FACULTAD DE ESTUDIOS ESTADÍSTICOS

GRADO EN ESTADISTICA APLICADA

Curso 2023/2024

Trabajo de Fin de Grado

TITULO: **ANÁLISIS DE CLUSTERS PARA LA SEGMENTACIÓN DE CÓDIGOS POSTALES EN BASE A CARACTERÍSITICAS SOCIO-DEMOGRÁFICAS EN ESPAÑA**

***Alumno*: Gabriel Pons Fontoira**

***Tutor*: Alicia Pérez Alonso**

Junio de 2024

UNIVERSIDAD COMPLUTENSE
MADRID

AGRADECIMIENTOS

Quisiera expresar mi más profundo agradecimiento a todas aquellas personas que han hecho posible la realización de este proyecto de fin de grado.

En primer lugar, agradezco a mi tutora del proyecto, la Dra. Alicia Pérez Alonso, por su apoyo, orientación y paciencia a lo largo de todo este proceso. Gracias por sus valiosos consejos y por dedicar tiempo a revisar y mejorar cada aspecto del proyecto.

También deseo agradecer a mi tutor de prácticas, Javier González Méndez, por brindarme la oportunidad de aplicar mis conocimientos en un entorno profesional y por su inestimable guía durante el periodo de prácticas. Su disposición para compartir su conocimiento y su apoyo constante han sido cruciales para mi aprendizaje y desarrollo profesional.

A mi familia, les expreso mi más sincero agradecimiento por su amor incondicional, comprensión y apoyo a lo largo de estos años. Gracias por creer en mí y por brindarme el ánimo necesario en los momentos difíciles. Su confianza y aliento han sido una fuente constante de motivación para alcanzar mis metas académicas.

¡Muchas gracias a todos!

ANÁLISIS DE CLUSTERS PARA LA SEGMENTACIÓN DE CÓDIGOS POSTALES EN BASE A CARACTERÍSITICAS SOCIO-DEMOGRÁFICAS EN ESPAÑA

RESUMEN

Este trabajo presenta la creación de una base de datos exhaustiva que engloba observaciones de códigos postales españoles. Esta iniciativa se materializó mediante la recopilación de datos de fuentes como el Instituto Nacional de Estadística, que proporciona información a nivel de sección censal sobre variables demográficas, de renta y fuentes de ingresos. Por otro lado, empleando *web scraping*, de la página web RealAdvisor se obtuvo el precio del metro cuadrado por código postal. Para unificar la unidad de observación geográfica, utilizando El Callejero, se escalaron las variables a nivel de código postal.

Aplicando el algoritmo de clustering K-Means a esta base de datos, se han seleccionado seis *clusters* distintos que reflejan diferentes perfiles socioeconómicos dentro de la población española. La originalidad y valor de este estudio radican en su enfoque metodológico simple, pero eficaz, y en la utilización combinada de algoritmos avanzados de análisis de datos. La identificación de *clusters* es una herramienta valiosa para la segmentación y análisis de datos, que facilita la comprensión de patrones económicos y sociales. Además, los resultados obtenidos son de suma importancia para diversas aplicaciones comerciales, ya que proporcionan una perspectiva detallada a nivel geográfico, permitiendo enriquecer con información adicional de alto valor estratégico a la empresa.

Palabras clave: Cluster, código postal, K-Means, algoritmo.

CLUSTER ANALYSIS FOR THE SEGMENTATION OF POSTAL CODES BASED ON SOCIO-DEMOGRAPHIC CHARACTERISTICS IN SPAIN

ABSTRACT

This project presents the creation of a database which holds observations of Spanish postal codes. This initiative was materialized by collecting data from sources such as the National Institute of Statistics, which provides information at census section level on demographic variables, income, and income sources. On the other hand, using web scraping, the price per square meter was obtained. To unify the variables, using El Callejero, the variables were scaled to the postal code level.

Applying the K-Means clustering algorithm to this database, six different clusters were selected to reflect different socioeconomic profiles within the Spanish population. The originality and value of this study lies in its simple but effective methodological approach and the combined use of advanced data analysis algorithms. The identification of clusters is a valuable tool for data segmentation and analysis, which facilitates the understanding of economic and social patterns. In addition, the results obtained are of great importance for various business applications, as they provide a detailed perspective at a geographic level, allowing to enrich the company with additional information of high strategic value.

Key words: cluster, postal code, K-Means, algorithm

1

ÍNDICE

1. INTRODUCCIÓN

1.1. Contexto

A diario, cuando escuchamos las noticias o se analizan datos zonales en España suelen estar agrupados por comunidades, provincias o incluso municipios. ¿Alguna vez te has preguntado si se puede reducir la información a unidades de terreno todavía más precisas? ¿Existe información relevante para estas zonas? ¿Tendrán estas zonas un perfil determinado y, por tanto, una posible clasificación? ¿Ser capaces de discernir entre zonas es útil?

En el 2024 firmé un contrato de prácticas con Generali Seguros como científico de datos y estas mismas dudas me surgieron durante un *brainstorming*. Si una cosa tengo clara, es que el ser humano sigue patrones y tendencias de comportamiento que le llevan a juntarse con otras personas con las que comparten características, intereses y hábitos. Además, está claro que en casa es donde uno busca seguridad y comodidad. Entonces, ¿Existen patrones a la hora de escoger vivienda?

1.2. Objetivo

La hipótesis principal de este trabajo es comprobar si a partir de la división del terreno español por unidades geográficas más pequeñas que los municipios, en particular el código postal (CP), se pueden obtener perfiles diferenciados de los CP en base a una serie de características socio-demográficas como la renta, fuentes de ingresos, edad y nacionalidad entre otras. Además, como objetivos secundarios, se espera que el desarrollo de este estudio ayude a potenciar proyectos de ventas y marketing, y que la creación de esta base de datos (BD) sea de utilidad para desarrollar otros estudios útiles para la empresa.

Otros estudios de *clustering* como Martori & Hoberg (2008) y Gómez-Barroso et al. (2015) utilizan como medida zonal la sección censal (SC). Los dos artículos analizan el uso de técnicas de estadística espacial, como el

diagrama de dispersión de Moran y los indicadores locales de asociación espacial para detectar conglomerados espaciales. Las técnicas utilizadas para detectar y caracterizar *clusters* en áreas urbanas incluyen el uso del Indicador Local de Asociación Espacial (LISA). Además, el primer artículo también añade un modelo econométrico espacial para analizar las características socioeconómicas de estas áreas.

A diferencia de estos artículos, en este proyecto se va a aportar un enfoque diferente utilizando el algoritmo de clusterización K-Means por su sencillez y facilidad de aplicación a la vez que su capacidad de aportar resultados concluyentes.

1.3. Organización

El resto del proyecto se organiza como sigue.

En el apartado 2, Metodología, se exponen las diferentes técnicas estadísticas aplicadas para realizar el análisis empírico.

En el apartado 3, Lenguaje de Programación, Software, Entorno de Desarrollo y Almacenamiento de Datos, se expone la tecnología utilizada para el desarrollo del trabajo. Microsoft Azure, Python y Excel son los programas y lenguajes principales del proyecto.

En el apartado 4, Creación de la Base de Datos, se desarrolla la base de datos. En la BD se recopilarán diferentes variables recogidas en el Instituto Nacional de Estadística (INE, 2024). Por otro lado, a partir de la web RealAdvisor (RealAdvisor, 2024), se captura el Precio del Metro Cuadrado realizando *web scrapping*. Esta variable, se acumula con una unidad geográfica superior a la sección censal, el código postal. Por lo tanto, el siguiente proceso será realizar una transformación de los datos ya almacenados como SC. Para ello, se utiliza la BD El Callejero, que almacena la relación existente entre SC y CP. Con esta información se ponderan todas las variables para así conseguir todas las observaciones identificadas como CP. Por último, para completar los valores perdidos, se realiza una imputación a partir de un *smoothing* por los CP cercanos

(vecinos). Para llevarlo a cabo, se utiliza otra BD, que contiene la relación de los CP cercanos.

El apartado 5, Análisis descriptivo de la Base de Datos, está dedicado al análisis descriptivo de la BD. En este, después de analizar el comportamiento de las variables, se estudia la matriz de correlaciones, buscando relaciones lineales entre variables y descartar así las que no aportan información por ser combinación lineal de otras variables. La última parte del análisis consiste en el tratamiento de *outliers*, que deben ser eliminados para la clusterización posterior. La idea original era detectar estos valores atípicos utilizando otro algoritmo de clusterización llamado DBSCAN. Aunque su desarrollo se ha incluido, no ha funcionado como se esperaba y, por tanto, los outliers han sido finalmente eliminados evaluando los *z-scores* con la BD normalizada.

En el apartado 6, Clusterización con K-Means, se muestra cómo se ha aplicado el algoritmo K-Means para agrupar los diferentes CP y los perfiles obtenidos de los resultados pertinentes.

El apartado 7, Conclusiones, presenta las conclusiones obtenidas a partir del trabajo realizado.

Este trabajo además incluye el Anexo, que contiene el acceso a los códigos almacenados en GitHub, archivos de Excel y gráficos descriptivos de todas las variables.

2. METODOLOGÍA

2.1. Metodología para tratamiento de *outliers*

La metodología más común para el tratamiento de *outliers* consiste en normalizar la BD (*z-scores*) y descartar todas las observaciones que tengan valores fuera del rango (-3,3) en cualquier variable. Esto es porque en la distribución Normal (0,1), la probabilidad de que una observación sea mayor o menor que ±3 es menor a 0,00135.

En vez de realizar este proceso en una primera instancia, el tratamiento de *outliers* se ha desarrollado a partir del algoritmo de clusterización no supervisado DBSCAN (Ester, et al., 1996), buscando un resultado más preciso. Para trabajar con él, se requiere el uso del paquete Scikit-*Learn* (Pedregosa et al., 2011). Este paquete se abrevia como *sklearn.*

DBSCAN es la abreviatura de las siglas Density-Based Spatial Clustering of Applications with Noise. El algoritmo diferencia en 3 tipos de observaciones:

- Observación central: son las observaciones con alta densidad de observaciones vecinas y, por tanto, los centros de los *clusters*.
- Observación periférica: son las observaciones que pertenecen a un *cluster*, pero no alcanzan la densidad suficiente para pertenecer a la categoría de observación central.
- Ruido: son las observaciones que no pertenecen a ningún *cluster*.

Para clasificar las observaciones el modelo necesita dos parámetros:

- ε: distancia entre observaciones para considerarse vecinas.
- Puntos mínimos (min_pts): número entero que marca el mínimo número de vecinos requerido para considerar una observación suficientemente densa como para considerarla observación central.

El algoritmo funciona como precede. Escoge un primer punto y lo evalúa, si es una observación central, comienza la creación del primer *cluster*. Recorre todos los puntos densamente alcanzables por este punto central y los clasifica. Si alguno de estos puntos es también un punto central, pasará a pertenecer a la misma categoría de *cluster* y procede a recorrer los puntos densamente alcanzables por este segundo punto central. Este proceso se repite de forma iterada, analizando todas las observaciones del conjunto de datos buscando un siguiente punto central, creando así los *clusters* restantes.

Posteriormente, a partir de tres medidas se evaluarán los parámetros: Índice de Silhouette (Rodríguez, 2023), Índice de Davies-Bouldin (Geeks for

Geeks, 2023) e Índice de Calinski-Harabasz (Alteryx, 2019). Partiendo de que cuando se realiza una clusterización buscamos clusters heterogéneos entre sí, pero con observaciones homogéneas en cada *cluster*, los 3 índices medirán cómo de alta es la heterogeneidad entre clusters y cómo de alta es la homogeneidad dentro de los mismos, pero a partir de diferentes desarrollos matemáticos. El Índice de Silhouette toma valores entre 1 y -1, siendo 1 la mejor métrica posible y -1 la peor, el de Davies-Bouldin indicará mejores resultados cuanto menor sea su valor, y el de Calinski-Harabasz será justo al contrario (cuanto mayor sea, mejor).

2.2. Metodología para Visualizar la Base de Datos Gráficamente

Nuestro conjunto de datos tiene más de 3 variables. Para poder visualizarlos, una opción es reducir la dimensionalidad a 3 o menos variables. Para ello, se aplica el Análisis de Componentes Principales (PCA).

El PCA es una técnica estadística utilizada para reducir la dimensionalidad, mientras se conserva la mayor cantidad posible de la variabilidad presente en esos datos. Este método transforma las variables originales en un nuevo conjunto de variables, llamadas componentes principales, que son ortogonales entre sí y están ordenadas de manera que las primeras capturan la mayor parte de la variabilidad de los datos originales.

El procedimiento es el siguiente. Se normaliza la BD y se calculan los autovalores y autovectores de la matriz de correlaciones de la BD ya normalizada. A partir de los autovalores se obtiene cúanta variabilidad es capaz de explicar cada componente y a partir de los autovectores se obtiene la nueva base ortogonal (Jolliffe, 2002).

En nuestro caso, cómo solo buscamos visualizar los datos, podemos quedarnos o con 2 o con 3 componentes principales.

Aunque parezca que a partir de esta técnica se puede proceder con el proyecto quitando ruido y redundancia, luego tiene un gran inconveniente: la

capacidad de explicar cada componente a la hora de aplicar los resultados a negocio. Si una componente ha agrupado 4 variables porque matemáticamente dan buenos resultados, pero que no se coordinan bien "en el mundo real" o son demasiado complejas, no se pueden crear reglas de negocio.

2.3. Metodología de Clusterización de la Base de Datos (K-Means)

El modelo K-Means es un algoritmo de clustering que tiene como objetivo principal dividir un conjunto de datos en K *clusters* con observaciones similares dentro del *cluster y* lo más dispares posibles entre los *clusters*. Cada *grupo* está representado por su centroide, que es la media de los puntos del mismo. El algoritmo busca que cada dato pertenezca al *cluster* con el centroide más cercano. La finalidad es minimizar la suma de las distancias al cuadrado entre cada dato y su centroide correspondiente.

Algunas de las ventajas de este modelo son su simplicidad, eficiencia y rapidez. El K-Means es fácil de implementar y computacionalmente eficiente, especialmente para grandes conjuntos de datos. Por otro lado, tiene algunas limitaciones. Requiere una selección a priori de K por el científico de datos, es decir, no existe una opción óptima objetiva. Además, es muy sensible a la selección inicial de los centroides, pudiendo afectar significativamente al resultado final. Esta última desventaja se solventa aplicando el algoritmo K-Means++, que busca los centroides más alejados posible unos de otros. El paquete de s*klearn* sí que utiliza esta versión mejorada del algoritmo. Otra desventaja son los *outliers*, que pueden influir considerablemente en los centroides, llevando a una representación distorsionada de los *clusters*. Por esto, se depuran antes de comenzar con este algoritmo.

Para escoger K analizaremos la relación entre el número de *clusters* y la suma de los errores cuadrados dentro de los *clusters.* Esto viene a ser la medida de la variabilidad dentro de cada *cluster*. Como se ha comentado

antes, se calcula como la suma de las distancias al cuadrado de cada punto de datos a su centroide correspondiente.

Para ello, se aplica la regla del codo (*elbow method*). Este procedimiento consiste en realizar los cálculos para diferentes K y así poder analizar el comportamiento del modelo en función de la K aportada (Kodinariya & Makwana, 2013).

3. LENGUAJE DE PROGRAMACIÓN, SOFTWARE, ENTORNO DE DESARROLLO Y ALMACENAMIENTO DE DATOS

Este proyecto, aparte de ser un trabajo de fin de grado (TFG), se implementará en Generali Seguros si aporta resultados concluyentes. Gracias a esto, el proyecto se realiza con las herramientas de trabajo proporcionadas por la empresa. Todos los archivos de datos descargados y creados se almacenan en un *Data Lake* de Microsoft Azure. El desarrollo del código se realiza a partir de *notebooks* en un entorno en la nube llamado Databricks (pertenece a Microsoft Azure). Databricks está basado en Apache Spark, pero acepta diferentes lenguajes de programación como R y Python. Apache Spark es un motor para procesar datos de magnitudes abrumadoras de forma rápida y sencilla. El proyecto en su gran mayoría está desarrollado a partir de *PySpark (*PySpark Overview, 2024), que es un paquete que nos permite trabajar con los *dataframes* de Spark en Python. Siempre que se crea un *dataframe* (DF) en Spark, se almacena en formato Parquet, ya que el acceso a las columnas es más rápido. Además, la librería *pandas* (McKinney, 2010) *también* se utiliza durante el proyecto. Al inicio de la creación de la BD para poder manejar archivos Excel cómodamente y más adelante para poder utilizar el paquete *sklearn*, que trabaja con los DF de *pandas* para aplicar técnicas de *machine learning* en Python. De manera auxiliar, se ha utilizado Microsoft Excel para diferentes tareas como la creación de tablas descriptivas o gráficos. Todo el código desarrollado durante el proyecto será almacenado en GitHub en formato de *notebooks* previamente ejecutados.

4. CREACIÓN DE LA BASE DE DATOS

La BD consiste en una unión de diferentes archivos pertenecientes al INE (INE, 2024) junto con información sobre el Precio del Metro Cuadrado extraída de la página web RealAdvisor.

Consta de 10554 observaciones y 25 variables contando con el identificador de las observaciones.

4.1. Datos obtenidos del INE

El apartado donde se encuentran los archivos del INE se llama **Atlas de Distribución de Renta de los Hogares** (INE, 2021). En esta sección la información viene agregada por unidades de terreno (municipio, distrito, sección censal, etc.). Buscando el mayor nivel de granularidad posible, se ha escogido la SC, que es la unidad más pequeña disponible.

Los datos por SC están almacenados en distintos archivos por provincias que a su vez también contienen observaciones de los municipios y distritos, y que más adelante serán eliminados. De los archivos disponibles se han escogido cuatro: **Indicadores de Renta Media y Mediana, Indicadores Demográficos, Distribución por Fuente de Ingresos y Porcentaje de Población por Unidad de Consumo Por Encima de un Umbral Determinado.**

Estos archivos se descargan en formato .xlsx de Excel. Se toma una medida sistemática que consiste en renombrar uno a uno siguiendo un mismo patrón y así evitar futuros problemas en la lectura de los archivos durante el bucle *for*. Las provincias siempre deben ir con mayúscula y tildadas. Los espacios son representados con '_' y según que archivo sea, le acompañará la siguiente desinencia: renta_media_mediana, FI, IUC e ID. Después se suben al *Data Lake* de Azure, para cargarlos en *Databricks* y poder manipularlos. Por ejemplo, los archivos de la provincia de Málaga serán: Málaga_ renta_media_mediana, Málaga_FI, Málaga_IUC, Málaga_ID.

Para poder generar la BD, tenemos que unir estos 208 archivos (52 provincias * 4 tipos de archivo). Para ello, utilizamos primero la librería *pandas* de Python y así estructurar todos los archivos de Excel de forma que sea viable concatenar cada grupo de archivos de una misma clase creando cuatro DF en total.

Estos archivos, aunque pueden parecer simples documentos de Excel, requieren tratamiento, ya que existen celdas combinadas, información no deseada, columnas innecesarias, etc. Esto obliga a analizar exhaustivamente los documentos antes de ponerse manos a la obra. Por ejemplo, todos los archivos contienen a pie de página anotaciones sobre alguna de las variables y muchas más columnas de las deseadas (información irrelevante de todos los años anteriores).

Para poder desarrollar un programa general para la lectura y estructuración de archivos, en el código existen varias líneas dedicadas a "casos base", es decir, ensayos de como estructurar bien los DF trabajando con una sola provincia (un único archivo), y más adelante se desarrolla el código genérico para esa clase de archivo a partir de un bucle *for*.

El procedimiento para unir los archivos de mismas características comparte la misma idea principal en los cuatro casos, que consiste en generar un DF vacío declarando las columnas correspondientes, y, utilizando la función *concat* de la librería *pandas,* añadir de forma iterada al DF todos los archivos previamente reestructurados. En la reestructuración de los datos, cabe destacar que los valores NA y los valores que venían con uno o dos puntos han sido reemplazados por ceros. Se ha comprobado que esto no supone un problema de identificación entre valores cero faltantes y valores cero numéricos, ya que los segundos no se dan en las variables. Además, todas las variables numéricas han sido transformadas a tipo *float* para evitar errores durante la transformación a Spark.

Una vez obtenidos los cuatro archivos como DF de Spark, los unimos usando la función *join* de la librería *Pyspark.* Parece obvio que, viniendo toda la información del INE y del mismo apartado **(Atlas de Distribución de Renta de los Hogares**), el número de secciones censales debería coincidir en los cuatro DF, pero sorprendentemente no. Por ejemplo, el DF que

contiene la información sobre la renta media y mediana, contiene más observaciones que el de ID. Por esto, se realiza un *full join* en todas las uniones, evitando así pérdida de información.

Todos los municipios, distritos y secciones censales están codificados. Más concretamente, el código de un municipio son cinco dígitos, el de un distrito son siete, y el de una SC son diez. En un distrito, de estos siete números, los cinco primeros coinciden con el código del municipio al que pertenecen y los dos restantes indican el distrito. Lo mismo ocurre con las SC. Los cinco primeros son del municipio, los dos siguientes son del distrito y los últimos tres marcan la SC.

Esta aclaración es importante debido a cómo está generada la variable Lugar en los archivos del INE, que no es más que el identificador de cada observación y la que se ha utilizado a la hora de realizar las uniones. El problema que tenemos con esta variable y nuestra BD es que existen observaciones de municipios y distritos que son irrelevantes, ya que se busca generar una BD exclusivamente de SC. Sólo son de utilidad las observaciones de secciones censales, ya que son las que contienen la información a estudiar.

Como se desea trabajar sólo con las secciones censales, se debe filtrar la BD eliminando estas observaciones. Para ello, debemos crear una nueva variable llamada Código transformando la variable Lugar.

En la Tabla 1, la variable Lugar se rige por la siguiente estructura (separada únicamente por espacios):

Código	Topónimo	Distrito o Sección	Tramo de Código
03904	San Isidro		
0390401001	San Isidro	sección	001

Tabla 1: San Isidro

Consta de cuatro partes: Código, Topónimo, Distrito o Sección y Tramo de Código. La primera fila es una observación del municipio de San Isidro (esta observación será una de las eliminadas, ya que sus datos son de municipio). La segunda fila es una observación de la SC 001 de San Isidro. Como se puede apreciar, la tercera parte de la variable Lugar aclara si es distrito o sección (en caso de ser un municipio la observación termina con el

topónimo como en el primer caso). La última parte es el Tramo de Código (001) que indica el distrito o sección correspondiente. En el ejemplo de la Tabla 1 se ve como la sección 001 de San Isidro comparte los cinco primeros dígitos del municipio de San Isidro. Además, también se puede extraer que esa SC pertenece al distrito 01 de San Isidro (dígitos de sexta y séptima posición).

Sabiendo esto, se puede filtrar la BD por las observaciones que contienen la palabra "sección" y después crear una nueva columna llamada "Código" con los 10 primeros caracteres de la variable Lugar, siendo esto el identificador de cada sección. Además, si se desease, también se puede crear una BD únicamente con municipios o únicamente con distritos.

Todas las variables han sido renombradas por motivos de comodidad y evasión de errores futuros con espacios y tildes. Más adelante, en el subapartado 4.5, se introduce un diccionario de datos con dichas transformaciones.

La variable Renta Mediana por Hogar está completamente vacía. Durante la ejecución del resto de apartados, se arrastra esta variable pese a no aportar información. Esta forma de proceder se debe a que al ser un proyecto que puede ser actualizado cada año, mantener la variable disponible es buena idea por si acaso en años posteriores el INE decide completarla. De igual manera, la variable del Porcentaje de Población Española tiene una gran suma de valores perdidos o en cero. Igualmente la arrastramos hasta el final de la creación de la BD y ya en el análisis descriptivo se decidirá si merece la pena trabajar con ella o no.

Por la misma razón, en esta BD, hemos incluido todas las posibles medidas de la renta teniendo así un abanico más grande de opciones. Esto parece ser un inconveniente, y es que, obviamente, si utilizamos todas en un modelo, van a estar muy correlacionadas. De todas formas, seleccionar la medida de renta que más atractiva parezca para el estudio y dejar de contar con el resto será la manera a proceder durante la modelización.

4.2. Precio Por Metro Cuadrado

Por otro lado, una posible variable de interés y con potencial es el Precio por Metro Cuadrado. Navegando por Internet, encontré la página RealAdvisor, que proporciona esta información para toda España. Para acceder a los datos, se programa un código de *web scraping* utilizando principalmente la librería *BeautifulSoup* (Richardson, 2004) de Python, obteniendo así la información a partir del código .html de la página.

Primero, accediendo a cada enlace de cada provincia, *scrapeamos* los enlaces de cada municipio de la provincia correspondiente y sus códigos postales. Una vez capturados, accedemos a estos enlaces y *scrapeamos* la información disponible: precio para piso y precio para chalet. Esta información queda grabada en un fichero .txt que leemos en Excel.

La razón de necesitar los dos precios es porque en las zonas rurales donde no exista precio por piso se tomará el precio del chalet. En caso de desconocer el precio por piso, se mantiene el precio para chalet como precio definitivo.

El *scrapeo* almacena los municipios con su código postal en la misma columna (igual que la variable Lugar en los archivos del INE). Para poder diferenciar las observaciones, en la Tabla 2 se muestra el procedimiento: se extrae de la columna el CP y se borran las posibles observaciones duplicadas.

Columna Original	Código Final
(28250) Torrelodones	28250
(28270) Galapagar	28270

Tabla 2: Extracción de código postal

Aunque realizar *web scraping* no es una práctica ilegal, muchas páginas bloquean la entrada continua desde un mismo dispositivo ya que lo detecta como un *bot*. Para evitar esto, en el código definimos una lista de dispositivos llamada *headers.* Al acceder continuamente a la página de RealAdvisor con un *header* diferente, no se nos bloquea el acceso.

4.3. Ponderación de la Base de Datos

Ahora debemos afrontar el siguiente problema. Las observaciones del INE se almacenan por SC y las del Precio por Metro Cuadrado por CP. Una SC es la unidad geográfica más pequeña para la que el INE recoge datos. Por otro lado, el CP es la unidad geográfica que utilizan los servicios de mensajería (como correos) y va incluido en todas las direcciones del país. Para poder unir estas BD en una sola, se debe escalar la información de las SC a CP, que pasará a ser nuestra unidad geográfica de observación. Para ello, utilizamos otra BD llamada El Callejero. En este caso, la BD ya estaba creada por Javier González y almacenada en el *Data Lake*. Basta con cargarla en el *notebook* para poder trabajar con ella.

El Callejero aporta la relación que existe entre una SC y su CP. Existen tres casos relacionales entre ambas. El primero y más obvio, es cuando un CP sólo contiene una SC (relación 1 a 1). El segundo, que son en los que se realizarán las próximas modificaciones, es cuando un CP abarca varias SC (relación n a 1). Por último, y menos obvio, es que existen casos en los que una SC pertenece a más de un CP (relación 1 a n).

Realizando un *join* sobre la BD de SC, se añade la columna con el código postal. Para los casos en los que la relación es 1 a 1, simplemente debemos dejar la observación tal y como está. En cambio, para los casos en los que la relación es n a 1, se pondera la variable en función de su población. Por ejemplo, la Renta Neta Media (RNM) para un CP se calcularía como se indica en la Ecuación 1:

$$\frac{RNM_{\text{sección }1} * población\ 1 + \ RNM_{\text{sección }2} * población\ 2 + \cdots + \ RNM_{\text{sección }n} * población\ n}{población\ 1 + población\ 2 + \cdots + población\ n}$$

Ecuación 1: Ponderación de variables

La estructura de la fórmula de la ecuación 1 se aplica a todas las variables siempre que se cumpla la relación adecuada.

Existe una tercera posibilidad bastante poco común donde la relación será al contrario (1 a n). La existencia de estos casos no interfiere con la técnica de ponderación, ya que, al fin y al cabo, a este tercer caso se le aplicará una de las dos opciones de ponderación anteriormente expuestas en función de sus características. Analicémoslo con un ejemplo sencillo: existe la SC 0104301002 con dos CP 01320 y 01322. La primera posibilidad es que estos CP pertenezcan al primer tipo (relación 1 a 1). Por lo tanto, los dos CP recibirán las variables de la misma SC sin causar problema alguno. En caso de que alguno de los CP también pertenezca a otra SC (relación n a 1), tampoco habrá problema. Se calculará una media ponderada por la población de las dos SC y obtendremos los valores de dicho CP.

4.4. Imputación por *smoothing*

La BD ya está casi terminada. Después de unir todos los datos, necesitamos imputar los valores faltantes para poder empezar a trabajar con ella descriptivamente. La técnica a utilizar es un *smoothing* a partir de los CP adyacentes. Al igual que El Callejero, la relación entre los CP y sus adyacentes está almacenada en un DF en el *Data Lake* y fue creado por Javier González.

La técnica consiste en completar los datos de una observación a partir de la información que se tiene en nuestra propia BD sobre todos sus "vecinos". Se considera CP adyacente (vecino) a todo CP que limite geográficamente con un CP concreto. Para que esta técnica sea más robusta, en caso de conocer la información de varios vecinos, se realiza una media y este será el valor imputado. También cabe la posibilidad de que no exista información de ningún CP adyacente para cierta variable. En estos casos, se imputa con la media de la variable por provincia. En caso de no tener información sobre una provincia en alguna columna, la imputación se realizará a partir de la media de la comunidad autónoma (CA). Por ejemplo, Guipúzcoa, que es la provincia con código 20, no tiene datos de las variables FI_salario, FI_pensiones y FI_desempleo. En caso de tampoco existir información de la CA, se imputa con la media de todo el conjunto de datos.

En la Tabla 3, se muestra el porcentaje de valores faltantes por variable.

Variable	Porcentaje de *Missings*
RNM_persona	0%
RNM_hogar	0%
RM_unidad_consumo	3%
R_med_unidad_consumo	3%
R_med_hogar	100%
RBM_persona	0%
RBM_hogar	0%
pob	0%
edad_media_P	0%
menor_18_P	0%
mayor_64_P	0%
M_tamano_hogar	0%
hogares_uni_P	0%
pob_esp_P	47%
FI_salario	4%
FI_pensiones	5%
FI_desempleo	5%
FI_otras_prestaciones	5%
FI_otros_ingresos	4%
pob_sup_140_med_P	23%
pob_sup_160_med_P	23%
pob_sup_200_med_P	23%

Tabla 3: Porcentaje de Valores Faltantes por Variable

4.5. Diccionario de Datos

Para arrojar claridad sobre esta BD, se desarrolla un archivo Excel con las variables con su nombre original en la Tabla 4. La Tabla 5 recoge el código de abreviaciones utilizado para nombrar las variables. Para algunas variables, se incluye además una aclaración adicional en la Tabla 6.

VARIABLES	
Codigo	
RNM_persona	Renta neta media por persona
RNM_hogar	Renta neta media por hogar
RM_unidad_consumo	Media de la renta por unidad de consumo
R_med_unidad_consumo	Mediana de la renta por unidad de consumo
R_med_hogar	Renta mediana por hogar

RBM_persona	Renta bruta media por persona
RBM_hogar	Renta bruta media por hogar
edad_media	Edad media de la población
menor_18_P	Porcentaje de población menor de 18 años
mayor_64_P	Porcentaje de población de 65 y más años
M_tamano_hogar	Tamaño medio del hogar
hogares_uni_P	Porcentaje de hogares unipersonales
pob	Población
pob_esp_P	Porcentaje de población española
FI_salario	Fuente de ingreso: salario
FI_pensiones	Fuente de ingreso: pensiones
FI_desempleo	Fuente de ingreso: prestaciones por desempleo
FI_otras_prestaciones	Fuente de ingreso: otras prestaciones
FI_otros_ingresos	Fuente de ingreso: otros ingresos
pob_sup_140_med_P	Por encima 140% mediana
pob_sup_160_med_P	Por encima 160% mediana
pob_sup_160_med_P	Por encima 200% mediana
Precio_m2	Precio por metro cuadrado

Tabla 4: Variables

ABREVIACIONES EN LAS VARIABLES
R: renta
N: neta
M: media
med: mediana
B: bruta
P: porcentaje
FI: fuente de ingresos
pob: población

Tabla 5: Abreviaciones

M_tamano_hogar	Número de personas que viven juntas
hogares_uni_P	Porcentaje de personas que viven solas
pob_sup_140_med_P	Porcentaje de población local por encima del umbral indicado
pob_sup_160_med_P	Porcentaje de población local por encima del umbral indicado
pob_sup_160_med_P	Porcentaje de población local por encima del umbral indicado

Tabla 6: Descripción de variables

5. ANÁLISIS DESCRIPTIVO DE LA BASE DE DATOS

Como ya se comentó en el subapartado 4.1, existían dos variables candidatas a retirase del estudio por la poca cantidad de datos que contenían. Definitivamente han sido suprimidas. La primera es Renta Mediana por Hogar, debido a que está completamente vacía. La segunda es Porcentaje de Población Española, que, aunque haya sido imputada junto al resto, contenía un 47% de observaciones nulas. Esto hace que la variable tenga muchos valores imputados a partir de la media de la provincia o incluso de la CA, relegándola a un nivel de calidad inferior al del resto de variables.

La BD final cuenta con 10554 observaciones y 22 columnas de las cuales 21 son variables numéricas, siendo la columna restante una columna identificadora de los códigos postales únicos.

5.1. Resumen Estadístico de las Variables

Para una lectura más cómoda de la Tabla 7, que es el resumen descriptivo de las variables, se ha dividido en 7 secciones.

Las variables sobre la Renta (Secciones 1 y 2), como siguen diferentes criterios, podría parecer que no se comportan igual. En el subapartado 5.2 se comprueba que tienen distribuciones similares. En estas variables, destacan mucho los extremos. La diferencia entre el mínimo y el primer cuartil es muy elevada en todas menos en la Renta mediana por Unidad de Consumo y la Renta Bruta Media por Persona. De la misma manera, la diferencia entre el tercer cuartil y el máximo en todas las variables también es muy elevado.

Esto es un indicio de que pueden existir muchos *outliers* con este tipo de variables.

Las variables Demográficas están en las Secciones 3 y 4. La gran mayoría de los CP tienen una Población entre 1500 y 4000 habitantes. El máximo de la variable es muy elevado, y la diferencia entre la mediana y la media también es muy grande. Esto quiere decir que esta variable contiene observaciones con valores muy altos, sesgando así la media. Las variables relacionadas con el hogar y la edad tienen un comportamiento standard.

En la Sección 5 y 6 están recogidas las variables de Fuentes de Ingresos. La Fuente de Ingresos: Salario, es lógico que se comporte de forma similar a las Rentas, y, por tanto, tiene también muchos valores fuera del rango intercuartílico. La otra variable que destaca en este mismo sentido es la de Fuente de Ingresos: Otros Ingresos.

En la Sección 6 y 7, están recogidas las variables de Porcentaje de Población Local por Encima de un Umbral Indicado. Aquí destaca que cuanto más alto es el umbral (140%, 160% y 200%), más disminuye la media y más aumenta la desviación típica.

Por último, también en la Sección 7, está el Precio del Metro Cuadrado. Esta es otra variable con comportamiento muy parecido al de las Rentas.

summary	RNM_persona	RNM_hogar	RM_unidad_consumo
count	10554	10554	10554
mean	12562.0365	29713.727	18022.75963
stddev	2471.65308	7239.86231	3932.997795
min	5343	14650	4179.83815
25%	10890.7231	24899	15538
50%	12296	28566	17452.05385
75%	13946.5649	32989	19885
max	31304.5895	88769	51150.23837

Sección 1 (Tabla 7)

summary	R_med_unidad_consumo	RBM_persona	RBM_hogar
count	10554	10554	10554
mean	16286.028	14850.5616	35214.32132
stddev	3406.15385	3557.0638	10336.2354
min	3927.55299	6617.76988	15475
25%	14006.0137	12431	28487
50%	15750	14312.6518	33263
75%	18042.7692	16681	39423.18151
max	38850	45383.5963	131719

Sección 2 (Tabla 7)

summary	pob	edad_media	menor_18_P
count	10554	10554	10554
mean	5418.48414	48.4564999	13.03632253
stddev	11143.3198	5.9455127	5.146477593
min	3	29.5196984	0.6
25%	642	43.8	9.1
50%	1489	47.7	13.3
75%	4081	53	16.82078775
max	133550	74.3	34.20385722

Sección 3 (Tabla 7)

summary	mayor_64_P	M_tamano_hogar	hogares_uni_P
count	10554	10554	10554
mean	27.2070887	2.34300869	34.82368934
stddev	9.54851056	0.30741525	9.487767119
min	1.73805623	1.28	10.3
25%	19.8607606	2.1237653	27.8
50%	25.9	2.33824313	33.8
75%	33.8	2.55458856	41.1
max	74.5	3.74	90

Sección 4 (Tabla 7)

summary	FI_salario	FI_pensiones	FI_desempleo
count	10554	10554	10554
mean	7634.74363	3742.79671	464.244042
stddev	2661.79649	1258.16705	190.2852747
min	1297.23002	450.110966	41.55091384
25%	5759	2830.78058	347.7654179
50%	7168.58563	3617	427
75%	9067.66137	4517.32399	527
max	26890.5287	8623	1505

Sección 5 (Tabla 7)

summary	Fl_otras_prestaciones	Fl_otros_ingresos	pob_sup_140_med_P
count	10554	10554	10554
mean	658.581146	1982.39155	23.28461671
stddev	189.936108	1237.18803	11.39703404
min	45.0405983	72	0.349593919
25%	534.15493	1223	15.2
50%	645.696203	1656	21.24826347
75%	766.971802	2357	29.17703552
max	1336	13848	77.77639913

Sección 6 (Tabla 7)

summary	pob_sup_160_med_P	pob_sup_200_med_P	PRECIO_m2_FINAL
count	10554	10554	10554
mean	16.2489649	7.87297067	1174.964694
stddev	9.58711951	6.44640403	799.3026606
min	0.39483549	0.06991878	8
25%	9.6	3.84321209	709
50%	14.3097978	6.21514176	933.4042553
75%	20.3794263	9.57833035	1366
max	72.6581923	60.5173825	9213

Sección 7 (Tabla 7)

Tabla 7: Resumen Descriptivo de las Variables

5.2. Distribución de las Variables

En el Anexo se han añadido los histogramas y diagramas de caja y bigotes para todas las variables.

Tanto en los histogramas como en los diagramas de caja y bigotes de las Rentas se aprecia que todas siguen una distribución muy similar, con muchos valores atípicos (sobre todo mayores). De las variables demográficas sólo destaca Población, que tiene la mayoría de sus observaciones con valores muy parecidos, pero tiene muchísimos *outliers*. De las difernetes variables del grupo Fuente de Ingresos destacan Fuente de Ingresos: Salario y Fuente de Ingresos: Otros Ingresos. Las dos tienen distribuciones parecidas a las de las Rentas. Las tres variables de Porcentaje de Población Local por Encima de un Umbral Indicado y el Precio por Metro Cuadrado tienen las 4 una gran masa de *outliers*.

Contando el número de observaciones con valores atípicos se obtienen un total de 956 *outliers* para el conjunto total de datos.

5.3. Análisis de Correlación Lineal Entre Variables

A partir de la matriz de correlaciones, graficamos un mapa de calor expuesto en la Ilustración 1. Se considerará que existe correlación lineal entre dos variables si se supera el valor de 0.6. Como era de esperar, se observa una clara correlación entre las diferentes medidas de las Rentas. Las variables de las Rentas, Fuente de Ingresos: Salario y las 3 variables de Porcentajes de Población Superior a un Umbral Determinado están también correlacionadas linealmente. Por otro lado, las variables Demográficas también están relacionadas entre sí. Más concretamente, las tres variables que hacen referencia a la edad están muy correlacionadas entre sí. Hay que destacar que estas mismas tienen una correlación suave con la variable de Fuente de Ingresos: Pensiones, ya que es un ingreso que sólo recibe la tercera edad. La variable del Tamaño del Hogar Medio está muy relacionada con el Porcentaje de Hogares Unipersonales.

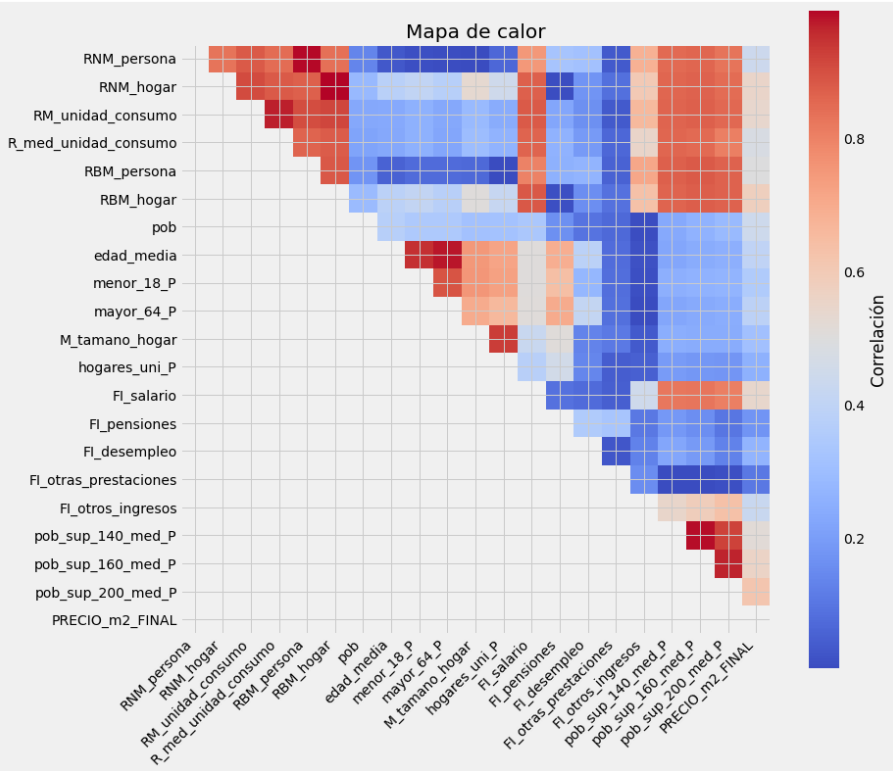


Ilustración 1: Mapa de calor

Parece lógico deshacerse de algunas de las variables mencionadas y mantener otras para conseguir variables independientes. De todas las opciones de las Rentas, se mantiene la Renta Mediana por Unidad de Consumo. Esta decisión tiene dos causas. La primera es que la métrica de los salarios es más precisa a partir de la mediana, ya que, con la media, los salarios muy elevados sesgan el resultado final y es muy común que esto ocurra. La segunda causa se fundamenta en el uso de escalas de equivalencia. Estas escalas permiten una comparación más precisa del gasto o ingreso entre hogares de diversos tamaños y composiciones. Según la teoría de economías de escala, el incremento en el tamaño del hogar no siempre resulta en un aumento proporcional del gasto para mantener la misma pauta de consumo. Esto se debe a que existen gastos compartidos que no crecen de manera proporcional al número de miembros del hogar, como los relacionados con la vivienda.

Por otro lado, la teoría de unidades de consumo equivalentes reconoce que las pautas de consumo de los niños difieren de las de los adultos. Al considerar estas diferencias, las escalas de equivalencia pueden proporcionar una visión más precisa del nivel de bienestar relativo entre hogares con distintas composiciones familiares. Esto quiere decir que son una herramienta más robusta y sensible para comparar el nivel de ingresos entre hogares, teniendo en cuenta tanto su tamaño como su composición (INE, s.f.).

Al priorizar la variable Renta Mediana por Unidad de Consumo, las tres variables de Porcentaje Superior a un Umbral Determinado y Fuente de Ingresos: Salario no pueden mantenerse en el estudio.

Por otro lado, o la variable Tamaño del Hogar Medio o el Porcentaje de Hogares Unipersonales se ha de eliminar. Se ha optado por mantener la primera por decisión de negocio (aportará más información para la venta de diferentes productos saber el Tamaño del Hogar Medio, pudiendo ver así zonas más familiares). Como las variables sobre la edad de la población están relacionadas entre ellas y con el Tamaño del Hogar Medio, se prescinde de ellas también.

Una vez realizada esta selección manual que toma como criterio la existencia de correlación por encima del 0.6, el conjunto de variables definitivo queda como muestra la Tabla 8:

VARIABLES	
cp	Código Postal
R_med_unidad_consumo	Mediana de la renta por unidad de consumo
M_tamano_hogar	Tamaño medio del hogar
pob	Población
FI_pensiones	Fuente de ingreso: pensiones
FI_desempleo	Fuente de ingreso: prestaciones por desempleo
FI_otras_prestaciones	Fuente de ingreso: otras prestaciones
FI_otros_ingresos	Fuente de ingreso: otros ingresos
Precio_m2	Precio por metro cuadrado

Tabla 8: Variables definitivas

5.4. Tratamiento de *Outliers*

Deshacerse de los valores atípicos es crucial antes de trabajar con un algoritmo como el K-Means, ya que el cálculo de los centroides se realiza en función de las distancias, haciéndolo así muy sensible a los *outliers*. Para ello, aplicamos el algoritmo DBSCAN.

Ya que el algoritmo funciona a partir de distancias, el primer paso a dar es normalizar las variables, evitando así que variables con unidades muy dispares distorsionen los resultados.

Como en algoritmos no supervisados de clusterización la selección de parámetros queda a elección del científico de datos, se realiza un *grid search* para analizar qué combinación paramétrica es más conveniente.

Con el *grid search* se visualiza el número de *clusters* que se crearán y cuantas observaciones contiene cada *cluster* (aunque el relevante es el primero, ya que debería contener el grueso de los datos).

Para acotar los valores que se introducirán en el *grid search*, a partir de la función *NearestNeighbors,* se obtienen las observaciones más cercanas

a cada una de ellas y las distancias correspondientes. Al graficarlo se obtiene la Ilustración 2.

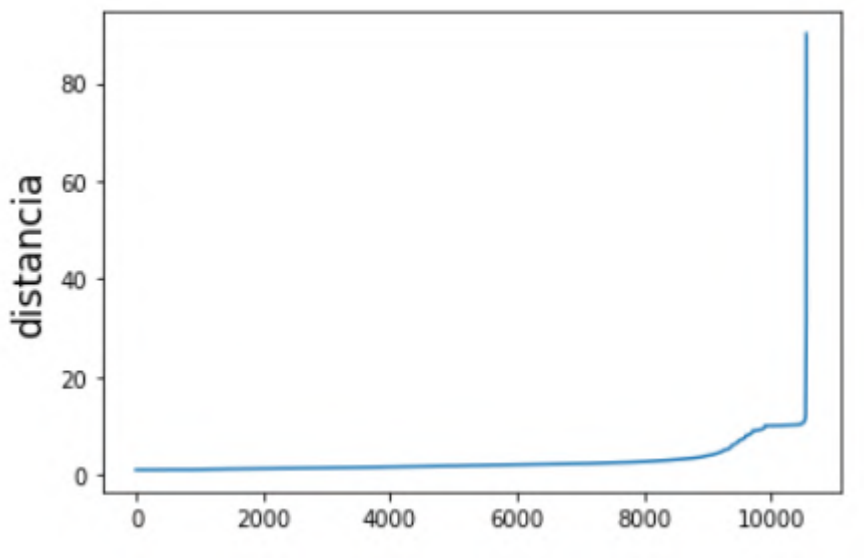


Ilustración 2: Distancias entre observaciones

Como se aprecia en la ilustración 2, las distancias preferentes se encuentran entre 3 y 10 y, por ende, el parámetro ε recorrerá valores desde 3 hasta 10 de décima en décima (3.1, 3.2, …, etc.) en el *grid search*. Así mismo, el número mínimo de puntos (min_pts), tomará valores desde 6 hasta 15. Con este proceso, se busca una tupla de parámetros que consiga generar un *cluster* principal suficientemente grande y que además el resto de *clusters* creados sean grupos de *outliers*. Sabiendo que durante el análisis descriptivo se han detectado alrededor de 960 *outliers* y con 10554 observaciones, sólo contemplaremos los casos en los que el *cluster* principal supere las 9000 observaciones (manteniendo un amplio margen de 500 observaciones aproximadamente).

	eps	min_samples	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
0	3	6	-0.837258	13.944864	1.8
1	3	7	-0.535276	16.319984	0.59
2	3.1	6	-0.794346	4.960815	5.28
3	3.1	7	-0.798465	11.469546	2.15
4	3.2	7	-0.870393	6.88788	4.51
5	3.3	7	-0.820472	5.833296	6.05
6	3.4	7	-0.762396	5.323283	7.13
7	4	8	-0.699044	16.427902	1.81
8	4.1	8	-0.8063	5.425897	7.59
9	4.1	9	-0.668648	12.746792	3.31
10	4.2	9	-0.860739	5.232631	7.27
11	4.3	9	-0.827289	5.594249	7.42
12	4.4	9	-0.803048	8.535534	8.83
13	4.5	9	-0.76682	6.771042	10.08
14	4.6	9	-0.729584	29.556536	10.39
15	5	10	-0.603438	14.581031	3.41
16	5.1	10	-0.826837	5.398644	8.73
17	5.1	11	-0.825902	9.814683	8.57
18	5.2	10	-0.736657	3.883362	11.71
19	5.2	11	-0.844792	7.023021	10.64
20	5.3	11	-0.854885	5.524224	11.88
21	5.4	11	-0.823299	4.194378	13.38
22	5.5	11	-0.798321	3.817096	13.13
23	5.6	11	-0.779407	3.830867	13.13
24	5.7	11	-0.769935	3.905405	13.59
25	5.8	11	-0.756885	3.779788	13.79
26	5.9	11	-0.745775	3.620201	14.38
27	6	11	-0.7279	3.580961	14.2
28	6	12	-0.422599	33.427952	0.13
29	6.1	12	-0.844322	6.359548	12.01
30	6.1	13	-0.768598	14.667573	9.14
31	6.2	12	-0.812338	4.228133	15.41
32	6.2	13	-0.852197	7.733939	14.31
33	6.3	12	-0.777332	4.026617	15.64
34	6.3	13	-0.841133	6.085837	17.26
35	6.4	12	-0.75774	3.977466	15.68
36	6.4	13	-0.825541	5.307814	17.49
37	6.5	12	-0.743239	3.852523	16.24
38	6.5	13	-0.823443	4.848409	18.06
39	6.6	12	-0.727828	4.065684	15.36
40	6.6	13	-0.820584	4.804494	17.98

41	6.7	13	-0.811686	4.79271	17.61
42	6.8	13	-0.808587	4.680634	17.54
43	6.9	13	-0.809137	4.679889	17.22
44	7	13	-0.801608	4.382116	17
45	7	14	-0.41992	31.581004	0.13
46	7.1	13	-0.766177	4.146587	15.31
47	7.1	14	-0.81894	5.663429	16.23
48	7.2	13	-0.744035	3.765731	15.6
49	7.2	14	-0.797704	5.457995	15.86
50	7.3	13	-0.712878	3.738836	15.98
51	7.3	14	-0.784929	4.866	17.06
52	7.4	14	-0.774755	4.782868	16.24
53	7.5	14	-0.770103	4.635512	15.29
54	7.6	14	-0.762906	4.678436	15.92
55	7.7	14	-0.76029	4.527875	16.23
56	7.8	14	-0.754237	4.337136	16.21
57	7.9	14	-0.754237	4.337136	16.21
58	8	14	-0.75087	4.237461	16.44
59	8.1	14	-0.72174	4.071672	16.19

Tabla 9: Evaluación de parámetros

En la Tabla 9, a partir del Índice de Silhouette se alcanza una conclusión clara: ninguna combinación de parámetros arroja un resultado sólido.

Aun así, con los mejores resultados de los Índices Davies-Bouldin y Calinski-Harabasz se intenta comprobar si los *clusters* diferentes del principal están siendo bien escogidos. Para ello, se visualizan gráficamente los datos realizando un Análisis de Componentes Principales, pudiendo así graficar en 3D las observaciones.

Aunque se han realizado varias pruebas gráficas con diferentes combinaciones paramétricas en el *notebook*, en la Tabla 9 están marcados en

azul los dos resultados que se van a mostrar las Ilustraciones 3 y 4 respectivamente:

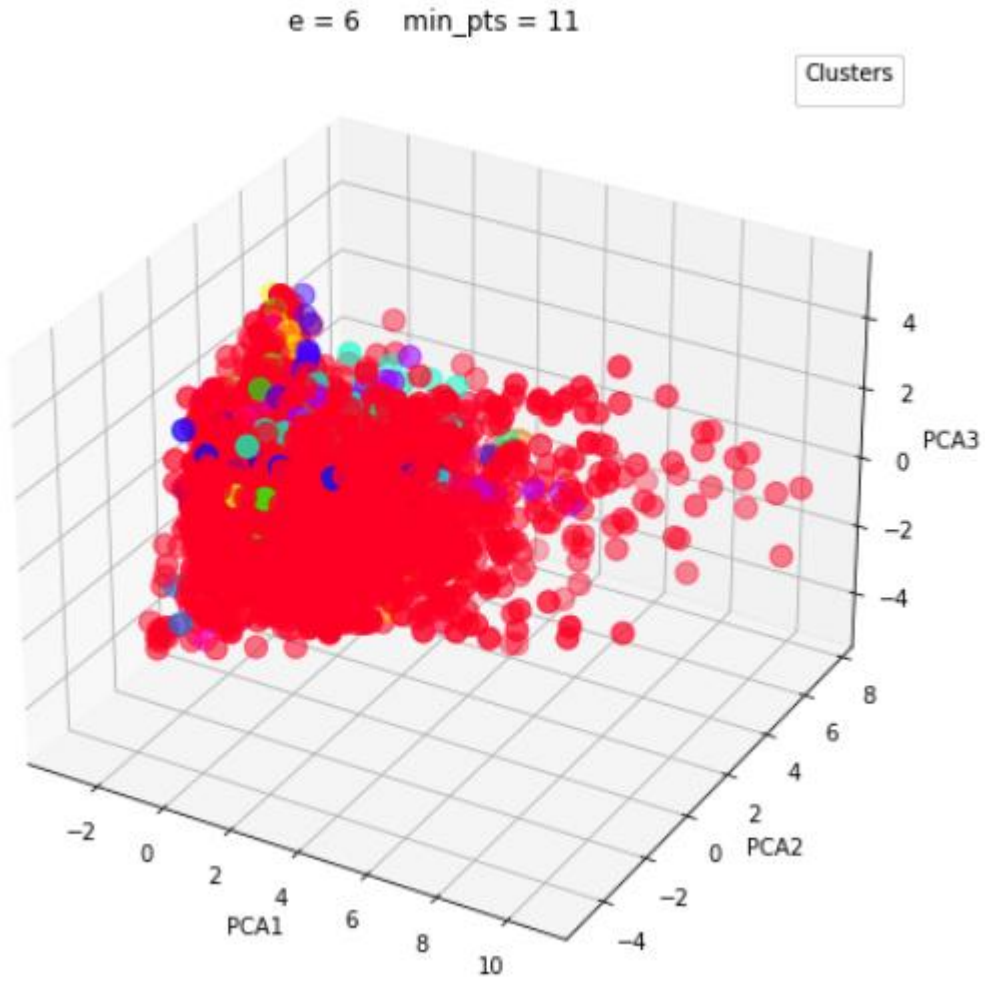


Ilustración 3: PCA clusterizado (6,11)

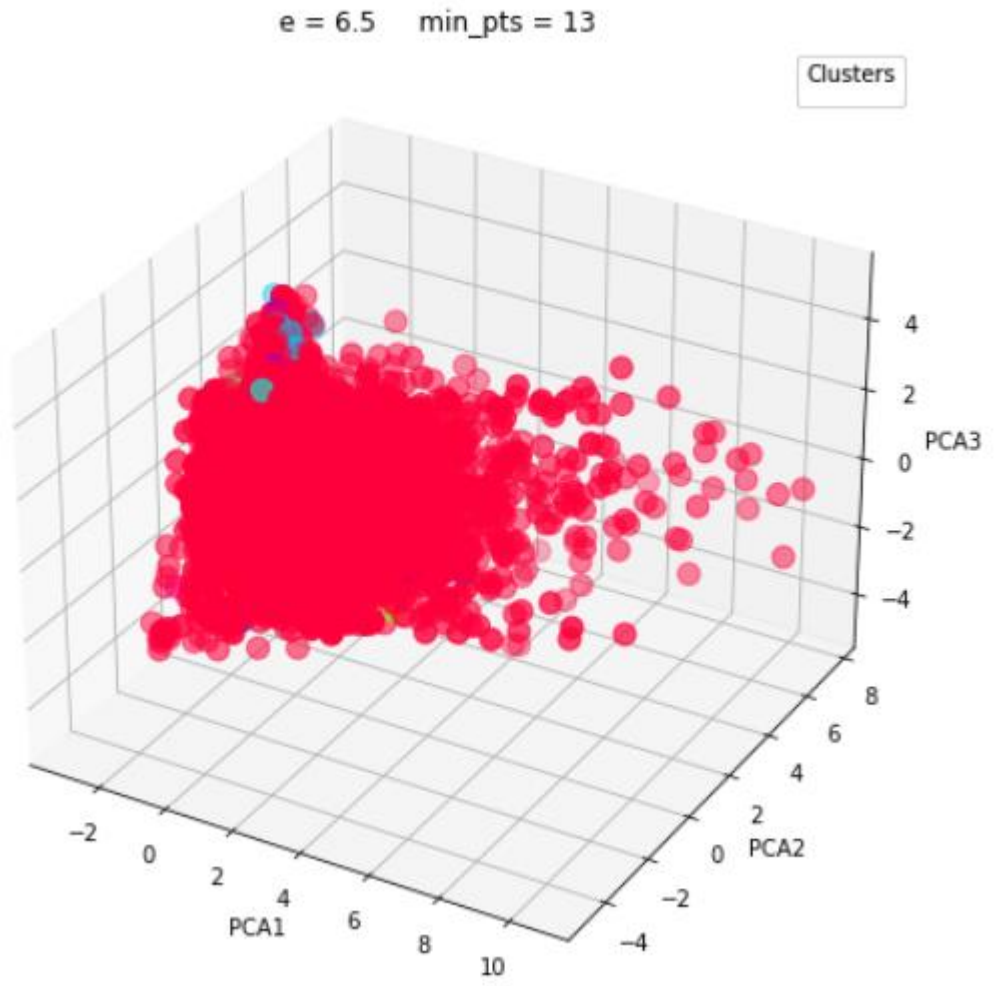


Ilustración 4: PCA clusterizado (6.5,13)

Se aprecia claramente en las Ilustraciones 3 y 4 cómo la zona derecha del gráfico contiene observaciones que son casos atípicos y en ninguna de las ilustraciones se los categoriza como un *cluster*, manteniéndose en el *cluster* principal. *Testeando* con otras combinaciones en las que el *cluster* principal es más pequeño, se consiguen resultados parecidos. Se obtienen muchos más *clusters,* pero los *outliers* se mantienen no detectados.

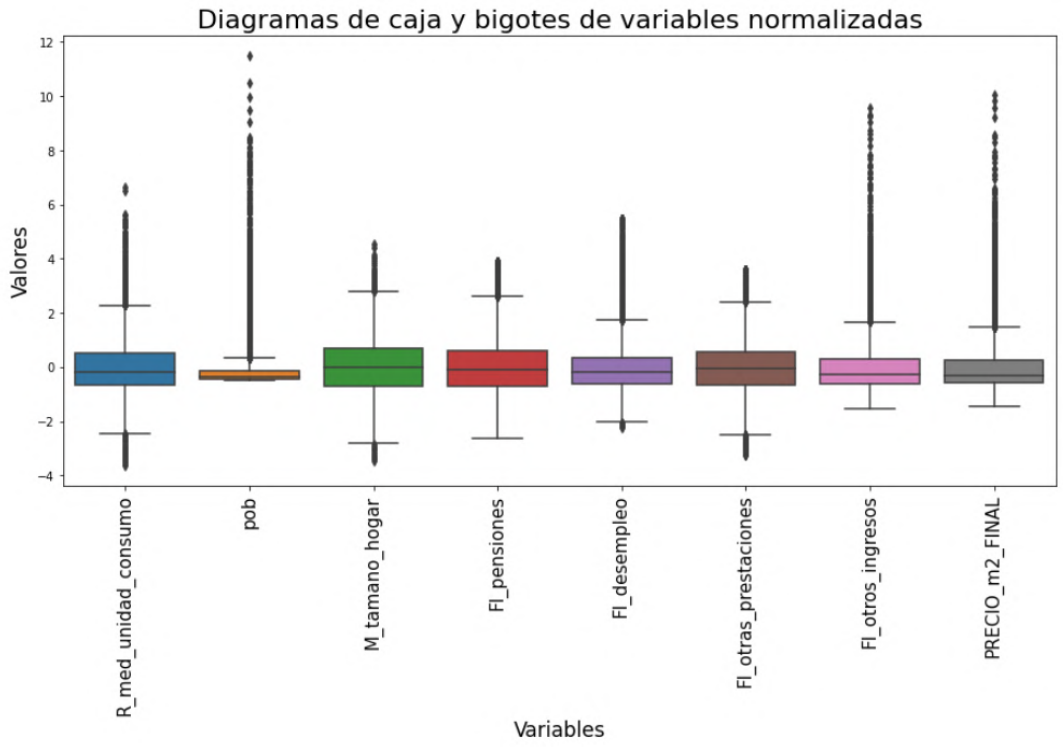


Ilustración 5: Diagramas de cajas normalizados

A partir de la ilustración 5, recalcamos y agrupamos en un sólo gráfico la existencia de varias observaciones atípicas ya mencionadas en el análisis de las distribuciones de las variables en el subapartado 5.2. Entonces, por lo acontecido después de aplicar el DBSCAN, se examinan los *z-scores* para eliminar las observaciones que toman valores fuera del rango (-3,3). Por lo tanto, el DF al que se le aplicará el algoritmo no supervisado K-Means contiene 9602 observaciones.

6. CLUSTERZACIÓN CON K-MEANS

6.1. Selección del número de *clusters* (K)

Aplicando la regla del codo explicada en el subapartado 2.3, se procede a analizar el comportamiento del modelo a partir de las Ilustraciones 6 y 7.

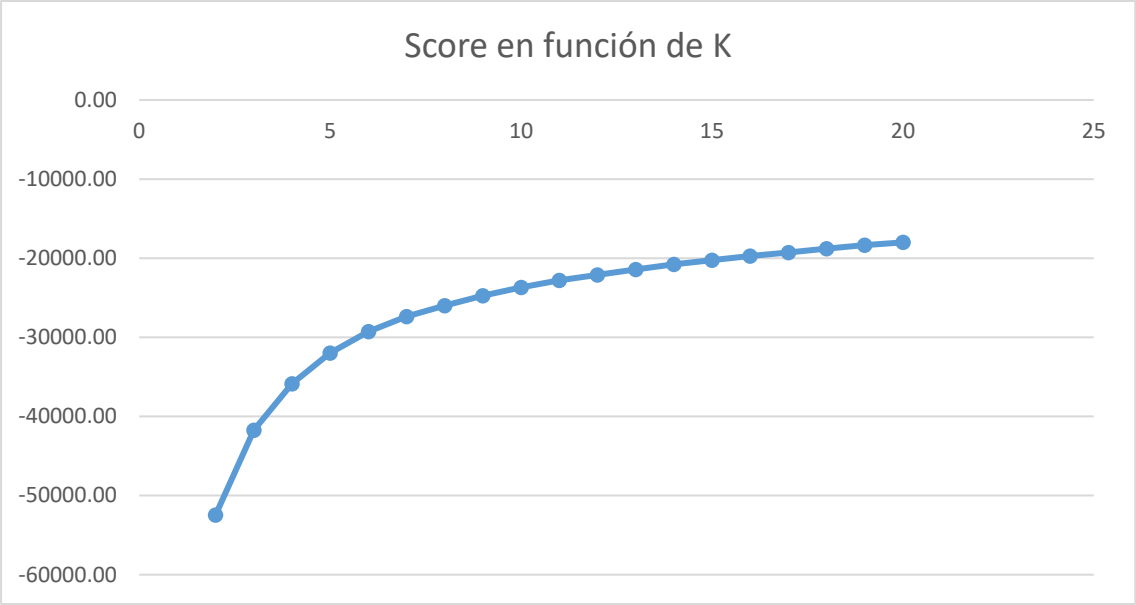


Ilustración 6: Puntuación del modelo en función de K

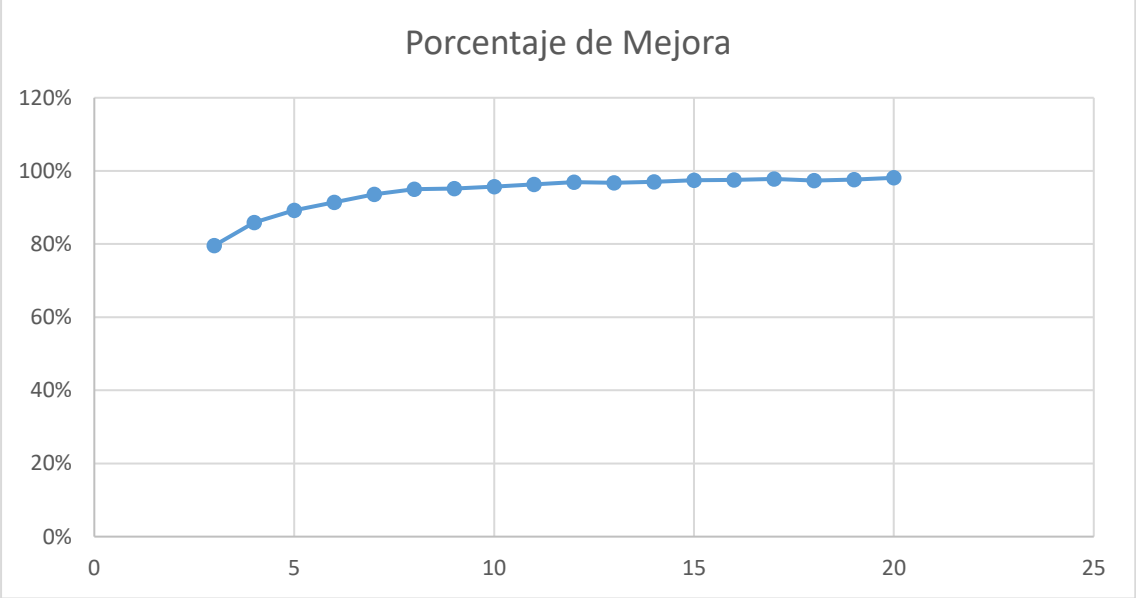


Ilustración 7: Porcentaje de mejora del modelo en función de K

En la ilustración 6, aunque el codo formado no es perfecto, sí que se aprecia una pequeña mejora a medida que aumenta el número de *clusters.* En cuanto al comportamiento del modelo con la ilustración 7, podemos apreciar el porcentaje de mejora que existe por cada aumento del número de K. Como la mejora de la puntuación es menor a medida que aumenta K, una

vez alcanzado K = 8 la mejora es pequeña. Además, cuanto más elevado sea el número de *clusters,* más complejo será interpretar cada uno de los grupos obtenidos, pudiendo crearse grupos demasiado similares o demasiado pequeños. Esto concuerda con el interés de negocio, que busca crear *clusters* suficientemente grandes para clasificar clientes (en este caso CP). Por lo tanto, seleccionamos K = 4,5, 6 y 7, y evaluamos la diferencia que hay respecto al grueso total de datos.

6.2. Análisis de *Clusters*

Este análisis se ha realizado en Excel y será incluido en Github junto con el resto de los códigos de programación del proyecto.

Para cada K se han introducido 2 tablas. La primera indica el número de observaciones que tiene cada *cluster*. La segunda compara la media global de las variables con la media de las variables en cada *cluster*, indicando la variación en porcentaje. Por ejemplo: en la Tabla 11, el *cluster* 0 tiene una Renta Mediana por Unidad de Consumo un 4% por debajo de la media global de esta variable.

Las Tablas 10 y 11 pertenecen a K = 4, las Tablas 12 y 13 a K = 5, las Tablas 14 y 15 a K = 6 y las Tablas 16 y 17 a K = 7.

- **K = 4**

Clusters	Observaciones	Frecuencia
0	1985	21%
1	2824	29%
2	2234	23%
3	2559	27%

Tabla 10: Clusters con K = 4

	R_med_unidad_consumo	pob	M_tamano_hogar	FI_pensiones	FI_desempleo	FI_otras_prestaciones	FI_otros_ingresos	PRECIO_m2_FINAL
0 VS TOT	-4%	32%	0%	-12%	13%	-2%	-6%	5%
1 VS TOT	1%	9%	1%	8%	-4%	3%	-7%	6%
2 VS TOT	-1%	-17%	-1%	-3%	-8%	-4%	3%	-11%
3 VS TOT	3%	-19%	0%	3%	2%	1%	9%	-1%

Tabla 11: Variación Respecto al Total (K = 4)

- ***K = 5***

Clusters	Observaciones	Frecuencia
0	1375	14%
1	2015	21%
2	2234	23%
3	2375	25%
4	2424	25%

Tabla 12: Clusters con K = 5

	R_med_unidad_consumo	pob	M_tamano_hogar	FI_pensiones	FI_desempleo	FI_otras_prestaciones	FI_otros_ingresos	PRECIO_m2_FINAL
0 VS TOT	-19%	-26%	-2%	-20%	57%	11%	-31%	-11%
1 VS TOT	2%	-56%	-9%	38%	-16%	26%	-11%	-23%
2 VS TOT	-1%	-17%	-1%	-3%	-8%	-4%	3%	-11%
3 VS TOT	-1%	89%	12%	-25%	2%	-5%	-22%	7%
4 VS TOT	-7%	-76%	-8%	5%	-21%	-21%	15%	-21%

Tabla 13: Variación Respecto al Total (K = 5)

- ***K = 6***

Clusters	Observaciones	Frecuencia
0	1408	15%
1	1202	13%
2	616	6%
3	1968	20%
4	2290	24%
5	2118	22%

Tabla 14: Clusters con K = 6

	R_med_unidad_consumo	pob	M_tamano_hogar	FI_pensiones	FI_desempleo	FI_otras_prestaciones	FI_otros_ingresos	PRECIO_m2_FINAL
0 VS TOT	-19%	-32%	-2%	-21%	57%	11%	-31%	-11%
1 VS TOT	29%	16%	8%	1%	-3%	-7%	66%	61%
2 VS TOT	6%	524%	11%	-20%	11%	10%	-19%	61%
3 VS TOT	2%	-58%	-9%	38%	-16%	26%	-11%	-23%
4 VS TOT	0%	-7%	11%	-21%	-3%	-9%	-17%	-1%
5 VS TOT	-8%	-79%	-10%	7%	-21%	-21%	17%	-22%

Tabla 15: Variación Respecto al Total (K = 6)

- **K = 7**

Clusters	Observaciones	Frecuencia
0	1720	18%
1	1816	19%
2	587	6%
3	1319	14%
4	1066	11%
5	2214	23%
6	880	9%

Tabla 16: Clusters con K = 7

	R_med_unidad_consumo	pob	M_tamano_hogar	FI_pensiones	FI_desempleo	FI_otras_prestaciones	FI_otros_ingresos	PRECIO_m2_FINAL
0 VS TOT	-12%	-77%	-11%	3%	-18%	-21%	2%	-22%
1 VS TOT	1%	-58%	-9%	38%	-15%	27%	-18%	-25%
2 VS TOT	5%	537%	11%	-19%	11%	11%	-20%	57%
3 VS TOT	-19%	-31%	-2%	-21%	59%	11%	-31%	-12%
4 VS TOT	16%	-52%	-3%	21%	-21%	-6%	70%	-4%
5 VS TOT	0%	-7%	11%	-21%	-3%	-8%	-18%	-3%
6 VS TOT	30%	41%	12%	-13%	5%	-12%	53%	86%

Tabla 17: Variación respecto al Total (K = 7)

También se han estudiado las diferencias entre *clusters* por cada K, pero no se han añadido para evitar la saturación de resultados.

Para simplificar la notación, a partir de aquí nos referimos a la Renta Mediana por Unidad de Consumo como Renta, ya que es la única variable de renta que hemos mantenido. Lo mismo para el Precio por Metro Cuadrado, que pasará a notarse como Precio. A las variables de Fuentes de Ingresos se las notará por el tipo de ingreso, es decir, Pensiones, Desempleo, Otras Prestaciones y Otros Ingresos.

Analizando todas las tablas calculadas en Excel se concluye qué número de *clusters* nos conviene más. Con K = 4 *clusters* el modelo no es capaz de diferenciar los grupos a partir de la Renta de cada CP, siendo esta una de las variables más importantes. Con K = 5 clusters, aunque ha mejorado la diferenciación grupal, comparando con la opción de K = 6 *clusters*, es sustancialmente peor. Con K = 5, aunque un *cluster* destaca con Rentas muy bajas, buscamos también *clusters* con Rentas elevadas. Con K = 6 *clusters*, sí que aparecen 2 con una Renta elevada (concretamente uno de ellos con un 29%). Además, con K = 6, se cumple una tendencia común: variables como el Precio y la Población elevadas implican rentas más altas. Estas zonas probablemente sean zonas urbanas de ciudades con mayor poder adquisitivo. A partir de K = 7 *clusters,* se puede apreciar en la Tabla 16 que ya encontramos 2 *clusters* por debajo del 10% de observaciones y los grupos creados no mejoran excesivamente a los del caso anterior. Definitivamente la K escogida es 6.

6.3. Perfil de los *Clusters*

A partir de la Tabla 15, se pueden intuir las diferencias entre los CP y crear así perfiles diferenciados.

- ***Cluster* 0:**

  Se caracteriza por tener una Renta bastante baja (-19% en media) además de una Población y Precio bajos. Además, el Desempleo es muy elevado. Esas características delatan CP rurales con poder adquisitivo muy bajo y probablemente poco desarrollo laboral fuera del sector primario. De antemano, sabemos que CA como Extremadura y Andalucía destacan por estas características. Si analizamos el número de CP pertenecientes a cada provincia en la Tabla 18 (se extraen 14 observaciones para su visualización), se aprecia que las provincias con código 6 y 11 (Badajoz y Cáceres) representan un 20% del total del *cluster*. La gran mayoría de provincias

andaluzas (18, 23, 14, 29, 41, 4, 21 y 11) también tienen un peso importante en este *cluster*, otorgando un 47% de observaciones a este grupo.

Este *cluster* queda clasificado como: **Agrupación de Códigos Postales Rurales más Desfavorecidos.**

Hay que recalcar que la información que recogemos de esta clusterización no es comprobar que una gran parte de esas provincias se comporta de esa manera, esto ya se sabía. Lo que demuestra la Tabla 18 es que hemos encontrado CP en otras zonas de España que se comportan como muchas zonas de Andalucía y Extremadura. A partir de un análisis por provincias, no hubieran sido detectadas.

PROV	count	frecuencia
6	176	11%
10	167	11%
18	143	9%
23	136	9%
38	95	6%
14	94	6%
29	87	6%
41	83	5%
35	66	4%
4	63	4%
21	61	4%
11	49	3%
2	44	3%

Tabla 18: Frecuencia de Provincias en el Cluster 0 (extracción para 14 observaciones)

- ***Cluster* 1:**

Es el *cluster* con Renta más elevada con gran diferencia. Por ejemplo, se encuentra un 29% por encima de la media de la Renta para la BD en su conjunto y, en comparación con el *cluster* con menor Renta en media (*cluster 0)*, existe una diferencia del 159%. Este *cluster* también alberga el Precio más elevados del

conjunto de datos junto con el *cluster 2.* Con una Población por encima de la media, está claro que estos CP son de zonas urbanas con un poder adquisitivo muy elevado. Por ejemplo, clasificando los *outliers* a partir de este modelo, los CP con las rentas más altas del país pasan a formar parte de este *cluster* (se verá en el siguiente subapartado en la Ilustración 8).

En la Tabla 19 se puede apreciar como Barcelona (provincia 8), encabeza el *cluster* 1 con un 13% de los códigos postales y está precedida por Navarra (provincia 31), con un 12%. Curiosamente Madrid (provincia 28), se encuentra en octavo puesto.

A este *cluster* lo bautizamos como: **Agrupación de Códigos Postales de la Élite Socioeconómica.**

PROV	count	frecuencia
8	192	13%
31	180	12%
17	174	12%
22	130	9%
7	94	6%
20	91	6%
25	86	6%
28	82	5%
39	36	2%
15	35	2%
1	33	2%
9	31	2%
33	31	2%

Tabla 19: Frecuencia de Provincias en el Cluster 1 (extracción para 14 observaciones)

- ***Cluster* 2:**

Lo que diferencia a este grupo del *cluster* 1 es, sobre todo, la cantidad de Población y la Renta. La Población es mucho más elevada (un 537% respecto a la media total) y la Renta más baja, aunque es un 6% más alta que la media total. Además, el *cluster* 2 tiene más ingresos por Desempleo y menos por Otros

ingresos. Estas características cuadran con la información de la Tabla 20, ya que Barcelona (provincia 8) y Madrid (provincia 28) están a la cabeza y son las provincias con las dos ciudades más pobladas en España. Además, estas ciudades también son conocidas por su alto poder adquisitivo. Esto quiere decir que los CP de estas dos ciudades que no alcanzan el nivel económico del *cluster* 1 se están introduciendo en este *cluster*.

Entonces, este *cluster* recibe el siguiente título: **Agrupación de Códigos Postales Urbanos de nivel Socioeconómico Alto.**

PROV	count	frecuencia
8	127	14%
28	111	12%
46	56	6%
29	50	6%
3	47	5%
41	42	5%
11	37	4%
30	34	4%
7	32	4%
35	29	3%
43	25	3%
18	20	2%
36	18	2%

Tabla 20: Frecuencia de Provincias en el Cluster 2 (extracción para 14 observaciones)

- **Cluster 3:**

Este grupo, al tener una Población y Precio bajos, denotan al igual que el *cluster* 0, zonas rurales (la Población media es más pequeña que la del *cluster* 0). La diferencia es que estas zonas tienen un poder adquisitivo por encima de la media. Además, destacan unos elevados ingresos por Pensiones y un Tamaño del Hogar Medio más reducido de lo habitual (de 1 a 2 personas en media), así que es probable que una gran parte de la población sea gente de edad avanzada).

En la Tabla 21 se puede apreciar como León (provincia 24) y Asturias (provincia 33) lideran este *cluster*. Esto cuadra con la descripción deducida a partir del estudio descriptivo.

El *cluster* queda bautizado como: **Agrupación de Códigos Postales Rurales con Residentes de Edad Avanzada.**

PROV	count	frecuencia
24	357	17%
33	306	14%
27	279	13%
32	218	10%
49	150	7%
15	94	4%
37	64	3%
25	62	3%
36	46	2%
39	45	2%
44	35	2%
34	34	2%
46	33	2%

Tabla 21: Frecuencia de Provincias en el Cluster 3 (extracción para 14 observaciones)

Reafirmando lo dicho para la Tabla 18 en el *cluster* 0, hay que comprender que las Tablas de la 18 a la 21 demuestran que el algoritmo ha detectado algunos CP que se comportan como zonas de las que sí conocemos su perfil.

- ***Cluster* 4:**

El *cluster* 4 podría considerarse el grupo estándar ya que no destaca especialmente. Todas las Fuentes de Ingresos son bajas, pero tanto la Renta como el Precio están muy cerca de la media y la Población está un poco por debajo pero no es nada exagerado.

Por lo tanto, este *cluster* se llamará: **Agrupación de Códigos Postales de nivel Socioeconómico Medio.**

- ***Cluster* 5:**

  El último grupo es el que contiene los CP con menor Población. Junto con un Precio bajo, también nos encontramos frente a un *cluster* rural. Se diferencia del *cluster* 3 gracias a la Renta, que es más baja, y a las Pensiones, que también son más bajas. Además, la gran diferencia que tiene con el *cluster* 0 es el Desempleo, que es un 21% menor que la media. Este dato sumado a un Tamaño del Hogar Medio bajo y unos ingresos por Pensiones por encima de la media remarcan valores de población de edad avanzada.

  Agrupación de Códigos Postales Rurales de Baja Densidad y Rentas Modestas será el nombre del último *cluster*.

6.4. Visualización de *Clusters*

Repitiendo el proceso realizado para visualizar la aplicación del DBSCAN, a partir de un PCA graficamos los diferentes *clusters* con *outliers* (Ilustración 8) y sin *outliers* (Ilustración 9).

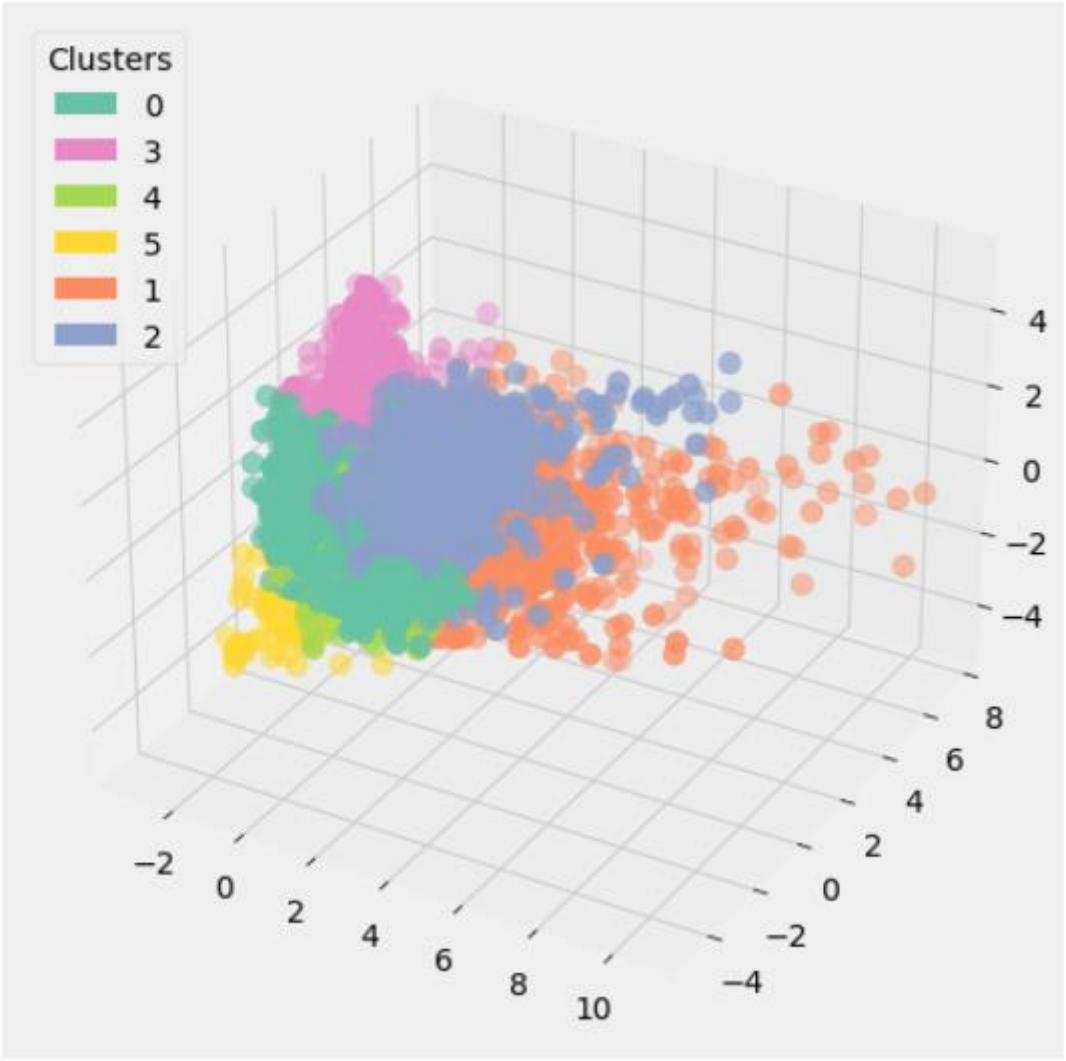


Ilustración 8: Visualización de clusters con outliers

En la Ilustración 8 se aprecia como los *outliers* más destacados (los más alejados de la masa de puntos en la derecha de la Ilustración) la mayoría se han incluido en el *cluster* 1. Los restantes se han añadido al *cluster* 2.

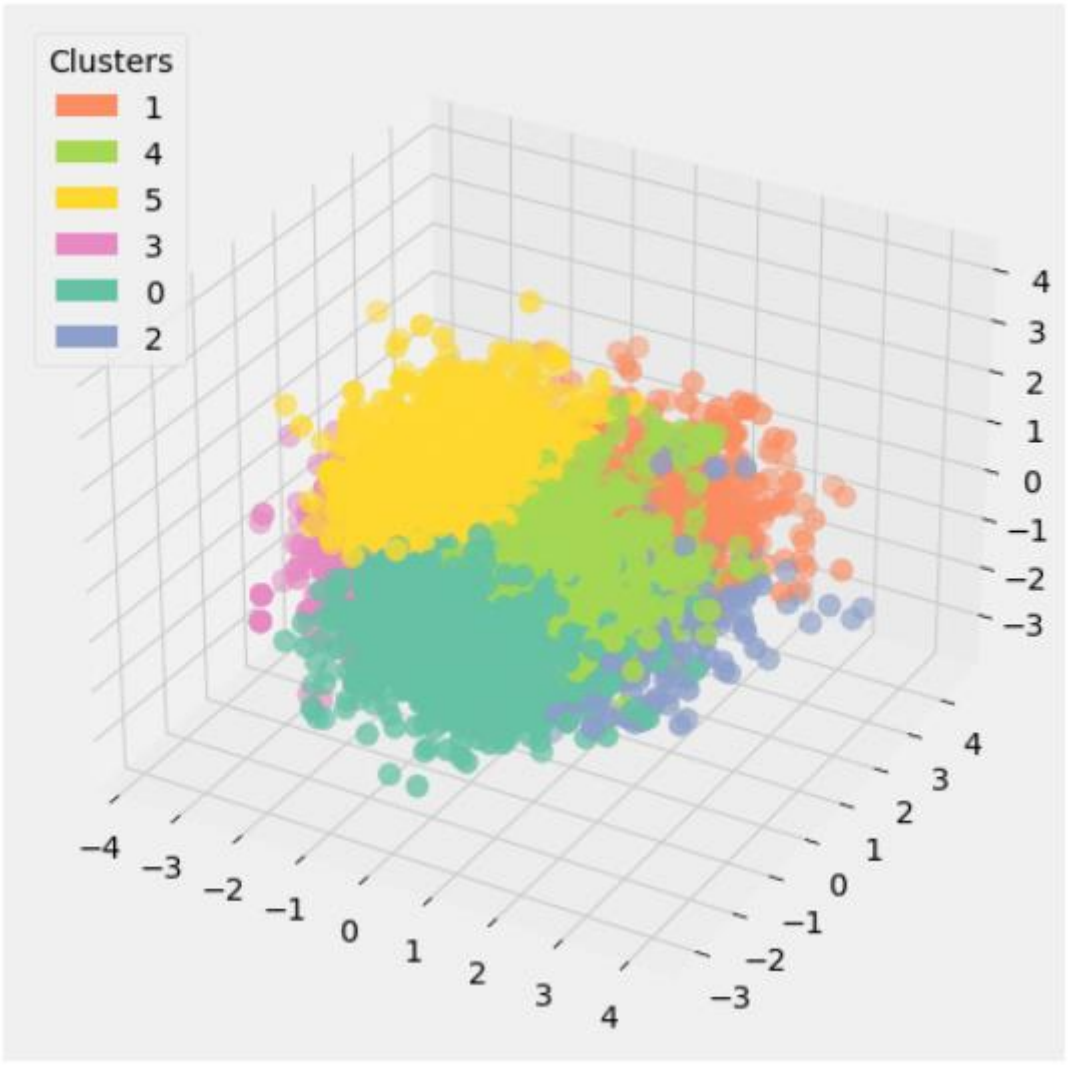


Ilustración 9: Visualzación de clusters sin outliers

En la Ilustración 9 se aprecia como el K-Means ha trabajado el conjunto de datos. Ha tomado centroides lo más alejados posibles y ha clasificado a cada observación en el *cluster* más cercano. Esta es una de las limitaciones del algoritmo, ya que los puntos con distancias parecidas a dos centroides diferentes pueden estar peor clasificados.

7. CONCLUSIONES

El proyecto realizado tenía como objetivo principal comprobar si se pueden obtener perfiles diferenciados de los diferentes CP en los que está

fraccionada España para un conjunto de variables socio-demográficas, confirmando nuestra hipótesis inicial. A través de análisis estadísticos, se ha conseguido categorizar en 6 grupos diferentes lo suficientemente significativos como para ser de utilidad en proyectos de negocio posteriores.

En relación al objetivo secundario, una manera de aplicar esta información es a partir de la venta cruzada de productos. Por ejemplo, cuando se conoce la zona residencial de un cliente, sabiendo su nivel socioeconómico, interesará invertir tiempo en intentar venderle un nuevo producto de inversión o no. Desde otra perspectiva, sería la no inversión de recursos en localidades sin interés. Por ejemplo, no se invertirá tiempo contactando con clientes de localidades de edades avanzadas para vender seguros de salud o de vida. También se pueden potenciar campañas de *marketing* local a través de carteles publicitarios, consiguiendo un impacto más directo.

En resumen, este TFG aporta evidencia relevante sobre la posibilidad de clasificar por CP todo el país y abre la puerta a nuevos proyectos que puedan retribuirse a partir de esta clasificación.

REFERENCIAS BIBLIOGRÁFICAS

El Callejero. (s. f.). https://elcallejero.es/

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 226-231. Portland, Estados Unidos. https://cdn.aaai.org/KDD/1996/KDD96-037.pdf

Geeks for Geeks. (2023, noviembre 23). Índice de Davies-Bouldin. https://www.geeksforgeeks.org/davies-bouldin-index/

Gómez-Barroso, D., Prieto-Flores, M.-E., Mellado San Gabino, A., & Moreno Jiménez, A. (2015). ANÁLISIS ESPACIAL DE LA MORTALIDAD POR ENFERMEDADES CARDIOVASCULARES EN LA CIUDAD DE MADRID, ESPAÑA. Revista Española de Salud Pública, 89(1), 27-37. https://www.scielosp.org/pdf/resp/2015.v89n1/27-37/es

Alteryx (2019). *Herramienta de diagnóstico de K-centroides*. https://help.alteryx.com/2020.1/es/K-Centroids_Diagnostics.htm#:~:text=El%20%C3%ADndice%20Calinski%2DHarabasz%20se,puntos%20dentro%20de%20un%20cl%C3%BAster

INE. (2024). https://www.ine.es/

INE. (2021). Atlas de Distribución de Renta de los Hogares. https://www.ine.es/dynt3/inebase/index.htm?padre=7132

INE. (s.f.). Escalas de equivalencia https://www.ine.es/DEFIne/es/concepto.htm?c=5228&op=30458#:~:text=Escala%20de%20equivalencia%20de%20la,(13%20a%C3%B1os%20y%20menos)

Jolliffe, I. T. (2002). *Principal Component Analysis* (2nd ed.). Springer. http://cda.psych.uiuc.edu/statistical_learning_course/Jolliffe%20I.%20Principal%20Component%20Analysis%20(2ed.,%20Springer,%202002)(518s)_MVsa_.pdf

Kodinariya, T., & Makwana, Dr. P. (2013). Review on determining number of Cluster in K-Means Clustering. IJARCSMS, 1(6), 90-95. https://www.researchgate.net/profile/Trupti-Kodinariya/publication/313554124_Review_on_Determining_of_Cluster_in_K-means_Clustering/links/5789fda408ae59aa667931d2/Review-on-Determining-of-Cluster-in-K-means-Clustering.pdf

Martori, J. C., & Hoberg, K. (2008). NUEVAS TÉCNICAS DE ESTADÍSTICA ESPACIAL PARA LA DETECCIÓN DE CLUSTERS RESIDENCIALES DE POBLACIÓN INMIGRANTE. REVISTA ELECTRÓNICA DE GEOGRAFÍA Y CIENCIAS SOCIALES, 12(263), 1-13. http://dspace.uvic.cat/xmlui/bitstream/handle/10854/2396/artconlli_a2008_martori_joan_carles_nuevas_tecnicas_estadistica.pdf?sequence=1&isAllowed=y

McKinney, W. (2010). Data structures for statistical computing in Python. En Proceedings of the 9th Python in Science Conference, 51-56. Austin, Texas. https://pdfs.semanticscholar.org/ef4e/f7f38bb907e5d7b4df3e6ff1db269d4970f5.pdf?_gl=1*1t91t2v*_ga*OTE5NDAxODkxLjE3MTcyMDIyOTM.*_ga_H7P4ZT52H5*MTcxNzIwMjI5My4xLjAuMTcxNzIwMjMwMy41MC4wLjA.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J.,

Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830. https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf

PySpark Overview. (2024). https://spark.apache.org/docs/latest/api/python/index.html

RealAdvisor. (2024, marzo 15). https://realadvisor.es/es/precios-viviendas

Richardson, L. (2004). https://www.crummy.com/software/BeautifulSoup/

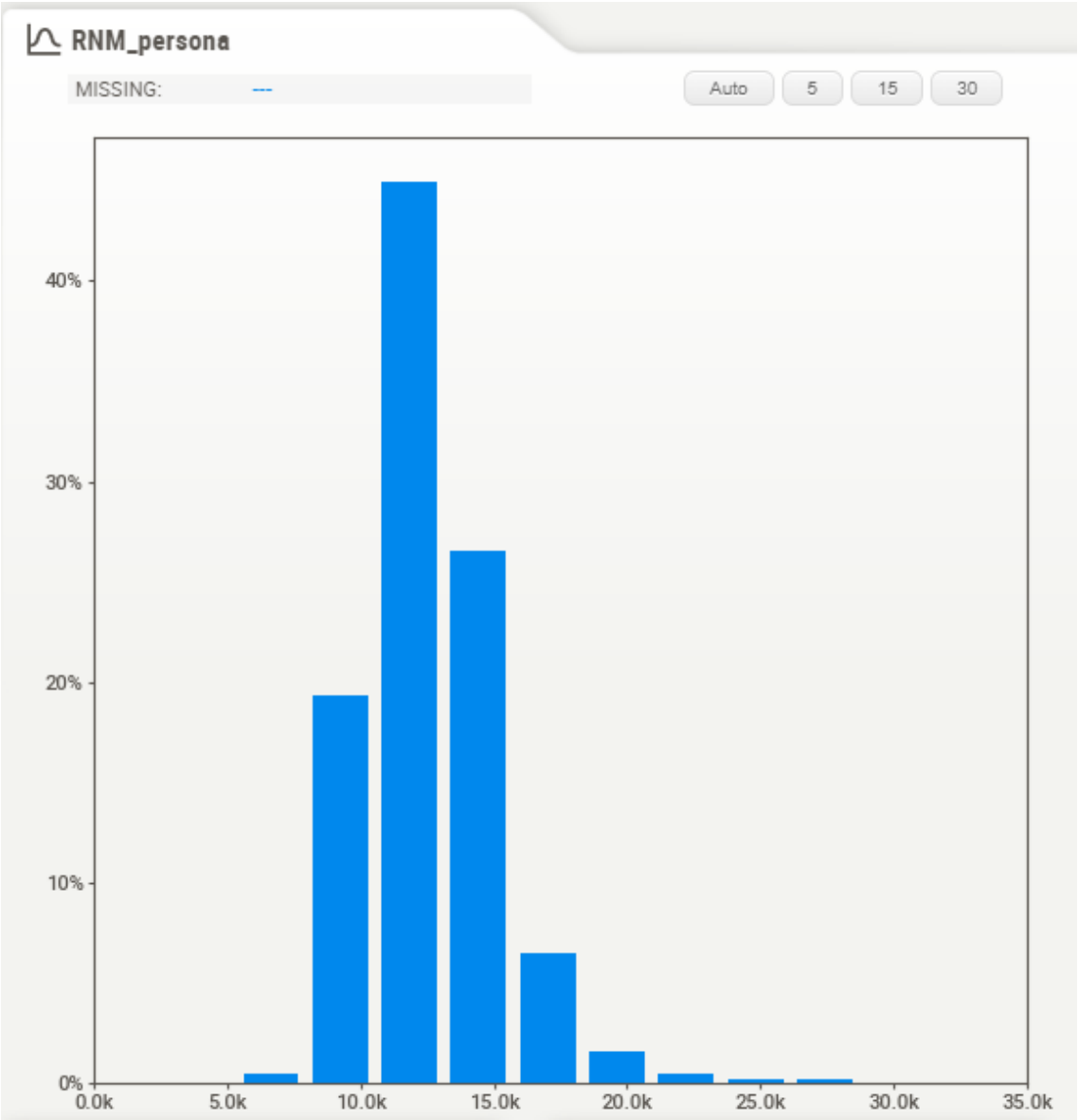
Rodríguez, D. (2023, junio 23). Número óptimo de clústeres con Silhouette e implementación en Python. https://www.analyticslane.com/2023/06/23/numero-optimo-de-clusteres-con-silhouette-e-implementacion-en-python/#google_vignette

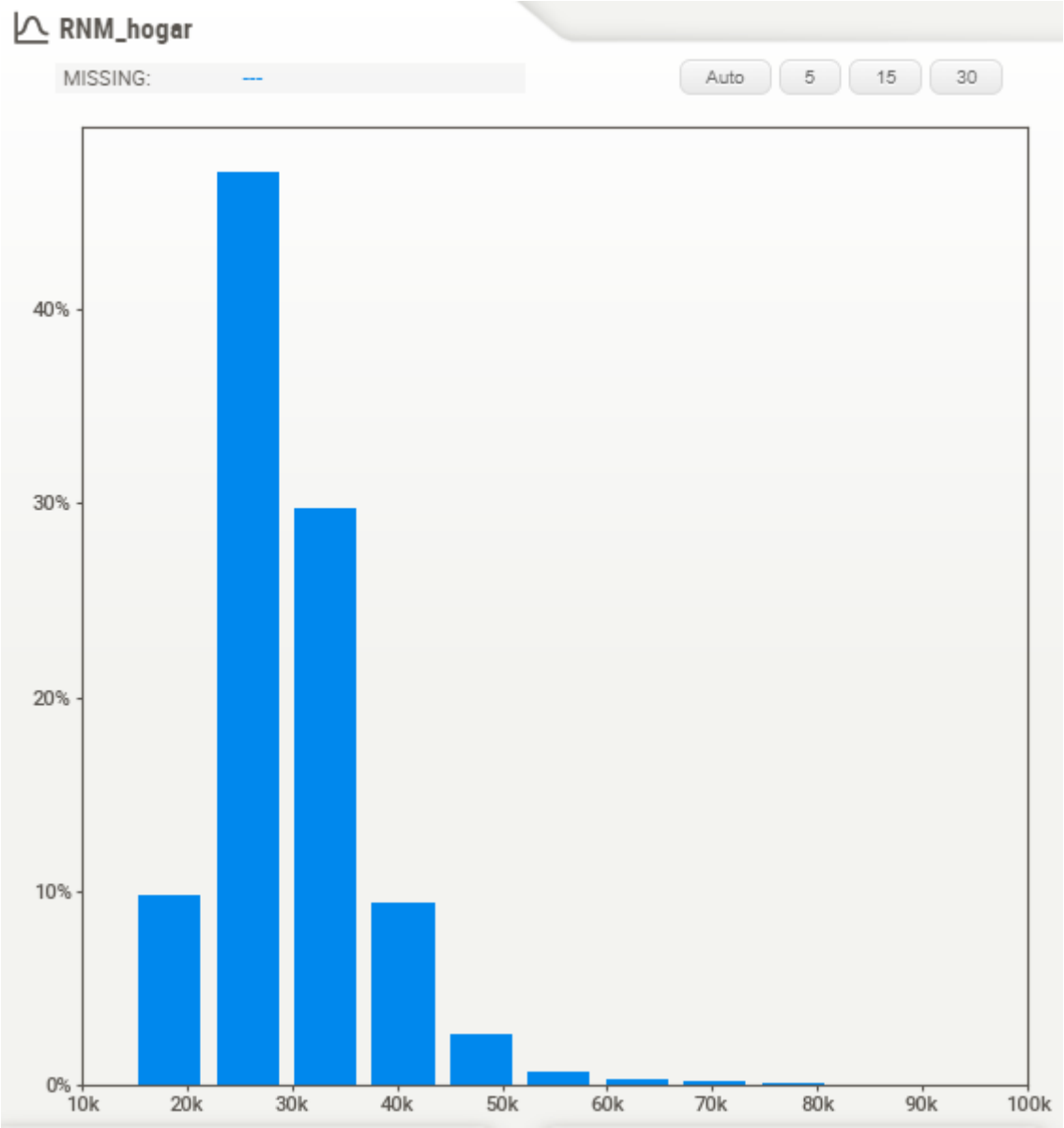
ANEXO

Enlace y código QR de Github: https://github.com/Fopoga/TFG-FILES

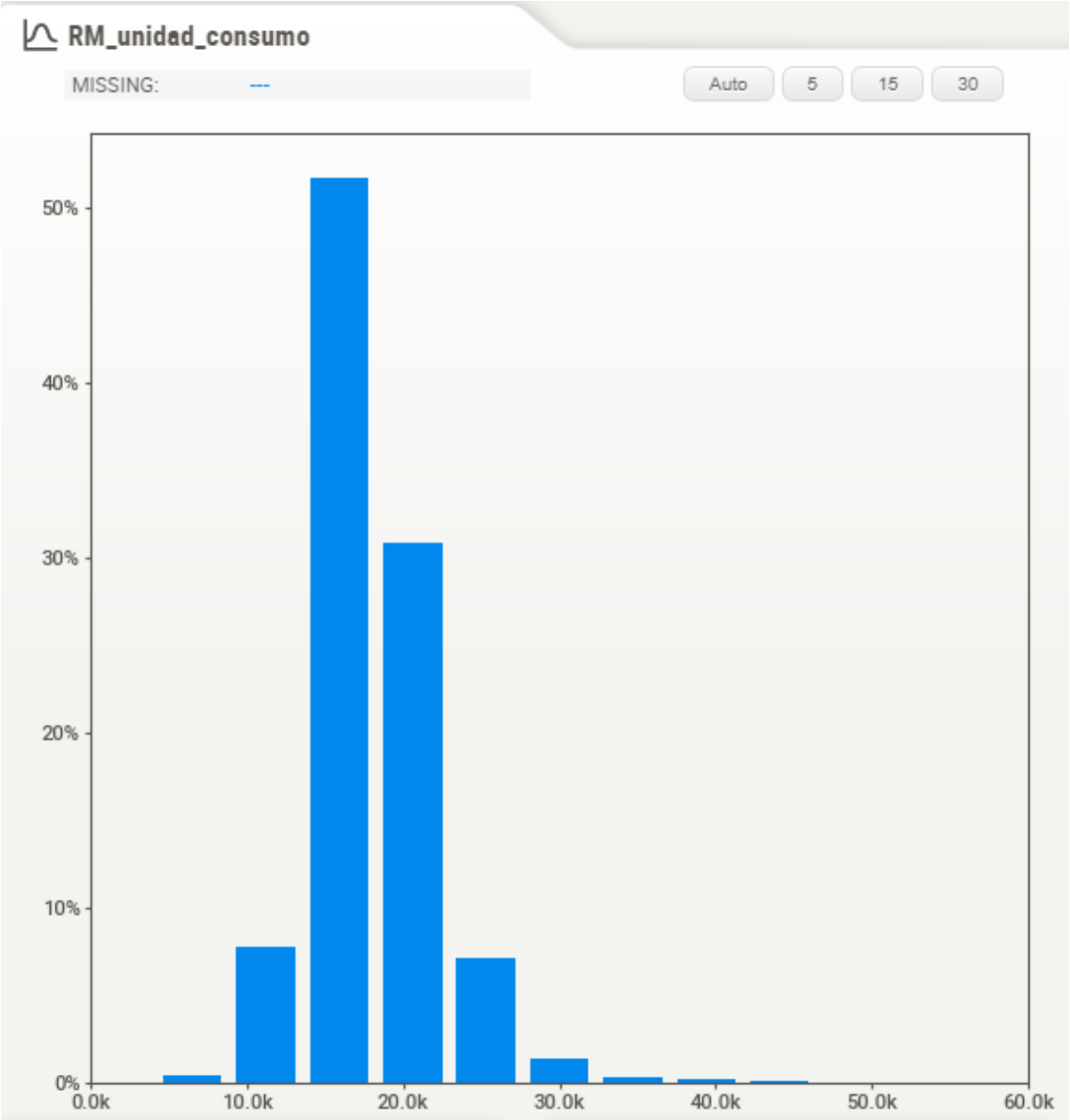
En este Anexo, se han incluido los histogramas y diagramas de caja mencionados en el subapartado 5.2

Histogramas

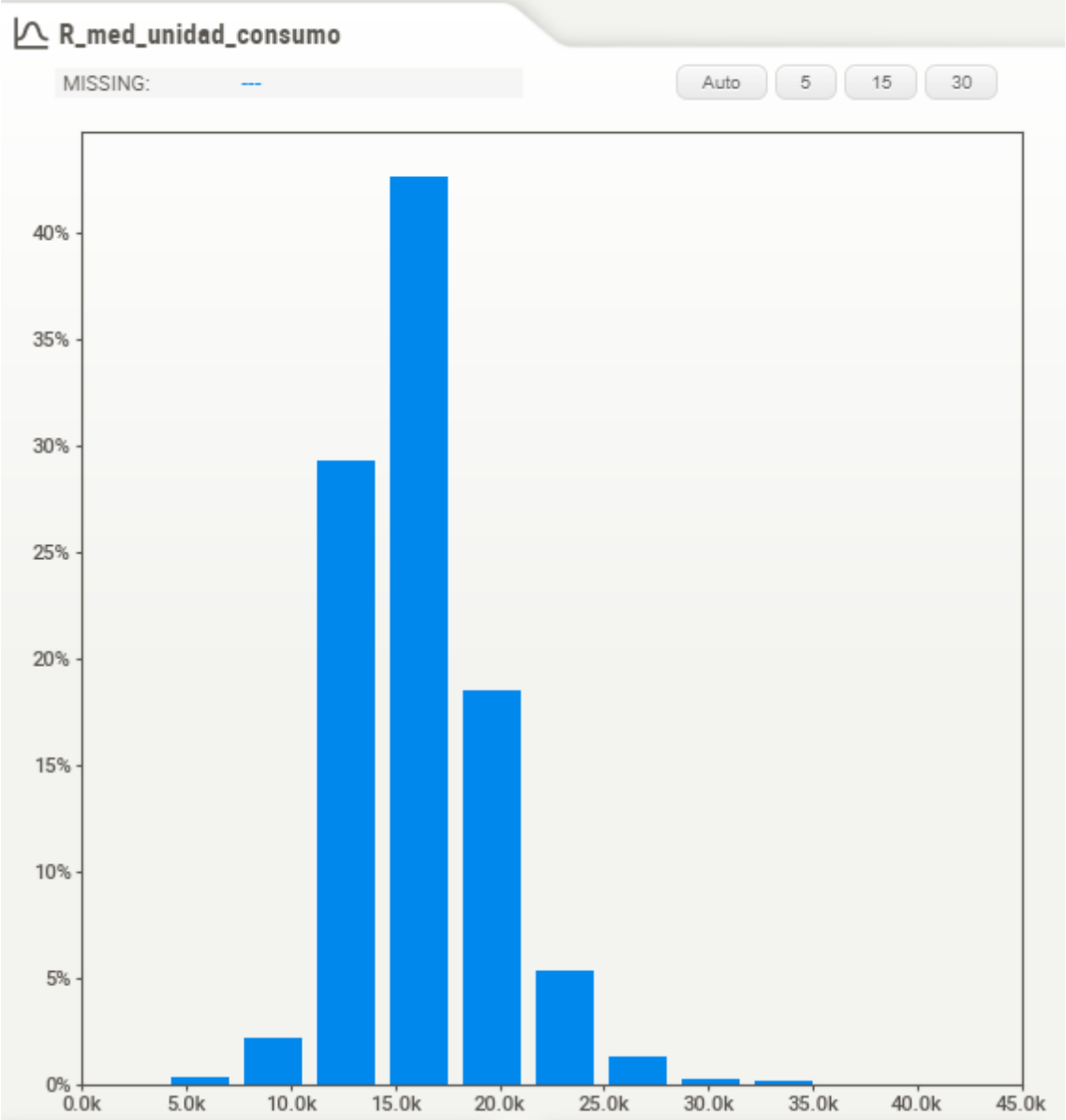



RNM_hogar
MISSING:

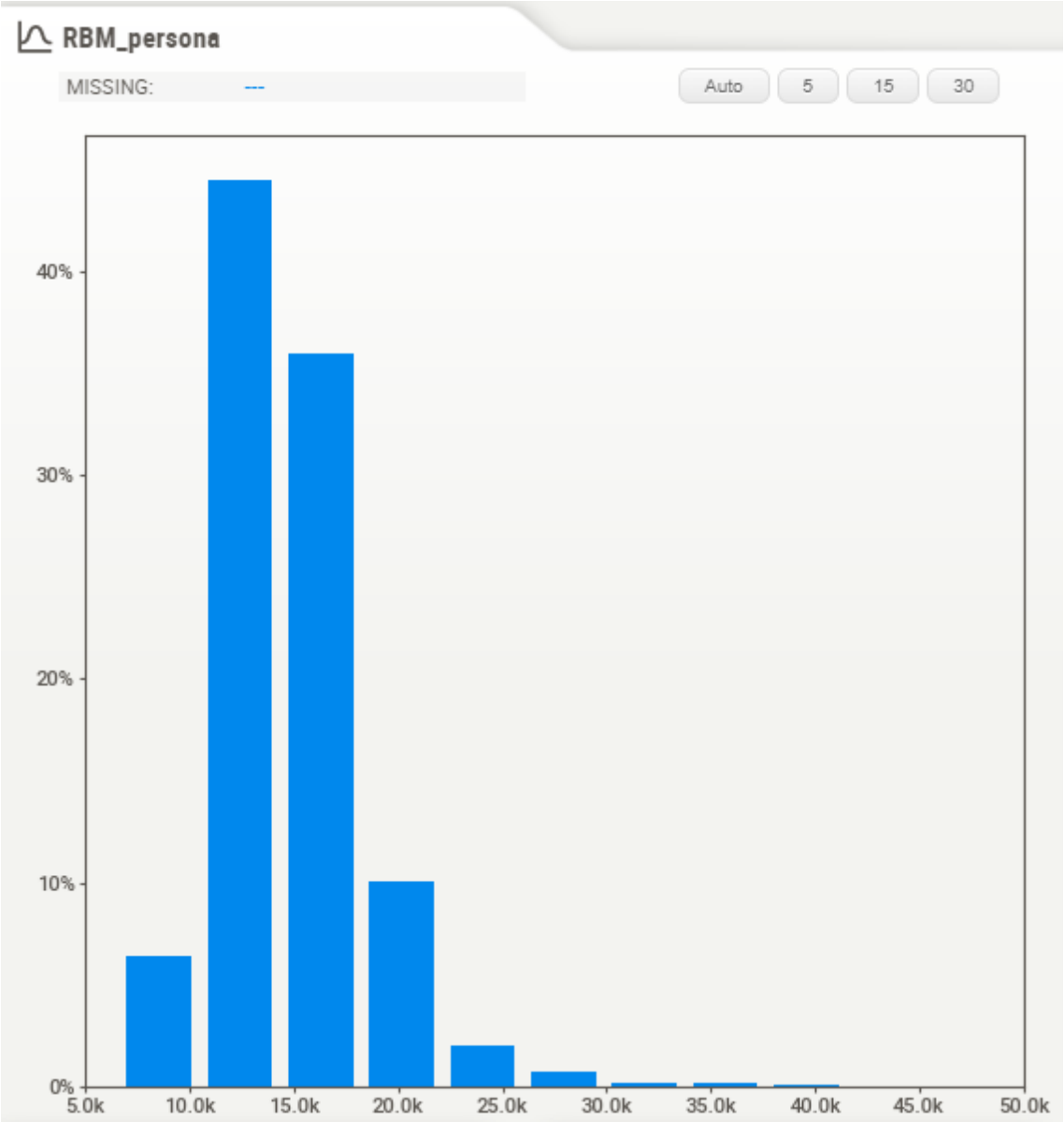
Auto
5
15
30
40%
30%
20%
10%
0%
10k
20k
30k
40k
50k
60k
70k
80k
90k
100k


RM_unidad_consumo
MISSING:

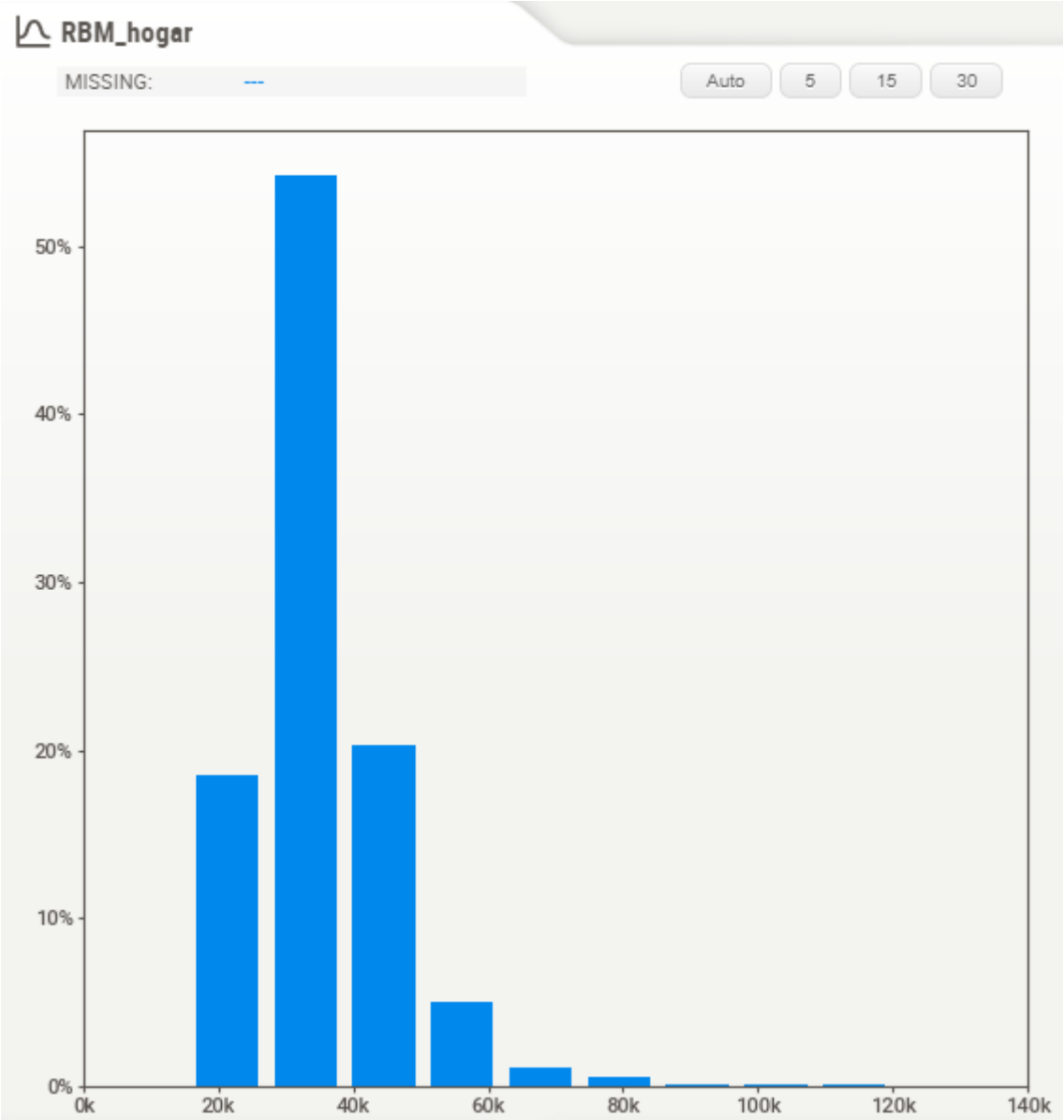
Auto
5
15
30
50%
40%
30%
20%
10%
0%
0.0k
10.0k
20.0k
30.0k
40.0k
50.0k
60.0k


R_med_unidad_consumo
MISSING:

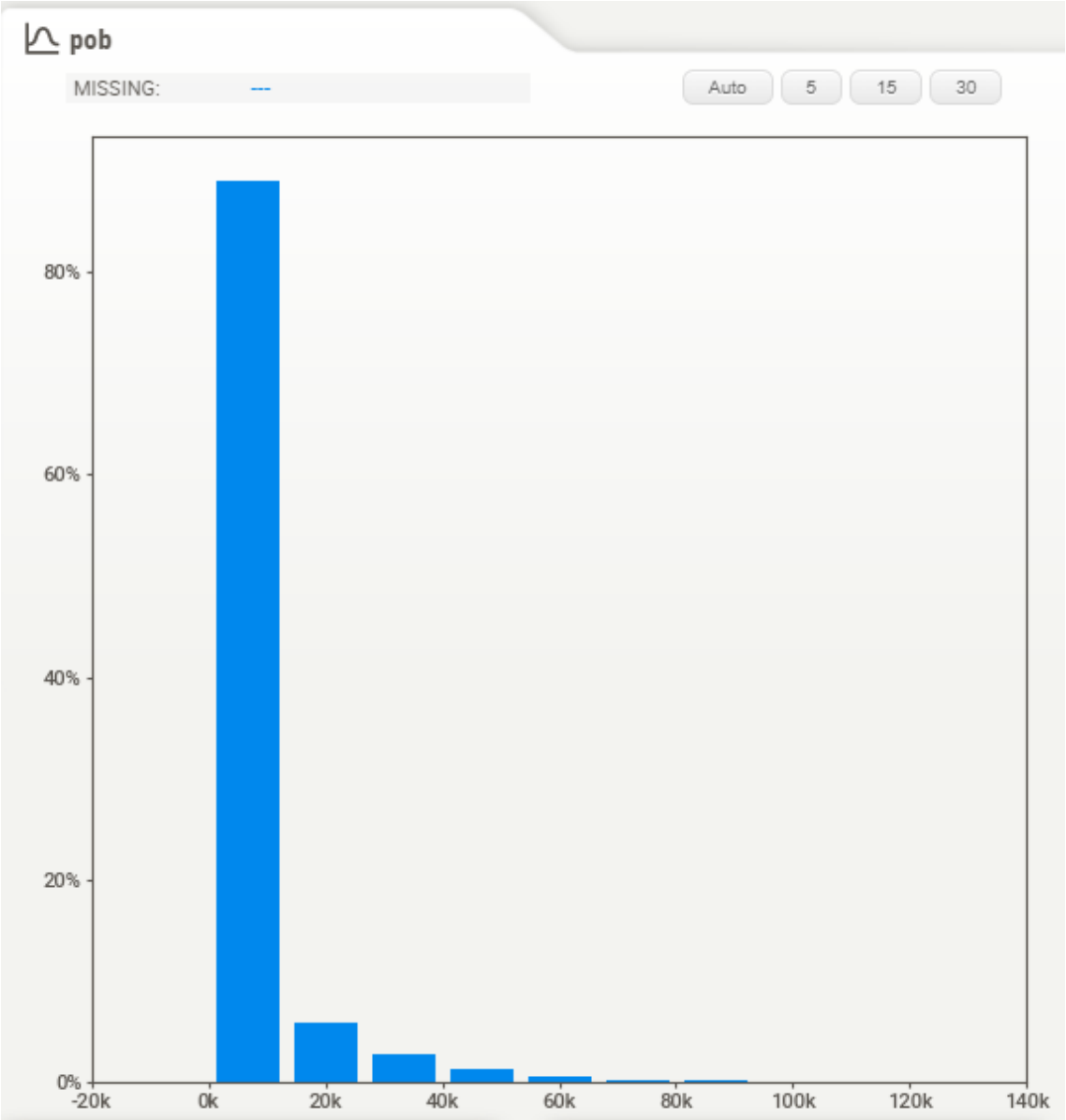
Auto
5
15
30
40%
35%
30%
25%
20%
15%
10%
5%
0%
0.0k
5.0k
10.0k
15.0k
20.0k
25.0k
30.0k
35.0k
40.0k
45.0k


RBM_persona
MISSING:

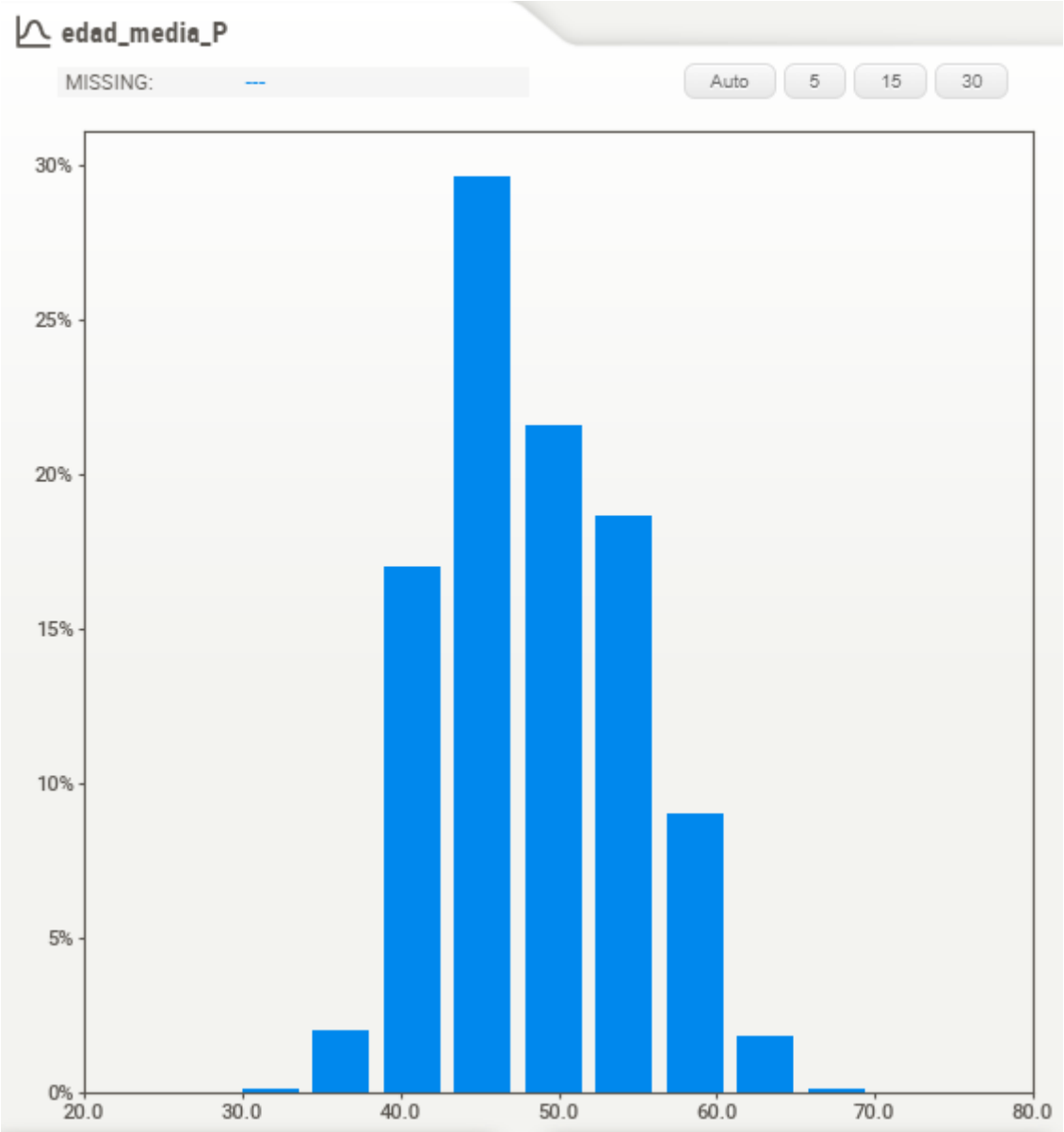
Auto
5
15
30
40%
30%
20%
10%
0%
5.0k
10.0k
15.0k
20.0k
25.0k
30.0k
35.0k
40.0k
45.0k
50.0k


RBM_hogar
MISSING:

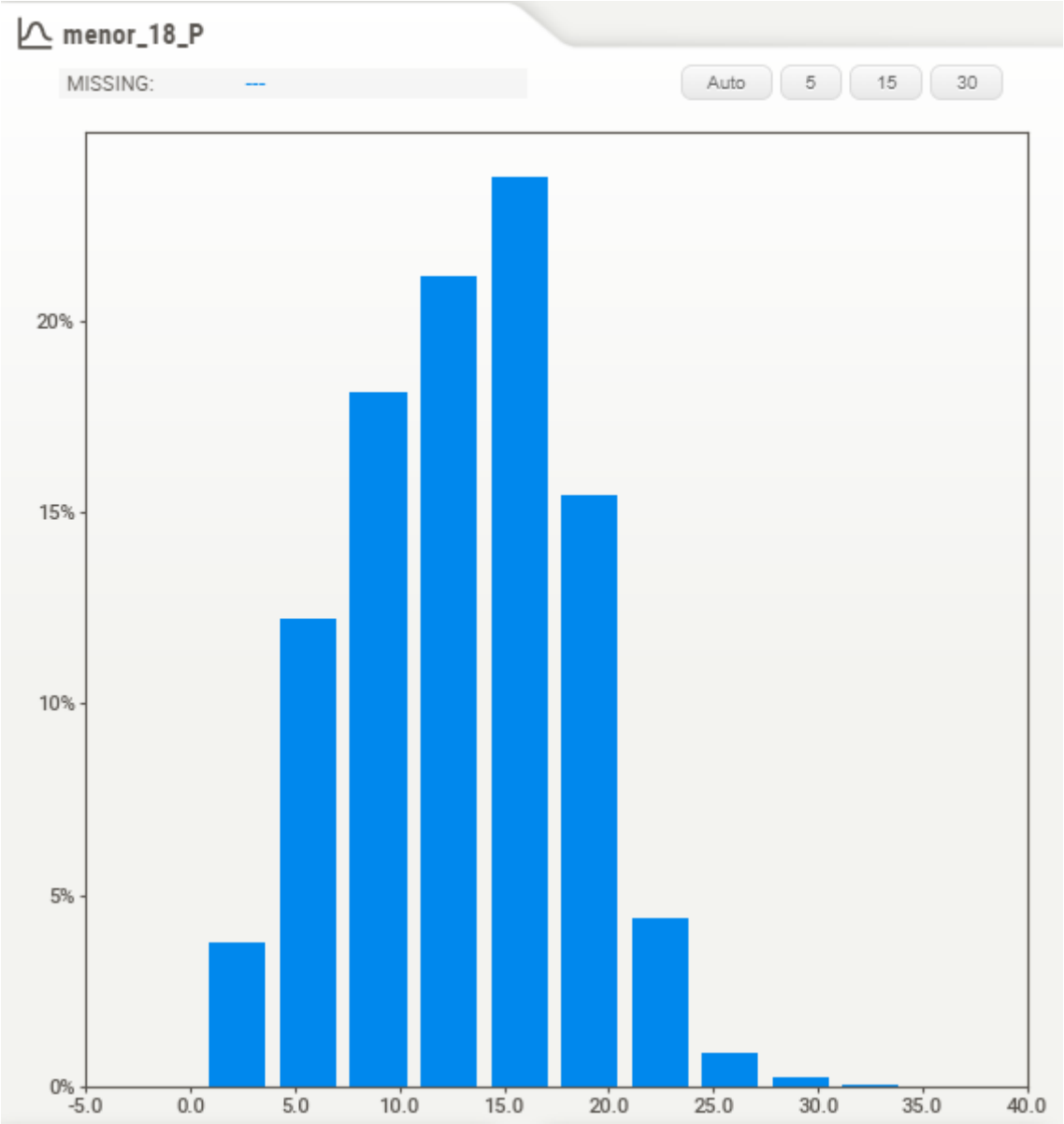
Auto
5
15
30
50%
40%
30%
20%
10%
0%
0k
20k
40k
60k
80k
100k
120k
140k


pob
MISSING:

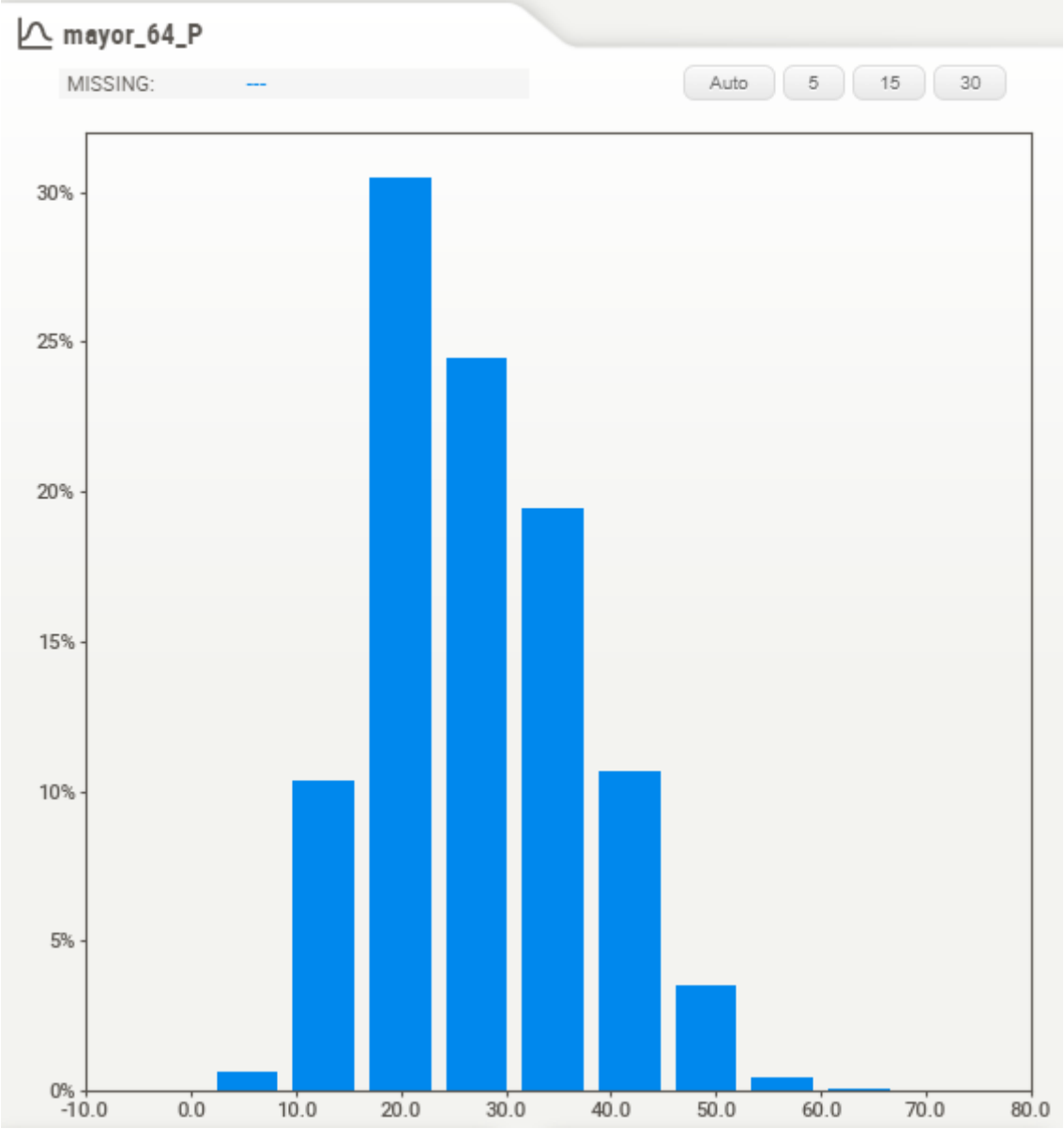
Auto
5
15
30
80%
60%
40%
20%
0%
-20k
0k
20k
40k
60k
80k
100k
120k
140k


edad_media_P
MISSING:

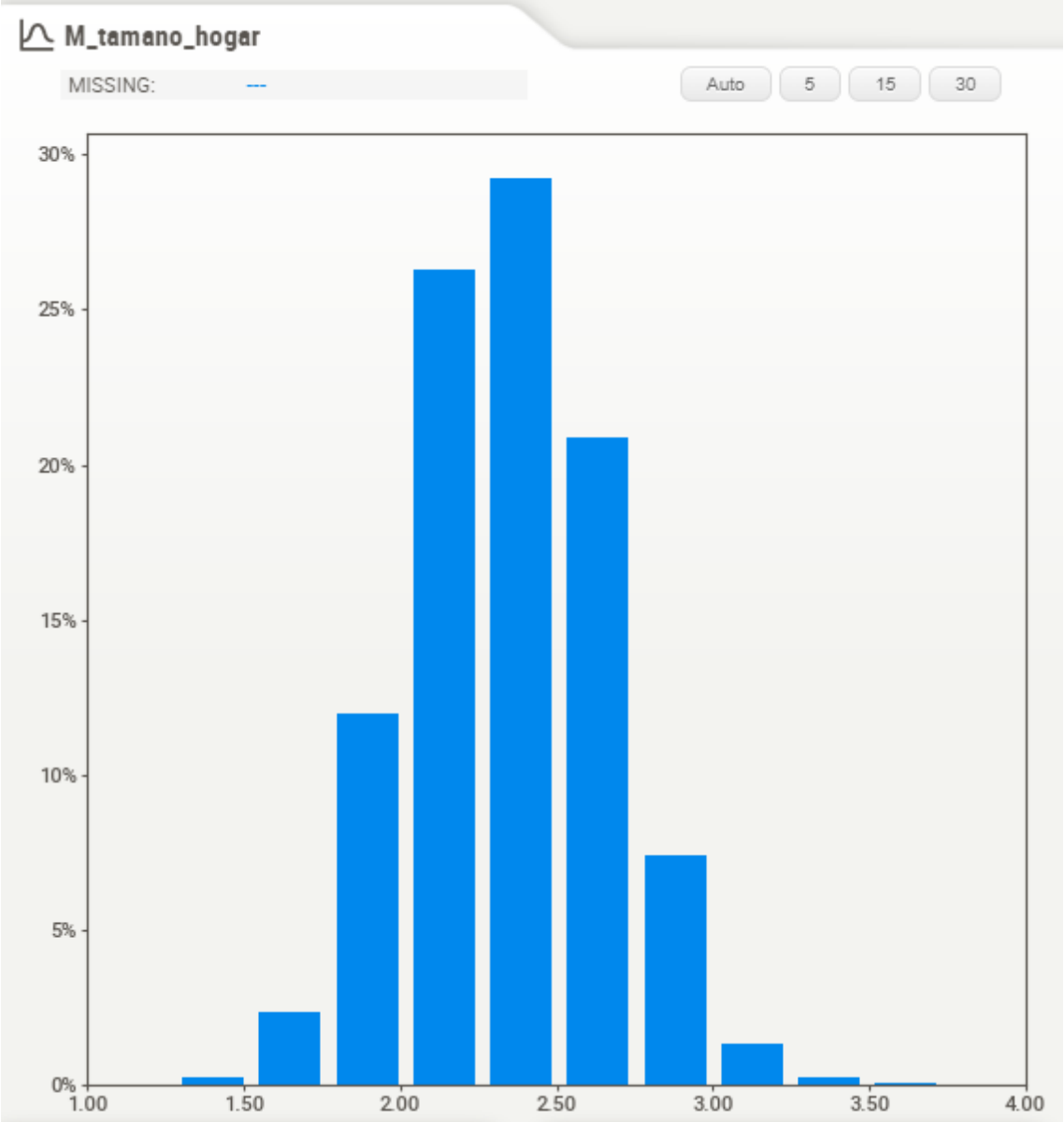
Auto
5
15
30
30%
25%
20%
15%
10%
5%
0%
20.0
30.0
40.0
50.0
60.0
70.0
80.0


menor_18_P
MISSING:

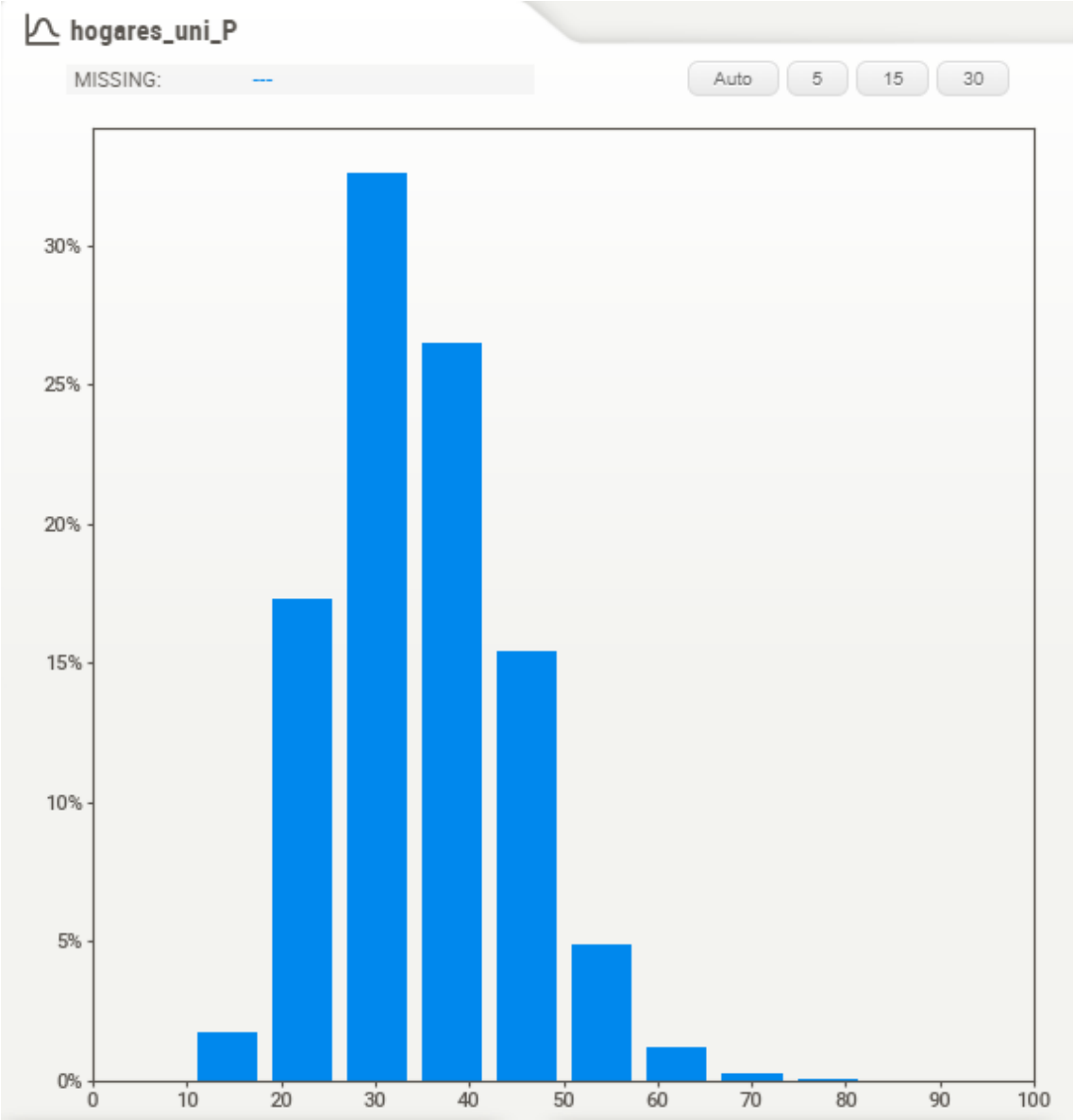
Auto
5
15
30
20%
15%
10%
5%
0%
-5.0
0.0
5.0
10.0
15.0
20.0
25.0
30.0
35.0
40.0


mayor_64_P
MISSING:

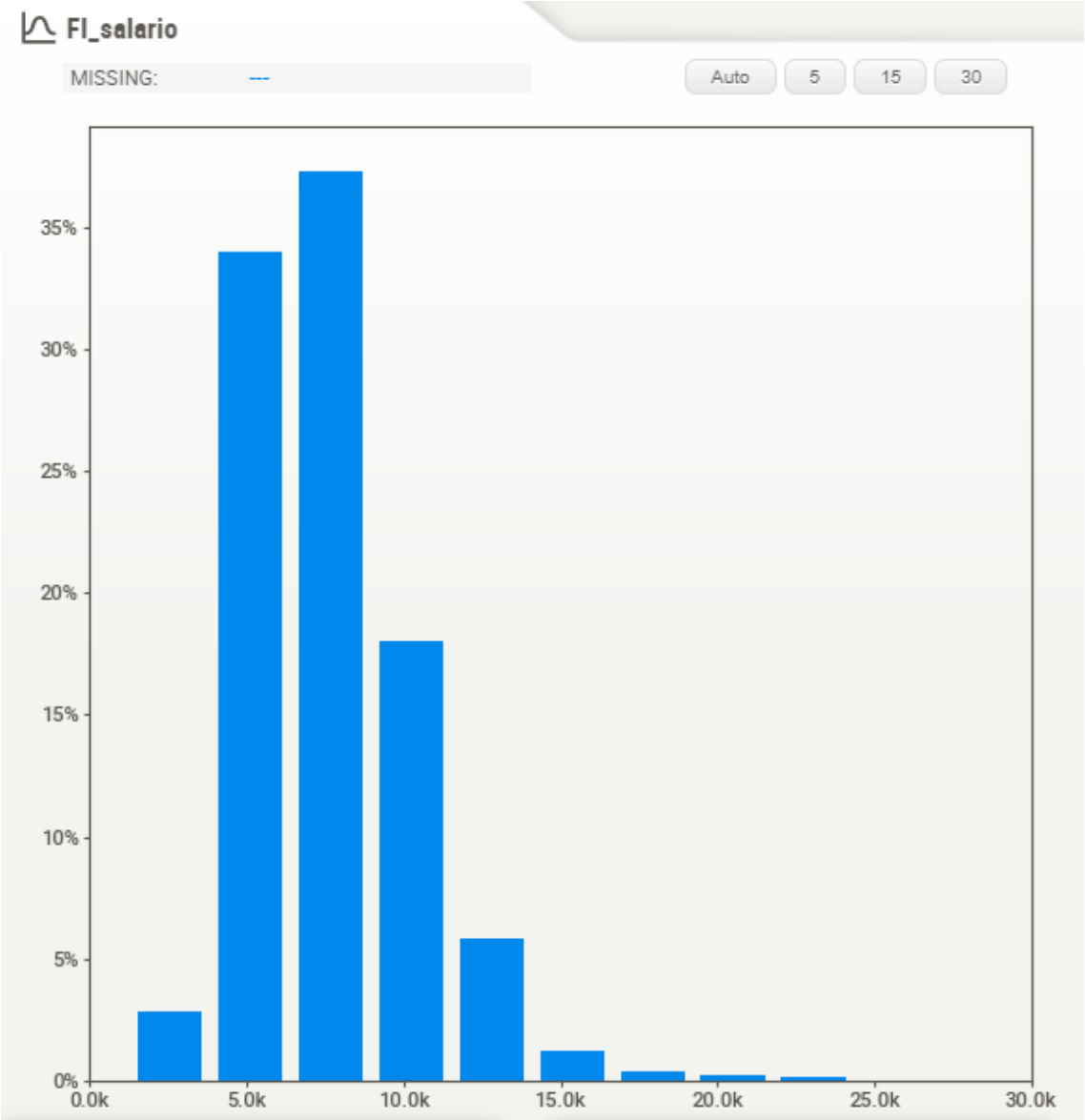
Auto
5
15
30
30%
25%
20%
15%
10%
5%
0%
-10.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0


M_tamano_hogar
MISSING:

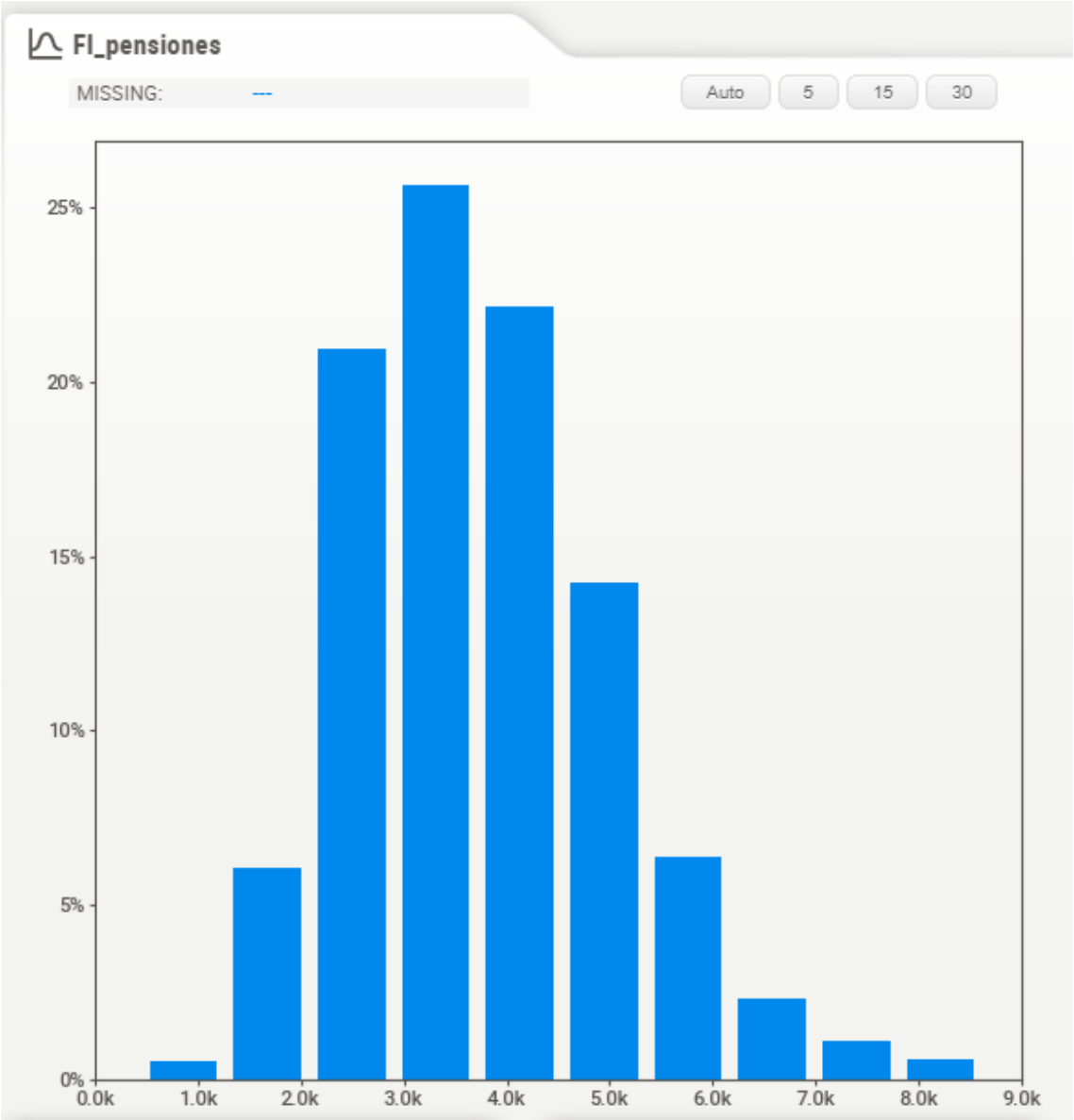
Auto
5
15
30
30%
25%
20%
15%
10%
5%
0%
1.00
1.50
2.00
2.50
3.00
3.50
4.00


hogares_uni_P
MISSING:

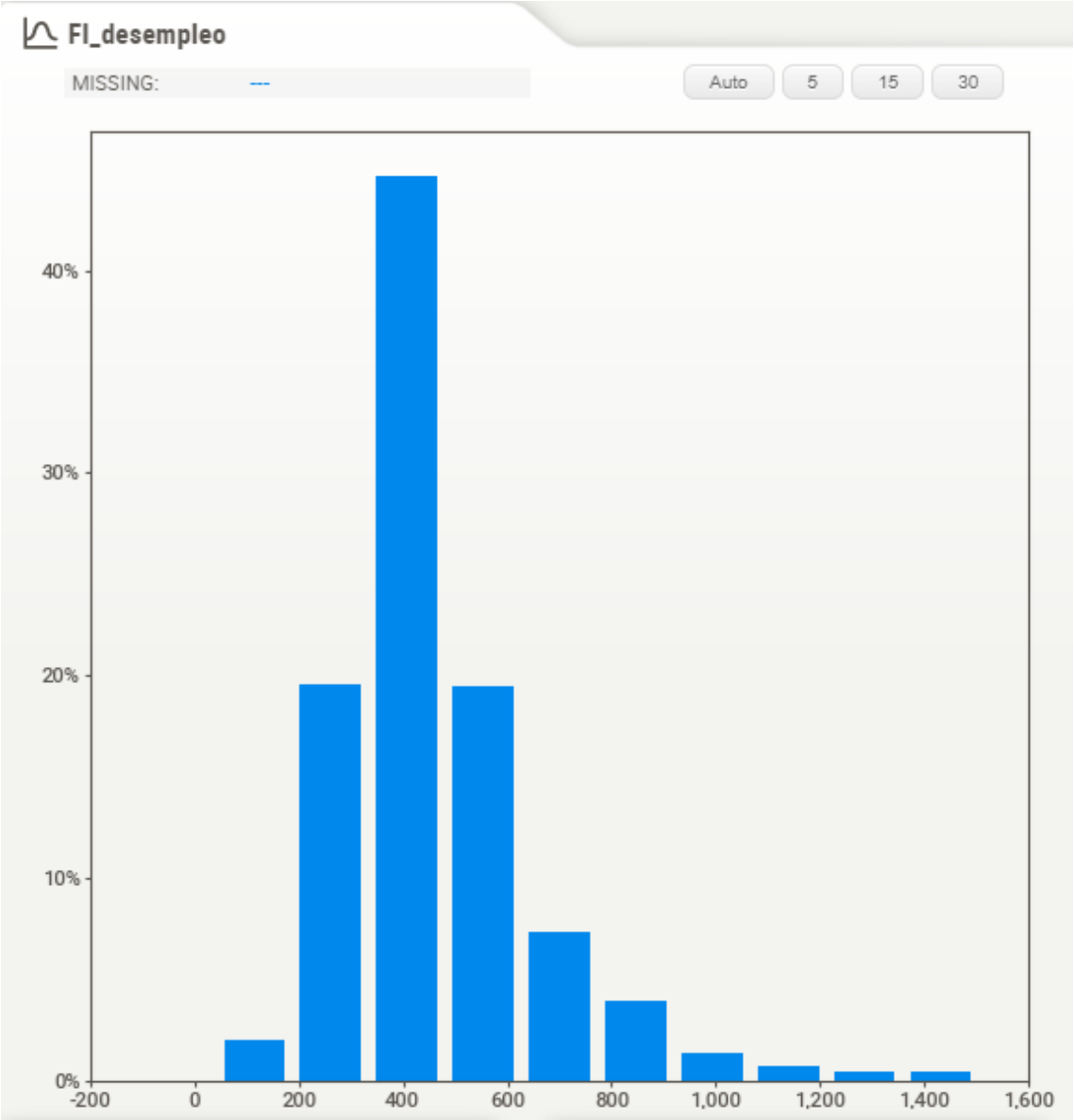
Auto
5
15
30
30%
25%
20%
15%
10%
5%
0%
0
10
20
30
40
50
60
70
80
90
100


Fl_salario
MISSING:

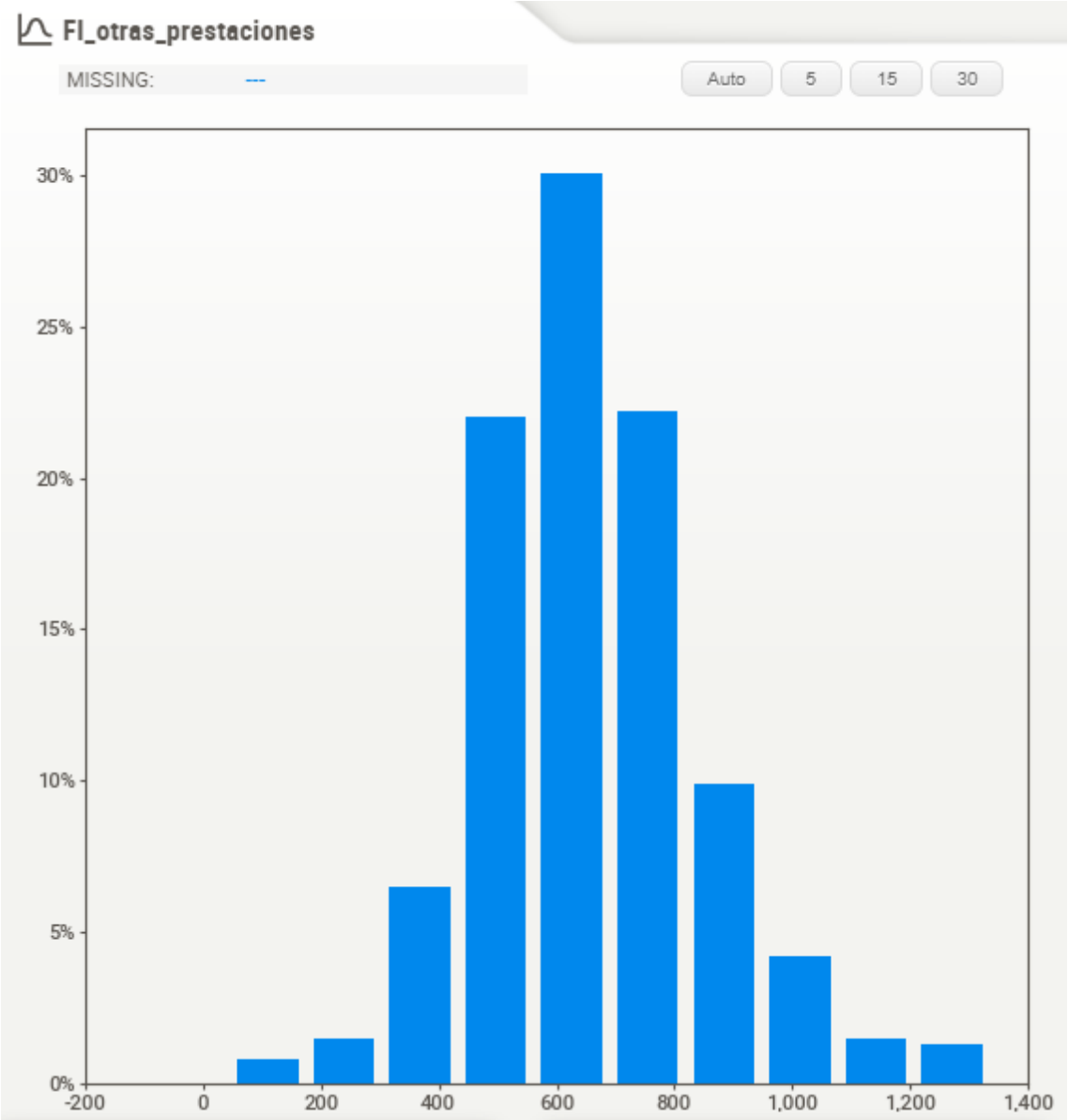
Auto
5
15
30
35%
30%
25%
20%
15%
10%
5%
0%
0.0k
5.0k
10.0k
15.0k
20.0k
25.0k
30.0k


Fl_pensiones
MISSING:

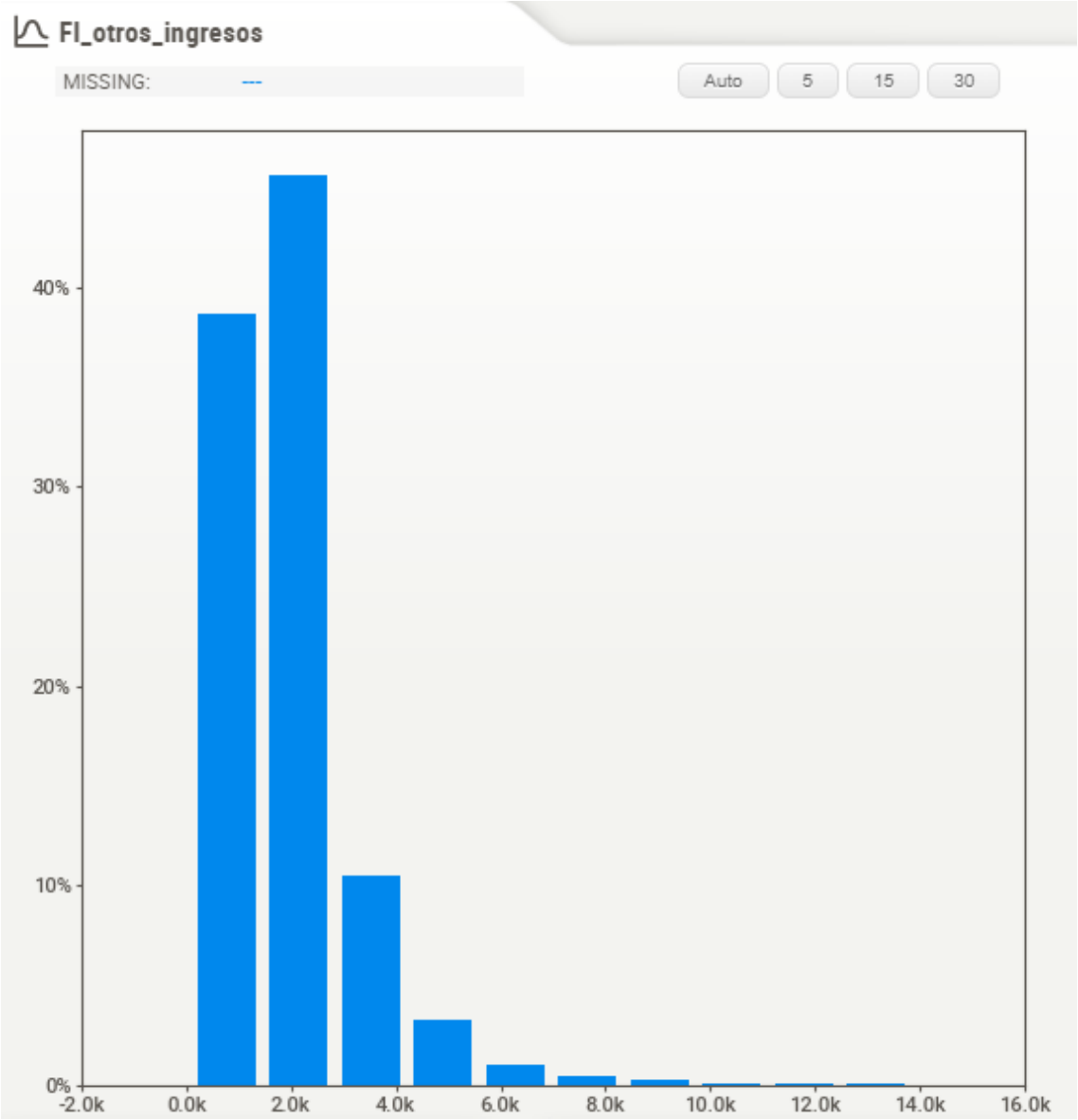
Auto
5
15
30
25%
20%
15%
10%
5%
0%
0.0k
1.0k
2.0k
3.0k
4.0k
5.0k
6.0k
7.0k
8.0k
9.0k


FI_desempleo
MISSING:

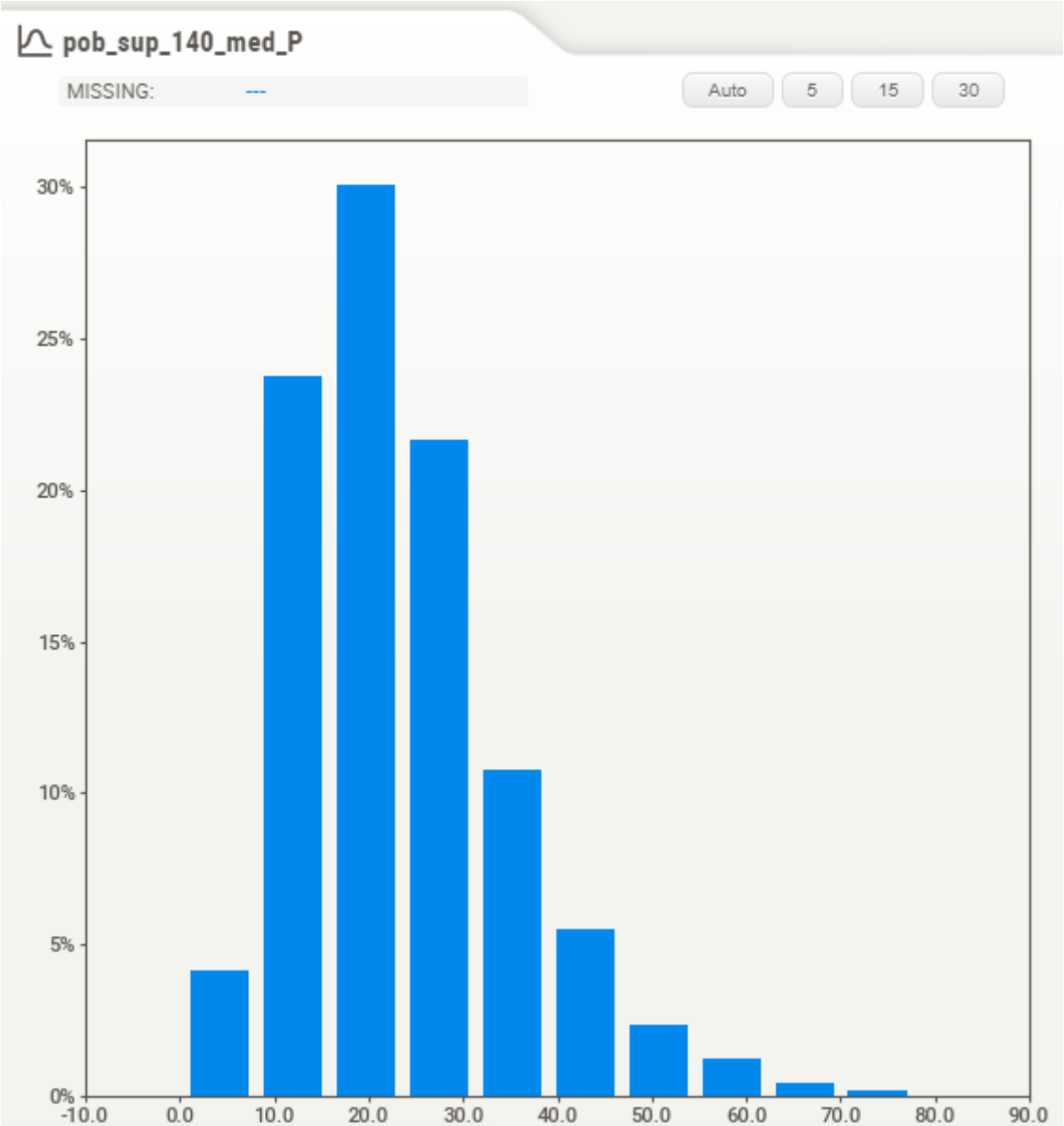
Auto
5
15
30
40%
30%
20%
10%
0%
-200
0
200
400
600
800
1,000
1,200
1,400
1,600


FI_otras_prestaciones
MISSING:

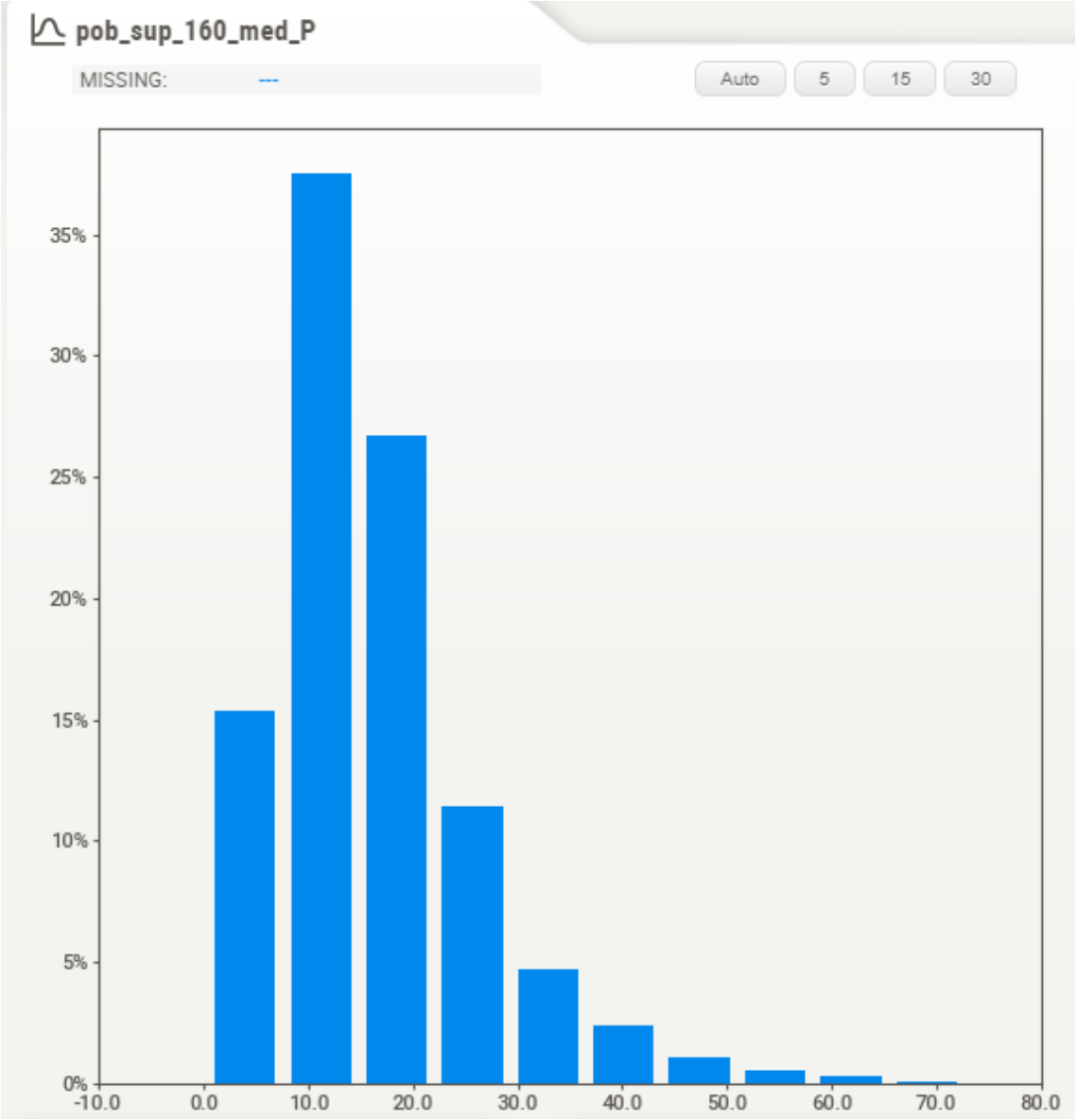
Auto
5
15
30
30%
25%
20%
15%
10%
5%
0%
-200
0
200
400
600
800
1,000
1,200
1,400


Fl_otros_ingresos
MISSING:

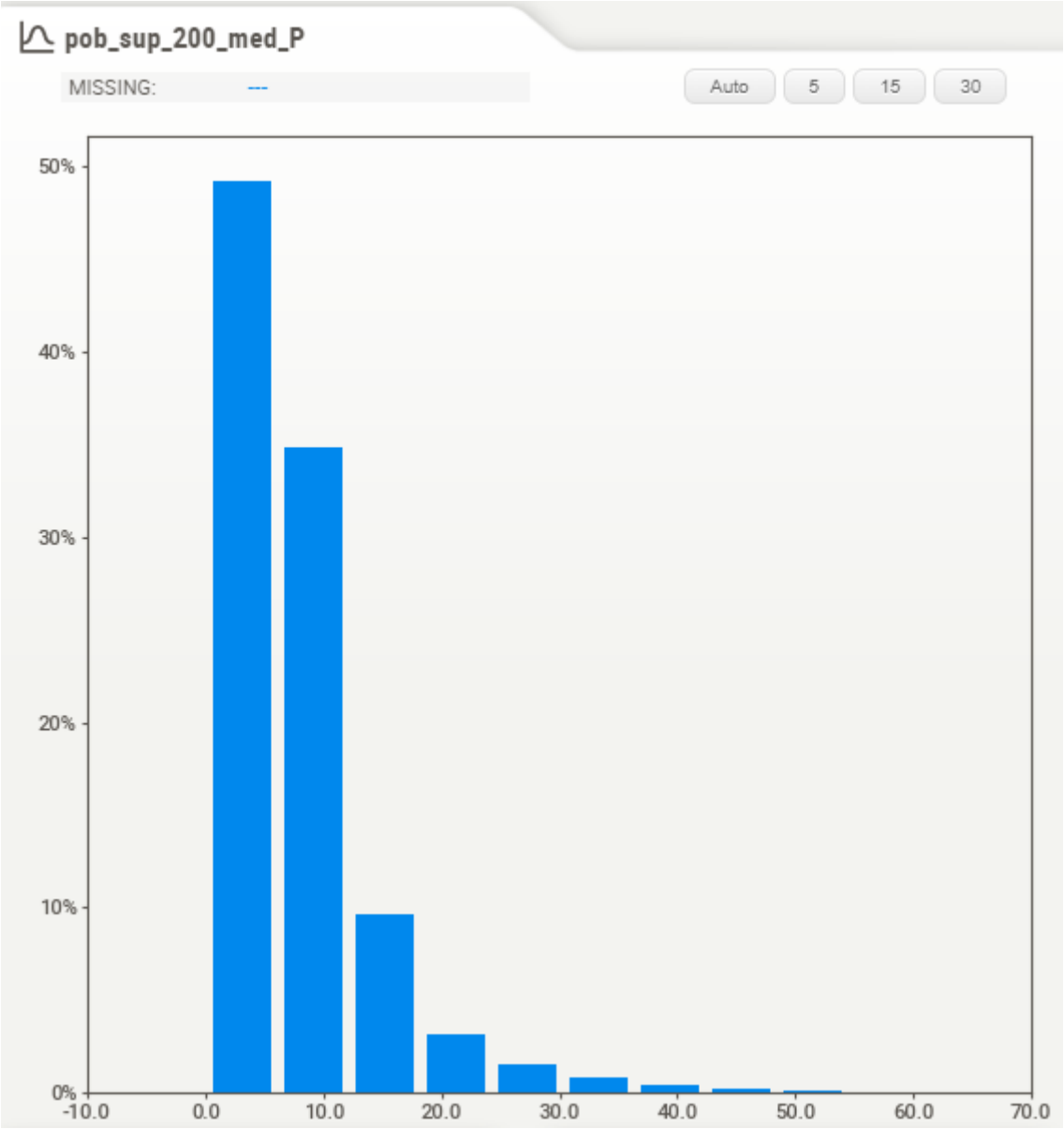
Auto
5
15
30
40%
30%
20%
10%
0%
-2.0k
0.0k
2.0k
4.0k
6.0k
8.0k
10.0k
12.0k
14.0k
16.0k


pob_sup_140_med_P
MISSING:

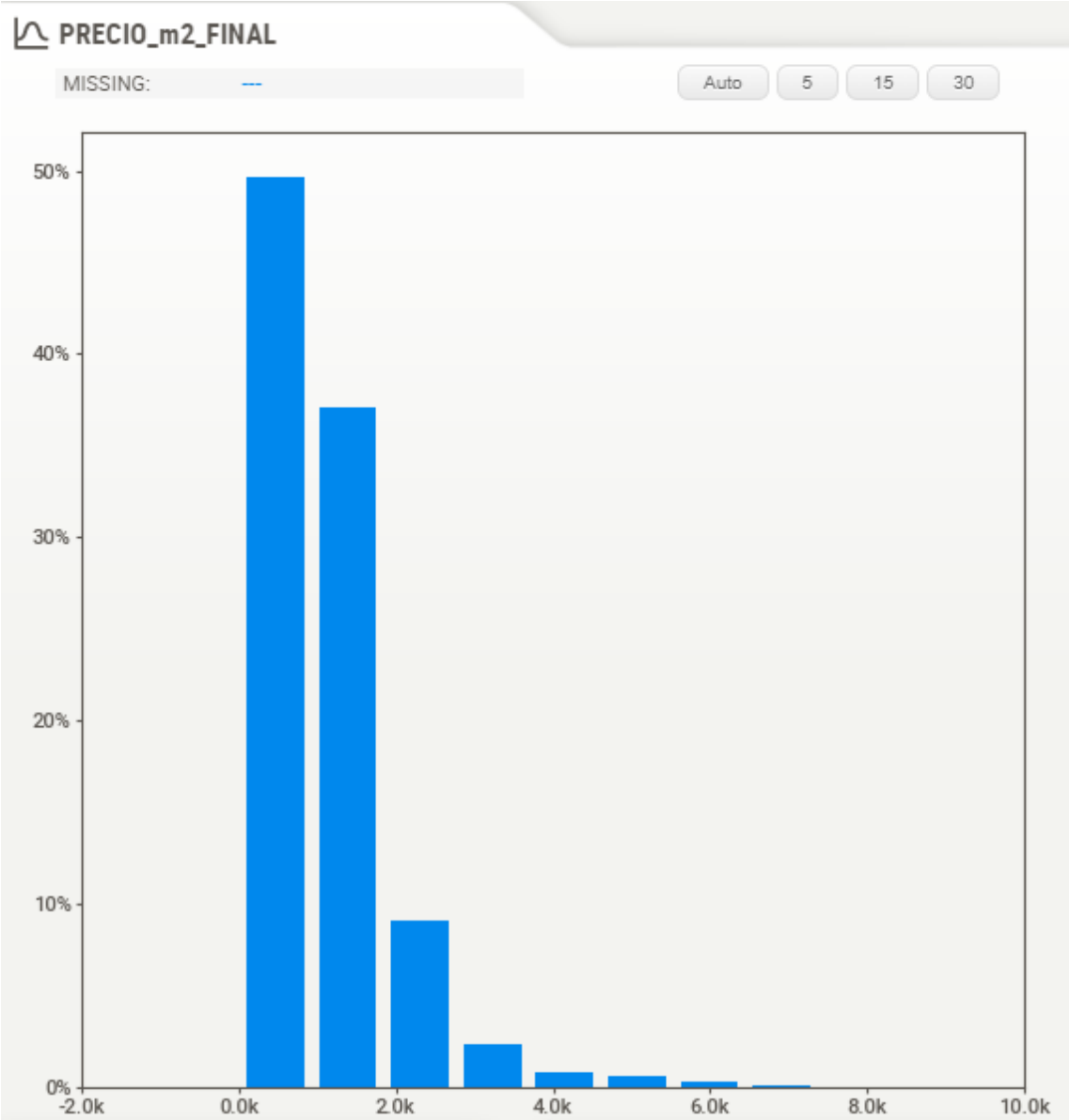
Auto
5
15
30
30%
25%
20%
15%
10%
5%
0%
-10.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0


pob_sup_160_med_P
MISSING:

Auto
5
15
30
35%
30%
25%
20%
15%
10%
5%
0%
-10.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0

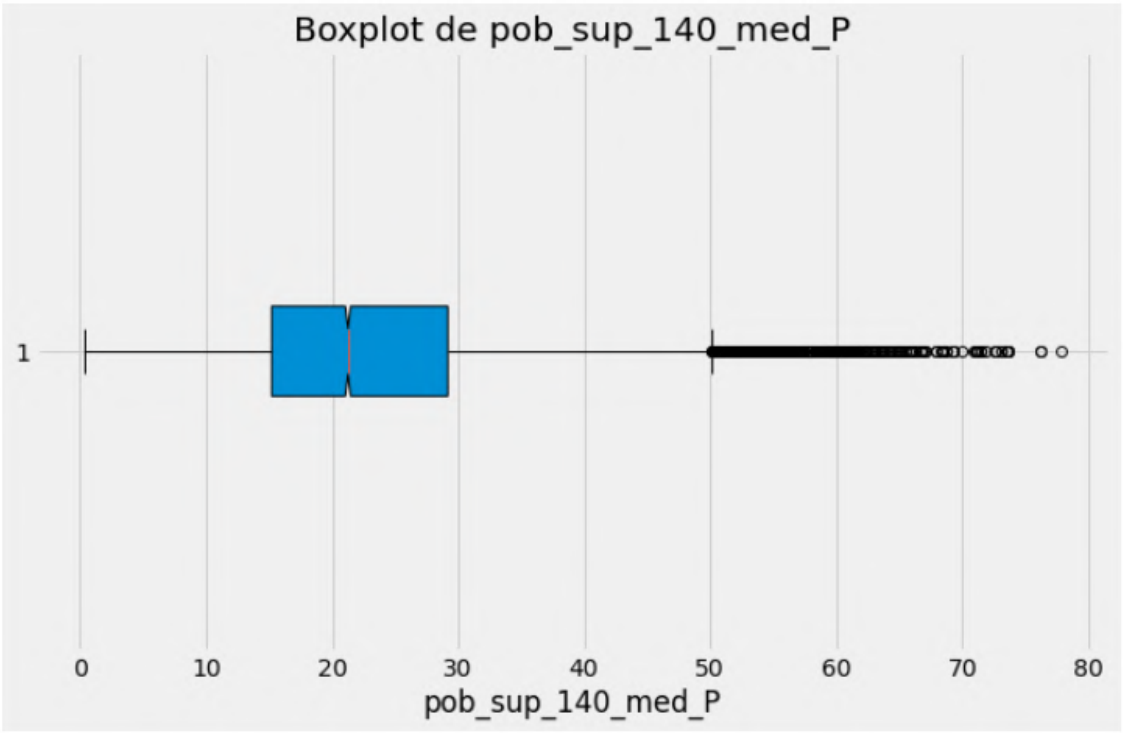

pob_sup_200_med_P
MISSING:

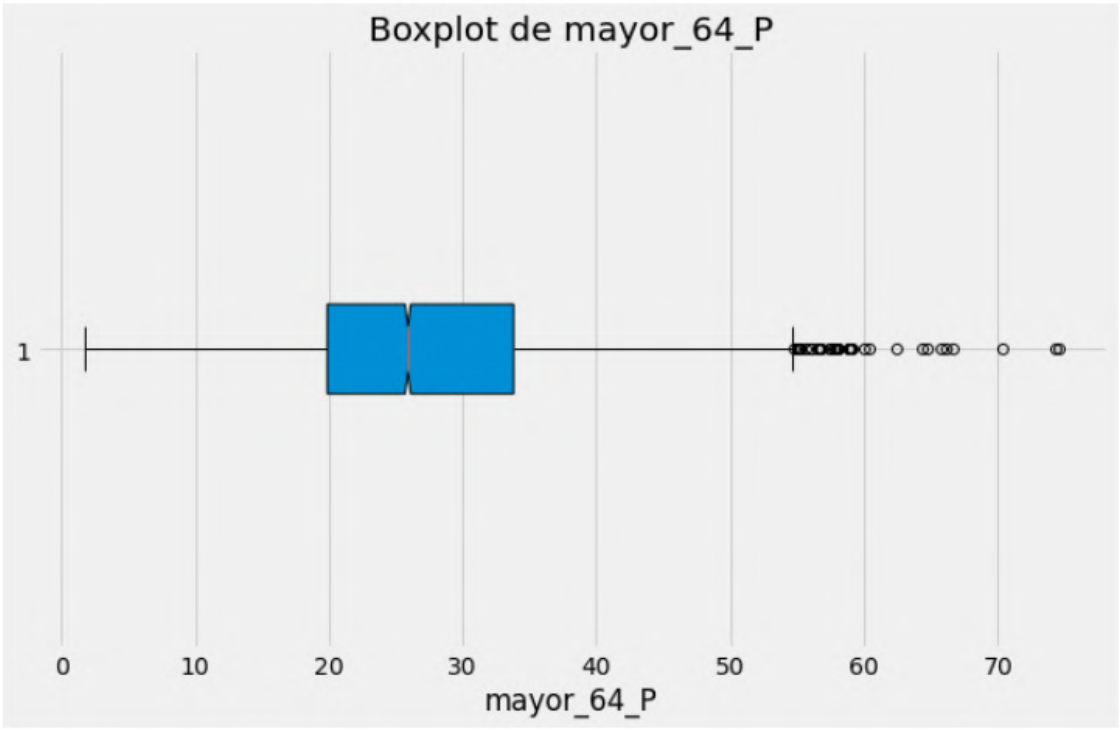
Auto
5
15
30
50%
40%
30%
20%
10%
0%
-10.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0

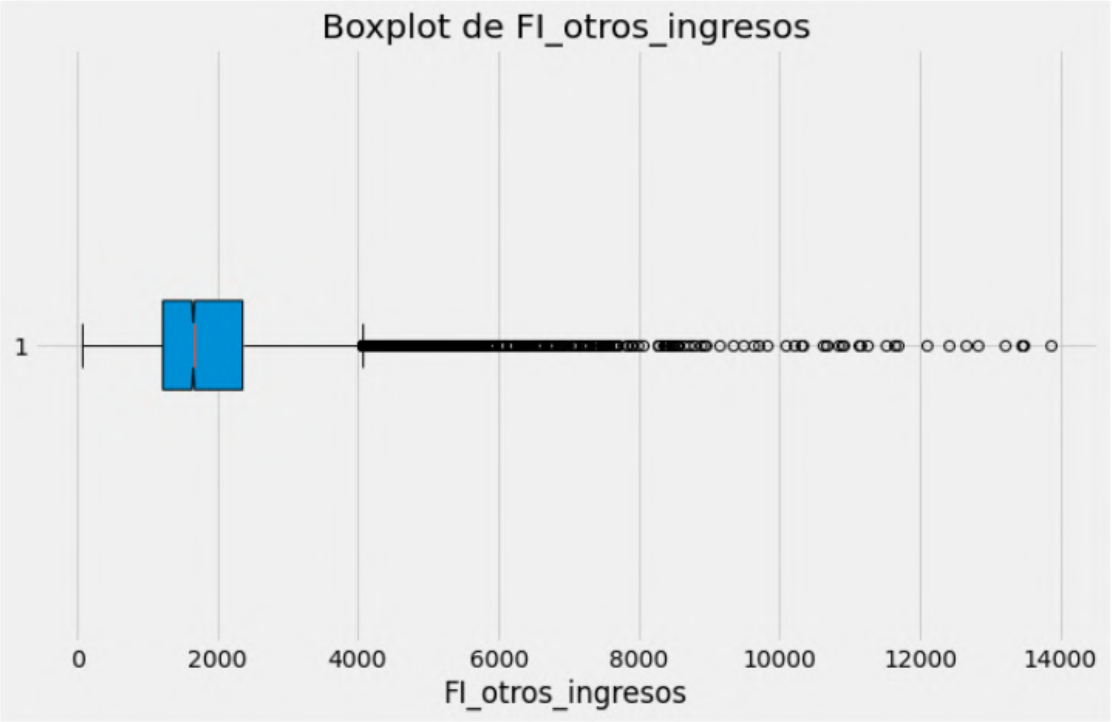

PRECIO_m2_FINAL
MISSING:

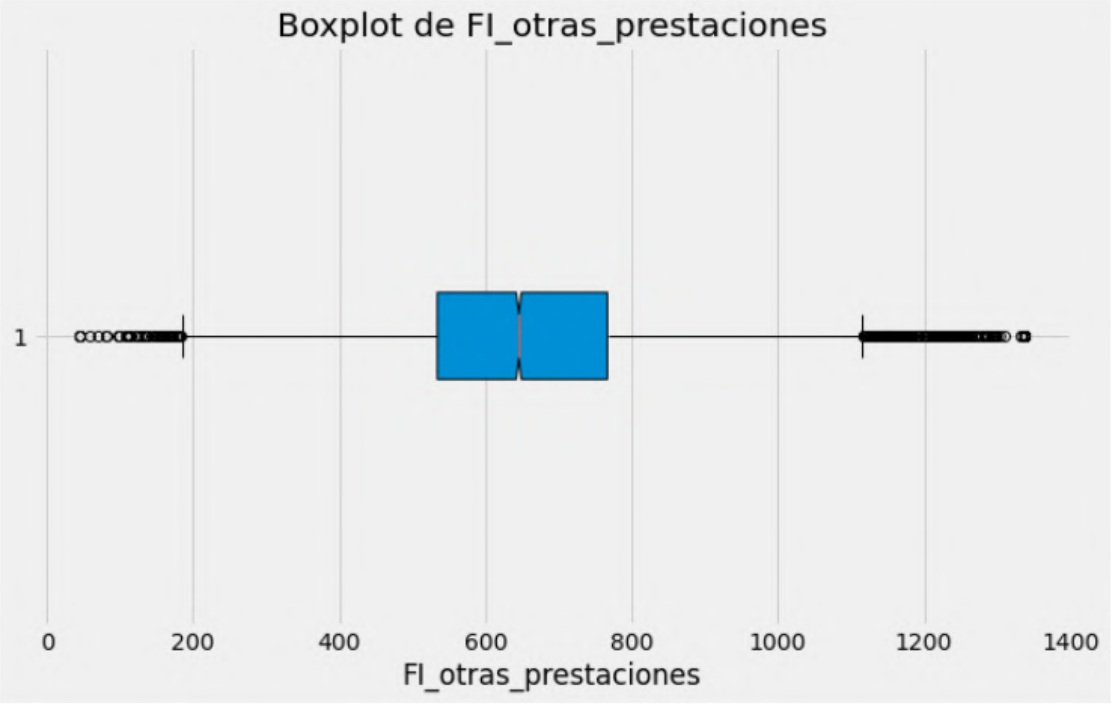
Auto
5
15
30
50%
40%
30%
20%
10%
0%
-2.0k
0.0k
2.0k
4.0k
6.0k
8.0k
10.0k

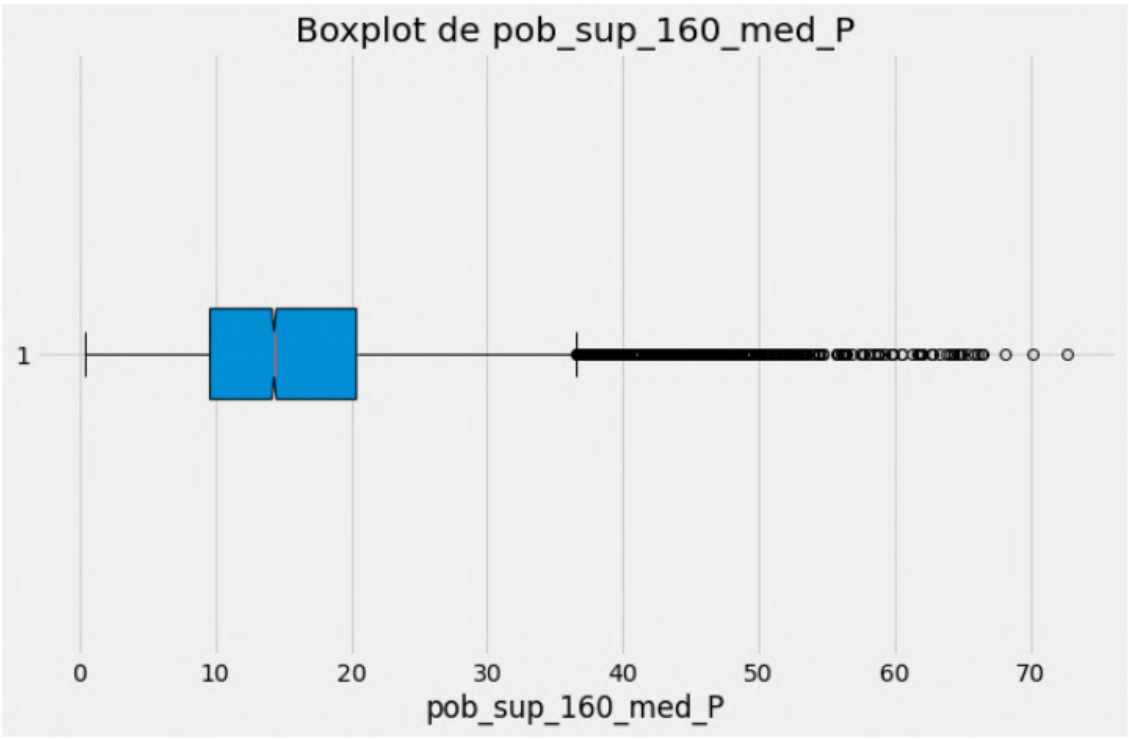
Diagramas de caja y bigotes

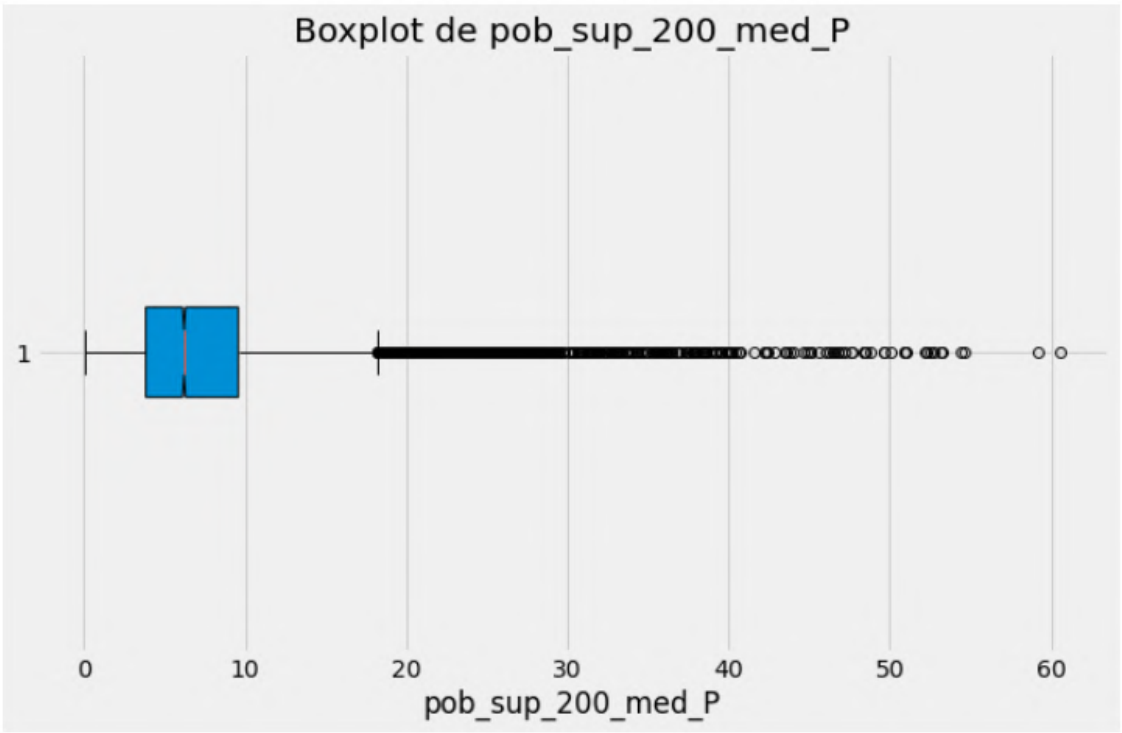


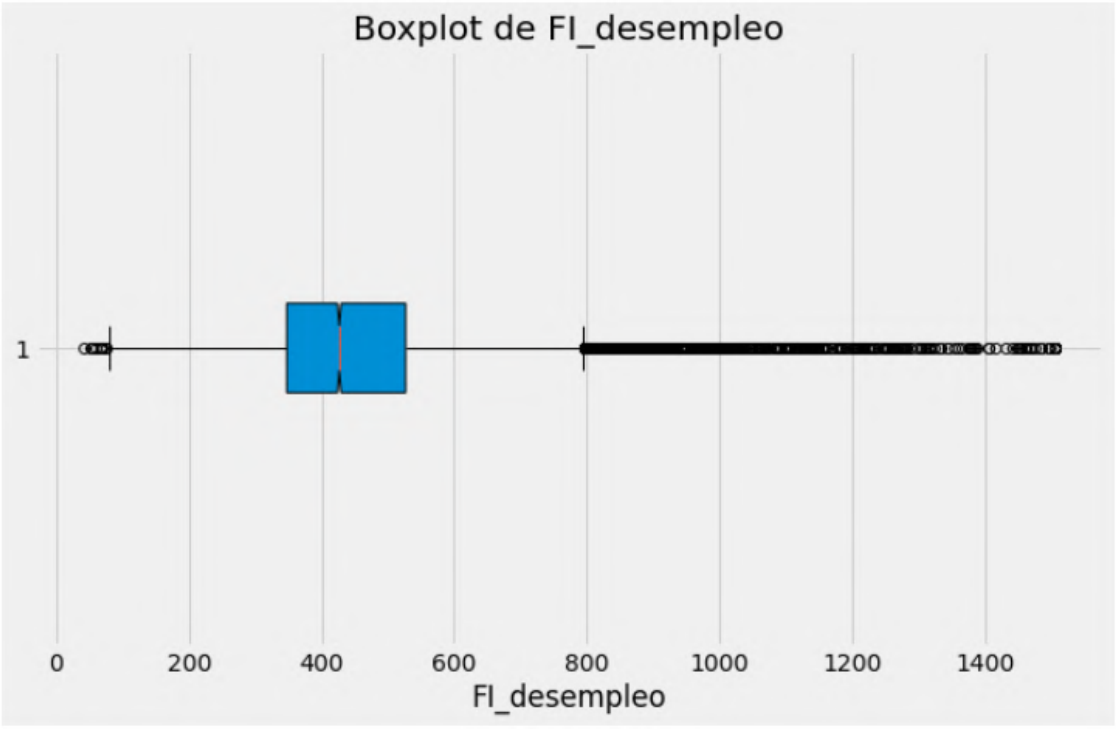


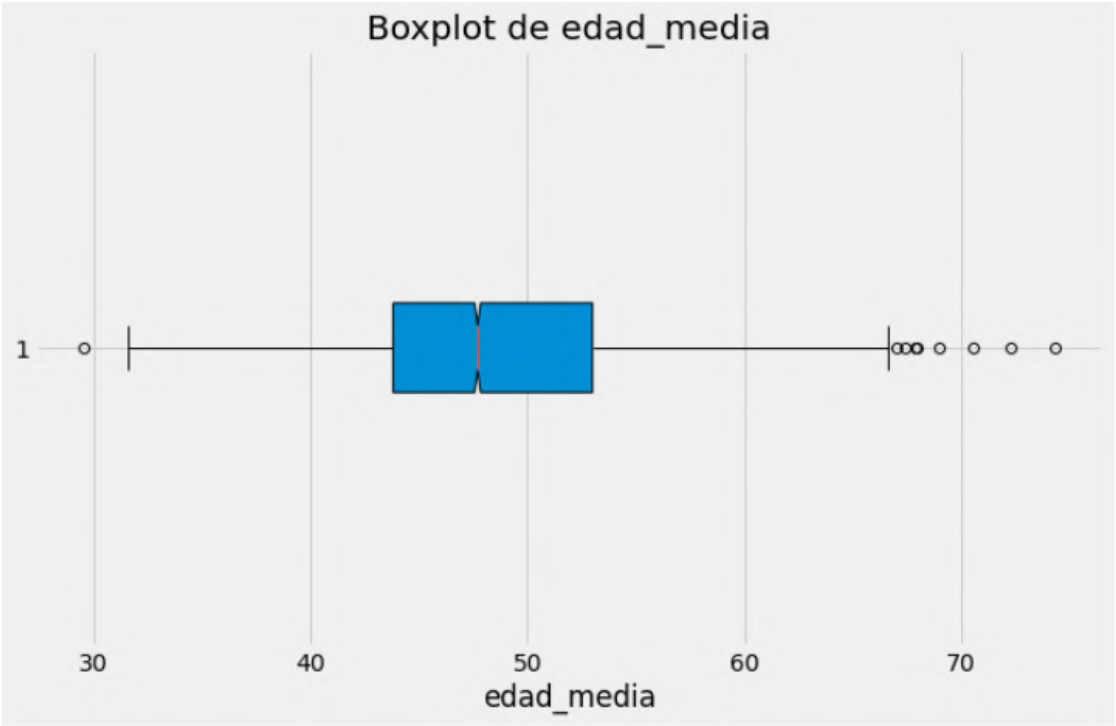

Boxplot de FI_otros_ingresos
1
0
2000
4000
6000
8000
10000
12000
14000
FI_otros_ingresos

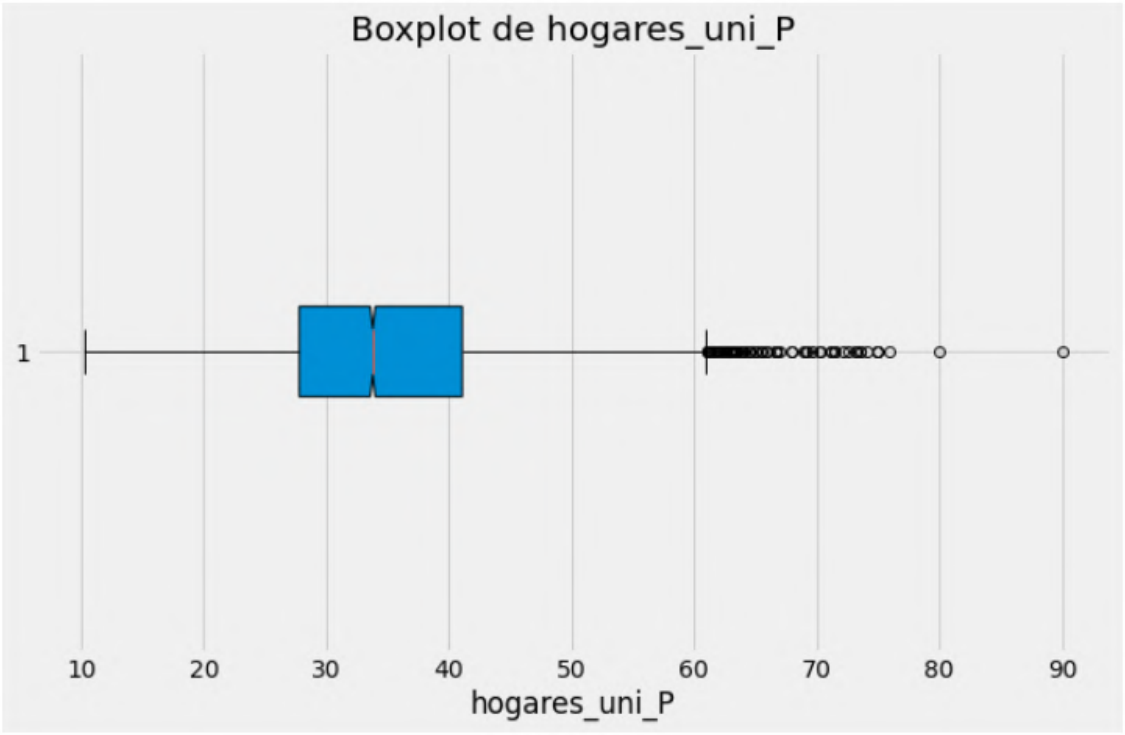

Boxplot de FI_otras_prestaciones
1
0
200
400
600
800
1000
1200
1400
FI_otras_prestaciones

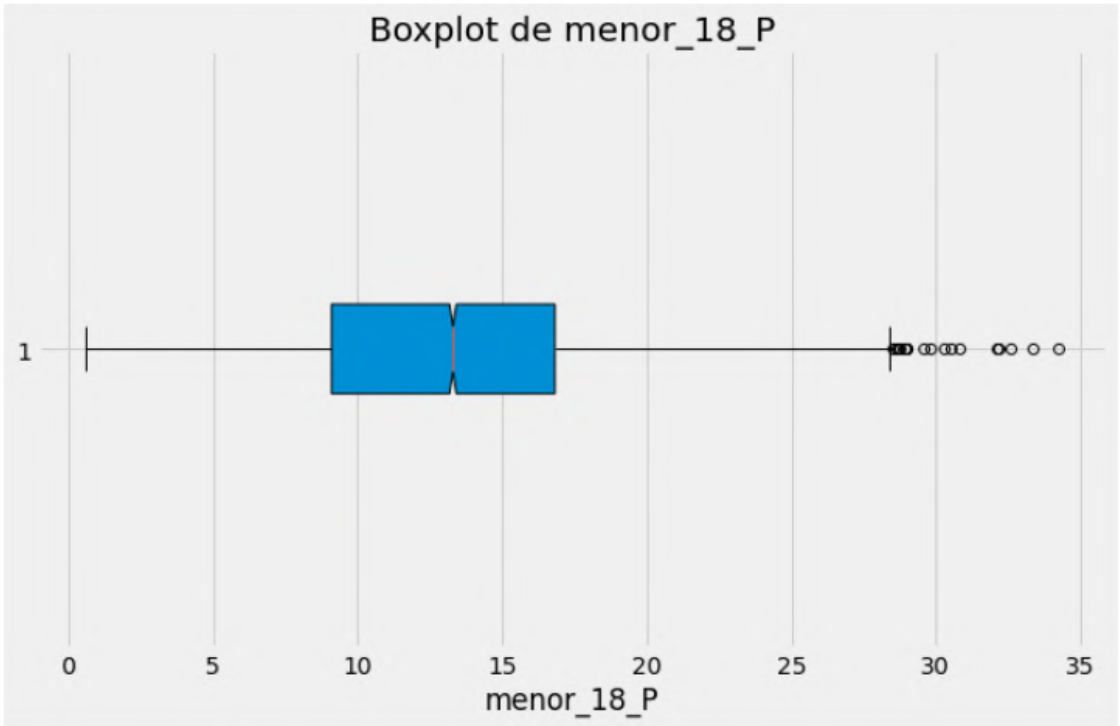

Boxplot de pob_sup_160_med_P
1
0
10
20
30
40
50
60
70
pob_sup_160_med_P

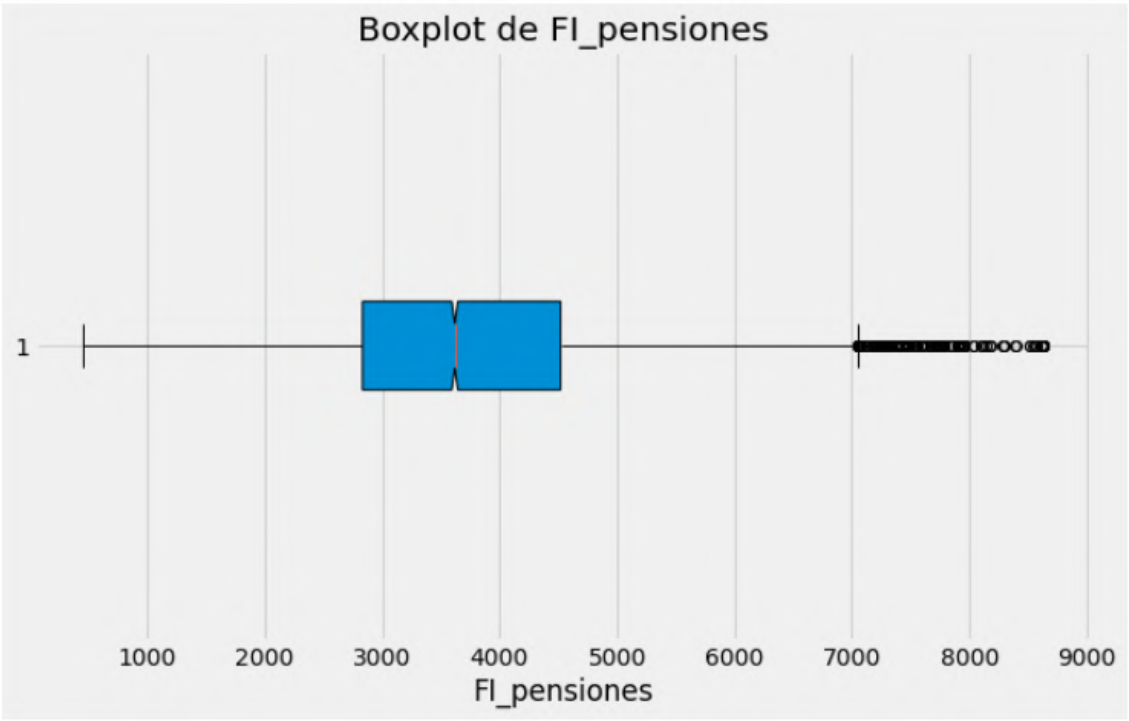

Boxplot de pob_sup_200_med_P
1
0
10
20
30
40
50
60
pob_sup_200_med_P

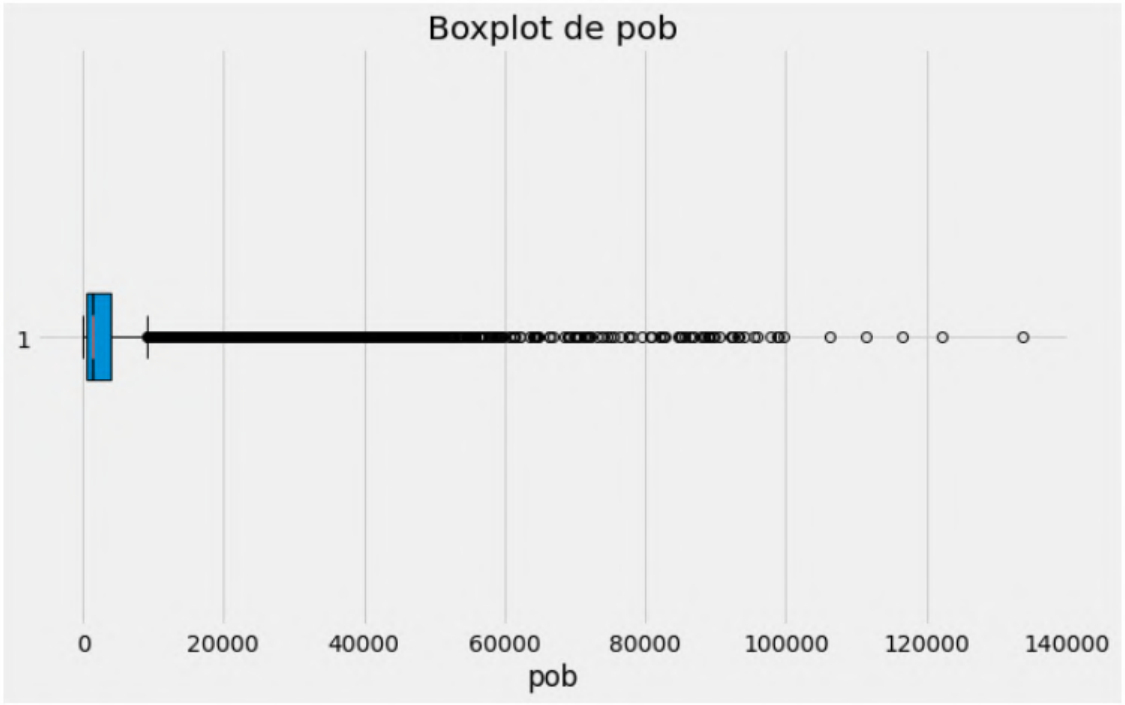

Boxplot de FI_desempleo
1
0
200
400
600
800
1000
1200
1400
FI_desempleo

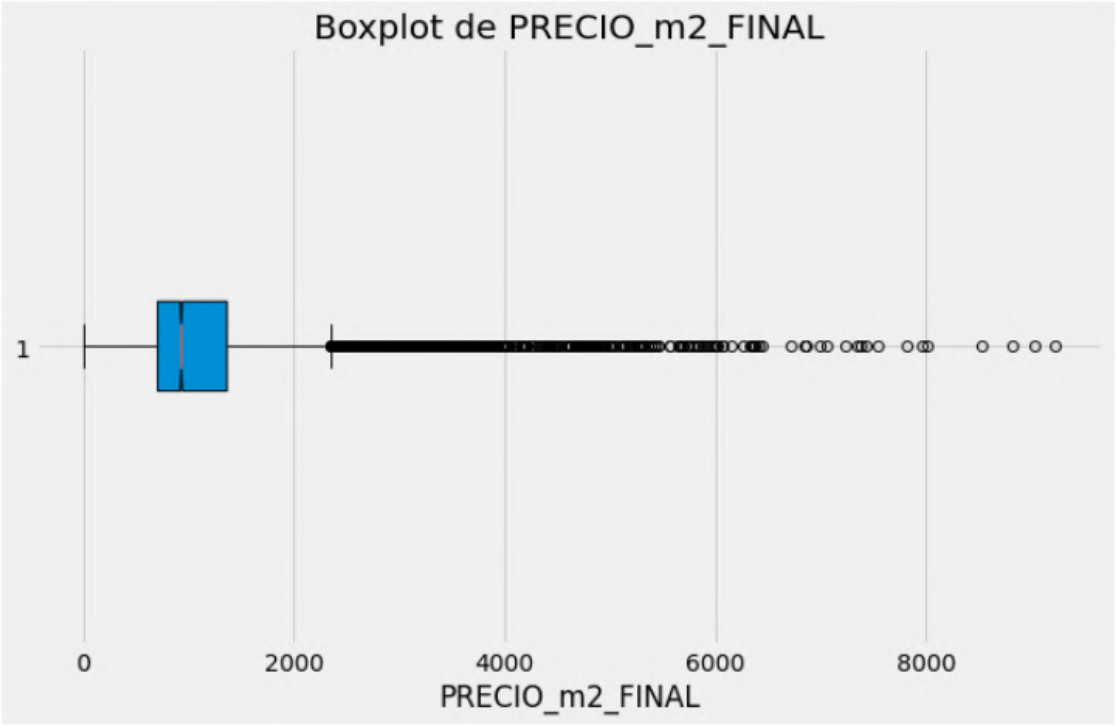

Boxplot de edad_media
1
30
40
50
60
70
edad_media

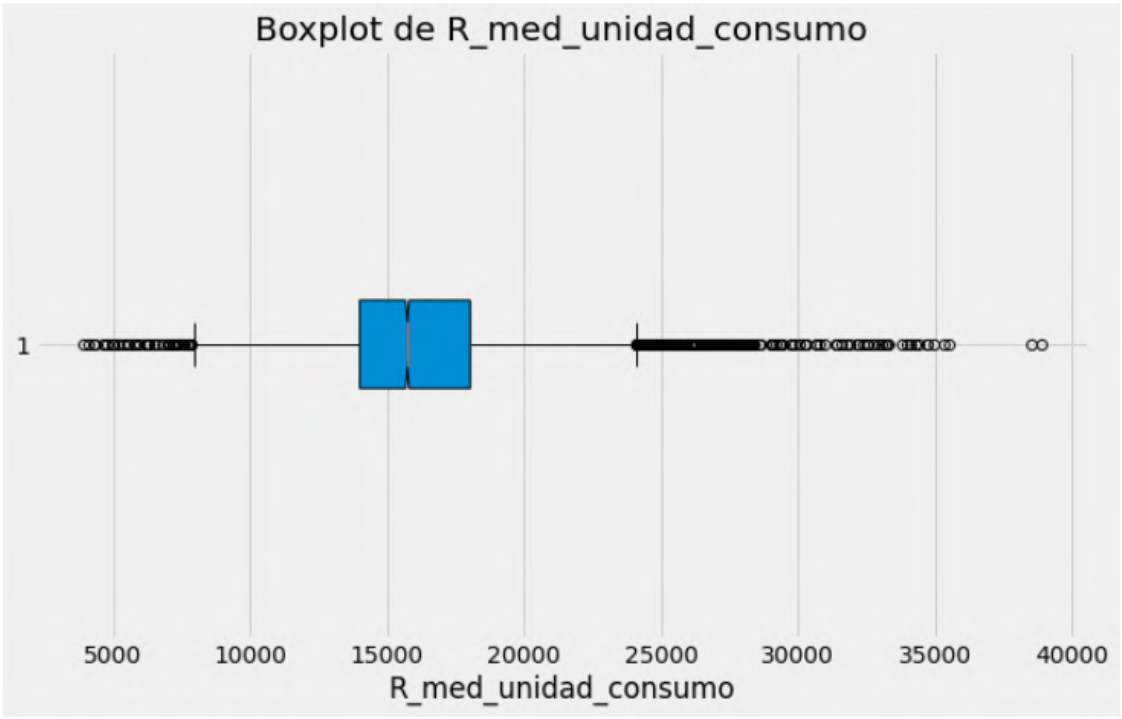

Boxplot de hogares_uni_P
1
10
20
30
40
50
60
70
80
90
hogares_uni_P

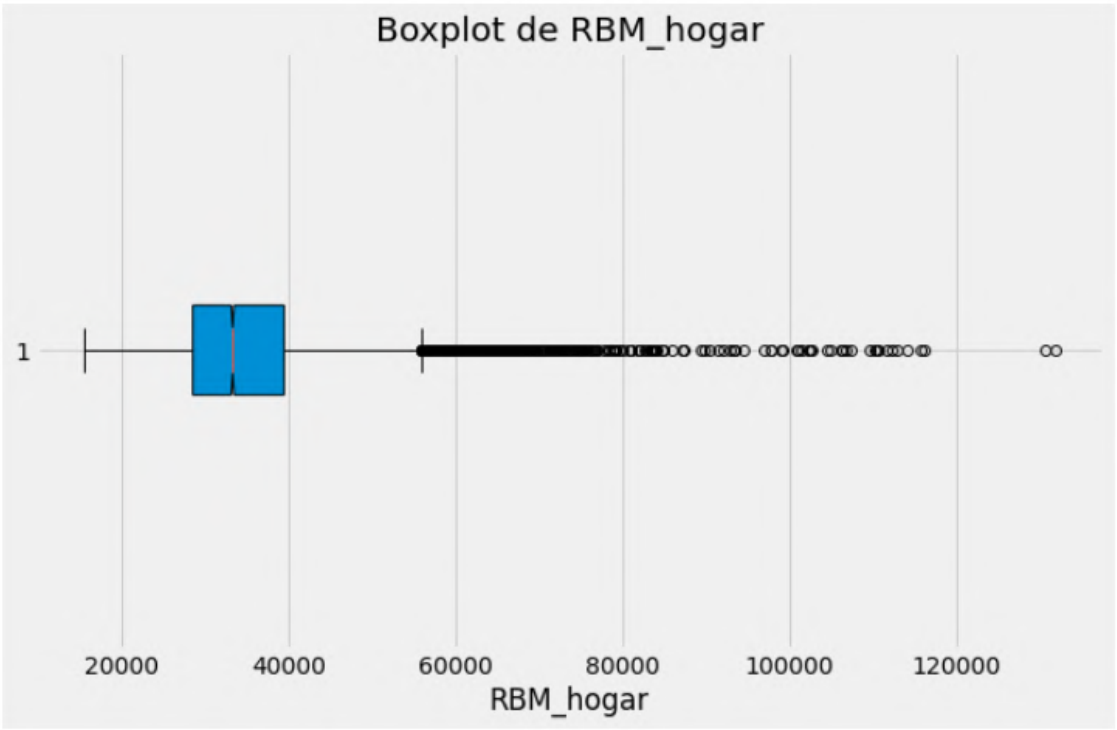

Boxplot de menor_18_P
1
0
5
10
15
20
25
30
35
menor_18_P

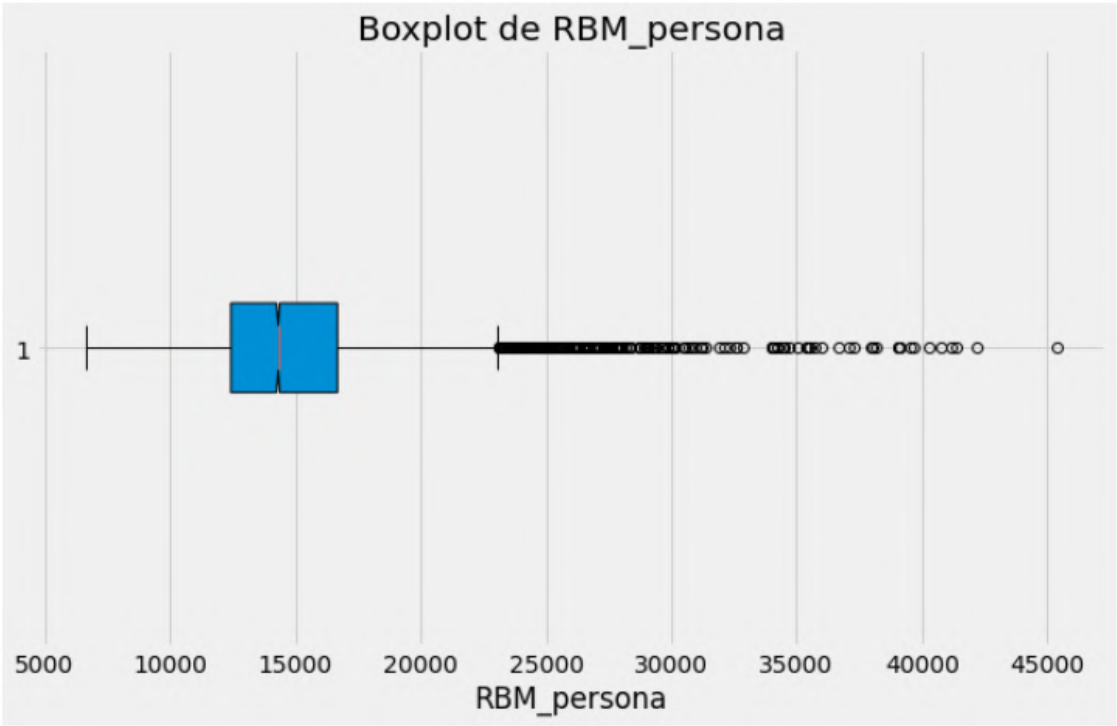

Boxplot de FI_pensiones
1
1000
2000
3000
4000
5000
6000
7000
8000
9000
FI_pensiones

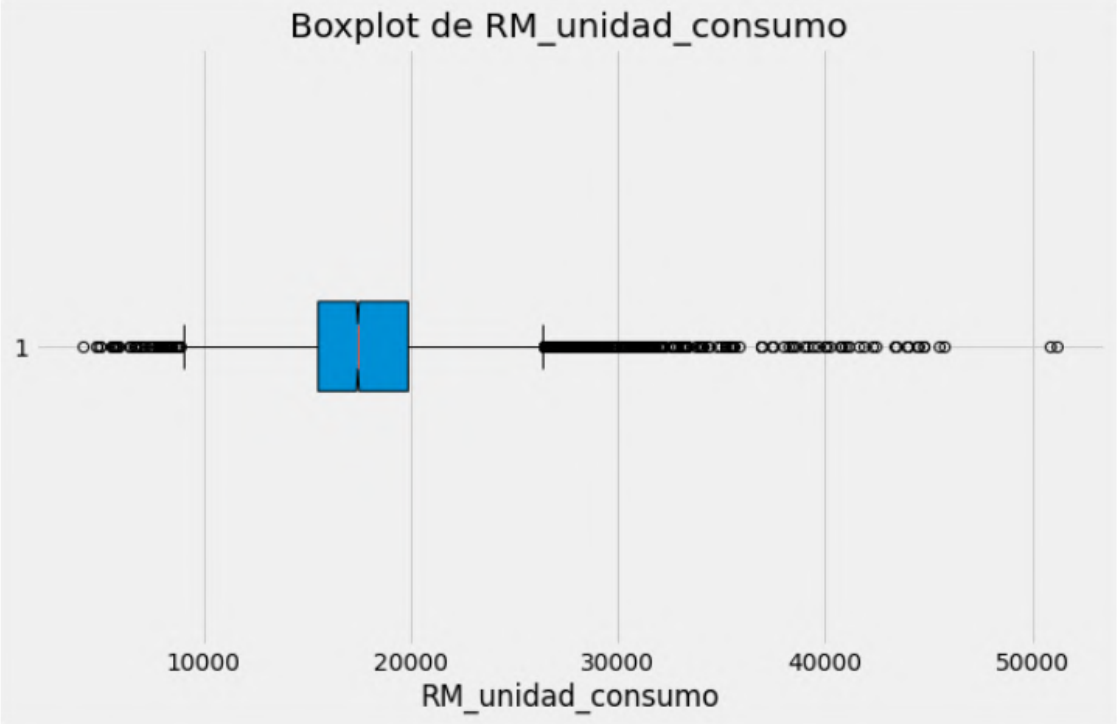

Boxplot de pob
1
0
20000
40000
60000
80000
100000
120000
140000
pob

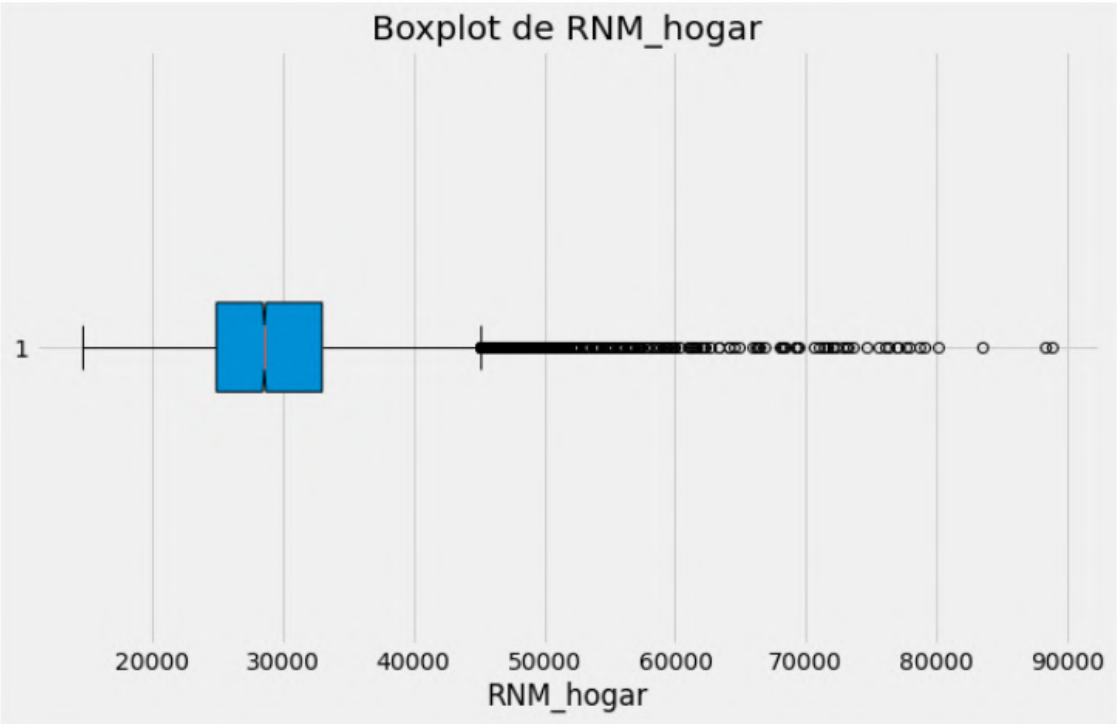

Boxplot de PRECIO_m2_FINAL
1
0
2000
4000
6000
8000
PRECIO_m2_FINAL

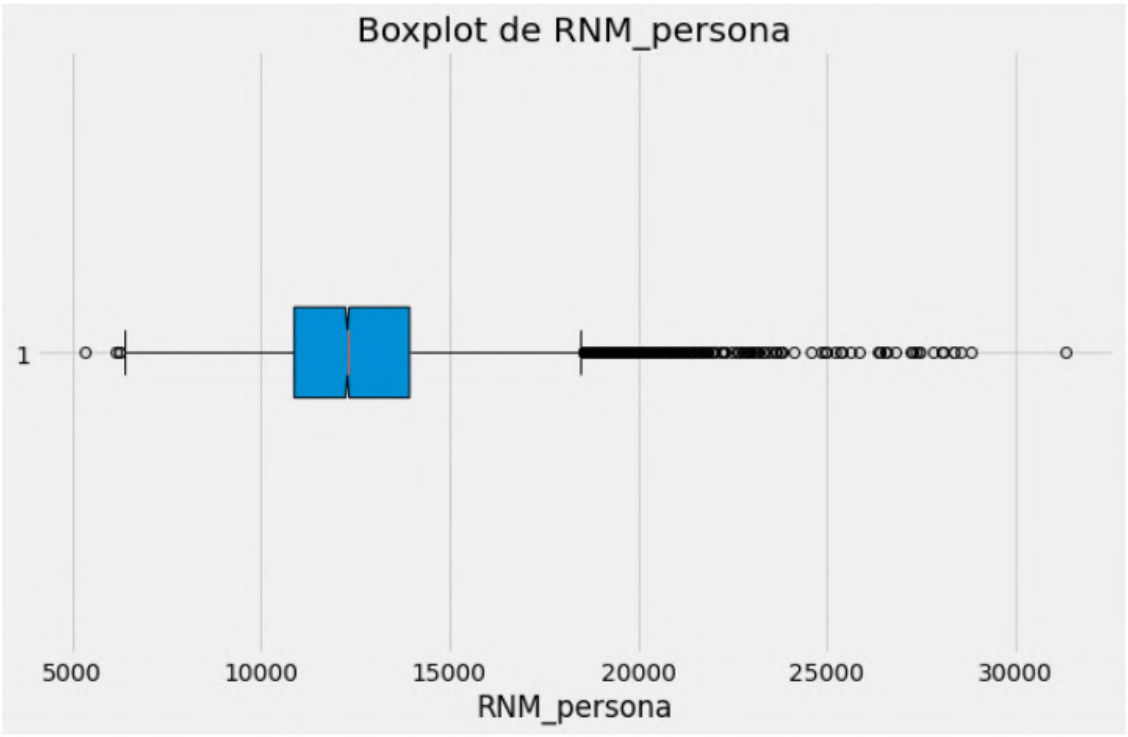

Boxplot de R_med_unidad_consumo
1
5000
10000
15000
20000
25000
30000
35000
40000
R_med_unidad_consumo

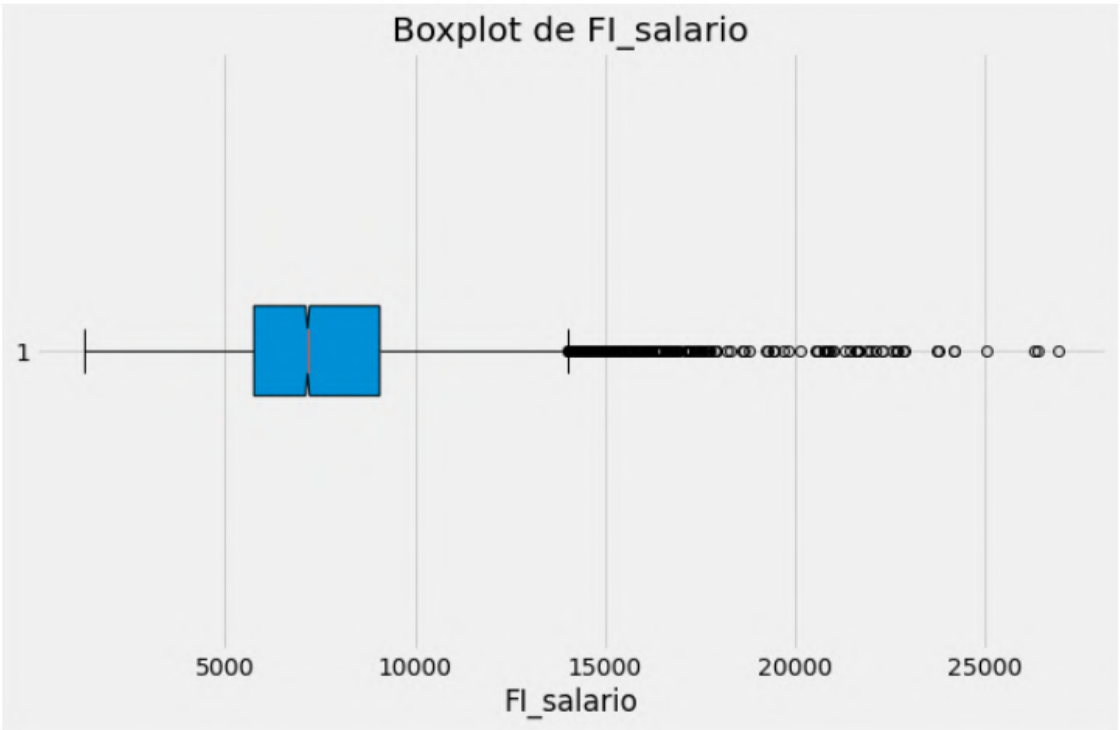

Boxplot de RBM_hogar
1
20000
40000
60000
80000
100000
120000
RBM_hogar

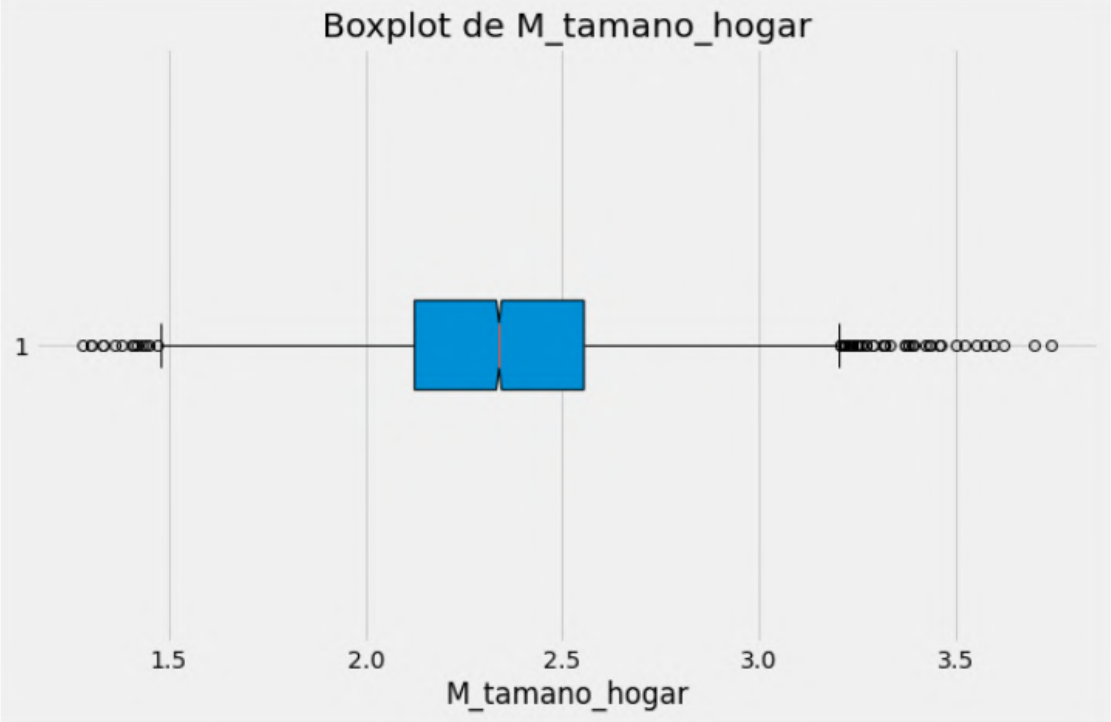

Boxplot de RBM_persona
1
5000
10000
15000
20000
25000
30000
35000
40000
45000
RBM_persona


Boxplot de RM_unidad_consumo
1
10000
20000
30000
40000
50000
RM_unidad_consumo


Boxplot de RNM_hogar
1
20000
30000
40000
50000
60000
70000
80000
90000
RNM_hogar


Boxplot de RNM_persona
1
5000
10000
15000
20000
25000
30000
RNM_persona


Boxplot de FI_salario
1
5000
10000
15000
20000
25000
FI_salario


Boxplot de M_tamano_hogar
1
1.5
2.0
2.5
3.0
3.5
M_tamano_hogar