theta}_{d}^{k} = \frac{1}{N_d}\left(\sum_{s_d} y_{di}^{k} + \sum_{s_d^c} \hat{y}_{di}^{k}\right)$$

Where,

$$\hat{y}_{di}^{k} = E_{\mathcal{M}}\left(y_{di}^{k} \mid x_d, \beta\right)$$

Estimation of θ_{di}^k

The estimation $\hat{\theta}_d^k$ simplifies to:

$$\hat{\theta}_d^k = \frac{1}{N_d} \sum_{i=1}^{N_d} \hat{y}_{di}^k$$

This approach allows for the estimation of the probability θ_d^k in domain d in dimension k using predictions and available data rather than relying on detailed individual information for all cases.

Prediction of "Hard Estimates"

- Hobza and Morales (2016) define "hard estimates" as binary values (0 or 1) that precisely indicate whether an individual has a specific feature regarding each multidimensional deprivation indicator.
- The estimation of θ_{di}^{k} reflects the probability that a specific unit i in domain d obtains the value 1 in dimension k.
- Therefore, $\hat{y}_{di}^{k} \sim Bernoulli(\hat{\pi}_{di}^{k})$ is defined, where $\hat{y}_{di}^{k}$ are the "hard" estimates.

Point Estimation of the MPI

The proposed procedure for estimating the MPI is as follows:

1. Use sample data to fit a Bernoulli unit-level logit model for each indicator. This is accomplished using the Markov Chain Monte Carlo (MCMC) algorithm with L iterations.
2. For each dimension k for which a unit-level Bernoulli logit model was estimated with L iterations, predict the values $\widehat{y}_{di}^{k}$ for each individual in the census. This will generate L random realizations of $\widehat{y}_{di}^{k}$.
3. Let $\widehat{y}_{di}^{kl}$ denote the l-th random realization of dimension k for individual i in domain d. Calculate $q_{di}^{l} = \sum_{k=1}^{K} w_k \cdot y_{di}^{kl}$.

Point Estimation of the MPI

From the calculated values for q_{di}, H_d^l, A_d^l, and MPI_d^l can be calculated using the equations:

$$MPI_d^l = \frac{1}{N_d} \sum_{i=1}^{N_d} c_{di}^l(z)$$

$$H_d^l = \frac{1}{N_d} \sum_{i=1}^{N_d} I\left(q_{di}^l \geq z\right) = \frac{N_d^l(z)}{N_d}$$

and

$$A_d^l = \sum_{i=1}^{N_d} \frac{c_{di}^l(z)}{N_d^l(z)}$$

Point Estimation of the MPI

4. The point estimation of H_d, A_d, and MPI_d in each small area d is calculated by taking the average over the L iterations:

$$\hat{H}_d = \frac{1}{L}\sum_{l=1}^{L} H_d^l,$$

$$\hat{A}_d = \frac{1}{L}\sum_{l=1}^{L} A_d^l$$

and

$$\widehat{MPI}_d = \frac{1}{L}\sum_{l=1}^{L} MPI_d^l$$

Estimation of the Variance for the MPI

5. Since the model was estimated using the MCMC algorithm, it is possible to estimate the estimation error as follows:

$$\widehat{Var}(\hat{H}_d) = \frac{1}{L}\sum_{l=1}^{L}\left(H_d^l - \hat{H}_d\right)^2$$

$$\widehat{Var}(\hat{A}_d) = \frac{1}{L}\sum_{l=1}^{L}\left(A_d^l - \hat{A}_d\right)^2$$

and

$$\widehat{Var}(\widehat{MPI}_d) = \frac{1}{L}\sum_{l=1}^{L}\left(MPI_d^l - \widehat{MPI}_d\right)^2$$

Multidimensional Poverty Index in Colombia

In Colombia, there are 9 indicators that are measured as deprivations: $y_{di}^k = 1$ if the person experiences the deprivation, and $y_{di}^k = 0$ if the person does not experience the deprivation.

The index requires information for each individual $i = 1, \dots, N_d$ in the domains $d = 1, \dots, D$, where N_d denotes the population size of domain d.

For this study, we use the value of 0.4 for z, meaning $I(\cdot)$ is equal to 1 when $q_{di} \geq 0.4$. The value of q_{di} in domain d is calculated as:

$$q_{di} = \frac{1}{16}(y_{di}^1 + y_{di}^2 + y_{di}^3 + y_{di}^4) + \frac{1}{12}(y_{di}^5 + y_{di}^6 + y_{di}^7) + \frac{1}{4}(y_{di}^8 + y_{di}^9)$$

Deprivations Calculated for Colombia

a. $y_{di}^1 =$ Deprivation in housing construction material.
b. $y_{di}^2 =$ Overcrowding in the household.
c. $y_{di}^3 =$ Lack of access to the Internet service.
d. $y_{di}^4 =$ Lack of access to electrical energy service.
e. $y_{di}^5 =$ Deprivation in sanitation.
f. $y_{di}^6 =$ Deprivation of access to clean drinking water.
g. $y_{di}^7 =$ Deprivation in health.
h. $y_{di}^8 =$ Deprivation of education.
i. $y_{di}^9 =$ Deprivation of employment and social protection.

Dimensions of Deprivations

The previous deprivations are grouped by dimensions as follows:

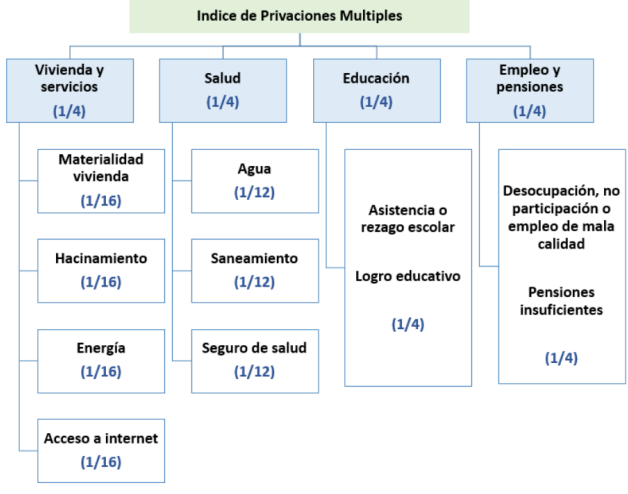


Figure 43: MPI dimensions and weights

Household Survey with Deprivation Indicators

In the following table, you can see a sample of the y_{di}^{k} indicators for Colombia.

Table 22: Deprivation Indices in Colombia

dam2	nbi_matviv_ee	nbi_hacina_ee	nbi_tic	nbi_agua_ee
05360	0	0	0	0
05360	0	0	0	0
05360	0	0	0	0
05360	0	0	0	0
05360	0	0	0	0
05360	0	0	1	0
05360	0	0	1	0
05360	0	0	1	0
05360	0	0	1	0
05360	0	0	1	0

Process for Aggregating the Surveys

Reading the survey and defining variables for aggregation

```
encuesta_ipm <-
  readRDS("www/07_Mod_IPM/encuesta_nbi.rds")

byAgrega <- c("dam", "dam2", "área", "sexo",
              "etnia", "anoest",  "edad")

names_ipm <- grep(pattern = "nbi",
                  names(encuesta_ipm), value = TRUE)
```

Process for Aggregating the Surveys

The process is repeated for each of the deprivations, and it is automated as follows:

```
encuesta_df <- map(
  setNames(names_ipm, names_ipm),
   function(y) {
    encuesta_ipm$temp <- as.numeric(encuesta_ipm[[y]])
    encuesta_ipm %>%
    group_by_at(all_of(byAgrega)) %>%
     summarise( n = n(), yno = sum(temp),
     ysi = n - yno, .groups = "drop"
    ) %>%
  inner_join(statelevel_predictors_df,
  by = c("dam", "dam2"))
})
saveRDS(encuesta_df,
        "www/07_Mod_IPM/03_tabla_encuesta_agg.rds")
```

Sample of the Resulting Datasets

The resulting datasets look like this:

Table 23: Deprivation in Housing Construction Material

dam2	area	sexo	etnia	anoest	n	yno	ysi	area1	sexo2
05001	0	1	3	1	3	0	3	0.9832	0.5299
05001	0	1	3	1	1	0	1	0.9832	0.5299
05001	0	1	3	1	1	1	0	0.9832	0.5299
05001	0	1	3	2	3	0	3	0.9832	0.5299
05001	0	1	3	2	2	0	2	0.9832	0.5299
05001	0	1	3	2	1	0	1	0.9832	0.5299
05001	0	1	3	2	3	0	3	0.9832	0.5299
05001	0	1	3	2	2	0	2	0.9832	0.5299
05001	0	1	3	3	6	1	5	0.9832	0.5299
05001	0	1	3	3	1	0	1	0.9832	0.5299

Define the Model

```
names_cov <-  statelevel_predictors_df %>%
  dplyr::select(-dam,-dam2) %>% names()
names_cov <- c("sexo","área",names_cov[16:19])
efec_aleat <-
  paste0("(1|",  c("dam", "etnia"), ")",
         collapse = "+")
formula_mod <-  formula(paste(
    " cbind(yno, ysi) ~", efec_aleat,
    "+",  paste0(names_cov,
           collapse = " + ")
  ))

formula_mod
```

```
cbind(yno, ysi) ~ (1 | dam) + (1 | etnia) + exo + área +
tasa_desocupacion + luces_nocturnas + cubrimiento_cultivo +
cubrimiento_urbano
```

Running the Models

```
plan(multisession, workers = 4)

fit <- future_map(encuesta_df, function(xdat){
stan_glmer(formula = formula_mod ,
  family = binomial(link = "logit"),
  data = xdat,
  cores = 4,
  chains = 4,
  iter = 500
)},
.progress = TRUE)

saveRDS(object = fit, "www/07_Mod_IPM/Modelos/fits_IPM.rds")
```

Predicting θ_{di}^{kl}

- The models were compiled separately, so we have an `.rds` object for each deprivation that makes up the MPI.
- The process is illustrated for water deprivation, but it is the same for the other deprivations.

Prediction on the Census Data

```
censo_ipm <-
  readRDS("www/07_Mod_IPM/04_tabla_censo.rds")

fit_agua <-
  readRDS(file = "www/07_Mod_IPM/Modelos/fit_agua.rds")

epred_mat_agua <-
  posterior_epred(
    fit_agua,
    newdata = poststrat_df,
    type = "response",
    allow.new.levels = TRUE
  )
```

Defining the Hard Estimates

In this code, we predict θ_{di}^{kl} for water deprivation using the compiled model. The predictions are stored in `epred_mat_agua`. We also create hard estimates by randomly sampling from these predictions and save them as `epred_mat_agua_dummy` in an RDS file.

```
epred_mat_agua_dummy <-
  rbinom(
    n = nrow(epred_mat_agua) * ncol(epred_mat_agua), 1,
         epred_mat_agua)

epred_mat_agua_dummy <- matrix(
  epred_mat_agua_dummy,
  nrow = nrow(epred_mat_agua),
  ncol = ncol(epred_mat_agua)
)
saveRDS(epred_mat_agua_dummy,
        "www/07_Mod_IPM/Dummys/epred_mat_agua_dummy.rds")
```

Calculating q^l_{di}

The calculation of q^l_{id} is a simple matrix operation.

```
chain_q  <-
  # Housing and services
  (1 / 16) * ( epred_mat_material_dummy +
               epred_mat_hacinamiento_dummy +
               epred_mat_energia_dummy +
               epred_mat_tic_dummy
  ) +
  # Health
  (1 / 12) * (epred_mat_agua_dummy +
              epred_mat_saneamiento_dummy +
              epred_mat_salud_dummy) +
  # Education
  (1 / 4) * epred_mat_educacion_dummy  +
  # Employment
  (1 / 4) * epred_mat_empleo_dummy
```

Calculating $I\left(q_{di}^{l} \geq z\right)$ and $c_{di}^{l}\left(z\right)$

Now, it is possible to calculate $I\left(q_{di}^{l} \geq z\right)$, taking $z = 0.4$ as the threshold.

```
chain_Ind <- chain_q
chain_Ind[chain_Ind < 0.4] <- 0
chain_Ind[chain_Ind != 0] <- 1
```

Next, we calculate $c_{di}^{l}\left(z\right)$.

```
chain_ci <- matrix(0, nrow = nrow(chain_q),
                   ncol = ncol(chain_q))
chain_ci[chain_Ind == 1] <- chain_q[chain_Ind == 1]
```

Results obtained in the first iterations

Table 24: Chains obtained

q1	q2	Ind1	Ind2	c1	c2	N
0.1875	0.6458	0	1	0.0000	0.6458	1
0.7083	0.3958	1	0	0.7083	0.0000	1
0.7708	0.5208	1	1	0.7708	0.5208	9
0.6250	0.4583	1	1	0.6250	0.4583	1
0.5625	0.3750	1	0	0.5625	0.0000	5
0.3125	0.6250	0	1	0.0000	0.6250	22
0.2500	0.3750	0	0	0.0000	0.0000	9
0.6250	0.2083	1	0	0.6250	0.0000	76
0.3958	0.7083	0	1	0.0000	0.7083	796
0.7083	0.6250	1	1	0.7083	0.6250	3549

Disaggregated MPI Estimates

To obtain disaggregated MPI estimates, a function was developed to facilitate the calculation, for example, by administrative division (*dam2*).

```
source("www/07_Mod_IPM/06_Estimar_ipm.R")
ipm_dam2 <- estime_IPM(
  poststrat = censo_ipm,
  chain_ci = chain_ci,
  chain_ind = chain_ind,
  byMap = "dam2"
) %>% data.frame()
```

Disaggregated MPI Estimates

Table 25: Estimates by administrative division

dam2	H	H_sd	A	A_sd	IPM	IPM_sd
05001	0.1247	0.0746	0.5502	0.0295	0.0684	0.0411
05002	0.6014	0.0781	0.6229	0.0258	0.3746	0.0510
05004	0.6555	0.0767	0.6291	0.0260	0.4124	0.0511
05021	0.5196	0.0807	0.6152	0.0236	0.3194	0.0494
05030	0.4771	0.0771	0.6029	0.0225	0.2874	0.0463
05031	0.5242	0.0780	0.6174	0.0231	0.3233	0.0471
05034	0.5397	0.0742	0.6161	0.0226	0.3323	0.0460
05036	0.5246	0.0785	0.6092	0.0242	0.3196	0.0496
05038	0.6856	0.0878	0.6300	0.0269	0.4320	0.0587
05040	0.5796	0.0799	0.6228	0.0231	0.3609	0.0511

IPM Estimates by Deprivation

- It is essential to analyze each dimension individually.
- This allows for a better understanding of the complexity of poverty and the design of effective strategies.
- “Hard estimates” are used to calculate the estimates for each deprivation.
- The process is applied similarly to all deprivations.

Estimation Process

To streamline the calculation process, the **aggregated_dim_ipm** function is created to perform the calculations. Here's how to use it:

```
source("www/07_Mod_IPM/07_Fun_agregado.r")

epred_mat_agua_dummy <-
readRDS("www/07_Mod_IPM/Dummys/epred_mat_agua_dummy.rds")

datos_dam_agua <-
  aggregated_dim_ipm(poststrat = censo_ipm,
            epredmat = epred_mat_agua_dummy,
            byMap = "dam2")
```

Estimation by Administrative Division

Table 26: Estimation by dam2 for water deprivation

dam2	estimate	estimate_se
05001	0.0012	0.0082
05002	0.1645	0.0753
05004	0.2000	0.0772
05021	0.1319	0.0572
05030	0.0925	0.0491
05031	0.1352	0.0562
05034	0.1391	0.0570
05036	0.1227	0.0625
05038	0.2084	0.0936
05040	0.1636	0.0652

Results for All Deprivations

Table 27: Point Estimate by Municipality and Dimension

dam2	Agua	Educacion	Empleo	Energia	Internet
05001	0.0012	0.2518	0.4725	0.0025	0.4783
05002	0.1645	0.6267	0.7798	0.2160	0.8866
05004	0.2000	0.6593	0.7985	0.2673	0.9089
05021	0.1319	0.5715	0.7442	0.1656	0.8532
05030	0.0925	0.5443	0.7119	0.1046	0.8295
05031	0.1352	0.5731	0.7481	0.1722	0.8542
05034	0.1391	0.5837	0.7512	0.1703	0.8599
05036	0.1227	0.5818	0.7383	0.1272	0.8660
05038	0.2084	0.6816	0.8165	0.2823	0.9242
05040	0.1636	0.6089	0.7721	0.2093	0.8784

Results for All Deprivations

Table 28: Estimation Error by Municipality and Dimension

dam2	Agua_se	Educacion_se	Empleo_se	Energia_se	Internet_se
05001	0.0082	0.0981	0.1089	0.0081	0.1138
05002	0.0753	0.0820	0.0718	0.0775	0.0470
05004	0.0772	0.0883	0.0697	0.0857	0.0428
05021	0.0572	0.0877	0.0777	0.0608	0.0622
05030	0.0491	0.0798	0.0777	0.0529	0.0604
05031	0.0562	0.0827	0.0764	0.0564	0.0614
05034	0.0570	0.0796	0.0710	0.0609	0.0549
05036	0.0625	0.0825	0.0719	0.0616	0.0508
05038	0.0936	0.0931	0.0797	0.0969	0.0397
05040	0.0652	0.0836	0.0750	0.0686	0.0531

Map of MPI Components

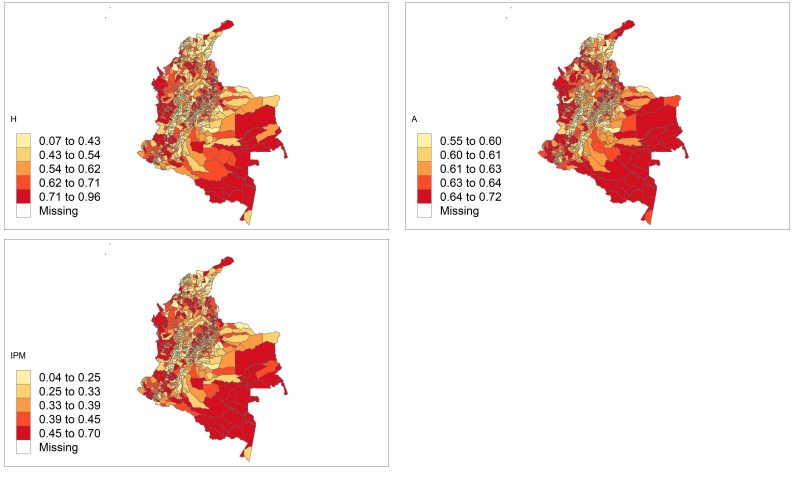


Figure 44: MPI Components

Map of Deprivations that Compose the MPI

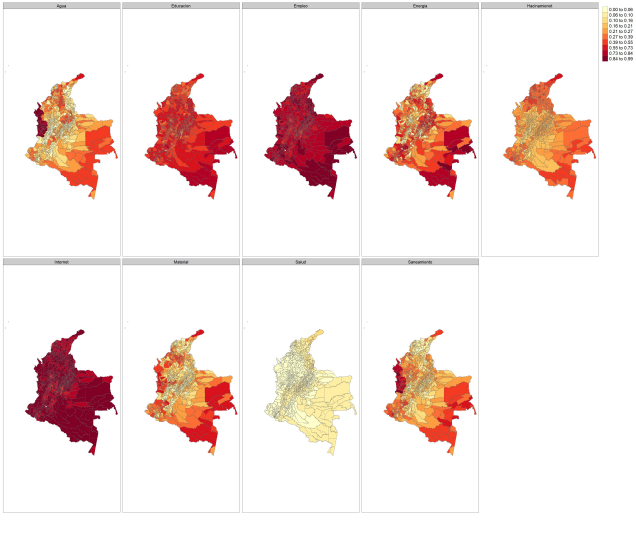


Figure 45: Deprivations of MPI

Area Model for Labor Market Statistics

Definition of the Multinomial Model

- Let K be the number of categories of the variable of interest $Y \sim multinomial\,(\theta)$, with $\theta = (p_1, p_2, ..., p_k)$ and $\sum_{k=1}^{K} p_k = 1$.
- Let N_i be the number of elements in the i-th domain, and N_{ik} be the number of elements that belong to the k-th category. Note that $\sum_{k=1}^{K} N_{ik} = N_i$ and $p_{ik} = \frac{N_{ik}}{N_i}$.
- Let $\hat{p}_{ik}$ be the direct estimate of p_{ik} and $v_{ik} = Var\,(\hat{p}_{ik})$, and denote the variance estimator as $\hat{v}_{ik} = \widehat{Var}\,(\hat{p}_{ik})$.

Considerations for the Multinomial Model

The design effect varies by category; therefore, the first step is to define the effective sample size per category.

The estimation of $\tilde{n}$ is given by

$$\tilde{n}_{ik} = \frac{(\tilde{p}_{ik} \times (1 - \tilde{p}_{ik}))}{\hat{v}_{ik}}$$

,

$$\tilde{y}_{ik} = \tilde{n}_{ik} \times \hat{p}_{ik}$$

then, $\hat{n}_i = \sum_{k=1}^{K} \tilde{y}_{ik}$ from which it follows that $\hat{y}_{ik} = \hat{n}_i \times \hat{p}_{ik}$.

Multinomial Area Model

Let $\theta = (p_1, p_2, p_3)^T = \left(\frac{N_{i1}}{N_i}, \frac{N_{i2}}{N_i}, \frac{N_{i3}}{N_i}\right)^T$; then the multinomial model for the i-th domain would be given by:

$$(\tilde{y}_{i1}, \tilde{y}_{i2}, \tilde{y}_{i3}) \mid \hat{n}_i, \theta_i \sim multinomial\,(\hat{n}_i, \theta_i)$$

Now, you can write p_{ik} as:

$\ln\left(\frac{p_{i2}}{p_{i1}}\right) = X_i^T \beta_2 + u_{i2}$ and $\ln\left(\frac{p_{i3}}{p_{i1}}\right) = X_i^T \beta_3 + u_{i3}$

Multinomial Area Model

Given the constraint $1 = p_{i1} + p_{i2} + p_{i3}$, then

$$p_{i1} + p_{i1}(e^{X_i^T \beta_2} + u_{i2}) + p_{i1}(e^{X_i^T \beta_3} + u_{i3})$$

from which it follows that

$$p_{i1} = \frac{1}{1 + e^{X_i^T \beta_2} + u_{i2} + e^{X_i^T \beta_2} + u_{i3}}$$

Multinomial Area Model

The expressions for p_{i2} and p_{i3} would be as follows:

$$p_{i2} = \frac{e^{X_i^T \beta_2} + u_{i2}}{1 + e^{X_i^T \beta_2} + u_{i2} + e^{X_i{}^T \beta_2} + u_{i3}}$$

$$p_{i3} = \frac{e^{X_i^T \beta_3} + u_{i3}}{1 + e^{X_i^T \beta_2} + u_{i2} + e^{X_i{}^T \beta_3} + u_{i3}}$$

Direct Estimation by Municipality

- A surveyed person can be in one of the following states: employed, unemployed, or inactive.
- For each domain, the number of employed, unemployed, and inactive people in each domain is calculated, and the proportion of people in each of these categories is estimated with their respective standard errors and design effects.

Selection of Domains

- Several quality measures are employed, including counting the number of domains that have two or more primary sampling units (PSUs), as well as a design effect greater than 1 and variances greater than 0.
- The selected domains are:
  - Domains with two or more PSUs.
  - Having a result in the DEFF.

The number of selected domains was 413.

STAN Programming for the Model: `functions`

Create a function that simplifies the calculations.

```
functions {
  matrix pred_theta(matrix Xp, int p, matrix beta){
  int D1 = rows(Xp);
  real num1[D1, p];
  real den1[D1];
  matrix[D1,p] theta_p;
  for(d in 1:D1){
    num1[d, 1] = 1;
    num1[d, 2] = exp(Xp[d, ] * beta[1, ]' ) ;
    num1[d, 3] = exp(Xp[d, ] * beta[2, ]' ) ;
    den1[d] = sum(num1[d, ]);
  }
  for(d in 1:D1){
    for(i in 2:p){
    theta_p[d, i] = num1[d, i]/den1[d];
    }
    theta_p[d, 1] = 1/den1[d];
   }
 return theta_p  ;
  }}
```

STAN Programming for the Model: data and parameters

```
data {
  int<lower=1> D; // Number of domains
  int<lower=1> P; // Categories
  int<lower=1> K; // Number of regressors
  int y_tilde[D, P]; // Data matrix
  matrix[D, K] X_obs; // Covariate matrix
  int<lower=1> D1; // Number of domains
  matrix[D1, K] X_pred; // Covariate matrix
}
parameters {
  matrix[P-1, K] beta; // Parameter matrix
  real<lower=0> sigma2_u1;
  real<lower=0> sigma2_u2;
  vector[D] u1;
  vector[D] u2;
}
```

STAN Programming for the Model: `transformed parameters`

```
transformed parameters {
  simplex[P] theta[D]; // Parameter vector
  real num[D, P];
  real den[D];
  real<lower=0> sigma_u1;
  real<lower=0> sigma_u2;
  sigma_u1 = sqrt(sigma2_u1);
  sigma_u2 = sqrt(sigma2_u2);
  for(d in 1:D){
    num[d, 1] = 1;
    num[d, 2] = exp(X_obs[d, ] * beta[1, ]' + u1[d]) ;
    num[d, 3] = exp(X_obs[d, ] * beta[2, ]' + u2[d]) ;
    den[d] = sum(num[d, ]);
  }
  for(d in 1:D){
    for(p in 2:P){
    theta[d, p] = num[d, p]/den[d];
    }
    theta[d, 1] = 1/den[d];
  }
}
```

STAN Programming for the Model: `model`

```
model {
 u1 ~ normal(0, sigma_u1);
 u2 ~ normal(0, sigma_u2);
 sigma2_u1 ~  inv_gamma(0.0001, 0.0001);
 sigma2_u2 ~  inv_gamma(0.0001, 0.0001);
 for(p in 2:P){
    for(k in 1:K){
      beta[p-1, k] ~ normal(0, 10000);
    }
}
for(d in 1:D){
    target += multinomial_lpmf(y_tilde[d, ] | theta[d, ]);
  }
}
```

STAN Programming for the Model: `generated quantitie`

```
generated quantities {
  matrix[D1,P] theta_pred;
  theta_pred = pred_theta(X_pred, P, beta);
}
```

Identifying Unobserved Domains

Select the variables for the model and create the covariate matrix.

```
names_cov <- c("dam2", "unemployment_rate", "overcrowding", "earth_floor",
X_pred <- anti_join(
    statelevel_predictors_df %>%
    select(all_of(names_cov)),
    indicador_dam1 %>% select(dam2)
)
```

Unobserved Domains

Create the covariate matrix for unobserved domains (`X_pred`) and observed domains (`X_obs`).

```
X_pred <- X_pred %>%
    data.frame() %>%
    select(-dam2) %>%
    as.matrix()

X_obs <- inner_join(
    indicador_dam1 %>%
    select(dam2, id_order),
    statelevel_predictors_df %>%
    select(all_of(names_cov))
) %>%
arrange(id_order) %>%
data.frame() %>%
select(-dam2, -id_order) %>%
as.matrix()
```

Calculating the Effective Sample Size (n_{eff}) and $\tilde{y}$

```
D <- nrow(indicador_dam1)
P <- 3 # Occupied, Unoccupied, Inactive.
Y_tilde <- matrix(NA, D, P)
n_tilde <- matrix(NA, D, P)
Y_hat <- matrix(NA, D, P)

# Effective sample size for Occupied
n_tilde[,1] <- (indicador_dam1$Ocupado * (1 - indicador_dam1$Ocupado)) /
  indicador_dam1$Ocupado_var
Y_tilde[,1] <- n_tilde[,1] * indicador_dam1$Ocupado
```

Calculating the Effective Sample Size (n_{eff}) and $\tilde{y}$

```
n_tilde[,2] <- (indicador_dam1$Desocupado * (1 - indicador_dam1$Desocupado)
  indicador_dam1$Desocupado_var

Y_tilde[,2] <- n_tilde[,2] * indicador_dam1$Desocupado

# Effective sample size for Inactive
n_tilde[,3] <- (indicador_dam1$Inactivo * (1 - indicador_dam1$Inactivo)) /
  indicador_dam1$Inactivo_var

Y_tilde[,3] <- n_tilde[,3] * indicador_dam1$Inactivo
```

Calculating $\hat{Y}$ and $\hat{n}_i$

```
ni_hat <- rowSums(Y_tilde)
Y_hat[,1] <- ni_hat * indicador_dam1$Ocupado
Y_hat[,2] <- ni_hat * indicador_dam1$Desocupado
Y_hat[,3] <- ni_hat * indicador_dam1$Inactivo
Y_hat <- ceiling(Y_hat)
```

Creating a data list for STAN

```
X1_obs <- cbind(matrix(1,nrow = D,ncol = 1),X_obs)
K = ncol(X1_obs)
D1 <- nrow(X_pred)
X1_pred <- cbind(matrix(1,nrow = D1,ncol = 1),X_pred)

sample_data <- list(D = D,
                    P = P,
                    K = K,
                    y_tilde = Y_hat,
                    X_obs = X1_obs,
                    X_pred = X1_pred,
                    D1 = D1)
```

Compiling the model in STAN

```
fit_mcmc2 <- stan(
file = "www/08_Mod_Trabajo/00_Multinomial_simple_no_cor.stan",
  data = sample_data,
 verbose = TRUE,
  warmup = 1000,# number of warmup iterations per chain
  iter = 2000,  # total number of iterations per chain
  cores = 4,    # number of cores (could use one per chain)
)

saveRDS(fit_mcmc2,
        "www/08_Mod_Trabajo/fit_multinomial_no_cor.Rds")
```

Model Validation

Model validation is essential to assess its ability to predict future outcomes accurately and reliably. In the case of an area model with a multinomial response, validation focuses on measuring the model's accuracy in predicting the different response categories.

Posterior Predictive Checking

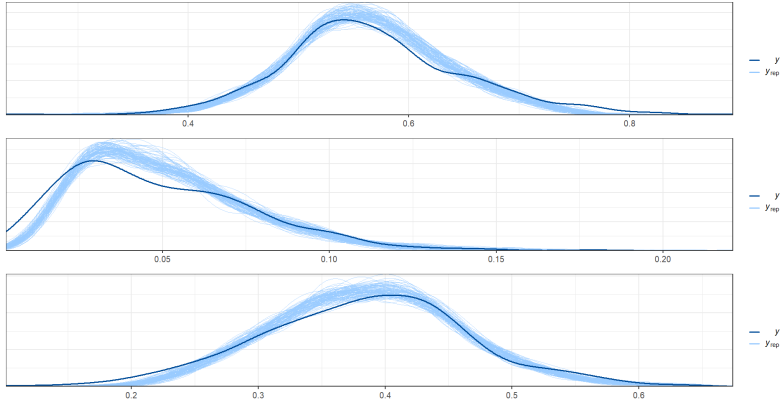


Figure 46: PPC Multinomial Model

Parameter Estimation

Parameter estimations are obtained directly from the chains generated by STAN. The process involves organizing the information to match each of the domains, and as a result, the following table is obtained:

Table 29: Multinomial Model Estimation

dam2	Ocupado_mod	Desocupado_mod	Inactivo_mod
05001	0.5773	0.0776	0.3451
05002	0.4675	0.0308	0.5017
05031	0.5401	0.0577	0.4022
05034	0.5581	0.0311	0.4108
05045	0.5037	0.0773	0.4189

Benchmarking Methodology

The benchmarking process must be performed for each of the categories following the steps that were previously carried out. The distribution of the weights is shown below.

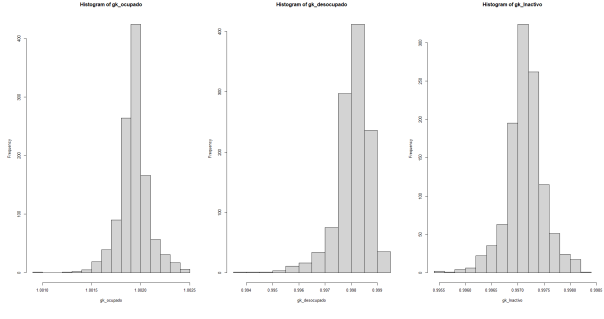


Figure 47: Weights Distribution

Labor Market Indicators Maps

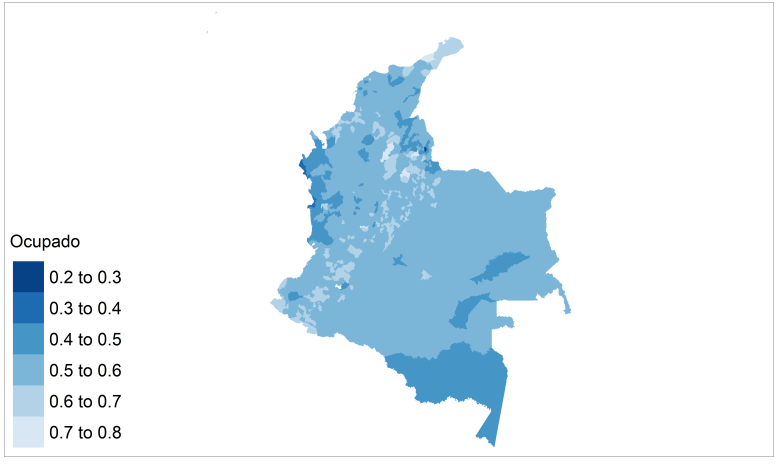


Figure 48: Employed Maps

Labor Market Indicators Maps

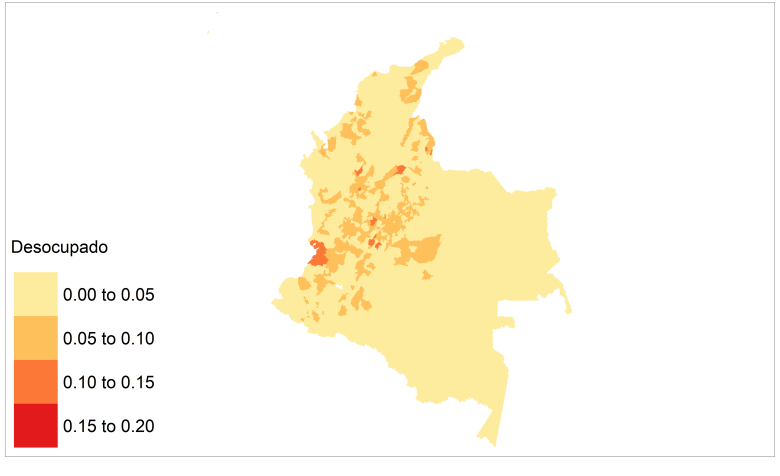


Figure 49: Unemployed Maps

Labor Market Indicators Maps

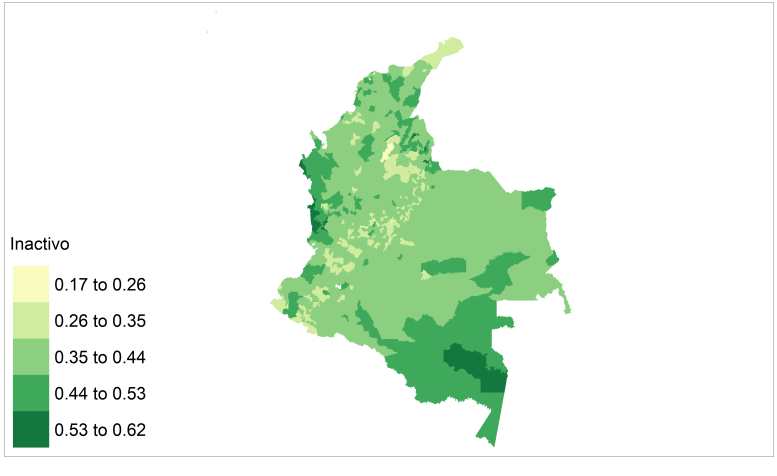


Figure 50: Inactive Maps

Thank You!

Email: andres.gutierrez@cepal.org