UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

TÉCNICAS DE ANÁLISIS DE CONGLOMERADOS PARA LA DEFINICIÓN DE ESTRATOS DE MUESTREO USANDO INFORMACIÓN ESPACIAL Y SOCIODEMOGRÁFICA DE MÉXICO

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN MATEMÁTICAS APLICADAS

PRESENTA:

JAZMÍN ALEJANDRA MARTÍNEZ GUERRERO

DIRECTOR DE LA TESIS:

DR. GONZALO PÉREZ DE LA CRUZ

Ciudad Universitaria, CD. MX 2024

"Para que serve o arrependimento, se isso não muda nada do que se passou? O melhor arrependimento é, simplesmente, mudar."

Agradecimientos

Quiero agradecer a mi papá, por su apoyo siempre incondicional. A mi mamá por la presión necesaria y un montón de cariño. Y a mi hermano, por ser una motivación en mi día a día para nunca rendirme.

A Nymeria por desvelarse conmigo todas las noches que le dedique a esta tesis.

Por siempre confiar en todo lo que soñé, cuidarme y guiarme hasta aquí; a Héctor. Sin él no hubiera tenido el valor para cambiarme de carrera. Gracias por priorizar mi felicidad, por hablar por mí cuando me daba miedo hablar y por soñar por mí cuando yo no quería soñar.

A mis amigos, que son uno de los pilares más importantes de mi vida y que sin ellos no podría haber sido la persona que soy ahora:

- A José y Rodrigo por haber sido mis mejores amigos durante toda la carrera, apoyarme, escucharme y enseñarme, sin ustedes la universidad no podría haber sido lo que fue.
- A Paolo por tantos momentos de diversión, aprendizaje y por siempre creer en mí.
- A Alberto porque pensé que había sido una suertuda por haberme quedado en esa clase de álgebra, pero la verdadera suerte fue encontrarte.
- A Pablo por la innumerables platicas aunque eso nos haya costado recursar álgebra lineal, por quererme a pesar de ser una mandona y sentirme la jefa del equipo de cálculo, por tantas aventuras juntos.
- A Bernardo por construir conmigo la amistad más leal que haya conocido.
- A Yadira por su amistad, apoyo incondicional y ser uno de mis ejemplos a seguir, te admiro tanto.
- A Karla por haberme apoyado y escuchado durante tanto tiempo.
- A Gema, Brenda, Leo, Ana, Mariana, Kenneth, André y a todas las personas que estuvieron, aunque ya no estén.

También quiero agradecer al Doctor Gónzalo, por su infinita paciencia a lo largo de este trayecto y siempre responder mis dudas, de verdad no pude tener un mejor asesor.

A mis profesores: Sergio López, Lizbeth Naranjo, Javier Fernandez, Diana Avella, Sebastián Velázquez y Yadira Rivas, por motivarme con su excepcional manera de enseñar.

Finalmente, quiero agradecer a la Universidad Nacional Autónoma de México, en especial a la Facultad de Ciencias, por darme la mayor cantidad de aprendizajes que he tenido en toda mi vida.

Índice general

Resumen

El objetivo de la tesis fue crear una estratificación de los municipios de los Estados Unidos Mexicanos que tomara en cuenta tanto características sociodemográficas como geográficas. Esto con la motivación de crear estratos de muestreo a nivel nacional como los que el INEGI obtiene en la primera etapa de su estratificación para el diseño de muestreo de su Muestra Maestra, es decir, cuando se crean los cuatro estratos sociodemográficos: bajo, medio bajo, medio alto y alto.

Se trabajó con la información de los municipios existentes hasta el 2020 según el Censo de Población y Vivienda realizado por el INEGI ese mismo año y para clasificarlos se utilizó el algoritmo de conglomerados jerárquico con restricciones espaciales utilizando el método de enlace Ward propuesto por Chavent et al. (2018).

Para utilizar el algoritmo son necesarias dos matrices: una de distancias geográficas y otra de distancias entre las características sociodemográficas, así como el peso de las observaciones w_i, un parámetro de mezcla α y el número de grupos k. Para las características sociodemográficas se utilizaron dos bases de datos: los Principales Resultados por Localidad del Censo de Población y Vivienda; diseñados por el INEGI, y los Indicadores de pobreza municipal 2015; creados por el CONEVAL. La información espacial de los municipios se obtuvo del Marco Geoestadístico diseñado por el INEGI.

Por practicidad se decidió darle el mismo peso w_i a todas las observaciones, sin embargo, respecto a la elección de los parámetros α y k se probó con diversos valores, con lo que se obtuvieron distintas formas de estratificar. Para elegir la mejor, se seleccionaron diez variables y se obtuvieron los estimadores de sus totales, utilizando el estimador Horvitz-Thompson en muestreo estratificado con un m.a.s al interior de cada estrato, se calculó la varianza poblacional para cada variable, y se conservó la estratificación que tuvo, en promedio, la menor varianza poblacional.

Para evaluar la estratificación resultante, se comparó la varianza poblacional obtenida de realizar un muestreo aleatorio simple al interior de cada estrato, con la varianza poblacional que se obtendría al llevar a cabo un m.a.s. El experimento se consideró satisfactorio puesto que la varianza obtenida con la estratificación fue menor. Aunado a esto, para analizar el desempeño de la estratificación se comparó su varianza poblacional con la de otras estratificaciones de interés como la estratificación por entidad federativa, por tipo de municipio y el cruce entre éstas. Todo el proceso se llevo a cabo utilizando el lenguaje de programación R.

Capítulo 1

Introducción

1.1. Motivación

En México, el Instituto Nacional de Estadística y Geografía (INEGI) es el encargado de llevar a cabo el Censo de Población y Vivienda cada diez años. Además, a partir de muestreos probabilísticos el INEGI genera información de diferentes temas cada cierto tiempo; por ejemplo, la Encuesta Nacional de Ocupación y Empleo (ENOE) se realiza trimestralmente, mientras que la Encuesta Nacional de Ingresos y Gastos de los Hogares (ENIGH) se lleva a cabo cada dos años. Aunque sería ideal realizar un censo cada vez que se quiere generar algún tipo de información y con ello obtener datos precisos y a un nivel muy desagregado, entrevistar a toda la población con estas periodicidades sería demasiado complicado y costoso. Por lo cual, basándose en la información del censo en su nivel más desagregado, el INEGI genera su Muestra Maestra, la cual está diseñada para ser representativa, es decir, para reflejar la información de toda la población. Así, cada vez que se quiere generar información el INEGI selecciona submuestras a través de la Muestra Maestra y con esto se evita tener que entrevistar a toda la población en cada ocasión, lo que resulta en una reducción de costos y una mayor eficiencia.

Actualmente, el INEGI está en proceso de actualización de su Muestra Maestra derivada del Censo 2020 (Rodriguez y Heredia, 2023), por lo que se sigue usando la de 2010. Según Landeros (2013) para el diseño de la Muestra Maestra del año 2010 el INEGI realizó los siguientes pasos:

1. **Creación de las Unidades Primarias de Muestreo**. El objetivo de crear Unidades Primarias de Muestreo (UPM) fue obtener agrupaciones de manzanas o localidades homogéneas en números de viviendas que facilitaran los recorridos de campo para realizar las encuestas y con ello la actualización de la muestra. Para agruparlas se utilizaron herramientas como Diagramas de Voronoi, Triangulación de Delaunay, Polígonos de Thiessen y Método de Recocido Simulado. En total se formaron 245,279 UPM.

2. **Estratificación de las UPM**.
   La estratificación se hizo de acuerdo a:

   - Las características sociodemográficas de los habitantes de las viviendas.
   - Las características físicas de las viviendas.

- La ubicación geográfica de las viviendas.

El proceso de estratificación se compuso de dos etapas: la primera consistió en clasificar a las UPM según su perfil sociodemográfico, empleando indicadores extraídos del censo. La segunda se realizó de acuerdo con su ubicación geográfica, la cual se deriva naturalmente de la división política del territorio nacional y del tamaño de las localidades.

Para la primera etapa de la estratificación se utilizó el método de conglomerados *K-means*, empleando 14 indicadores de población y 19 de vivienda, los cuales fueron construidos con la información del Censo de Población y Vivienda 2010. Como resultado se obtuvieron cuatro estratos a nivel nacional: bajo, medio bajo, medio alto y alto.

En la segunda fase de la estratificación, se llevó a cabo una clasificación adicional de los estratos formados en la primera etapa. En esta segunda clasificación, cada estrato fue diferenciado según la entidad federativa a la que pertenecía, y a su vez, según el tamaño de la localidad a la que estaba asociado. Este proceso resultó en la formación de un total de 746 estratos.

3. **Tamaño de la muestra**.
   Una vez estratificadas las UMP se calculó el tamaño de muestra necesario para obtener resultados con un nivel de precisión y confiabilidad específicos, considerando la variabilidad de la población y el diseño de muestreo.

Así, la Muestra Maestra del INEGI se compone de un conjunto de UPM seleccionadas aleatoriamente y cuando se quiere generar o actualizar información se toma una submuestra de viviendas de estas UPM en una segunda etapa (Rodriguez y Heredia, 2023).

1.2. Objetivo general

El objetivo de este trabajo es probar un método de aprendizaje estadístico no supervisado para clasificar a los municipios de los Estados Unidos Mexicanos, de acuerdo al estrato sociodemográfico al que pertenecen y utilizando la información de su ubicación geográfica para complementar este proceso de clasificación. Esta idea surgió con la motivación de querer obtener estratos de muestreo a nivel nacional como los que genera el INEGI en la primera etapa de la estratificación para el diseño de su Muestra Maestra, es decir, cuando se crean los cuatro estratos sociodemográficos a nivel nacional. Cabe destacar que no se pretende sustituir el trabajo realizado por el INEGI, sino sólo probar una metodología para la definición de los estratos.

Se utilizó información a nivel municipal ya que es la información más desagregada y completa disponible para el público en general, y se trabajó con la información de los municipios existentes hasta el 2020 según el Censo de Población y Vivienda realizado por el INEGI de dicho año. Para clasificar los municipios se utilizó el algoritmo de análisis de conglomerados jerárquico con restricciones espaciales, empleando el método de enlace

Ward. Este método fue propuesto por Chavent et al. (2018), para el cual son necesarios los siguientes elementos:

- Una matriz de distancias entre las características sociodemográficas.
- Una matriz de distancias geográficas.
- Un parámetro de mezcla $\alpha \in [0,1]$, en donde si $\alpha = 0$ significa que los grupos son creados sólo considerando las características sociodemográficas y si $\alpha = 1$ quiere decir que todo el peso se le da a la ubicación geográfica.
- Los pesos de las observaciones, denotados como w_i.
- El número k de grupos en los que se clasificarán las observaciones.

Para las características sociodemográficas se utilizaron dos bases de datos: los Principales Resultados por Localidad del Censo de Población y Vivienda 2020 [1] y los Indicadores de pobreza municipal 2015 [2]. Para las distancias geográficas, la información espacial de los municipios se obtuvo del Marco Geoestadístico diseñado por el INEGI [3] y las distancias fueron obtenidas de dos formas distintas; en la primera se calculó la distancia recta entre los centroides de los municipios, y la segunda se basó en analizar la adyacencia entre los municipios.

Respecto a los parámetros; por practicidad se le dio el mismo peso w_i a todas las observaciones, para el parámetro α se decidió aprovechar el poder computacional actual y utilizar una malla de 21 puntos, y para determinar el número de grupos k se utilizaron diversas técnicas como el método del codo, la silueta y gap, con lo que al final se decidió experimentar con $k = 3$, 4 y 5. Debido a que se ejecutó el algoritmo con las combinaciones de estos parámetros y las dos matrices de distancias, se originaron 126 distintas estratificaciones.

Para elegir la mejor estratificación lo ideal sería obtener muestras a nivel hogar/persona y evaluar las estratificaciones emulando lo que sería el uso que tiene la Muestra Maestra. Dado que esto no es posible se utilizó la base de datos de las Estadísticas Censales a Escalas Geoelectorales [4] ya que es un nivel más desagregado y público que a nivel municipal y con esto se pueden tomar muestras para evaluar las estratificaciones.

Primero se clasificaron a los distritos electorales de acuerdo a las 126 estratificaciones obtenidas anteriormente y después se seleccionaron diez variables: población de 15 años o más con primaria incompleta, población de 8 a 14 años que no sabe leer ni escribir, población de 15 años o más analfabeta, población con discapacidad, población de 3 años y más que habla alguna lengua indígena, población sin afiliación a servicios de salud, viviendas particulares habitadas con piso de tierra, viviendas particulares habitadas que no disponen

[1] Disponible en:
https://www.inegi.org.mx/programas/ccpv/2020/

[2] Disponible en:
https://www.coneval.org.mx/Medicion/Paginas/Programas_BD_municipal.aspx

[3] Disponible en:
https://www.inegi.org.mx/app/biblioteca/ficha.html?upc=889463807469

[4] Disponible en: https://www.inegi.org.mx/programas/ccpv/2020/

de energía eléctrica, viviendas particulares habitadas que no disponen de agua entubada, y viviendas particulares habitadas que no disponen de drenaje. Para cada variable, en cada estratificación, se calculó el estimador para totales Horvitz-Thompson en muestreo estratificado con muestreo aleatorio simple (m.a.s) al interior de cada estrato, para así poder obtener la varianza de cada estimador y conservar aquella estratificación en donde, en general, para todas las variables el estimador tuviera la menor varianza poblacional. La estratificación elegida fue la obtenida con $\alpha = 0.25$, $k = 5$ y la matriz de distancias entre centroides.

Con el propósito de analizar el desempeño de la estratificación elegida se comparó la varianza poblacional resultante de esta estratificación con la de un m.a.s, ambos diseños con el mismo tamaño de muestra. En este caso, como lam varianza de la estratificación seleccionada fue menor que la del m.a.s., se consideró exitosa en el sentido de que se redujo el error cuadrático medio de los estimadores.

Adicionalmente, se clasificó a los municipios de acuerdo a otras estratificaciones que podrían resultar de interés (por ejemplo, la obtenida solo con la división por entidad federativa), y de forma análoga se calcularon las varianzas poblacionales para éstas nuevas estratificaciones y se compararon con la de la estratificación elegida. Se observó que la estratificación elegida también ayudo a mejorar la precisión de los estimadores.

La tesis está organizada de la siguiente manera. En el capítulo dos se dan las nociones básicas de muestreo. En el capítulo tres se define qué es el aprendizaje estadístico no supervisado y en qué reside una de sus técnicas principales: el análisis de conglomerados jerárquico. También se describe cómo considerar el uso de restricciones espaciales en dicho método. En el capítulo cuatro se realizan distintas estratificaciones de los municipios de los Estados Unidos Mexicanos y se evalúa su desempeño para elegir la mejor. Además, se comparan los resultados obtenidos con la estratificación elegida contra los que se hubieran obtenido al realizar un muestreo aleatorio simple y otras estratificaciones como por entidad federativa, tipo de municipio y el cruce entre éstas. Finalmente, en el capítulo cinco se dan comentarios finales.

Capítulo 2

Conceptos básicos de muestreo

En este capítulo se dará una introducción básica acerca de muestreo: qué es un diseño muestral, en qué consisten los diseños muestrales con estratificación, qué es un estimador y para qué sirve, el estimador Horvitz-Thompson para totales y una forma para calcular el tamaño de la muestra. Se comenzará definiendo algunos conceptos básicos, basados en Lohr (2019, pag. 3).

- **Población**: Se define como población al conjunto U y sus elementos se denotan como $\{u_1, u_2, ..., u_N\}$, donde N es el tamaño de la población.
- **Muestra**: Subconjunto de una población.
- **Parámetro**: Comúnmente denotado por θ, es la medida numérica que resume la información de una característica de la población. Por ejemplo, la edad promedio de los estudiantes inscritos en una universidad o el ingreso salarial total por familia.
- **Censo**: Al ejercicio de medir una o más características en todos los elementos de una población se le llama censo.

Debido a la dificultad y el costo asociados con la realización de un censo, surge la alternativa de tomar mediciones solo en un subconjunto de la población, conocido como muestra. El reto principal del muestreo es obtener una muestra que refleje la información correspondiente al parámetro de interés de toda la población. Ya sea en el censo o al utilizar una muestra, se asume que las mediciones necesarias para calcular los parámetros se llevan a cabo sin error, lo que significa que los valores de las mediciones $y_1, ..., y_N$ son fijos.

Las estimaciones del parámetro de interés a menudo presentan un error aleatorio inherente debido a que se basan únicamente en un subconjunto de la población. La magnitud de este error dependerá del diseño de muestreo utilizado (cómo se selecciona la muestra) y del estimador utilizado.

2.1. Diseños de muestreo

Sea U la población de interés de tamaño N, donde sus elementos se denotan como $u_1, u_2, ..., u_N$; y cada unidad u_k tiene un valor fijo asociado y_k.

El objetivo es estimar el parámetro de interés θ, que es una función de $y_1, ..., y_N$, usando una muestra s, la cual es cualquier subconjunto de U.

Nótese que hay 2^N diferentes subconjuntos de U, los cuales conforman el conjunto potencia S:

$$S = \{\emptyset, \{1\}, \{2\}, ..., \{1,2\}, ..., \{N-1, N\}, ..., \{1,2,..,N\}\},$$

donde la cardinalidad de S es $|S| = \sum_{i=0}^{N} \binom{N}{i} = 2^N$.

Definición 2.1.1 ***Estimador.***
Un estimador de θ, denotado como $\hat{\theta}$, es cualquier función de los valores de y asociados a los elementos en la muestra s seleccionada:

$$\hat{\theta} = f(y_k, u_k \in s).$$

Notar que $\hat{\theta}$ tomará un valor diferente para cada muestra s. Se denotará al tamaño de la muestra s como n_s.

Definición 2.1.2 ***Diseño muestral.***
Un diseño muestral es una función $p(.)$ que representa una función de probabilidad sobre S:

$$p(s) = P(S = s) \ \forall s \in S,$$

donde $0 \leq p(s) \leq 1 \ \forall s \in S \ y \ \sum_{s \in S} p(s) = 1.$

Esta definición permite que haya elementos en S cuya probabilidad $p(s)$ sea cero, lo cual implica que no se podrían observar en la práctica. Ya que estos casos no son los de interés, se define como S_0 al subconjunto de S tal que $0 < p(s)$, dichos elementos conforman el conjunto de muestras posibles.

Definición 2.1.3 ***Diseño muestral probabilístico.***
Un diseño muestral probabilístico es un diseño muestral $p(.)$ que cumple que:

$$\forall \ u_k \in U \ \exists \ s \in S_0 \ni \ u_k \in s, \tag{2.1}$$

es decir, cada elemento de la población tiene probabilidad positiva de ser seleccionado en la muestra.

Definición 2.1.4 ***Probabilidad de inclusión.***

Sea $u_k \in U$, el evento "se observa una muestra que contiene al elemento u_k" se denotará como "$u_k \in S$". Es de interés obtener la probabilidad de este evento, la cual se conoce como probabilidad de inclusión del elemento u_k y se denota como π_k.

Definición 2.1.5 ***Función indicadora.***
Para cada elemento $u_k \in U$, la función indicadora se define como:

$$I_k = \begin{cases} 1, & si \ u_k \in S \\ 0, & en \ otro \ caso. \end{cases} \tag{2.2}$$

Es decir, $I_k = 1$ representa que se ha observado una muestra que contiene a la unidad u_k. Por lo que π_k se puede obtener fácilmente a partir de I_k y de un diseño de muestreo definido, usando la regla de probabilidad total sobre la variable aleatoria S:

$$\pi_k = P(u_k \in S) = P(I_k = 1) = \sum_{s \in S_0} I(u_k \in s) p(s). \tag{2.3}$$

De forma similar se puede obtener la probabilidad de inclusión de los elementos u_k y u_l, esto significa que se observó al elemento u_k y al elemento u_l, es decir, "$u_k, u_l \in S$", a esta probabilidad se le denota por π_{kl}. Entonces:

$$\pi_{kl} = P(u_k \in S, u_l \in S) = P(I_k = 1, I_l = 1) = \sum_{s \in S_0} I(u_k \in s, u_l \in s) p(s). \tag{2.4}$$

Se tiene el caso particular cuando $k = l$:

$$\pi_{kk} = P(u_k \in S) = P(I_k = 1) = \pi_k. \tag{2.5}$$

Una vez que se selecciona la muestra s, se tiene que:

$$I_k| \text{ “S=s”} = \begin{cases} 1, & \text{si } u_k \in s \\ 0, & \text{en otro caso.} \end{cases} \tag{2.6}$$

En muchos casos, las variables aleatorias asociadas a la muestra se podrán escribir en términos de las funciones indicadoras. Por ejemplo:

$$n_s = \sum_{u_k \in U} I_k. \tag{2.7}$$

Con función de probabilidad:

$$P(n_s = n) = \sum_{s \in S_0} I(n_s = n) p(s), \qquad n \in \{0, ..., N\}. \tag{2.8}$$

Incluso, para calcular la esperanza de n_s puede ser más conveniente usar 2.7 en lugar de la función de probabilidad, pues:

$$\begin{aligned} E(n_s) &= E(\sum_{u_k \in U} I_k) \\ &= \sum_{u_k \in U} E(I_k) \\ &= \sum_{u_k \in U} \pi_k. \end{aligned} \tag{2.9}$$

Resultado 2.1.1 *(Propiedades de las funciones indicadoras). Sea un diseño de muestreo $p(.)$ y sus respectivas probabilidades de inclusión de primer y segundo orden. Las funciones indicadoras definidas en 2.2 están asociadas a variables aleatorias que cumplen que $\forall u_k, u_l \in U$:*

$$E(I_k) = \pi_k, \tag{2.10}$$

$$V(I_k) = \pi_k(1-\pi_k), \tag{2.11}$$

$$Cov(I_k, I_l) = \pi_{kl} - \pi_k\pi_l. \tag{2.12}$$

Demostración. Por definición se tiene que $I_k \sim Bernoulli(\pi_k)$, $u_k \in U$, pues $I_k = 1$ representa que se ha observado una muestra que contiene a la unidad u_k. De lo anterior es directo que $E(I_k) = \pi_k$ y $V(I_k) = \pi_k(1-\pi_k)$. Por otro lado:

$$\begin{aligned} Cov(I_k, I_l) &= E(I_k I_l) - E(I_k)E(I_l) \\ &= P(I_k = 1, I_l = 1) - \pi_k\pi_l \\ &= \pi_{kl} - \pi_k\pi_l. \end{aligned}$$

Además, sean $I_1, ..., I_N$ las variables indicadoras de los N elementos en la población $\{u_1, u_2, ..., u_N\}$ y $\{a_k, b_k; k = 1, ..., N\}$ constantes. Entonces:

$$Cov(\sum_{u_k \in U} a_k I_k, \sum_{u_l \in U} b_l I_l) = \sum_{u_k \in U}\sum_{u_l \in U} a_k b_l Cov(I_k, I_l).$$

Esta ecuación se puede separar en dos casos, cuando $u_k = u_l$ y cuando $u_k \neq u_l$, así:

$$\begin{aligned} \sum_{u_k \in U}\sum_{u_l \in U} a_k b_l Cov(I_k, I_l) &= \sum_{u_k \in U} a_k b_k Cov(I_k, I_k) + \sum_{u_k \in U}\sum_{u_l \in U, l \neq k} a_k b_l Cov(I_k, I_l) \\ &= \sum_{u_k \in U} a_k b_k V(I_k) + \sum_{u_k \in U}\sum_{u_l \in U, l \neq k} a_k b_l Cov(I_k, I_l) \\ &= \sum_{u_k \in U} a_k b_k \pi_k(1-\pi_k) + \sum_{u_k \in U}\sum_{u_l \in U, l \neq k} a_k b_l (\pi_{kl} - \pi_k\pi_l). \end{aligned}$$

Por lo tanto,

$$Cov(\sum_{u_k \in U} a_k I_k, \sum_{u_l \in U} b_l I_l) = \sum_{u_k \in U} a_k b_k \pi_k(1-\pi_k) + \sum_{u_k \in U}\sum_{u_l \in U, l \neq k} a_k b_l (\pi_{kl} - \pi_k\pi_l). \tag{2.13}$$

Existen distintos diseños de muestreo, entre los más populares se encuentran el muestreo Bernoulli, el muestreo aleatorio simple con y sin reemplazo, y el muestreo sistemático. En este trabajo se explicará el muestreo aleatorio simple sin reemplazo, ya que es el que se utilizó en el experimento.

2.1.1. Muestreo aleatorio simple sin reemplazo *(m.a.s)*

Este diseño de muestreo es el más sencillo de todos y es muy usado (aunque costoso), ya que permite la obtención de expresiones sencillas para los estimadores de totales, en particular, aquellos basados en el estimador Horvitz–Thompson. Considerando una población de tamaño N, $U = \{u_1, u_2, ..., u_N\}$, y un valor n fijo, $n \leq N$, el *m.a.s* es aquel en donde cada muestra de tamaño n tiene la misma probabilidad de ser seleccionada, es decir,

$$S_0 = \{s_1 = \{1, 2, ..., n\}, ..., s_{\binom{N}{n}} = \{N - n + 1, .., N\}\}, \quad p(s_i) = \frac{1}{\binom{N}{n}} \ \forall s_i \in S_0. \tag{2.14}$$

Con este diseño se cumple que $\pi_k = \frac{n}{N}$ y que $\pi_{kl} = \frac{n(n-1)}{N(N-1)}$, $u_k \neq u_l$, $u_k, u_l \in U$.

Para tomar una muestra aleatoria simple, se necesita una lista de todas las unidades en la población; esta lista es el llamado marco de muestreo.

El método más simple para seleccionar una muestra con este diseño consiste en generar N números aleatorios entre 0 y 1, luego seleccionar las unidades correspondientes a los n números aleatorios más pequeños para que sean la muestra. Por ejemplo, si $N = 10$ y $n = 4$, generamos 10 números aleatorios entre 0 y 1:

unidad i	1	2	3	4	5	6	7	8	9	10
número aleatorio	0.837	0.636	0.465	0.609	0.154	0.766	0.821	0.713	0.987	0.469

Los 4 números aleatorios más pequeños son 0.154, 0.465, 0.469 y 0.609, lo que resulta en la muestra con las unidades u_3, u_4, u_5 y u_{10}.

2.2. El estimador Horvitz-Thompson para totales

De acuerdo a la definición 2.1.1, un estimador es una función que se aplica a los datos muestrales para obtener una aproximación de un parámetro poblacional desconocido θ.

Muchas veces se está interesado en calcular el total de una variable y. Esto puede ser el total de viviendas con electricidad o el total de personas que terminaron la educación media superior. Horvitz y Thompson (1951) propusieron un estimador para totales $\theta = \sum_{k=1}^{N} y_k$ también conocido como $\pi-$estimador, el cual continua siendo en la actualidad uno de los más importantes. Sea U la población de interés, se define el $\pi-$estimador para totales como:

$$\hat{\theta} = \hat{t}_{y\pi} = \sum_{u_k \in S} \frac{y_k}{\pi_k} = \sum_{u_k \in S} w_k y_k, \tag{2.15}$$

donde y_k es el valor asociado al elemento de la población u_k. En este estimador, $\frac{1}{\pi_k} = w_k$ actúa como ponderador del valor y_k. Al conjunto $\{w_1, w_2, ..., w_N\}$ se le conoce como el conjunto de factores de expansión o pesos muestrales, los cuales se usan para asignar un peso relativo a cada unidad de la muestra.

2.2.1. Propiedades básicas

El estimador $\hat{t}_{y\pi}$ tiene las siguientes propiedades:

1. *Es un estimador insesgado para t_y, es decir,*

$$E(\hat{t}_{y\pi}) = t_y = \sum_{k=1}^{N} y_k. \tag{2.16}$$

2.

$$V(\hat{t}_{y\pi}) = \sum_{k=1}^{N}\sum_{l=1}^{N} \Delta_{kl}\frac{y_k}{\pi_k}\frac{y_l}{\pi_l}, \tag{2.17}$$

donde $\Delta_{kl} = \pi_{kl} - \pi_k\pi_l$.

Demostración. Primero se observa que:

$$\hat{t}_{y\pi} = \sum_{u_k\in S}\frac{y_k}{\pi_k} = \sum_{u_k\in U} I_k\frac{y_k}{\pi_k},$$

es decir, es un estimador lineal en términos de las variables aleatorias $I_1, I_2, ..., I_N$.

I. $$\begin{aligned} E(\hat{t}_{y\pi}) &= E(\sum_{u_k\in U} I_k\frac{y_k}{\pi_k}) \\ &= \sum_{u_k\in U} E(I_k\frac{y_k}{\pi_k}) \\ &= \sum_{u_k\in U}\frac{y_k}{\pi_k}E(I_k) \\ &= \sum_{u_k\in U}\frac{y_k}{\pi_k}\pi_k \\ &= \sum_{u_k\in U} y_k = t_y \end{aligned}$$

II. $$\begin{aligned} V(\hat{t}_{y\pi}) &= V(\sum_{u_k\in S}\frac{y_k}{\pi_k}) \\ &= V(\sum_{u_k\in U} I_k\frac{y_k}{\pi_k}) \\ &= Cov(\sum_{u_k\in U} I_k\frac{y_k}{\pi_k}, \sum_{u_l\in U} I_l\frac{y_l}{\pi_l}) \end{aligned}$$

Como $\frac{y_k}{\pi_k}, k = 1, ..., N$, son constantes, se utiliza 2.13, entonces:

$$V(\hat{t}_{y\pi}) = \sum_{u_k\in U}\frac{y_k}{\pi_k}^2(1-\pi_k) + \sum_{u_k\in U}\sum_{u_l\in U, l\neq k}\frac{y_k}{\pi_k}\frac{y_l}{\pi_l}(\pi_{kl} - \pi_k\pi_l). \tag{2.18}$$

A $V(\hat{t}_{y\pi})$ se le conoce como la **varianza poblacional** o teórica y se debe notar que depende de todas las observaciones, es decir, de todos los valores $y_1, ..., y_N$, por lo que no se puede calcular con una muestra especifica.

Resultado 2.2.1 *Por ser un estimador insesgado, el error cuadrático medio del estimador HT es igual a su varianza, ya que:*

$$\begin{aligned} MSE(\hat{t}_{y\pi}) &= E\left[(\hat{t}_{y\pi} - t_y)^2\right] \\ &= E(\hat{t}_{y\pi}^2) - 2t_y E(\hat{t}_{y\pi}) + t_y^2 + E(\hat{t}_{y\pi})^2 - E(\hat{t}_{y\pi})^2 \\ &= E(\hat{t}_{y\pi}^2) - E(\hat{t}_{y\pi})^2 + \left[E(\hat{t}_{y\pi}) - t_y\right]^2 \end{aligned}$$

Como $E(\hat{t}_{y\pi}) = t_y$, *entonces* $\left[E(\hat{t}_{y\pi}) - t_y\right]^2 = 0$, *y por lo tanto*

$$MSE(\hat{t}_{y\pi}) = V(\hat{t}_{y\pi}). \tag{2.19}$$

2.2.2. Estimador HT bajo un m.a.s

Si se tiene una población $U = \{u_1, u_2, ..., u_N\}$ y una muestra de tamaño fijo n, donde el diseño de muestreo es aleatorio simple sin reemplazo, el estimador para totales se transforma en:

$$\hat{t} = \sum_{u_k \in S} \frac{y_k}{\pi_k} = \frac{N}{n} \sum_{u_k \in S} y_k = N\bar{y}_S,$$

esto porque $\pi_k = \frac{n}{N}$ y $\pi_{kl} = \frac{n(n-1)}{N(N-1)}, k \neq l, u_k, u_l \in U$.

Partiendo de 2.18 y utilizando este diseño de muestreo, se puede encontrar una expresión más sencilla para $V(\hat{t}_{y\pi})$, tal que:

$$V(\hat{t}_{y\pi}) = \frac{N^2}{n}\left(1 - \frac{n}{N}\right) S_{yU}^2. \tag{2.20}$$

Donde $S_{yU}^2 = \frac{\sum_{k=1}^{N}(y_k - \bar{y_U})^2}{(N-1)}$ y $\bar{y_U} = \frac{\sum_{k=1}^{N} y_k}{N}$.

También se puede obtener que:

$$\hat{V}(\hat{t_{y\pi}}) = \frac{N^2}{n}\left(1 - \frac{n}{N}\right) S_{yS}^2. \tag{2.21}$$

Donde $S_{yS}^2 = \frac{\sum_{k \in S}(y_k - \bar{y_S})^2}{(n-1)}$ y $\bar{y_S} = \frac{\sum_{k \in S} y_k}{n}$.

2.3. Determinación del tamaño de la muestra

Una de las preguntas fundamentales en muestreo es qué tamaño de muestra n es el adecuado. Para responder esta interrogante se debe decidir qué cantidad de error en las estimaciones es tolerable y equilibrarlo con el costo asociado a tomar una muestra de ese tamaño. Para esto se deben de tener definidos los parámetros de interés, el diseño muestral y los estimadores a usar. El tamaño de muestra se debe calcular para cada parámetro de interés y cada dominio de estudio. En general, el cálculo con respecto a los dominios de estudio se realiza de forma jerárquica, empezando con los dominios de estudio de menor cardinalidad para que de esta manera se vayan acumulando los tamaños de muestra.

El tamaño de muestra hace referencia a las unidades en la población de interés que se deben seleccionar. Por ejemplo, si el parámetro de interés fuera el número de hogares

con electricidad, el tamaño de la muestra se calcularía sobre las viviendas. En cambio, si el parámetro de interés fuera el número de personas con secundaria incompleta, el tamaño de la muestra se calcularía sobre los individuos.

Por otro lado, dependiendo del diseño de muestreo cada estimador tiene su propia distribución y por esto, para un tamaño de muestra dado, cada uno proporciona un diferente desempeño y una diferente incertidumbre. La precisión de los estimadores se puede acotar considerando:

$$P\left(|\hat{\theta}-\theta| \leq k\sqrt{V(\hat{\theta})}\right) = 1-\alpha, \tag{2.22}$$

donde k se determina dependiendo del valor de α y de la distribución de $\hat{\theta}$. En particular, para el estimador Horvitz-Thompson y n grande, se suele usar $k = z_{1-\frac{\alpha}{2}}$. Además

$$k\sqrt{V(\hat{\theta})}, \tag{2.23}$$

es una cota en términos absolutos sobre el posible error $|\hat{\theta}-\theta|$ que se observaría con una confianza $1-\alpha$ establecida. Es decir, si se repitiera la obtención de una muestra B veces considerando un diseño de muestreo fijo, entonces se espera que en $(1-\alpha)x100\,\%$ de las B muestras, $|\hat{\theta}-\theta|$ fuera menor o igual a $k\sqrt{V(\hat{\theta})}$.

Con todo esto, se puede proporcionar una cota sobre el error absoluto deseado d. Por ejemplo, si es de interés que $|\hat{\theta}-\theta| \leq d$, esto se puede lograr con una confianza de $1-\alpha$ si

$$k\sqrt{V(\hat{\theta})} \leq d. \tag{2.24}$$

Para la determinación del tamaño de muestra se requiere encontrar una relación de $k\sqrt{V(\hat{\theta})}$ con el valor n asociado al tamaño de la muestra. En general esto no es fácil, pero hay estimadores y diseños de muestreo en donde sí es posible.

2.3.1. Cálculo del tamaño de muestra a partir del estimador HT para totales y con un diseño de muestreo aleatorio simple

Como en este caso el tamaño de muestra se quiere calcular utilizando el estimador HT para totales, en lo siguiente se considera que $k = z_{1-\frac{\alpha}{2}}$, $\theta = t_y$ y $\hat{\theta} = \hat{t}_{\pi y} = \sum_{u_k \in S} \frac{y_k}{\pi_k}$, por lo que la ecuación 2.24 se convierte en:

$$k\sqrt{\frac{N^2}{n}(1-\frac{n}{N})S^2_{yU}} = k\sqrt{N^2(\frac{1}{n}-\frac{1}{N})S^2_{yU}} \leq d \tag{2.25}$$

y despejando n se obtiene:

$$n \geq \frac{1}{\frac{d^2}{N^2k^2S^2_{yU}}+\frac{1}{N}} = \frac{(\frac{kNS^2_{yU}}{d})^2}{1+\frac{1}{N}(\frac{kNS^2_{yU}}{d})^2}, \tag{2.26}$$

donde n, el menor valor que cumple esa desigualdad, es el tamaño de muestra recomendado.

2.4. Muestreos estratificados

En la práctica, a veces se cuenta con información adicional la cual puede ayudar a mejorar el diseño de muestreo. Si la variable en la que se está interesado toma diferentes valores promedio en distintas subpoblaciones, es posible que se puedan obtener estimaciones más precisas de las cantidades tomando una muestra aleatoria estratificada. Esto consiste en dividir a la población en H subpoblaciones llamadas estratos.

Definición 2.4.1 ***Diseños estratificados***
Sea

$$U = \{u_1, u_2, ..., u_N\}$$

la población de interés de tamaño N. La población U se particiona en H subconjuntos, denotados como $U_h,\ h = 1, ..., H$. *Estos subconjuntos son llamados estratos y cumplen por definición:*

$$\bigcup_{h=1}^{H} U_h = U,\ U_i \neq \emptyset\ \ y\ \ U_i \cap U_j = \emptyset\ \forall\ i \neq j; i, j = 1, ..., H.$$

El número de unidades en el estrato h se denota como N_h y $\sum_{h=1}^{H} N_h = N$.
Se dice que un diseño es estratificado si en cada estrato h:

- Se selecciona una muestra aleatoria s_h de tamaño $n_h > 0$, siguiendo un diseño de muestreo probabilístico $p_h(s_h)$.
- La selección de la muestra en ese estrato se realiza de forma independiente del resto de estratos.

Es decir, a cada estrato o subconjunto U_h se le considera como una población por sí mismo. Con estas condiciones la muestra s se obtiene como:

$$s = \bigcup_{h=1}^{H} s_h,$$

además $n_s = \sum_{h=1}^{H} n_h$, y dada la independencia en la selección de la muestra entre estratos se tiene que:

$$P(S = s) = p(s) = \prod_{h=1}^{H} p_h(s_h).$$

Con esta definición, el diseño de muestreo $p_h(.)$ usado en cada estrato podría ser diferente.

2.4.1. El estimador Horvitz-Thompson en muestreo estratificado con m.a.s al interior de cada estrato

Resultado 2.4.1 *El estimador Horvitz-Thompson en un diseño de muestreo con estratificación y m.a.s al interior de cada estrato se puede expresar como:*

$$\hat{t}_{y\pi} = \sum_{h=1}^{H} \hat{t}_{h\pi} = \sum_{h=1}^{H} N_h \sum_{u_k \in S_h} \frac{y_k}{n_h} = \sum_{h=1}^{H} N_h \bar{y}_{S_h}, \tag{2.27}$$

donde $\hat{t}_{h\pi} = \sum_{u_k \in S_h} \frac{y_k}{\pi_k}$ *es el estimador de* $t_h = \sum_{u_k \in Uh} y_k$.
Además:

$$V(\hat{t}_{y\pi}) = \sum_{h=1}^{H} V(\hat{t}_{h\pi}) = \sum_{h=1}^{H} \frac{N_h^2}{n_h}\left(1 - \frac{n_h}{N_h}\right) S_{yU_h}^2 = \sum_{h=1}^{H} \frac{N_h^2}{n_h} S_{yU_h}^2 - \sum_{h=1}^{H} N_h S_{yU_h}^2, \tag{2.28}$$

donde $V(\hat{t}_{h\pi})$ *es la varianza de* $\hat{t}_{h\pi}$, $S_{yU_h}^2 = \frac{\sum_{u_k \in U_h}(y_k - \bar{y}_{U_h})^2}{N_h - 1}$ *y* $\bar{y}_{U_h} = \frac{\sum_{u_k \in U_h} y_k}{N_h}$.

Demostración. Dado que $s = \bigcup_{h=1}^{H} s_h$, entonces:

$$\begin{aligned} \hat{t}_{y\pi} &= \sum_{u_k \in S} \frac{y_k}{\pi_k} \\ &= \sum_{h=1}^{H} \sum_{u_k \in S_h} \frac{y_k}{\pi_k} \\ &= \sum_{h=1}^{H} \hat{t}_{h\pi}. \end{aligned} \tag{2.29}$$

Usando la independencia de los diseños de muestreo de cada estrato se obtiene lo correspondiente a la varianza.

2.4.2. Construcción de los estratos

Los aspectos que se consideran para la construcción de estratos son los siguientes:

- **Reducción del ECM de los estimadores.** Si se desea estimar el total de la variable y, se sugiere construir los estratos de modo que la variable sea muy homogénea en cada uno de ellos. Para lograr esto, se puede utilizar información auxiliar de una o más variables que estén relacionadas con la variable de interés. De esta forma, utilizando cualquier método de aprendizaje no supervisado para clasificación, se pueden identificar grupos homogéneos en relación a esas variables, lo que a su vez producirá cierta homogeneidad con respecto a la variable y.

- **Aprovechar divisiones ya creadas.** Es posible utilizar divisiones previamente establecidas, que históricamente presentan comportamientos diferenciados sobre la variable de estudio. Por ejemplo, áreas geográficas como entidades federativas o características demográficas como el sexo y la edad.

- **Mejorar la representación y credibilidad del estudio.** En un muestreo unietápico sin estratificación no es posible garantizar que la muestra seleccionada contenga unidades de diferentes grupos que se deseen representar; mientras que en un muestreo estratificado, se podrían incluir estratos construidos de manera que esos grupos se vean representados. Por ejemplo, en un muestreo aleatorio simple de hogares en el territorio nacional, se podría obtener una muestra donde no existan hogares de cierta entidad federativa, mientras que si se crean estratos a partir de las entidades federativas esto no podría pasar.

- **Coincidencia entre estratos y dominios de estudio**. En múltiples ocasiones, se busca generar resultados para distintos dominios de estudio o subpoblaciones. Dado que la selección en cada estrato es independiente y se asegura la selección de muestra en cada uno, estos podrían coincidir con dominios de estudio, facilitando la estimación, selección de la muestra y definición del tamaño de la muestra.

- **Facilitar la estimación y selección de la muestra.** En términos de mejorar la precisión de las estimaciones en un muestreo, una opción es utilizar un muestreo proporcional al tamaño, sin embargo, la selección y estimación de la varianza pueden ser complicadas. Por otro lado, el uso de un muestreo estratificado podría ser beneficioso ya que se puede aprovechar la información auxiliar representada en la variable tamaño durante la construcción de los estratos, y el diseño dentro de cada estrato podría ser un aleatorio simple, lo que facilita la estimación.

- **Costo**. En general, la implementación de un diseño estratificado puede ser costosa. Además, el número de estratos define un límite inferior en el tamaño de la muestra, ya que es necesario seleccionar muestra en cada estrato, entonces $n \geq H$. Además, para estimar la varianza en cada estrato, se recomienda seleccionar más de una unidad por estrato y considerar la posibilidad de que exista no respuesta. De lo contrario, podría ser necesario fusionar estratos para poder estimar las varianzas.

2.5. Efecto de diseño

En 1951, Cornfield sugirió medir la eficiencia de un diseño de muestreo mediante la razón entre la varianza que se obtendría de un muestreo aleatorio simple y la varianza obtenida a partir del diseño de muestreo elegido, ambos con n unidades de observación. En 1965, Kish llamó al recíproco de la razón de Cornfield como *deff* y lo utilizó para resumir el efecto del diseño en la varianza de la estimación (Sarndal et al., 1992, pag. 53).

Cuando el *deff* es menor a 1, significa que el diseño de muestreo es más eficiente que un m.a.s, y hay un efecto significativo del diseño en la varianza. Sin embargo, cuando el *deff* es mayor que 1, indica que el diseño de muestreo aumenta la varianza en comparación con un m.a.s, lo que significa que se necesita un tamaño de muestra mayor para lograr la misma precisión en las estimaciones. El *deff* es calculado de la siguiente manera:

$$deff = \frac{V_{\text{complejo}}(\hat{\theta})}{V_{\text{mas}}(\hat{\theta})}. \tag{2.30}$$

Este trabajo se centrará en el primer aspecto, es decir, usar una técnica de análisis de conglomerados para la definición de estratos con el objetivo de reducir el ECM de los estimadores. Este capítulo se baso principalmente en Pérez de la Cruz (2023), Tillé (2020, sección 3.1) y Lohr (2019, caps. 1, 2 y 4). Se recomienda al lector revisar estos para detalles que no se exponen en este trabajo.

Capítulo 3

Aprendizaje estadístico no supervisado

En este capítulo se definirá qué es el aprendizaje estadístico no supervisado y en qué consiste una de sus técnicas principales: el análisis de conglomerados jerárquico. También se describirá cómo considerar el uso de restricciones espaciales en dicho método. En esta sección se usará la notación n, p y k como en la mayoría de libros sobre análisis de conglomerados, pese a que en el capítulo anterior tenían otro significado.

De acuerdo con James et al. (2021), el aprendizaje estadístico se refiere a un conjunto de herramientas cuyo objetivo principal es obtener el mayor conocimiento posible de determinados datos. Puede ser clasificado como aprendizaje estadístico supervisado y no supervisado. En general, el aprendizaje supervisado consiste en construir un modelo estadístico para predecir o estimar un *output* basado en uno o más *inputs*. Por otro lado, el aprendizaje no supervisado no tiene asociada una variable respuesta, por lo que no interesa realizar predicciones y en su lugar se enfoca en encontrar subgrupos homogéneos dentro de una población y en hallar patrones en las características de los datos. Las dos técnicas más populares para esto son el análisis de conglomerados y el análisis de componentes principales.

3.1. Análisis de conglomerados jerárquico

El objetivo principal del análisis de conglomerados es descubrir grupos que sean de interés en los datos. En particular, dado un conjunto de datos con n observaciones y donde para cada observación se tiene la información de p variables, la meta es identificar k grupos con base en las p variables tales que:

- Cada grupo debe contener al menos una observación.
- Cada observación pertenezca a un sólo grupo.

Esto define una partición del conjunto de n unidades en k subconjuntos llamados conglomerados o clústers. Como una gran cantidad de particiones pueden ser encontradas, la búsqueda se restringe a encontrar grupos en donde, respecto a las p variables, las observaciones sean similares entre ellas y diferentes a las de otros grupos.

Actualmente hay un interés creciente en el análisis de conglomerados, pues hay una necesidad de encontrar grupos en distintos campos de estudio. Por ejemplo, en marketing, con base en variables como características demográficas y de comportamiento, se pueden clasificar a los clientes en distintos grupos y así realizar una publicidad especifica para cada uno o identificar a clientes potenciales. En medicina se puede utilizar en la investigación sobre cáncer para clasificar a los pacientes en subgrupos según su perfil de expresión génica y esto puede ser útil para identificar el perfil molecular de pacientes con buen o mal pronóstico, así como para entender la enfermedad y definir el tratamiento más adecuado.

A pesar de que el análisis de conglomerados es bastante intuitivo, su formalización no es sencilla. De hecho posee problemas y retos con los que el usuario debe lidiar. El primero de ellos es que no se existe un conocimiento a priori de la existencia y del número de grupos. Otro problema es el cómo evaluar los grados de similitud o disimilitud entre las observaciones ya que, al contrario del aprendizaje supervisado, en esta clase de técnicas no se cuenta con una variable respuesta por lo que es difícil medir su eficacia. Finalmente, hablando del método jerárquico, una decisión relevante concierne a la elección del método de enlace y la distancia a usar. La peculiaridad del método jerárquico es que no conduce a una sola partición con un número determinado de conglomerados, sino que produce una serie de particiones definidas por una estructura jerárquica (Giordani et al., 2020), por lo que puede ser más amigable que otros métodos ya que no requiere elegir un número k previo de grupos (como en el método de *K-means*). Para funcionar, este método requiere una matriz de distancias entre las observaciones.

3.1.1. Algoritmo

El análisis de conglomerados jerárquico produce una serie de particiones donde los dos conglomerados más similares son sucesivamente fusionados.

Sea D_n la matriz con las distancias entre cada par de observaciones y D_k, $k \leq n$, una matriz de distancias entre los k conglomerados, el algoritmo es el siguiente:

0. Se comienza con la partición en donde cada una de las n observaciones es tratada como un conglomerado. Sea $r = 0$.
1. De acuerdo con D_{n-r}, se fusionan los dos conglomerados con la mínima distancia entre ellos, con lo que resulta una nueva partición de $n - r - 1$ conglomerados.
2. Se obtiene una nueva matriz de distancias entre conglomerados D_{n-r-1}.
3. Si $n - r - 1 = 1$ parar, de otra forma $r = r + 1$, y repetir los pasos 1 y 2.

En el último paso, D_2 tiene orden (2×2) y contiene las medidas de distancia entre los dos conglomerados que son fusionados para obtener la partición trivial y final con un conglomerado de n observaciones. Existen diversas formas de calcular D_k, las cuales dependen del método de enlace que se utilice (James et al., 2021).

3.1.2. Métodos de enlace

El punto crucial en el análisis de conglomerados jerárquico radica en la elección del método para calcular las distancias entre conglomerados (también llamado método de enlace). Los más utilizados se encuentran resumidos en la Figura 3.1 y asumen que ya se ha definido una distancia $d()$ a usar. En esta figura se considera que A y B son conglomerados cualquiera con $i \in A$ y $j \in B$.

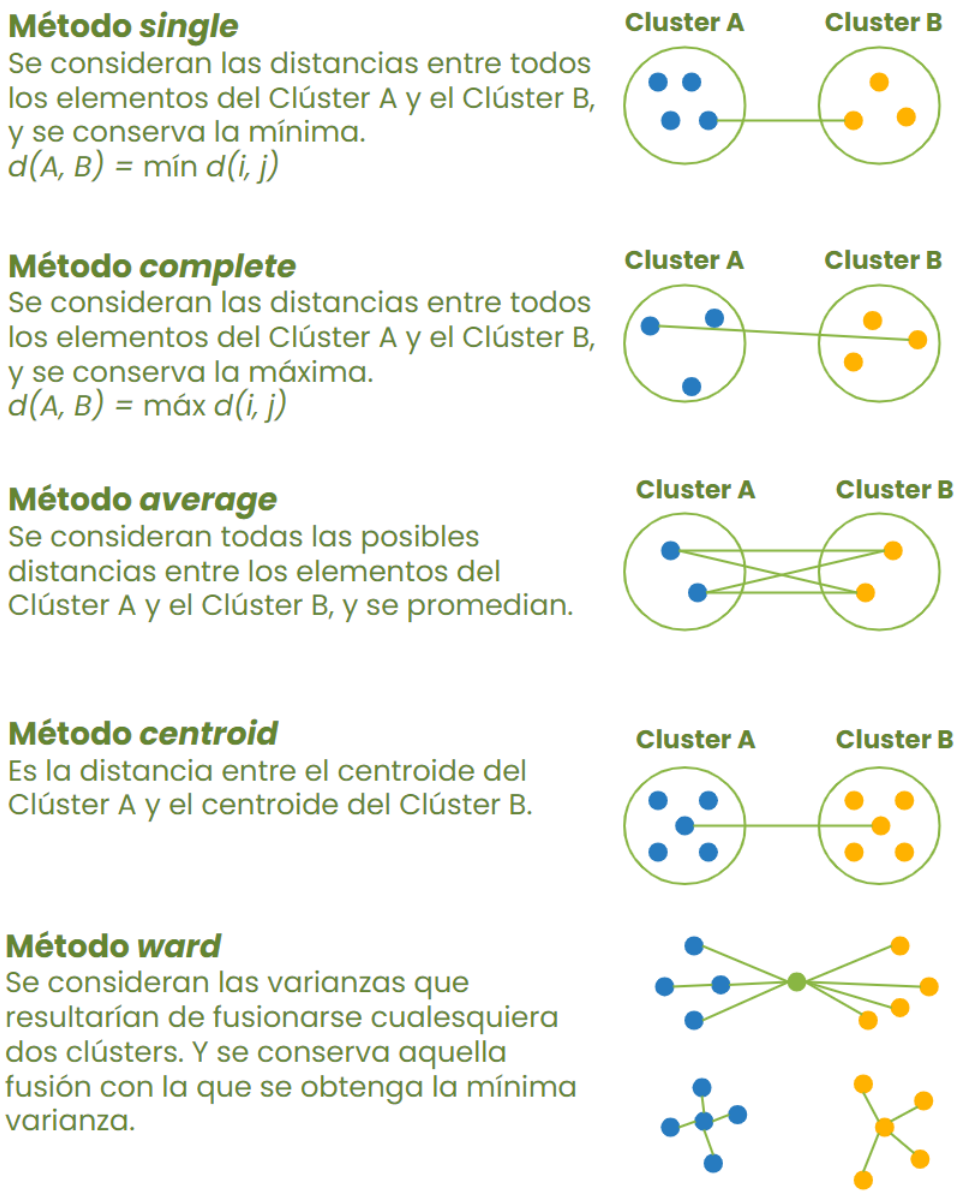


Figura 3.1: Resumen de los métodos más usados para calcular la distancia entre conglomerados. Imagen de autoría propia.

Con cada método se obtienen distintas ventajas y desventajas, y se deberá usar el más conveniente dependiendo del contexto. A continuación se enlistan las más comunes.

- Método *single*: En este método la distancia entre los conglomerados es la distancia entre sus elementos más cercanos, así que solo controla la similitud entre los vecinos más próximos. Por esta razón tiene buen desempeño con conglomerados no elípticos (siempre y cuando la distancia entre ellos no sea tan pequeña) y no tiende a separar grupos grandes, sin embargo, es sensible a *outliers*. También suele separar mal los grupos si hay ruido entre ellos, ya que puede provocar la fusión prematura de grupos con pares cercanos incluso si esos grupos son bastante diferentes en general.

- Método *complete*: La proximidad entre dos grupos es igual a la distancia entre sus dos objetos más distantes, por lo que tiende a producir conglomerados más compactos y no es tan sensible al ruido entre ellos ni a valores atípicos. Funciona mejor en formas elípticas, pero tiende a clasificar mal conglomerados grandes.

- Método *average*: Toma todos los pares de puntos entre un conglomerado y el otro y calcula el promedio de estas distancias, por esta razón suele ser más estable que los métodos *single* y *complete*, y no es tan sensible al ruido entre conglomerados ni a valores atípicos. Sin embargo, los resultados pueden ser variados, por ejemplo, en la Figura 3.2 al utilizarlo se separaron mal los grupos grandes y las formas no elípticas.

- Método *centroid*: La proximidad entre dos grupos es la distancia entre sus centroides geométricos. Estos grupos pueden estar fragmentados y a menos que sus figuras centrales estén separadas entre sí la unión será consistente. No es tan sensible al ruido entre conglomerados ni a valores atípicos. Esta es una de las técnicas menos utilizadas en la práctica ya que es caro computacionalmente.

- Método *ward*: Tiene como objetivo minimizar la varianza total dentro del grupo. Dados k conglomerados, este método permite la reducción a $k-1$ grupos considerando la unión de todas las posibles parejas de conglomerados que pueden ser formadas y conservando aquella en donde el valor de la suma de las desviaciones al cuadrado (ESS)[1] se minimice. Fue creado por Ward J. en 1963[2], donde:

$$ESS = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 . \tag{3.1}$$

  A pesar de que funciona mejor con conglomerados elípticos no es sensible al ruido entre ellos ni a valores atípicos. Tampoco tiende a separar grupos grandes y suele ser el mejor método para separar conjuntos de datos donde hay mucho ruido.

Con el propósito de ilustrar las ventajas y desventajas de cada método de enlace descrito previamente se simularon cuatro conjuntos de datos en dos dimensiones. Utilizando la distancia euclidiana, a cada conjunto de datos se le aplicaron los métodos de enlace: *single*, *complete*, *average*, *centroid* y *ward*, y se obtuvieron los resultados mostrados en la Figura 3.2. En el anexo B se detalla cómo fueron simulados los conjuntos de datos.

[1] Sum of the squared deviations, por sus siglas en inglés.
[2] En *Hierarchical Grouping to Optimize an Objective Function* por Ward (1963).

Como se puede observar en la Figura 3.2, con cada método de enlace se generaron distintos resultados, los cuales se pueden resumir como sigue:

- En el conjunto A los datos se clasificaron en tres grupos y debido al ruido entre ellos el método *single* no logró clasificar bien y utilizando este método se obtuvieron dos grupos con una sola observación. El resto de los métodos lo realizó de forma satisfactoria.
- Dado que la forma de los datos en el conjunto B no es elíptica, todos los métodos a excepción del *single* tienen problemas con la clasificación. Sin embargo, el método *Average* casi logra una clasificación correcta, a excepción de un conjunto de puntos naranjas en el triángulo de abajo.
- Debido a que no hay ruido entre grupos en el conjunto C, el método *single* es el que logra una mejor clasificación.
- Para el conjunto D, los métodos que realizan bien la clasificación son el *single* y el *ward*, esto puede ser debido a la forma de los datos y a que no hay ruido entre grupos.

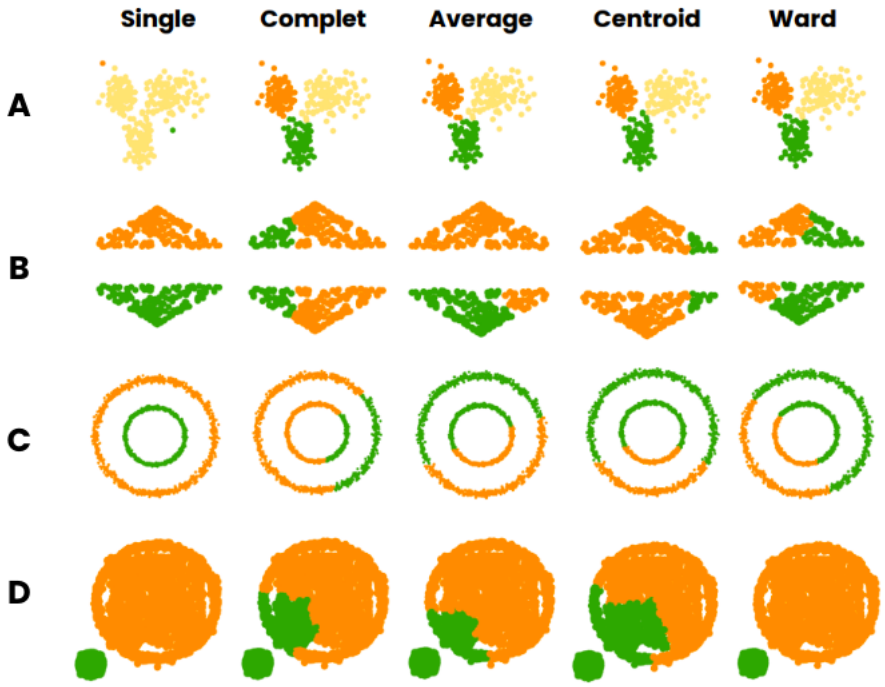


Figura 3.2: Ilustración de los distintos conglomerados obtenidos después de aplicar los métodos de enlace utilizando la distancia euclidiana para cuatro conjuntos de datos simulados. El conjunto A está compuesto de 3 nubes de puntos con ruido entre ellos. El conjunto B está conformado por dos triángulos, uno de ellos es un reflejo del otro y están dispuestos con separación entre sí. El conjunto C contiene dos circunferencias, una adentro de otra y separadas entre sí. El conjunto D exhibe dos círculos rellenos de puntos, colocados con cierta separación. Imagen de autoría propia.

Como se ilustró a través de este ejercicio, cada método de enlace tiene sus ventajas y desventajas, y dependiendo del caso de uso variará el método que se deba implementar. Es importante recordar que en la práctica no es común ver conjuntos de datos con tan sólo dos dimensiones, ni en donde los grupos se puedan diferenciar tan fácilmente a simple vista, así que se procede a explorar los diferentes métodos de enlaces y analizar los resultados.

3.1.3. Elección del número de conglomerados

Usualmente en las aplicaciones de la vida real el número de grupos k es desconocido, por lo que determinarlo es un problema fundamental en el análisis de conglomerados. Desafortunadamente no hay una respuesta definitiva para esta interrogante, el número óptimo de grupos resulta ser subjetivo y depende del contexto. En muchas aplicaciones lo importante es que los grupos sean interpretables y útiles.

Para el método jerárquico la solución más simple consiste en inspeccionar el dendrograma resultante para ver si sugiere un número particular de grupos, sin embargo, esto también es arbitrario ya que muchas veces depende de quien lo mire. Por lo tanto, para resolver esta cuestión se han desarrollado distintos métodos, entre los más populares se encuentran el del codo, la silueta y el estadístico Gap. Según Kassambara (2017), adicionalmente a estos hay más de treinta métodos que han sido publicados. En esta sección se explicarán los tres antes mencionados.

Método del codo

Sea $\{X_{ij}\}$, $i = 1,2,...,n,\ \ j = 1,2,...,p$, un conjunto de datos con n observaciones y p características, agrupados en k conglomerados $C_1, C_2, ..., C_k$, donde C_r denota el conglomerado r y c_r a su centroide. Denotemos por $d(i, c_r)$ a la distancia euclidiana entre la observación i y el centroide c_r, entonces:

$$D_r = (\sum_{i \in C_r} d(i, c_r))^2 \tag{3.2}$$

es la suma al cuadrado de las distancias de todos los puntos de C_r a su centroide. Y se define WSS como:

$$WSS = \sum_{r=1}^{k} (D_r), \tag{3.3}$$

es decir, la suma de todas las $Dr's$.

Por lo tanto, WSS [3] mide la compacidad entre los conglomerados y por consiguiente se quiere que esta sea lo más pequeña posible. Lógicamente, mientras más grupos haya, más pequeña será, entonces ¿Cómo elegir hasta qué k detenerse?. El método del codo, desarrollado por Thorndike en 1953, responde a esta interrogante. Consiste en ver a la WSS como función del número de conglomerados k y elegir la k tal que al añadir otro conglomerado WSS no disminuya considerablemente, del tal forma que en ese punto se vea una forma similar a la de un codo (Ver Figura 3.3).

[3] *Within-clúster sum of squares*, por sus siglas en inglés

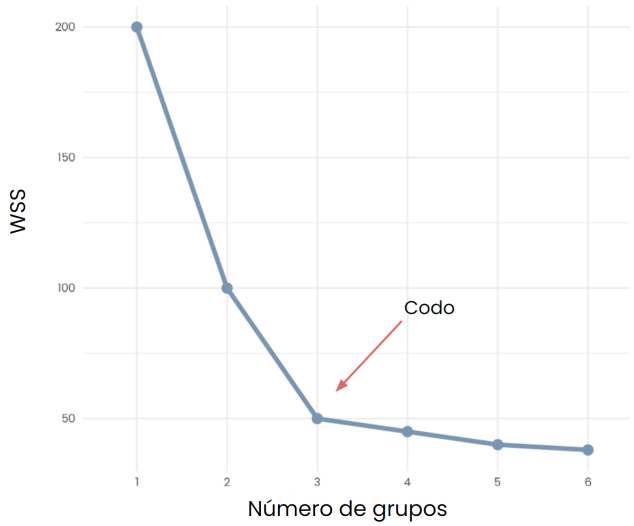


Figura 3.3: Ejemplo del método del codo, donde se determina que el número de conglomerados es tres.

Método de la silueta

Este método fue propuesto por Rousseeuw en 1987. Consiste en tener $\{X_{ij}\}$, $i = 1,2,...,n,\ j = 1,2,...,p$, un conjunto de datos con p características y n observaciones, escalados y agrupados en k conglomerados $C_1, C_2, ..., C_k$, donde el conglomerado r se denota con C_r y su cardinalidad $|C_r| = n_r$.
Denotemos por $d(i,i')$ a la distancia euclidiana entre las observaciones i e i', entonces si i es un punto en el conglomerado C_r:

$$a_i = \left(\sum_{i' \in C_r} d(i,i') \right) \frac{1}{n_r} \tag{3.4}$$

es el promedio de las distancias entre i y todos los puntos en el conglomerado r. Y

$$b_i = \text{mín} \left\{ \left(\sum_{j \in C_s} d(i,j) \right) \frac{1}{n_s} \middle| \ C_s \neq C_r \right\} \tag{3.5}$$

es el promedio de las distancias entre i y todos los puntos del conglomerado C_s más cercano a C_r.

Entonces, el ancho de la silueta para el punto i se define como $S_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$, donde $S_i \in [-1,1]$ y se puede interpretar como sigue:

- Si S_i es cercano a 1 significa que la observación i está bien agrupada.
- Si S_i es cercano a 0 significa que la observación i se encuentra entre dos conglomerados.
- Si S_i resulta negativo significa que la observación i probablemente está mal agrupada.

Por lo que S_i mide que tan bien el objeto i coincide con su conglomerado asignado y se quiere que en promedio el ancho de la silueta de cada conglomerado sea tan amplio como

sea posible. Este se define como el promedio de S_i para todos los objetos i pertenecientes a ese conglomerado. Finalmente, se puede considerar el promedio general del ancho de la silueta como el promedio de los S_i para todos los objetos i en el conjunto de datos. En general, para diferentes valores de k se tendrán distintos promedios generales $\bar{s}(k)$. Una forma de elegir la k adecuada es seleccionando aquella k cuyo valor $\bar{s}(k)$ sea el más grande posible.

Método del estadístico Gap

Supongamos que se tiene un conjunto de datos $\{X_{ij}\}$, $i = 1,2,...,n$, $j = 1,2,...,p$, con n observaciones y p características, agrupados en k conglomerados $C_1, C_2, ..., C_k$, donde C_r denota el conglomerado r y n_r la cardinalidad de cada conglomerado. Sea $d(i,i')$ la distancia euclidiana entre las observaciones i e i', entonces:

$$D_r = \sum_{i,i' \in C_r} d(i,i') \tag{3.6}$$

es la suma de las distancias entre todas las parejas de puntos en el conglomerado r. Con lo cual se obtiene:

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r. \tag{3.7}$$

A grandes rasgos, la idea de este método es estandarizar la gráfica de $log(W_k)$ comparando el cambio de dispersión dentro del conglomerado y el cambio esperado bajo una distribución de referencia. Donde el número óptimo de conglomerados es el valor de k para el cual el $log(W_k)$ caiga lo más alejado posible por debajo de esta curva de referencia. Entonces, la estadística Gap se define como:

$$Gap(k) = E_n^*\{log(W_k)\} - log(W_k), \tag{3.8}$$

donde E_n^* denota el $log(W_k)$ bajo la distribución de referencia. El k elegido será aquel que maximice el valor de $Gap_n(k)$.

Respecto a la distribución de referencia, la manera más simple de crearla es hacerlo con una distribución uniforme, donde el máximo y el mínimo de la distribución estarán dados por el máximo y el mínimo de los puntos observados.[4] $E_n^*\{log(W_k)\}$ se estima realizando B copias de la distribución de referencia y obteniendo el promedio del $log(W_k)$. Y finalmente, se elige la k más pequeña tal que Gap(k) ≥ Gap($k+1$) - s_{k+1}. Este método fue desarrollado por Tibshirani et al. (2001) como una alternativa al método del codo.

Para el experimento práctico de este trabajo, la intención es probar con todas las $k's$ sugeridas por estos métodos, aunque la elección final estará relacionada con el objetivo de construir estratos que reduzcan el ECM de los estimadores.

3.2. Análisis de conglomerados jerárquico con restricciones espaciales usando el método Ward

En algunos problemas de conglomerados es importante aplicar restricciones sobre las posibles soluciones. El tipo más común es la de contigüidad (sobre espacio o tiempo), tales

[4] Otra elección para la distribución de referencia se puede consultar en Tibshirani et al. (2001).

restricciones ocurren cuando los objetos en un conglomerado no sólo necesitan ser similares entre ellos sino también abarcar un conjunto contiguo de objetos.

Para definir qué es un conjunto contiguo, es necesario primero considerar la contigüidad entre cada par de objetos, la cual está dada por una matriz $\mathbf{C} = (c_{ij})_{nxn}$ donde $c_{ij} = 1$ si i y j son contiguos y 0 si no. Así, un conglomerado C es contiguo si existe un camino entre cualquier par de objetos en C.

En Chavent et al.(2018) se desarrolló un algoritmo de conglomerados jerárquico que incluye restricciones espaciales. Este se emplea usando el método de enlace Ward, dos matrices de distancias (no necesariamente euclidianas) D_0 y D_1, un parámetro de mezcla $\alpha \in [0,1]$ y pesos para las observaciones w_i (los cuales pueden no ser uniformes). La primera matriz corresponde a las distancias usando las p características antes descritas y la segunda a las distancias geográficas, por ejemplo la de contigüidad. El parámetro de mezcla α establece la importancia de las restricciones geográficas, cuando α incrementa, se le da más importancia a las distancias en D_1 y viceversa. La idea es determinar un valor de α el cual incremente la contigüidad espacial sin deteriorar demasiado la calidad de la solución con base en las otras variables de interés.

Sea $\{X_{il}\}$ un conjunto de n observaciones y p características ($l = 1,...,p$), donde w_i es el peso de la i-ésima observación para $i = 1,2,...,n$. Consideremos dos matrices de distancias: $D_0 = [d_{0,ij}]$ y $D_1 = [d_{1,ij}]$, $i, j = 1,...,n$. Para un valor dado de α el algoritmo funciona como sigue, considerando que la partición en k grupos será indexada por: $\mathscr{P}_k^{\alpha}$.

Definición 3.2.1 *La pseudoinercia mezclada del conglomerado C_k^{α} (llamada inercia mezclada de aquí en adelante) se define como:*

$$I_{\alpha}(C_k^{\alpha}) = (1-\alpha) \sum_{i \in C_k^{\alpha}} \sum_{j \in C_k^{\alpha}} \frac{w_i w_j}{2\mathscr{M}_k^{\alpha}} d_{0,ij}^2 + \alpha \sum_{i \in C_k^{\alpha}} \sum_{j \in C_k^{\alpha}} \frac{w_i w_j}{2\mathscr{M}_k^{\alpha}} d_{1,ij}^2. \tag{3.9}$$

En donde $\mathscr{M}_k^{\alpha} = \sum_{i \in C_k^{\alpha}} w_i$, es la suma de los pesos de todos los elementos en el conglomerado C_k^{α} y $d_{0,ij}$ ($d_{1,ij}$) es la distancia normalizada entre las observaciones i y j en D_0 (D_1).

Entonces, cuando los pesos son uniformes ($w_i = \frac{1}{n}$), la pseudoincercia queda definida como:

$$I_{\alpha}(C_k^{\alpha}) = (1-\alpha) \sum_{i \in C_k^{\alpha}} \sum_{j \in C_k^{\alpha}} \frac{1}{2n|C_k^{\alpha}|} d_{0,ij}^2 + \alpha \sum_{i \in C_k^{\alpha}} \sum_{j \in C_k^{\alpha}} \frac{1}{2n|C_k^{\alpha}|} d_{1,ij}^2. \tag{3.10}$$

Ya que:

$$\mathscr{M}_k^{\alpha} = \sum_{i \in C_k^{\alpha}} W_i = \frac{|C_k|}{n}, \tag{3.11}$$

y por lo tanto:

$$\frac{w_i w_j}{2\mathscr{M}_k^{\alpha}} = \frac{\frac{1}{n^2}}{\frac{2|C_k|}{n}} = \frac{n}{2n^2|C_k|} = \frac{1}{2n|C_k|}. \tag{3.12}$$

Definición 3.2.2 *La pseudoinercia mezclada de la partición* $\mathscr{P}_k^\alpha = (C_1^\alpha, ..., C_k^\alpha)$ *es la suma de la inercia mezclada de sus conglomerados:*

$$W_\alpha(\mathscr{P}_k^\alpha) = \sum_{k=1}^{K} I_\alpha(C_k^\alpha). \tag{3.13}$$

Para obtener una nueva partición $\mathscr{P}_k^\alpha$ con k grupos a partir de la partición $\mathscr{P}_{k+1}^\alpha$, la idea es agregar dos conglomerados A y B de $\mathscr{P}_{k+1}^\alpha$ tal que la partición resultante tenga la mínima inercia mezclada. Así, la optimización del problema puede ser expresada como sigue:

$$\min_{A,B \in \mathscr{P}_{k+1}^\alpha} I_\alpha(A \cup B) - I_\alpha(A) - I_\alpha(B). \tag{3.14}$$

3.2.1. Algoritmo

- Paso $k = n$: inicialización.
  Las distancias se pueden reescalar entre 0 y 1 para que tengan el mismo orden de magnitud.
  La partición inicial $\mathscr{P}_n^\alpha =: \mathscr{P}_n$ en n conglomerados (es decir, donde cada conglomerado contiene una observación) es única y por lo tanto no depende de α.

- Paso $k = n-1, ..., 2$: se obtiene la partición en k conglomerados a partir de la partición de $k+1$ conglomerados.
  En cada paso k, el algoritmo agrega dos conglomerados A y B de $\mathscr{P}_{k+1}^\alpha$ de acuerdo con el problema de optimización en 3.14 de tal forma que, de todas las uniones posibles, la unión seleccionada tenga la inercia mezclada más pequeña.

  Más precisamente el algoritmo agrega dos conglomerados A y B tal que la medida de agregación correspondiente

  $$\delta_\alpha(A, B) := W_\alpha(\mathscr{P}_{k+1}\alpha) - W_\alpha(\mathscr{P}_k\alpha) = I_\alpha(A \cup B) - I_\alpha(A) - I_\alpha(B) \tag{3.15}$$

  sea mínima.

- Paso $k = 1$: detener. Se obtiene la partición en un conglomerado $\mathscr{P}_1^\alpha =: \mathscr{P}_1$. Nótese que dicha partición es única y no depende de α.

Observaciones.

- El algoritmo previo difiere de aplicar directamente el método de Ward a la matriz obtenida a través de la combinación $D_\alpha = (1-\alpha)D_0 + \alpha D_1$.
  El beneficio principal del algoritmo propuesto por Chavent et al. (2018) es que el parámetro de mezcla α controla la parte de la inercia entre D_0 y D_1 en la ecuación 3.2.2. Este no es el caso cuando se aplica directamente el método de Ward a D_α ya que este está basado en una única inercia.

- Cuando $\alpha = 0$ ($\alpha = 1$) el algoritmo jerárquico sólo está basado en la matriz de distancias D_0 (D_1). Un procedimiento propuesto para determinar el valor apropiado para α se puede encontrar en *ClustGeo: an R package for hierarchical clústering with spatial constraints* de Chavent et al. (2018). Sin embargo, en este trabajo se aprovechará el poder computacional y se probará con una malla para distintos valores de α.

Capítulo 4

Clasificación de los municipios de los Estados Unidos Mexicanos a través del método de conglomeración jerárquica con restricciones espaciales

En este capítulo se presenta el resultado de aplicar el método de conglomerados jerárquico con restricciones espaciales para crear una estratificación de los municipios de los Estados Unidos Mexicanos que tomó en cuenta tanto características sociodemográficas como geográficas. Esto con la motivación de crear estratos de muestreo a nivel nacional como los que el INEGI obtiene en la primera etapa de su estratificación para el diseño de muestreo de su Muestra Maestra, es decir, cuando se crean los cuatro estratos sociodemográficos: bajo, medio bajo, medio alto y alto. Se trabajó con la información de los municipios existentes hasta el 2020 según el Censo de Población y Vivienda realizado por el INEGI ese mismo año.

Como se detalló en la sección anterior, para poder utilizar el algoritmo de conglomerados jerárquico con restricciones espaciales son necesarias dos matrices: una que posea las distancias entre las características sociodemográficas y otra que incluya las distancias geográficas. Primero se explicará cómo se seleccionaron y procesaron los datos para las características sociodemográficas.

4.0.1. Características sociodemográficas

Para las características sociodemográficas fueron utilizadas las siguientes bases de datos:

- **Indicadores de Pobreza Municipal 2015**[1]. Esta base fue creada por el CONEVAL y es la segunda edición, teniendo como predecesor los Indicadores de Pobreza Municipal 2010. Los indicadores sirven para dar cuenta de la situación de pobreza que existe en el país, los cuales son una herramienta útil par el diseño y la evaluación de

[1] Disponible en: https://www.coneval.org.mx/Medicion/Paginas/PobrezaInicio.aspx

políticas públicas destinadas a la superación de la pobreza acorde con las características de cada región (CONEVAL, 2018, pag 8). La base de datos está constituida por 37 variables y 2,457 municipios.

- **Principales Resultados por Localidad del Censo de Población y Vivienda 2020**[2]. Estos resultados surgen del Censo de Población y Vivienda, realizado por el INEGI, el cual es el proyecto estadístico de mayor relevancia en el país, pues es la fuente con mayor nivel de desagregación geográfica que proporciona información sobre la dimensión, estructura y distribución en el territorio nacional de la población, y permite cuantificar las viviendas y sus características (INEGI, 2021 a, pag 11). La base de datos contiene lo resultados del Censo a nivel localidad y municipal y consta de 232 variables y 2,469 municipios.

La base de datos de los Principales Resultados por Localidad, como su nombre lo dice, se encuentra desglosada a nivel localidad, pero también contiene información a nivel municipal, así que esta base se filtró para conservar únicamente la información de los 2,469 municipios.

Para unir las dos bases se utilizó el nombre de cada estado y municipio como llave primaria, por lo cual fue necesario limpiar estos nombres.

Limpieza de datos

En ambas bases de datos se limpiaron los nombres de los estados y municipios, ya que en muchos casos no coincidían y, sin embargo, se trataba de los mismos, por ejemplo: *Distrito Federal* y *Ciudad de México*. Esta limpieza incluyó la sustitución de abreviaciones en los nombres, por ejemplo: *Gral. Bravo* por *General Bravo*.

Fueron limpiados cuatro nombres de entidades federativas y 31 nombres de municipios para que las bases pudieran ser unidas.

Imputación de datos faltantes

Como se mencionó anteriormente, se trabajó con la información de los municipios creados hasta el 2020 según el Censo de Población y Vivienda de dicho año, sin embargo, de 2015 a 2020 se crearon 12 municipios, por lo que estos no aparecen en los Indicadores de Pobreza Municipal 2015, pero sí en el Censo 2020 del INEGI, mismos que se pueden consultar en el Tabla 4.1. Dado que estos municipios fueron creados a partir de la partición y fusión de otros, se decidió imputar la información en los Indicadores de Pobreza realizando un promedio de los valores de aquellos municipios de los que provienen. [3]

Así mismo existen 11 municipios en la base de datos de los Indicadores de Pobreza que no contienen información (ver Tabla 4.2). Para imputarla, se promedió la información de los municipios circundantes.

[2]Disponible en: https://www.inegi.org.mx/programas/ccpv/2020/

[3]Esta información se obtuvo consultando https://www.inegi.org.mx/app/ageeml/

Procesamiento de variables

Puesto que la mayoría de los datos se muestran en totales absolutos, es difícil hacer comparaciones entre los municipios y por consiguiente se eligió convertir los datos a proporciones. Para la creación de las proporciones en los Indicadores de Pobreza se dividió la información de cada variable entre la población total del respectivo municipio. Sin embargo, en los datos del Censo no todas las variables tienen la misma población de referencia, por lo que cada una se dividió entre el total que hiciera sentido. Por ejemplo, la *población de 12 años o más* se dividió entre la *población total*, mientras que el *total de viviendas habitadas* se dividió entre el *total de viviendas*. En el caso de las variables asociadas a promedios como el *grado promedio de escolaridad*, las variables se normalizaron restándoles el valor mínimo y dividiéndolas entre el rango para que así todos los datos de las variables estuvieran escalados entre 0 y 1.

Municipios creados de 2015 a 2020

Estado	Municipio	Estado de origen	Municipio(s) de origen
Baja California	San Quintín	Baja California	Ensenada
Campeche	Seybaplaya	Campeche	Champotón Campeche
Chiapas	Capitán Luis Ángel Vidal	Chiapas	Siltepec
Chiapas	Rincón Chamula San Pedro	Chiapas	Jitotol Rayón Tapilula Pueblo Nuevo - Solistahuacán
Chiapas	El Parral	Chiapas	Villaflores Villa Corzo
Chiapas	Emiliano Zapata	Chiapas	Venustiano Carranza Chiapa de Corzo Acala
Chiapas	Mezcalapa	Chiapas	Ocozocoautla de Espinosa Tecpatán Ostuacán
Chiapas	Honduras de la Sierra	Chiapas	Siltepec
Morelos	Coatetelco	Morelos	Miacatlán
Morelos	Xoxocotla	Morelos	Puente de Ixtla
Morelos	Hueyapan	Morelos	Tetela del Volcán
Quintana Roo	Puerto Morelos	Quintana Roo	Benito Juárez

Tabla 4.1: Municipios nuevos que aparecen en la información del Censo, pero no en los Indicadores de pobreza municipal.

Municipios sin información

Estado	Municipio	Estado	Municipios vecinos
Chihuahua	Buenaventura	Chihuahua	Ascensión Nuevo Casas Grandes Galeana Ignacio Zaragoza Namiquipa Chihuahua Ahumada
Chihuahua	Carichí	Chihuahua	Guerrero Bocoyna Guachochi Nonoava San Francisco de Borja Cusihuiriachi
Chihuahua	Santa Isabel	Chihuahua	Chihuahua Riva Palacio Cuauhtémoc Gran Morelos Dr. Belisario Domínguez Satevó
Chihuahua	Temósachic	Chihuahua	Madera Moris Ocampo Guerrero Matachí Namiquipa Gómez Farías
		Sonora	Sahuaripa Yécora
Chihuahua	Urique	Chihuahua	Guazapares Batopilas de Manuel - Gómez Morín Guachochi Bocoyna Maguarichi Choix

Estado	Municipio	Estado	Municipios vecinos
Oaxaca	Matías Romero Avendaño	Veracruz	Jesús Carranza
		Oaxaca	San Juan Cotzocón San Juan Mazatlán San Juan Guichicovi Santa María Petapa El Barrio de la Soledad Santa María Chimalapa
Oaxaca	San Francisco Chindúa	Oaxaca	San Pedro Topiltepec Santiago Nejapilla Santo Domingo - Tlatayápam San Francisco Nuxaño San Miguel Tecomatlán San Mateo Etlatongo San Juan Sayultepec
Oaxaca	Santa María Chimalapa	Oaxaca	Matías Romero Avendaño El Barrio de la Soledad Asunción Ixtaltepec San Miguel Chimalapa
		Chiapas	Cintalapa
		Veracruz	Las Choapas Uxpanapa Jesús Carranza
Oaxaca	Santa María Petapa	Oaxaca	Matías Romero Avendaño San Juan Guichicovi Santo Domingo Petapa El Barrio de la Soledad
Puebla	San Nicolás de los Ranchos	Estado de México	Amecameca Atlautla
		Puebla	Tochimilco Tianguismanalco Nealtican Calpan Huejotzingo
Sonora	Plutarco Elías Calles	Sonora	Puerto Peñasco Caborca

Tabla 4.2: Municipios en los Indicadores de pobreza municipal que no contienen información.

Elección de las variables sociodemográficas

Para la elección de las variables sociodemográficas fueron tomados en cuenta los indicadores empleados en la estratificación de la Muestra Maestra 2010 ver INEGI (2019, pag 57) los cuales fueron los siguientes:

Mnemónico	Descripción
	Proporción de Población
PPSSNOSP	Que tiene derecho a recibir servicios médicos en alguna institución de salud pública o privada excepto seguro popular.
PPDER_SS	Derechohabiente a servicios de salud.
PDP3A14A	De 3 a 14 años de edad que asiste a la escuela.
PDP15A24A	De 15 a 24 años de edad que asiste a la escuela.
PDP8A14ALF	De 8 a 14 años de edad que saben leer o escribir.
PDP15YM_SE	De 15 años o más de edad que aprobaron algún grado de escolaridad diferente al nivel preescolar.
PP15PRI_CO	De 15 años o más de edad que tienen como máxima escolaridad 6 grados aprobados en primaria.
PP15SEC_CO	De 15 años o más de edad que tienen como máxima escolaridad 3 grados aprobados en secundaria.
PGDO_ESC	Grado promedio de escolaridad.
PPEA	De 12 años y más que trabajaron; tenían trabajo pero no trabajaron o; buscaron trabajo en la semana de referencia.
PPEA_F	Femenina de 12 años y más que trabajaron; tenían trabajo pero no trabajaron o; buscaron trabajo en la semana de referencia.
PTASAOCUPA	Tasa de ocupación.
TOCU12A17	No ocupada de 12 a 17 años de edad entre la población de 12 a 17 años de edad.
PPOMAYED	Ocupada de 18 y más años de edad entre la población ocupada.
	Proporción de Viviendas Particulares Habitadas
PVIVSINH	Que no tienen hacinamiento.
PVPH_PISDT	Que tienen piso de cemento o firme, madera, mosaico u otro material.
PVPH2YMASD	Que usan para dormir entre 2 y 25 cuartos.
PVPH_2MASC	Que tienen más de un cuarto.
PVPH3YMASC	Que tienen entre 3 y 25 cuartos.
PVPH_C_ELE	Que disponen de luz eléctrica.
PVPHAGUADV	Que tienen disponibilidad de agua entubada dentro de la vivienda,
PVPHAGUADV	o fuera de la vivienda pero dentro del terreno.
PVPH_EXCSA	Que tienen excusado, retrete, sanitario, letrina u hoyo negro.
PVPHDRENAJ	Que tienen drenaje conectado a la red pública, fosa séptica, barranca, grieta, rio, lago o mar.
PVDRERED	Que disponen de drenaje conectado a la red pública.
PVEXCAGU	Que disponen de excusado con descarga directa de agua.
PVPH_CSERV	Que disponen de luz eléctrica, agua entubada dentro o fuera de la vivienda, pero dentro del terreno, así como drenaje.
PSIN_HASIN	Que no se encuentran en situación de hacinamiento a nivel manzana.
	Proporción de Viviendas Particulares Habitadas que disponen de:
PVPH_TV	Televisor.
PVPH_AUTOM	Automóvil o camioneta.
PVPH_CEL	Teléfono celular.
PVCELFIJ	Teléfono celular y teléfono fijo.
PV4ELEC	Radio, televisor, refrigerador y lavadora.
PVRADTEL	Radio y televisor.
PVPHCBIEN	Todos los bienes.

Tabla 4.3: Indicadores utilizados en la Muestra Maestra 2010.

Sin embargo, para fines de esta tesis sólo se incluyeron algunos, ya que se le otorgó importancia a también representar diversos grupos vulnerables en la población, por ejemplo: población en situación de pobreza, población vulnerable, población de la tercera edad, población que habla alguna lengua indígena, población afromexicana, población con discapacidad o limitación, y población con rezago educativo. Así como también fueron tomadas en cuenta variables de interés popular como: viviendas con acceso a internet o viviendas con teléfono celular. Las variables elegidas, 45 en total, se muestran en la Tabla 4.4, donde también se presentan algunas estadísticas considerando el procesamiento antes descrito.

Nombre de Variable	**Descripción**	**Media**	**Desviación Estándar**	**Rango**
pobreza_pob	Población en situación de pobreza	0.66	0.21	(0.03 - 1)
pobreza_e_pob	Población en situación de pobreza extrema	0.2	0.18	(0 - 0.97)
pobreza_m_pob	Población en situación de pobreza moderada	0.46	0.12	(0.02 - 0.82)
vul_ing_pob	Población vulnerable por ingreso	0.04	0.04	(0 - 0.24)
npnv_pob	Población no pobre y no vulnerable	0.08	0.1	(0 - 0.66)
ic_segsoc_pob	Población con carencia por acceso a la seguridad social	0.74	0.16	(0.06 - 0.97)
ic_ali_pob	Población con carencia por acceso a la alimentación	0.24	0.12	(0 - 0.86)
carencias_pob	Población con al menos una carencia social	0.88	0.13	(0.32 - 1)
carencias3_pob	Población con al menos tres carencias sociales	0.36	0.21	(0.01 - 0.98)
plb_pob	Población con ingreso inferior a la línea del bienestar	0.69	0.19	(0.04 - 1)
plbm_pob	Población con ingreso inferior a la línea del bienestar mínimo	0.37	0.23	(0.01 - 0.99)
POB65_MAS	Población de 65 años o más	0.1	0.04	(0.01 - 0.32)
PNACOE	Población nacida en otra entidad	0.09	0.09	(0 - 0.7)
P3YM_HLI	Población de 3 años y más que habla alguna lengua indígena	0.17	0.27	(0 - 0.94)
POB_AFRO	Población que se considera afromexicana o afrodescendiente	0.02	0.07	(0 - 0.96)
PCON_DISC	Población con discapacidad	0.06	0.03	(0 - 0.32)
PCON_LIMI	Población con limitación	0.13	0.04	(0.01 - 0.37)
PCLIM_PMEN	Población con algún problema o condición mental	0.01	0	(0 - 0.05)
P15YM_AN	Población de 15 años y más analfabeta	0.07	0.05	(0 - 0.35)
P15YM_SE	Población de 15 años y más sin escolaridad	0.07	0.04	(0 - 0.28)
P15PRI_IN	Población de 15 años y más con primaria incompleta	0.11	0.04	(0 - 0.28)
P15SEC_IN	Población de 15 años y más con secundaria incompleta	0.02	0.01	(0 - 0.09)
P18YM_PB	Población de 18 años y más con educación posbásica	0.19	0.09	(0.01 - 0.76)
GRAPROES	Grado promedio de escolaridad	0.4	0.13	(0 - 1)
PEA	Población de 12 años y más económicamente activa	0.43	0.08	(0.05 - 0.64)
POCUPADA	Población de 12 años y más ocupada	0.42	0.08	(0.05 - 0.62)
PSINDER	Población sin afiliación a servicios de salud	0.24	0.11	(0.01 - 0.84)
PAFIL_IPRIV	Población afiliada a servicios de salud en una institución privada	0.01	0.02	(0 - 0.58)

HOGJEF_F	Hogares censales con persona de referencia mujer	0.29	0.06	(0.09 - 0.5)
PRO_OCUP_C	Promedio de ocupantes por cuarto en viviendas particulares habitadas	0.33	0.14	(0 - 1)
VPH_PISOTI	Viviendas particulares habitadas con piso de tierra	0.08	0.09	(0 - 0.67)
VPH_1CUART	Viviendas particulares habitadas con sólo un cuarto	0.08	0.06	(0 - 0.49)
VPH_S_ELEC	Viviendas particulares habitadas que no disponen de energía eléctrica	0.02	0.03	(0 - 0.49)
VPH_AGUAFV	Viviendas particulares habitadas que no disponen de agua entubada en el ámbito de la vivienda	0.06	0.09	(0 - 0.79)
VPH_NODREN	Viviendas particulares habitadas que no disponen de drenaje	0.14	0.19	(0 - 0.99)
VPH_NDEAED	Viviendas particulares habitadas que no disponen de energía eléctrica, agua entubada, ni drenaje	0.01	0.01	(0- 0.28)
VPH_NDACMM	Viviendas particulares habitadas que no disponen de automóvil o camioneta, ni de motocicleta o motoneta	0.59	0.2	(0.04 - 0.98)
VPH_SNBIEN	Viviendas particulares habitadas sin ningún bien	0.05	0.08	(0 - 0.55)
VPH_REFRI	Viviendas particulares habitadas que disponen de refrigerador	0.74	0.21	(0.04 - 0.99)
VPH_LAVAD	Viviendas particulares habitadas que disponen de lavadora	0.56	0.24	(0 - 0.95)
VPH_TV	Viviendas particulares habitadas que disponen de televisor	0.81	0.15	(0.08 - 0.98)
VPH_PC	Viviendas particulares habitadas que disponen de computadora, laptop o tablet	0.18	0.13	(0 - 0.85)
VPH_CEL	Viviendas particulares habitadas que disponen de teléfono celular	0.73	0.18	(0.02 - 0.97)
VPH_INTER	Viviendas particulares habitadas que disponen de Internet	0.27	0.18	(0 - 0.92)
VPH_SINCINT	Viviendas particulares habitadas sin computadora ni Internet	0.68	0.18	(0.03 - 1)

Tabla 4.4: Resumen estadístico de las variables seleccionadas para usarse como características sociodemográficas.

Breve análisis exploratorio de las variables sociodemográficas seleccionadas

Para tener una comprensión general de la base de datos se realizaron cuatro mapas a través del software *qgis* y se colorearon los municipios de acuerdo a cuatro variables: Población en situación de pobreza, población de 3 años y más que habla alguna lengua indígena, grado promedio de escolaridad y población sin afiliación a servicios de salud. Esto se hizo utilizando la clasificación por quintiles, con las proporciones obtenidas anteriormente, excepto los años promedio de escolaridad ya que esta variable es más fácil de interpretar cuando no está escalada. Además, se dan algunos comentarios sobre las variables relacionadas con las antes mencionadas.

En la Figura 4.1 se puede observar que los estados en donde la población tiene un promedio mayor de años de escolaridad son la Ciudad de México, Baja California Sur, Baja California, Tlaxcala y Quintana Roo, mientras que los que tienen menos promedio de años de escolaridad son Chiapas, Oaxaca, Guerrero, Michoacán y Puebla.

Los tres municipios en los que la población tiene, en promedio, más años de escolaridad son: Benito Juárez en Ciudad de México, San Pedro Garza García en Nuevo León y Miguel Hidalgo en Ciudad de México, con 14.55, 13.16 y 13.11 años respectivamente. En contraste los municipios en los que la población, en promedio, tiene menos años de escolaridad son: Cochoapa el Grande en Guerrero, San Martín Peras en Oaxaca y Batopilas de Manuel Gómez Morín en Chihuahua con 3.4, 3.48 y 3.59 años respectivamente.

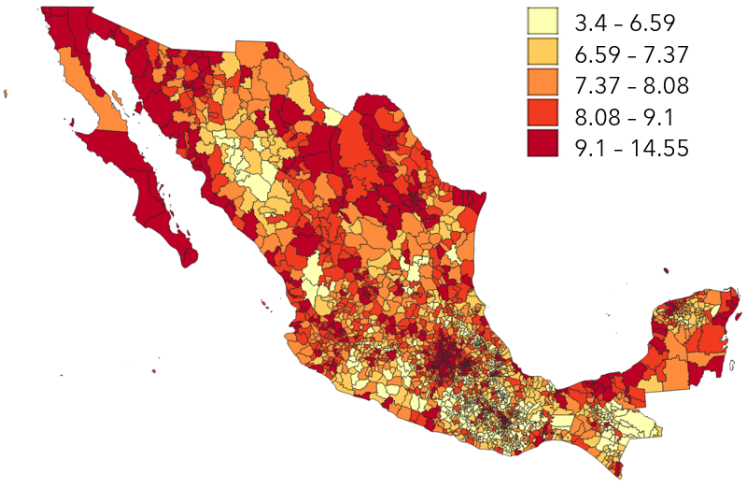


Figura 4.1: Años promedio de escolaridad.

Por su parte Santa María la Asunción en Oaxaca es el municipio del país con mayor proporción (34.61%) de población con 15 años y más analfabeta. Seguido por San Miguel Santa Flor (32.42%) también en Oaxaca y Cochoapa el Grande (28.47%) en Guerrero.

Respecto a la población de 15 años y más sin escolaridad San Miguel Santa Flor encabeza la lista, con 28.22% de su población en esta situación, seguido por Santa María la Asunción (27.92%) y San Juan Mixtepec -Dto. 26 - (27.18%), los tres municipios pertenecientes al estado de Oaxaca.

En la Figura 4.2 se puede observar que la población de 3 años y más que habla alguna lengua indígena predomina en el sur este del país, en su mayoría en los estados de Yucatán, Oaxaca y Chiapas. Mientras que en Aguascalientes menos del 0.01% de la población de 3 años y más habla alguna lengua indígena, seguido por Coahuila y Guanajuato, que también presentan una menor prevalencia en el uso de lenguas indígenas.

En 39 municipios (ver Tabla 4.5) de la República Mexicana, más del 90% de la población de 3 años y más habla alguna lengua indígena, 34 de los cuales pertenecen al estado Oaxaca, tres a Chiapas y dos a Puebla.

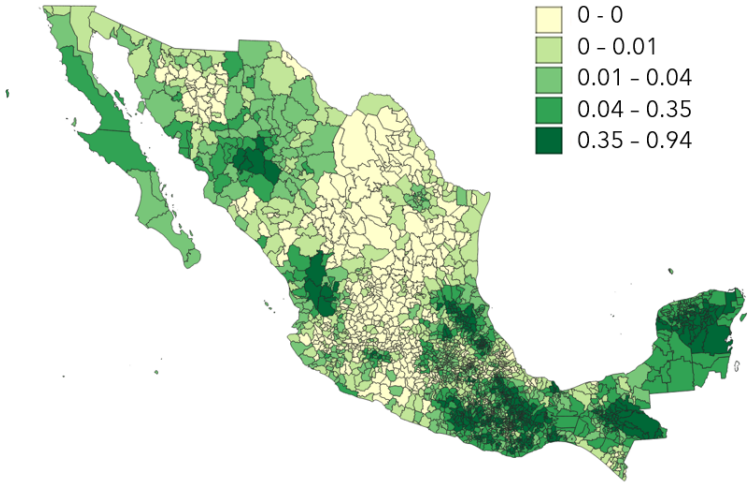


Figura 4.2: Porcentaje de población de 3 años y más que habla alguna lengua indígena.

Municipio	%	Municipio	%
Chiapas		San Pedro Ocotepec	92.22
Aldama	91.17	San Pedro Quiatoni	90.47
Chamula	90.38	San Pedro Yaneri	92.39
Zinacantán	91.43	San Vicente Lachixío	90.52
Oaxaca		Santa Ana Yareni	91.99
Abejones	91.32	Santa Catalina Quierí	94.3
Mixistlán de la Reforma	92.72	Santa Inés Yatzeche	92.95
San Bartolomé Quialana	90.75	Santa Lucía Miahuatlán	91.82
San Francisco Logueche	90.51	Santa María Temaxcalapa	91.92
San José Lachiguiri	92.38	Santa María Texcatitlán	90.07
San Juan Mixtepec -Dto. 26 -	91.27	Santa María Tlahuitoltepec	90.04
San Juan Petlapa	91.85	Santiago Atitlán	91.45
San Juan Yaeé	92.36	Santiago Lalopa	91.65
San Juan Yatzona	93.18	Santiago Texcalcingo	90.22
San Lucas Camotlán	92.06	Santiago Yaitepec	90.22
San Lucas Quiaviní	92.21	Santo Domingo Roayaga	90.01
San Martín Peras	90.33	**Puebla**	
San Mateo del Mar	90.13	Atlequizayan	90.05
San Miguel Aloápam	90.42	Camocuautla	91.59
San Miguel Quetzaltepec	92.89		
San Miguel Yotao	92.48		
San Pablo Tijaltepec	91.28		
San Pedro Mixtepec -Dto. 26 -	90.95		

Tabla 4.5: Municipios en donde más del 90% de la población de 3 años o más habla alguna lengua indígena, ordenados alfabéticamente por entidad federativa.

Respecto a la población que se considera afromexicana o afrodescendiente, ésta predomina en el estado de Oaxaca, en municipios como San Juan Bautista Lo de Soto (con 95.69% de la población considerándose así), Santa María Cortijo (93.72%) y Santiago Tapextla (92.85%).

Por otro lado, el 70.4% de la población en Tizayuca Hidalgo nació en otra entidad, así como 65.68% de Pueblo Viejo Veracruz y 63.34% de Hidalgo en Coahuila.

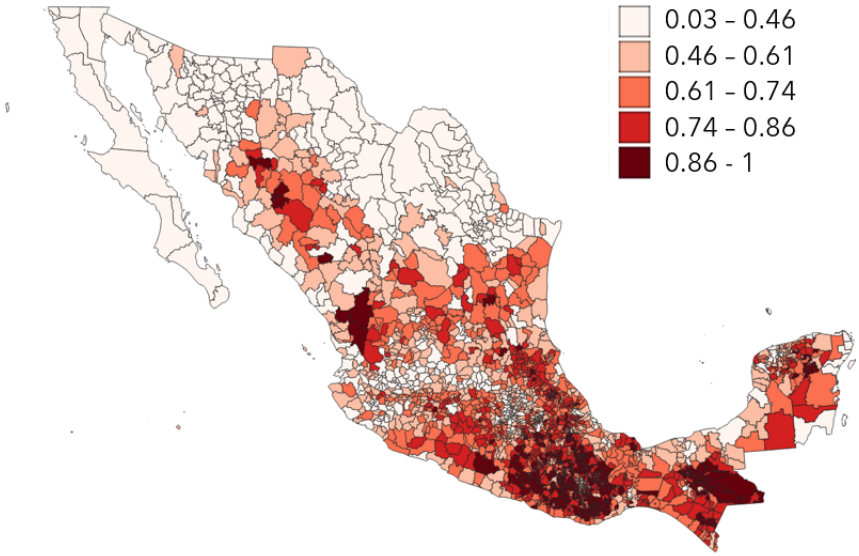


Figura 4.3: Tasa de Pobreza municipal.

La Figura 4.3 muestra la distribución de la pobreza por municipio. Siendo Baja California, Baja California Sur y Ciudad de México los estados menos pobres del país, mientras que un poco más del 50% de los municipios que componen a los estados de Chiapas y Oaxaca tienen un índice de pobreza del 0.86 al 1, es decir, más del 86% de la población en al menos la mitad de los municipios que componen a Chiapas y Oaxaca vive en situación de pobreza.

Aunado a esto, 190 municipios del país, tienen a más del 95% de su población en pobreza extrema, estos pertenecen a los estados de Oaxaca (140), Chiapas (32), Veracruz (7), Guerrero (6) y Puebla (5).

En contraste, los tres municipios con mayor porcentaje de población no pobre y no vulnerable son Benito Juárez con 66.2% de la población, seguido por Miguel Hidalgo (57.7%) y Apodaca (53.1%), los dos primeros pertenecientes a la Ciudad de México y el tercero a Nuevo León.

En la Figura 4.4 se puede observar que los estados con más población sin afiliación a los servicios de salud son: Michoacán, Hidalgo y el Estado de México. Respecto a los municipios, Santa Ana Ateixtlahuaca en Oaxaca lidera esta lista con 83.86% de sus habitantes sin acceso a servicios de salud, seguido por Santa María Ixcatlán (77.87%) y Villa Díaz Ordaz (71.93%), también en Oaxaca. En contraste, los demás municipios del país tienen menos del 66.49% de su población sin afiliación a los servicios de salud.

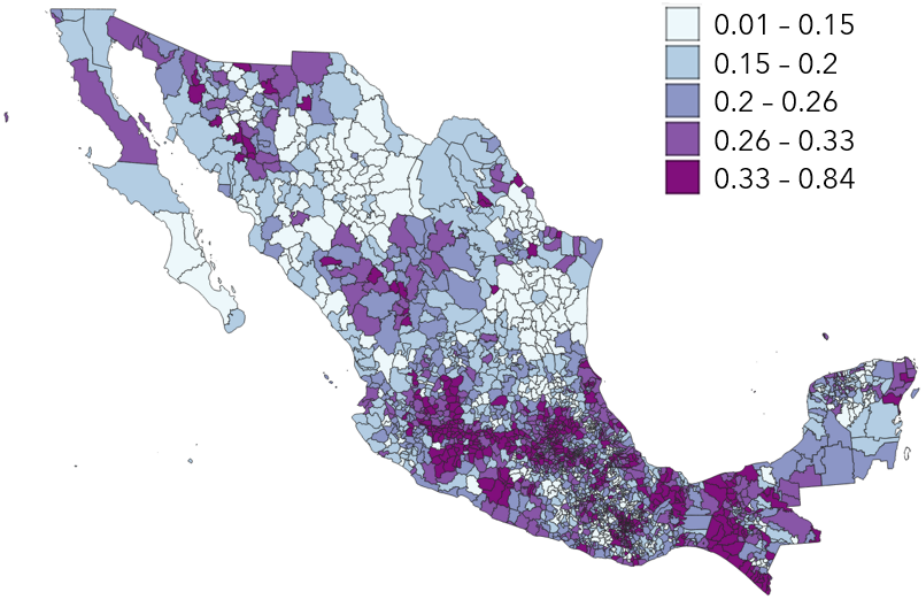


Figura 4.4: Proporción municipal de población sin afiliación a servicios de salud.

Correlación entre las variables sociodemográficas

Por otro lado, se calculó la correlación de Pearson entre pares de variables (ver Figura 4.5). Entre algunos resultados interesantes, se encontró que existe correlación positiva de 0.78 entre las viviendas sin internet y la pobreza poblacional. La variable con la que más se correlaciona la población de 3 años y más que habla alguna lengua indígena es con la población en pobreza extrema (0.71). Y el grado promedio de escolaridad tiene un alto grado de correlación con las viviendas particulares habitadas que disponen de computadora, laptop o tablet (0.89), así como con la población no pobre y no vulnerable (0.79).

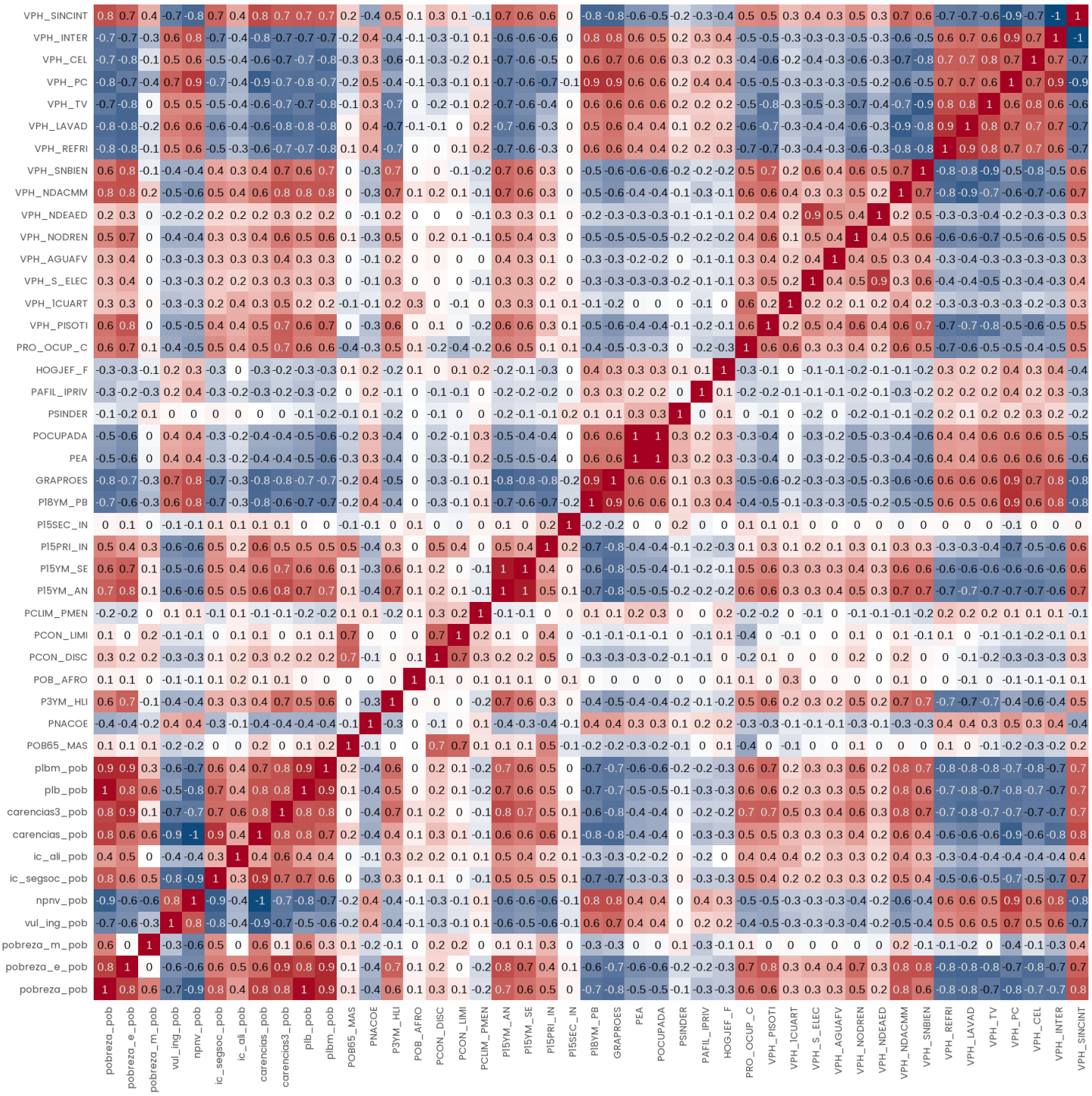


Figura 4.5: Heatmap presentando la correlación de Pearson entre pares de variables socio-demográficas. Se incluye un gradiente rojo a azul, en donde el color blanco significa que la correlación es cero, entre más rojo significa que la correlación es más positiva y entre más azul significa que la correlación es más negativa. Los datos fueron redondeados a un décimal.

4.0.2. Características geográficas

Para la base de datos con las ubicaciones geográficas se utilizó el Marco Geoestadístico[4], desarrollado por el INEGI. El Marco Geoestadístico Nacional es un sistema único y de carácter nacional para referenciar correctamente la información estadística con los lugares geográficos correspondientes; esto se entiende como la delimitación de la República Mexicana en tres niveles de desagregación, llamadas Áreas Geoestadísticas: Estatal (AGEE), Municipal (AGEM) y Básica (AGEB) (INEGI, 2010, pag 2). En este caso fue utilizado a nivel municipal y se compone de 2,469 municipios, los mismos que aparecen en los Principales Resultados por Localidad del Censo de Población y Vivienda 2020.

Específicamente se utilizó el archivo shape que contiene la información de todos los municipios del país hasta el 2020, con 2,469 municipios en total, y a través de la librería `rgdal` (Bivand et al., 2023) estos datos fueron transformados al formato WGS84.

4.1. Matrices de distancias

Una vez que se definieron las características geográficas y sociodemográficas que se usarán, se procedió a calcular las matrices de sus distancias.

4.1.1. Distancias entre las características sociodemográficas

Para el cálculo de las distancias sociodemográficas se decidió usar la distancia euclidiana, ya que todos los datos son continuos y están escalados.

4.1.2. Distancias geográficas

Respecto a las distancias geográficas, se probó con dos formas distintas de determinarlas. La primera consistió en calcular la distancia euclidiana en línea recta entre los centroides de los municipios, ya que otra opción habría sido calcular las distancias entre las carreteras o caminos que conectan a los municipios, pero no se disponía de esa información. La segunda forma de calcular las distancias geográficas se basó en analizar la adyacencia entre municipios.

Matriz de distancias euclidianas: Se utilizó la función `st_centroid` de la librería `sf` (Pebesma, 2018) para obtener los centroides de cada municipio y luego se calculó la distancia euclidiana en metros entre cada par de centroides.

Matriz de distancias binarias: Para cada municipio se construyó una lista de vecinos basándose en las regiones con limites contiguos, es decir, que compartieran al menos un punto límite. Esto se hizo a través de la función `poly2nb` de la librería `spdep` (Bivand, 2022); después se creó una matriz de adyacencia utilizando la función `nb2mat` y, puesto que se consideró a cada municipio vecino de sí mismo, se colocaron unos en la diagonal. Como se quería obtener una matriz de distancias, se creo una matriz de unos y se le restó la matriz de adyacencias. De esta manera se obtuvo una matriz compuesta por ceros y unos. Así, si la entrada (i, j) es igual a 0 significa que los municipios i y j son vecinos, y si la entrada (i, j) es igual a 1 significa que no son vecinos.

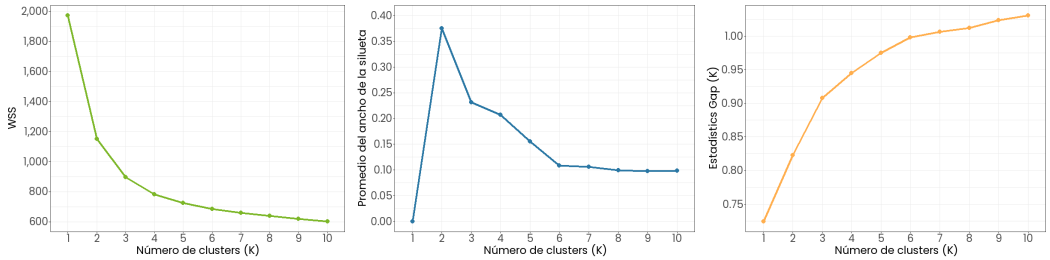
[4]Disponible en: https://www.inegi.org.mx/app/biblioteca/ficha.html?upc=889463807469

4.2. Parámetros para el uso del algoritmo

Para el método de conglomeración jerárquica con restricciones espaciales, además de las matrices de distancias son necesarios otros tres parámetros: los pesos de las observaciones w_i, $i = 1, ..., n$, un parámetro $\alpha \in [0, 1]$ el cual determina la importancia que se le da a las ubicaciones geográficas, y el número de grupos k. Por simplicidad se decidió darle el mismo peso w_i a todas las observaciones así que los parámetros por elegir fueron α y k.

Parámetro α se optó por aprovechar el poder computacional actual y probar con una malla desde $\alpha = 0$, hasta $\alpha = 1$, con brincos de 0.05, es decir, 21 valores en total. Se eligió este número ya que no se vio diferencia utilizando una malla más fina.

Para elegir el número de grupos k primero se graficó el dendrograma (Figura 4.7) en el cual se pueden observar 3, 4 o 6 grupos, sin embargo, con el fin de evitar decisiones subjetivas basadas únicamente en la observación visual del dendrograma se decidió probar con los distintos métodos previamente descritos en el capítulo 3 para tener distintas sugerencias sobre qué k usar. Para esto se utilizó únicamente la base de datos de las características sociodemográficas y la función `fviz_nbclust` de la librería `factoextra` (Kassambara et al., 2020).



(a) Método del codo (b) Método de la silueta (c) Método Gap

Figura 4.6: Resultados de utilizar diferentes métodos para la elección de k.

- **Método del codo**: En la Figura 4.6 (a) se puede observar que la pendiente disminuye considerablemente a partir de $k = 4$.
- **Método Silhouette**: Como se puede ver en la Figura 4.6 (b), el número recomendado de conglomerados por este método puede ser dos o seis.
- **Método estadística de Gap**: Como se observa en 4.6 (c), el número de grupos sugerido para este caso es $k = 10$.

Resumiendo los resultados, el dendrograma mostrado en la Figura 4.7 sugiere 3, 4 o 6 grupos, sin embargo, con el método del codo se sugirieron 4 grupos, con el de la silueta 2 o 6 y con el de gap 10. Como los resultados fueron muy distintos se decidió probar con $k = 3$, 4 y 5. Se eligió usar una k pequeña ya que el INEGI sólo creó cuatro estratos sociodemográficos en su Muestra Maestra de 2012. [5]

[5]Para más detalles consultar: *Cómo se hace la ENOE*, INEGI. (2019).

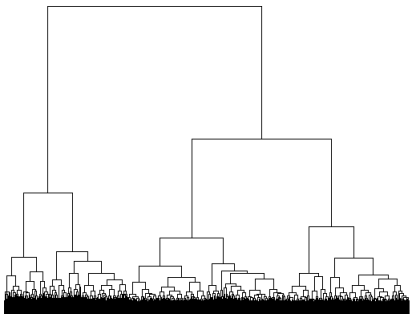


Figura 4.7: Dendrograma obtenido al aplicar el método de conglomerados jerárquico con el método de enlace ward a las características sociodemográficas, el cual sugiere utilizar 3, 4 o 6 grupos.

4.3. Clasificación de los municipios

Para la ejecución del algoritmo de conglomerados jerárquico se utilizó la paquetería `ClustGeo` (Chavent et al., 2021), empleando la función `hclustgeo` para generar el dendrograma y la función `cutree` para elegir el número de grupos. Es importante recordar que la función `hclustgeo` utiliza la distancia euclidiana, con el método de enlace ward para ejecutar el algoritmo de conglomerados jerárquico con restricciones espaciales. Como se tenían 2 matrices con distancias geográficas, 21 opciones de $\alpha's$ y 3 opciones para la k, el algoritmo se ejecutó de 126 formas distintas.

4.3.1. Elección de los parámetros α y k

Si se tuviera acceso a los datos con los que cuenta el INEGI (en particular a los del censo), se podrían obtener muestras a nivel hogar/persona, y se evaluarían las estratificaciones emulando lo que sería el uso que tiene la Muestra Maestra (que es ser usada para conseguir muchas submuestras de viviendas/personas para encuestas sobre varios temas) y así se elegiría la mejor estratificación. Dado que eso no es posible, se decidió usar la información de las Estadísticas Censales a escala geoelectorales [6], ya que es un nivel más desagregado y público que a nivel municipal, de donde se pueden tomar muestras para evaluar las estratificaciones. Para más detalles sobre la creación de esta base consultar la Metodología para el Cálculo de las Estadísticas Censales a Escalas Geoelectorales 2020 (INEGI, 2021 b).

Las Estadísticas Censales a Escala Geoelectorales contienen las mismas variables que los Principales Resultados por Localidad del Censo de Población y Vivienda 2020, pero los datos se desglosan por distritos electorales, en lugar de en localidades. Se decidió utilizar esta base porque es la información con mayor nivel de granularidad disponible y porque es una partición de los resultados a nivel municipal del Censo de Población y Vivienda 2020, en donde cada sección pertenece a un único municipio.

[6] Disponible en: https://www.inegi.org.mx/programas/ccpv/2020/

De las Estadísticas Censales se seleccionaron diez variables y se obtuvieron los estimadores de sus totales utilizando el estimador Horvitz-Thompson en muestreo estratificado con m.a.s al interior de cada estrato, para después calcular la varianza poblacional de cada estimador. Se eligió la α, k y matriz de distancias geográficas de aquella estratificación con la que se obtuvo la menor varianza.

La intención fue que las variables seleccionadas consideraran distintos grupos vulnerables, como por ejemplo: población con rezago educativo (`P15PRI_IN`, `P8A14AN` y `P15YM_AN`), población con discapacidad (`PCON_DISC`), población que habla alguna lengua indígena (`P3YM_HLI`), población sin afiliación a servicios de salud (`PSINDER`) y población en situación de pobreza (`VPH_PISODT`, `VPH_S_ELEC`, `VPH_AGUAFV` y `VPH_NODREN`). Nótese que todas pertenecen al Censo de Población y Vivienda pues la información de los Indicadores de Pobreza sólo tienen como máximo nivel de segmentación el municipal.

La descripción de cada variable y su descripción se enlistan a continuación:

1. `P15PRI_IN`: Población de 15 años o más con primaria incompleta.
2. `P8A14AN`: Población de 8 a 14 años que no sabe leer ni escribir.
3. `P15YM_AN`: Población de 15 años o más analfabeta.
4. `PCON_DISC`: Población con discapacidad.
5. `P3YM_HLI`: Población de 3 años y más que habla alguna lengua indígena.
6. `PSINDER`: Población sin afiliación a servicios de salud.
7. `VPH_PISODT`: Viviendas particulares habitadas con piso de tierra.
8. `VPH_S_ELEC`: Viviendas particulares habitadas que no disponen de energía eléctrica.
9. `VPH_AGUAFV`: Viviendas particulares habitadas que no disponen de agua entubada.
10. `VPH_NODREN`: Viviendas particulares habitadas que no disponen de drenaje.

Para realizar lo anterior lo primero que se hizo fue identificar a qué estratificación pertenecía cada sección geolectoral. Ya que las claves de los municipios en las bases de datos a nivel sección y nivel municipio no coinciden, de nuevo fue necesario utilizar como llave primaria el nombre del estado y municipio, para así poder unir ambas bases y saber a qué agrupación pertenecía cada sección. Se limpiaron los nombres y se logró clasificar a $68,786$ secciones. De esas, se conservaron aquellas con viviendas y población mayor a cero, es decir, $68,710$ secciones.

Lo siguiente fue elegir el tamaño de muestra n, recordando que se está utilizando el estimador HT para totales con un diseño de m.a.s dentro de cada estrato, la n se puede calcular a partir de la ecuación 2.26, es decir:

$$n \geq \frac{\left(\frac{kNS_{yU}^2}{d}\right)^2}{1+\frac{1}{N}\left(\frac{kNS_{yU}^2}{d}\right)^2}, \tag{4.1}$$

con $k = 1.96$ y $N = 68,710$. Donde n corresponde al tamaño de muestra **total** en el experimento, considerando como población a las secciones geoelectorales.

Para obtener d (el error absoluto deseado) se calculó el valor real de cada variable (es decir su suma) y multiplicó por 0.05, de esta forma el error absoluto no será mayor al 5%. Los resultados de cada n sugerida para cada variable se resumen en la Tabla 4.6.

Variable	∑	d	n
P15PRI_IN	7,726,165	386,308	1,822
P8A14AN	412,433	20,622	7250
P15YM_AN	4,451,780	222,589	3,446
PCON_DISC	6,176,499	308,825	1,488
P3YM_HLI	7,354,108	367,705	16,375
PSINDER	32,979,915	1,648,996	2,747
VPH_PISODT	33,818,918	1,690,946	2,324
VPH_S_ELEC	268,564	13,428	13,026
VPH_AGUAFV	1,215,109	60,755	15,573
VPH_NODREN	1,497,889	74,894	10,749

Tabla 4.6: Total real de cada variable, en donde también se muestra el error deseado d y la sugerencia de n para el tamaño de muestra.

Con el fin abarcar todos los resultados se conservó el tamaño de muestra sugerido más grande, es decir, $n = 16,375$ secciones. Este se refiere al tamaño total de la muestra, por lo que se debe calcular el tamaño para cada estrato; para esto se decidió asignar un tamaño de muestra proporcional al tamaño de cada N_h.

Con esto ya se tienen todos los componentes para poder calcular las varianzas poblacionales según la ecuación 2.28:

$$V(\hat{t}_{y\pi}) = \sum_{h=1}^{H} \frac{N_h^2}{n_h}\left(1 - \frac{n_h}{N_h}\right) S_{yU_h}^2,$$

donde $S_{yU_h}^2 = \frac{\sum_{u_k \in U_h}(y_k - \bar{y}_{U_h})^2}{N_h - 1}$ y $\bar{y}_{U_h} = \frac{\sum_{u_k \in U_h} y_k}{N_h}$.

Se obtuvo una base de datos, en donde las columnas fueron los parámetros utilizados, es decir, 10 columnas correspondientes a cada variable seleccionada, y los renglones a la varianza poblacional obtenida con cada estratificación, o sea la combinación de α, k y matriz de distancias geográficas. Así, (i, j) corresponde a la varianza poblacional del parámetro j con la estratificación i.

Para poder comparar las varianzas de cada parámetro, la base se ordenó de menor a mayor de acuerdo a la varianza poblacional obtenida. Lo cual dio como resultado 10 ordenamientos distintos, y a cada estratificación se le asignó un puntaje de acuerdo a la suma de su lugar en los ordenamientos, por ejemplo, si una agrupación estuvo siete veces en tercer lugar, dos en el segundo, y una en el sexto, obtuvo: $(7 * 3) + (2 * 2) + 6 = 31$ puntos. Al final se conservó la estratificación que tuvo menos puntos, es decir, la que en general tiene menos varianza poblacional en todos los estimadores.

En la Figura 4.8 se pueden observar los puntajes obtenidos con cada estratificación por cada variable, de la cual se puede rescatar lo siguiente:

- El puntaje máximo fue de 1,196, que se alcanzó con la estratificación que utilizó distancias binarias (es decir la matriz de adyacencia entre municipios), $\alpha = 1$ y $k = 3$.
- El promedio del puntaje para las estratificaciones fue de 613 puntos.
- El puntaje mínimo fue de 99 puntos, el cual pertenece a la **estratificación elegida, resultante de utilizar k=5, α=0.25 y la base de distancias geográficas calculadas por centroides.**

La $k = 5$ quiere decir que los municipios se clasificaron en 5 grupos, y $\alpha = 0.25$ puede ser interpretado como que se le dio un 25% de peso a las distancias geográficas y el 75% restante a las características sociodemográficas.

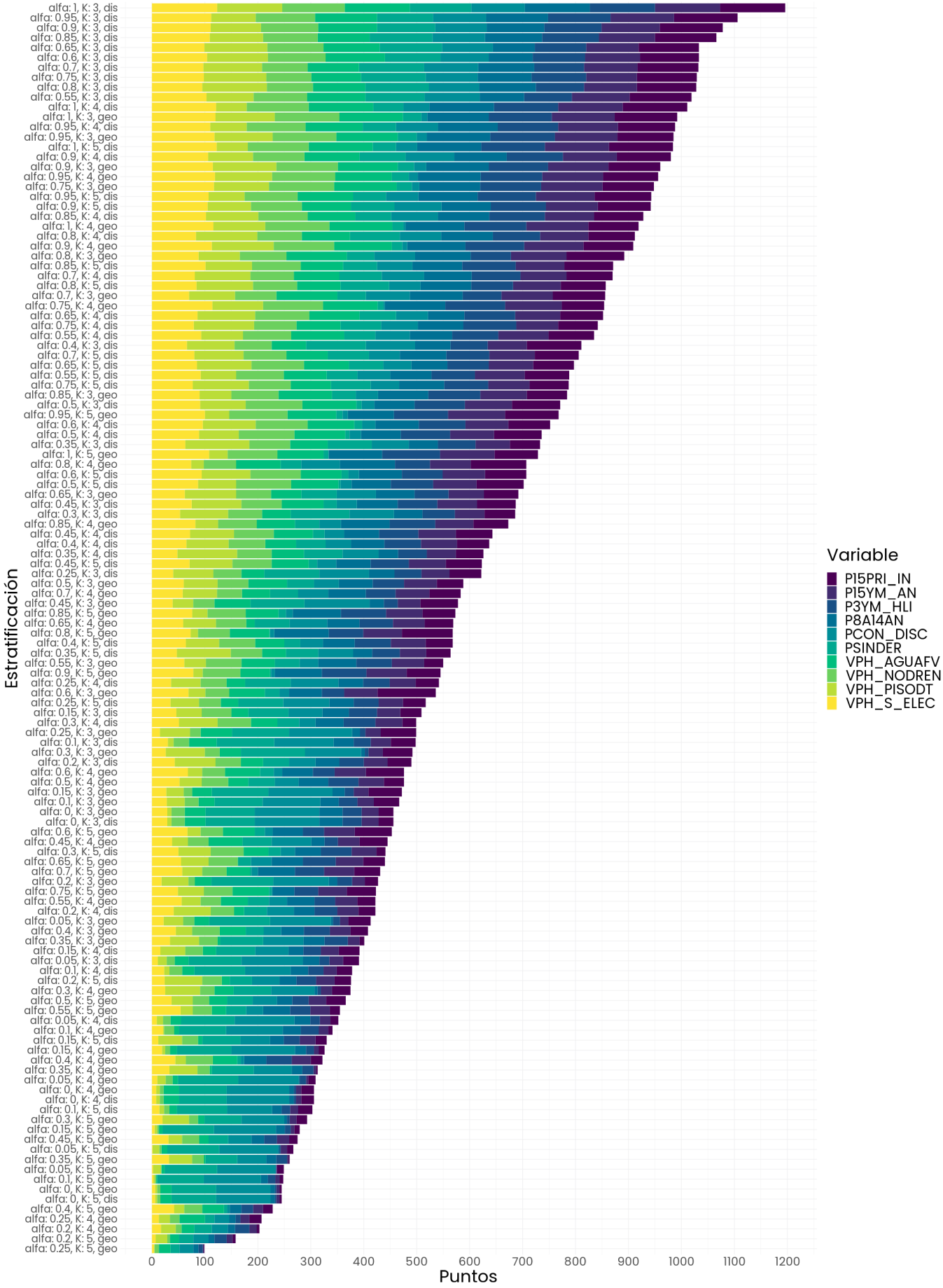


Figura 4.8: Gráfica en donde muestra el puntaje general para cada estratificación. La elegida es la del menor puntaje.

4.4. Evaluación de la estratificación y descripción de los resultados

4.4.1. Evaluación del modelo

Una vez elegida la mejor estratificación, se evaluó si usarla sería mejor que usar muestreo aleatorio simple. Para esto, se calculó la varianza poblacional resultante de tomar una muestra con un m.a.s. Los resultados se muestran en la Tabla 4.7 [7]. En general, los rankings que ocupa la estratificación elegida son buenos, con el lugar 26 como el más bajo para el parámetro total de población con discapacidad, lo cual significa que para ese parámetro habían 25 estratificaciones mejores que la elegida (en el contexto de tener una menor varianza poblacional) de un total de 126. Por otro lado, lo más recalcable es que **las varianzas poblacionales en todos los parámetros fueron menores utilizando la estratificación con muestreo aleatorio simple al interior de cada estrato que realizando un muestreo aleatorio simple**, logrando una disminución de más del 30% en la varianza poblacional de dos parámetros. Esto significa que la estratificación obtenida fue exitosa, ya que se logró disminuir el error cuadrático medio de los estimadores.

Variable	Ranking	Varianza estratificación	Varianza M.A.S	Mejoró	Reducción
P15PRI_IN	1	0.63	0.72	TRUE	-12.51%
P15YM_AN	2	0.31	0.46	TRUE	-31.94%
P3YM_HLI	8	4.68	7.46	TRUE	-37.19%
P8A14AN	10	0.01	0.01	TRUE	-15.62%
PCON_DISC	26	0.37	0.37	TRUE	-1.00%
PSINDER	17	19.58	19.96	TRUE	-1.91%
VPH_AGUAFV	22	0.18	0.19	TRUE	-5.24%
VPH_NODREN	8	0.14	0.18	TRUE	-23.30%
VPH_PISODT	1	17.30	17.64	TRUE	-1.94%
VPH_S_ELEC	4	0.01	0.01	TRUE	-11.27%

Tabla 4.7: Comparación de resultados de realizar el muestreo estratificado y un muestreo aleatorio simple.

Como se detalló en el Capítulo 2 sección 2.5 el *deff* indica cuánto aumenta la varianza de una estimación debido a su diseño de muestreo en comparación con la varianza que se obtendría si se utilizara un diseño de muestreo aleatorio simple, ambos diseños con el mismo tamaño de muestra. En la Tabla 4.8 se muestran los resultados de calcular el *deff* con la estratificación seleccionada. En todos los caso resultó ser menor a uno, lo cual indica una mayor precisión en comparación con el muestreo aleatorio simple.

[7] Los cambios porcentuales fueron calculados sin redondear las varianzas, las varianzas que se muestran están redondeadas a 2 decimales.

P15PRI_IN	P15YM_AN	P3YM_HLI	P8A14AN	PCON_DISC
0.87	0.68	0.63	0.84	0.99
PSINDER	**VPH_AGUAFV**	**VPH_NODREN**	**VPH_PISODT**	**VPH_S_ELEC**
0.98	0.95	0.77	0.98	0.89

Tabla 4.8: Resultados del *deff*.

4.4.2. Comparación de la estratificación resultante con otras estratificaciones de interés

Se clasificaron a los municipios de acuerdo a otras estratificaciones que podrían resultar de interés, para luego, con las mismas diez variables utilizadas previamente, calcular las varianzas poblacionales de cada estimador, y así poder comparar la varianza poblacional de cada estratificación.

Las estratificaciones comparadas fueron las siguientes:

a). Estratificación obtenida en la tesis (5 estratos).

b). Estratificación por entidades federativas (32 estratos).

c). Estratificación por el tipo de municipio (3 estratos).

d). Estratificación por el cruce de las entidades federativas con el tipo de municipio (65 estratos).

e). Estratificación por el cruce de las entidades federativas y la obtenida en la tesis (90 estratos).

f). Estratificación por el cruce de las entidades federativas, tipo de municipio y la obtenida en la tesis (148 estratos).

Para realizar la clasificación de los municipios en c) se utilizó la base de datos de los Principales Resultados por Localidad del Censo de Población y Vivienda. Se trabajó únicamente con la información de las localidades, categorizando a cada localidad como rural o urbana en función de su población. Con población menor a 2,500 personas las localidades se categorizaron como rurales, y en otro caso como urbanas.

Una vez que se categorizaron las localidades, se procedió a la clasificación de los municipios en tres grupos: Totalmente rural, predominantemente rural y predominantemente urbano. Se consideró que un municipio era totalmente rural si el 100% de sus localidades eran rurales. Si al menos 50% de las localidades eran rurales, se clasificó como predominantemente rural, y si menos del 50% de la población residía en localidades rurales, se categorizó como municipio predominantemente urbano. Resultando:

- 809 municipios totalmente rurales.
- 1,616 municipios predominantemente rurales.
- 44 municipios predominantemente urbanos.

Por otro lado, la estratificación en f) es la que más se aproxima a lo que realizó el INEGI en 2010 para obtener sus 746 estratos, sin embargo, es importante recordar que el INEGI obtuvo los estratos a través de UPM en lugar de municipios, y clasificó primero en cuatro estratos por nivel sociodemográfico y luego los diferenció de acuerdo a la entidad federativa y tamaño de localidad, utilizando cuatro tamaños de localidad.

Los resultados de las varianzas poblacionales para cada estratificación se muestran en la Tabla 4.9.

Variable	a)	b)	c)	d)	e)	f)
P15PRI_IN	0.63	0.62	0.7	0.61	0.57	**0.57**
P8A14AN	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
P15YM_AN	0.31	0.35	0.45	0.35	0.28	**0.27**
PCON_DISC	0.37	0.35	0.37	0.35	**0.34**	**0.34**
P3YM_HLI	4.68	6.34	7.32	6.28	4.01	**3.91**
PSINDER	19.58	17.93	19.75	17.64	17.38	**17.21**
VPH_PISODT	17.3	16.86	17.47	16.7	16.47	**16.38**
VPH_S_ELEC	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
VPH_AGUAFV	0.18	0.18	0.19	0.18	**0.17**	**0.17**
VPH_NODREN	0.14	0.17	0.18	0.16	**0.13**	**0.13**

Tabla 4.9: Tabla de comparación de varianzas de las estratificaciones.

Como es natural, la estratificación con menor varianza fue la resultante de tener más estratos (columna f, Tabla 4.9), sin embargo, la varianza no disminuye radicalmente si es comparada con la varianza de la estratificación obtenida en la tesis (columna a). Incluso hay casos en donde el uso de solo la estratificación de la tesis obtiene mejores resultados que usar b, c y d, como en `P15YM_AN`, `P3YM_HLI` y `VPH_NODREN`.

4.4.3. Descripción de la estratificación obtenida

Con la estratificación seleccionada los 2,469 municipios fueron clasificados en 5 grupos:198 (8.02%) en el grupo 1, 304 (12.3%) en el grupo 2, 959 (38.8%) en el grupo 3, 758 (30.7%) en el grupo 4 y 250 (10.1%) en el grupo 5. Ver Figura 4.9.

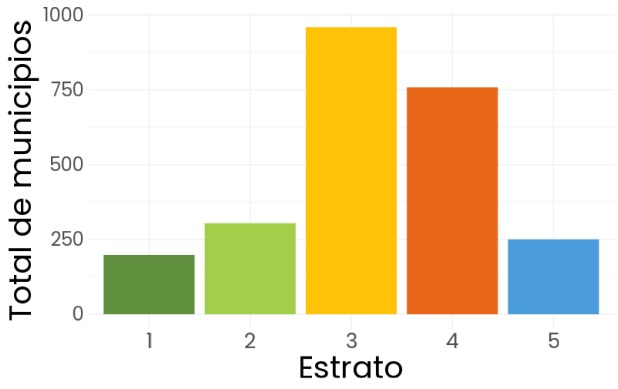


Figura 4.9: Distribución de los municipios en la estratificación.

Análisis de componentes principales

Para tener una primera visualización de cómo están distribuidos los grupos se decidió usar los componentes principales.

Los componentes principales se obtienen mediante una combinación lineal de las variables originales y sirven como una técnica de reducción de dimensionalidad y también como una de visualización. El primer componente principal representa la dirección en la cual los datos muestran la mayor dispersión o variabilidad, mientras que el segundo componente principal captura la siguiente mayor cantidad de variabilidad no explicada por el primer componente, y así sucesivamente (James et al., 2021).

Para saber qué tanto influyen las variables en cada componente, se debe de obtener la correlación de las variables con el componente de interés. Entre más grande la correlación de la variable x con el componente y significa que ésta tiene más peso en ese componente. Usualmente se considera que la correlación es pequeña si es menor a 0.5.

Se obtuvieron los componentes principales de las 45 variables sociodemograficas. Las observaciones sobre el primer y segundo componente principal se muestran en la Figura 4.10, diferenciando a cada grupo con un color diferente.

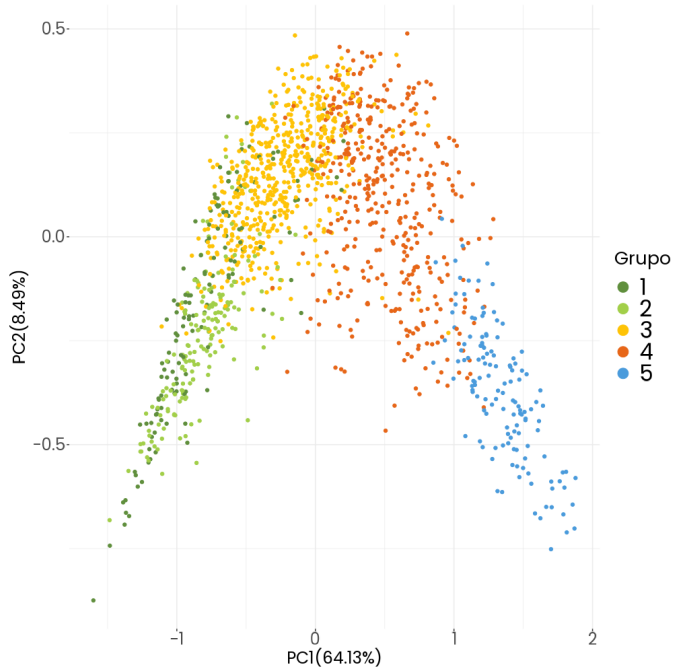


Figura 4.10: Gráfica del primer y segundo componente principal de los datos sociodemográficos utilizando la estratificación seleccionada. Se puede observar que los datos del grupo 1, 2 y 3 tienen valores negativos en el PC1, eso significa que las variables con mayor correlación positiva con el PC1 tendrán menores valores en estos grupos, y al contrario, las variables con mayor correlación negativa tendrán mayores valores en estos grupos. Por el contrario, los grupos 4 y 5 tienen valores positivos en el PC1, eso quiere decir que las variables correlacionadas con el PC1 positivamente tendrán un mayor valor en los grupos 4 y 5.

Las correlaciones entre cada variable y el primer componente principal (CP1) se muestran en la Tabla 4.10. La variable con mayor correlación en sentido positivo es la pobreza_pob, esto significa que a mayor valor en el CP1, mayor pobreza. Y de acuerdo a la Figura 4.10 esto quiere decir que los grupos 1 y 2 tienen menor pobreza (ya que sus valores en el CP1 son negativos), mientras que el grupo 5 es el que tiene mayor pobreza.

Por otro lado, la variable VPH_PC tiene correlación negativa, la interpretación es que a mayores valores en el CP1, menores en la variable, esto quiere decir que los municipios pertenecientes al grupo 5 y 4 tienen menos viviendas con computadoras en comparación a los municipios pertenecientes a los grupos 1, 2 y 3.

De hecho, todas las variables con alta correlación en sentido positivo están asociadas a mayor pobreza o carencias, mientras que las que tienen sentido negativo están asociadas a tener bienes básicos. Es decir, a mayor valor en el CP1 se tendrá un menor porcentaje en cuanto a tener computadora, lavadora, internet, televisión y celular; además de un menor grado escolar. Con esto en mente, el primer componente principal resumen muchas variables en términos de falta de bienes o mayor pobreza, de manera que un mayor valor en el CP1 indica mayor falta de bienes y pobreza.

Variable	**Correlación**	**Variable**	**Correlación**
pobreza_pob	0.91	POB65_MAS	0.15
pobreza_e_pob	0.90	PCON_LIMI	0.09
carencias3_pob	0.90	P15SEC_IN	0.09
plbm_pob	0.89	POB_AFRO	0.08
VPH_SINCINT	0.88	PCLIM_PMEN	-0.18
plb_pob	0.87	PSINDER	-0.18
carencias_pob	0.85	PAFIL_IPRIV	-0.32
P15YM_AN	0.84	HOGJEF_F	-0.34
VPH_NDACMM	0.80	PNACOE	-0.47
VPH_SNBIEN	0.77	POCUPADA	-0.63
P15YM_SE	0.76	PEA	-0.65
VPH_PISOTI	0.74	vul_ing_pob	-0.74
ic_segsoc_pob	0.73	P18YM_PB	-0.82
P3YM_HLI	0.68	VPH_REFRI	-0.83
PRO_OCUP_C	0.65	npnv_pob	-0.83
VPH_NODREN	0.65	VPH_CEL	-0.83
P15PRI_IN	0.60	VPH_TV	-0.83
ic_ali_pob	0.48	VPH_INTER	-0.84
VPH_S_ELEC	0.46	VPH_LAVAD	-0.85
VPH_AGUAFV	0.42	GRAPROES	-0.88
VPH_NDEAED	0.37	VPH_PC	-0.89
VPH_1CUART	0.35		
pobreza_m_pob	0.30		
PCON_DISC	0.27		

Tabla 4.10: Correlación entre variables sociodemográficas y el primer componente principal. Sólo se incluyen valores diferentes a cero, después de dos decimales.

En la Tabla 4.11 se muestran las correlaciones entre las variables y el segundo componente principal. A diferencia del primer componente principal, en el segundo las variables no tienen correlaciones demasiado altas, las más destacables son POB65_MAS, PCON_DISC y PCON_LIMI, todas ellas negativas. La interpretación de esto es que los municipios pertenecientes a los grupos 1, 2 y 5 tienen más población de 65 años o más que los municipios en los grupos 3 y 4. Lo mismo pasa para la población con discapacidad y población con limitación.

Variable	Correlación	Variable	Correlación
PRO_OCUP_C	0.45	PNACOE	0.10
VPH_SNBIEN	0.41	P15YM_SE	0.08
npnv_pob	0.35	P15YM_AN	0.08
VPH_PISOTI	0.34	PEA	0.06
P3YM_HLI	0.34	POCUPADA	0.04
VPH_NDEAED	0.32	PSINDER	-0.06
VPH_S_ELEC	0.32	plb_pob	-0.08
P18YM_PB	0.29	pobreza_pob	-0.12
VPH_PC	0.26	ic_segsoc_pob	-0.19
GRAPROES	0.24	VPH_SINCINT	-0.20
vul_ing_pob	0.24	PCLIM_PMEN	-0.23
VPH_1CUART	0.22	VPH_TV	-0.24
VPH_NODREN	0.22	VPH_LAVAD	-0.27
VPH_INTER	0.22	VPH_REFRI	-0.32
pobreza_e_pob	0.21	carencias_pob	-0.33
VPH_AGUAFV	0.20	pobreza_m_pob	-0.54
VPH_NDACMM	0.18	P15PRI_IN	-0.57
PAFIL_IPRIV	0.16	PCON_LIMI	-0.60
carencias3_pob	0.14	PCON_DISC	-0.63
ic_ali_pob	0.11	POB65_MAS	-0.68

Tabla 4.11: Correlación entre variables sociodemográficas y el segundo componente principal. Sólo se incluyen valores diferentes a cero, después de dos decimales.

Análisis descriptivo por grupos

La información de los grupos resultantes se resumió en las Tablas 4.12 y 4.13. La primera tabla muestra el resumen estadístico por cada grupo, considerando la media y el rango. Por su parte, la Tabla 4.13 muestra las proporciones de cada grupo por entidad federativa.

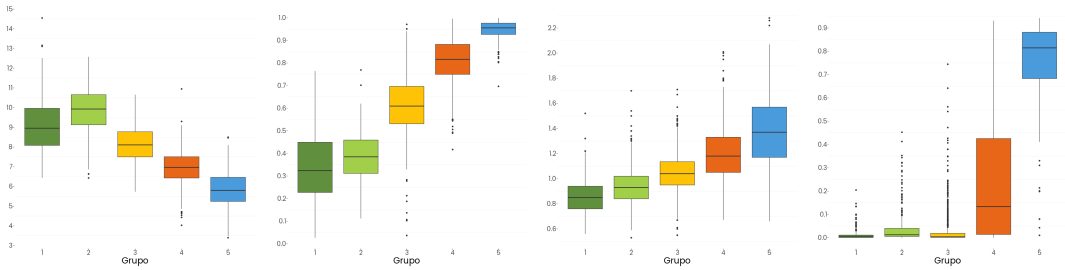
De la Tabla 4.12 se puede deducir lo siguiente:

- Los grupos con menor población en pobreza son el 1 y 2, mientras que en los grupos 3, 4 y 5 la población tiene más carencias.
- En promedio hay más población de 65 años y más en el grupo 4 (12%) y menor en el grupo 2 (9%).
- La población con alguna discapacidad o limitación es proporcional en todos los grupos (aproximadamente 6% de población con discapacidad y 12% con limitación).

- El grupo 2 es el que lidera el promedio de años de escolaridad y menor porcentaje de población con primaria y secundaria incompleta, seguido por el grupo 1 y 3.

- El grupo 2 es el que tiene más población ocupada, en promedio 47%, mientras que el grupo 3 es el que menos tiene (30%).

- En general el grupo 1 y 2 son los que tienen más bienes como refrigerador, lavadora, computadora, teléfono celular e internet.



(a) GRAPROES (b) pobreza_pob (c) PRO_OCUP_C (d) P3YM_HLI

Figura 4.11: Diagramas de cajas de las variables: Grado promedio de escolaridad, pobreza poblacional, promedio de ocupantes por cuarto y población de 3 años o más que habla alguna lengua indígena.

Algunas de las variables mostradas en la Tabla 4.12 se seleccionaron y en la Figura 4.11 se presenta la información mediante box plots, de lo cual se puede extraer lo siguiente:

En (a), se puede observar que el grupo 2 cuenta con el mayor grado promedio de escolaridad, con una mediana de 9.93 años y un mínimo de 6.43 años, mientras que el grupo 5 alcanza solamente un máximo de 8.5 años y mediana de 5.8 años.

Respecto a la pobreza poblacional en (b) se observa que el grupo 5 tiene el primer lugar, seguido por los grupos 4, 3, 2 y 1, en términos de mediana. En el grupo 5, en promedio, el 95% de la población vive en pobreza, mientras que en el grupo 1, solamente el 34% de la población lo hace. Y como se observa en (c) el grupo 5 también tiene mayor promedio de ocupantes por cuarto (1.39 en promedio).

En (d), se muestra que la pobreza poblacional y la población de tres años o más que habla alguna lengua indígena están correlacionadas. De hecho, el grupo 5, siendo el más pobre, también es el que tiene el mayor porcentaje de población hablante de lengua indígena.

Variable	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
pobreza_pob	0.34 (0.03 - 0.76)	0.38 (0.11 - 0.77)	0.61 (0.04 - 0.97)	0.81 (0.42 - 1)	0.94 (0.7 - 1)
pobreza_e_pob	0.03 (0 - 0.23)	0.04 (0.01 - 0.13)	0.11 (0 - 0.48)	0.29 (0.04 - 0.77)	0.56 (0.23 - 0.97)
pobreza_m_pob	0.3 (0.03 - 0.57)	0.34 (0.1 - 0.64)	0.5 (0.04 - 0.78)	0.52 (0.22 - 0.82)	0.38 (0.02 - 0.67)
vul_ing_pob	0.07 (0 - 0.24)	0.09 (0.01 - 0.23)	0.04 (0 - 0.2)	0.01 (0 - 0.11)	0 (0 - 0.03)
npnv_pob	0.22 (0.02 - 0.66)	0.23 (0.02 - 0.44)	0.08 (0 - 0.25)	0.02 (0 - 0.13)	0 (0 - 0.05)
ic_segsoc_pob	0.55 (0.2 - 0.91)	0.51 (0.2 - 0.85)	0.76 (0.36 - 0.94)	0.82 (0.06 - 0.96)	0.86 (0.29 - 0.97)
ic_ali_pob	0.15 (0.01 - 0.37)	0.18 (0.03 - 0.57)	0.23 (0 - 0.53)	0.27 (0.01 - 0.71)	0.36 (0.02 - 0.86)
carencias_pob	0.71 (0.32 - 0.98)	0.68 (0.39 - 0.96)	0.88 (0.58 - 1)	0.97 (0.8 - 1)	0.99 (0.92 - 1)
carencias3_pob	0.12 (0.01 - 0.5)	0.14 (0.03 - 0.4)	0.28 (0.03 - 0.67)	0.49 (0.11 - 0.81)	0.69 (0.36 - 0.98)
plb_pob	0.41 (0.04 - 0.77)	0.48 (0.16 - 0.8)	0.65 (0.05 - 0.98)	0.82 (0.45 - 1)	0.95 (0.72 - 1)
plbm_pob	0.13 (0.01 - 0.43)	0.14 (0.03 - 0.5)	0.28 (0.02 - 0.88)	0.5 (0.08 - 0.96)	0.78 (0.44 - 0.99)
POB65_MAS	0.11 (0.03 - 0.22)	0.09 (0.01 - 0.27)	0.1 (0.04 - 0.23)	0.12 (0.01 - 0.32)	0.1 (0.03 - 0.27)
PNACOE	0.11 (0.01 - 0.63)	0.16 (0 - 0.7)	0.1 (0 - 0.66)	0.06 (0 - 0.45)	0.02 (0 - 0.14)
P3YM_HLI	0.01 (0 - 0.2)	0.04 (0 - 0.45)	0.03 (0 - 0.74)	0.24 (0 - 0.93)	0.76 (0.01 - 0.94)
POB_AFRO	0.01 (0 - 0.11)	0.02 (0 - 0.24)	0.02 (0 - 0.39)	0.04 (0 - 0.96)	0.02 (0 - 0.38)
PCON_DISC	0.06 (0.02 - 0.12)	0.05 (0.02 - 0.1)	0.06 (0.02 - 0.13)	0.07 (0 - 0.32)	0.06 (0 - 0.21)
PCON_LIMI	0.13 (0.07 - 0.24)	0.12 (0.06 - 0.2)	0.12 (0.04 - 0.23)	0.14 (0.02 - 0.37)	0.12 (0.01 - 0.28)
PCLIM_PMEN	0.01 (0 - 0.02)	0.01 (0 - 0.02)	0.01 (0 - 0.03)	0.01 (0 - 0.05)	0.01 (0 - 0.03)
P15YM_AN	0.03 (0 - 0.1)	0.03 (0.01 - 0.08)	0.05 (0.01 - 0.16)	0.1 (0 - 0.24)	0.16 (0.02 - 0.35)
P15YM_SE	0.03 (0 - 0.12)	0.03 (0.01 - 0.09)	0.05 (0.01 - 0.17)	0.09 (0 - 0.25)	0.14 (0.02 - 0.28)
P15PRI_IN	0.09 (0.01 - 0.22)	0.06 (0.02 - 0.25)	0.1 (0.03 - 0.23)	0.13 (0 - 0.28)	0.13 (0.04 - 0.26)
P15SEC_IN	0.03 (0 - 0.09)	0.02 (0.01 - 0.05)	0.02 (0.01 - 0.06)	0.02 (0 - 0.09)	0.03 (0.01 - 0.06)
P18YM_PB	0.26 (0.05 - 0.76)	0.32 (0.09 - 0.56)	0.2 (0.07 - 0.41)	0.14 (0.02 - 0.31)	0.09 (0.01 - 0.23)
GRAPROES	0.51 (0.27 - 1)	0.57 (0.27 - 0.82)	0.43 (0.21 - 0.65)	0.32 (0.06 - 0.68)	0.22 (0 - 0.46)
PEA	0.44 (0.28 - 0.64)	0.48 (0.28 - 0.56)	0.45 (0.15 - 0.58)	0.4 (0.05 - 0.58)	0.32 (0.11 - 0.58)
POCUPADA	0.43 (0.26 - 0.62)	0.47 (0.27 - 0.55)	0.44 (0.14 - 0.57)	0.4 (0.05 - 0.58)	0.3 (0.09 - 0.58)
PSINDER	0.21 (0.04 - 0.57)	0.24 (0.05 - 0.44)	0.28 (0.03 - 0.66)	0.23 (0.02 - 0.78)	0.18 (0.01 - 0.84)
PAFIL_IPRIV	0.02 (0 - 0.38)	0.02 (0 - 0.17)	0.01 (0 - 0.08)	0 (0 - 0.16)	0 (0 - 0.58)
HOGJEF_F	0.29 (0.15 - 0.48)	0.31 (0.13 - 0.43)	0.29 (0.09 - 0.45)	0.27 (0.12 - 0.50)	0.26 (0.11 - 0.50)
PRO_OCUP_C	0.19 (0.02 - 0.57)	0.24 (0 - 0.67)	0.3 (0.01 - 0.67)	0.38 (0.08 - 0.85)	0.49 (0.07 - 1)
VPH_PISOTI	0.02 (0 - 0.31)	0.02 (0 - 0.09)	0.05 (0 - 0.36)	0.11 (0 - 0.62)	0.24 (0.02 - 0.67)
VPH_1CUART	0.04 (0.01 - 0.34)	0.06 (0.01 - 0.49)	0.06 (0.01 - 0.32)	0.1 (0 - 0.37)	0.09 (0.01 - 0.44)
VPH_S_ELEC	0.02 (0 - 0.12)	0.01 (0 - 0.03)	0.01 (0 - 0.14)	0.02 (0 - 0.34)	0.05 (0 - 0.49)
VPH_AGUAFV	0.02 (0 - 0.16)	0.02 (0 - 0.18)	0.05 (0 - 0.79)	0.08 (0 - 0.62)	0.13 (0 - 0.7)
VPH_NODREN	0.05 (0 - 0.43)	0.03 (0 - 0.18)	0.07 (0 - 0.78)	0.19 (0 - 0.98)	0.46 (0.01 - 0.99)
VPH_NDEAED	0 (0 - 0.04)	0 (0 - 0.02)	0 (0 - 0.13)	0.01 (0 - 0.15)	0.02 (0 - 0.28)
VPH_NDACMM	0.35 (0.07 - 0.66)	0.44 (0.15 - 0.75)	0.5 (0.17 - 0.93)	0.72 (0.04 - 0.97)	0.88 (0.51 - 0.98)
VPH_SNBIEN	0.01 (0 - 0.1)	0.01 (0 - 0.03)	0.02 (0 - 0.21)	0.07 (0 - 0.39)	0.22 (0.02 - 0.55)
VPH_REFRI	0.92 (0.62 - 0.99)	0.9 (0.72 - 0.98)	0.83 (0.42 - 0.98)	0.64 (0.07 - 0.95)	0.35 (0.04 - 0.82)
VPH_LAVAD	0.78 (0.32 - 0.95)	0.78 (0.53 - 0.94)	0.67 (0.17 - 0.93)	0.43 (0.05 - 0.88)	0.14 (0 - 0.57)
VPH_TV	0.91 (0.37 - 0.98)	0.93 (0.7 - 0.97)	0.88 (0.54 - 0.97)	0.74 (0.08 - 0.93)	0.51 (0.1 - 0.95)
VPH_PC	0.3 (0.05 - 0.85)	0.37 (0.08 - 0.69)	0.2 (0.01 - 0.42)	0.09 (0 - 0.31)	0.04 (0 - 0.17)
VPH_CEL	0.88 (0.52 - 0.97)	0.9 (0.39 - 0.96)	0.82 (0.22 - 0.94)	0.64 (0.02 - 0.95)	0.44 (0.07 - 0.94)
VPH_INTER	0.41 (0.03 - 0.92)	0.51 (0.1 - 0.83)	0.31 (0.02 - 0.69)	0.16 (0 - 0.55)	0.08 (0 - 0.45)
VPH_SINCINT	0.53 (0.06 - 0.88)	0.43 (0.14 - 0.88)	0.64 (0.3 - 0.96)	0.8 (0.03 - 1)	0.9 (0.55 - 1)

Tabla 4.12: El resumen estadístico de las variables de acuerdo al estrato de pertenencia de los municipios. El primer número corresponde a la media, y los números entre paréntesis corresponden al rango, es decir, el valor mínimo y el máximo.

Estado	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
Aguascalientes	-	4 (36.36%)	7 (63.64%)	-	-
Baja California	6 (100%)	-	-	-	-
Baja California Sur	5 (100%)	-	-	-	-
Campeche	-	5 (41.67%)	4 (33.33%)	3 (25%)	-
Chiapas	-	1 (0.81%)	38 (30.65%)	55 (44.35%)	30 (24.19%)
Chihuahua	16 (23.88%)	37 (55.22%)	1 (1.49%)	10 (14.93%)	3 (4.48%)
Ciudad de México	16 (100%)	-	-	-	-
Coahuila	32 (84.21%)	2 (5.26%)	4 (10.53%)	-	-
Colima	-	4 (40%)	6 (60%)	-	-
Durango	27 (69.23%)	-	11 (28.21%)	-	1 (2.56%)
Guanajuato	-	5 (10.87%)	41 (89.13%)	-	-
Guerrero	-	-	27 (33.33%)	37 (45.68%)	17 (20.99%)
Hidalgo	-	11 (13.1%)	51 (60.71%)	21 (25%)	1 (1.19%)
Jalisco	-	30 (24%)	93 (74.4%)	1 (0.8%)	1 (0.8%)
Michoacán	-	3 (2.65%)	105 (92.92%)	5 (4.42%)	-
Morelos	-	6 (16.67%)	29 (80.56%)	1 (2.78%)	-
México	-	36 (28.8%)	71 (56.8%)	18 (14.4%)	-
Nayarit	1 (5%)	3 (15%)	13 (65%)	2 (10%)	1 (5%)
Nuevo León	6 (11.76%)	20 (39.22%)	25 (49.02%)	-	-
Oaxaca	-	31 (5.44%)	67 (11.75%)	305 (53.51%)	167 (29.3%)
Puebla	-	5 (2.3%)	79 (36.41%)	123 (56.68%)	10 (4.61%)
Querétaro	-	4 (22.22%)	13 (72.22%)	1 (5.56%)	-
Quintana Roo	-	7 (63.64%)	-	4 (36.36%)	-
San Luis Potosí	-	3 (5.17%)	41 (70.69%)	13 (22.41%)	1 (1.72%)
Sinaloa	17 (94.44%)	1 (5.56%)	-	-	-
Sonora	72 (100%)	-	-	-	-
Tabasco	-	1 (5.88%)	15 (88.24%)	1 (5.88%)	-
Tamaulipas	-	6 (13.95%)	37 (86.05%)	-	-
Tlaxcala	-	16 (26.67%)	42 (70%)	2 (3.33%)	-
Veracruz	-	26 (12.26%)	85 (40.09%)	83 (39.15%)	18 (8.49%)
Yucatán	-	33 (31.13%)	-	73 (68.87%)	-
Zacatecas	-	4 (6.9%)	54 (93.1%)	-	-

Tabla 4.13: Número y porcentaje de municipios, en cada estado, pertenecientes a cada estrato.

De toda la información previa para cada grupos se puede resumir lo siguiente:

- **Grupo 1:** Consta de **198 municipios (8.02% del total)**. Es el grupo con menor población en situación en pobreza y el segundo con más años promedio de escolaridad. Baja California, Baja California Sur, Ciudad de México y Sonora están conformados en su totalidad por municipios en este grupo, mientras que Sinaloa y Coahuila poseen el 94.4% y 84.2% de sus municipios en este grupo respectivamente. Cabe resaltar que sólo 10 estados de la república mexicana cuentan con algún municipio en este estrato.

- **Grupo 2:** Lo constituyen **304 municipios (12.3% del total)**. Es el grupo con mayores índices de educación, es decir, es el grupo con menos población con primaria y secundaria incompleta, así como con más población de 18 años y más con educación posbásica y con mayor promedio de escolaridad. Es el segundo grupo con menor población en situación de pobreza y en algunos casos el grupo con menos carencias, por ejemplo es el grupo que tiene más viviendas con televisión, celular, internet y computadora. Más de la mitad de los municipios de Chihuahua y Quintana Roo pertenecen a este grupo.

- **Grupo 3:** Se compone de **959 municipios (38.8% del total)**. Es el tercer grupo menos vulnerable y pobre, y también el tercero con mayores índices de escolaridad. En este grupo están categorizados la mayoría de municipios y sólo 7 estados no tienen ningún municipio perteneciente a este grupo. Guanajuato (89%), Michoacán (92%), Tabasco (88%) y Tamaulipas (86%) tienen más del 85% de sus municipios en este grupo.
- **Grupo 4:** Con **758 municipios (30.7% del total)**. Es el segundo grupo más pobre y vulnerable. Es el grupo con más población en situación de pobreza moderada y el segundo con menor índice de educación. Yucatán (69%), Puebla (57%) y Oaxaca (54%) poseen más del 50% de sus municipios con esta clasificación.
- **Grupo 5: Constituye el grupo más pobre y vulnerable**, contiene **250 municipios (10.1% del total)** y sólo 11 estados son parte de este estrato, entre los cuales se encuentran Oaxaca, Chiapas y Guerrero con más del 20% de sus municipios en esta estratificación. Es el grupo con más hablantes de lengua indígena, mayor población con carencias y menor grado de escolaridad.

En la Figura 4.12 se muestran los municipios de México utilizando la clasificación resultante de la estratificación seleccionada. Se puede observar que el grupo 1 y 2 están principalmente en el norte del país, mientras que el 3 abarca el centro, y el cuatro el sur.

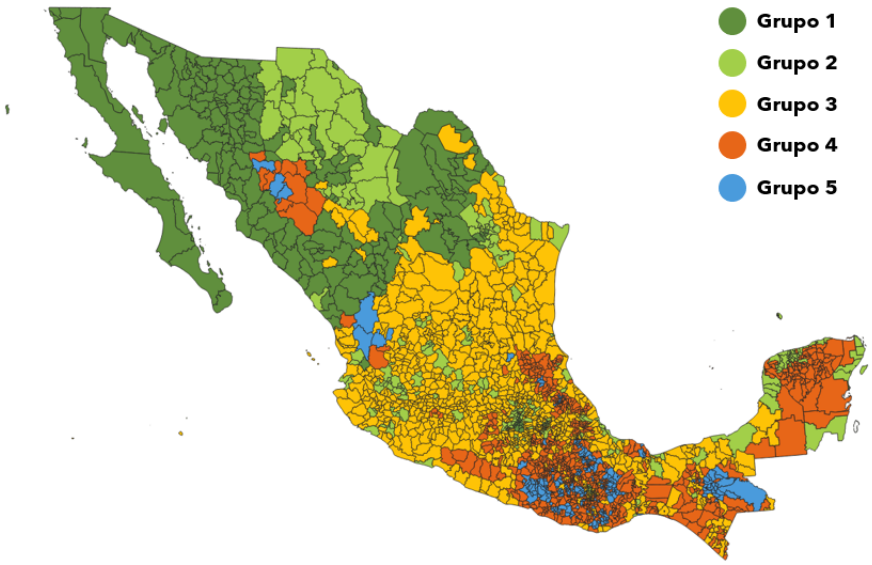


Figura 4.12: Mapa de los Estados Unidos Mexicanos utilizando la estratificación elegida.

Capítulo 5

Conclusiones

En esta sección se discutirán las decisiones tomadas para la realización de esta tesis, así como las dificultades enfrentadas a lo largo del proceso.

La motivación de esta tesis fue crear una estratificación de los municipios en los Estados Unidos Mexicanos similar a como lo hace el INEGI en la primera etapa de su estratificación, es decir, cuando se crean los cuatro estratos a nivel nacional, pero en este caso incluyendo restricciones geográficas desde un inicio.

La estratificación se realizó a nivel municipal debido a que es la información más desagregada y completa para el público en general, utilizando la base de los Principales Resultados por Localidad del Censo de Población y Vivienda 2020. No se hizo a nivel localidad, ya que en las localidades más chicas no aparece la información completa. El hacerlo de esta manera también sirvió para poder utilizar los Indicadores de Pobreza Municipal 2015 del CONEVAL y así contar con más información sociodemográfica para hacer la estratificación.

El utilizar dos bases de datos de distintas fuentes representó un reto ya que nos enfrentamos a lo siguiente:

a). Unir las bases de datos por los nombres de los estados y municipios fue complicado debido a que no todos los nombres coinciden y están sucios, por ejemplo, en la base de datos del INEGI existen municipios aparentemente duplicados ya que ambos aparecen dos veces con el mismo nombre y en el mismo estado: San Juan Mixtepec y San Pedro Mixtepec. Buscando estos nombres en los Indicadores de Pobreza Municipal se observa que los nombres correctos son San Juan Mixtepec -Dto. 08 -, San Juan Mixtepec -Dto. 26 -, San Pedro Mixtepec -Dto. 22 - y San Pedro Mixtepec -Dto. 26 -. Para diferenciar a cuál correspondía cada municipio, se obtuvo la población total y se relacionó a aquel municipio del INEGI en el que la población total fuera similar a la población total registrada por el CONEVAL.

b). Una vez realizado lo anterior, se logró intersecar a 2,433 municipios de los 2,469 totales. Para poder unir todos fue necesario limpiar los nombres quitando abreviaciones o simplemente modificando los nombres, por ejemplo Heroica Ciudad de Juchitán de Zaragoza por Juchitán de Zaragoza.

c). Había datos faltantes en la base de datos del CONEVAL, algunos correspondían a mu-

nicipios formados entre 2015 y 2020, y otros simplemente no tenían información. La información fue imputada promediando los datos de los municipios vecinos, esto se hizo así porque muchas veces municipios vecinos tienen características similares. En total se imputaron 23 municipios faltantes utilizando la información de otros 90 municipios.

Una vez que las bases de datos fueron unidas, la mayoría de variables fueron transformadas a proporciones ya que así es más fácil hacer comparaciones entre municipios porque aveces los municipios varían mucho respecto al total de su población. Esto podría haberse omitido o incluso trabajar con otro tipo de transformaciones, sin embargo, se hizo así porque se pretendía que la comparación entre municipios fuera fácil e intuitiva.

Ya que de unir ambas bases resultaron 246 variables, se seleccionaron sólo algunas. Esto porque muchas están correlacionadas y no es práctico incluirlas todas. Se seleccionaron algunas de las variables que se utilizaron para la realización de la Muestra Maestra del INEGI, pero también se agregaron otras que podrían resultar de interés actual o que involucran a grupos vulnerables para que estos quedaran representados en cada grupo. Sin embargo, es importante notar que el probar distintas variables podría generar distintas estratificaciones.

Se probó con dos tipos de distancias para los datos geográficos, sin embargo, pueden explorarse otras formas de distancias, por ejemplo, tomando otro punto de referencia en los municipios (no el centroide) o calcular la distancia con las rutas que comunican a los municipios.

Respecto a los parámetros requeridos por el algoritmo:

- Se probó con distintos métodos para seleccionar el número adecuado de grupos, y se llegó a una decisión final considerando lo realizado por en INEGI que fue crear cuatro estratos en la primera etapa de la estratificación. En una segunda etapa, como se vio en la comparación de la estratificación resultante con otras estratificaciones de interés, se podría diferenciar cada grupo por tipo de municipio o por estado. También se podría haber elegido un número grande de grupos desde el principio, como $32 * 4 = 128$ que es el resultado máximo de lo que correspondería diferenciar los cuatro estratos por cada entidad federativa.
- Para la elección de α se probó con varias mallas, pero se vio que no había gran diferencia entre usar una malla de 21 puntos o más, por lo que se conservó esta.
- Se le dio el mismo peso a todas las características geográficas, sin embargo, para otros proyectos, algunas variables podrían tener más importancia que otras por lo que podrían ponderarse de distinta manera.

En cuanto al tiempo de ejecución del código, es muy rápido, excepto en la parte que se ejecutan los métodos para sugerir el número de grupos, en especial el método gap que tarda aproximadamente 40 minutos. Todo lo demás no toma más de 3 minutos.

Respecto a la elección de la mejor estratificación otro de los retos de esta tesis fue que incluso tratándose del mismo Censo, a distintas escalas, las claves de los municipios y estados no corresponden, por lo que se tuvo que hacer la unión nuevamente por nombre

del estado y municipio, teniendo que realizar un proceso parecido al que se realizó para obtener la base de datos de las características sociodemograficas (cuando se unieron lo Indicadores de Pobreza Municipal y los Principales Resultados del Censo 2020), ya que algunos nombres en la base de datos a nivel localidad y geolectoral cambian, aunque ambos provengan del Censo 2020 realizado por el INEGI.

Por simplicidad se eligió estimador HT para totales ya que es fácil calcular la varianza poblacional, sin embargo, se puede elegir otro estimador distinto (como promedios o porcentajes) e incluso combinarlos.

Se eligieron diez variables que se consideraron de importancia para seleccionar la estratificación, sin embargo, se pudieron seleccionar cualquiera otras e incluso todas. Se hizo así porque se pretendía que la estratificación elegida tuviera poca varianza poblacional en las variables de interés. También se pudo haber elegido otra forma de determinar la mejor estratificación, sin embargo, la idea de los rankings era ver en promedio cuál estratificación tenía menos varianza en todas las variables.

Los números de muestra sugeridos por el método empleado fueron muy variables, pero se decidió conservar el más grande para abarcar todos los demás casos.

Los resultados presentados en esta tesis muestran que la estratificación seleccionada, mediante el método de conglomerados jerárquico con restricciones espaciales ha sido efectiva para mejorar la precisión de las estimaciones de la población en comparación con el diseño de muestreo aleatorio simple sin reemplazo. Esto se evidencia por la menor varianza poblacional asociada al diseño de muestreo con estratificación. En algunos casos, se mejoró la precisión de los estimadores hasta en un 30 %. Otro indicador que demuestra que se mejoró la precisión de los estimadores fue el *deff*, pues en todos los casos los resultados fueron menores a uno.

Finalmente, ya que usualmente los municipios cercanos tienen condiciones similares y eso se podría aprovechar, se consideró que el usar restricciones geográficas fue de utilidad.

Cabe resaltar que el objetivo de este trabajo no fue replicar el trabajo que realiza el INEGI, sino evaluar un método particular de análisis de conglomerados para la creación de estratos de muestreo, utilizando solamente información de acceso público.

Capítulo 6

Anexo A: Nociones básicas

Análisis de componentes principales

Si se quisieran visualizar n observaciones con p características como parte del análisis exploratorio, se podría hacer examinando las gráficas $2D$ de los datos; cada una con dos características a la vez y así resultando $\binom{p}{2} = \frac{p(p-1)}{2}$ gráficas. Por ejemplo si $p = 20$ resultarían 190 gráficas, de las cuales ninguna de ellas sería informativa ya que sólo contendrían una pequeña fracción del total de la información en los datos.

Por lo tanto es requerido un mejor método para visualizar las n observaciones cuando p es grande. Lo ideal sería encontrar una representación de baja dimensionalidad que contuviera tanta información como fuese posible; el análisis de componentes principales provee una herramienta para hacerlo.

La idea es que cada una de las n observaciones viven en una espacio $p-$dimensional, pero no todas las dimensiones son igual de importantes. Así el análisis de componentes principales (PCA, por sus siglas en inglés) se encarga de encontrar las dimensiones más importantes, donde la importancia se mide por la variación de las observaciones a lo largo de cada dimensión. Cada una de las dimensiones encontradas por el PCA es una combinación lineal de las p características. A continuación se explica cómo encontrar estas dimensiones (o componentes principales).

El primer componente principal (Z_1) de un conjunto de características $x_1, x_2, ..., x_p$ es la combinación lineal normalizada:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + ... + \phi_{p1} X_p, \tag{6.1}$$

tal que tiene la varianza más grande. Normalizada significa que $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. Los elementos $\phi_{11}, \phi_{21}, ..., \phi_{p1}$ son llamados los *loadings* del primer componente principal.

Cálculo de los componentes principales

Para calcular el primer componente principal de un conjunto de datos $\mathbf{X}$ de $n \times p$, todos los datos son centrados con media cero y luego se busca la combinación lineal:

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + ... + \phi_{p1} x_{ip}, \tag{6.2}$$

tal que tenga la mayor varianza muestral, sujeta a $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. Es decir, el vector *loading* del primer componente principal ($\phi_1 = [\phi_{11} \ \phi_{21} \ ... \ \phi_{p1}]^T$) resuelve el problema de optimización:

$$\max_{\phi_{11},...,\phi_{p1}} \{\frac{1}{n}\sum_{i=1}^{n}\big(\sum_{j=1}^{p}\phi_{j1}x_{ij}\big)^2\} \text{ sujeto a } \sum_{j=1}\phi_{j1}^2 = 1. \tag{6.3}$$

De la ecuación 6.2, la función objetivo se puede escribir como $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$. Como $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$, el promedio de $z_{11},...,z_{n1}$ será cero también. Por eso la función objetivo de 6.3 es solo la varianza muestral de los n valores de z_{i1}. Los valores $z_{11},...,z_{n1}$ son llamados los *scores* del componente principal. El problema 6.3 puede ser resuelto con álgebra lineal utilizando los vectores propios. El vector *loading* ϕ_1 define la dirección en la que que los datos varían más.

Sea **X** una matriz que representa a un conjunto de n observaciones y p características, donde x_{ki} es la $k-$ésima observación con la característica i. Para obtener los componentes principales se debe realizar lo siguiente:

1. Se calcula **A**: la matriz de covarianzas de **X**, donde

$$cov(X_i, X_j) = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \bar{X}_i)(x_{kj} - \bar{X}_j). \tag{6.4}$$

2. Se calculan los valores y vectores propios de **A**.

3. Los vectores propios se ordenan según la magnitud de su valor propio. Y cada uno corresponde a los diferentes componentes principales.

Proporción de Varianza explicada

Es natural preguntarse qué tanta información en un conjunto de datos se pierde al proyectar las observaciones en los primeros componentes principales, esto se traduce a saber cuánta varianza no está contenida en los primeros componente principales. Para esto se utiliza la proporción de varianza explicada, la cuál esta definida como:

$$\sum_{j=1}^{p}\text{Var}(\mathbf{A}_j) = \sum_{j=1}^{p}\frac{1}{n}\sum_{i=1}^{n}x_{ij}^2, \tag{6.5}$$

y la varianza explicada por el $m-$ésimo componente principal es:

$$\frac{1}{n}\sum_{i=1}^{n}z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\big(\sum_{j=1}^{p}\phi_{jm}x_{ij}\big)^2. \tag{6.6}$$

Por lo tanto, la proporción de varianza explicada (PVE por sus siglas en inglés) del $m-$ésimo componente principal está dada por:

$$\frac{\sum_{i=1}^{n}(\sum_{j=1}^{p}\phi_{jm}x_{ij})^2}{\sum_{j=1}^{p}\sum_{i=1}^{n}x_{ij}^2}. \tag{6.7}$$

La PVE de cada componente principal es una cantidad positiva. Para calcular PVE acumulada de los M componentes principales, simplemente se suma la ec. 6.7 hasta los primeros M componentes.

Capítulo 7

Anexo B: Simulación de bases de datos para ilustrar los métodos de enlace

Para lo mostrado en la Figura 3.2 se simularon cuatro bases de datos. Para la primera se utilizó la distribución normal, creando tres conjuntos de datos en dos dimensiones. Para la segunda se utilizó la función `mlbench.shapes` de la librería `mlbench` (Leisch et al., 2021), la cual permite generar conjuntos de datos con distintas formas geométricas. Y los dos últimos conjuntos de datos se simularon a través de funciones de senos y cosenos. El código para simular las bases de datos, agruparlas y graficarlas de acuerdo a cada método de enlace se muestra a continuación.

```
pacman::p_load(NbClust,ClustGeo,ggplot2,readxl,purrr,
               cluster,factoextra,tidyr,dplyr,scales,
               mlbench,showtext,paletteer)

font_add_google(c("Poppins"))
showtext_auto()


metodos <- c("single","complete","average","centroid","ward")

# esta funci n obtiene la agrupaci n para los m todos:
# single, complete, average, centroid y ward
obten_clusters <- function(data, k){

  distance <- dist(scale(data), method = 'euclidean')
  complete <- hclust(distance, method = 'complete')
  data$complete <- cutree(complete, k = k)

  average <- hclust(distance, method = 'average')
  data$average <- cutree(average, k = k)

  single <- hclust(distance, method = 'single')
```

```
23    data$single <- cutree(single, k = k)
24
25    centroid <- hclust(distance, method = 'centroid')
26    data$centroid <- cutree(centroid, k = k)
27
28    ward <- hclust(distance, method = 'ward.D')
29    data$ward <- cutree(ward, k = k)
30
31    data <- data%>%
32      mutate(complete = as.factor(complete),
33             average = as.factor(average),
34             single = as.factor(single),
35             centroid = as.factor(centroid),
36             ward = as.factor(ward))
37    return(data)
38  }
39
40  # funci n para crear las gr ficas con distintos m todos
41  obten_Grafica <- function(data,metodo_enlace,tamanio_punto){
42    # la variable x debe llamarse V1
43    # la variable y debe llamarse V2
44    G <- ggplot(data,aes(x=V1,y=V2))+
45      theme_void()+
46      theme(text=element_text(family="Poppins", size=10),
47            plot.title = element_text(face="bold", size=10),
48            legend.title = element_blank(),
49            axis.title = element_blank(),
50            axis.text = element_blank(),
51            panel.grid.major = element_blank(),
52            panel.grid.minor = element_blank(),
53            panel.border = element_blank(),
54            panel.background = element_blank(),
55            legend.position="none")
56
57    if(metodo_enlace=="single"){
58      G <- G+geom_point(aes(color=factor(single)),size=tamanio_
          punto)+
59        labs(title="Clusters creados con el m todo single")
60      return(G)
61    }
62    else if(metodo_enlace=="complete"){
63      G <- G+geom_point(aes(color=factor(complete)),size=tamanio_
          punto)+
64        labs(title="Clusters creados con el m todo complete")
65      return(G)
66    }
67    else if(metodo_enlace=="average"){
68      G <- G+geom_point(aes(color=factor(average)),size=tamanio_
```

```
        punto)+
      labs(title="Clusters creados con el m todo average")
    return(G)
  }
  else if(metodo_enlace=="centroid"){
    G <- G+geom_point(aes(color=factor(centroid)),size=tamanio_
       punto)+
      labs(title="Clusters creados con el m todo cetroid")
    return(G)
  }

  else if(metodo_enlace=="ward"){
    G <- G+geom_point(aes(color=factor(ward)),size=tamanio_
       punto)+
      labs(title="Clusters creados con el m todo ward")
    return(G)
  }else{
    G <- paste(metodo_enlace, "No es un m todo v lido.",
      "Elige un m todo v lido: single, complete, centroid,
         ward o average")

  }
  return(print(G))

}

# paletas de colores:
paletteer_d("ggthemes::excel_Vapor_Trail")
paletteer_d("ggthemes::excel_Depth")
paletteer_d("ggthemes::excel_Gallery")


# Tres grupos con ruido
set.seed(6)
data <- data.frame(V1=c(rnorm(100,5,2),
                        rnorm(100,1,1),
                        rnorm(100,-2,1)),

                   V2=c(rnorm(100,5,1),
                        rnorm(100,1,1),
                        rnorm(100,5,1)))
plot(data)
data_with_clusters <- obten_clusters(data,3)

for(m in metodos){
  g <- obten_Grafica(data_with_clusters,m,8)+
    scale_color_manual(values=c("#FFE082","#54A021","#FF8F00"))
       +
```

```
    labs(title="")
  plot(g)
  ggsave(g,filename = paste0("tres_clusters_",m,".png"),
         device = "png",width = 5,height = 5, dpi=150)
}


# triangulos
base <- mlbench.shapes(n=800)
data <- as.data.frame(base$x)%>%
  rename(V1=x4)%>%
  filter(V1>0,
         V2>0)

data <- rbind(data%>%
                mutate(V2=V2-.5),
              data%>%
                mutate(V2=(V2-.5)*-1))
plot(data)

data_with_clusters <- obten_clusters(data,2)
for(m in metodos){
  g <- (obten_Grafica(data_with_clusters,m,8)+
         scale_color_manual(values=c("#FF8F00","#54A021")))+
    labs(title="")
  plot(g)
  ggsave(g,filename = paste0("triangulos_",m,".png"),
         device = "png",width = 5,height = 5, dpi=150)
}


# dos clusters de diferentes tama os (uno muy grande)
genera_circulos <- function(r,a){
  set.seed(6)
  theta = runif(1000, 0,2*pi)
  x = cos(theta) + rnorm(100, 0, 0.03)
  y = sin(theta) + rnorm(100, 0, 0.03)
  data <- data.frame(V1=x,
                     V2=y)

  d<-data*r
  plot(d)
  set.seed(6)
  circulo <- rbind(data.frame(V1=runif(1000,-a,a),
                              V2=runif(1000,-a,a)),d)
  return(circulo)
}
data <- rbind(genera_circulos(10,8),
```

```
                genera_circulos(2,1.8)-11)
plot(data)

data_with_clusters <- obten_clusters(data,2)
for(m in metodos){
  g <- obten_Grafica(data_with_clusters,m,8)+
    scale_color_manual(values=c("#FF8F00","#54A021"))+
    labs(title="")
  plot(g)
  ggsave(g,filename = paste0("dos_clusters_",m,".png"),
         device = "png",width = 5,height = 5, dpi=150)
}


# Dos circulos uno dentro y otro fuera
theta = runif(1000, 0,2*pi)
x = cos(theta) + rnorm(100, 0, 0.03)
y = sin(theta) + rnorm(100, 0, 0.03)
data <- data.frame(V1=x,
           V2=y)

data <- rbind(data, data*0.5)
plot(data)
data_with_clusters <- obten_clusters(data,2)
for(m in metodos){
  g <- (obten_Grafica(data_with_clusters,m,3)+
         scale_color_manual(values=c("#FF8F00","#54A021")))+
    labs(title="")

  plot(g)
  ggsave(g,filename = paste0("circulos_",m,".png"),
         device = "png",width = 5,height = 5, dpi=150)
}
```

Capítulo 8

Anexo C: Código Tesis

El código utilizado en esta tesis se muestra a continuación. Todos los datos son públicos y se han referenciado las fuentes a lo largo del documento.

El archivo `Comparacion_Variables.xlsx` que fue creado para filtrar las variables seleccionadas se puede encontrar en *https:github.comJaz2608Tesis*. Sin embargo, los resultados se pueden replicar sin éste.

```
pacman::p_load(readr,magrittr,dplyr,tidyr,purrr,tibble,
               ggplot2,lubridate,leaflet,rgdal,sf,rgdal)


# Establece la configuraci n del lenguaje
Sys.setlocale(locale="es_ES.UTF-8")

input <- "C:/Users/amartinez/Documents/Proyecto"
output <- "C:/Users/amartinez/Documents/Proyecto/imagenes"
setwd(input)

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    * *
# 0.0  Se cargan los poligonos de los municipios ======
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    * *

# Marco Geoestad stico. Censo de Poblaci n y Vivienda 2020
# Se descarg  de:
#https://www.inegi.org.mx/app/biblioteca/ficha.html?upc
   =889463807469

# Directorio donde se encuentran los poligonos de los
   municipios
setwd(paste0(input,"/data/poligonos_inegi"))

# Poligonos de todos los municipios del pais
datos_espaciales <- readOGR("00mun.shp",
```

```
                              verbose = FALSE,
                              encoding = "UTF-8")

dim(datos_espaciales@data) # son 2469 municipios

# Se transforman los datos al formato WGS84
datos_espaciales %<>% spTransform(CRS("+proj=longlat +datum=
    WGS84"))

# Se grafica el mapa de la rep blica dividido por municipios
plot(datos_espaciales, main = "Mapa de la rep blica mexicana
      \ndividido por municipios",
      xlab = "Longitud", ylab = "Latitud")


# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    * *
# 0.1  Se cargan  las distancias ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    * *

# Se calculan las matrices D0 y D1 por distancias y por vecinos

# Se ubican los centroides y calculando las distancias
# Se convierte a un objeto espacial
datos_espaciales.sf <- sf::st_as_sf(datos_espaciales)

# Calcula los centroides geogr ficos
Centroids <- st_centroid(datos_espaciales.sf)
# plot(Centroids)
# plot(datos_espaciales.sf)

# Calcula la distancia hacia los crentroides
D1.geo <- st_distance(Centroids) # Matriz de distancias D1
D1.geo <- as.dist(D1.geo)

# Se construye una lista de vecinos basandose en las regiones
    con
# limites contiguos
list.nb <- spdep::poly2nb(datos_espaciales)

# Se contrsuye la matriz de adyacencias
A <- spdep::nb2mat(list.nb, style="B")
# Se pone unos en la diagonal
diag(A) <- 1

D1.dis <- 1-A # Esto se hace porque se esta calculando la
    matriz
```

```
# de distancias
D1.dis <- as.dist(D1.dis)

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 1.1:  Integraci n y limpieza de los datos ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# Se cargan y limpian los datos que sirven para calcular
# las distancias sociodemograficas

pacman::p_load(dplyr,stringr,stringi,readxl,tidyr)
options(scipen=999)

# Se cargan los datos del coneval
# https://www.coneval.org.mx/Medicion/Paginas/Programas_BD_municipal.aspx
setwd(paste0(input,"/data/coneval"))
coneval <- read.csv("indicadores de pobreza municipal 2015.csv",
                    encoding = "latin1")

# Se cargan los datos del inegi a nivel municipal
# https://www.inegi.org.mx/programas/ccpv/2020/#datos_abiertos
setwd(paste0(input,"/data/censo_inegi"))
temp <- list.files(pattern = ".csv")
myfiles <- lapply(temp, read.csv, encoding = "UTF-8")
inegi <- do.call(rbind, myfiles)
rm(temp);rm(myfiles)

# Existen columnas en donde las proporciones para las variables
# del coneval son calculadas
# previamente, se descartan ya que estan redondeadas
coneval <- coneval %>%
  select(-c("pobreza", "pobreza_e","pobreza_m","vul_car",
            "vul_ing","npnv","ic_rezedu","ic_asalud",
            "ic_segsoc","ic_cv","ic_sbv","ic_ali",
            "carencias","carencias3","plb","plbm"))

# El nombre de la columna X.U.FEFF.ENTIDAD en la base del inegi
# se reemplaza por ENTIDAD
names(inegi)[which(names(inegi)=="X.U.FEFF.ENTIDAD")] = "ENTIDAD"

# Se conserva  nicamente  el total por municipio en la base del inegi,
# y se descartan variables que no son de inter s
inegi <- inegi%>% filter(NOM_LOC=='Total del Municipio')%>%
```

```
  select(-c("LOC", "NOM_LOC" ,
            "LONGITUD", "LATITUD", "ALTITUD") )

# En el INEGI y el coneval las claves de los municipios
# son distintas, as  que esas no sirven para hacer un join
summary(coneval$MUN)
summary(inegi$MUN)


# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 1.2:  Cambios de los nombres en municipios y estados ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
unique(inegi$NOM_ENT)
unique(coneval$entidad_federativa)

# Se reemplazan los nombres de los estados
inegi<-inegi%>%
  mutate(NOM_ENT=case_when(
    NOM_ENT=='Michoac n de Ocampo' ~ 'Michoac n',
    NOM_ENT=='Veracruz de Ignacio de la Llave' ~ 'Veracruz',
    NOM_ENT=='Coahuila de Zaragoza' ~ 'Coahuila',
    TRUE ~ NOM_ENT))

coneval<- coneval%>%
  mutate(entidad_federativa=case_when(
    entidad_federativa=='Distrito Federal'~'Ciudad de M xico',
    TRUE ~ entidad_federativa))%>%
  rename(NOM_ENT=entidad_federativa,
         NOM_MUN=municipio,
         ENTIDAD=clave_entidad,
         MUN=clave_municipio)

# Con esto se ve si las combinaciones entre los nombres de los
# estados y municipios son  nicos  para poderlos usar
# como llave primaria

inegi%>%
  group_by(NOM_ENT,NOM_MUN)%>%
  summarise(total=n())%>%
  filter(total>1)

coneval%>%
  group_by(NOM_ENT,NOM_MUN)%>%
  summarise(total=n())%>%
  filter(total>1)

```

```
156 # Con el coneval s  se puede hacer una llave unica uniendo
157 # los nombres, pero para el inegi se tienen que arreglar
158
159
160 # "San Juan Mixtepec"
161 inegi%>%
162   filter(NOM_MUN=="San Juan Mixtepec")
163 coneval %>%
164   filter(str_detect(NOM_MUN, "San Juan Mixtepec"))
165
166 # "San Pedro Mixtepec"
167 inegi%>%
168   filter(NOM_MUN=="San Pedro Mixtepec")
169 coneval %>%
170   filter(str_detect(NOM_MUN, "San Pedro Mixtepec"))
171
172 # Se puede inferir por su poblaci n cu l es cu l
173 inegi<-inegi%>%
174   mutate(NOM_MUN=case_when(
175     MUN=='208' & NOM_ENT== 'Oaxaca'~ 'San Juan Mixtepec -Dto.
          08 -',
176     MUN=='209' & NOM_ENT== 'Oaxaca'~ 'San Juan Mixtepec -Dto.
          26 -',
177     MUN=='318' & NOM_ENT== 'Oaxaca'~ 'San Pedro Mixtepec -Dto.
          22 -',
178     MUN=='319' & NOM_ENT== 'Oaxaca'~ 'San Pedro Mixtepec -Dto.
          26 -',
179     TRUE ~ NOM_MUN)
180   )
181
182 # Se quitan las claves del coneval ya que ahora no ser n
     necesarias
183 coneval <- coneval%>% select(-c("ENTIDAD","MUN"))
184
185 # Ahora se pueden unir las bases de datos
186 interseccion_inegi_coneval <- merge(coneval, inegi,
187                                     by=c("NOM_ENT","NOM_MUN"))
188 # notemos que:
189 #   en el inegi hay 2469 renglones
190 #   en el coneval hay 2457 renglones
191 #   en la intersecci n hay 2433 renglones
192
193
194 # Vemos aquellos que est n en el coneval pero no en la
     interseccion
195 dif_coneval <- setdiff(coneval%>%select(NOM_ENT, NOM_MUN),
196                        interseccion_inegi_coneval%>%
197                          select(NOM_ENT, NOM_MUN))%>%
```

```
198   arrange((NOM_MUN))
199
200 # se tienen 24 registros que estan en el coneval
201 # pero no en la interseccion
202
203 # Vemos aquellos que est n en el inegi pero no en la
      interseccion
204 dif_inegi <-setdiff(inegi%>%select(NOM_ENT, NOM_MUN),
205                     interseccion_inegi_coneval%>%
206                       select(NOM_ENT, NOM_MUN))%>%
207                      arrange((NOM_MUN))
208 # se tienen 36 registros que estan en el inegi
209 # pero no en la intersecci n
210
211
212 # Se tienen que cambiar aquellos que sean los mismos pero el
      nombre
213 coneval <- coneval %>%
214   mutate(NOM_MUN=
215            case_when(
216              NOM_MUN=='Acambay' & NOM_ENT=='M xico' ~
217                'Acambay de Ru z Casta eda',
218              NOM_MUN=='Batopilas' & NOM_ENT=='Chihuahua' ~
219                'Batopilas de Manuel G mez Mor n',
220              NOM_MUN=='Dr. Arroyo'& NOM_ENT=='Nuevo Le n' ~
221                'Doctor Arroyo',
222              NOM_MUN=='Dr. Coss'& NOM_ENT=='Nuevo Le n' ~
223                'Doctor Coss',
224              NOM_MUN=='Dr. Gonz lez'& NOM_ENT=='Nuevo Le n' ~
225                'Doctor Gonz lez',
226              NOM_MUN=='Carmen'& NOM_ENT=='Nuevo Le n' ~
227                'El Carmen',
228              NOM_MUN=='Gral. Bravo'& NOM_ENT=='Nuevo Le n' ~
229                'General Bravo',
230              NOM_MUN=='Gral. Escobedo'& NOM_ENT=='Nuevo Le n'
                   ~
231                'General Escobedo',
232              NOM_MUN=='Gral. Ter n'& NOM_ENT=='Nuevo Le n' ~
233                'General Ter n',
234              NOM_MUN=='Gral. Trevi o'& NOM_ENT=='Nuevo Le n'
                   ~
235                'General Trevi o',
236              NOM_MUN=='Gral. Zaragoza'& NOM_ENT=='Nuevo Le n'
                   ~
237                'General Zaragoza',
238              NOM_MUN=='Gral. Zuazua'& NOM_ENT=='Nuevo Le n' ~
239                'General Zuazua',
240              NOM_MUN=='Heroica Ciudad de Juchit n de Zaragoza'
```

```
                &
241               NOM_ENT=='Oaxaca' ~
242               'Juchit n de Zaragoza',
243             NOM_MUN=='Jonacatepec'& NOM_ENT=='Morelos'  ~
244               'Jonacatepec de Leandro Valle',
245             NOM_MUN=='Jos  Joaquin de Herrera'&
246               NOM_ENT=='Guerrero' ~
247               'Jos  Joaqu n de Herrera',
248             NOM_MUN=='Medell n' & NOM_ENT=='Veracruz'~
249               'Medell n de Bravo',
250             NOM_MUN=='San Mateo Yucutind ' & NOM_ENT=='Oaxaca
                  '~
251               'San Mateo Yucutindoo',
252             NOM_MUN=='Santiago Chazumba'& NOM_ENT=='Oaxaca' ~
253               'Villa de Santiago Chazumba',
254             NOM_MUN=='Silao' & NOM_ENT=='Guanajuato' ~
255               'Silao de la Victoria',
256             NOM_MUN=='Tlaltizap n'& NOM_ENT=='Morelos' ~
257               'Tlaltizap n de Zapata',
258             NOM_MUN=='Tezoatl n de Segura y Luna'&
259               NOM_ENT=='Oaxaca' ~
260               'Heroica Villa Tezoatlan De Segura Y Luna, Cuna
                     De La Independencia De Oaxaca',
261             NOM_MUN=='Tlaltizap n'& NOM_ENT=='Morelos' ~
262               'Tlaltizap n de Zapata',
263             NOM_MUN=='Tlaquepaque'& NOM_ENT=='Jalisco'  ~
264               'San Pedro Tlaquepaque',
265             NOM_MUN=='Villa de Tututepec de Melchor Ocampo'&
                   NOM_ENT=='Oaxaca' ~
266               'Villa de Tututepec',
267             NOM_MUN=='Zacualpan'& NOM_ENT=='Morelos' ~
268               'Zacualpan de Amilpas',
269             TRUE ~ NOM_MUN)
270   )
271
272 inegi<- inegi%>%
273   mutate(NOM_MUN = case_when(NOM_MUN == "Heroica Villa
          Tezoatl n de Segura y Luna, Cuna de"  ~
274     'Heroica Villa Tezoatlan De Segura Y Luna, Cuna De La
          Independencia De Oaxaca',
275      TRUE ~ NOM_MUN))
276
277 # Si se vuelven a correr las intersecciones en el coneval hay 0
278 interseccion_inegi_coneval <- merge(coneval, inegi,
279                                     by=c("NOM_ENT","NOM_MUN"))
280 dif_coneval <- setdiff(coneval%>%select(NOM_ENT, NOM_MUN),
281                        interseccion_inegi_coneval%>%
282                          select(NOM_ENT, NOM_MUN))%>%
```

```
283   arrange((NOM_MUN))
284 dif_inegi <-setdiff(inegi%>%select(NOM_ENT, NOM_MUN),
285                     interseccion_inegi_coneval%>%
286                       select(NOM_ENT, NOM_MUN))%>%
287   arrange((NOM_MUN))
288
289
290 # Existen 12 municipios nuevos, que aparecen en la encuesta del
      inegi 2020
291 # pero no aparecen en la encuesta del coneval del 2015
292
293 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      * *
294 # 2.0: Join entre coneval e inegi====
295 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      * *
296 inegi_y_coneval <- merge(inegi, coneval,
297                          by=c("NOM_ENT","NOM_MUN"),
298                          all.x=TRUE)
299
300 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      * *
301 # 2.1: Transformar las variables del INEGI a proporciones ====
302 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      * *
303
304
305 # Los datos del inegi van desde POBTOT hasta TAMLOC, saldr n
     warnings porque TAMLOC no es n merica
306 subdata_inegi <- inegi[which(colnames(inegi_y_coneval)=="POBTOT
    "):
307                          which(colnames(inegi_y_coneval)=="
                               TAMLOC")]
308 # Se convierten los datos a numericos
309 subdata_inegi <- as.data.frame(sapply(subdata_inegi, as.numeric
    ))
310
311
312 # Se usa un archivo auxiliar para calcular las tasas del inegi,
313 # en este archivo se define entre qu  variable se divide cada
     una
314 setwd(input)
315 para_tasas_inegi <- read_excel("Comparacion_Variables.xlsx")%>%
316   select(Variables,dividir_entre)%>%
317   drop_na()
318 names(para_tasas_inegi) <- c("variable","division")
319
320 # Se realiza la conversion
```

```
for(v in 1:dim(para_tasas_inegi)[1]){
  variable <- para_tasas_inegi$variable[v]
  division <- para_tasas_inegi$division[v]
  if(division=="no se divide"){
    subdata_inegi[variable]<-subdata_inegi[variable]
  }else{
    subdata_inegi[variable]<-subdata_inegi[variable]/
      subdata_inegi[division]
  }
};rm(v)



# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 2.2: Transformar las variables del CONEVAL a proporciones ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

# Las variables del coneval se divien entre 'poblacion', a partir de la variable
# pobreza_pob y hasta plbm_pob, se agrega tambi n el estado y municipio
# El coneval tiene 12 na'as y 11 "n.d", en estos renglones no se pueden
# hacer operaciones
subdata_coneval <- inegi_y_coneval[
              which(colnames(inegi_y_coneval)=="poblacion"):
              which(colnames(inegi_y_coneval)=="plbm_pob")]

subdata_coneval <- as.data.frame(sapply(subdata_coneval, as.numeric))


# se calculan las tasas desde pobreza_pob hasta la  ltima columna
for (i in names(subdata_coneval)[-1]){
  subdata_coneval[i] <-  (subdata_coneval[i])/
    (subdata_coneval["poblacion"])
}
rm(i) # Salen warnings por los NA's de los estados que no tienen info para el coneval


# Se agrega la poblaci n total, el estado y municipio a las dos subdatas
# y al del inegi tambi n se le agrega las claves
```

```r
subdata_coneval <- cbind(inegi_y_coneval[
            which(colnames(inegi_y_coneval)=="NOM_MUN")],
              subdata_coneval)
subdata_coneval <- cbind(inegi_y_coneval[
            which(colnames(inegi_y_coneval)=="NOM_ENT")],
             subdata_coneval)

subdata_inegi <- cbind(inegi[which(colnames(inegi)=="MUN")],
                       subdata_inegi)
subdata_inegi <- cbind(inegi[which(colnames(inegi)=="ENTIDAD")
   ],
                       subdata_inegi)
subdata_inegi <- cbind(inegi[which(colnames(inegi)=="NOM_MUN")
   ],
                       subdata_inegi)
subdata_inegi <- cbind(inegi[which(colnames(inegi)=="NOM_ENT")
   ],
                       subdata_inegi)



# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    * *
# 3.1: Imputar los datos faltantes ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    * *


# Se ve si hay datos faltantes
which(apply(subdata_inegi, 2, function(x) any(is.na(x)))==TRUE)
# Hay datos faltantes en TAMLOC pero esa se quitar

which(apply(subdata_coneval, 2, function(x) any(is.na(x)))==
   TRUE)
length(which(apply(subdata_coneval, 2, function(x) any(is.na(x)
   ))==TRUE))
# Existen 17 datos faltantes para todas las columnas en los
   datos del coneval

# Estos son:
faltantes <- which((rowSums(is.na(subdata_coneval)) > 0) ==
   TRUE) #162 en total
faltantes
length(faltantes) #23 datos faltantes en la base del coneval

# Se van a modificar las variables del coneval, es decir, desde
    pobreza_pob
# hasta plbm_pob
```

```r
i <- which(names(subdata_coneval)=="pobreza_pob")
j <- which(names(subdata_coneval)=="plbm_pob")

# Son 12 los municipios que surgieron entre 2015 y 2020 por lo que
# aparecen en la base de datos del inegi pero no del coneval
# Esta informaci n se obtuvo de AGEEML_2021682212978.csv


# * * * * * * * * * * * * * * * * * * * * * * San Quint n - Baja California
# Proviene de: (1)
# NOM_MUN: Ensenada
# NOM_ENT: Baja California

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Baja California' &
                      subdata_coneval['NOM_MUN'] == 'San Quint n'),i:j] <- subdata_coneval[
                      which(subdata_coneval['NOM_ENT'] == 'Baja California' &
                      subdata_coneval['NOM_MUN'] == 'Ensenada'), i:j]

# * * * * * * * * * * * * * * * * * * * * * * Seybaplaya - Campeche
# Proviene de: (2)
# NOM_MUN: Champot n
# NOM_ENT: Campeche

imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Campeche' &
           subdata_coneval['NOM_MUN'] == 'Champot n'),i:j]

# NOM_MUN: Campeche
# NOM_ENT: Campeche
imputar <- rbind(imputar, subdata_coneval[
  which(subdata_coneval['NOM_ENT'] == 'Campeche' &
  subdata_coneval['NOM_MUN'] == 'Campeche'),i:j])


imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))


subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Campeche' &
  subdata_coneval['NOM_MUN'] == 'Seybaplaya'),i:j] <- imputar
```

```
436
437 rm(imputar)
438
439
440 # * * * * * * * * * * * * * * * * * * Capit n Luis  ngel  Vidal -
      Chiapas
441 # Proviene de: (1)
442 # NOM_MUN: Siltepec
443 # NOM_ENT: Chiapas
444
445 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
446                 subdata_coneval['NOM_MUN'] == 'Capit n Luis
                        ngel  Vidal'), i:j] <- subdata_coneval[
447                 which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
448                 subdata_coneval['NOM_MUN'] == 'Siltepec'),i:j]
449
450
451 # * * * * * * * * * * * * * * * * * * Rinc n Chamula San Pedro -
      Chiapas
452 # Proviene de: (4)
453
454 # NOM_MUN: Jitotol
455 # NOM_ENT: Chiapas
456
457 imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
      'Chiapas' &
458           subdata_coneval['NOM_MUN'] == 'Jitotol'),i:j]
459
460 # NOM_MUN: Ray n
461 # NOM_ENT: Chiapas
462
463 imputar <- rbind(imputar, subdata_coneval[which(subdata_coneval
      ['NOM_ENT'] == 'Chiapas' &
464           subdata_coneval['NOM_MUN'] == 'Ray n'),i:j])
465
466 # NOM_MUN: Tapilula
467 # NOM_ENT: Chiapas
468
469 imputar <- rbind(imputar, subdata_coneval[which(subdata_coneval
      ['NOM_ENT'] == 'Chiapas' &
470           subdata_coneval['NOM_MUN'] == 'Tapilula'),i:j])
471
472
473 # NOM_MUN: Pueblo Nuevo Solistahuac n
474 # NOM_ENT: Chiapas
475
476 imputar <- rbind(imputar, subdata_coneval[which(subdata_coneval
      ['NOM_ENT'] == 'Chiapas' &
```

```
            subdata_coneval['NOM_MUN'] == 'Pueblo Nuevo
               Solistahuac n'),i:j])

imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))


subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
             subdata_coneval['NOM_MUN'] == 'Rinc n Chamula San
                Pedro'),i:j] <- imputar

rm(imputar)

# * * * * * * * * * * * * * * * * * * * * El Parral - Chiapas
# Proviene de: (2)

# NOM_MUN: Villaflores
# NOM_ENT: Chiapas

imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
   'Chiapas' &
            subdata_coneval['NOM_MUN'] == 'Villaflores'),i:j]

# NOM_MUN: Villa Corzo
# NOM_ENT: Chiapas

imputar <- rbind(imputar, subdata_coneval[which(subdata_coneval
   ['NOM_ENT'] == 'Chiapas' &
            subdata_coneval['NOM_MUN'] == 'Villa Corzo'),i:j])


imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))


subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
                        subdata_coneval['NOM_MUN'] == 'El
                           Parral'),i:j] <- imputar

rm(imputar)


# * * * * * * * * * * * * * * * * * * * * Emiliano Zapata -
   Chiapas
# Proviene de: (3)

# NOM_MUN: Venustiano Carranza
# NOM_ENT: Chiapas
```

```
519
520 imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
      'Chiapas' &
521             subdata_coneval['NOM_MUN'] == 'Venustiano Carranza')
                ,i:j]
522
523 # NOM_MUN: Chiapa de Corzo
524 # NOM_ENT: Chiapas
525
526 imputar <- rbind(imputar, subdata_coneval[
527           which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
528           subdata_coneval['NOM_MUN'] == 'Chiapa de Corzo'),i:j
              ])
529
530 # NOM_MUN: Acala
531 # NOM_ENT: Chiapas
532
533 imputar <- rbind(imputar, subdata_coneval[which(subdata_coneval
      [
534             'NOM_ENT'] == 'Chiapas' &
535             subdata_coneval['NOM_MUN'] == 'Acala'),i:j])
536
537 imputar <- sapply(imputar, as.numeric)
538 imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))
539
540
541 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
542              subdata_coneval['NOM_MUN'] == 'Emiliano Zapata'),
543              i:j] <- imputar
544
545 rm(imputar)
546
547 # * * * * * * * * * * * * * * * * * * * * * Mezcalapa - Chiapas
548 # Proviene de: (3)
549
550 # NOM_MUN: Ocozocoautla de Espinosa
551 # NOM_ENT: Chiapas
552
553 imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
      'Chiapas' &
554             subdata_coneval['NOM_MUN'] == 'Ocozocoautla de
                Espinosa'),i:j]
555
556
557 # NOM_MUN: Tecpat n
558 # NOM_ENT: Chiapas
559
560 imputar <- rbind(imputar, subdata_coneval[
```

```
561             which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
562             subdata_coneval['NOM_MUN'] == 'Tecpat n'),i:j])
563
564 # NOM_MUN: Ostuac n
565 # NOM_ENT: Chiapas
566
567 imputar <- rbind(imputar, subdata_coneval[
568             which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
569             subdata_coneval['NOM_MUN'] == 'Ostuac n'),i:j])
570
571 imputar <- sapply(imputar, as.numeric)
572 imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))
573
574
575 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
576             subdata_coneval['NOM_MUN'] == 'Mezcalapa'),i:j] <-
                 imputar
577
578 rm(imputar)
579
580
581 # * * * * * * * * * * * * * * * * * * Honduras de la Sierra -
       Chiapas
582 # Proviene de: (1)
583
584 # NOM_MUN: Siltepec
585 # NOM_ENT: Chiapas
586
587 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
588                 subdata_coneval['NOM_MUN'] == 'Honduras de la
                     Sierra'), i:j] <- subdata_coneval[
589                 which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
590                 subdata_coneval['NOM_MUN'] == 'Siltepec'),i:j]
591
592
593 # * * * * * * * * * * * * * * * * * * * * Coatetelco - Morelos
594 # Proviene de: (1)
595
596 # NOM_MUN: Miacatl n
597 # NOM_ENT: Morelos
598
599 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Morelos' &
600                 subdata_coneval['NOM_MUN'] == 'Coatetelco'),i:j
                     ] <- subdata_coneval[
601                 which(subdata_coneval['NOM_ENT'] == 'Morelos' &
602                 subdata_coneval['NOM_MUN'] == 'Miacatl n'),i:j
                     ]
603
```

```
# * * * * * * * * * * * * * * * * * * * Xoxocotla - Morelos
# Proviene de: (1)

# NOM_MUN: Puente de Ixtla
# NOM_ENT: Morelos

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Morelos' &
                subdata_coneval['NOM_MUN'] == 'Xoxocotla'),i:j]
                    <- subdata_coneval[
                which(subdata_coneval['NOM_ENT'] == 'Morelos' &
                subdata_coneval['NOM_MUN'] == 'Puente de Ixtla'
                   ),i:j]


# * * * * * * * * * * * * * * * * * * * Hueyapan - Morelos
# Proviene de: (1)

# NOM_MUN: Tetela del Volc n
# NOM_ENT: Morelos

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Morelos' &
                subdata_coneval['NOM_MUN'] == 'Hueyapan'),i:j]
                   <- subdata_coneval[
                which(subdata_coneval['NOM_ENT'] == 'Morelos' &
                subdata_coneval['NOM_MUN'] == 'Tetela del
                   Volc n'),i:j
                        ]

# * * * * * * * * * * * * * * * * * * * Puerto Morelos - Quintana
    Roo
# Proviene de: (1)

# NOM_MUN: Benito Ju rez
# NOM_ENT: Quintana Roo

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Quintana
   Roo' &
                subdata_coneval['NOM_MUN'] == 'Puerto Morelos')
                   , i:j] <- subdata_coneval[
                which(subdata_coneval['NOM_ENT'] == 'Quintana
                   Roo' &
                subdata_coneval['NOM_MUN'] == 'Benito Ju rez')
                   ,i:j]

rm(i);rm(j)
```

```
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# * * * * * * * Otros municipios que no tienen datos en el coneval * * * * * * *
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

faltantes <- which(is.na(subdata_coneval$vul_car_pob)==TRUE)
length(faltantes)

subdata_coneval[faltantes,c("NOM_ENT","NOM_MUN")]
i <- which(names(subdata_coneval)=="pobreza_pob")
j <- which(names(subdata_coneval)=="plbm_pob")

subdata_coneval[faltantes,c("NOM_ENT","NOM_MUN")]


# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * Buenaventura - Chihuahua
# Vecinos: (7) LISTO ***
#   Ascensi n - Chihuahua
#   Nuevo Casas Grandes - Chihuahua
#   Galeana - Chihuahua
#   Ignacio Zaragoza - Chihuahua
#   Namiquipa - Chihuahua
#   Chihuahua - Chihuahua
#   Ahumada - Chihuahua


#   Ascensi n - Chihuahua
imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
                           subdata_coneval['NOM_MUN'] == 'Ascensi n'),i:j]
#   Nuevo Casas Grandes - Chihuahua
imputar <- rbind(imputar, subdata_coneval[
           which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
           subdata_coneval['NOM_MUN'] == 'Nuevo Casas Grandes'),i:j])
#   Galeana - Chihuahua
imputar <- rbind(imputar, subdata_coneval[
           which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
           subdata_coneval['NOM_MUN'] == 'Galeana'),i:j])
#   Ignacio Zaragoza - Chihuahua
imputar <- rbind(imputar, subdata_coneval[
           which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
           subdata_coneval['NOM_MUN'] == 'Ignacio Zaragoza'),i:j])
```

```
683 #   Namiquipa - Chihuahua
684 imputar <- rbind(imputar, subdata_coneval[
685             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
686             subdata_coneval['NOM_MUN'] == 'Namiquipa'),i:j])
687 #   Chihuahua - Chihuahua
688 imputar <- rbind(imputar, subdata_coneval[
689             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
690             subdata_coneval['NOM_MUN'] == 'Chihuahua'),i:j])
691 #   Ahumada - Chihuahua
692 imputar <- rbind(imputar, subdata_coneval[which(subdata_coneval
        ['NOM_ENT'] == 'Chihuahua' &
693             subdata_coneval['NOM_MUN'] == 'Ahumada'),i:j])
694
695 imputar <- sapply(imputar, as.numeric)
696 imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))
697
698
699 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chihuahua'
        &
700                 subdata_coneval['NOM_MUN'] == 'Buenaventura'),i
                        :j] <- imputar
701
702 rm(imputar)
703
704 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        Carich  - Chihuahua
705 # Vecinos: (6) LISTO ***
706 #   Guerrero - Chihuahua
707 #   Bocoyna - Chihuahua
708 #   Guachochi - Chihuahua
709 #   Nonoava - Chihuahua
710 #   San Francisco de Borja - Chihuahua
711 #   Cusihuiriachi - Chihuahua
712
713 #   Guerrero - Chihuahua
714 imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
        'Chihuahua' &
715             subdata_coneval['NOM_MUN'] == 'Guerrero'),i:j]
716 #   Bocoyna - Chihuahua
717 imputar <- rbind(imputar, subdata_coneval[
718             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
719             subdata_coneval['NOM_MUN'] == 'Bocoyna'),i:j])
720 #   Guachochi - Chihuahua
721 imputar <- rbind(imputar, subdata_coneval[
722             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
723             subdata_coneval['NOM_MUN'] == 'Guachochi'),i:j])
724 #   Nonoava - Chihuahua
725 imputar <- rbind(imputar, subdata_coneval[
```

```
726             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
727             subdata_coneval['NOM_MUN'] == 'Nonoava'),i:j])
728 #   San Francisco de Borja - Chihuahua
729 imputar <- rbind(imputar, subdata_coneval[
730             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
731             subdata_coneval['NOM_MUN'] == 'San Francisco de
                    Borja'),i:j])
732 #   Cusihuiriachi - Chihuahua
733 imputar <- rbind(imputar, subdata_coneval[which(subdata_coneval
    ['NOM_ENT'] == 'Chihuahua' &
734             subdata_coneval['NOM_MUN'] == 'Cusihuiriachi'),i:j])
735
736 imputar <- sapply(imputar, as.numeric)
737 imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))
738
739 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chihuahua'
    &
740                       subdata_coneval['NOM_MUN'] == 'Carich
                          '),i:j] <- imputar
741
742 rm(imputar)
743
744
745 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    Santa Isabel - Chihuahua
746 # Vecinos: (6) LISTO ***
747 #   Chihuahua - Chihuahua
748 #   Riva Palacio - Chihuahua
749 #   Cuauht moc - Chihuahua
750 #   Gran Morelos - Chihuahua
751 #   Dr. Belisario Dom nguez - Chihuahua
752 #   Satev  - chihuahua
753
754
755 #   Chihuahua - Chihuahua
756 imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
    'Chihuahua' &
757             subdata_coneval['NOM_MUN'] == 'Chihuahua'),i:j]
758 #   Riva Palacio - Chihuahua
759 imputar <- rbind(imputar, subdata_coneval[
760             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
761             subdata_coneval['NOM_MUN'] == 'Riva Palacio'),i:j])
762 #   Cuauht moc - Chihuahua
763 imputar <- rbind(imputar, subdata_coneval[
764             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
765             subdata_coneval['NOM_MUN'] == 'Cuauht moc'),i:j])
766 #   Gran Morelos - Chihuahua
767 imputar <- rbind(imputar, subdata_coneval[
```

```
768             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
769             subdata_coneval['NOM_MUN'] == 'Gran Morelos'),i:j])
770 #   Dr. Belisario Dom nguez - Chihuahua
771 imputar <- rbind(imputar, subdata_coneval[
772             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
773             subdata_coneval['NOM_MUN'] == 'Dr. Belisario
                    Dom nguez'),i:j])
774 #   Satev  - chihuahua
775 imputar <- rbind(imputar, subdata_coneval[
776             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
777             subdata_coneval['NOM_MUN'] == 'Satev '),i:j])
778
779 imputar <- sapply(imputar, as.numeric)
780 imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))
781
782 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chihuahua'
    &
783                 subdata_coneval['NOM_MUN'] == 'Santa Isabel'),i:
                       j] <- imputar
784
785 rm(imputar)
786
787
788 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    Tem sachic - Chihuahua
789 # Vecinos: (9) LISTO ***
790 #   Madera - Chihuahua
791 #   Moris - Chihuahua
792 #   Ocampo - Chihuahua
793 #   Guerrero - Chihuahua
794 #   Matach  - Chihuahua
795 #   Namiquipa - Chihuahua
796 #   G mez Far as - Chihuahua
797 #   Sahuaripa - Sonora - 052
798 #   Y cora - Sonora - 069
799
800 #   Madera - Chihuahua 1
801 imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
    'Chihuahua' &
802             subdata_coneval['NOM_MUN'] == 'Madera'),i:j]
803 #   Moris - Chihuahua 2
804 imputar <- rbind(imputar, subdata_coneval[
805              which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
806              subdata_coneval['NOM_MUN'] == 'Moris'),i:j])
807 #   Ocampo - Chihuahua 3
808 imputar <- rbind(imputar, subdata_coneval[
809             which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
810             subdata_coneval['NOM_MUN'] == 'Ocampo'),i:j])
```

```
#   Guerrero - Chihuahua 4
imputar <- rbind(imputar, subdata_coneval[
           which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
           subdata_coneval['NOM_MUN'] == 'Guerrero'),i:j])
#   Matach  - Chihuahua 5
imputar <- rbind(imputar, subdata_coneval[
           which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
           subdata_coneval['NOM_MUN'] == 'Matach '),i:j])
#   Namiquipa - Chihuahua 6
imputar <- rbind(imputar, subdata_coneval[
           which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
           subdata_coneval['NOM_MUN'] == 'Namiquipa'),i:j])
#   G mez Far as - Chihuahua 7
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
            subdata_coneval['NOM_MUN'] == 'G mez Far as'),i:j
               ])
#   Sahuaripa - Sonora 8
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Sonora' &
            subdata_coneval['NOM_MUN'] == 'Sahuaripa'),i:j])
#   Y cora - Sonora 9
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Sonora' &
            subdata_coneval['NOM_MUN'] == 'Y cora'),i:j])

imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chihuahua'
    &
                subdata_coneval['NOM_MUN'] == 'Tem sachic'),i:
                   j] <- imputar

rm(imputar)

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    Urique - Chihuahua
# Vecinos:(6) LISTO ***
#   Guazapares - Chihuahua
#   Batopilas de Manuel G mez Mor n - Chihuahua
#   Guachochi - Chihuahua
#   Bocoyna - Chihuahua
#   Maguarichi - Chihuahua
#   Choix - Sinaloa


#   Guazapares - Chihuahua
```

```
imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
            subdata_coneval['NOM_MUN'] == 'Guazapares'),i:j]
#   Batopilas de Manuel G mez Mor n - Chihuahua
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
            subdata_coneval['NOM_MUN'] == 'Batopilas de Manuel G mez Mor n'),i:j])
#   Guachochi - Chihuahua
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
            subdata_coneval['NOM_MUN'] == 'Guachochi'),i:j])
#   Bocoyna - Chihuahua
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
            subdata_coneval['NOM_MUN'] == 'Bocoyna'),i:j])
#   Maguarichi - Chihuahua
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
            subdata_coneval['NOM_MUN'] == 'Maguarichi'),i:j])
#   Choix - Sinaloa
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Sinaloa' &
            subdata_coneval['NOM_MUN'] == 'Choix'),i:j])

imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Chihuahua' &
                subdata_coneval['NOM_MUN'] == 'Urique'),i:j] <-
                    imputar

rm(imputar)

# * * * * * * * * * * * * * * * * * * * * * * * *  Mat as Romero Avenda o - Oaxaca
# Vecinos:(7) LISTO*** , en realidad 5 porque 427 no tiene info , ni el 407 *
#   091 - Veracruz - Jes s Carranza
#   190 - Oaxaca - San Juan Cotzoc n
#   207 - Oaxaca -  San Juan Mazatl n
#   198 - Oaxaca -  San Juan Guichicovi
#   427 - Oaxaca - Santa Mar a Petapa *
#   010 - Oaxaca - El Barrio de la Soledad
#   407 - Oaxaca - Santa Mar a Chimalapa *


```

```
#   091 - Veracruz - Jes s Carranza 1
imputar <- subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Veracruz' &
            subdata_coneval['NOM_MUN'] == 'Jes s Carranza'),i:
               j]
#   190 - Oaxaca - San Juan Cotzoc n 2
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Juan Cotzoc n')
               ,i:j])
#   207 - Oaxaca -  San Juan Mazatl n 3
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Juan Mazatl n')
               ,i:j])
#   198 - Oaxaca -  San Juan Guichicovi 4
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Juan Guichicovi'
               ),i:j])

#   427 - Oaxaca - Santa Mar a Petapa  (no tiene datos para
   imputar as  que se descarta)
#imputar <- rbind(imputar, subdata_coneval[which(subdata_
   coneval['NOM_ENT'] == 'Oaxaca' &
#                                          subdata_coneval['
   NOM_MUN'] == 'Santa Mar a Petapa'),i:j])

#   010 - Oaxaca - El Barrio de la Soledad 5
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'El Barrio de la
               Soledad'),i:j])

#   407 - Oaxaca - Santa Mar a Chimalapa, tampoco tiene datos
#imputar <- rbind(imputar, subdata_coneval[which(subdata_
   coneval['NOM_ENT'] == 'Oaxaca' &
#                                          subdata_coneval['
   NOM_MUN'] == 'Santa Mar a Chimalapa'),i:j])


imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
                subdata_coneval['NOM_MUN'] == 'Mat as Romero
                   Avenda o'),i:j] <- imputar

```

```r
rm(imputar)

# * * * * * * * * * * * * * * * * * * * * * * *  San Francisco Chind a - Oaxaca
# Vecinos: (7) LISTO ***
#   332 - Oaxaca -  San Pedro Topiltepec
#   479 - Oaxaca -  Santiago Nejapilla
#   518 - Oaxaca - Santo Domingo Tlatay pam
#   147 - Oaxaca - San Francisco Nuxa o
#   281 - Oaxaca -  San Miguel Tecomatl n
#   250 - Oaxaca - San Mateo Etlatongo
#   215 - Oaxaca - San Juan Sayultepec


#   332 - Oaxaca -  San Pedro Topiltepec 1
imputar <- subdata_coneval[
           which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
           subdata_coneval['NOM_MUN'] == 'San Pedro Topiltepec'),i:j]
#   479 - Oaxaca -  Santiago Nejapilla 2
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'Santiago Nejapilla'),i:j])
#   518 - Oaxaca - Santo Domingo Tlatay pam   3
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'Santo Domingo Tlatay pam'),i:j])
#   147 - Oaxaca - San Francisco Nuxa o 4
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Francisco Nuxa o'),i:j])
#   281 - Oaxaca -  San Miguel Tecomatl n 5
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Miguel Tecomatl n'),i:j])
#   250 - Oaxaca - San Mateo Etlatongo 6
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Mateo Etlatongo'),i:j])
#   215 - Oaxaca - San Juan Sayultepec 7
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Juan Sayultepec'
```

```
                ),i:j])
975
976 imputar <- sapply(imputar, as.numeric)
977 imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))
978
979 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
980             subdata_coneval['NOM_MUN'] == 'San Francisco
                Chind a'),i:j] <- imputar
981
982 rm(imputar)
983 # * * * * * * * * * * * * * * * * * * * * * * * * * * *  Santa Mar a
    Chimalapa - Oaxaca
984 # Vecinos: (8) LISTO *
985 #   057 - Oaxaca -  Mat as Romero Avenda o
986 #   010 - Oaxaca -  El Barrio de la Soledad
987 #   005 - Oaxaca -  Asunci n Ixtaltepec
988 #   265 - Oaxaca -  San Miguel Chimalapa
989 #   017 - Chiapas - Cintalapa
990 #   061 - Veracruz -  Las Choapas
991 #   210 - Veracruz - Uxpanapa
992 #   091 - Veracruz - Jes s Carranza
993
994
995 #   057 - Oaxaca -  Mat as Romero Avenda o 1
996 imputar <- subdata_coneval[
997             which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
998             subdata_coneval['NOM_MUN'] == 'Mat as Romero
                Avenda o'),i:j]
999 #   010 - Oaxaca -  El Barrio de la Soledad 2
1000 imputar <- rbind(imputar, subdata_coneval[
1001             which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
1002             subdata_coneval['NOM_MUN'] == 'El Barrio de la
                Soledad'),i:j])
1003 #   005 - Oaxaca -  Asunci n Ixtaltepec 3
1004 imputar <- rbind(imputar, subdata_coneval[
1005             which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
1006             subdata_coneval['NOM_MUN'] == 'Asunci n Ixtaltepec
                '),i:j])
1007 #   265 - Oaxaca -  San Miguel Chimalapa 4
1008 imputar <- rbind(imputar, subdata_coneval[
1009             which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
1010             subdata_coneval['NOM_MUN'] == 'San Miguel Chimalapa
                '),i:j])
1011 #   017 - Chiapas - Cintalapa 5
1012 imputar <- rbind(imputar, subdata_coneval[
1013             which(subdata_coneval['NOM_ENT'] == 'Chiapas' &
1014             subdata_coneval['NOM_MUN'] == 'Cintalapa'),i:j])
1015 #   061 - Veracruz -  Las Choapas 6
```

```
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Veracruz' &
            subdata_coneval['NOM_MUN'] == 'Las Choapas'),i:j])
#   210 - Veracruz - Uxpanapa 7
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Veracruz' &
            subdata_coneval['NOM_MUN'] == 'Uxpanapa'),i:j])
#   091 - Veracruz - Jes s Carranza 8
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Veracruz' &
            subdata_coneval['NOM_MUN'] == 'Jes s Carranza'),i:
               j])

imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
                      subdata_coneval['NOM_MUN'] == 'Santa
                         Mar a Chimalapa'),i:j] <- imputar

rm(imputar)
# * * * * * * * * * * * * * * * * * * * * * * * * * Santa Mar a
   Petapa - Oaxaca
# Vecinos: (4) LISTO *
#   057 - Oaxaca -  Mat as Romero Avenda o
#   198 - Oaxaca -  San Juan Guichicovi
#   513 - Oaxaca -  Santo Domingo Petapa
#   010 - Oaxaca -  El Barrio de la Soledad


#   057 - Oaxaca -  Mat as Romero Avenda o 1
imputar <- subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'Mat as Romero
               Avenda o'),i:j]
#   010 - Oaxaca -    San Juan Guichicovi 2
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'San Juan Guichicovi'
               ),i:j])
#   513 - Oaxaca -  Santo Domingo Petapa 3
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
            subdata_coneval['NOM_MUN'] == 'Santo Domingo Petapa
               '),i:j])
#   010 - Oaxaca -  El Barrio de la Soledad 4
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
```

```
1058              subdata_coneval['NOM_MUN'] == 'El Barrio de la
                     Soledad'),i:j])
1059
1060 imputar <- sapply(imputar, as.numeric)
1061 imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))
1062
1063 subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Oaxaca' &
1064                       subdata_coneval['NOM_MUN'] == 'Santa
                             Mar a Petapa'),i:j] <- imputar
1065 rm(imputar)
1066
1067 # * * * * * * * * * * * * * * * * * * * * * * * * * * *  San Nicol s de
       los Ranchos - Puebla
1068 # Vecinos: (7) LISTO ***
1069 #   009 - NOM_ENT de M xico - Amecameca
1070 #   015 - NOM_ENT de M xico - Atlautla
1071 #   188 - Puebla - Tochimilco
1072 #   175 - Puebla - Tianguismanalco
1073 #   102 - Puebla - Nealtican
1074 #   026 - Puebla - Calpan
1075 #   074 - Puebla - Huejotzingo
1076
1077 #   009 - NOM_ENT de M xico - Amecameca 1
1078 imputar <- subdata_coneval[
1079              which(subdata_coneval['NOM_ENT'] == 'M xico' &
1080              subdata_coneval['NOM_MUN'] == 'Amecameca'),i:j]
1081 #   015 - NOM_ENT de M xico - Atlautla 2
1082 imputar <- rbind(imputar, subdata_coneval[
1083              which(subdata_coneval['NOM_ENT'] == 'M xico' &
1084              subdata_coneval['NOM_MUN'] == 'Atlautla'),i:j])
1085 #   188 - Puebla - Tochimilco 3
1086 imputar <- rbind(imputar, subdata_coneval[
1087              which(subdata_coneval['NOM_ENT'] == 'Puebla' &
1088              subdata_coneval['NOM_MUN'] == 'Tochimilco'),i:j])
1089 #   175 - Puebla - Tianguismanalco 4
1090 imputar <- rbind(imputar, subdata_coneval[
1091              which(subdata_coneval['NOM_ENT'] == 'Puebla' &
1092              subdata_coneval['NOM_MUN'] == 'Tianguismanalco'),i:
                    j])
1093 #   102 - Puebla - Nealtican 5
1094 imputar <- rbind(imputar, subdata_coneval[
1095              which(subdata_coneval['NOM_ENT'] == 'Puebla' &
1096              subdata_coneval['NOM_MUN'] == 'Nealtican'),i:j])
1097 #   026 - Puebla - Calpan 6
1098 imputar <- rbind(imputar, subdata_coneval[
1099              which(subdata_coneval['NOM_ENT'] == 'Puebla' &
1100              subdata_coneval['NOM_MUN'] == 'Calpan'),i:j])
1101 #   074 - Puebla - Huejotzingo 7
```

```
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Puebla' &
            subdata_coneval['NOM_MUN'] == 'Huejotzingo'),i:j])

imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))

subdata_coneval[which(subdata_coneval['NOM_ENT'] == 'Puebla' &
                subdata_coneval['NOM_MUN'] == 'San Nicol s de
                   los Ranchos'),i:j] <- imputar
rm(imputar)


# * * * * * * * * * * * * * * * * * * * * * * *  General Plutarco
    El as Calles - Sonora
# Vecinos: (2) LISTO ***
#   048 - Sonora - Puerto Pe asco
#   017 - Sonora - Caborca

#   048 - Sonora - Puerto Pe asco
imputar <- subdata_coneval[which(subdata_coneval['NOM_ENT'] ==
   'Sonora' &
            subdata_coneval['NOM_MUN'] == 'Puerto Pe asco'),i:
               j]
#   017 - Sonora - Caborca
imputar <- rbind(imputar, subdata_coneval[
            which(subdata_coneval['NOM_ENT'] == 'Sonora' &
            subdata_coneval['NOM_MUN'] == 'Caborca'),i:j])

imputar <- sapply(imputar, as.numeric)
imputar <- as.data.frame(t(apply((imputar),MARGIN=2,FUN=mean)))

subdata_coneval[
  which(subdata_coneval['NOM_ENT'] == 'Sonora' &
  subdata_coneval['NOM_MUN'] == 'General Plutarco El as Calles
     '),i:j] <- imputar
rm(imputar)

rm(i);rm(j)


# Comprobaci n de que ya no hay NOM_MUNs faltantes de datos
faltantes <- which(is.na(subdata_coneval$vul_ing_pob)==TRUE)
length(faltantes)
rm(faltantes)


# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
* *
# 3.2: Juntar todos los datos municipales INEGI y CONEVAL ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
setwd(input)
variables_inegi <- read_excel("Comparacion_Variables.xlsx")

# Se comprueban las dimensiones:

# Se deben tener  2469 renglones
dim(subdata_coneval)[1]==2469
dim(subdata_inegi)[1]==2469

# las columnas deben de ser las mismas
dim(inegi)[2]==dim(subdata_inegi)[2]
dim(coneval)[2]==dim(subdata_coneval)[2]

# se debe pegar por NOM_ENT y NOM_MUN
names(subdata_inegi)
names(subdata_coneval)

# Se deben de juntar los datos de las proporciones en una sola base de nuevo
data_tasas <- merge(subdata_coneval,subdata_inegi, by=c("NOM_ENT","NOM_MUN"))
nombres_limpios <- data_tasas%>%select(NOM_ENT,NOM_MUN, ENTIDAD, MUN)
# Menos dos porque son las llaves
dim(inegi)[2]+dim(coneval)[2]-2==dim(data_tasas)[2]


# El  ltimo  detalle es hacer que todas las variables queden entre
# 0 y 1, estas son los promedios
min_max_norm <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

data_tasas %>% select_if(~any(. > 1))
# identificamos los nombres y buscamos alguna coincidencia con pro (usualmente a s  se llaman
#
    los promedios)
mayores_a_1<-names(data_tasas %>% select_if(~any(. > 1)))
mayores_a_1[grep("PRO",mayores_a_1)]

# tambi n se checa a "mano"
```

```r
mayores_a_1
# y los que son promedios son:
# PROM_HNV: Promedio de hijas e hijos nacidos vivos
# GRAPROES
# GRAPROES_F
# GRAPROES_M
# PROM_OCUP
# PRO_OCUP_C


data_tasas <- data_tasas%>%
  mutate(PROM_HNV=min_max_norm(data_tasas$PROM_HNV),
         GRAPROES=min_max_norm(data_tasas$GRAPROES),
         GRAPROES_F=min_max_norm(data_tasas$GRAPROES_F),
         GRAPROES_M=min_max_norm(data_tasas$GRAPROES_M),
         PROM_OCUP=min_max_norm(data_tasas$PROM_OCUP),
         PRO_OCUP_C=min_max_norm(data_tasas$PRO_OCUP_C))

# Se borran variables que ya no se utilizar n
rm(list=setdiff(ls(), c("inegi","coneval","data_tasas",
                        "inegi_y_coneval","claves_ENT_MUN",
                        "nombres_limpios",
                        "input","output",
                        "D1.dis","D1.geo")))

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 4  Obtener la clasificaci n socioeconomica ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
pacman::p_load(NbClust,ClustGeo,ggplot2,readxl,purrr,
               cluster,factoextra,tidyr,dplyr,scales, ape)

setwd(input)



# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 4.1  Se seleccionan las variables ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
obten_data <- function(var){
  variables_inegi <- read_excel("Comparacion_Variables.xlsx",
                                sheet = "Variables")

  var_conservadas_inegi <- variables_inegi%>%
    select(Variables, var)%>%
```

```
1228      drop_na()
1229
1230
1231    # necesitamos todas las variables del coneval, excepto
            poblacion
1232    variables_coneval <- names(coneval)[names(coneval) != "
            poblacion"]
1233    variables_inegi <- var_conservadas_inegi$Variables
1234
1235    # ahora se conservan  nicamente  esas variables, menos el
            nombre de
1236    # la entidad y del municipio
1237    data <- data_tasas%>%
1238      select(c(variables_coneval,variables_inegi))
1239
1240    return(data)
1241
1242  }
1243
1244  sociodemo<-obten_data("Nuevo")
1245
1246  datos_sociodemo <- sociodemo%>%
1247    select(-c(vul_car_pob,ic_rezedu_pob,ic_asalud_pob,ic_cv_pob,
1248              ic_sbv_pob))
1249  rm(sociodemo)
1250  names(datos_sociodemo)
1251
1252
1253  # * * * * * * * * * * * * * * * * * * * * * * * * * * * *
1254  # 4.2  Resumen de las variables utilizadas  ====
1255  # * * * * * * * * * * * * * * * * * * * * * * * * * * * *
1256
1257  # Para calcular las medias
1258  datos_sociodemo %>%
1259    select(-NOM_ENT, -NOM_MUN)%>%
1260    summarise(across(everything(),~round(mean(.), 2)))
1261
1262  # Para calcular la desviaci n estandar
1263  datos_sociodemo %>%
1264    select(-NOM_ENT, -NOM_MUN)%>%
1265    summarise(across(everything(),~round(sd(.), 2)))
1266
1267  # Para el rango
1268  datos_sociodemo %>%
1269    select(-NOM_ENT, -NOM_MUN)%>%
1270    summarise(across(everything(),~round(min(.), 2)))
1271
1272  datos_sociodemo %>%
```

```r
  select(-NOM_ENT, -NOM_MUN)%>%
  summarise(across(everything(),~round(max(.), 2)))

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 4.3  Descripci n de las variables sociodemogr ficas  ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

# Se clasifican los a os promedio de escolaridad, para poder
# dar datos de cu nto % del municipio pertenece a cada quintil

inegi%>%
  select(NOM_ENT, NOM_MUN, GRAPROES)%>%
  mutate(GRAPROES = as.numeric(GRAPROES))%>%
  group_by(NOM_ENT)%>%
  summarise(promedio_estado = mean(GRAPROES))%>%
  arrange((promedio_estado))

# Para saber los municipios en donde m s a os de escolaridad
   se tiene
inegi%>%
  select(NOM_ENT, NOM_MUN, GRAPROES)%>%
  mutate(GRAPROES = as.numeric(GRAPROES))%>%
  arrange(desc(GRAPROES))


# P15YM_AN Poblaci n de 15 a os y m s analfabeta
datos_sociodemo%>%
  select(NOM_ENT, NOM_MUN, P15YM_AN)%>%
  arrange(desc(P15YM_AN))%>%
  top_n(10)%>%
  mutate(P15YM_AN = 100*round(P15YM_AN,4))


# P15YM_SE Poblaci n de 15 a os y m s sin escolaridad
datos_sociodemo%>%
  select(NOM_ENT, NOM_MUN, P15YM_SE)%>%
  arrange(desc(P15YM_SE))%>%
  top_n(10)%>%
  mutate(P15YM_SE = 100*round(P15YM_SE,4))

# Se clasifican la poblaci n de habla indigena de acuerdo a
   los quintiles
datos_sociodemo%>%
  select(NOM_ENT, NOM_MUN, P3YM_HLI)%>%
  group_by(NOM_ENT)%>%
  mutate(total_municipios = n())%>%
  mutate(P3YM_HLI = as.numeric(P3YM_HLI))%>%
  mutate(P3YM_HLI = round(P3YM_HLI,2))%>%
```

```
  mutate(clasificacion = case_when(
    P3YM_HLI <= 0 ~ "1",
    P3YM_HLI > 0 & P3YM_HLI <= 0.01 ~ "2",
    P3YM_HLI > 0.01 & P3YM_HLI <= 0.04 ~ "3",
    P3YM_HLI > 0.04 & P3YM_HLI <= 0.35 ~ "4",
    TRUE ~ "5"
  ))%>%
  group_by(NOM_ENT,total_municipios,clasificacion)%>%
  summarise(total_grupo=n())%>%
  mutate(porcentaje_grupo = 100 * total_grupo / total_
     municipios)%>%
  filter(clasificacion %in% c("1"))%>%
  filter(porcentaje_grupo > 80) %>%
  arrange(porcentaje_grupo)

poblacion90ind <- datos_sociodemo%>%
  select(NOM_ENT, NOM_MUN, P3YM_HLI)%>%
  mutate(P3YM_HLI = as.numeric(P3YM_HLI))%>%
  filter(P3YM_HLI>.9)%>%
  mutate(P3YM_HLI = 100*round(P3YM_HLI,4))

table(poblacion90ind$NOM_ENT)


# POB_AFRO Poblaci n que se considera afromexicana o
   afrodescendiente
datos_sociodemo%>%
  select(NOM_ENT, NOM_MUN, POB_AFRO)%>%
  arrange(desc(POB_AFRO))%>%
  top_n(10)%>%
  mutate(POB_AFRO = 100*round(POB_AFRO,4))


# PNACOE  Poblaci n nacida en otra entidad
datos_sociodemo%>%
  select(NOM_ENT, NOM_MUN, PNACOE)%>%
  arrange(desc(PNACOE))%>%
  top_n(10)%>%
  mutate(PNACOE = 100*round(PNACOE,4))

# pobreza_pob Poblaci n en situaci n de pobreza
pobreza_extrema <- datos_sociodemo%>%
  select(NOM_ENT, NOM_MUN, pobreza_pob)%>%
  arrange(desc(pobreza_pob))%>%
  filter(pobreza_pob > 0.95)%>%
  mutate(pobreza_pob = 100*round(pobreza_pob,4))
table(pobreza_extrema$NOM_ENT)

```

```
1365
1366 # Poblaci n sin afiliaci n a servicios de salud
1367 datos_sociodemo%>%
1368   select(NOM_ENT, NOM_MUN, PSINDER)%>%
1369   group_by(NOM_ENT)%>%
1370   mutate(total_municipios = n())%>%
1371   mutate(PSINDER = as.numeric(PSINDER))%>%
1372   mutate(PSINDER = round(PSINDER,2))%>%
1373   mutate(clasificacion = case_when(
1374     PSINDER < 0.15 ~ "1",
1375     PSINDER >= 0.15 & PSINDER < 0.2 ~ "2",
1376     PSINDER >= 0.2 & PSINDER < 0.26 ~ "3",
1377     PSINDER >= 0.26 & PSINDER < 0.33 ~ "4",
1378     TRUE ~ "5"
1379   ))%>%
1380   group_by(NOM_ENT,total_municipios,clasificacion)%>%
1381   summarise(total_grupo=n())%>%
1382   mutate(porcentaje_grupo = 100 * total_grupo / total_
       municipios)%>%
1383   filter(clasificacion %in% c("5"))%>%
1384   filter(porcentaje_grupo > 40) %>%
1385   arrange(porcentaje_grupo)
1386
1387
1388 # PSINDER Poblaci n sin afiliaci n a servicios de salud
1389 datos_sociodemo%>%
1390   select(NOM_ENT, NOM_MUN, PSINDER)%>%
1391   arrange(desc(PSINDER))%>%
1392   mutate(PSINDER = 100*round(PSINDER,4))
1393
1394 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      * *
1395 # 4.4  Se busca una K adecuada ====
1396 # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      * *
1397 # Para el Dendrograma
1398 D0 <- dist(datos_sociodemo%>%
1399              select(- NOM_ENT,- NOM_MUN),
1400            method="euclidean")
1401
1402 # gr fica del dendrograma
1403 distancia <- dist(subset(datos_sociodemo,
1404                          select = -c(NOM_ENT, NOM_MUN)))
1405 tree <- hclust(distancia, method = "ward.D2")
1406 plot(tree, cex = 0.6, hang = -1)
1407
1408 # Con el m todo del codo
1409 elbow <- fviz_nbclust(x = datos_sociodemo%>%
```

```
                          select(- NOM_ENT,- NOM_MUN),
                        FUNcluster = hcut, linecolor="#91B83F",
                        method = "wss", k.max = 10)

ggplot(data=elbow$data, aes(x=clusters, y=y, group=1))+
  labs(title = "N mero  ptimo  de clusters con el m todo del
     codo",
       x="N mero de clusters (K)", y="WSS")+
  geom_point(size=3, color="#7CB728")+
  geom_line(size=1.5, color="#7CB728")+
  theme_bw()+
  theme(text = element_text(family="Poppins"),
        plot.title = element_blank(),
        axis.title = element_text(size=19),
        axis.text = element_text(size=18))+
  scale_y_continuous(label=comma,  n.breaks=8)


setwd(output)
ggsave(filename = "codo.png",
       device = "png",width = 8,height = 6, dpi=100)

# Con el m todo silhouette
silhouette <- fviz_nbclust(x = datos_sociodemo%>%
                             select(- NOM_ENT,- NOM_MUN),
                           FUNcluster = hcut, linecolor="#3
                              FB8AF",
                           method = "silhouette", k.max = 10)

ggplot(data=silhouette$data, aes(x=clusters, y=y, group=1))+
  labs(title = "N mero  ptimo  de clusters con el m todo de
     la silueta",
       x="N mero de clusters (K)", y="Promedio del ancho de la
           silueta")+
  geom_point(size=3, color="#2A77A4")+
  geom_line(size=1.5, color="#2A77A4")+
  theme_bw()+
  theme(text = element_text(family="Poppins"),
        plot.title = element_blank(),
        axis.title = element_text(size=19),
        axis.text = element_text(size=18))+
  scale_y_continuous(label=comma,  n.breaks=8, limits = c
     (0,0.4))
setwd(output)
ggsave(filename = "silueta.png",
       device = "png",width = 8,height = 6, dpi=100)

# Con el m todo gap
```

```r
gap <- fviz_nbclust(x = datos_sociodemo%>%
                      select(- NOM_ENT,- NOM_MUN ),
                    FUNcluster = hcut, linecolor="#F08080",
                    method = "gap_stat", k.max = 10)


ggplot(data=gap$data, aes(x=clusters, y=gap, group=1))+
  labs(title = "N mero  ptimo  de clusters con el m todo gap",
       x="N mero de clusters (K)", y="Estad stics Gap (K)")+
  geom_point(size=3, color="#FFAE4D")+
  geom_line(size=1.5, color="#FFAE4D")+
  theme_bw()+
  theme(text = element_text(family="Poppins"),
        plot.title = element_blank(),
        axis.title = element_text(size=19),
        axis.text = element_text(size=18))+
  scale_y_continuous(label=comma,  n.breaks=8)

setwd(output)
ggsave(filename = "gap.png",
       device = "png",width = 8,height = 6, dpi=100)

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 4.5  Creaci n de clusters ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

# Se crea la secuencia de 21 alfas
malla <- seq(0,1,by=0.05)

# Funci n para ejecutar el algoritmo y guardar los resultados con cada alfa
malla_alfas_ks <- function(d0,d1,Ks){
  data_aux <- data.frame(matrix(nrow = dim(data_tasas)[1]))
  alphas <- malla
  data <- data_tasas%>%
    select(NOM_ENT,NOM_MUN,ENTIDAD,MUN)

  for(K in Ks){
    for(i in 1:length(alphas)){
      a <- alphas[i]
      tree <- hclustgeo(d0, d1, alpha=a)
      clusters <- data.frame(cutree(tree, K))
      name <- paste0("alfa_",a,".K_",K)
      names(clusters) <- name
      data <- cbind(data,clusters)
```

```
    }
  }



  return(data)
}


# D0, D1.dis, D1.geo
D0_D1.dis <- malla_alfas_ks(D0,D1.dis,Ks=c(3,4,5))
D0_D1.geo <- malla_alfas_ks(D0,D1.geo,Ks=c(3,4,5))



# # se guardan los datos que nos interesan
rm(list=setdiff(ls(), c("D0_D1.dis",
                        "D0_D1.geo",
                        "data_tasas",
                        "nombres_limpios",
                        "inegi","coneval",
                        "inegi_y_coneval",
                        "datos_sociodemo",
                        "claves_ENT_MUN","input","output")))

# * * * * * * * * * * * * * * * * * * * * * * * * * *
# 5.0 Elecci n de cl steres ====
# * * * * * * * * * * * * * * * * * * * * * * * * * *

#  Se cargan los datos del censo con los que se calcular  la
   varianza poblacional
setwd(paste0(input,"/data/eceg_2020_csv/conjunto_de_datos/"))

seccion <- read.csv("INE_SECCION_2020.csv")%>%
  mutate(across(c(where(is.character)), as.numeric))%>%
  rename(MUN=MUNICIPIO)


# se comprueba que la base de secci n sea una partici n de la
    del inegi por municipios
sum(seccion$POBTOT)
sum(data_tasas$POBTOT)

dim(seccion%>%
      group_by(ENTIDAD,MUN)%>%
      summarise(total=n()))
# pero las claves de los municipios no coinciden por lo que no
   se podr n unir de esta forma
```

```r
summary(seccion$MUN)
summary(data_tasas$MUN)

# por ejemplo:
inegi_y_coneval%>%
  filter(ENTIDAD==3)%>%
  distinct(NOM_ENT, NOM_MUN, ENTIDAD, MUN)

seccion%>%
  filter(ENTIDAD==3)%>%
  distinct(ENTIDAD,MUN)


# Como la clave de municipio no coincide se tiene que hacer una limpieza
# de los nombres para poder hacer el join


# * * * * * * * * * * * * * * * * * * * * * * * * *
# 5.1 Limpieza en los nombres de secciones ====
# * * * * * * * * * * * * * * * * * * * * * * * * *


# Como la clave de municipio no coincide en ambas bases se tiene que hacer una limpieza
# de los nombres para poder hacer el join

obten_secciones<-function(){
  # Se ocupa el catalogo de las secciones para poder tener sus nombres
  setwd(paste0(input,"/data/eceg_2020_csv/catalogos/"))
  cat_secciones_2020 <- read.csv("cat_secciones_2020.csv",
                                 encoding = "latin1")%>%
    distinct(CVE_ENT,CVE_MUN,DESC_MUN)%>%
    rename(ENTIDAD=CVE_ENT, NOM_MUN=DESC_MUN)%>%
    mutate(NOM_MUN=str_to_title(NOM_MUN),
           NOM_MUN=chartr("          ", "aeiou", NOM_MUN),
           NOM_MUN=chartr("          ", "AEIOU", NOM_MUN))

  # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  # Se hacen los cambios generales que se hicieron en INEGI - municipios
  # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  # Los que dicen Gral. se reemplazan por General y Dr. por Doctor
  cat_secciones_2020$NOM_MUN <- gsub("Gral.",
                                     "General", cat_secciones_
                                       2020$NOM_MUN)
```

```
cat_secciones_2020$NOM_MUN <- gsub("Dr.",
                                   "Doctor", cat_secciones_
                                      2020$NOM_MUN)


# Los municipios de San Juan Mixtepec y San Pedro Mixtepec
   vienen repetidos
cat_secciones_2020 <- cat_secciones_2020%>%
  mutate(NOM_MUN=case_when(
    CVE_MUN=='208' & ENTIDAD== '20'~ 'San Juan Mixtepec -Dto.
        08 -',
    CVE_MUN=='209' & ENTIDAD== '20'~ 'San Juan Mixtepec -Dto.
        26 -',
    CVE_MUN=='318' & ENTIDAD== '20'~ 'San Pedro Mixtepec -Dto
       . 22 -',
    CVE_MUN=='319' & ENTIDAD== '20'~ 'San Pedro Mixtepec -Dto
       . 26 -',
    TRUE ~ NOM_MUN)
  )%>%
  mutate(NOM_MUN=
           case_when(
             ENTIDAD==5 & NOM_MUN=="Cuatrocienegas" ~
               "Cuatro Cienegas",

             ENTIDAD==7 & NOM_MUN=="Villacomaltitlan" ~
               "Villa Comaltitlan",

             ENTIDAD==8 & NOM_MUN=="Doctor Belisario
                Dominguez" ~
               "Dr. Belisario Dominguez",

             ENTIDAD==10 & NOM_MUN=="Simon Bolivar" ~
               "General Simon Bolivar",

             ENTIDAD==17 & NOM_MUN=="Jonacatepec" ~
               "Jonacatepec De Leandro Valle",

             ENTIDAD==20 & NOM_MUN=="H Villa Tezoatlan Segura
                 Y Luna Cuna Ind Oax" ~
               "Heroica Villa Tezoatlan De Segura Y Luna,
                  Cuna De La Independencia De Oaxaca",
             ENTIDAD==20 & NOM_MUN=="Heroica Ciudad De
                Juchitan De Zaragoza" ~
               "Juchitan De Zaragoza",
             ENTIDAD==20 & NOM_MUN=="Santiago Chazumba" ~
               "Villa De Santiago Chazumba",

             ENTIDAD==29 & NOM_MUN=="Zitlaltepec De Trinidad
```

```
                    Sanchez Santos" ~
1622                  "Ziltlaltepec De Trinidad Sanchez Santos",
1623
1624
1625                ENTIDAD==30 & NOM_MUN=="Cosamaloapan" ~
1626                  "Cosamaloapan De Carpio",
1627                ENTIDAD==30 & NOM_MUN=="Ozuluama" ~
1628                  "Ozuluama De Mascare as",
1629                ENTIDAD==30 & NOM_MUN=="Zontecomatlan" ~
1630                  "Zontecomatlan De Lopez Y Fuentes",
1631
1632                ENTIDAD==19 & NOM_MUN=="Carmen" ~
1633                  "El Carmen",
1634                TRUE ~ NOM_MUN
1635              ))%>%
1636    rename(MUN=CVE_MUN)
1637
1638  # # secciones es la base que ya tiene ENTIDAD  y NOM_MUN, lo
       que permitir  unirla con las bases de partici n
1639  secciones <- merge(seccion,cat_secciones_2020,by=c("ENTIDAD",
       "MUN"),
1640                     all.x = TRUE)%>%
1641    filter(! ( (ENTIDAD==4 & MUN==12) |
1642               (ENTIDAD==20 & MUN==317) |
1643               (ENTIDAD==20 & MUN==316) ) )
1644
1645  return(secciones)
1646 }
1647 secciones <- obten_secciones()
1648
1649
1650
1651 # Haciendo una exploraci n de la base de datos, nos percatamos
       que
1652 # algunos datos son negativos, los quitamos
1653 secciones <- secciones%>%
1654   filter(TVIVPARHAB>0,
1655          POBTOT>0)
1656
1657 limpia_nombres<-function(data){
1658   df <- data%>%
1659     mutate(NOM_MUN=str_to_title(NOM_MUN),
1660            NOM_MUN=chartr("          ", "aeiou", NOM_MUN),
1661            NOM_MUN=chartr("          ", "AEIOU", NOM_MUN))
1662   return(df)
1663 }
1664 D0_D1.dis <- limpia_nombres(D0_D1.dis)
1665 D0_D1.geo <- limpia_nombres(D0_D1.geo)
```

```r

# * * * * * * * * * * * * * * * * * * * * * * * * * *
# 5.2 Estimando el tama o de muestra ====
# * * * * * * * * * * * * * * * * * * * * * * * * * *

tamanio_muestra <- function(k,N,S_yu,d){
  numerador <- ((k * N * S_yu) / d)^2
  denominador <- 1 + ( (1/N) * ( ( (k*N*S_yu)/d ) ^2) )
  return(numerador/denominador)
}

calculo_d <- function(var){
  suma <- as.character(prettyNum(sum(secciones[var]),
                                 big.mark=",", scientific=FALSE
                                    ))
  print(paste0("El valor verdadero de ",var , " es ", suma))
  d <- round(sum(secciones[var])*0.05,0)
  print(paste0("Por lo que d es igual a ",
               as.character(prettyNum(d, big.mark=",",
                                      scientific=FALSE))))
  return(d)
}
k =  1.96
N = dim(secciones)[1]



# Los parametros que se tratar n de estimar son:

# 1. Poblaci n con 15 a os o m s con primaria incompleta
# data_tasas | secciones
# P15PRI_IN | P15PRI_IN
tm_P15PRI_IN = tamanio_muestra(k,N,
                               sqrt(var(secciones$P15PRI_IN)),
                               calculo_d("P15PRI_IN"))

# 2. Poblaci n de 8 a 14 a os que no sabe leer ni escribir
# data_tasas | secciones
# P8A14AN | P8A14AN
tm_P8A14AN = tamanio_muestra(k,N,
                             sqrt(var(secciones$P8A14AN)),
                             calculo_d("P8A14AN"))

# 3. Poblaci n de 15 a os o m s analfabeta
# data_tasas | secciones
# P15YM_AN | P15YM_AN
tm_P15YM_AN = tamanio_muestra(k,N,
```

```
                                sqrt(var(secciones$P15YM_AN)),
                                calculo_d("P15YM_AN"))

# 4. Poblaci n con discapacidad
# data_tasas | secciones
# PCON_DISC | PCON_DISC
tm_PCON_DISC = tamanio_muestra(k,N,
                                 sqrt(var(secciones$PCON_DISC)),
                                 calculo_d("PCON_DISC"))

# 5. Poblaci n de 3 a os y m s que habla alguna lengua
   ind gena
# data_tasas | secciones
# P3YM_HLI | P3YM_HLI
tm_P3YM_HLI = tamanio_muestra(k,N,
                                sqrt(var(secciones$P3YM_HLI)),
                                calculo_d("P3YM_HLI"))

# 6. Poblaci n sin afiliaci n a servicios de salud
# data_tasas | secciones
# PSINDER | PSINDER
tm_PSINDER = tamanio_muestra(k,N,
                               sqrt(var(secciones$PSINDER)),
                               calculo_d("PSINDER"))


# 7. Viviendas particulares habitadas con piso de tierra
# data_tasas | secciones
# VPH_PISODT | VPH_PISODT
tm_VPH_PISODT = tamanio_muestra(k,N,
                                  sqrt(var(secciones$VPH_PISODT))
                                     ,
                                  calculo_d("VPH_PISODT"))

# 8. Viviendas particulares habitadas que no disponen
# de energ a electrica
# data_tasas | secciones
# VPH_S_ELEC | VPH_S_ELEC
tm_VPH_S_ELEC = tamanio_muestra(k,N,
                                  sqrt(var(secciones$VPH_S_ELEC))
                                     ,
                                  calculo_d("VPH_S_ELEC"))

# 9. Viviendas particulares habitadas que no disponen de agua
   entubada
# data_tasas | secciones
# VPH_AGUAFV | VPH_AGUAFV
tm_VPH_AGUAFV = tamanio_muestra(k,N,
```

```
1757                                    sqrt(var(secciones$VPH_AGUAFV))
                                           ,
1758                                    calculo_d("VPH_AGUAFV"))
1759
1760 # 10. Viviendas particulares habitadas que no disponen de
        drenaje
1761 # data_tasas | secciones
1762 # VPH_NODREN | VPH_NODREN
1763 tm_VPH_NODREN = tamanio_muestra(k,N,
1764                                    sqrt(var(secciones$VPH_NODREN))
                                           ,
1765                                    calculo_d("VPH_NODREN"))
1766
1767 # * * * * * * * * * * * * * * * * * * * * * * * * * * *
1768
1769 # El n mero m s grande de tama o de muestra
1770 # Es el que se tomar
1771
1772 max(tm_P15PRI_IN,
1773     tm_P8A14AN,
1774     tm_P15YM_AN,
1775     tm_PCON_DISC,
1776     tm_P3YM_HLI,
1777     tm_PSINDER,
1778     tm_VPH_PISODT,
1779     tm_VPH_S_ELEC,
1780     tm_VPH_AGUAFV,
1781     tm_VPH_NODREN)
1782
1783 # Por lo tanto, el valor de la n debe de ser: 16,375.28
1784
1785 # * * * * * * * * * * * * * * * * * * * * * * * * * * *
1786 n <- round(max(tm_P15PRI_IN,
1787                tm_P8A14AN,
1788                tm_P15YM_AN,
1789                tm_PCON_DISC,
1790                tm_P3YM_HLI,
1791                tm_PSINDER,
1792                tm_VPH_PISODT,
1793                tm_VPH_S_ELEC,
1794                tm_VPH_AGUAFV,
1795                tm_VPH_NODREN),0)
1796 n
1797 # * * * * * * * * * * * * * * * * * * * * * * * * * * *
1798
1799
1800
1801 # * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```r
# 5.3 Obtenci n de varianzas poblacionales ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * *

# Se calcula la varianza poblacional para cada par metro

variables <- c("P15PRI_IN",
               "P8A14AN",
               "P15YM_AN",
               "PCON_DISC",
               "P3YM_HLI",
               "PSINDER",
               "VPH_PISODT",
               "VPH_S_ELEC",
               "VPH_AGUAFV",
               "VPH_NODREN")

dimension <- 63



obten_df_varianzas_poblacionales<-function(variables,data){
  # variable <- variables [1]
  # data <- D0_D1.dis
  # i <-57
  obten_varianzas_poblacionales <- function(data,variable){
    # Lo primero es hacer el join entre secciones y
    # la forma de particionar
    base <- merge(secciones%>%
                    select(c("ENTIDAD","NOM_MUN","SECCION",
                       variable)),
                  data%>%
                    select(-c(MUN,NOM_ENT)), by=c("ENTIDAD","
                       NOM_MUN"),
                  all.x=TRUE)

    # Las primeras columnas estar n ordenadas de esta forma:
    #   ENTIDAD | NOM_MUN | SECCION | Variable
    # se debe de hacer la agrupaci n a partir de la columnas 5

    obten_VP <- function(base,variable,i){
      # la i es de donde va a correr el for, es para que
      # agarre la forma de agrupar
      base_temp <- base[,c(1:4,i)] # con esto ya tengo
      # las agrupaciones de las secciones
      names(base_temp)[5]<-"Cluster"
      #Asignaci n del tama o de muestra en cada estrato
      #Proporcional a Nh, es decir nh ~ n * Nh/N
      varianza_poblacional <-
```

```
1848          base_temp%>%
1849          group_by(Cluster)%>%
1850          summarise(Nh=n(), # total de poblaci n por cada grupo
1851                    nh=ceiling(n*Nh/N), # se calcula la n de cada
                          h
1852                    Syh2=var(eval(parse(text=variable))),
1853                    #varianza del conjunto h
1854                    Vh=(Nh^2/nh)*(1-nh/Nh)*Syh2 )%>%
1855          #calculo de la varianza
1856          mutate_if(is.numeric, ~replace(., is.na(.), 0))
1857          #existen grupos con una observaci n por lo cual
1858        # la varianza poblacional de ah  debe de ser cero
1859
1860        VP <- sum(varianza_poblacional$Vh)/
1861          sum(varianza_poblacional$Nh)^2
1862        VP
1863        return(VP)
1864      }
1865
1866      varianzas_poblacionales <- data.frame()
1867
1868      for(i in 5:dim(base)[2]){
1869        varianzas_poblacionales <- rbind(varianzas_poblacionales,
1870                                         data.frame(
1871                                         agrupacion = names(base)
                                               [i],
1872                         varianza_poblacional = obten_VP(base,
                                 variable,i))
1873        )
1874      }
1875      names(varianzas_poblacionales)[2]<-paste0("vp_",variable)
1876      return(varianzas_poblacionales)
1877    }
1878
1879    df_varianzas_poblacionales <- data.frame(matrix(NA,
1880                                    nrow = dimension, ncol = 0))
1881
1882    for(variable in variables){
1883
1884      df_varianzas_poblacionales <- cbind(df_varianzas_
             poblacionales,
1885                                        obten_varianzas_poblacionales
                                               (
1886                                          data,variable)[,2] )
1887    }
1888    names(df_varianzas_poblacionales) <- variables
1889    df_varianzas_poblacionales
1890
```

```r

  nombres_agrupaciones<-data.frame()
  for(i in 5:dim(data)[2]){
    nombres_agrupaciones <- rbind(nombres_agrupaciones,
                                  data.frame(
                                  agrupacion = names(data)[i])
    )
  }
  nombres_agrupaciones

  df_varianzas_poblacionales <- cbind(
                                  nombres_agrupaciones,
                                  df_varianzas_poblacionales)
  return(df_varianzas_poblacionales)
}

vp_D0_D1.dis <- obten_df_varianzas_poblacionales(variables,D0_
   D1.dis)
vp_D0_D1.geo <- obten_df_varianzas_poblacionales(variables,D0_
   D1.geo)

vp_D0_D1.dis$data_frame <- "vp_D0_D1.dis"
vp_D0_D1.geo$data_frame <- "vp_D0_D1.geo"


# Se une toda la informaci n en un  nico  data frame
data <- rbind(vp_D0_D1.dis,
              vp_D0_D1.geo)


# * * * * * * * * * * * * * * * * * * * * * * * * *
# 6.0  Elecci n de la partici n ====
# * * * * * * * * * * * * * * * * * * * * * * * * *

# Se ve el minimo para cada variable y a que agrupaci n
# y base de datos pertenece

for(i in variables){
  print(paste0("Para la variable: ", i))
  print(data%>%
          filter(eval(parse(text=i))==min(data[i])))
}



df_eleccion<-data.frame()

# * * * * * * * * * * * * * * * * * * * * * * * * *
```

```r
# 6.1  Reankeo de variables  ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

# Se rankea para cara variable, qui n tiene la menor
# varianza poblacional
for(i in variables){
  df_eleccion<-rbind(df_eleccion,
                     data%>%
                       select(agrupacion,data_frame,i)%>%
                       mutate(rank = dense_rank(eval(
                         parse(text=i))))%>%
                       select(agrupacion,data_frame,rank)%>%
                       mutate(variable = i)%>%
                       arrange(rank))

}


# Se obtiene la varianza con un muestreo aleatorio simple
df_VarMAS <- data.frame()
for(i in variables){
  df_VarMAS <- rbind(df_VarMAS,
                     data.frame(variable = i,
                    VarMAS = as.numeric((1/n)*
                                (1-(n/N))*var(secciones[i])) ) )
  #VarMAS = as.numeric((N^2/n)*(1-(n/N))*var(secciones[i])) ) )
}


# Se conserva
suma_rank <- df_eleccion%>%
  group_by(agrupacion,data_frame)%>%
  summarise(puntos=sum(rank))%>%
  mutate(estratificacion = paste0(agrupacion,"_",data_frame))
     %>%
  mutate(estratificacion = gsub("vp_D0_D1.", "",
     estratificacion))%>%
  mutate(estratificacion = gsub("alfa_", "alfa: ",
     estratificacion))%>%
  mutate(estratificacion = gsub(".K_", ", K: ", estratificacion
     ))%>%
  mutate(estratificacion = gsub("_", ", ", estratificacion))%>%
  arrange(puntos)


suma_rank
suma_rank$estratificacion

```

```
# Se realiza una gr fica para que se puedan visualizar los
    resultados
# m s f cilmente

resumen_ranks <- df_eleccion%>%
  mutate(estratificacion = paste0(agrupacion,"_",data_frame))
     %>%
  mutate(estratificacion = gsub("vp_D0_D1.", "",
     estratificacion))%>%
  mutate(estratificacion = gsub("alfa_", "alfa: ",
     estratificacion))%>%
  mutate(estratificacion = gsub(".K_", ", K: ", estratificacion
     ))%>%
  mutate(estratificacion = gsub("_", ", ", estratificacion))%>%
  select(estratificacion, variable, rank)%>%
  mutate(estratificacion = factor(estratificacion, levels=suma_
     rank$estratificacion))


# colores <- c("#F2BED1","#D988B9","#9D76C1","#176B87",
#              "#64CCC5","#EAFFD0", "#A2C579","#D2DE32",
#              "#FFE17B","#FD8D14")

library(paletteer)
colores <- paletteer_c("grDevices::Viridis", 10)

# Stacked barplot with multiple groups
ggplot(data=resumen_ranks, aes(x=rank,
                               y=estratificacion,
                               fill=variable)) +
  scale_fill_manual(values=colores)+
  geom_bar(stat="identity")+
  theme_bw()+
  labs(x = "Puntos", y = "Estratificaci n")+
  theme(text = element_text(family="Poppins"),
        plot.title = element_text(size=20,hjust = 0.5),
        axis.title.x = element_text(size=30),
        axis.title.y = element_text(size=30),
        axis.text.y = element_text(size=18,
                                   colour = "#505050"),
        axis.text.x = element_text(size=20,
                                   colour = "#505050"),
        panel.background = element_rect(fill = "white",
                                        colour = "white"),
        plot.background = element_rect(fill = "white",
                                       colour = "white"),
        panel.grid = element_line(color = "#F1F1F2",
                                  linetype = 1),
```

```
        legend.title = element_text(size=30),
        legend.text = element_text(size=25),
        panel.border = element_blank(),
        axis.ticks = element_blank(),
        axis.line = element_blank())+
  scale_x_continuous(n.breaks = 10)+
  guides(fill=guide_legend(title="Variable"))

# setwd(output)
#
# ggsave(filename = "Ranks_variables.png",
#        device = "png",width = 22, height =30 , dpi=100)

# Para saber cu ntos puntos obtuvo cada estratificaci n
resumen_ranks%>%
  group_by(estratificacion)%>%
  summarise(total= sum(rank))%>%
  arrange(desc(total))

mean((resumen_ranks%>%
        group_by(estratificacion)%>%
        summarise(total= sum(rank))%>%
        arrange(desc(total)))$total)

# La que obtuvo menos puntos fue la de alfa=0.25 K=5
# y usando la distancia entre centroides
data%>%
  filter(agrupacion=="alfa_0.25.K_5",
         data_frame=="vp_D0_D1.geo")

# Se ve el promedio de las varianzas as  como los rangos
for(i in 2:11){
  print(paste0(names(data)[i], ": ", "Promedio: ",
               round(mean(data[,i]),2),
               " Rango: ", "(",
               round(max(data[,i]),2),
               "-", round(min(data[,i]),2), ")"))

  print("-----------")
}

# Varianza poblacional para cada variable:
data%>%
  select(c("data_frame", "agrupacion" , variables))%>%
  filter((agrupacion == "alfa_0.25.K_5") &
           (data_frame=="vp_D0_D1.geo"))%>%
  mutate(across(where(is.numeric), round, 4))
```

```r
# Rank qe ocupa cada variable
ranks <- c()
for(i in variables){
  print(i)
  row <- data%>%
    select(agrupacion,data_frame,i)%>%
    mutate(rank = dense_rank(eval(parse(text=i))))%>%
    select(agrupacion,data_frame,rank)%>%
    arrange(rank)%>%
    filter((agrupacion == "alfa_0.25.K_5") &
             (data_frame=="vp_D0_D1.geo"))

  print(row)
  ranks <- append(ranks, row$rank)

}
mean(ranks)


# * * * * * * * * * * * * * * * * * * * * * * * * *
# 6.2  Evaluaci n del modelo  ====
# * * * * * * * * * * * * * * * * * * * * * * * * *
a <- "alfa_0.25.K_5"
df <- "vp_D0_D1.geo"
decimales <- 2

merge(
  data%>%
    filter(agrupacion==a,
           data_frame==df)%>%
    pivot_longer(!c(agrupacion,data_frame),
                 names_to="variable", values_to="varianza")%>%
    select(variable,varianza),
  df_VarMAS,
  by="variable"
)%>%
  mutate(mejoro=varianza<VarMAS)%>%
  # mutate(radio = varianza/VarMAS)
  mutate(reduccion=100*((VarMAS-varianza)/VarMAS))%>%
  mutate(varianza=round(varianza,decimales),
         VarMAS=round(VarMAS,decimales),
         reduccion=round(reduccion,decimales))
  # select(-mejoro)%>%
  # mutate(across(where(is.numeric), ~round(., 2)))

# * * * * * * * * * * * * * * * * * * * * * * * * *
# 7.0  Evaluaci n de otras estratificaciones  ====
# * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
2119
2120
2121 # * * * * * * * * * * * * * * * * * * * * * * * * * *
2122 # 7.1 Clasificaci n de municipios r/u  ====
2123 # * * * * * * * * * * * * * * * * * * * * * * * * * *
2124
2125 # Informaci n de las localidades obtenida de:
2126 # https://www.inegi.org.mx/programas/ccpv/2020/#datos_abiertos
2127 # Principales resultados por localidad (ITER)
2128 # Estados Unidos Mexicanos
2129
2130
2131 setwd(paste0(input,"/data/localidades/"))
2132 localidades_raw <- read.csv("conjunto_de_datos_iter_00CSV20.csv
       ")
2133
2134 # Localidades urbanas
2135 # Son aqu llas que tienen una poblaci n mayor o igual
2136 # a 2 500 habitantes o que sean cabeceras
2137 # municipales, independiente de su poblaci n.
2138 # Las localidades urbanas se representan en forma de pol gono.
2139 # Localidades rurales
2140 # Son todas las que tienen una poblaci n menor a 2 500
2141 # habitantes y no son cabeceras municipales.
2142 # Las localidades rurales se representan con un tri ngulo
2143 # o con un punto.
2144 #
2145 # https://www.inegi.org.mx/rnm/index.php/catalog/315/download/
       9636
2146
2147
2148 # Municipio totalmente rural =
2149 # 100% de la poblaci n en localidades rurales
2150 # Municipio predominantemente rural =
2151 # m s del 50% y menos del 100% de la poblaci n en localidades
       rurales
2152 # Municipio predominantemente urbano =
2153 # menos del 50% de la poblaci n en localidades rurales
2154
2155 # Aqu  tampoco coinciden las claves entre las bases de
       localidades y secciones
2156 summary(localidades_raw$MUN)
2157 summary(secciones$MUN)
2158
2159
2160
2161 localidades <- localidades_raw%>%
2162   filter_at(vars(NOM_LOC, NOM_ENT, NOM_MUN),
```

```
2163              ~!grepl("Total", .))%>%
2164   select(ENTIDAD,MUN,NOM_ENT, NOM_MUN,POBTOT)%>%
2165   mutate(POBTOT = as.numeric(POBTOT))%>%
2166   mutate(clasificacion_localidad = case_when(
2167     POBTOT<2500 ~ "rural",
2168     TRUE ~ "urbana"
2169   ))%>%
2170   group_by(ENTIDAD,MUN,NOM_ENT,NOM_MUN,clasificacion_localidad)
       %>%
2171   summarise(clasificacion_total = n())%>%
2172   group_by(ENTIDAD,MUN) %>%
2173   mutate(porcentaje = round(100 *
2174         clasificacion_total / sum(clasificacion_total),4))
2175
2176
2177 municipios_clasificados_raw <- localidades%>%
2178   group_by(ENTIDAD,MUN,NOM_ENT,NOM_MUN)%>%
2179   filter(porcentaje == max(porcentaje))%>%
2180   mutate(clasificacion_municipio = case_when(
2181     clasificacion_localidad == "rural" &
2182       porcentaje == 100  ~ "Totalmente rural",
2183     clasificacion_localidad == "rural" &
2184       (porcentaje > 50 & porcentaje <= 100)  ~
2185         "Predominantemente rural",
2186     TRUE ~ "Predominantemente urbano"
2187   ))%>%
2188   distinct(ENTIDAD,MUN,NOM_ENT,NOM_MUN,clasificacion_municipio)
2189
2190
2191 # Se deben de limpiar los nombres de los municipios
2192 # en la base de localidades
2193 clean_names_municipios_localidades<-function(){
2194   # Se ocupa el catalogo de las secciones para poder
2195   # tener sus nombres
2196   municipios <- municipios_clasificados_raw%>%
2197     mutate(NOM_MUN=str_to_title(NOM_MUN),
2198            NOM_MUN=chartr("          ", "aeiou", NOM_MUN),
2199            NOM_MUN=chartr("          ", "AEIOU", NOM_MUN))%>%
2200     rename(CVE_MUN = MUN)
2201
2202   # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
2203   # Se hacen los cambios generales que se hicieron
2204   # en INEGI - municipios
2205   # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
2206
2207   # Los que dicen Gral. se reemplazan por General y Dr. por
       Doctor
2208   municipios$NOM_MUN <- gsub("Gral.", "General", municipios$NOM
```

```
    _MUN)
municipios$NOM_MUN <- gsub("Dr.", "Doctor", municipios$NOM_
    MUN)

# Los municipios de San Juan Mixtepec y San Pedro Mixtepec
    vienen repetidos
municipios_clean <- municipios%>%
  mutate(NOM_MUN=case_when(
    CVE_MUN=='208' & ENTIDAD== '20'~
      'San Juan Mixtepec -Dto. 08 -',
    CVE_MUN=='209' & ENTIDAD== '20'~
      'San Juan Mixtepec -Dto. 26 -',
    CVE_MUN=='318' & ENTIDAD== '20'~
      'San Pedro Mixtepec -Dto. 22 -',
    CVE_MUN=='319' & ENTIDAD== '20'~
      'San Pedro Mixtepec -Dto. 26 -',
    TRUE ~ NOM_MUN)
  )%>%
  mutate(NOM_MUN=
           case_when(
             ENTIDAD==5 & NOM_MUN=="Cuatrocienegas" ~
               "Cuatro Cienegas",

             ENTIDAD==7 & NOM_MUN=="Villacomaltitlan" ~
               "Villa Comaltitlan",

             ENTIDAD==8 & NOM_MUN=="Doctor Belisario
                Dominguez" ~
               "Dr. Belisario Dominguez",

             ENTIDAD==10 & NOM_MUN=="Simon Bolivar" ~
               "General Simon Bolivar",

             ENTIDAD==17 & NOM_MUN=="Jonacatepec" ~
               "Jonacatepec De Leandro Valle",

             ENTIDAD==20 &
             NOM_MUN=="H Villa Tezoatlan Segura Y Luna Cuna
                Ind Oax" ~
             "Heroica Villa Tezoatlan De Segura Y Luna, Cuna
                De La Independencia De Oaxaca",
             ENTIDAD==20 &
             NOM_MUN=="Heroica Ciudad De Juchitan De Zaragoza
                " ~
             "Juchitan De Zaragoza",
             ENTIDAD==20 &
             NOM_MUN=="Santiago Chazumba" ~
             "Villa De Santiago Chazumba",
```

```
2250
2251                ENTIDAD==29 &
2252                NOM_MUN=="Zitlaltepec De Trinidad Sanchez Santos
                       " ~
2253                "Ziltlaltepec De Trinidad Sanchez Santos",
2254
2255
2256                ENTIDAD==30 &
2257                NOM_MUN=="Cosamaloapan" ~
2258                "Cosamaloapan De Carpio",
2259                ENTIDAD==30 &
2260                NOM_MUN=="Ozuluama" ~
2261                "Ozuluama De Mascare as",
2262                ENTIDAD==30 &
2263                NOM_MUN=="Zontecomatlan" ~
2264                "Zontecomatlan De Lopez Y Fuentes",
2265
2266                ENTIDAD==19 & NOM_MUN=="Carmen" ~ "El Carmen",
2267                TRUE ~ NOM_MUN
2268             ))%>%
2269    rename(MUN=CVE_MUN)%>%ungroup()
2270
2271
2272  return(municipios_clean)
2273 }
2274 municipios_clasificados <- clean_names_municipios_localidades()
2275
2276
2277 # Se obtiene un resumen de las clasificaciones de los
       municipios
2278 municipios_clasificados%>%
2279   group_by(clasificacion_municipio)%>%
2280   summarise(total=n())
2281
2282  # Se comprueba que no haya municipios que no vayan a matchear
       en el join
2283 rbind(secciones%>%distinct(ENTIDAD, NOM_MUN)%>%
2284         mutate(base="secciones"),
2285       municipios_clasificados%>%
2286        distinct(ENTIDAD, NOM_MUN)%>%
2287         mutate(base="municipios clasi"))%>%
2288   group_by(ENTIDAD,NOM_MUN)%>%
2289   mutate(total=n())%>%
2290   arrange(total)
2291
2292
2293 # Se crean las diferentes estratificaciones
2294 # Recordando la estratificaci n elegida
```

```r
a <- "alfa_0.25.K_5"
df <- "vp_D0_D1.geo"



secciones_clasificadas <- secciones%>%
  left_join(municipios_clasificados,
            by=c("ENTIDAD","NOM_MUN"))%>%
  left_join(D0_D1.geo%>%select(ENTIDAD, NOM_MUN, a),
            by=c("ENTIDAD","NOM_MUN"))%>%
  mutate(Entidad_TipoMunicipio =
           paste0("Entidad: ",ENTIDAD, " - ",
           "Tipo Municipio: ", clasificacion_municipio))%>%
  rename(Cluster = a)%>%
  mutate(Entidad_EstratificacionTesis =
           paste0("Entidad: ", ENTIDAD, " ",
           "Cluster: ", Cluster))%>%
  mutate(Entidad_TipoMunicipio_EstratificacionTesis =
           paste0("Entidad: ", ENTIDAD, " ",
                  "Tipo Municipio: ", clasificacion_municipio,
                    " ",
                  "Cluster: ", Cluster))


# 32
dim(table(secciones_clasificadas$ENTIDAD))
# 3
dim(table(secciones_clasificadas$clasificacion_municipio))
# 32 * 3
dim(table(secciones_clasificadas$Entidad_TipoMunicipio))
# 32 * 5
dim(table(secciones_clasificadas$Entidad_EstratificacionTesis))
# 32 * 5 * 3
dim(table(secciones_clasificadas$Entidad_TipoMunicipio_
   EstratificacionTesis))

resumen <- secciones_clasificadas%>%select(ENTIDAD,NOM_MUN,
                                clasificacion_municipio,
                                   Cluster,
                                Entidad_TipoMunicipio,
                                Entidad_EstratificacionTesis,
                                Entidad_TipoMunicipio_
                                   EstratificacionTesis)


# resumen%>%
#   filter(Cluster==3)%>%
#   left_join(D0_D1.geo%>%distinct(NOM_ENT, NOM_MUN, ENTIDAD,
```

```
    MUN))%>%
#    distinct(NOM_ENT)

# * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 7.2 C lculo de vairanzas con diferentes estratificaciones
    ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * *

# La variables de las que se calcular n las varianzas

variables <- c("P15PRI_IN",
               "P8A14AN",
               "P15YM_AN",
               "PCON_DISC",
               "P3YM_HLI",
               "PSINDER",
               "VPH_PISODT",
               "VPH_S_ELEC",
               "VPH_AGUAFV",
               "VPH_NODREN")

# Funci n para obtener las varianzas, recibe la agrupaci n
# por la que se calcular n las varianzas

obten_varianzas <- function(base, agrupacion){
  obten_VP <- function(variable,agrupacion){
    #Asignaci n del tama o de muestra en cada estrato
    #Proporcional a Nh, es decir nh ~ n * Nh/N
    varianza_poblacional <-
      base%>%
      group_by(eval(parse(text=agrupacion)))%>%
      summarise(Nh=n(), # total de poblaci n por cada grupo
                nh=ceiling(n*Nh/N), # se calcula la n de cada h
                Syh2=var( eval(parse(text=variable)) ),  #
                   varianza del conjunto h
                Vh=(Nh^2/nh)*(1-nh/Nh)*Syh2 )%>% #calculo de la
                    varianza
      mutate_if(is.numeric, ~replace(., is.na(.), 0)) #existen
         grupos con una observaci n
    # por lo cual la varianza poblacional de ah  debe de ser
       cero

    VP <- sum(varianza_poblacional$Vh)/sum(varianza_poblacional
       $Nh)^2
    VP
    return(VP)
  }

```

```
  varianzas_df <- data.frame(matrix(nrow = 0, ncol = 2))
  names(varianzas_df) <- c("Variable","Varianza")

  for(variable in variables){
    row <- data.frame(Variable = variable,
                      Varianza = obten_VP(variable, agrupacion)
                        )
    varianzas_df <- rbind(varianzas_df, row)
  }
  return(varianzas_df)
}

# Estratificaci n de la tesis
varianza_estratificacion <- obten_varianzas(secciones_
   clasificadas,"Cluster")%>%
  rename(varianza_estratificacion = Varianza)
# Tiene que dar lo mismo obtenido anteriormente
data%>%
  filter(agrupacion==a,
         data_frame==df)
varianza_estratificacion

# Varianza por estado
# Tiene 32 estratos
varianza_entidad <- obten_varianzas(
  secciones_clasificadas,"ENTIDAD")%>%
  rename(varianzas_entidad = Varianza)

# Varianza por la clasificaci n del municipio
# Tiene 3 estratos
varianza_clasificacionMunicipio <- obten_varianzas(
  secciones_clasificadas,"clasificacion_municipio")%>%
  rename(varianza_clasificacionMunicipio = Varianza)

# Varianza por entidad y tipo de municipio
# Tiene 32*4 estratos
varianza_entidad_clasificacionMunicipio <- obten_varianzas(
  secciones_clasificadas,"Entidad_TipoMunicipio")%>%
  rename(varianza_entidad_clasificacionMunicipio = Varianza)

# Varianza por entidad y estratificaci n tesis
# Tiene 32*5 estratos
varianza_entidad_estratificacion <- obten_varianzas(
  secciones_clasificadas,"Entidad_EstratificacionTesis")%>%
  rename(varianza_entidad_estratificacion = Varianza)

# Varianza por entidad, tipo de municipio y estratificaci n
   tesis
```

```r
# Tiene 32*4*5 estratos
varianza_entidad_clasificacionMunicipio_estratificacion <-
   obten_varianzas(
  secciones_clasificadas,"Entidad_TipoMunicipio_
     EstratificacionTesis")%>%
  rename(varianza_entidad_clasificacionMunicipio_
     estratificacion = Varianza)

varianzas <- varianza_estratificacion%>%
  left_join(varianza_entidad)%>%
  left_join(varianza_estratificacion)%>%
  left_join(varianza_clasificacionMunicipio)%>%
  left_join(varianza_entidad_clasificacionMunicipio)%>%
  left_join(varianza_entidad_estratificacion)%>%
  left_join(varianza_entidad_clasificacionMunicipio_
     estratificacion)

varianzas <- varianzas %>%
  mutate_if(is.numeric, round, digits=2)



# * * * * * * * * * * * * * * * * * * * * * * * * *
# 8.0 Descripci n de los resultados  ====
# * * * * * * * * * * * * * * * * * * * * * * * * *

# * * * * * * * * * * * * * * * * * * * * * * * * *
# 8.1 Correlaciones variables sociodemo  ====
# * * * * * * * * * * * * * * * * * * * * * * * * *
library(reshape2);library(showtext)
font_add_google(c("Special Elite"))
font_add_google(c("Poppins"))
showtext_auto()

heat <- melt(datos_sociodemo%>%
               mutate(id=paste0(NOM_ENT,"_",NOM_MUN))%>%
               select(-c(NOM_ENT,NOM_MUN) ) )

# Se obtiene la correlacion
cormat <- round(cor(datos_sociodemo%>%select(-c(NOM_ENT,NOM_MUN
   ) )),2)
melted_cormat <- melt(cormat)


ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()+
  # scale_fill_gradient(low = "yellow", high = "red", na.value
     = NA)+
```

```
  scale_fill_gradient2(low = "#004779", high = "#A50021",mid =
     "white",
                       na.value = NA,guide = "colourbar",
                       aesthetics = "fill")+
  labs(title = "Correlaci n entre variables
       sociodemogr ficas con el m todo de pearson")+
  theme_bw()+
  theme(text = element_text(family="Poppins"),
        plot.title = element_text(size=20,hjust = 0.5),
        axis.title = element_blank(),
        axis.text.x = element_text(size=15,
                                   family="Special Elite",
                                   angle=90, hjust = 1),
        axis.text.y = element_text(size=15,
                                   family="Special Elite"),
        panel.background = element_rect(fill = "white",
                                        colour = "white"),
        plot.background = element_rect(fill = "white",
                                       colour = "white"),
        panel.grid = element_blank(),
        legend.title = element_blank(),
        legend.text = element_text(size=14),
        panel.border = element_blank(),
        axis.ticks = element_blank(),
        axis.line = element_blank())

# setwd(output)
# ggsave(filename = "correlaciones.png",
#        device = "png",width = 10,height = 10 , dpi=100)


# * * * * * * * * * * * * * * * * * * * * * * * * *
# 8.2 PCA  ====
# * * * * * * * * * * * * * * * * * * * * * * * * *
estratificacion <- D0_D1.geo[c("alfa_0.25.K_5","ENTIDAD","MUN")
   ]%>%
  left_join(nombres_limpios)


datos_estratos <- merge(estratificacion, datos_sociodemo,
                        by=c("NOM_ENT", "NOM_MUN"))%>%
  rename(grupos="alfa_0.25.K_5")%>%
  mutate(grupos=as.character(grupos))%>%
  mutate(grupos=case_when(
    grupos == "3" ~ "1",
    grupos == "1" ~ "2",
    grupos == "2" ~ "3",
    TRUE ~ grupos
```

```
  ))%>%
  mutate(grupos=factor(grupos,levels=c("1","2","3","4","5")))

# Se hace el analisis de pca
library(ggfortify)

df_PCA <- datos_estratos%>%
  select(-c(NOM_ENT,NOM_MUN,ENTIDAD,MUN,grupos))

pca.obj <- prcomp(df_PCA)

colores <- c("#608F3D","#A2CF49","#FEC306","#E76618","#4A9BDC")
ggplot(data = data.frame('Grupo' = datos_estratos$grupos,
                         pca.obj$x[,1:2]) ) +
  geom_point(aes(x = PC1, y = PC2, col = Grupo)) +
  theme_bw() +
  scale_color_manual(values=colores)+
  labs(x="PC1(64.13%)", y="PC2(8.49%)")+
  theme(text = element_text(family="Poppins"),
        plot.title = element_text(size=20,hjust = 0.5),
        axis.title = element_text(size=19),
        legend.title = element_text(size=20),
        legend.text = element_text(size=25),
        axis.text = element_text(size=18),
        panel.border = element_blank(),
        panel.background = element_blank())+
  guides(color = guide_legend(override.aes = list(size = 5)))


# setwd(output)
# ggsave(filename = "PCA.png",
#        device = "png",width = 10,height = 10 , dpi=100)


# Se obtiene la correlaci n entre las variables y el PCA1

library("FactoMineR")
res.pca <- PCA(df_PCA, graph = FALSE)
res.desc <- dimdesc(res.pca, axes = c(1,2), proba = 0.05)

corr_PCA1 <- as.data.frame(res.desc$Dim.1)%>%
  select(quanti.correlation)%>%
  mutate(quanti.correlation = round(quanti.correlation,2))
corr_PCA1


corr_PCA2 <- as.data.frame(res.desc$Dim.2)%>%
  select(quanti.correlation)%>%
```

```
  mutate(quanti.correlation = round(quanti.correlation,2))
corr_PCA2


# * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 8.3 Total de municipios por grupo  ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * *

total_municipios <- datos_estratos%>%
  group_by(grupos)%>%
  summarise(total=n())%>%
  mutate(porcentaje = round(100*total/sum(total),2))

names(total_municipios) = str_to_title(names(total_municipios))
total_municipios
ggplot(data=total_municipios, aes(x=Grupos, y=Total,
                                  fill=Grupos)) +
  scale_fill_manual(values=colores)+
  geom_bar(stat="identity")+
  theme_bw()+
  labs(x = "Estrato", y = "Total de municipios")+
  theme(text = element_text(family="Poppins"),
        plot.title = element_text(size=20,hjust = 0.5),
        axis.title.x = element_text(size=30),
        axis.title.y = element_text(size=30),
        axis.text.y = element_text(size=18,
                                   colour = "#505050"),
        axis.text.x = element_text(size=20,
                                   colour = "#505050"),
        panel.background = element_rect(fill = "white",
                                        colour = "white"),
        plot.background = element_rect(fill = "white",
                                       colour = "white"),
        panel.grid = element_line(color = "#F1F1F2",
                                  linetype = 1),
        legend.title = element_text(size=30),
        legend.text = element_text(size=25),
        panel.border = element_blank(),
        axis.ticks = element_blank(),
        legend.position = "none",
        axis.line = element_blank())+
  scale_y_continuous(n.breaks = 5)

setwd(output)
ggsave(filename = "total_grupos_barras.png",
       device = "png", width = 8, height = 5, dpi=100)


```

```
2609
2610 # * * * * * * * * * * * * * * * * * * * * * * * * * * *
2611 # 8.4 BoxPlots  ====
2612 # * * * * * * * * * * * * * * * * * * * * * * * * * * *
2613 obten_boxplot <- function(base){
2614   g <- ggplot(base, aes(x=grupos, y=var, fill=grupos)) +
2615     geom_boxplot()+
2616     scale_fill_manual(values=colores)+
2617     theme_bw()+
2618     scale_y_continuous(n.breaks=10)+
2619     theme(text = element_text(family="Poppins"),
2620           plot.title = element_blank(),
2621           axis.title = element_text(size=25),
2622           legend.title = element_text(size=20),
2623           legend.text = element_text(size=18),
2624           axis.text = element_text(size=18, color="#4D4D4D"),
2625           panel.grid.major = element_blank(),
2626           panel.border = element_blank(),
2627           legend.position = "none")+
2628     labs(y='', x="Grupo")
2629   return(g)
2630 }
2631
2632 # Grado promedio de escolaridad
2633 obten_boxplot(datos_estratos%>%select(grupos, ENTIDAD, MUN)%>%
2634                 left_join(inegi)%>%select(grupos,GRAPROES)%>%
2635                 mutate(GRAPROES = as.numeric(GRAPROES))%>%
2636                 rename(var = GRAPROES))
2637
2638 setwd(output)
2639 ggsave(filename = "Box_plot_GRAPROES.png",
2640        device = "png",width = 10, height = 10 , dpi=100)
2641
2642
2643 # Pobreza poblacional
2644 obten_boxplot(datos_estratos%>%select(grupos,pobreza_pob)%>%
2645                 mutate(pobreza_pob = as.numeric(pobreza_pob))
                        %>%
2646                 rename(var = pobreza_pob))
2647
2648 setwd(output)
2649 ggsave(filename = "Box_plot_pobreza_pob.png",
2650        device = "png",width = 10, height = 10 , dpi=100)
2651
2652
2653 # Promedio de ocupantes por cuarto
2654 obten_boxplot(datos_estratos%>%select(grupos, ENTIDAD, MUN)%>%
2655                 left_join(inegi)%>%select(grupos,PRO_OCUP_C)%>%
```

```
                mutate(PRO_OCUP_C = as.numeric(PRO_OCUP_C))%>%
                rename(var = PRO_OCUP_C))

setwd(output)
ggsave(filename = "Box_plot_PRO_OCUP_C.png",
       device = "png",width = 10, height = 10 , dpi=100)


# Poblaci n de habla ind gena
obten_boxplot(datos_estratos%>%select(grupos,P3YM_HLI)%>%
                mutate(P3YM_HLI = as.numeric(P3YM_HLI))%>%
                rename(var = P3YM_HLI))

setwd(output)
ggsave(filename = "Box_plot_P3YM_HLI.png",
       device = "png",width = 10, height = 10 , dpi=100)


# * * * * * * * * * * * * * * * * * * * * * * * * * * *
# 8.5 Descripci n de los boxplots  ====
# * * * * * * * * * * * * * * * * * * * * * * * * * * *

# Para los a os promedio de escolaridad
datos_estratos%>%select(grupos, ENTIDAD, MUN)%>%
  left_join(inegi)%>%select(grupos,GRAPROES)%>%
  mutate(GRAPROES = as.numeric(GRAPROES))%>%
  group_by(grupos)%>%
  summarise(promedio  = mean(GRAPROES),
            minimo = min(GRAPROES),
            maximo = max(GRAPROES),
            mediana = median(GRAPROES))


# Pobreza poblacional
datos_estratos%>%select(grupos,pobreza_pob)%>%
  mutate(pobreza_pob = as.numeric(pobreza_pob))%>%
  group_by(grupos)%>%
  summarise(promedio  = mean(pobreza_pob),
            minimo = min(pobreza_pob),
            maximo = max(pobreza_pob),
            mediana = median(pobreza_pob))

# Ocupantes por cuarto
datos_estratos%>%select(grupos, ENTIDAD, MUN)%>%
  left_join(inegi)%>%select(grupos,PRO_OCUP_C)%>%
  mutate(PRO_OCUP_C = as.numeric(PRO_OCUP_C))%>%
  group_by(grupos)%>%
  summarise(promedio  = mean(PRO_OCUP_C),
```

```
2704              minimo = min(PRO_OCUP_C),
2705              maximo = max(PRO_OCUP_C),
2706              mediana = median(PRO_OCUP_C))
2707
2708
2709 # * * * * * * * * * * * * * * * * * * * * * * * * * *
2710 # 8.6 # Municipios en cada grupo por estado  ====
2711 # * * * * * * * * * * * * * * * * * * * * * * * * * *
2712 estratos_por_estados <- datos_estratos%>%
2713   group_by(NOM_ENT, grupos)%>%
2714   summarise(total=n())%>%
2715   group_by(NOM_ENT)%>%
2716   mutate(porcentaje = total /sum(total))%>%
2717   mutate(info = paste0(total ,
2718                        " (", 100 * round(porcentaje,4), "%)"))
                           %>%
2719   select(grupos, info)%>%
2720   pivot_wider(names_from = grupos, values_from = info)%>%
2721   rename(grupo_1 = '1',
2722          grupo_2 = '2',
2723          grupo_3 = '3',
2724          grupo_4 = '4',
2725          grupo_5 = '5')%>%
2726   select(NOM_ENT, grupo_1, grupo_2, grupo_3, grupo_4, grupo_5)
      %>%
2727   mutate_all(~replace(., is.na(.), "-"))
2728
2729 #write.csv(estratos_por_estados,"estratos_por_estados.csv",row.
    names=FALSE)
2730
2731
2732 # * * * * * * * * * * * * * * * * * * * * * * * * * *
2733 # 8.7 # Tabla de variables promedio(rango)  ====
2734 # * * * * * * * * * * * * * * * * * * * * * * * * * *
2735
2736 names(datos_estratos)
2737
2738 medias_grupos <- datos_estratos%>%
2739   select(-NOM_ENT, -NOM_MUN, -ENTIDAD, -MUN)%>%
2740   group_by(grupos)%>%
2741   mutate(grupos = case_when(
2742     grupos == "1" ~ "grupo_1",
2743     grupos == "2" ~ "grupo_2",
2744     grupos == "3" ~ "grupo_3",
2745     grupos == "4" ~ "grupo_4",
2746     grupos == "5" ~ "grupo_5"
2747   ))%>%
2748   summarise(across(everything(), mean))
```

```r

min_grupos <- datos_estratos%>%
  select(-NOM_ENT, -NOM_MUN, -ENTIDAD, -MUN)%>%
  group_by(grupos)%>%
  mutate(grupos = case_when(
    grupos == "1" ~ "grupo_1",
    grupos == "2" ~ "grupo_2",
    grupos == "3" ~ "grupo_3",
    grupos == "4" ~ "grupo_4",
    grupos == "5" ~ "grupo_5"
  ))%>%
  summarise(across(everything(), min))


max_grupos <- datos_estratos%>%
  select(-NOM_ENT, -NOM_MUN, -ENTIDAD, -MUN)%>%
  group_by(grupos)%>%
  mutate(grupos = case_when(
    grupos == "1" ~ "grupo_1",
    grupos == "2" ~ "grupo_2",
    grupos == "3" ~ "grupo_3",
    grupos == "4" ~ "grupo_4",
    grupos == "5" ~ "grupo_5"
  ))%>%
  summarise(across(everything(), max))

nombres_variables <- names(datos_estratos)[-c(1:5)]

medias_t <- as.data.frame(t(medias_grupos)[-1,])%>%
  mutate(V1 = as.numeric(V1),
         V2 = as.numeric(V2),
         V3 = as.numeric(V3),
         V4 = as.numeric(V4),
         V5 = as.numeric(V5))%>%
  mutate(variable = nombres_variables)%>%
  mutate(across(where(is.numeric), round, digits=2))

min_t <- as.data.frame(t(min_grupos)[-1,])%>%
  mutate(V1 = as.numeric(V1),
         V2 = as.numeric(V2),
         V3 = as.numeric(V3),
         V4 = as.numeric(V4),
         V5 = as.numeric(V5))%>%
  mutate(variable = nombres_variables)%>%
  mutate(across(where(is.numeric), round, digits=2))

max_t <- as.data.frame(t(max_grupos)[-1,])%>%
  mutate(V1 = as.numeric(V1),
```

```
          V2 = as.numeric(V2),
          V3 = as.numeric(V3),
          V4 = as.numeric(V4),
          V5 = as.numeric(V5))%>%
  mutate(variable = nombres_variables)%>%
  mutate(across(where(is.numeric), round, digits=2))


names(medias_t) <- c(paste0(c("grupo_1","grupo_2",
                              "grupo_3", "grupo_4",
                              "grupo_5"),"_media"),"variable")
names(min_t) <- c(paste0(c("grupo_1", "grupo_2",
                           "grupo_3", "grupo_4",
                           "grupo_5"),"_min"),"variable")
names(max_t) <- c(paste0(c("grupo_1", "grupo_2",
                           "grupo_3", "grupo_4",
                           "grupo_5"),"_max"),"variable")


resumen_variables <- medias_t%>%
  left_join(min_t)%>%
  left_join(max_t)%>%
  mutate(grupo_1 = paste0(grupo_1_media,
                          " (", grupo_1_min,
                          " - ", grupo_1_max,")"))%>%
  mutate(grupo_2 = paste0(grupo_2_media,
                          " (", grupo_2_min,
                          " - ", grupo_2_max,")"))%>%
  mutate(grupo_3 = paste0(grupo_3_media,
                          " (", grupo_3_min,
                          " - ", grupo_3_max,")"))%>%
  mutate(grupo_4 = paste0(grupo_4_media,
                          " (", grupo_4_min,
                          " - ", grupo_4_max,")"))%>%
  mutate(grupo_5 = paste0(grupo_5_media,
                          " (", grupo_5_min,
                          " - ", grupo_5_max,")"))%>%
  select(variable, grupo_1, grupo_2, grupo_3, grupo_4, grupo_5)


# write.csv(resumen_variables,"resumen_grupos.csv",row.names =
   FALSE)
```

Capítulo 9

Bibliografía

- Bivand, R. (2022). R Packages for Analyzing Spatial Data: A Comparative Case Study with Areal Data Geographical Analysis, 54(3), 488-518. https://doi.org/10.1111/gean.12319
- Bivand, R. y Keitt, T., Rowlingson, B. (2023). rgdal: Bindings for the 'Geospatial' Data Abstraction Library. R package version 1.6-4. https://CRAN.R-project.org/package=rgdal
- Chavent, M., Kuentz-Simonet, V., Labenne, A. y Saracco, J. (2018). ClustGeo: an R package for hierarchical clustering with spatial constraints. *Computational Statistics*, 33, 1799–1822. https://doi.org/10.1007/s00180-018-0791-1
- Chavent, M., Kuentz, V., Labenne, A. y Saracco, J. (2021). ClustGeo: Hierarchical Clustering with Spatial Constraints. R package version 2.1. https://CRAN.R-project.org/package=ClustGeo
- CONEVAL. (2018). *Informe de pobreza en los municipios de México 2015*. CONEVAL.
- Giordani, P., Ferraro, M. y Martella, F. (2020). *An Introduction to Clustering with R*. Springer.
- INEGI. (2010). *Manual de cartografía geoestadística*. INEGI.
- INEGI. (2019). *Cómo se hace la ENOE 2019*. INEGI.
- INEGI. (2021) a). *Censo de Población y Vivienda 2020: marco conceptual*. INEGI.
- INEGI. (2021) b). *Metodología para el cálculo de las Estadísticas Censales a Escalas Geoelectorales 2020*. INEGI.
- INEGI. (2022). *Programa anual de trabajo*. INEGI.
- James, G., Witten, D. y Hastie, T. (2021). An introduction to Statistical Learning With applications in R. Springer.
- Kassambara, A. (2017). *Practical Guide To Cluster Analysis in R*. STHDA.

- Kassambara, A. y Mundt, F. (2020). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.7. https://CRAN.R-project.org/package=factoextra
- Landeros, A. (2013). *Reunión nacional de estadística*. INEGI. https://www.inegi.org.mx/contenidos/eventos/2013/fne/P-AnaMariaLanderos.pdf
- Leisch, F. y Dimitriadou, E. (2021). *Machine Learning Benchmark Problems*. R package version 2.1
- Lohr, S. (2019). *Sampling Design and Analysis*. Wiley.
- Pebesma, E., (2018). Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446. https://doi.org/10.32614/RJ-2018-009
- Pérez de la Cruz, G. (2023). *Notas de clase. Muestreo. Semestre 2022-1*. Facultad de Ciencias, UNAM.
- Rodriguez Muñoz, J. y Heredia Hernández, O. (11 de junio de 2023). *La muestra maestra de viviendas: Diseño, actualización y uso* [Archivo de Vídeo]. Youtube. Canal Seminario de Estadística y Actuaría Facultad de Ciencias https://www.youtube.com/watch?v=U7RYp1x-VP0ab_channel =SeminariodeEstad%C3%ADsticayActuar%C3%ADaFacultaddeCiencias
- Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7
- Sarndal, C., Swensson, B. y Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer.
- Tillé, Y. (2020). *Sampling and estimation from finite populations*. Wiley.
- Thorndike, R. (1953). Who belongs in the family. *Psychometrika*, 18, 267-276. https://doi.org/10.1007/BF02289263
- Tibshirani, R., Walther, G. y Hastie, T. (2001). Estimating the Number of Clusters in a Data Set Via the Gap Statistic. *Journal of the Royal Statistical Society Series B*, 63(2), 411-423. https://doi.org/10.1111/1467-9868.00293
- Ward, J. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58 (301), 236-244. https://doi.org/10.1080/01621459.1963.10500845