

Modelos multinivel con enfoque bayesiano para la estimación de la población

Andrés Gutiérrez y Stalyn Guerrero

CEPAL - Unidad de Estadísticas Sociales

Tabla de contenidos I

Introducción

Introducción a los modelos de población

Modelo multinivel para censos

Predicción de la estructura poblacional

Introducción

Por qué SAE

- ▶ Necesidad de contar con información desagregada
 - ▶ Relevante para las políticas públicas
 - ▶ Mandato ODS de “no dejar a nadie atrás”
- ▶ Más allá de lo que las encuestas permiten por sí solas
 - ▶ Generar cifras para áreas donde la encuesta no alcanza la calidad requerida
 - ▶ Generar cifras para áreas no cubiertas por la encuesta
 - ▶ Reducir el error de estimación de las cifras

Ejemplo: Número de DAM1 según calidad de las estimaciones

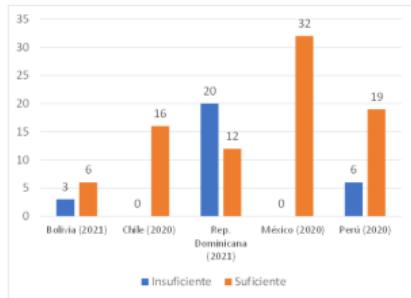


Figura 1: Tasa de desocupación, 15 y más años

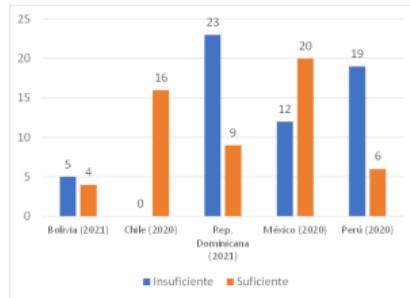


Figura 2: Tasa de desocupación femenina, 15 y más años

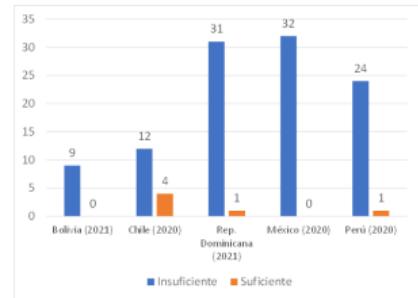


Figura 3: Tasa de desocupación femenina, 25 a 29 años

Cómo funciona SAE

“Área pequeña”:

- ▶ Dominio para el cual no es posible obtener estimaciones directas confiables
- ▶ La subpoblación de interés puede ser un área geográfica o un subgrupo socioeconómico.

Fuentes auxiliares:

- ▶ Censos, registros administrativos, datos satelitales, otras encuestas, etc., que tengan representatividad al nivel del “área pequeña” deseada.

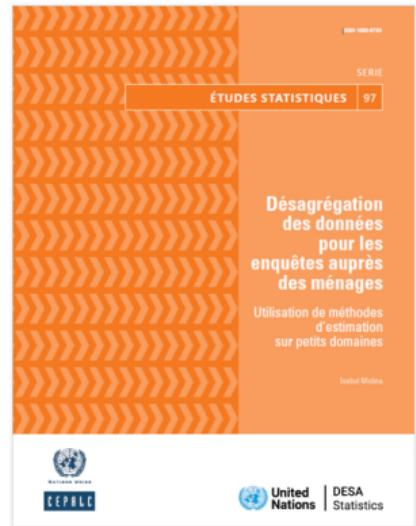
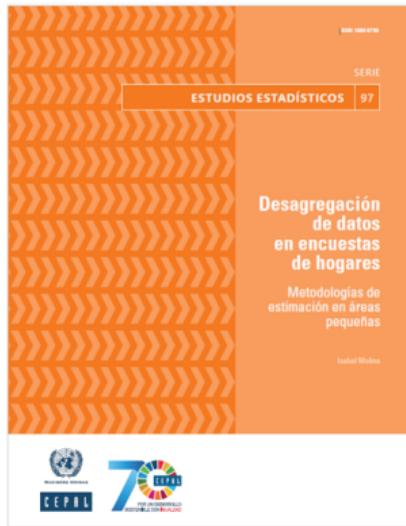
Cómo funciona SAE

Proceso

- ▶ Identificar las variables auxiliares x disponibles en la encuesta y en las fuentes auxiliares.
- ▶ Estimar un modelo para predecir la variable de interés y con la encuesta ($y = f(x)$).
- ▶ Aplicar los parámetros estimados a la fuente complementaria.
- ▶ Medir el error de estimación y evaluar la confiabilidad de los resultados.

SAE en CEPAL

- Inicio: Documento “Desagregación de datos en encuestas de hogares” (2019)
- Programación en R de los métodos SAE: pobreza extrema, pobreza e ingresos



SAE en CEPAL

- ▶ Proyecto DA13 “Innovative approaches for examining inequality through integration of different data sources in Latin America and the Caribbean” (2021-2024)
- ▶ Proyecto con UNFPA
- ▶ Cursos
 - ▶ Presenciales
 - ▶ eLearning: español + inglés (con UNSD y UNFPA)
- ▶ Asistencias técnicas
- ▶ Seminarios nacionales y regionales
- ▶ Publicaciones
- ▶ Estimaciones SAE de CEPAL

Cursos

- ▶ eLearning en español
 - ▶ 2 (2021), 1 (2022), 1 (2023)
- ▶ eLearning en inglés (con UNSD y UNFPA)

Características

- ▶ Materiales de lectura
- ▶ Videos grabados (50 videos de aproximadamente 10-15 minutos cada uno), organizados en 10 módulos
- ▶ Materiales de evaluación que incluyen evaluaciones semanales calificadas por computadora, dos proyectos de medio término y un proyecto final
- ▶ Código en lenguaje R
- ▶ Dos versiones: auto-guiado y con tutor

Asistencias técnicas

- ▶ Ecuador: tasas de desnutrición a nivel cantonal (ODS 2)
- ▶ Chile: tasas de pobreza desagregada a nivel comunal (ODS 1)
- ▶ Colombia: mapa de pobreza municipal (ODS 1)
- ▶ Chile: tasas de victimización desagregada a nivel comunal (ODS 5 y 16)
- ▶ Perú: indicadores de planificación familiar a nivel provincial (ODS 3)
- ▶ Colombia: tasas de acceso a la justicia a nivel municipal (ODS 16)
- ▶ República Dominicana: tasa de informalidad a nivel municipal (ODS 8)
- ▶ Guatemala: Proporción de personas mal nutridas (ODS 2)
- ▶ FAO LATAM: Proporción de personas en inseguridad alimentaria (ODS 2)

Seminarios

Seminarios y talleres nacionales

- ▶ Ecuador (enero 2023)
- ▶ Perú (feb 2023)
- ▶ República Dominicana (jul 2023)
- ▶ Guatemala (oct 2023)
- ▶ Chile (jul 2022)
- ▶ Uruguay (mar 2022)
- ▶ Colombia (abr 2023)
- ▶ México (nov 2023)
- ▶ Jamaica (sep 2023)

Seminarios

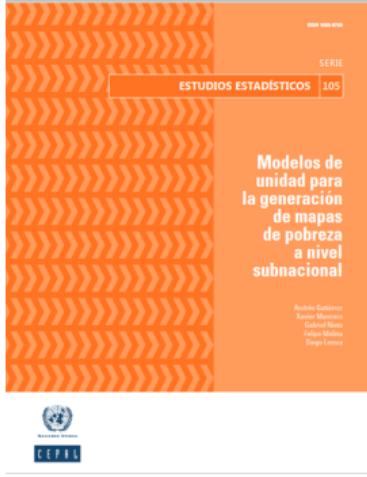
Seminarios regionales

- ▶ Nov 2018: Chile
- ▶ Jun 2023: Brasil (con Cetic.br/nic.br)

Webinars

- ▶ Mapas de pobreza mediante SAE (RTC) (abr 2021)
- ▶ Mapas de pobreza mediante SAE (con ISWGHS) (jul 2021)
- ▶ Taller de expertos (con ISWGHS) (nov 2023)

Publicaciones



Poverty mapping in Latin America: ECLAC experiences on small area estimation

Aníbal Gutiérrez,¹ Xavier Manzana and Sabrina Gómez
Social Statistics Unit, Statistics Division, Economic Commission for Latin America and the Caribbean, Santiago, Chile

Abstract. Poverty mapping is a valuable tool for government and international organizations to identify the geographical areas where poverty is more prevalent and more severe. This approach can facilitate design and implementation of poverty reduction policies more effectively. This article presents the main experience of the Statistics Division of the UN Economic Commission for Latin America (ECLAC) in poverty mapping, based on the use of small area estimation (SAE) methods to estimate poverty at the household survey, census and satellite imagery to derive poverty estimates at the provincial and municipal levels, which could not be obtained using the traditional unit of reference.

Keywords: Poverty mapping, Poorest-EEM, small area estimation, unit-level models

1. Introduction

Most Latin American and Caribbean countries regularly implement nationally representative household surveys to measure living conditions, including poverty and income inequality. These surveys generally be disaggregated geographically by urban and rural areas, as well as by administrative units such as departments, provinces or regions. However, when it comes to poverty mapping, the small area estimation of such indicators, usually are subject to lower levels of accuracy, especially when using Small Area Estimation (SAE) models to allow obtaining such disaggregation while maintaining precision.

This article presents the main experience of the Statistics Division of the UN Economic Commission for Latin America and the Caribbean (ECLAC) in poverty mapping, based on the use of geographically disaggregated poverty indicators based on household depositary income in accordance with the

¹Corresponding author: Aníbal Gutiérrez, Social Statistics Unit, Statistics Division, Economic Commission for Latin America and the Caribbean, Dag Hammarskjöld Library, Viajante, Bogotá, Colombia. Email: anibal.gutierrez@cepal.org.ve

ISSN 0013-0803 - 105 - Author's Retention © ECLAC 2022. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, CC BY-NC 4.0.

Temas estadísticos de la CEPAL

No. 4 | Diciembre de 2021

Estimaciones subnacionales de la pobreza para América Latina

Contenidos

1. Introducción
2. Descripción de las estimaciones de pobreza en América Latina
3. Fuentes de información
4. Método de estimación
5. Procesamiento de los datos
6. Resultados

Los esfuerzos de gobierno a nivel subnacional son fundamentalmente intergubernativos para que los gobiernos puedan diseñar, indicar y monitorear de manera más efectiva las políticas de desarrollo, desigualdades y riesgos a los hogares y grupos de población en donde viven. En la presente nota se presenta la metodología de mapeo de la pobreza basada en métodos de estimación en áreas pequeñas enfocadas por la División de Estadística de la CEPAL. Los resultados principales reducen la imprecisión de los censos y se utilizan imágenes satelitales para obtener estimaciones directas de las unidades de población y sus entornos, para las cuales las estimaciones directas que provienen de los censos no son suficientemente precisas ni tienen suficiente precisión.

4. Muestra de datos: En la presente se presentan los resultados obtenidos por este metodología para los países de Chile, Colombia y Perú.

Estimación desagregada para indicadores de planificación familiar en Colombia y Perú

Utilizando técnicas de estimación en áreas pequeñas para el caso de Perú y Colombia

Aníbal Gutiérrez
Gaelina Heredia
Felipe Molina
Sabrina Gómez
Walter Mendez
Carlos Ramírez
Juliana Gómez

CEPAL

UNFPA

Figura 4: Publicaciones

Desagregación de indicadores regionales

Fuentes de información

- ▶ Banco de Datos de Encuestas de Hogares (BADEHOG)
 - ▶ Variables armonizadas para comparabilidad regional de indicadores
- ▶ Banco de datos censales de CELADE
 - ▶ Procesamiento mediante librería REDATAM para R
- ▶ Imágenes satelitales
 - ▶ Acceso a Google Earth Engine desde R con librería rgee
 - ▶ Luces nocturnas, suelo cultivado, suelo urbano, índice de modificación humana, accesibilidad hospitalares, etc

Desagregación de indicadores regionales

Procedimiento

- ▶ Estandarización y homologación de covariables
- ▶ Actualización de conteos de población según marginales de la encuesta de hogares
- ▶ Definición de los modelos para los indicadores de interés
- ▶ Predicción en áreas pequeñas y estimación del ECM
 - ▶ Inicialmente Bootstrap
 - ▶ Actualmente modelo bayesiano
- ▶ Validación de supuestos del modelo y “benchmarking”
- ▶ Generación de mapas

Áreas en desarrollo

- ▶ Modelo SAE multinomial para indicadores de mercado laboral
 - ▶ Estimación conjunta de condición de actividad (ocupado, desocupado, fuera de la fuerza de trabajo)
- ▶ Estimación SAE para índice de pobreza multidimensional
 - ▶ Estimación de cada una de las privaciones que componen el IPM, para obtener su aporte marginal

CEPALSTAT



Figura 5: CEPALSTAT

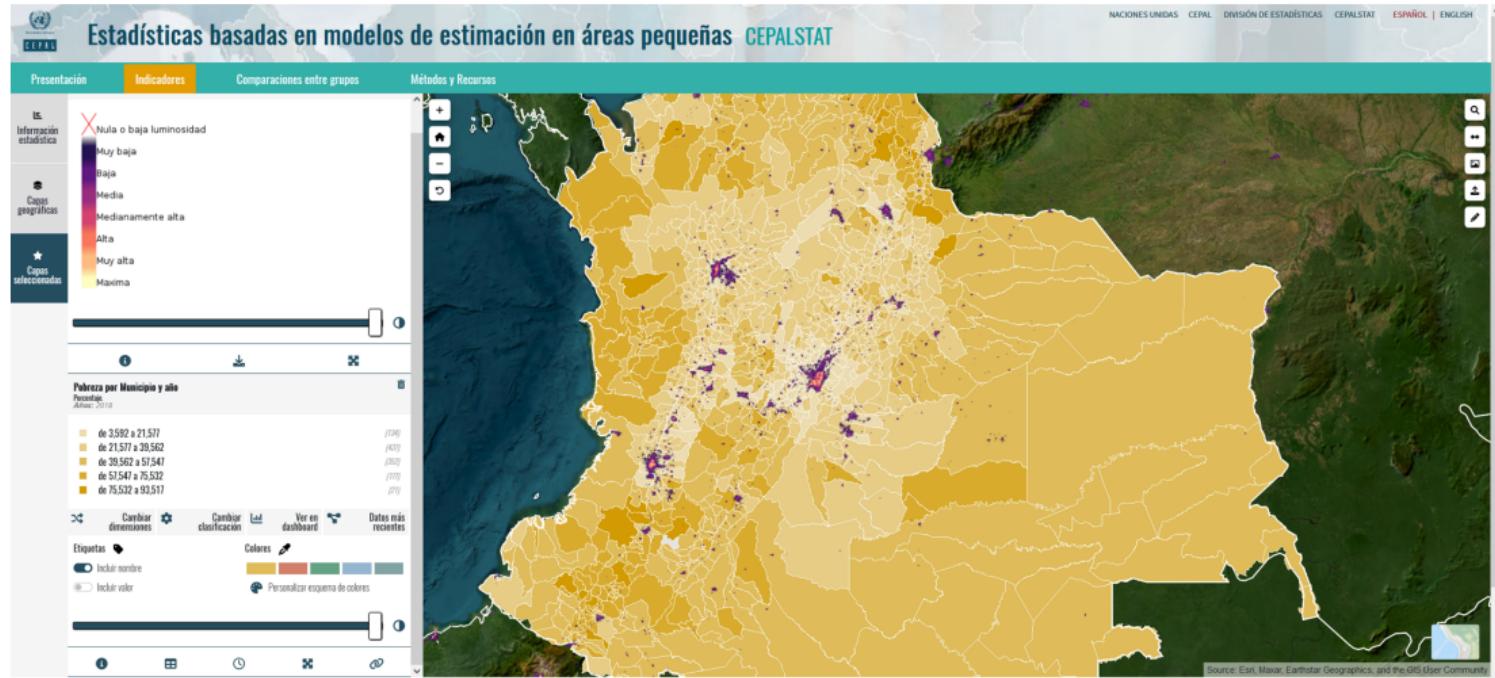


Figura 6: CEPALSTAT

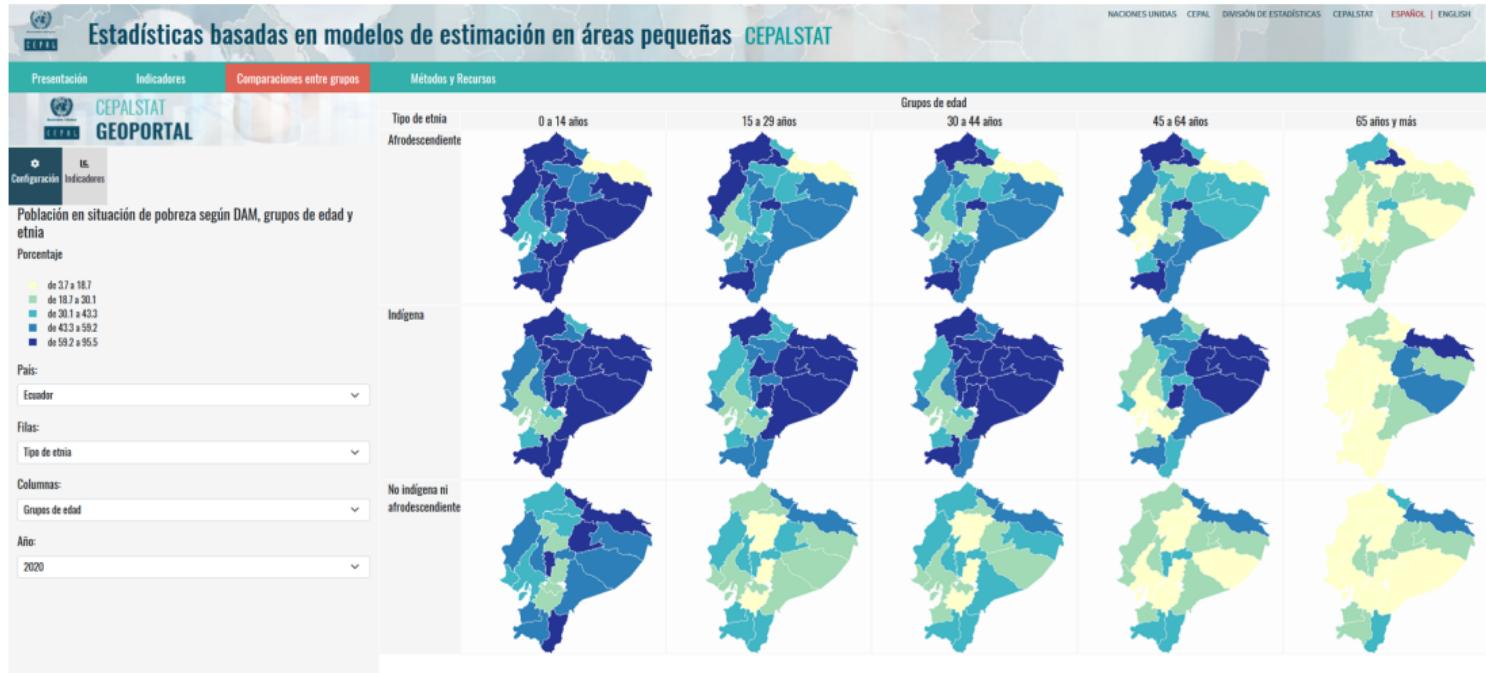


Figura 7: CEPALSTAT

Desarrollos en 2023

1. En 2023, la CEPAL se unió a la FAO para apoyar en las estimaciones de FIES en municipios de países de Latinoamérica.
 - ▶ Las estimaciones de FIES provienen de modelos de tipo Rasch.
 - ▶ Enfoques frecuentistas y bayesianos.
2. En 2023, la CEPAL recibió la visita de la OIT y acordó explorar enfoques novedosos para la estimación de los métodos de EAC de mercado y trabajo en América Latina y el Caribe.
 - ▶ Tasa de desempleo, tasa de participación, entre otros indicadores.
 - ▶ Modelos bayesianos con efectos aleatorios correlacionados.

Nuestra visión a futuro

1. La División de Estadística de la CEPAL aspira a consolidarse como la unidad líder en la desagregación de estimaciones en la región de América Latina y el Caribe (ALC).
 - ▶ Al centrarnos en mejorar la calidad y la solidez de nuestro trabajo estadístico, nos esforzamos por proporcionar información y datos valiosos que puedan informar eficazmente las políticas y los procesos de toma de decisiones.
2. Reconocemos la importancia de la colaboración y el intercambio de conocimientos, y nuestra División está comprometida a fomentar asociaciones con otros programas, fondos, agencias y partes interesadas de las Naciones Unidas.
 - ▶ Buscamos activamente oportunidades para cooperar e intercambiar conocimientos con el fin de promover la adopción e implementación de métodos SAE en todos los países de todo el mundo.

Nuestra visión a futuro

3. Nuestro objetivo es hacer que los métodos de integración de datos sean accesibles y aplicables en diferentes contextos.
- La División de Estadística de la CEPAL busca contribuir al desarrollo de marcos estadísticos integrales y precisos que faciliten el monitoreo y la evaluación de los indicadores socioeconómicos, apoyen la formulación de políticas basadas en evidencia y fomenten el desarrollo sostenible a nivel mundial.

Introducción a los modelos de población

Modelo WorldPop

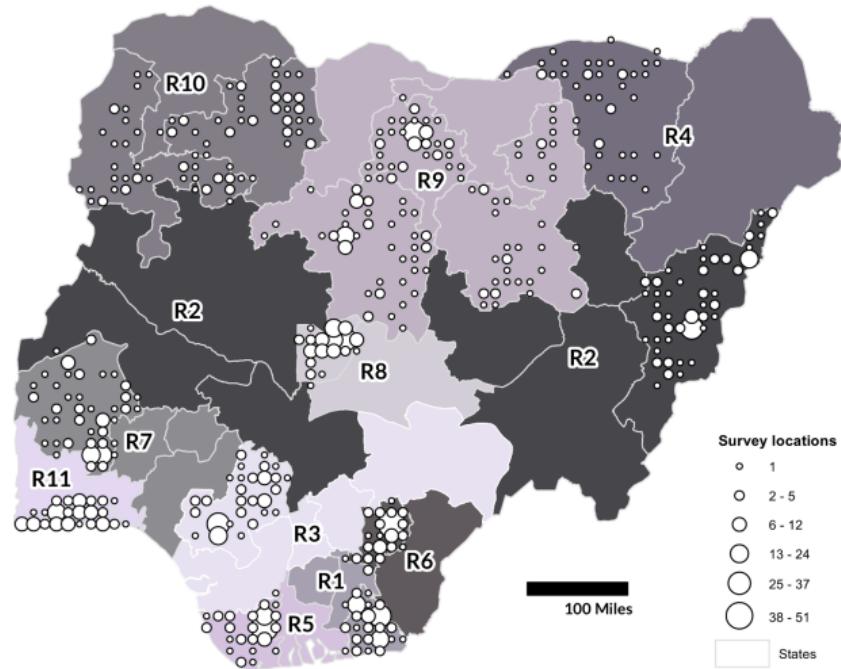


Fig. 1. Map of Nigeria showing locations of microcensus surveys as the number of survey locations within each 20-km grid cell. Labeling (R1 to R11) and shading of states indicate regions used for modeling.

Modelo WorldPop

NATURE COMMUNICATIONS | <https://doi.org/10.1038/s41467-022-29094-x>

ARTICLE

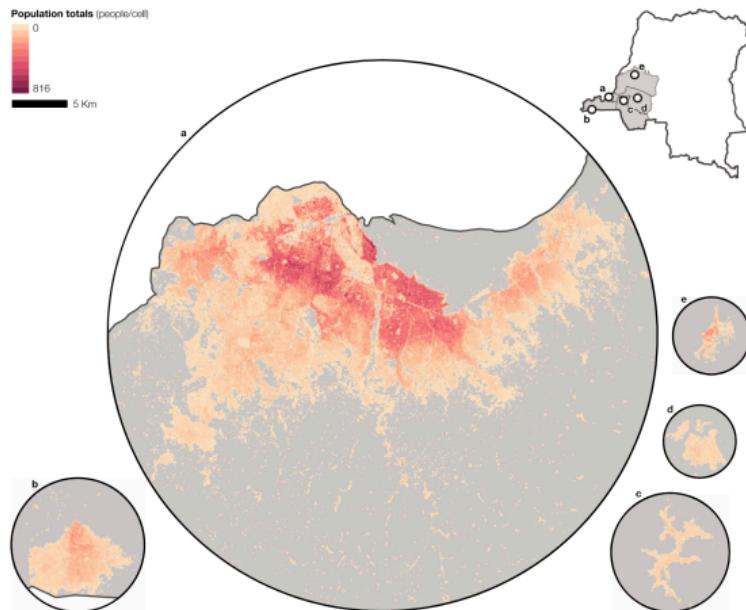


Fig. 1 Gridded population estimates in selected cities. Estimated population totals (people/cell) in the capital cities of the provinces of (a) Kinshasa (Kinshasa), (b) Kongo Central (Matadi), (c) Kwango (Kenge), (d) Kwilu (Bandundu), and (e) Mai-Ndombe (Inongo). The estimates represent the mean of the posterior distribution ($n=10,000$). The map of the DRC shows the extent of the five provinces defining the study region in gray.

Asistencias técnicas en modelación de población

- ▶ Costa Rica
- ▶ Ecuador
- ▶ República Dominicana
- ▶ Jamaica
- ▶ Barbados
- ▶ Haití

Limitaciones identificadas en la base.



Figura 8: Mapa del sector censal Rural

Limitaciones identificadas en la base.



Figura 9: Mapa del sector censal Urbano

Filtrado y Refinamiento de Datos del Censo

En el proceso de mejorar y depurar la base de datos del censo, es fundamental establecer reglas consistentes y replicables. En este contexto, el procedimiento de “Filtrado y Refinamiento de Datos del Censo” se vuelve esencial para mejorar la calidad de los datos y eliminar información irrelevante.

A continuación veremos el conjunto de filtros realizados para obtener una base de datos con información precisa y valiosa de los datos del censo.

Aplicación del primer filtro

Categorización de hogares con residentes y determinación del estado de greenpoint

1. Creamos una nueva columna llamada 'personas' para categorizar los hogares como con residentes ('si') o vacíos ('no') según el número total de residentes en cada hogar.
2. Se determina el estado de 'greenpoint' de cada hogar en función de condiciones específicas. Si el valor de 'greenpoint' es '0' y el valor de 'personas' es 'si', asignamos un valor de '1' a la columna 'greenpoint'.

greenpoint: La casa está censada en el mapa de puntos.

Características de los hogares según el estado de greenpoint:

En la siguiente tabla se resumen las características de los hogares en función de su estado de greenpoint. Calcula el número mínimo y máximo de residentes en los hogares, cuenta los valores faltantes para el número total de residentes y proporciona el recuento total de hogares para cada categoría de greenpoint.

greenpoint	min	max	num_na	total
0	0	0	0	471456
1	0	261	0	1053477
NA	NA	NA	218308	218308

Validación de entre la ocupación y el estado de greenpoint:

Se genera una tabla de contingencia para explorar la relación entre la ocupación y el estado de greenpoint. Esto proporciona una representación visual de cómo se distribuyen estas dos variables entre los hogares.

	0	1	NA
1	0	772625	0
2	471456	3853	0
3	0	92465	0
4	0	31187	0
5	0	52463	0
6	0	23845	0
7	0	2168	0
8	0	74871	0
9	0	0	218308
NA	0	0	0

Aplicación del Segundo Filtro

Aplicamos el segundo filtro para categorizar aún más los hogares en función de criterios adicionales. Introducimos el estado de “greenpoint2” para describir detalladamente el estado de los hogares basándonos en diferentes criterios, como el número de residentes, los resultados de las entrevistas y la ocupación de la vivienda.

Código para la aplicación del Segundo Filtro

```
censo2 <- censo1 %>% mutate(  
  greenpoint2 = case_when(  
  
    H01A_TOTAL_PERSONAS > 0 ~ "Censado con informacion n>0",  
  
    RESUL_ENTREVISTA_VIV %in% c(1) &  
    H01A_TOTAL_PERSONAS == 0 ~ "Censado con informacion n=0",  
  
    RESUL_ENTREVISTA_VIV %in% c(3, 4) ~ "Sin informacion pero n>0",  
  
    is.na(greenpoint) & is.na(personas) ~ "Sin informacion pero n>=0",  
  
    V02_OCUPACION_VIVIENDA == "8" ~ "Sin informacion pero n>=0",  
  
    TRUE ~ "Resto"  
  )  
)
```

Resumen de datos basados en 'greenpoint2'

Resumimos los datos basados en la variable 'greenpoint2' actualizada. Estos datos nos ayudan a comprender el impacto del filtro en la clasificación de los hogares.

greenpoint2	n	percentage
Censado con informacion n=0	175921	10.09
Censado con informacion n>0	776478	44.54
Sin informacion pero n>0	285810	16.40
Sin informacion pero n>=0	505032	28.97

Proceso de estimación en STAN

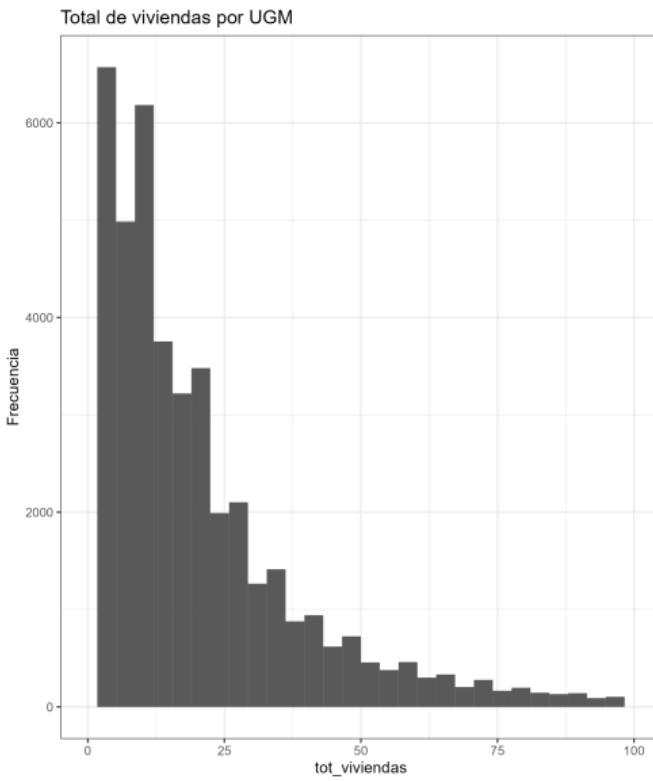
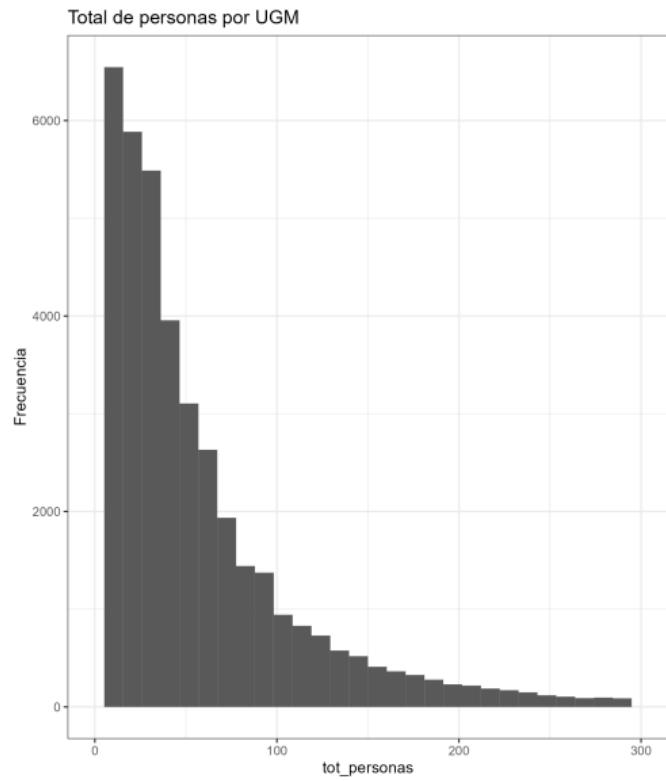
Sea Y el conteo de personas censadas por sector censal del país. Aquí incluimos las viviendas con cero personas.

```
dataPois <-  
  readRDS("Recursos/00_Intro_bayes/Poisson/conteo_viviendas.rds")
```

Tabla 1: Contenzo de personas y viviendas

DIST_ID	ID_UGM	tot_personas	tot_viviendas
10101	000001	9	2
10101	000002	0	5
10101	000003	0	1
10101	000004	0	6
10101	000005	0	1
10101	000006	0	1
10101	000007	1	1
10101	000008	0	1

Histograma con el conteo de personas



Estandarización y validación de covariables

Imagenes satelitales

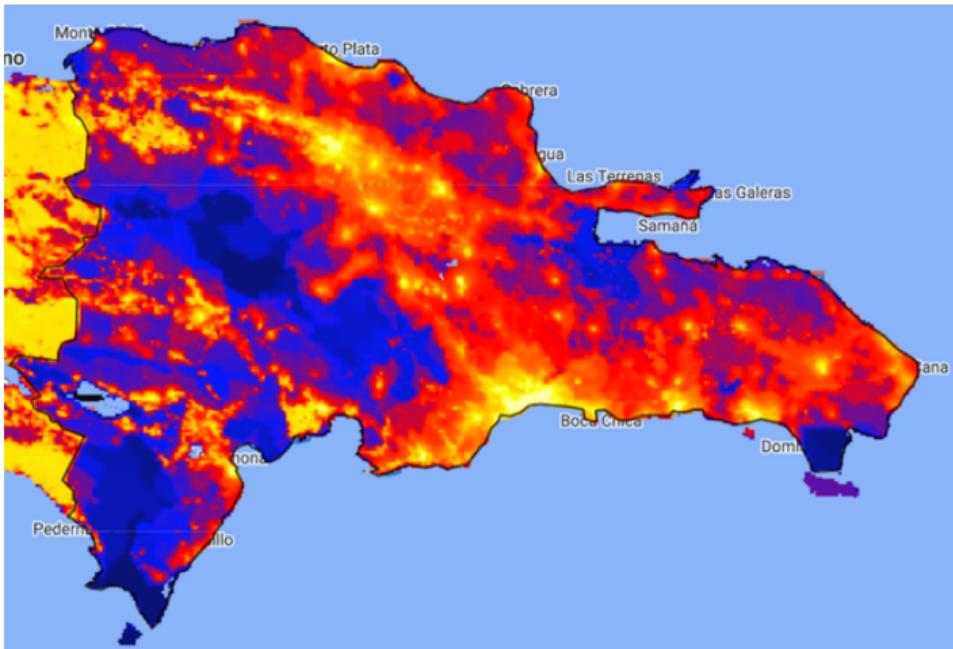


Figura 10: Modificación humana

Imagenes satelitales

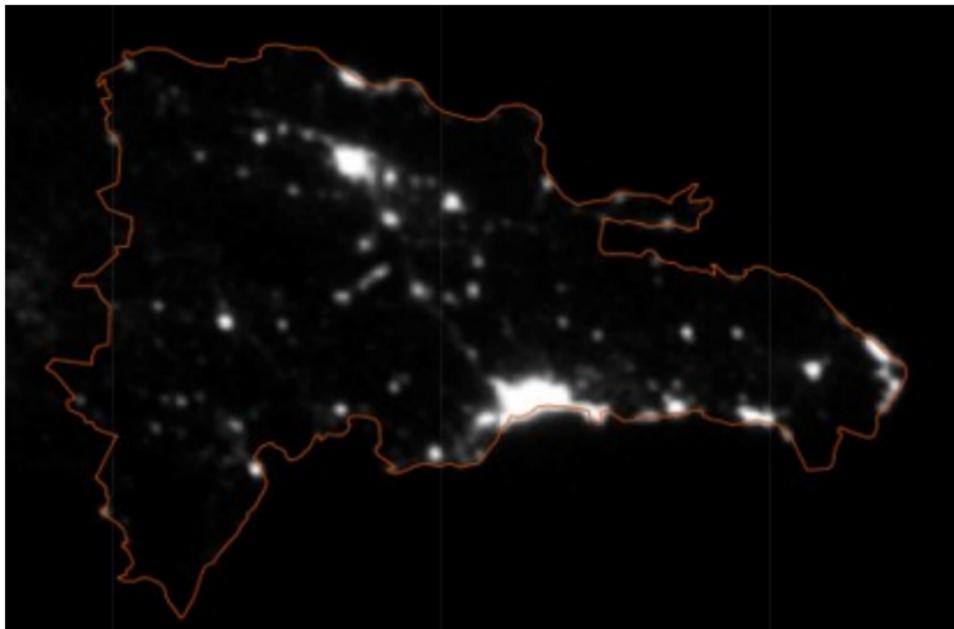


Figura 11: Luces nocturnas

Imagenes satelitales

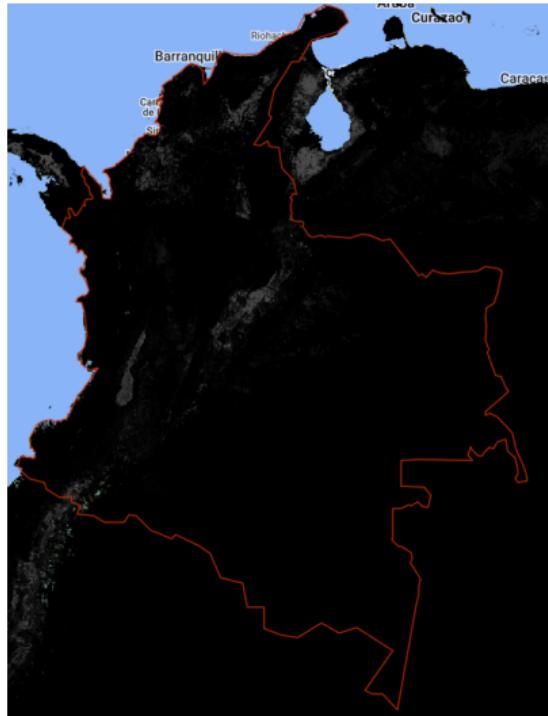


Figura 12: Suelo cultivos

Imagenes satelitales

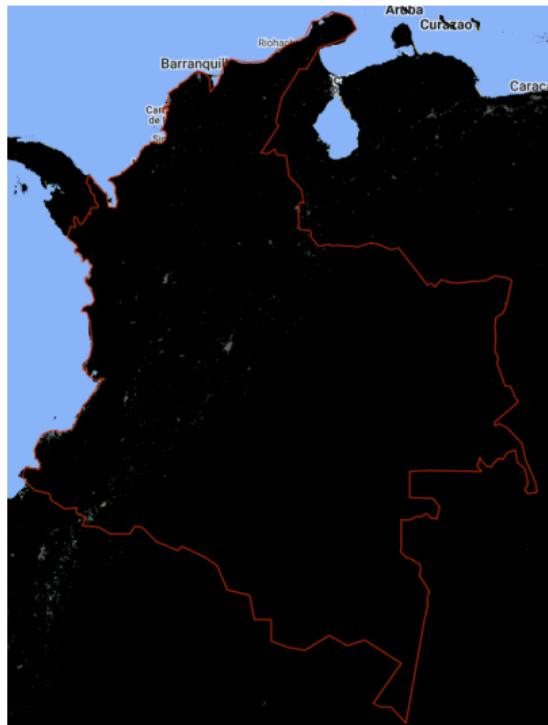


Figura 13: Suelo urbano

Listado de variables usadas en el modelo

- ▶ sector_censal_peligrosidad
- ▶ sector_censal_problema_de_acceso
- ▶ sector_censal_riesgos_amenazas
- ▶ sector_censal_cobertura_telecomunicaciones
- ▶ dist_permisos_de_construccion_2011_2022
- ▶ dist_poblacion_proyeccion_ajustada_2022
- ▶ dist_poblacion_ccss_abril_2023
- ▶ dist_matriculaEducacionPrimaria_2021
- ▶ GHS_BUILT_S_E2020_GLOBE_R2023A_5367_CRI (superficie urbanizada en m²)
- ▶ urban_coverfraction
- ▶ crops_coverfraction

Listado de variables usadas en el modelo

- ▶ asent (Asentamientos Informales)
- ▶ ppp_CRI_v2 (Densidad de población en modelo Wpop)
- ▶ elev (Altitud del área en metros sobre el nivel del mar)
- ▶ indig (Define si un área esta dentro o fuera de un área indígena)
- ▶ aprot (Define si un área esta dentro o fuera de un área protegida)
- ▶ ebais_tt (Tiempo de viaje al EBAI mas próximo en minutos)
- ▶ escu_tt (Tiempo de viaje a la escuela mas próximo en minutos)
- ▶ igl_tt (Tiempo de viaje a la iglesia as próximo en minutos)
- ▶ dist_nl_mean

Estandarización y validación de covariables

- ▶ De manera similar a como se validaron las variables del censo, el conjunto de covariables pasa por un proceso de validación. Esto implica asegurar la uniformidad en la longitud de identificadores como sector censal , Cantones, regiones, etc.
- ▶ Se lleva a cabo una validación para identificar valores faltantes (NAs) en el conjunto de datos.
- ▶ Posteriormente, se realiza un análisis descriptivo de los datos.

Estandarización y validación de covariables

- ▶ Comenzamos creando un resumen que incluye los nombres de las columnas y sus tipos de datos correspondientes.
- ▶ Luego, mejoramos este resumen añadiendo una columna que indica el tipo de datos de cada columna. Seguido de esto se crean las estadísticas de resumen como promedios, desviación estándar, máximos, mínimos y otros.
- ▶ El proceso continua con la estandarización del tipo de información (carácter o numérico)

Numéricas: Se estandarizan a escala de media cero y desviación estándar 1

Carácter: La longitud de los caracteres sea igual dentro de la variable.

Valores descriptivos de la base de sector censal

Tabla 2: Valores descriptivos de la base de sector censal (Carácter)

Nombre	Num_nas_char	leng_min	leng_max
UGM_ID	0	8	8
un_id	0	5	5
PROV_ID	0	1	1
CANT_ID	0	3	3
DIST_ID	0	5	5
ugm_peligrosidad	0	1	1
ugm_problema_de_acceso	0	1	1
ugm_riesgos_amenazas	0	1	1
ugm_cobertura_telecomunicaciones	0	1	1
asent	0	1	1
indig	0	1	1
aprot	0	1	1
dist_codigo_urbanidad	0	1	1
ugm_sin_info	0	1	1

Valores descriptivos de la base de sector censal

Tabla 3: Valores descriptivos de la base de sector censal (Numéricas)

Nombre	Num_nas	Valor_sd	Valor_Mediana	Valor_Media	Valor_Minimo	Valor_Maximo
ugm_viviendas_totales_censo	0	1	-0.3219	0	-0.8631	10.154
ugm_viviendas_ocupadas_censo	0	1	-0.3283	0	-0.8307	11.578
ugm_viviendas_desocupadas_censo	0	1	-0.3693	0	-0.5644	24.207
ugm_area_m2	0	1	-0.1084	0	-0.1112	180.185
ppp_CRI_v2	0	1	-0.4165	0	-0.8036	8.259
elev	0	1	0.1824	0	-1.2981	4.304
dist_permisos_de_construccion_2011_2022	0	1	-0.3123	0	-1.1128	4.403
dist_poblacion_proyeccion_ajustada_2022	0	1	-0.3263	0	-1.1282	4.016
dist_poblacion_ccss_abril_2023	0	1	-0.2821	0	-1.1436	3.603
dist_matriculaEducacionPrimaria_2021	0	1	-0.3061	0	-1.1416	3.433
GHS_BUILT_S_E2020_GLOBE_R2023A_5367_CRI	0	1	-0.1111	0	-1.0875	3.290
urban_coverfraction	0	1	0.1048	0	-1.0940	1.126
crops_coverfraction	0	1	-0.6641	0	-0.6641	1.906
ebais_tt	0	1	-0.3487	0	-1.4834	17.526
escu_tt	0	1	-0.3230	0	-0.3964	30.619
igl_tt	0	1	-0.2914	0	-0.3571	33.169
prov_nl_mean	0	1	-0.1938	0	-1.4204	1.216
cant_nl_mean	0	1	-0.6202	0	-0.8261	2.203
dist_nl_mean	0	1	-0.5312	0	-1.0179	1.498
wpop_sum	0	1	-0.3126	0	-0.6765	42.657

Modelo multinivel para censos

Caso de estudio

- ▶ Los modelos que se implementaron, aprovecharon una variedad de fuentes de datos, como el pre-censo, el censo, registros administrativos e información geoespacial. Que fue preparada previamente.
- ▶ La información geoespacial se ajustó según los sectores censales, que desempeñaron un papel clave como sectores censales y áreas de empadronamiento.
- ▶ Se implementó un modelo bayesiano similar a los del Proyecto WorldPop de la Universidad de Southampton en Inglaterra para predecir el número de personas en viviendas no visitadas, ocupadas pero con habitantes ausentes o que rechazaron responder al cuestionario censal.
- ▶ Este modelo se basa en la suposición de que el número de personas en una vivienda de una sector censal sigue una distribución de Poisson.

Caso de estudio

La ecuación básica del modelo es:

$$\begin{aligned}Y_{ij} &\sim \text{Poisson}(N_{ij} \times D_j) \\ \log(D_j) &= x_{ij}^t \beta + z_{ij}^t \gamma \\ \beta &\sim N(0, 10000) \\ \gamma &\sim N(0, 10000)\end{aligned}$$

donde Y_{ij} es el número de personas que habitan en la i-ésima vivienda de la j-ésima sector censal , N_{ij} es el número de viviendas en esa UGM (conocido desde el censo y el precenso), D_j es la densidad poblacional promedio en la UGM.

Note que D_j se relaciona por medio de la función logaritmo con los correspondientes efectos fijos $x_{ij}^t \beta$ y los efectos aleatorios $z_{ij}^t \gamma$

Objetivo.

Estimar el total de personas en el país, es decir,

$$Y = \sum_U Y_d$$

donde Y_d es total de personas en la d -ésima UGM

Note que,

$$Y = \sum_{U_d} Y_d + \sum_{U_d^c} Y_d$$

Estimador de Y

El estimador de Y esta dado por:

$$\hat{Y}_d = \sum_{U_d} Y_d + \sum_{U_d^c} \hat{y}_d$$

donde

$$\hat{y}_d = E_{\mathcal{M}}(Y_d | X_d, \beta)$$

,

donde \mathcal{M} hace referencia a la medida de probabilidad inducida por el modelamiento.
De esta forma se tiene que,

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

Modelo en Stan

```
data {  
    int<lower=1> D; // Número de sector censal  
    int<lower=1> K; // Cantidad de regresores  
    int<lower=1> Kz; // Cantidad de efectos aleatorios  
    int<lower=0> Y_obs [D]; // conteos de poblacion por UGM  
    int<lower=0> V_obs [D]; // Número de viviendas censadas  
    matrix[D, K] X_obs; // matriz de covariables  
    matrix[D, Kz] Z_obs; // matriz de dummis  
}  
  
parameters {  
    vector[K] beta; // matriz de parámetros  
    vector[Kz] gamma; // Efectos aleatorios  
    real<lower=0> densidad [D];  
    real<lower=0> sigma;  
}
```

Modelo en Stan

```
transformed parameters {
  real<lower=0> lambda [D];
  vector[D] lp; // vector de parámetros

  lp = X_obs * beta + Z_obs * gamma;
  for(d in 1:D){
    lambda[d] = densidad[d] * V_obs[d];
  }
}
```

Modelo en Stan

```
model {  
    // Prior  
    gamma ~ normal(0, 10);  
    beta ~ normal(0, 1000);  
    sigma ~ inv_gamma(0.001, 0.001);  
  
    // Likelihood  
    for (d in 1:D) {  
        Y_obs[d] ~ poisson(lambda[d]);  
    }  
  
    // Log-normal distribution for densidad  
    for (d in 1:D) {  
        densidad[d] ~ lognormal(lp[d], sigma);  
    }  
}
```

Consideraciones para los modelos.

Durante el proceso de validación de la información censal se hizo la clasificación de los registros en 4 categorías, que debemos utilizar para generar resultados considerando estas clasificaciones:

Modelo 1: Considera las viviendas con información (Ocupadas y desocupadas).

- ▶ Censado con información $n=0$
- ▶ Censado con información $n>0$

Modelo 2: Considera las viviendas con personas presentas

- ▶ Censado con información $n>0$

Proceso de estimación de los modelos en R y STAN

Lectura de la información censal y las covariables que fueron previamente estandarizada y debidamente organizada.

```
censo_vivienda <-  
  readRDS("Recursos/03_Model_for_people/censo_viviendas.rds")  
Base_ugms <-  
  readRDS("Recursos/03_Model_for_people/Base_ugms_estandarizada.rds")
```

Seleccionado los datos para el Modelo 1

```
base_ugm_estima_todas <- censo_vivienda %>%  
  filter( !greenpoint2 %in% c("Sin informacion pero n>0",  
                            "Sin informacion pero n>=0")) %>%  
  group_by(UGM_ID) %>%  
  summarise(tot_personas = sum(H01A_TOTAL_PERSONAS),  
            tot_viviendas = n())  
base_ugm_estima_todas <-  
  inner_join(base_ugm_estima_todas, Base_ugms, by = "UGM_ID")
```

Preparando datos para STAN

```
Y_obs <- base_ugm_estima_todas$tot_personas  
N_obs <- base_ugm_estima_todas$tot_viviendas
```

Efectos aleatorio

```
Z_obs <- model.matrix(UGM_ID ~  
                      -1 +  
                      (PROV_ID) +  
                      (CANT_ID) +  
                      (DIST_ID) ,  
                      data = base_ugm_estima_todas)
```

Efectos Fijos

```
X_obs <- model.matrix( UGM_ID~ dist_codigo_urbanidad +
ugm_peligrosidad + ugm_problema_de_acceso +
ugm_riesgos_amenazas + ugm_cobertura_telecomunicaciones +
dist_permisos_de_construccion_2011_2022 +
dist_poblacion_proyeccion_ajustada_2022 +
dist_poblacion_ccss_abril_2023 +
dist_matriculaEducacionPrimaria_2021 + dist_codigo_urbanidad +
GHS_BUILT_S_E2020_GLOBE_R2023A_5367_CRI +
urban_coverfraction + crops_coverfraction + asent +
ppp_CRI_v2 + elev + indig + aprot + ebais_tt +
escu_tt + igl_tt + dist_nl_mean,
data = base_ugm_estima_todas
) %>% as.matrix()
```

Definiendo el sample_data para STAN

```
sample_data <- list(  
  D = nrow(X_obs) , # Número de UGM  
  K = ncol(X_obs), # Número de efectos fijos  
  Kz = ncol(Z_obs), # Número de efectos aleatorios  
  Y_obs = Y_obs,    # Conteo de personas por UGM  
  V_obs = N_obs,    # Conteo de personas Por UGM  
  X_obs = X_obs %>% as.matrix(),  
  Z_obs = Z_obs %>% as.matrix()  
)
```

Ejecutando el modelo en STAN

```
options(mc.cores = parallel::detectCores())
rstan::rstan_options(auto_write = TRUE) # speed up running time

fit_poisson_todas <- stan(
  file = "Recursos/03_Model_for_people/02_Modelo_worldpop.stan",
  # Stan program
  data = sample_data, # named list of data
  verbose = TRUE,
  warmup = 10000, # number of warmup iterations per chain
  iter = 15000,   # total number of iterations per chain
  cores = 4,      # number of cores (could use one per chain)
)

saveRDS(fit_poisson_todas,
        "Recursos/03_Model_for_people/fit_poisson_todas_worldpop.rds")
```

Seleccionando los datos para el Modelo 2

```
base_ugm_estima_ocupadas <- censo_vivienda %>%
  filter(greenpoint2 %in% c("Censado con informacion n>0",
                            "Papel n>0")) %>%
  group_by(UGM_ID) %>%
  summarise(tot_personas = sum(H01A_TOTAL_PERSONAS),
            tot_viviendas = n())

base_ugm_estima_ocupadas <-
  inner_join(base_ugm_estima_ocupadas,
             Base_ugms, by = "UGM_ID")
```

Preparando datos para STAN

```
Y_obs <- base_ugm_estima_ocupadas$tot_personas  
N_obs <- base_ugm_estima_ocupadas$tot_viviendas
```

Efectos aleatorio

```
Z_obs <- model.matrix(UGM_ID ~  
                      -1 +  
                      (PROV_ID) +  
                      (CANT_ID) +  
                      (DIST_ID) ,  
                      data = base_ugm_estima_ocupadas)
```

Efectos Fijos

```
X_obs <- model.matrix( UGM_ID~ dist_codigo_urbanidad +
ugm_peligrosidad + ugm_problema_de_acceso +
ugm_riesgos_amenazas + ugm_cobertura_telecomunicaciones +
dist_permisos_de_construccion_2011_2022 +
dist_poblacion_proyeccion_ajustada_2022 +
dist_poblacion_ccss_abril_2023 +
dist_matriculaEducacionPrimaria_2021 + dist_codigo_urbanidad +
GHS_BUILT_S_E2020_GLOBE_R2023A_5367_CRI +
urban_coverfraction + crops_coverfraction + asent +
ppp_CRI_v2 + elev + indig + aprot + ebais_tt +
escu_tt + igl_tt + dist_nl_mean,
data = base_ugm_estima_ocupadas
) %>% as.matrix()
```

Definiendo el sample_data para STAN

```
sample_data <- list(  
  D = nrow(X_obs) , # Número de UGM  
  K = ncol(X_obs), # Número de efectos fijos  
  Kz = ncol(Z_obs), # Número de efectos aleatorios  
  Y_obs = Y_obs,    # Conteo de personas por UGM  
  V_obs = N_obs,    # Conteo de personas Por UGM  
  X_obs = X_obs %>% as.matrix(),  
  Z_obs = Z_obs %>% as.matrix()  
)
```

Ejecutando el modelo en STAN

```
options(mc.cores = parallel::detectCores())
rstan::rstan_options(auto_write = TRUE) # speed up running time

fit_poisson_todas <- stan(
  file = "Recursos/03_Model_for_people/02_Modelo_worldpop.stan",
  # Stan program
  data = sample_data, # named list of data
  verbose = TRUE,
  warmup = 10000, # number of warmup iterations per chain
  iter = 15000,   # total number of iterations per chain
  cores = 4,      # number of cores (could use one per chain)
)

saveRDS(fit_poisson_todas,
        "Recursos/03_Model_for_people/fit_poisson_ocupadas_worldpop.rds")
```

Pasos para la predicción de la población

- ▶ Después de esperar un tiempo prudente (15 días o más por modelo) se procede a obtener $\hat{y}_d = E_{\mathcal{M}}(Y_d | X_d, \beta)$ para cada UGM para cada modelo.
- ▶ La predicción por vivienda se hace siguiendo las siguientes reglas

$$\hat{Y}_d = \begin{cases} \hat{y}_{mod1} & \text{greenpoint2 == "Sin informacion pero n}\geq 0\text{"} \\ \hat{y}_{mod2} & \text{greenpoint2 == "Sin informacion pero n}>0\text{"} \\ Y_d & \text{en otro caso} \end{cases}$$

- ▶ Siguiendo una regla similar se le asignan los Margenes de Error (ME)

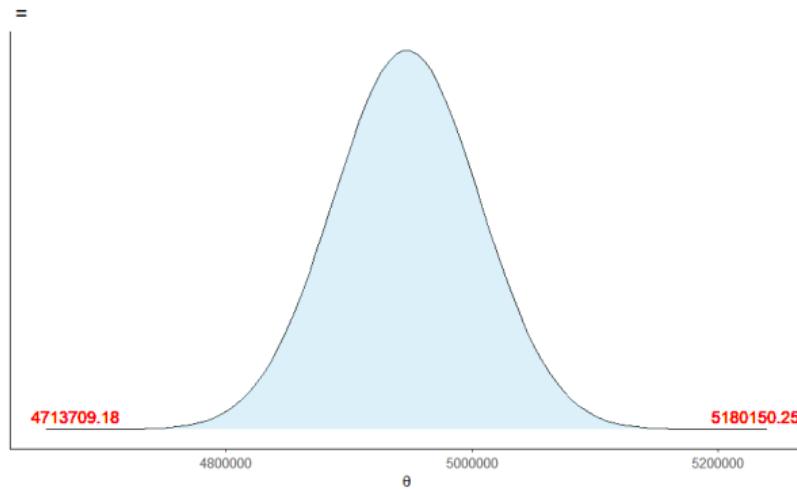
$$\hat{Y}_d^{ME} = \begin{cases} \hat{y}_{mod1}^{ME} & \text{greenpoint2 == "Sin informacion pero n}\geq 0\text{"} \\ \hat{y}_{mod2}^{ME} & \text{greenpoint2 == "Sin informacion pero n}>0\text{"} \\ 0 & \text{en otro caso} \end{cases}$$

Estimaciones agregadas

- ▶ Ahora se presenta la metodología utilizada para obtener estimaciones en varios niveles de agregación, empleando un conjunto de funciones personalizadas diseñadas para este conjunto de datos en particular.
- ▶ Estas funciones facilitan el proceso de generar predicciones y visualizaciones, lo que nos permite analizar de manera efectiva las estimaciones de población.
 - ▶ **plot_densidad:** Esta función genera un gráfico de la densidad de una distribución normal con la media y la desviación estándar especificadas. Además, resalta un intervalo específico de la distribución con un área sombreada y segmentos en el gráfico.
 - ▶ **Pred_totPob:** Esta función realiza cálculos y visualizaciones relacionados con los datos de población total en un censo.

Predicción del total poblacional

```
p1 <- Pred_totPob(censo_vivienda, agrega = NULL, Plot = TRUE)
```



total	SE	LimInf	LimSup	Len_IC
4946930	142208	4713709	5180150	466441

Predicción del total por provincia

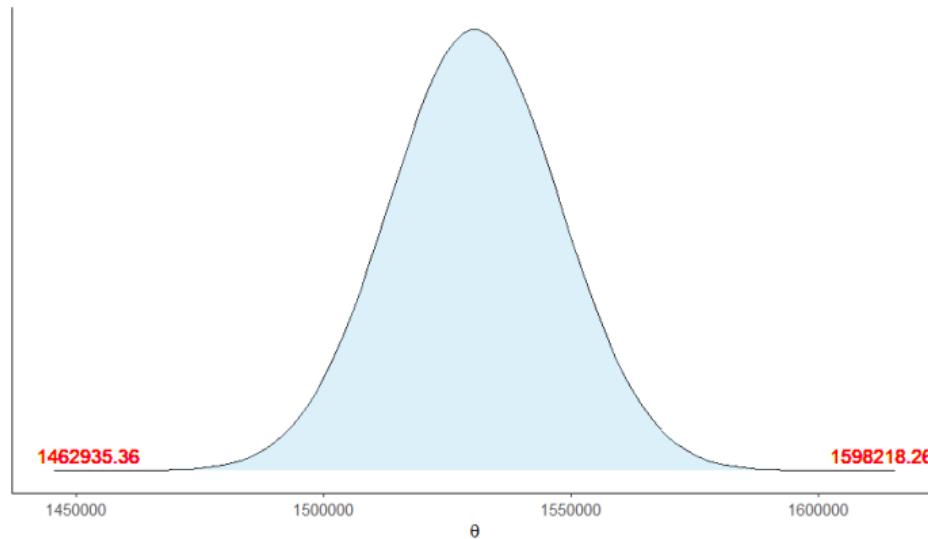
```
tab <- Pred_totPob(censo_vivienda, agrega = "PROV_ID", Plot = FALSE)
```

PROV_ID	total	SE	LimInf	LimSup	Len_IC
1	1530577	41245	1462935	1598218	135283
2	1081286	34928	1024003	1138568	114565
3	556988	12840	535930	578046	42116
4	488545	19040	457320	519770	62451
5	376403	11949	356808	395999	39192
6	462539	12164	442590	482487	39897
7	450592	10042	434123	467061	32938

Predicción del total por provincia

```
p1 <- Pred_totPob(censo_vivienda, agrega = "PROV_ID", filtro = "1",
                    Plot = TRUE)
```

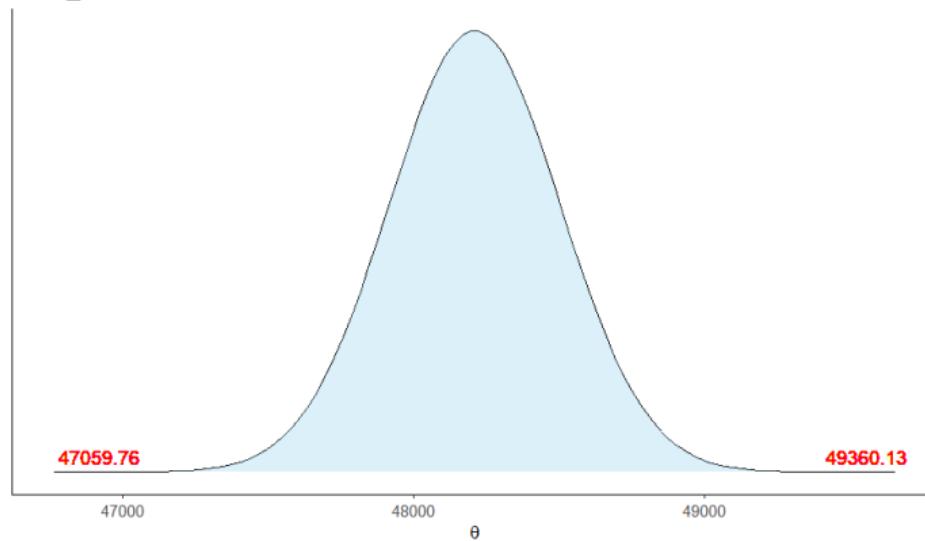
PROV_ID = 1



Predicción del total por distrito

```
p1 <- Pred_totPob(censo_vivienda, agregar = "DIST_ID",  
                    Plot = TRUE, filtro = "10110")
```

DIST_ID = 10110



Predicción de la estructura poblacional

Modelo multinomial para estimar personas por rango de edad y sexo

En el paso final, modelamos la estructura demográfica de las Unidades Geográficas Pequeñas a nivel de provincia. Esta estructura consta de 40 subgrupos basados en la combinación de género y grupos de edad. Para estimar la demografía de hogares no observados, utilizamos la predicción condicional basada en los resultados del paso anterior.

Modelo multinomial para estimar personas por rango de edad y sexo

Dada la naturaleza del fenómeno en estudio, se considera adecuada una distribución multinomial para el recuento de individuos en los 40 grupos. el modelo utilizado es el siguiente:

- ▶ G_i sigue una distribución Multinomial $Multinomial(p_{G_i})$.
- ▶ Las log-odds de p_{G_i} están modeladas por $x_i^T \beta$.

En este contexto, G_i representa el recuento de cada uno de los cuarenta grupos demográficos. p_{G_i} es un vector de tamaño 40 que contiene las probabilidades de que una persona clasificada en cada grupo resida en un hogar.

Modelo multinomial: Procesamiento en R

El código elimina todos los objetos del entorno de R actual, asegurando un punto de partida limpio para las operaciones posteriores.

```
library(tidyverse)      # Data manipulation and visualization
library(data.table)     # Fast data manipulation
library(openxlsx)       # Reading Excel files
library(magrittr)        # Pipe operator
library(lme4)           # For fitting linear mixed-effects models
library(nnet)            # For fitting neural networks
cat("\f")    # Clears console output
```

Modelo multinomial: Lectura de datos en R

Para la lectura de los datos hacemos uso del siguiente código

```
censo_vivienda <-  
  readRDS("Recursos/04_Multinomial/02_censo_vivienda_personas.rds")
```

Ahora, filtrar los datos del censo para conservar columnas relevantes para el análisis de edad y género.

```
censo_vivienda_age_gender <- censo_vivienda %>%  
  filter( !greenpoint2 %in% c("Sin informacion pero n>0",  
                            "Sin informacion pero n>=0")) %>%  
  dplyr::select(PROV_ID, HOMBRES_GRUPO1_sum:MUJERES_GRUPO20_sum)
```

Modelo multinomial: Preparación de datos:

El código crea un nuevo conjunto de datos, `censo_vivienda_edad_sexo`, filtrando entradas específicas de `censo_vivienda`. Se seleccionan columnas relacionadas con grupos de edad y género y provincias. Luego, este conjunto de datos filtrado se agrega a nivel de `PROV_ID` utilizando la función `summarise_if`.

```
# Summarizing the age and gender data by grouping  
# it based on province (PROV_ID).  
censo_personas <- censo_vivienda_age_gender %>%  
  group_by(PROV_ID) %>%  
  summarise_if(is.numeric, sum)
```

Modelo multinomial

Se crea un modelo multinomial utilizando la función `multinom`. Este modelo predice la distribución de grupos de edad y género dentro de los hogares según la provincia (`PROV_ID`). El modelo se almacena en la variable `model`.

```
# Fitting a multinomial model to estimate the
# distribution of age and gender within households,
# using province as the predictor.
model <- multinom(
  censo_personas[,-1] %>% as.matrix() ~ censo_personas$PROV_ID)

saveRDS(model, "Recursos/04_Multinomial/Multinomial_model.rds")
```

Modelo multinomial: Predicción

La función `predict` se utiliza para predecir las probabilidades de distribución para el modelo multinomial. Los resultados de la predicción no se muestran aquí, pero se pueden obtener utilizando la función `predict`.

```
model <- readRDS("Recursos/04_Multinomial/Multinomial_model.rds")
predict(model, type = "probs") %>%
  data.frame() %>% select(1:3) %>% tba()
```

HOMBRES_GRUPO1_sum	HOMBRES_GRUPO2_sum	HOMBRES_GRUPO3_sum
0.0360	0.0332	0.0364
0.0195	0.0311	0.0370
0.0158	0.0311	0.0332
0.0161	0.0274	0.0320
0.0255	0.0323	0.0389
0.0243	0.0330	0.0379
0.0264	0.0412	0.0459

Construcción de la base censal con predicciones del modelo multinomial

Para la creación de la base se siguen los siguientes pasos:

1. **Lectura de Datos del Censo:** Lee los datos del censo de viviendas y personas desde un archivo en formato RDS llamado “02_censo_vivienda_personas.rds”.
2. **Lectura del Modelo Multinomial Previamente Creado:** Lee el modelo multinomial previamente creado desde un archivo en formato RDS llamado “Multinomial_model.rds”.
3. **Cálculo de Probabilidades:** Calcula las probabilidades de pertenecer a cada categoría de resultado utilizando el modelo multinomial. Estas probabilidades se almacenan en un nuevo DataFrame llamado “probabilidad”.

Construcción de la base censal con predicciones del modelo multinomial

4. **Copia de los Datos del Censo:** Crea una copia de los datos del censo de viviendas y personas en un nuevo `data.frame` llamado “`censo_vivienda_pred`”. Esto se hace para trabajar con una copia de los datos originales sin alterar los datos originales.
5. **Cálculo de la Suma de Recuentos Predichos:** Calcula la suma de los recuentos predichos de personas en el censo y la muestra. Este cálculo utiliza la columna “`pred_conteos`” en el `data.frame` “`censo_vivienda_pred`”.
6. **Identificación de Nombres de Columnas de Grupos de Edad:** Identifica las columnas en el `data.frame` “`censo_vivienda_pred`” que corresponden a los grupos de edad. Estas columnas se almacenan en la variable “`var_grupo`” para su posterior uso.

Código de R para la creación de la base censal.

```
censo_vivienda <-  
  readRDS("Recursos/04_Multinomial/02_censo_vivienda_personas.rds")  
  
# Reading the previously created multinomial model.  
model <- readRDS("Recursos/04_Multinomial/Multinomial_model.rds")  
  
# Calculate probabilities for each outcome category using the model.  
probabilidad <-  
  predict(model, type = "probs") %>% as.data.frame() %>%  
  select_all(~paste0(.,"_prob")) %>%  
  mutate(PROV_ID = as.character(1:7))
```

Código de R para la creación de la base censal.

```
# Create a copy of the census data to work with.  
censo_vivienda_pred <- censo_vivienda  
  
# Calculate the sum of predicted counts.  
sum(censo_vivienda_pred$pred_conteos)  
  
# Identify column names corresponding to age groups.  
var_grupo <- grep(x = names(censo_vivienda_pred),  
                    pattern = "*_GRUPO\\d{,2}_sum$",  
                    value = TRUE)
```

Resultados para la Provincia 1

Filtrado de Datos del Censo para PROV_ID = 1 y Categorías Específicas de ‘greenpoint2’: Filtra los datos del censo para aquellas observaciones donde el valor de la columna “PROV_ID” es igual a “1” y la columna “greenpoint2” tiene uno de los siguientes valores: “Sin informacion pero n>=0” o “Sin informacion pero n>0”. Los resultados de este filtro se almacenan en un nuevo DataFrame llamado “PROV_1”.

```
# Filter census data for PROV_ID = 1 and specific
# greenpoint2 categories.
PROV_1 <- censo_vivienda_pred %>% filter(
  PROV_ID == "1" ,
  greenpoint2 %in% c("Sin informacion pero n>=0",
                     "Sin informacion pero  n>0")
)
```

Resultados para la Provincia 1

Cálculo de Recuentos Predichos para Grupos de Edad en PROV_1: Calcula los recuentos predichos para cada grupo de edad en “PROV_1”. Utiliza la matriz de recuentos predichos “pred_conteos” en “PROV_1” y multiplica esta matriz por la matriz de probabilidades de pertenecer a cada grupo de edad específico en la provincia 1. Los resultados se almacenan en las columnas correspondientes a grupos de edad.

```
# Calculate predicted counts for each age group in PROV_1.  
PROV_1[, var_grupo] <-  
  matrix(PREV_1$pred_conteos, nrow = nrow(PREV_1)) %*%  
  matrix(as.numeric(probabilidad[1, paste0(var_grupo, "_prob")]),  
        ncol = 40)
```

Resultados para la Provincia 1

Cálculo del Límite Inferior de Recuentos Predichos para Grupos de Edad en PROV_1: Calcula el límite inferior de los recuentos predichos para cada grupo de edad en “PROV_1”. Utiliza la matriz de límites inferiores “MEInf_pred_conteos” en “PROV_1” y realiza una operación similar a la anterior para obtener los recuentos inferiores para cada grupo de edad.

```
# Calculate lower bound of predicted counts for each
# age group in PROV_1.
PROV_1[, paste0(var_grupo, "_MEInf")] <-
  matrix(PROV_1$MEInf_pred_conteos, nrow = nrow(PROV_1)) %*%
  matrix(as.numeric(probabilidad[1, paste0(var_grupo, "_prob")]), 
        ncol = 40)
```

Resultados para la Provincia 1

Cálculo del Límite Superior de Recuentos Predichos para Grupos de Edad en PROV_1: Calcula el límite superior de los recuentos predichos para cada grupo de edad en “PROV_1”. Utiliza la matriz de límites superiores “MESup_pred_conteos” en “PROV_1” y realiza una operación similar a la anterior para obtener los recuentos superiores para cada grupo de edad.

```
# Calculate upper bound of predicted counts for
# each age group in PROV_1.
PROV_1[, paste0(var_grupo, "_MESup")] <-
  matrix(PROV_1$MESup_pred_conteos, nrow = nrow(PROV_1)) %*%
  matrix(as.numeric(probabilidad[1, paste0(var_grupo, "_prob")]),
        ncol = 40)
```

Ten en cuenta que el código está etiquetado se repite para cada una de las provincias.

Consolidando la base con las provincias

Combinar los resultados de las provincias con información faltante.

```
prov_sin_informacion <-  
  list(  
    PROV_1,  
    PROV_2,  
    PROV_3,  
    PROV_4,  
    PROV_5,  
    PROV_6,  
    PROV_7) %>% bind_rows()
```

Validaciones

1. Filtrado y Recopilación de Datos para Todas las Provincias con Información Faltante (Missing Information)

```
# Filter and gather data for all provinces with missing information
PROV_todas <-
  censo_vivienda_pred %>%
  filter(greenpoint2 %in% c("Sin informacion pero n>=0",
                            "Sin informacion pero n>0"))
```

2. Cálculo de Sumas por Fila y Suma Total de Recuentos Predichos para Provincias con Información Faltante

```
# Calculate row sums and total sum of predicted counts for provinces
# with missing information
rowSums(prov_sin_informacion[, var_grupo])
sum(prov_sin_informacion[, var_grupo])
sum(PROV_todas$pred_conteos)
```

Validaciones

3. Filtrado de Datos para Provincias con Información del Censo Completa

(Censada): Filtra los datos de censo para todas las provincias donde el valor de la columna “greenpoint2” no corresponde a “Sin informacion pero n \geq 0” ni “Sin informacion pero n $>$ 0”. Los resultados se almacenan en un nuevo DataFrame llamado “PROV_censada”.

Validaciones

- ▶ Inicializar columnas para los límites inferior y superior de los recuentos previstos

```
# in provinces with complete census
PROV_censada[,paste0(var_grupo, "_MEInf")] <- 0
PROV_censada[,paste0(var_grupo, "_MESup")] <- 0

# Combine data frames of provinces with complete and missing
# information for each age group
censo_vivienda_grupo_edad <-
  bind_rows(PROV_censada, prov_sin_informacion) %>%
  dplyr::select(un_ID,
                var_grupo,
                paste0(var_grupo, "_MEInf"),
                paste0(var_grupo, "_MESup"))
```

Guardar resultados

```
# Inner join the census data with the grouped age data and  
# save the result  
readRDS("Recursos/04_Multinomial/04_censo_vivienda_personas.rds") %>%  
  inner_join(censo_vivienda_grupo_edad) %>%  
  saveRDS(  
    "Recursos/04_Multinomial/05_censo_vivienda_personas_grupo_edad.rds")
```

Estimaciones Agregadas modelo multinomial.

Para el proceso se crearon las siguientes funciones:

- ▶ **plot_piramide_pob**: Genera un gráfico de pirámide de población con barras e intervalos de confianza.
- ▶ **piramide_pob**: Calcula y visualiza la pirámide de población a partir de los datos del censo.

```
source("Recursos/04_Multinomial/01_Agregados.R")
```

```
censo_vivienda <-
```

```
  readRDS("Recursos/04_Multinomial/05_censo_vivienda_personas_grupo_edad.rds")
```

Predicción del total de personas por rango de edad y sexo

```
p1 <- piramide_pob(  
  censo = censo_vivienda,  
  Plot = FALSE,  
  agrega = NULL,  
  filtro = NULL  
) %>% head(15) %>% tba()  
saveRDS(p1, "Recursos/04_Multinomial/04a_piramide.rds")
```

Predicción del total de personas por rango de edad y sexo

grupo	total	SE	LimInf	LimSup	Len_IC
HOMBRES_GRUPO1	132716	3539	126912	138519	11606
HOMBRES_GRUPO2	167537	4578	160029	175045	15016
HOMBRES_GRUPO3	182328	5217	173772	190884	17112
HOMBRES_GRUPO4	180166	5189	171656	188676	17020
HOMBRES_GRUPO5	191782	5695	182442	201123	18681
HOMBRES_GRUPO6	196682	5950	186923	206441	19518
HOMBRES_GRUPO7	197411	5831	187848	206974	19127
HOMBRES_GRUPO8	191008	5695	181669	200348	18678
HOMBRES_GRUPO9	156523	4346	149395	163650	14255
HOMBRES_GRUPO10	136570	3926	130131	143008	12877
HOMBRES_GRUPO11	135995	3837	129702	142289	12587
HOMBRES_GRUPO12	131889	3723	125784	137994	12211
HOMBRES_GRUPO13	124940	3636	118977	130903	11927
HOMBRES_GRUPO14	106462	3190	101230	111694	10464
HOMBRES_GRUPO15	72515	2030	69185	75844	6659

Predicción del total de personas por rango de edad y sexo

```
p1 <- piramide_pob(  
  censo = censo_vivienda,  
  Plot = TRUE,  
  agrega = NULL,  
  filtro = NULL)
```

Predicción del total de personas por rango de edad y sexo

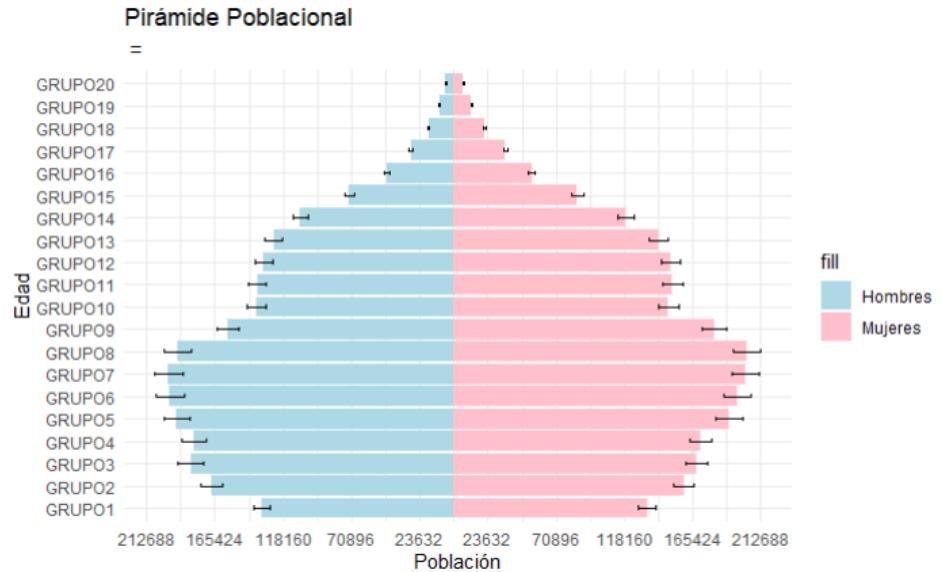


Figura 14: Resultado nacional

Predicción del total de personas por rango de edad y sexo

```
p1 <- piramide_pob(  
  censo = censo_vivienda,  
  Plot = TRUE,  
  agrega = "PROV_ID",  
  filtro = "1"  
)
```

Predicción del total de personas por rango de edad y sexo

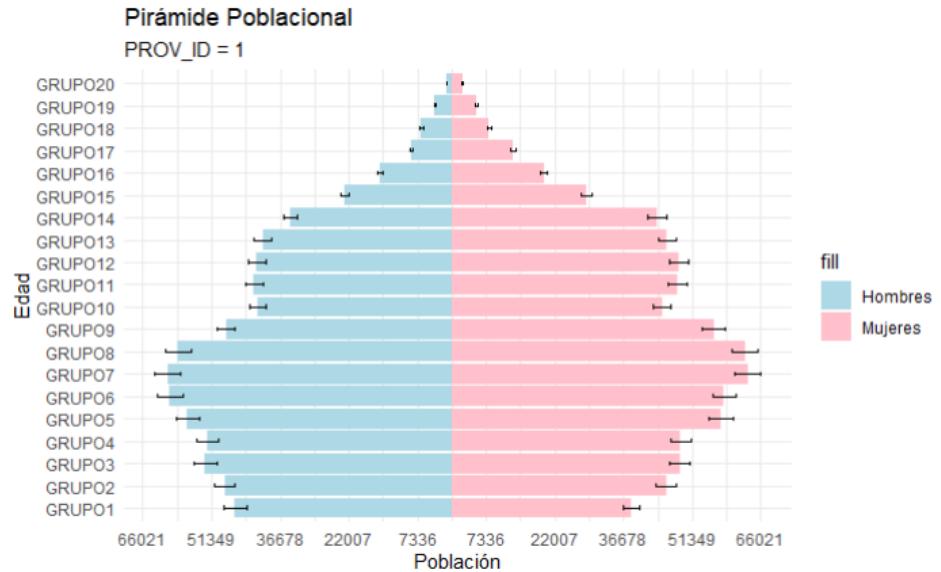


Figura 15: Resultado para la provincia 1

Predicción del total de personas por rango de edad y sexo

```
p1 <- piramide_pob(  
  censo = censo_vivienda,  
  Plot = TRUE,  
  agrega = "CANT_ID",  
  filtro = "101"  
)
```

Predicción del total de personas por rango de edad y sexo

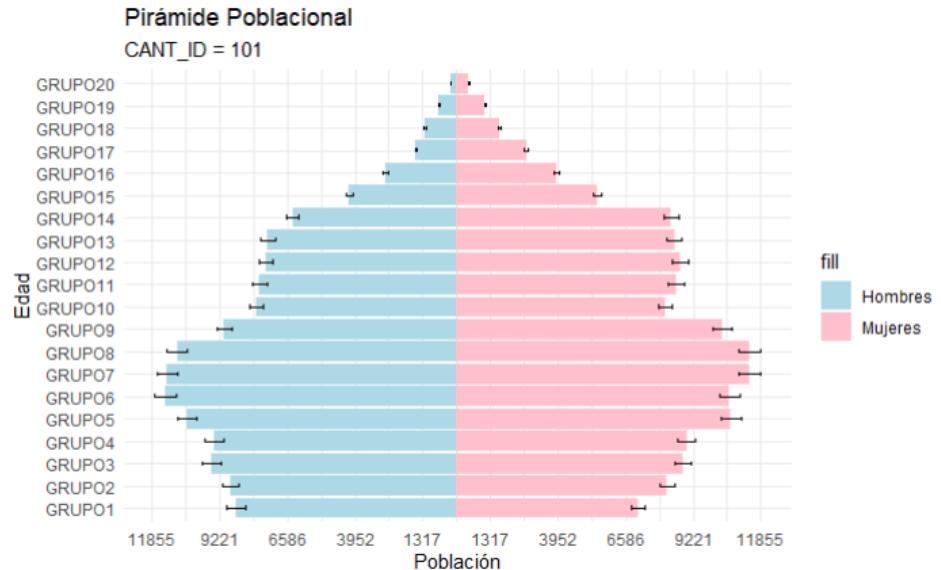


Figura 16: Resultado para la canton 101

¡Gracias!

Email: andres.gutierrez@cepal.org