

# Modelos multinivel con enfoque bayesiano para la estimación de la población

Andrés Gutiérrez

CEPAL - Unidad de Estadísticas Sociales

# Tabla de contenidos I

Introducción

Introducción al pensamiento bayesiano.

Base de datos Censo de Costa Rica

Estandarización y validación de covariables

Modelo multinivel para censos

# 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 / Example: Number of MAD1 according to quality of estimates

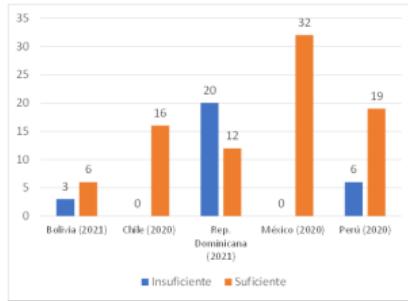


Figura 1: Tasa de desocupación, 15 y más años

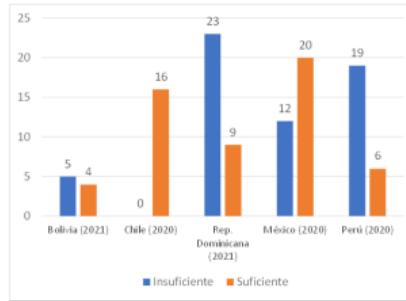


Figura 2: Tasa de desocupación femenina, 15 y más años

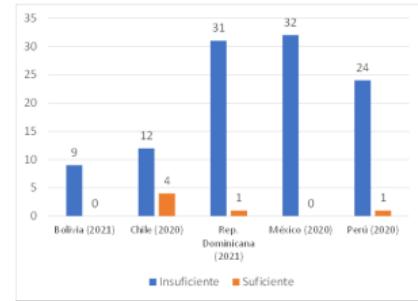


Figura 3: Tasa de desocupación femenina, 25 a 29 años

# 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:

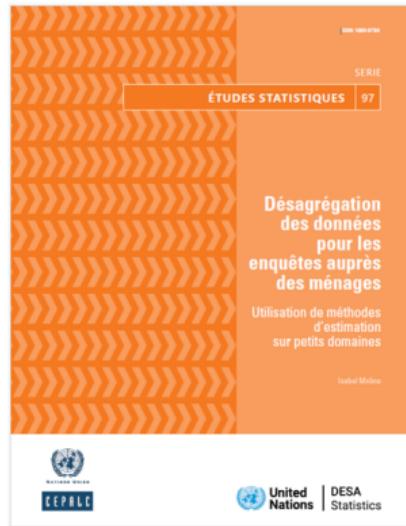
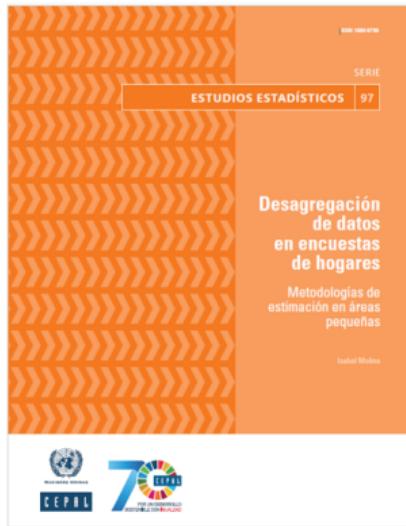
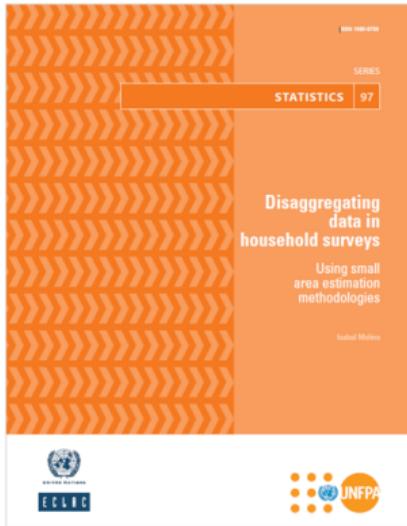
- ▶ Censos, registros administrativos, datos satelitales, otras encuestas, etc., que tengan representatividad al nivel del “área pequeña” deseada.

## 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

Algunos ejemplos:

- ▶ Ecuador: tasas de desnutrición a nivel cantonal (ODS 2)
  - ▶ Modelos FH, BHF
- ▶ Chile: tasas de pobreza desagregada a nivel comunal (ODS 1)
  - ▶ Modelo FH
- ▶ Colombia: mapa de pobreza municipal (ODS 1)
  - ▶ Modelo Census Best (Guadarrama)
- ▶ Chile: tasas de victimización desagregada a nivel comunal (ODS 5 y 16)
  - ▶ Modelo FH, EBP
- ▶ Perú: indicadores de planificación familiar a nivel provincial (ODS 3)
  - ▶ Modelo EBP
- ▶ Colombia: tasas de acceso a la justicia a nivel municipal (ODS 16)
  - ▶ Modelo FH

# Seminarios

## **Seminarios y talleres nacionales**

- ▶ Chile (nov 2022)
- ▶ Uruguay (nov 2022)
- ▶ Colombia (abr 2023)
- ▶ México (nov 2023)

## **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



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



Figura 5: CEPALSTAT

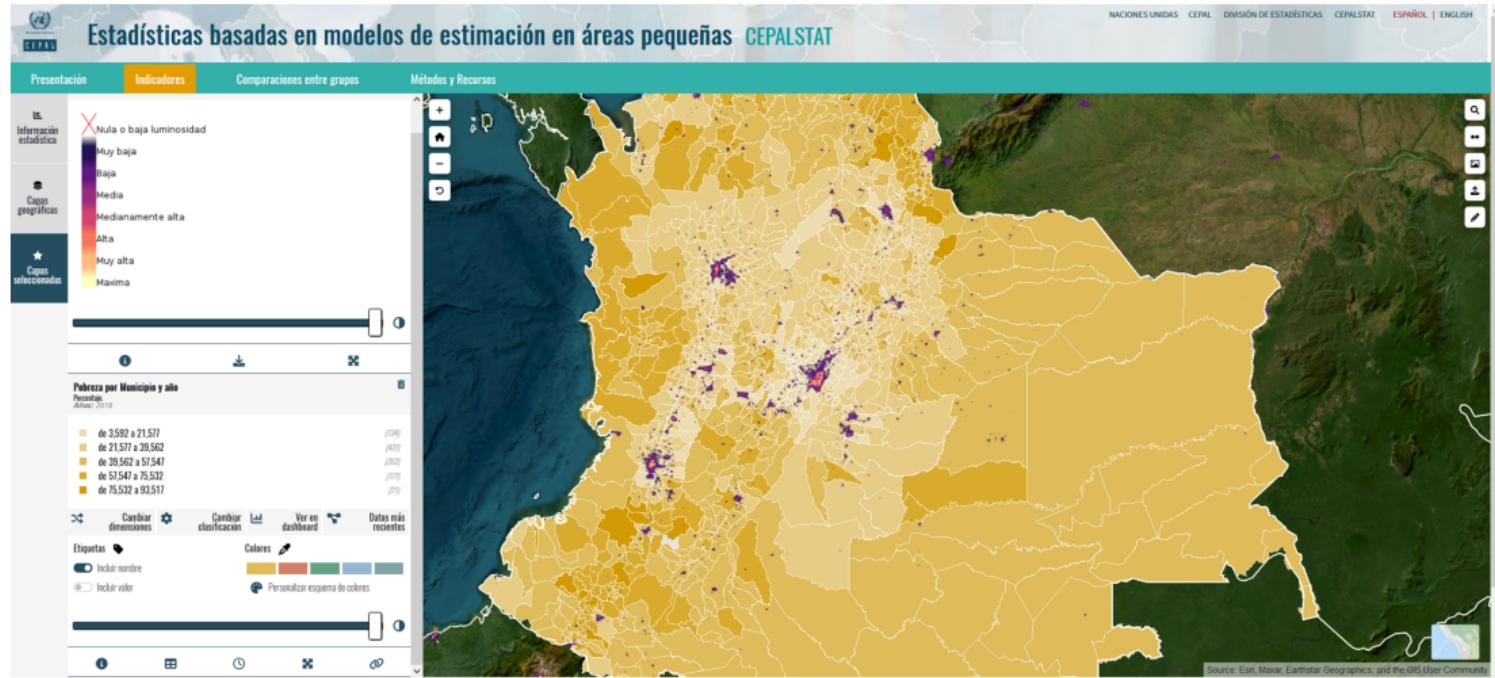


Figura 6: CEPALSTAT

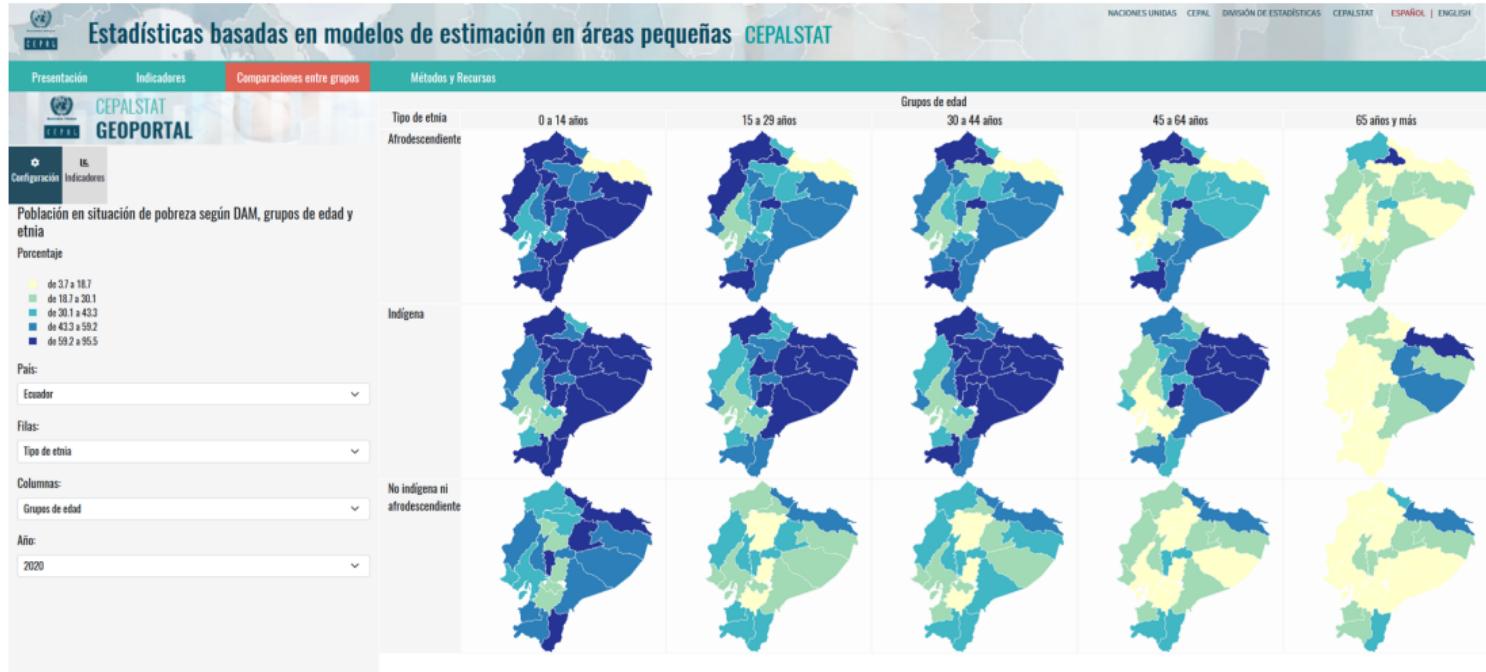


Figura 7: CEPALSTAT

## Desarrollos en 2023

1. En 2023, la CEPAL se unió a la FAO para apoyar al gobierno chileno en las estimaciones de FIES en municipios.
  - ▶ 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.

## Desarrollos en 2023

3. En 2023, la División de Estadística de las Naciones Unidas, la CEPAL y el UNFPA lanzaron la Capacitación Global sobre Métodos SAE donde los investigadores más influyentes del mundo grabaron sesiones de capacitación.
  - ▶ Dos versiones de la Capacitación Global (autodidacta y guiada).
  - ▶ Partha Lahiri, Isabel Molina, Pedro Silva, Mónica Pratesi, entre otros.
  - ▶ 10 módulos.

## And this is only the beginning

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.

And this is only the beginning

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 al pensamiento bayesiano.

# Modelos de población con el enfoque de **Tom**

Y te levantas un día...

- ▶ Y te sientes un poco raro, y débil. Vas al médico y te hacen exámenes. Uno de ellos te marca positivo para una enfermedad muy rara que solo afecta al 0.1% de la población.

**No son buenas noticias.**

- ▶ Vas al consultorio del médico y le preguntas qué tan específico es el examen. Te dice que es muy preciso; identifica correctamente al 99% de la gente que tiene la enfermedad.

# Y conoces a Thomas...

Esta es la información que tienes:

- $P(E) = 0.001$
- $P(+|E) = 0.99$
- $P(-E) = 0.999$
- $Pr(+|-E) = 0.01$

Además, por el teorema de probabilidad total

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

La regla de Bayes afirma lo siguiente:

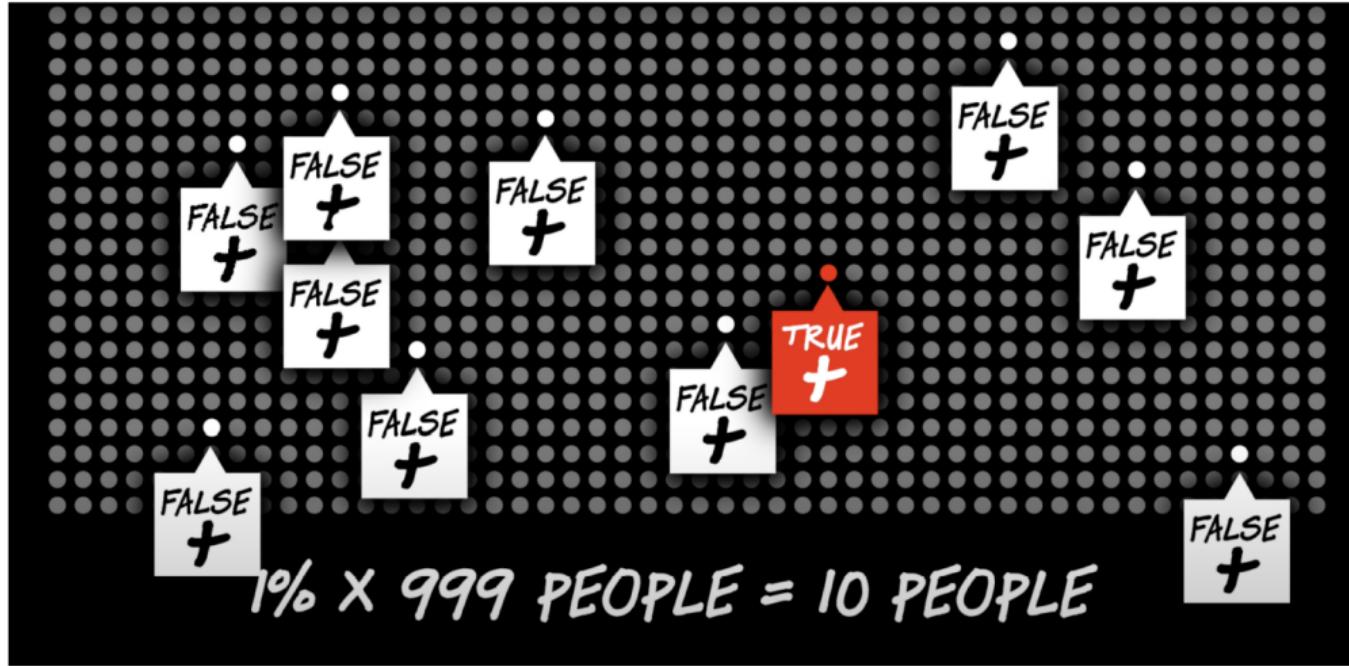
$$Pr(E|+) = \frac{Pr(+|E) \times Pr(E)}{Pr(+)}$$

Por lo tanto:

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



¿cómo funciona?



¿cómo funciona?



1 IN 11 PEOPLE = 9%

Y pides una segunda opinión

- ▶ Y esta vez el médico ordena que vuelves a realizarte ese mismo examen... y vuelves a marcar positivo para esa enfermedad.
- ▶ **Y vuelves a preguntarte:** *¿cuál es la probabilidad de que tenga esa enfermedad?*

Esta vez, has actualizado tu información sobre  $Pr(E)$ , pues ya marcaste positivo en un examen

$$Pr(E) = 0.09 \text{ Y } Pr(-E) = 0.91$$

Por lo tanto:

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

## Elementos de la regla de Bayes

En términos de inferencia para  $\theta$ , es necesario encontrar la distribución de los parámetros condicionada a la observación de los datos. Para este fin, es necesario definir la distribución conjunta de la variable de interés con el vector de parámetros.

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

- ▶ La distribución  $p(\theta)$  se le conoce con el nombre de distribución previa.
- ▶ El término  $p(Y | \theta)$  es la distribución de muestreo, verosimilitud o distribución de los datos.
- ▶ La distribución del vector de parámetros condicionada a los datos observados está dada por

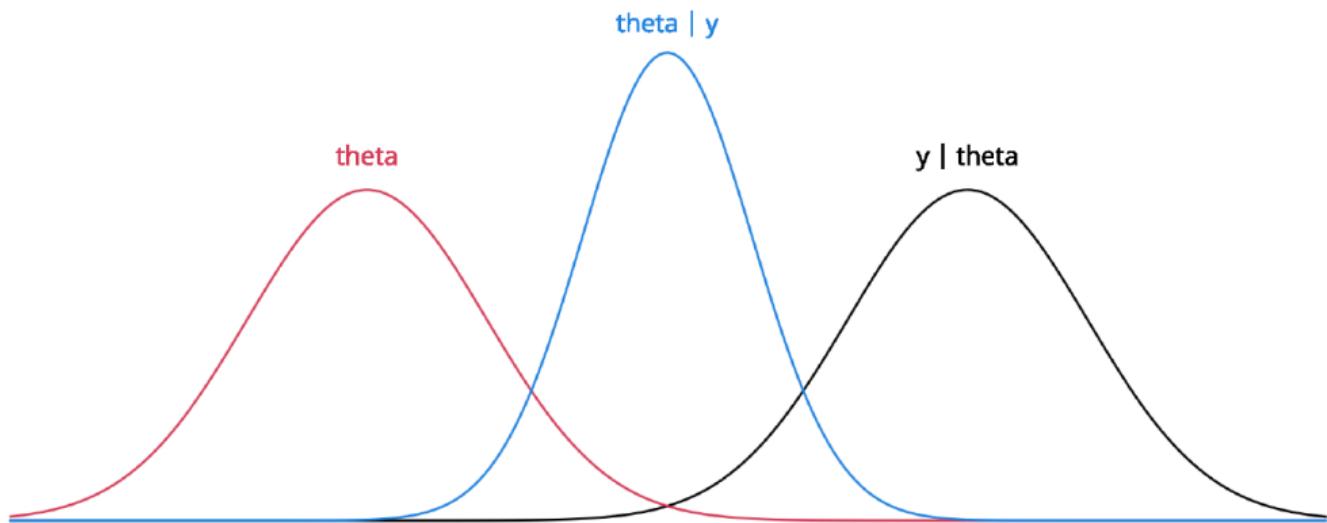
$$p(\theta | Y) = \frac{p(\theta, Y)}{p(Y)} = \frac{p(\theta)p(Y | \theta)}{p(Y)}$$

## Regla de Bayes

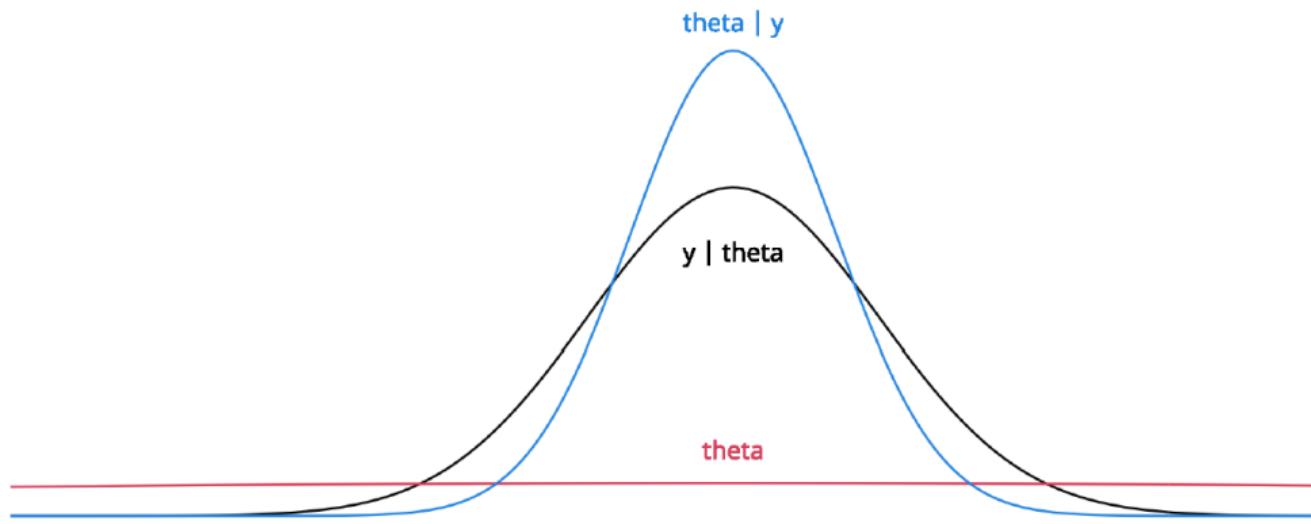
- ▶ El término  $p(\theta | Y)$  se le conoce con el nombre de distribución **posterior**.
- ▶ El denominador no depende del vector de parámetros y considerando a los datos observados como fijos, corresponde a una constante y puede ser obviada. Luego,

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

## Distribución previa informativa para $\theta$



## Distribución previa NO informativa para $\theta$



## Modelo de área Poisson

Suponga que  $Y = \{Y_1, \dots, Y_n\}$  es una muestra aleatoria de variables con distribución Poisson con parámetro  $\theta$ , la función de distribución conjunta o la función de verosimilitud está dada por

$$\begin{aligned} p(Y \mid \theta) &= \prod_{i=1}^n \frac{e^{-\theta} \theta^{y_i}}{y_i!} I_{\{0,1,\dots\}}(y_i) \\ &= \frac{e^{-n\theta} \theta^{\sum_{i=1}^n y_i}}{\prod_{i=1}^n y_i!} I_{\{0,1,\dots\}^n}(y_1, \dots, y_n) \end{aligned}$$

donde  $\{0, 1 \dots\}^n$  denota el producto cartesiano  $n$  veces sobre el conjunto  $\{0, 1 \dots\}$ .

El parámetro  $\theta$  está restringido al espacio  $\Theta = (0, \infty)$ .

## Distribución previa para $\theta$

- La distribución previa del parámetro  $\theta$  dada por

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

- La distribución posterior del parámetro  $\theta$  está dada por

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

Base de datos Censo de Costa Rica

Limitaciones identificadas en la base.



Figura 8: Mas de UGMs Rural

## Limitaciones identificadas en la base.



Figura 9: Mas de UGMs 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.

## Lectura de Datos de Viviendas sin Coordenadas.

- ▶ Importación de datos de viviendas desde un archivo CSV.

```
Viviendas_sin_coordenadas <-
  read_csv2("Recursos/02_Census_Filters/Data/Viviendas sin coordenadas.csv")
```

- ▶ Transformación de datos al formato requerido.
- ▶ Creación de variables como ID de provincia, cantón y distrito.

```
Viviendas_sin_coordenadas %<-%>
  transmute(
    LLAVEV,
    PROV_ID = str_sub(CODIGO_PCD, 1,1),
    CANT_ID = str_sub(CODIGO_PCD, 1,3),
    DIST_ID = as.character(CODIGO_PCD),
    UGM_ID = paste0(CODIGO_PCD , ID_UGM),
    H01A_TOTAL_PERSONAS = H01A_TOTAL_RESIDENTES_HAB)
```

## Lectura de Datos del Censo Estandarizado.

Leer los datos del censo estandarizado desde un archivo RDS (R Data Serialization) almacenado. De manera similar a los pasos anteriores, ajustamos los códigos del UGM para mantener la consistencia de los datos.

```
censo1 <-  
  readRDS(  
    "Recursos/02_Census_Filters/Data/censo_estandarizado.rds")
```

## Incorporación de la Base de Edad y Sexo.

En esta sección, incorporamos la base de edad y sexo en el análisis. La base de edad y sexo se lee desde un archivo RDS almacenado.

```
censo_sexo_edad <-
  readRDS(
    "Recursos/02_Census_Filters/Data/Censo con grupos por sexo.rds")
```

## Unión Interna para Agregar la Base de Edad y Sexo

- ▶ Cálculo de la diferencia en el recuento de filas entre las bases.
- ▶ Identificación de hogares censados en papel no incluidos.
- ▶ Comparación del número de filas con los datos del censo y viviendas sin coordenadas.

```
censo1 <- inner_join( censol, censo_sexo_edad,  
join_by( un_ID, PROV_ID, CANT_ID,  
DIST_ID, UGM_ID, LLAVEV, V01_TIPO_VIVIENDA,  
V02_OCUPACION_VIVIENDA  
)  
)
```

## 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.

# Análisis de la distribución de greenpoints

1. Se calcula la distribución del estado de greenpoint entre los hogares.
2. Agrupa los datos según el estado de 'greenpoint' y cuenta el número de hogares en cada categoría.
3. Calcula la distribución porcentual para cada categoría.

greenpoint	n	percentage
0	471456	27.04
1	1053477	60.43
NA	218308	12.52

## 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"  
  )  
)
```

## Aplicación del segundo filtro: Criterio WorldPop

Incluimos todos los hogares con la variable WorldPop (WP) que se encuentran dentro de 1 desviación estándar de su valor promedio. Sin embargo, si estos hogares tienen cero residentes en la variable de interés, marcamos esa variable como “No Disponible” (NA).

Las estadísticas resumen para la variable ‘wpop\_sum’ se muestran en la siguiente tabla.

media	sd	min	max
96.97	143.2	0	6214

Estas estadísticas nos ayudan a establecer los umbrales para el filtro y se guardan en un archivo de resumen.

## Cálculo de umbrales inferiores y superiores

Utilizamos las estadísticas resumen para calcular los umbrales inferiores y superiores.

```
li <- 96.96515 - 143.1986 * 1 # Umbral inferior  
ls <- 96.96515 + 143.1986 * 1 # Umbral superior
```

Identificamos y contamos los hogares que cumplen con los criterios del segundo filtro.  
Nos enfocamos en los hogares con cero residentes ('H01A\_TOTAL\_PERSONAS')  
pero que tienen valores de 'wpop\_sum' fuera de los umbrales calculados.

## Aplicando los umbrales inferiores y superiores

Para validar el resultado de aplicar el criterio de WorldPop se realiza la siguiente tabla.

```
filter_2_counts <- censo2 %>%
  filter(H01A_TOTAL_PERSONAS == 0, wpop_sum > ls | wpop_sum < li) %>%
  group_by(V02_OCUPACION_VIVIENDA) %>% summarise(n = n())
filter_2_counts
```

V02_OCUPACION_VIVIENDA	n
2	129652
3	22968
4	8210
5	10514
6	4635
7	532
8	17160

## Aplicación del segundo filtro y actualización de columnas

Finalmente, aplicamos el segundo filtro a los hogares y actualizamos las columnas 'greenpoint2' y 'Filtros'.

```
censo3 <- censo2 %>% mutate(  
  greenpoint2 = case_when(  
    H01A_TOTAL_PERSONAS == 0 &  
    (wpop_sum > ls | wpop_sum < li) ~ "Sin información pero n>=0",  
    TRUE ~ greenpoint2  
)  
,  
  Filtros = case_when(  
    H01A_TOTAL_PERSONAS == 0 &  
    (wpop_sum > ls | wpop_sum < li) ~ "Criterion WorldPop",  
    TRUE ~ NA_character_  
)  
)
```

## 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

## Resumen de estadísticas basadas en 'greenpoint2'

- ▶ Calculamos estadísticas adicionales para las categorías de 'greenpoint2'.
- ▶ Estos datos son esenciales para comprender la distribución de residentes en los hogares filtrados.

greenpoint2	min	max	num_na	total
Censado con informacion n=0	0	0	0	212980
Censado con informacion n>0	1	261	0	776478
Sin informacion pero n>0	0	0	0	341804
Sin informacion pero n>=0	NA	NA	218308	411979

## Definición del Tercer Filtro

- ▶ El tercer filtro aborda los hogares dentro de las UGM que fueron encuestados después de un intervalo mayor a 20 días y, a pesar de estar clasificados como desocupados, existe incertidumbre sobre su estado de ocupación.
- ▶ Para la implementar el filtro se contaba con el archivo 'Desocupadas fuera periodo.xlsx' que recopila información sobre los hogares que estaban desocupados pero fueron visitados fuera del intervalo estándar.
- ▶ Utilizando la información recopilada actualizamos las columnas 'greenpoint2' y 'Filtros' según los criterios especificados.

## Implementando el tercer filtro

```
censo4 <- censo3 %>% mutate(  
  greenpoint2 = case_when(  
    UGM_ID %in% upms_reporte$UGM_ID &  
    H01A_TOTAL_PERSONAS == 0 ~ "Sin informacion pero n>=0",  
    TRUE ~ greenpoint2  
,  
  Filtros = case_when(  
    UGM_ID %in% upms_reporte$UGM_ID &  
    H01A_TOTAL_PERSONAS == 0 ~ "Fuera de periodo (20 días)",  
    TRUE ~ Filtros  
)  
)
```

## Aplicación de filtros adicionales y creación de valores en ‘Filtros’

Refinando aún más los datos mediante la aplicación de filtros adicionales. Los valores en ‘Filtros’ se actualizan en función de diversas condiciones, como el número de residentes, el resultado de la entrevista ('RESUL\_ENTREVISTA\_VIV') y la ocupación de la vivienda ('V02\_OCUPACION\_VIVIENDA').

```
censo4 %<~% mutate(Filtros = case_when(  
  is.na(Filtros) & H01A_TOTAL_PERSONAS > 0 ~ "Número de personas mayor a 0",  
  
  is.na(Filtros) & RESUL_ENTREVISTA_VIV %in% c(1) &  
  H01A_TOTAL_PERSONAS == 0 ~ "Entrevista igual a 1 y número de personas igual a 0"  
  
  is.na(Filtros) & RESUL_ENTREVISTA_VIV %in% c(3,4) ~ "Entrevista es 3 o 4",  
  
  is.na(Filtros) & is.na(greenpoint) & is.na(personas) ~ "Sin conteo de personas",  
  
  is.na(Filtros) & V02_OCUPACION_VIVIENDA == "8" ~ "Ocupación de la vivienda es 8",  
  
  TRUE ~ Filtros  
)
```

## Resumen de datos basados en la variable 'greenpoint2'

Este resumen nos ayuda a comprender el impacto del tercer 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

## Resumen de datos basados en 'greenpoint2' y 'Filtros'

Generamos un resumen adicional que considera la combinación de las variables 'greenpoint2' y 'Filtros'. Esto proporciona información más detallada sobre cómo el Criterio WorldPop afecta a las categorías existentes.

greenpoint2	Filtros	n	percentage
Censado con informacion n=0	Entrevista igual a 1 y Número de personas igual a 0	175921	10.092
Censado con informacion n>0	Número de personas mayor a 0	776478	44.542
Sin informacion pero n>0	Entrevista es 3 o 4	285810	16.395
Sin informacion pero n>=0	Criterio WorldPop	135370	7.765
Sin informacion pero n>=0	Fuera de periodo(20 días)	151354	8.682
Sin informacion pero n>=0	Sin conteo de personas	218308	12.523

**En un proceso adicional incorporamos las entrevistas realizadas en papel y se hace la validación de identificadores duplicados**

# Tabla de resumen finales

greenpoint2	Filtros	min	max	num_na	total
Censado con informacion n=0	Entrevista igual a 1 y Número de personas igual a 0	0	0	0	175921
Censado con informacion n>0	Número de personas mayor a 0	1	261	0	776478
Sin informacion pero n>0	Entrevista es 3 o 4	0	0	0	285810
Sin informacion pero n>=0	Criterio WorldPop	0	0	0	135370
Sin informacion pero n>=0	Fuera de periodo(20 días)	0	0	0	151354
Sin informacion pero n>=0	Sin conteo de personas	NA	NA	218308	218308

## Proceso de estimación en STAN

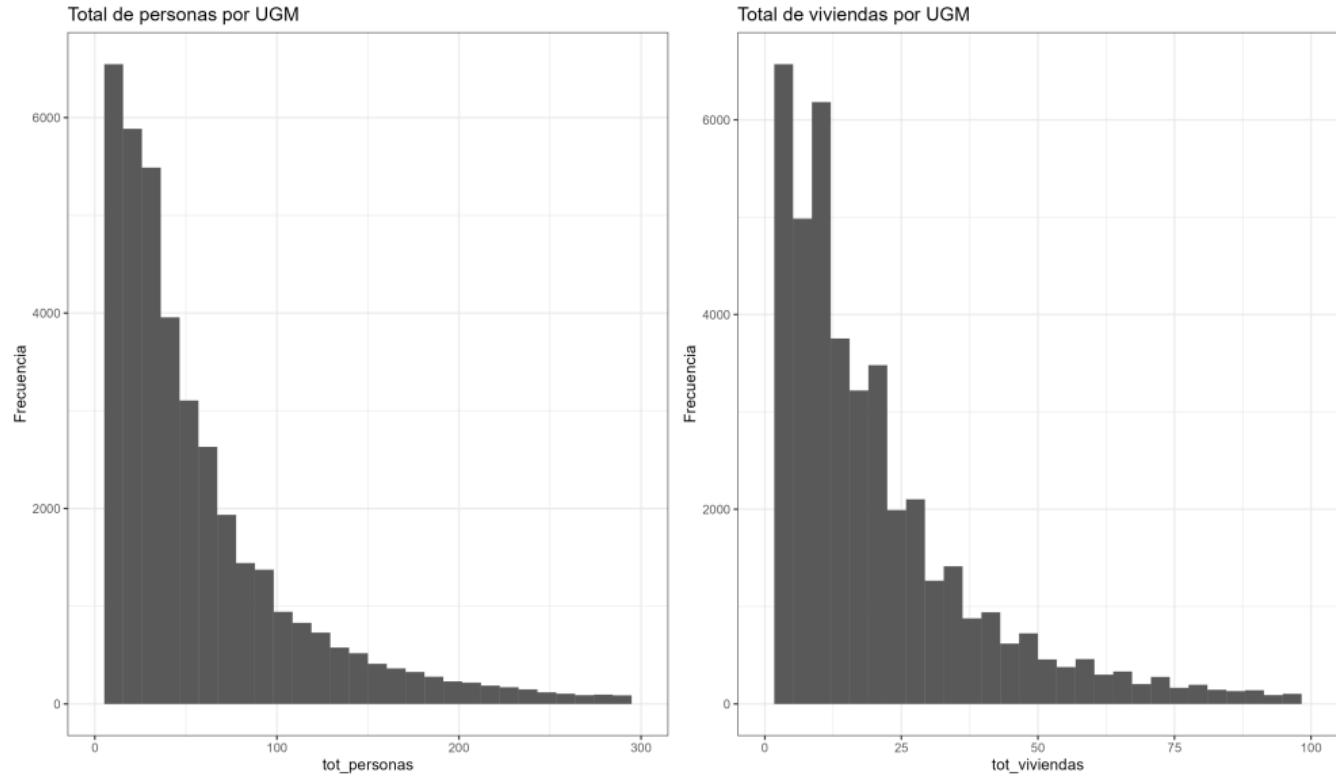
Sea  $Y$  el conteo de personas censadas por UGM del país. Aquí incluimos las viviendas con cero personas.

```
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 conteno de personas



## Modelo escrito en código STAN

```
data {
    int<lower=0> n;          // Número de áreas geograficas
    int<lower=0> y[n];      // Conteos por 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 de longitud n
    for(ii in 1:n){
        ypred[ii] = poisson_rng(theta);
    }
}
```

# Preparando datos para código STAN

## ► Organizando datos para STAN

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

## ► Ejecutando el código de STAN

```
stan_pois <- "Recursos/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,  
        "Recursos/00_Intro_bayes/Poisson/model_poisson.rds")
```

## Resultados de la estimación del parámetro $\theta$

```
model_poisson <- readRDS(  
    "Recursos/00_Intro_bayes/Poisson/model_poisson.rds")  
tabla_posi <- summary(model_poisson,  
    pars = c("theta"))$summary  
tabla_posi %>% tba()  
saveRDS(tabla_posi, "Recursos/00_Intro_bayes/Poisson/04_tabla_theta.rds")
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
theta	56.04	0.0014	0.0355	55.97	56.02	56.04	56.06	56.11	678.9	1.008

## Convergencias de las cadenas el parámetro $\theta$

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

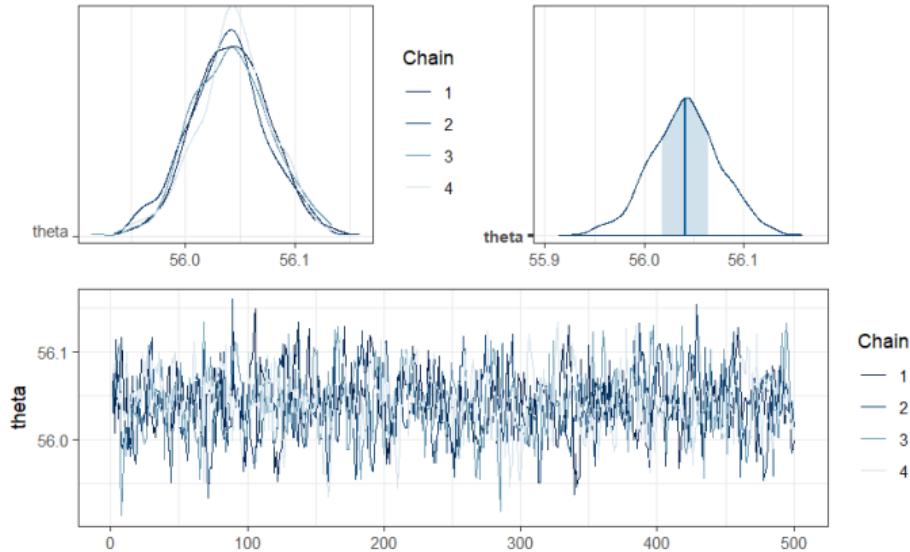


Figura 10: Cadenas evaluación de las convergencias de  $\theta$

## Chequeo predictivo posterior

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

rowsrandom<-sample(nrow(y_pred_B),300)

y_pred2<-y_pred_B[rowsrandom,]

p1<- ppc_dens_overlay(y =as.numeric(dataPois$tot_personas), y_pred2)
p1 + xlim(0,300)
```

## Chequeo predictivo posterior

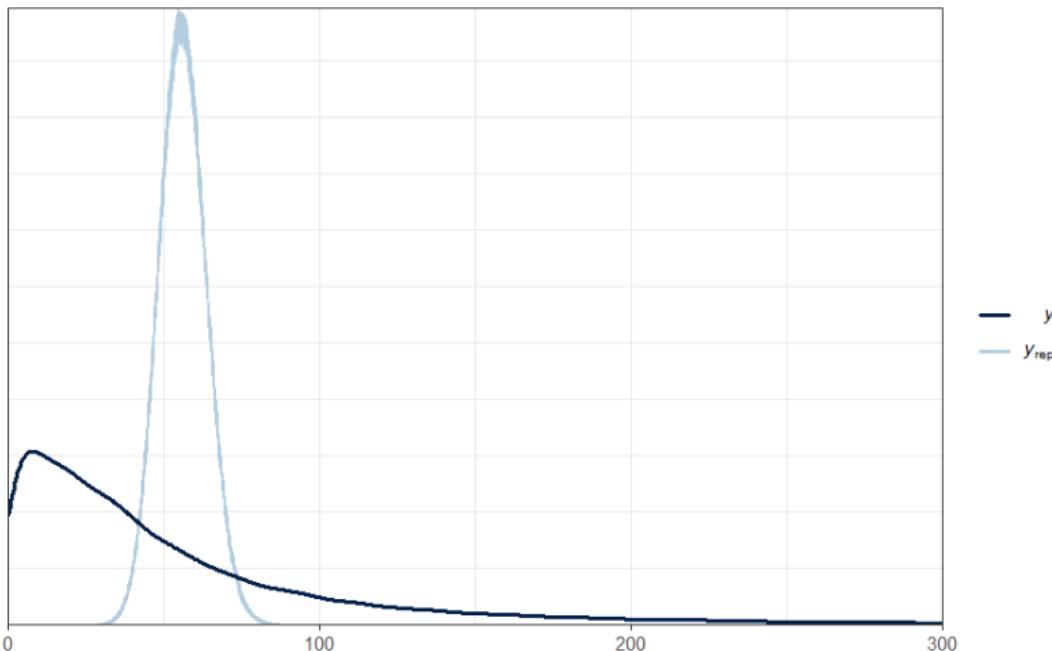


Figura 11: Chequeo predictivo posterior para el total de personas por UGM

## Proceso de estimación en **STAN** (total de viviendas)

- Sea  $Y$  el conteo de viviendas ocupadas con personas presentes censadas por UGM.

**Empleando un proceso igual que el caso anterior se realiza la estimación del modelo para la variable  $Y$**

- Organizando datos para STAN

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

## Ejecutando el código de STAN

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

# Resultados de la estimación del parámetro $\theta$

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
theta	21.07	8e-04	0.0217	21.03	21.05	21.07	21.08	21.11	683.1	1.001

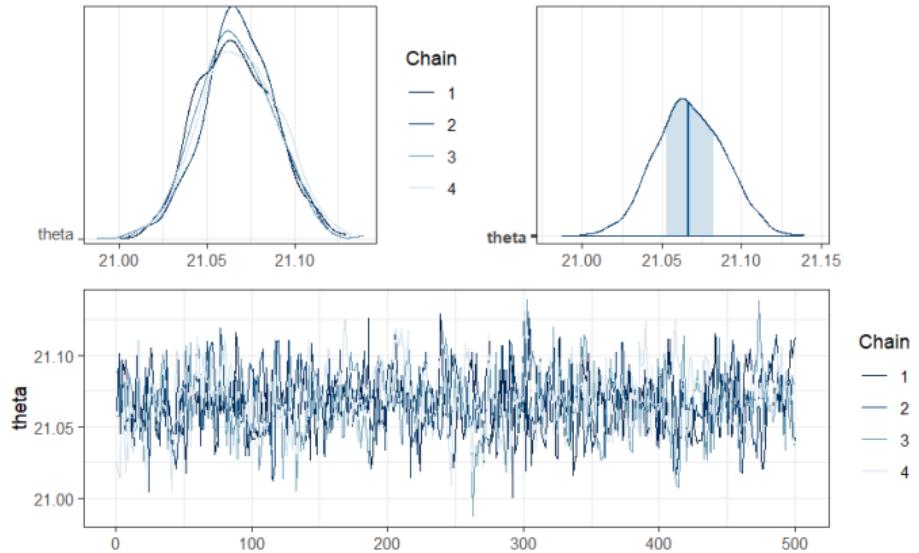


Figura 12: Evaluación de la convergencias de  $\theta$

## Chequeo predictivo posterior

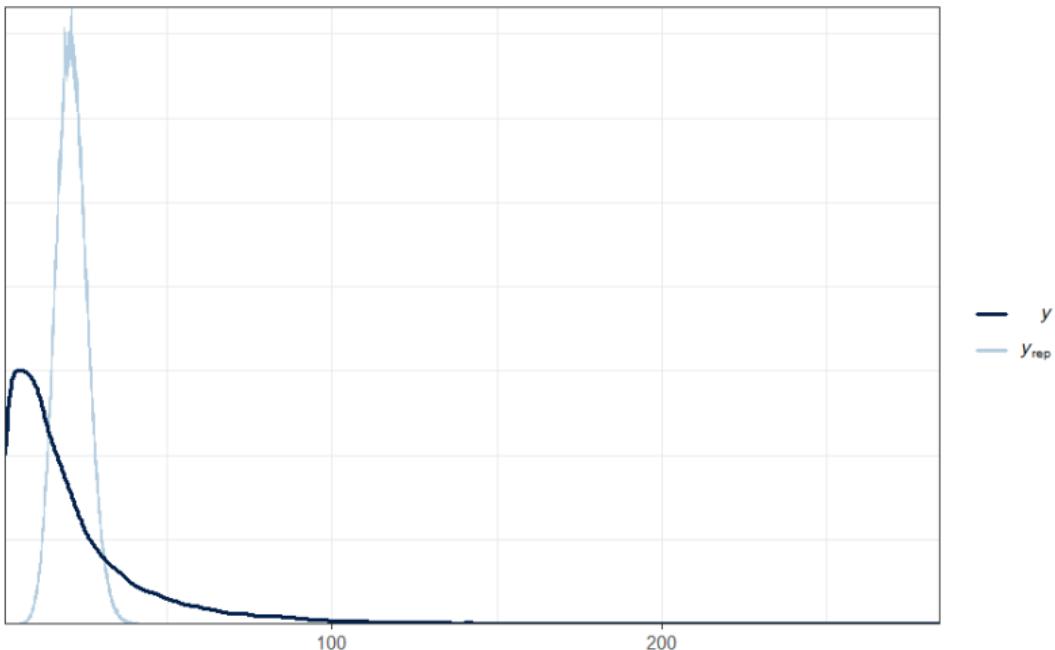


Figura 13: Chequeo predictivo posterior para el total de viviendas por UGM

## Estandarización y validación de covariables

## 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 UGM, Cantos, 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 UGM

Tabla 2: Valores descriptivos de la base de UGM (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 UGM

Tabla 3: Valores descriptivos de la base de UGM (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 Costa Rica

- ▶ 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 las unidades geoestadísticas mínimas (UGM), 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 UGM sigue una distribución de Poisson.

## Caso de estudio Costa Rica

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 UGM,  $N_{ij}$  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 Costa Rica, 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 UGMs  
    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  $n>0$

## Listado de variables usadas en el modelo

- ▶ 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
- ▶ GHS\_BUILT\_S\_E2020\_GLOBE\_R2023A\_5367\_CRI (superficie urbanizada en m<sup>2</sup>)
- ▶ 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

# 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 UGM 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

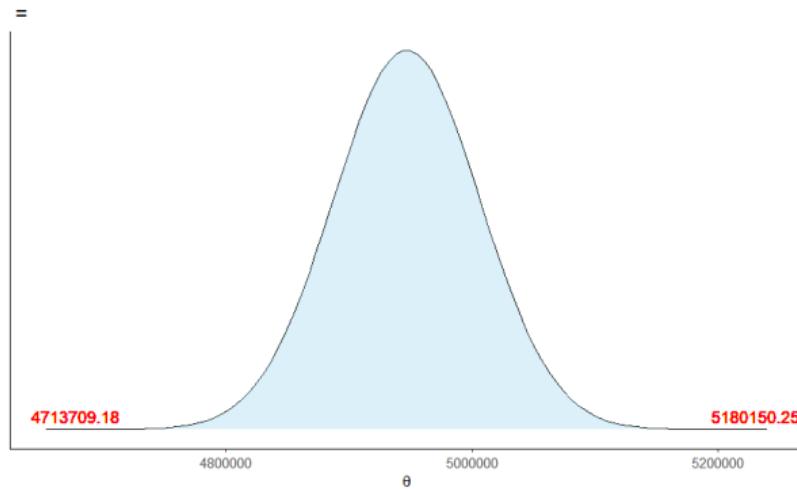
- ▶ Ahora se presenta la metodología utilizada para obtener estimaciones en varios niveles de agregación, empleando un conjunto de funciones personalizadas diseñadas para este conjunto de datos en particular.
- ▶ Estas funciones facilitan el proceso de generar predicciones y visualizaciones, lo que nos permite analizar de manera efectiva las estimaciones de población.

```
source("Recursos/03_Model_for_people/01_Funcion_agregados.R")
censo_vivienda <-
  readRDS("Recursos/03_Model_for_people/05_censo_vivienda_personas_grupo_e")
```

- ▶ **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

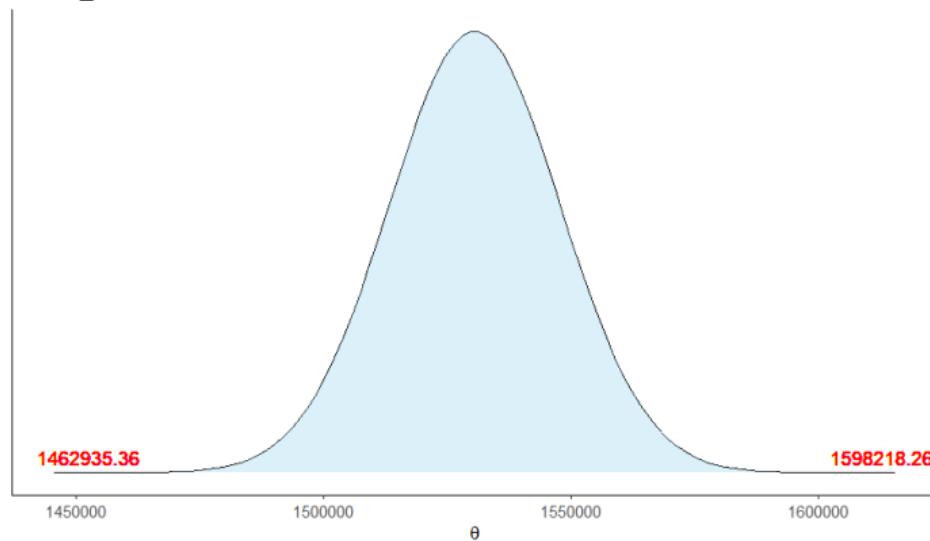
```
tab <- Pred_totPob(censo_vivienda, agrega = "PROV_ID", Plot = FALSE)
```

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)
```

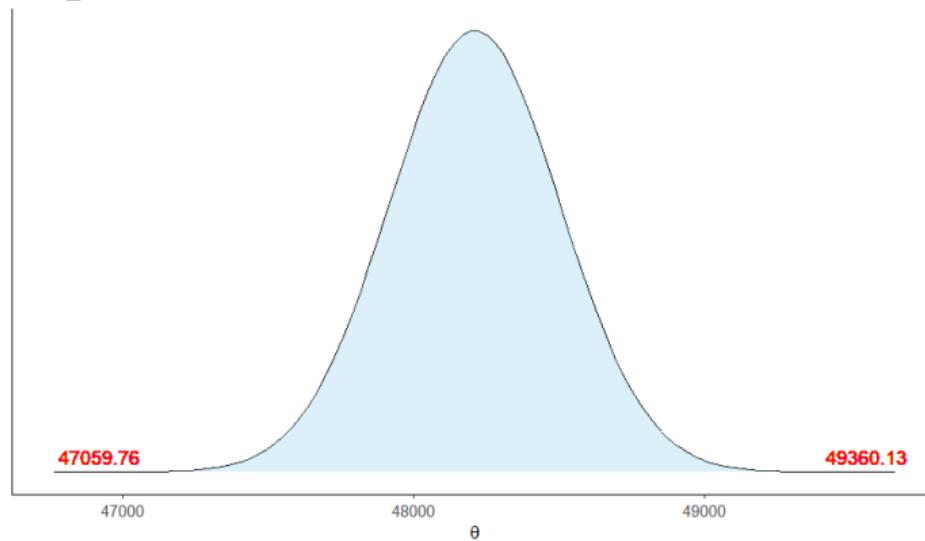
PROV\_ID = 1



## Predicción del total por distrito

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

DIST\_ID = 10110



## 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")
```

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

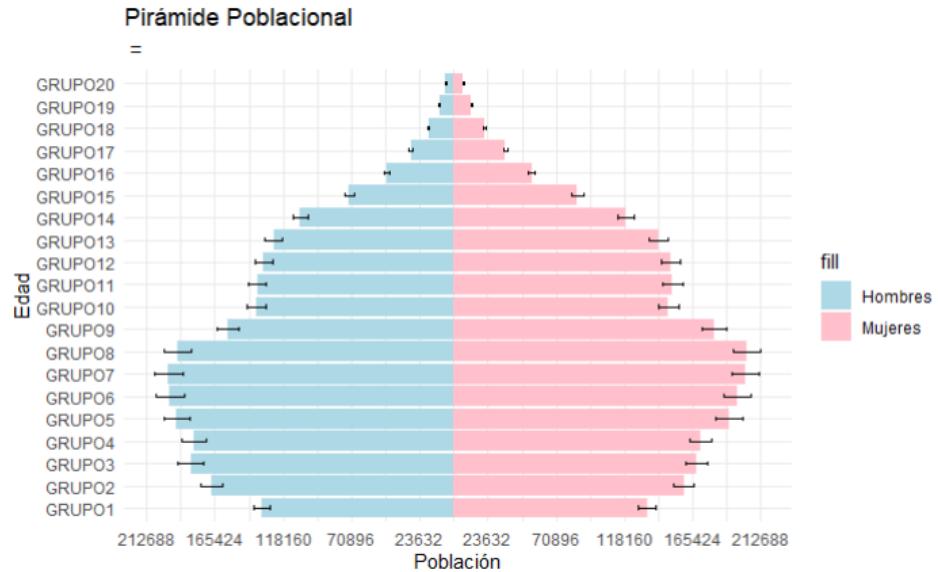


Figura 14: Resultado nacional

## 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

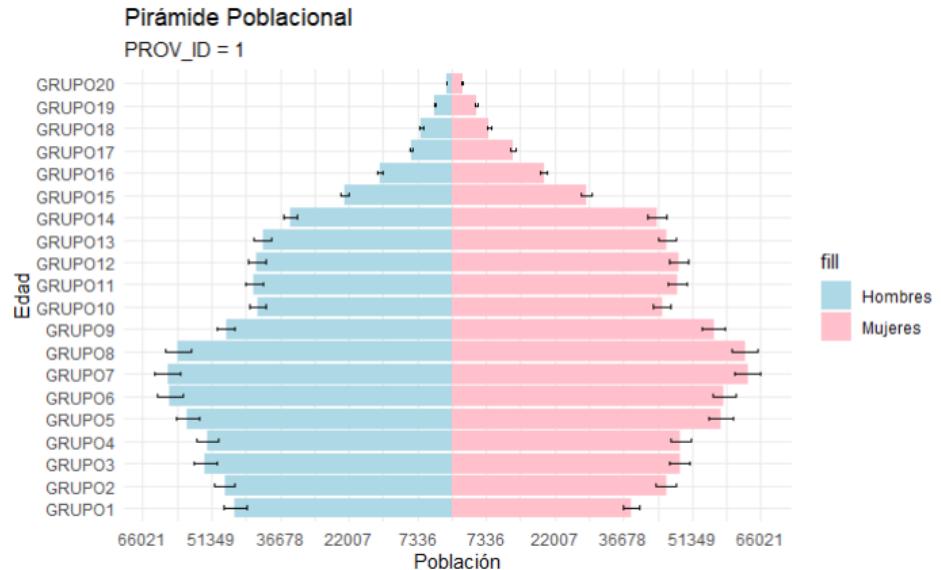


Figura 15: Resultado para la provincia 1

## 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

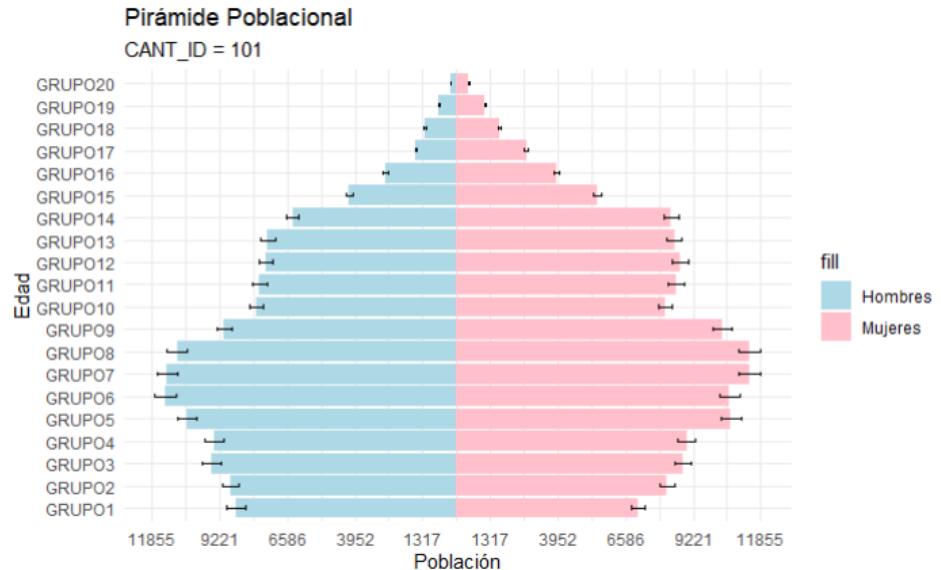


Figura 16: Resultado para la canton 101

¡Gracias!

*Email:* andres.gutierrez@cepal.org