Desagregación de Estimaciones en Áreas Pequeñas un enfoque bayesiano

CEPAL - Unidad de Estadísticas Sociales

Tabla de contenidos I

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:

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

Además, por el teorema de probabilidad total

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

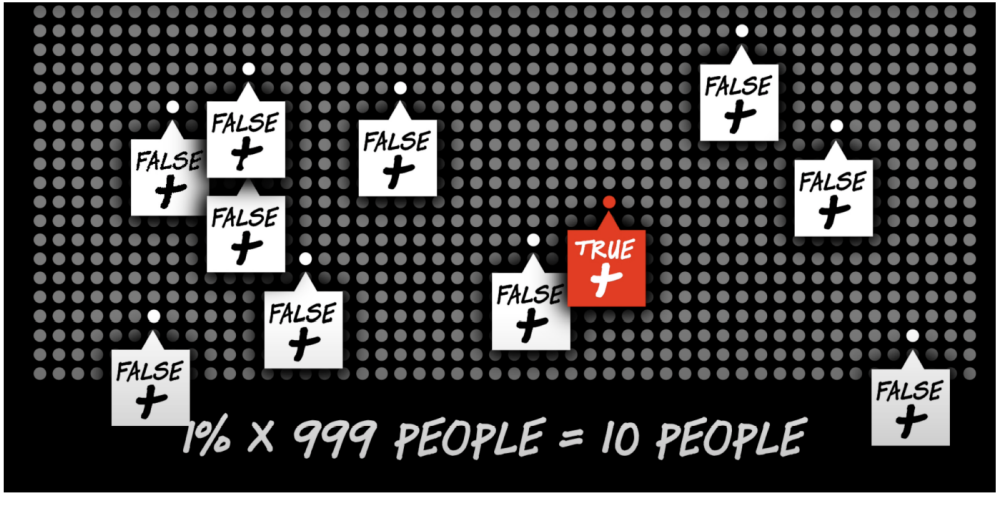
La regla de Bayes afirma lo siguiente:

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

Por lo tanto:

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

¿cómo funciona?



¿cómo funciona?

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

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

Por lo tanto:

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

Elementos de la regla de Bayes

En términos de inferencia para θ, 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.

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

- La distribución $p(\theta)$ se le conoce con el nombre de distribución previa.
- El término $p(Y \mid \theta)$ 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

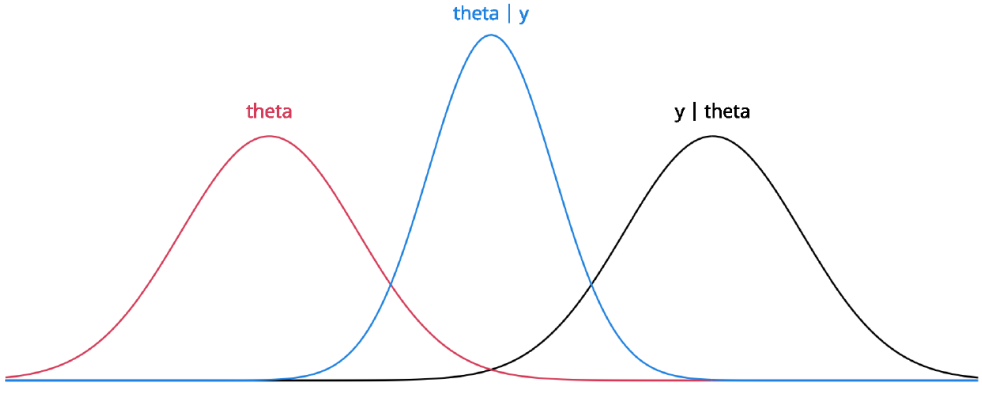
$$p(\theta \mid Y) = \frac{p(\theta, Y)}{p(Y)} = \frac{p(\theta)p(Y \mid \theta)}{p(Y)}$$

Regla de Bayes

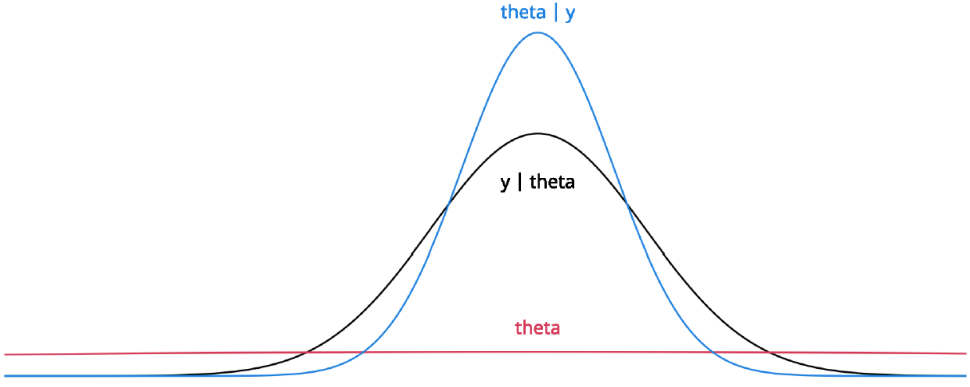
- El término $p(\theta \mid Y)$ se le conoce con el nombre de distribución ***posterior***.
- El denominador no depende del vector de parámetros y considerando a los datos observados como fijos, corresponde a una constante y puede ser obviada. Luego,

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

Distribución previa informativa para θ



Distribución previa NO informativa para θ



Modelo de área Poisson

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

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

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

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

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

Distribución previa para θ

- La distribución previa del parámetro θ dada por

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

- La distribución posterior del parámetro θ está dada por

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

Base de datos Censo de Costa Rica

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 un 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( censo1, 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.

Codígo 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)  ~ "Criterio 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**

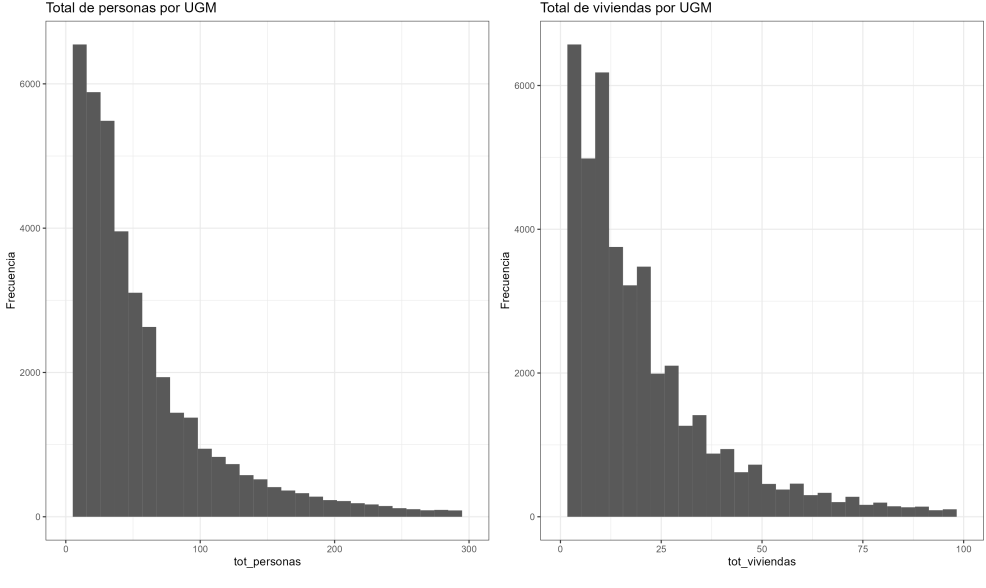
Sea Y el conteo de personas censadas por UGM del país. Aquí incluimos las viviendas con cero personas.

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

Tabla 1: Conteno 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

Histrograma 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 θ

```
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 θ

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

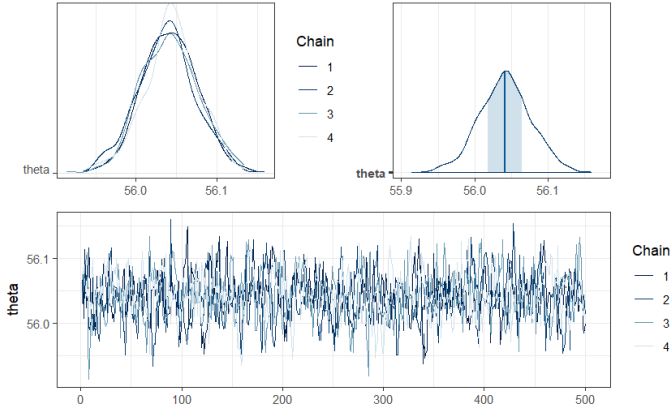


Figura 1: Cadenas evaluación de las convergencias de θ

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

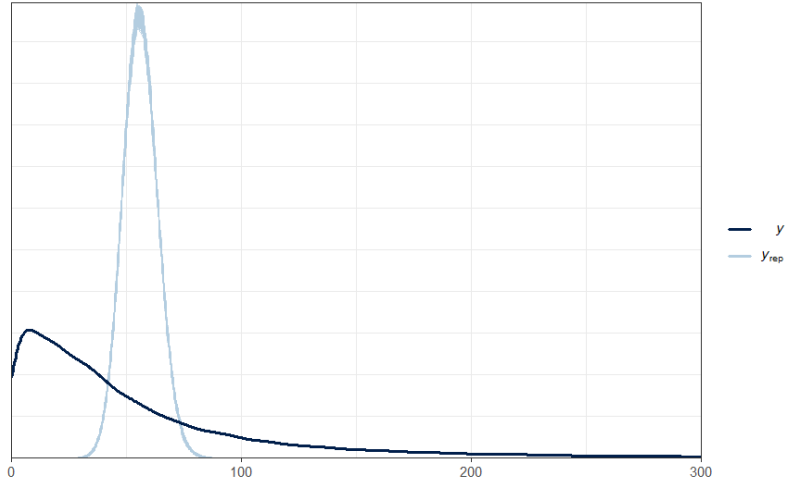


Figura 2: Chequeo predictivo posterior para el total de personas por UGM

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 θ

	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

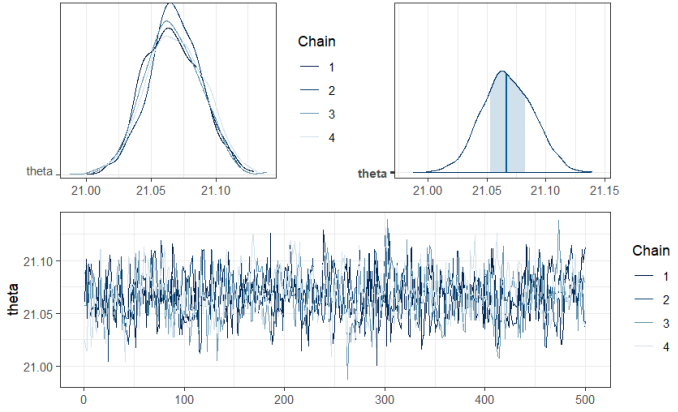


Figura 3: Evaluación de la convergencias de θ

Chequeo predictivo posterior

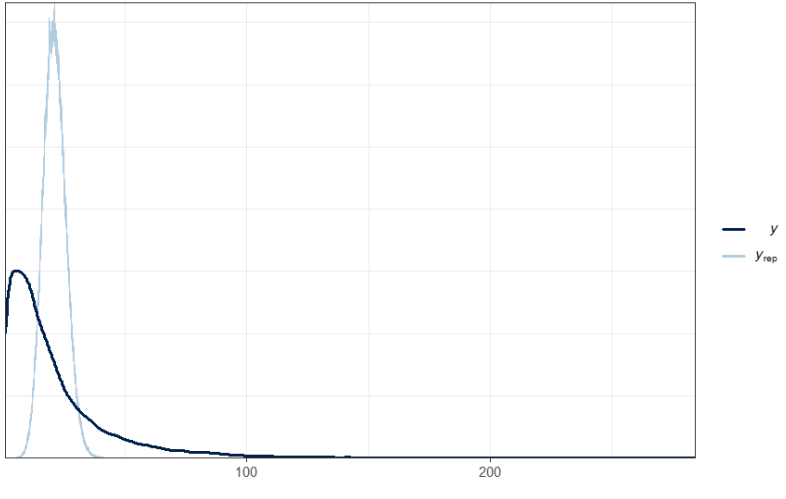


Figura 4: Chequeo predictivo posterior para el total de viviendas por UGM

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_matricula_educacion_primaria_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:

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

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 $z_{ij}^t\gamma$

Obejtivo.

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

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

,

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

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_matricula_educacion_primaria_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")
```

Seleccionado 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_matricula_educacion_primaria_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 $\hat{y}_d = E_{\mathcal{M}}\left(Y_d \mid X_d, \beta\right)$ para cada UGM para cada modelo.
- La predicción por UGM se hace siguiendo las siguientes reglas

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

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

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

Estimaciones agregadas

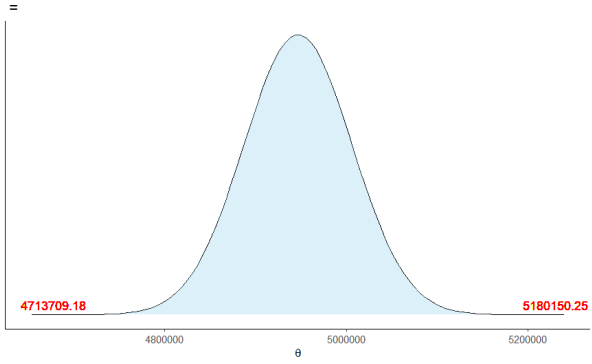
- Ahora se presenta la metodología utilizada para obtener estimaciones en varios niveles de agregación, empleando un conjunto de funciones personalizadas diseñadas para este conjunto de datos en particular.
- Estas funciones facilitan el proceso de generar predicciones y visualizaciones, lo que nos permite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

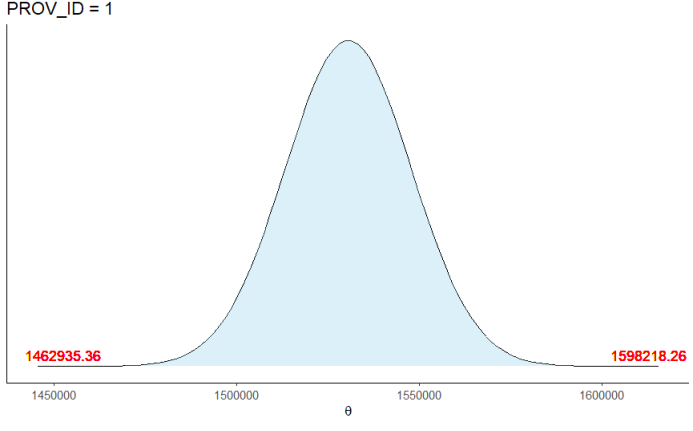
Predicción del total por provincia

```
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)
```



Predicción del total por distrito

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