uc3m | Universidad Carlos III de Madrid

Máster Universitario en Ciencias Actuariales y Financieras
Curso Académico 2022-2023

Trabajo Fin de Máster

"Segmentación de una cartera de autos empleando técnicas de Machine Learning con aprendizaje no supervisado"

Alicia Pérez Sánchez

Tutor/es

José Miguel Rodríguez - Pardos del Castillo

Jesús Ramón Simón del Potro

Madrid, a junio de 2023

DETECCIÓN DEL PLAGIO

La Universidad utiliza el programa **Turnitin Feedback Studio** para comparar la originalidad del trabajo entregado por cada estudiante con millones de recursos electrónicos y detecta aquellas partes del texto copiadas y pegadas. Copiar o plagiar en un TFM es considerado una **Falta Grave**, y puede conllevar la expulsión definitiva de la Universidad.

Texto primera página:

RESUMEN

Cada vez más, las empresas cuya actividad se engloba dentro del sector asegurador necesitan de la Inteligencia Artificial para entender mejor su negocio y poder, de esta manera, alcanzar sus objetivos de rentabilidad de manera más eficiente.

Este trabajo tiene como finalidad ayudar a una determinada compañía de seguros de no vida en la identificación de grupos de clientes para poder poseer un mayor conocimiento de sus asegurados, así como utilizar dicha segmentación para estrategias de gestión de cartera, además de acciones de tarifa y suscripción.

Para ello, se ha hecho uso de una base de datos relativos a una serie de características declaradas por los asegurados de dicha compañía que poseen un seguro de auto para, posteriormente, emplear métodos no supervisados de Inteligencia Artificial.

Palabras claves

Auto, Inteligencia Artificial, Clúster, K-Means, K-Prototypes.

ABSTRACT

Increasingly, companies whose activity is included in the insurance sector need Artificial Intelligence to better understand their business and, thus, be able to achieve their profitability objectives in a more efficiently way.

The purpose of this work is to help a certain non-life insurance company in the identification of customer groups in order to have a better knowledge of its policyholders and use the segmentation for portfolio management strategies and tariff actions and subscription.

For doing so, it has been used a database of different characteristics declared by the policyholders of the company who have car insurance, using unsupervised methods of Artificial Intelligence.

Keywords

Auto, Artificial Intelligence, Cluster, K-Means, K-Prototypes.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

1.1 Motivación del trabajo

En la actualidad, la adopción de la Inteligencia Artificial en las empresas es fundamental para asegurar su supervivencia en el futuro. De esta manera, las compañías aseguradoras están haciendo uso de ella con el fin de mejorar sus operaciones y servicios, ofreciendo mayor precisión y personalización, a la vez que eficiencia en las diferentes áreas.

Así, los principales ámbitos donde las compañías aseguradoras aplican y se espera que perfeccionen los diferentes algoritmos de la Inteligencia Artificial son:

- Procesos de suscripción y siniestros: la Inteligencia Artificial mejorará los procesos de suscripción tanto a nivel de eficiencia como de automatización. Además, se espera que permita identificar y suscribir riesgos emergentes y descubra nuevas fuentes de ingresos.

  Por otro lado, gracias a la Inteligencia Artificial se agilizarán los procesos de reclamaciones, pues esta es capaz de analizar y procesar dichas reclamaciones automáticamente, sin necesidad de la intervención humana.

- Análisis de riesgos y tarificación: gracias a la Inteligencia Artificial permitirá a las compañías la tarificación de los riesgos a los que hacen frente, pudiendo adaptar mejor el precio a cada póliza en función de factores de riesgo más específicos y personalizados. De esta forma, la Inteligencia Artificial es capaz de analizar gran cantidad de datos pudiendo encontrar patrones y correlaciones que, en muchos casos, la especie humana no sería capaz de identificarlos.

- Detección de conductas fraudulentas: al igual que en el punto anterior, al poder la Inteligencia Artificial analizar grandes volúmenes de datos, es capaz de detectar patrones y comportamientos fraudulentos, identificando falsas reclamaciones y comportamientos por parte de los asegurados.

- Mejora de la experiencia del cliente: a través de la Inteligencia Artificial se acelerará la resolución de las consultas de los clientes. A su vez, mejorarán en los procesos de realización de cotizaciones del seguro, proporcionará

información acerca de determinadas pólizas ya existentes, así como asistencia en tiempo real.

Por otro lado, la Inteligencia Artificial permitirá a las aseguradoras comprender mejor las necesidades y preferencias que tengan sus clientes ofreciendo recomendaciones individualizadas de productos y servicios, además de perfeccionar las pólizas en cartera.

A pesar de todas las ventajas que pueda brindar la Inteligencia Artificial a las compañías aseguradoras, hay que tener en cuenta que presenta numerosos retos para el sector seguros como, por ejemplo, cuestiones éticas o privacidad de los datos de los asegurados.

A raíz de la importancia de actualizarse para no dejar de ser competitivos en el mercado, surgió la idea de realizar una identificación de los clientes de una entidad aseguradora de no vida de manera técnica y estadística, con el objetivo de ser más eficiente en la segmentación de estos para realizar mejores tomas de decisión dentro de la empresa.

1.2 Objetivos

La aseguradora la cual ha transferido sus datos y se ha hecho uso de ellos para realizar el trabajo aquí presente, en la actualidad, con el objetivo de llevar a cabo la identificación de sus clientes, está empleando una variable de segmentación que fue confeccionada en función de la experiencia previa del equipo y la cual depende de la fidelización del cliente con el banco (scoring bancario, cuyos inputs no son conocidos), edad, antigüedad de carné del conductor y potencia del vehículo.

Con el fin de mejorar la segmentación de clientes de la compañía aseguradora se intentarán encontrar patrones para poder crear nuevos grupos de manera que, obteniendo esta información, la aseguradora pueda tomar mejores estrategias de gestión de cartera, así como llevar a cabo acciones de tarifa y suscripción.

Para alcanzar este objetivo se decidió que lo más adecuado era emplear técnicas de aprendizaje no supervisado, esto es, la utilización de técnicas de análisis clúster.

1.3 Estructura de la memoria

El Trabajo Fin de Máster se dividirá en las siguientes secciones:

- Base teórica de los métodos de clasificación: se explican y desarrollan todos los posibles tipos de clúster que pueden ser empleados para alcanzar el objetivo de segmentación.
- Estudio y depuración de datos: antes de elegir los algoritmos que se utilizarán, es necesario comprender y analizar la base de datos empleada, lo que incluye estudiar de manera exhaustiva todas las variables que vayan a ser adoptadas. Asimismo, se realizarán todos los ajustes necesarios para que los resultados del algoritmo sean lo más precisos posibles.
- Aplicación de las técnicas clúster: una vez analizados los datos, se determinará qué tipo de técnica se usará para poder conseguir nuestro propósito, además de analizar los resultados que arrojen dichas técnicas.
- Conclusiones: finalmente, se expondrán las conclusiones que se obtengan del proyecto.

2. BASE TEÓRICA DE LOS MÉTODOS DE CLASIFICACIÓN

Tal y como define Villardón, el Análisis de Clústers, también llamado Análisis de Conglomerados, es un tipo de técnica de Análisis Exploratorio de Datos que proporciona soluciones a problemas de clasificación (Villardón, 2007).

Los Análisis de Clúster pretenden ordenar objetos, ya sean personas, variables, clientes, entre otros, en grupos, de manera que todos los integrantes que conforman cada grupo tengan un grado de similitud alto en comparación con el que tienen con los miembros del resto de agrupaciones. En otras palabras, el clúster se define como la clase en la que se engloban sus miembros.

Tal y como exponen James et all. los métodos de clustering no corresponden a un único campo, sino que son utilizados para diferentes fines. De esta manera, existen numerosos enfoques de clustering, pero son dos tipos los que prevalecen sobre el resto: los agrupamientos jerárquicos y los agrupamientos no jerárquicos (James, G., Witten, D., Hastie, T., & Tibshirani, R., 2013).

Asimismo, existen otras dos categorías que, aunque menos conocidas, no menos importantes, merecen la pena destacar. Por un lado, los métodos de clustering basados en densidades, y por otro, los métodos de clustering espectral.

2.1 Métodos supervisados vs Métodos no supervisados

El aprendizaje automático, o también llamado Machine Learning, es una rama de la Inteligencia Artificial. Su peculiaridad reside en que las máquinas aprenden por sí mismas sin necesidad de ser programadas. Es decir, el aprendizaje automático es capaz de identificar patrones por sí solo y poder realizar predicciones.

Existen principalmente dos tipos de aprendizaje en el Machine Learning: aprendizaje supervisado y aprendizaje no supervisado.

El aprendizaje supervisado desarrolla un modelo matemático sobre la base de un conjunto de datos que contiene inputs y outputs (International Journal for Research in Applied Science & Engineering Technology, June 2020).

Así, el objetivo del aprendizaje supervisado es entrenar un modelo que aprenda a mapear las entradas a las salidas correctas. En la fase de entrenamiento, el modelo establece los parámetros o reglas internas para reducir la diferencia entre las salidas predichas y reales de los datos empleados en la fase de entrenamiento.

Dentro del aprendizaje supervisado se aprecian dos tipos de problemas:

- Clasificación: se trata de determinar una de las diferentes clases predefinidas a una instancia de entrada. Es decir, intentar predecir la categoría en la que va a clasificarse una observación según unas características dadas.
- Regresión: en la regresión el objetivo es predecir un valor numérico y continuo. A diferencia de la clasificación, lo que se intenta predecir no es una categoría, sino un valor numérico.

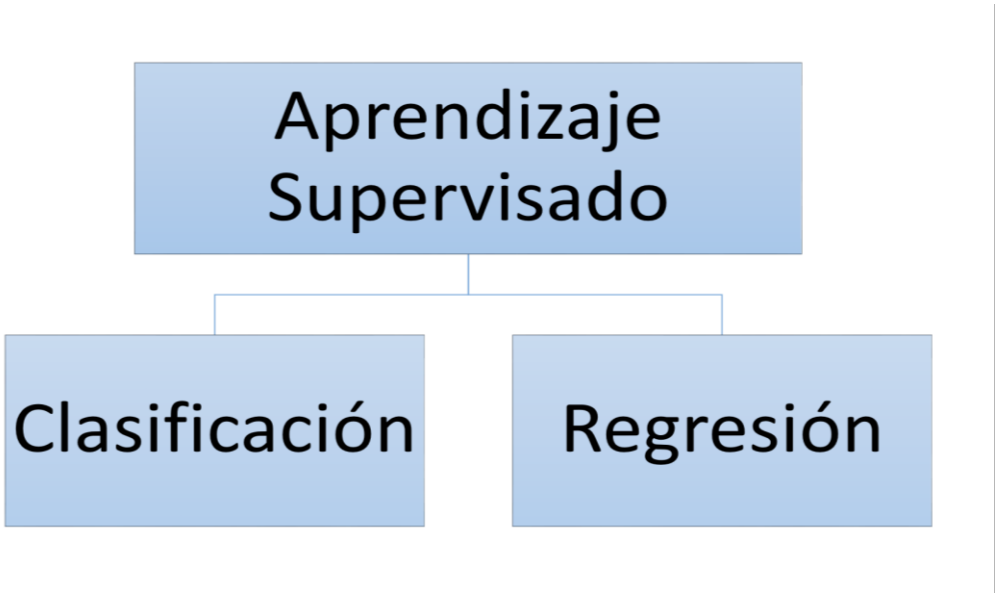


Figura 1. Aprendizaje Supervisado. Elaboración Propia.

Dentro de los algoritmos que se incluyen en el aprendizaje supervisado están: Regresión Lineal, Regresión Logística, Clasificador Bayesiano Ingenuo (Naive Bayes), Árboles de Decisión, Redes Neuronales, entre otros.

El aprendizaje no supervisado desarrolla un modelo que se construye únicamente a través de inputs (a diferencia del supervisado donde sí eran necesarios tanto inputs como outputs). Es decir, se utilizan datos sin etiquetar.

Su propósito es utilizar algoritmos de tal forma que se consigan agrupaciones de datos o patrones no conocidos sin necesitar para ello de la inteligencia humana. Dicho de otro modo, el fin último del empleo del aprendizaje no supervisado es identificar y comprender la estructura propia del conjunto de datos para obtener similitudes y diferencias en la información.

Dentro del aprendizaje no supervisado se distinguen principalmente dos tipos:

- Reducción de la dimensionalidad: es decir, técnicas estadísticas que permiten mapear una base de datos a subespacios de tamaño menor obtenidos del espacio inicial. Una de las principales técnicas de reducción de la dimensionalidad es el llamado Análisis de Componentes Principales (PCA), herramienta que, aunque sea englobada dentro del aprendizaje no supervisado, es necesaria para el procesado de datos previo a la ejecución del algoritmo del aprendizaje supervisado.
- Análisis Clúster: se denominan todos aquellos métodos que tratan de averiguar subgrupos desconocidos en una base de datos dada. Entre ellos destacan K-Means, K-Modes o DBSCAN. Este tipo de aprendizaje será el llevado a cabo en el trabajo con el fin de segmentar la cartera de autos de la entidad aseguradora.

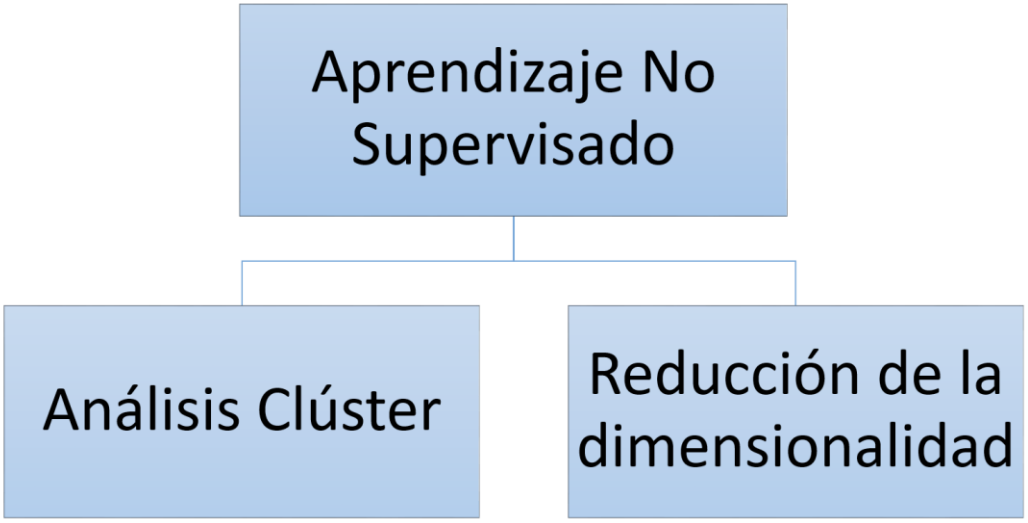


Figura 2. Aprendizaje No Supervisado. Elaboración Propia.

En resumen, la principal diferencia que existe entre el aprendizaje supervisado y el aprendizaje no supervisado es la necesidad de tener datos etiquetados en los métodos supervisados.

Además, el aprendizaje supervisado normalmente crea modelos para clasificar datos o realizar predicciones mientras que el no supervisado, en la mayoría de los casos, se emplea para entender las posibles relaciones que hay en los datos a estudiar.

Por último, destacar que las conclusiones obtenidas por lo métodos de aprendizaje supervisado son mucho más objetivas que las que pueda dar cualquier método no supervisado, pudiendo en muchos casos no llegar a ninguna conclusión con estos últimos.

A continuación, se analizarán los tipos de Análisis de Clúster más relevantes donde se detallará en qué consisten y cuál es su funcionamiento. La elección de unos u otros dependerá del tipo de dato que se emplee.

2.2 Clustering no jerárquico

Son aquellos clústeres (también conocidos como clustering particional) cuyo algoritmo se basa en agrupar los datos en un único nivel, sin existir ningún tipo de estructura jerárquica. En este tipo de datos se divide la muestra de datos en distintos grupos sin basarse en ningún nivel de jerarquía entre ellos.

En general, una de las peculiaridades que posee este tipo de clústeres es la eficiencia con la que se ejecutan sus algoritmos, pues son capaces de trabajar con gran cantidad de datos que contienen diversos campos sin necesidad de requerir mucho tiempo para la ejecución y espacio para el almacenamiento de los datos.

Los clústeres no jerárquicos, por lo general, suelen necesitar de un punto relevante al inicio antes de realizar la partición, el llamado centroide. Es a partir de este cuando, de manera iterativa, se va asignando a cada observación de la base de datos el clúster que tenga su centroide menos alejado de dicha observación, minimizando de esta manera la función objetivo.

Lo que hace que difieran unos tipos de algoritmos con otros son los pasos a posteriori que optimizan la función objetivo.

2.2.1 K-Means

Es el algoritmo no jerárquico más popular a día de hoy pues la idea que subyace de él es la más sencilla en comparación con otros tipos de algoritmos.

La finalidad del algoritmo de K-Means es agrupar un conjunto de elementos N que se encuentran contenidos en un espacio dimensional i en K clústeres (MacKay, D. J. ,2003).

El funcionamiento del algoritmo de K-Means es el siguiente:

- Elección del número de clústeres: en primer lugar, hay que elegir el número de clústeres k que se quieren crear.
- Inicialización: aleatoriamente se eligen k puntos que servirán como centroides iniciales, tantos como clústeres se quieran crear.
- Asignación de puntos: todos los puntos de la base de datos son asignados al centroide que se encuentra más cercano a ellos. Para poder llevar a cabo este paso es necesario calcular la distancia entre el punto y todos los centroides elegidos para, posteriormente, asignarlo al centroide cuya distancia calculada sea menor.
- Actualización de los centroides: tras haber sido fijados todos los puntos a uno de los centroides, hay que recalcular la posición de cada centroide. Para ello, se calcula el promedio de los valores que están integrados en cada clúster y dicho punto se convertirá en el nuevo centroide.
- Repetición de los pasos 3 y 4: se repite la asignación de puntos y la actualización de centroides hasta que se alcance un punto donde los centroides converjan y no se produzcan cambios significativos en la asignación de los puntos.
- Creación de los clústeres: finalmente, se obtienen los k clústeres distintos, estando asignada cada observación a uno de ellos.

En resumen, lo que pretende el algoritmo de K-Means es minimizar la suma de distancias al cuadrado entre cada observación y el centroide del grupo asignado. Es decir, se busca minimizar la varianza intra-clúster y maximizar la distancia entre grupos.

A pesar de las ventajas que pueda suponer este método, ya que es computacionalmente muy eficiente y capaz de trabajar con grandes conjuntos de datos, además de poderse

aplicar a datos muy diversos entre sí, pues es independiente del dominio, presenta algunos inconvenientes y limitaciones.

Por un lado, los resultados pueden variar en función de los centroides iniciales elegidos. Por otro, es necesario conocer de antemano cuál es el número de clústeres deseado, por lo que es fundamental realizar un análisis exploratorio previo de los datos. Además, es un tipo de algoritmo muy sensible a los datos atípicos, sensible a la escala en la que se encuentren los datos y asume que la forma de los clústeres es esférica.

Aunque, debido a su sencillez y eficiencia, es el método más extendido para la creación de agrupaciones de bases de datos.

2.2.2 K-Modes

El algoritmo de K-Modes es una variación del algoritmo K-Means, pero presentando como principal diferencia el tipo de datos a emplear ya que el primero permite trabajar con bases de datos de variables únicamente categóricas, mientras que el segundo sólo es posible su aplicación con variables numéricas.

El objetivo del algoritmo K-Modes es muy similar al del K-Means. Lo que se pretende es agrupar el conjunto de datos en k grupos diferentes, pero con la peculiaridad de que cada grupo está representado por el valor más repetido en cada característica categórica dentro del grupo.

El funcionamiento del algoritmo de K-Modes es el siguiente:

- Elección del número de clústeres: en primer lugar, al igual que en el algoritmo K-Means, hay que elegir el número de clústeres k que se quieren crear.
- Inicialización: se seleccionan k observaciones de manera aleatoria, tantas como clústeres se quieran crear, que servirán como representantes de cada tipo de grupo. Son los denominadas k modos iniciales.
- Asignación de puntos: cada observación del conjunto de datos es asignada al modo que se encuentre más cercano, utilizando una medida de disimilitud únicamente válida para variables categóricas. Normalmente, esta medida es la denominada Hamming que consiste en asignar valor 1 a aquellas características que difieren entre la observación y el modo.

- Actualización de los modos: cuando se ha finalizado el paso en el que todas las observaciones han sido asignadas a un modo, hay que recalcular la posición de cada modo. Para ello se selecciona el valor más repetido en cada atributo categórico entre las observaciones que han sido asignadas al modo.
- Repetición de los pasos 3 y 4: ambos procedimientos se repiten iterativamente hasta que los modos converjan y no se produzcan cambios significativos en la asignación de los puntos.
- Creación de los clústeres: una vez terminado el algoritmo, se crean los k grupos diferentes entre sí, donde cada uno de ellos se caracteriza por tener un modo determinado.

Sin embargo, el algoritmo K-Modes presenta ciertas desventajas, algunas de ellas comunes al algoritmo K-Means. La más destacada es la sensibilidad a la inicialización, pues dependiendo de cuáles sean los modos iniciales, los resultados pueden ser unos u otros. Asimismo, si los datos poseen muchas categorías puede ralentizar la ejecución del algoritmo, además de que las la distancia de Hamming no refleja el nivel de diferencias entre las categorías.

A pesar de las limitaciones anteriormente descritas, el algoritmo K-Modes presenta varias ventajas a tener en cuenta. En primer lugar, a diferencia de otros algoritmos de clustering que se basan en datos numéricos, K-Modes permite la utilización de datos categóricos sin necesidad de transformarlos.

Por otro lado, es robusto ante datos faltantes y en general, es mucho más eficiente computacionalmente en comparación con otros métodos (por ejemplo, el algoritmo K-Prototypes).

2.2.3 K-Prototypes

El algoritmo de K-Prototypes es una extensión del algoritmo de K-Means que se caracteriza por poder trabajar no sólo con datos numéricos, sino con bases de datos mixtos, esto es, datos tanto numéricos como categóricos. El algoritmo K-Prototypes hace uso no sólo la distancia euclídea utilizada en el algoritmo K-Means, sino también una medida de disimilitud para variables categóricas.

La base de este algoritmo reside en la combinación de características del algoritmo K-Means para datos numéricos con las características propias del algoritmo K-Modes para datos categóricos.

El objetivo del método K-Prototypes es hallar k centroides que minimicen la suma de las distancias intra-grupos, distancias cuyo cálculo se basa en las variables numéricas y variables categóricas. Es decir, minimiza la varianza intra-clúster para las variables numéricas y minimiza la medida de disimilitud, esta es, la distancia de Hamming, para las variables categóricas.

Por tanto, este tipo de algoritmo hace que tenga una relevancia mayor que el resto de métodos de clustering ya que en el mundo real es muy común trabajar con bases de datos que contengan tanto variables categóricas como variables numéricas (Huang, Z. ,1998).

El funcionamiento del algoritmo de K-Prototypes es el siguiente:

- Elección del número de clústeres: al igual que en los otros dos algoritmos explicados, hay que elegir el número de clústeres k que se quieren crear.
- Inicialización: se seleccionan k observaciones de manera aleatoria que serán los centroides, tantas como números de clústeres se quieran crear, y que se utilizarán como representantes de cada tipo de grupo.
- Asignación de puntos: cada observación de la base de datos con la que se trabaje se asigna al centroide seleccionado aleatoriamente más cercano empleando para ello la medida de distancia más adecuada. Como en el algoritmo K-Prototypes se tienen datos mixtos, es decir, datos numéricos y datos categóricos, para los primeros se suele utilizar la distancia euclídea, aunque existen otros tipos de distancia que se pueden emplear. Para los segundos, se suele utilizar la medida de Hamming, aunque se puede emplear otro tipo de medida de disimilitud.
- Actualización de los centroides: una vez terminado el paso anterior, se actualizan los centroides de tal manera que se vuelven a calcular los valores medios de los datos numéricos y los modos en los datos categóricos en cada clúster.

- Repetición de los pasos 3 y 4: ambos procedimientos se repiten iterativamente hasta que los centroides converjan y no se produzcan cambios significativos en la asignación de los puntos.
- Creación de los clústeres: finalmente, se crean los k grupos diferentes entre sí, donde cada uno de ellos se caracterizará por tener unas características propias.

En resumen, el algoritmo de K-Prototypes utiliza conjuntos de datos que combinan atributos numéricos con categóricos para comprender patrones y relaciones no conocidas en un primer momento. Su importancia recae en el hecho de que, en el mundo real, las bases de datos más comunes no albergan únicamente datos numéricos o categóricos, sino ambos simultáneamente.

En cuanto a las ventajas de este tipo de algoritmo, permite la identificación de patrones complejos que se encuentran en bases de datos mixtas. Es capaz de identificar relaciones y dependencias de datos de diferente naturaleza. Además, las agrupaciones serán más precisas que en otros tipos de clústeres ya que no sólo considerará un tipo de dato, sino que serán de tipo mixto.

Si se atienden a las desventajas. muchas de ellas son comunes a los otros tipos de algoritmos ya descritos. Por ejemplo, el algoritmo K-Prototypes es sensible a cuál sea la inicialización elegida, sus resultados dependerán del número de clústeres elegidos y dependiendo de cuál sea la escala de los datos impactará de manera diferente en las distancias y, eso a su vez, en la formación de agrupaciones.

Por otro lado, este tipo de método tiene mayor complejidad computacional que los algoritmos K-Means y K-Modes al tener que trabajar con datos de diferente índole lo que se traducirá en un mayor tiempo de ejecución y necesidad de ordenadores con mayor memoria.

A pesar de estas limitaciones, el algoritmo K-Prototypes permite obtener resultados de calidad al no necesitar aplicar tantas transformaciones en los conjuntos de datos con los que se va a trabajar, por lo que lo hace de él un método muy práctico en objetivos de creación de agrupaciones.

2.3 Clustering jerárquico

El objetivo de los métodos jerárquicos puede ser por un lado crear, a partir de una agrupación de clústeres, uno nuevo o dividir uno ya existente en varios. De esta manera, al realizar este proceso de aglomeración o división sucesivamente, se quiere minimizar algún tipo de distancia o maximizar algún tipo de similitud.

Dentro de los métodos jerárquicos existen dos subtipos:

- Métodos jerárquicos aglomerativos: también llamados ascendentes. El proceso se inicia con tantos grupos como observaciones se tengan. A partir de estos individuos iniciales se van creando los diferentes clústeres de manera ascendente de manera que al finalizar el proceso todas estas observaciones se agrupan en un mismo grupo.
- Métodos jerárquicos divisivos: también llamados descendentes. En este caso, el proceso se inicia con un agregado del conjunto de observaciones, y a partir de esta agrupación inicial, se van formando grupos cada vez de menor tamaño como consecuencia de un proceso de sucesivas divisiones.

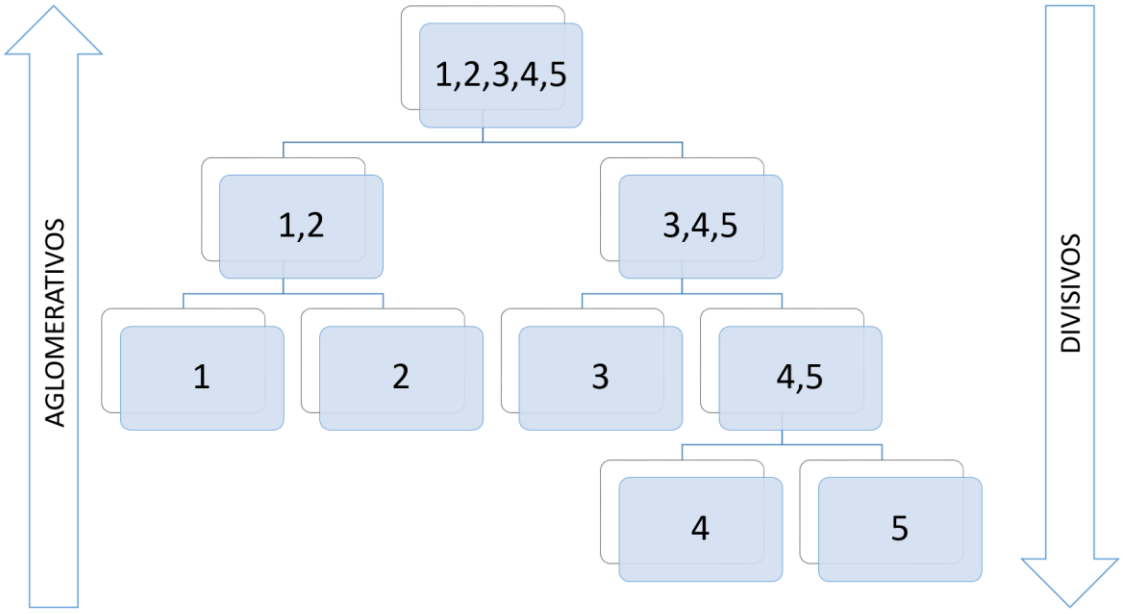


Figura 3. Métodos Jerárquicos. Elaboración Propia.

Tanto los métodos jerárquicos aglomerativos como los métodos jerárquicos divisivos, normalmente se representan con un dendrograma que muestra las diferentes asociaciones o disociaciones que se van creando. Es decir, organiza los datos en grupos o subgrupos hasta el nivel de detalle que se quiera alcanzar y permite, no sólo distinguir las relaciones de agrupación entre los datos, sino también las posibles relaciones entre grupos.

En los métodos jerárquicos aglomerativos el algoritmo que hay detrás de ellos es el siguiente:

- Inicialización y cálculo de la matriz de distancias: hay que calcular la matriz de distancias entre el conjunto de pares de las observaciones, pues se considera que cada observación del conjunto de datos es un clúster individual. Las medidas más habituales que se emplean son la distancia euclídea, la de Manhattan, la de Minkowski o la de correlación, entre otras.
- Construcción del dendrograma: el dendrograma empieza, como se ha explicado anteriormente, con cada observación como un clúster individual. A continuación, se agrupan los pares de puntos que más cercanos estén en base a la matriz de distancias.
- Actualización de la matriz de distancias: tras la agrupación de los dos clústeres del paso anterior, la matriz de distancias tiene que ser actualizada teniendo en cuenta esta fusión. Es decir, se vuelve a calcular la matriz de distancias teniendo en cuenta el nuevo clúster y los antiguos. Por tanto, la nueva matriz de distancias tendrá un clúster menos.
- Repetición de los pasos 3 y 4: se van fusionando los clústeres hasta que finalmente sólo quede un único clúster que agrupe a todas las observaciones.

Por otro lado, en los métodos jerárquicos divisivos el algoritmo que hay detrás es idéntico que en los métodos jerárquicos aglomerativos, con la única diferencia de que se ejecuta en sentido opuesto.

Es decir, se consideran todas las observaciones como un único clúster y en cada iteración se separan aquellas que no sean similares hasta llegar a tantos clústeres como observaciones haya.

En resumen, hay que tener en cuenta que antes de realizar cualquier agrupamiento jerárquico es necesario establecer la matriz de proximidad entre cada observación empleando para ello una función de distancia. Posteriormente, hay que actualizar la matriz para establecer la distancia que hay entre cada clúster.

Sin embargo, hay varias maneras de medir la vinculación que existe entre los clústeres. Es decir, hay que definir qué significa "proximidad".

Existen, por tanto, diferentes métodos de vinculación (*linkage methods*):

- Método del vecino más próximo o Enlace simple: es la distancia más pequeña que existe entre un par de puntos en dos clústeres.
- Método del vecino más lejano o Enlace completo: es la distancia más lejana que existe entre un par de puntos en dos clústeres.
- Método de agrupación de vinculación promedio o Vinculación inter-grupo: consiste en sumar la distancia de cada par de observaciones en cada clúster y dividir entre el total de pares para obtener la distancia media entre agrupamientos.
- Método de Ward o Varianza mínima: tiene en cuenta todos los clústeres. Se calcula la suma de distancias cuadradas dentro de los agrupamientos y las fusiona para reducirlas. De esta manera, se reduce la varianza de cada clúster creado.
- Método del Centroide: encuentra el centroide del clúster uno y dos y se calcula la distancia entre ellos antes de unirse.

Los métodos jerárquicos tienen, al igual que otros algoritmos, ciertas limitaciones y desventajas. Una de las más importantes es que no se puede emplear para grandes bases de datos ya que tienen un elevado costo computacional, además de que la visualización de los clústeres en los dendrogramas puede ser una tarea difícil y compleja. También, se ven afectados por la escala y distancia que se use por lo que es muy importante llevar a cabo un buen procesado previo de los datos.

Aunque los métodos jerárquicos también presentan una serie de ventajas. Una de ellas es el hecho de que a diferencia de los clústeres no jerárquicos no es necesario especificar el número de clústeres de antemano ya que, gracias al dendrograma, se

puede elegir el nivel de clústeres según las necesidades y objetivos del análisis. Además, las inicializaciones aleatorias no afectan a las conclusiones de los clústeres por lo que los resultados obtenidos son más sólidos y consistentes.

2.4 Clustering basados en densidades

Todos los métodos hasta ahora analizados se basan en que la forma que se obtiene de los clústeres es esférica. Sin embargo, puede ocurrir que las agrupaciones presenten otro tipo de formas como son formas en S u ovales.

Si los datos utilizados presentan esta naturaleza puede suceder que la segmentación sea incorrecta ya que se incluirán valores atípicos o ruido en los grupos. (Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X., 2017).

Para solventar este problema, esto es, poder encontrar clústeres que tengan formas arbitrarias, las agrupaciones se pueden crear como regiones densas en el espacio de datos y separadas por regiones dispersas.

La definición de densidad en este contexto se define como el número de observaciones dentro de un radio especificado.

El tipo de clúster más relevante dentro de los clústeres basados en densidades es el algoritmo DBSCAN donde se pueden encontrar clústeres con formas arbitrarias sin verse afectado por el ruido.

2.4.1 DBSCAN

El algoritmo DBSCAN (Density-Based Spatial Clustering of Applications with Noise) es el más usado dentro de la categoría de clustering de densidades.

Con este método se pueden encontrar distancias cercanas o valores atípicos, es decir, puntos que no se pueden clasificar en ninguno de los clústeres y que son considerados como ruido.

La idea que subyace del algoritmo DBSCAN es que, para todos los puntos de un clúster, la vecindad un determinado radio tiene que abarcar por lo menos un mínimo de puntos.

En el algoritmo DBSCAN se necesitan dos parámetros a elegir antes de la creación de los clústeres:

- Épsilon (eps): es la distancia a la que tienen que estar los puntos entre sí para ser incluidos dentro de un clúster. Es decir, un par de puntos se consideran vecinos si la distancia entre ellos es menor o igual al valor de épsilon.
- Puntos mínimos (minPts): total de punto mínimos que tiene que haber para poder formar un clúster. Si no se alcanzan este número mínimo, entonces no hay clúster por densidad.

Hay que destacar que en el algoritmo DBSCAN existen 3 tipos de puntos:

- Punto de Núcleo: un punto es considerado como núcleo cuando dentro del radio épsilon tiene al menos el número de puntos especificados como puntos mínimos. El punto de núcleo siempre se considerará como región densa.
- Punto de Borde: un punto es considerado como borde cuando el número de puntos dentro del radio épsilon es menor que el de puntos mínimos establecido, pero se encuentra en la vecindad con un punto considerado como núcleo.
- Punto de Ruido: un punto de ruido son aquellos puntos que no pueden ser clasificados como puntos de núcleo ni puntos de borde.

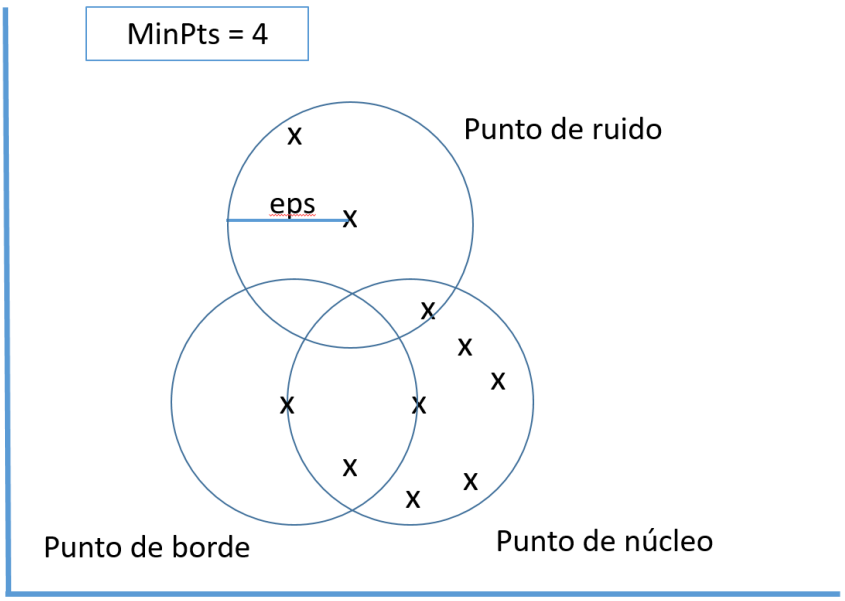


Figura 4. Tipos de puntos en algoritmo DBCSAN. Elaboración Propia.

El funcionamiento del algoritmo de DBSCAN es el siguiente:

- Elección de parámetros: antes de iniciar la ejecución del algoritmo hay que seleccionar cuáles van a ser los parámetros de las variables épsilon y los puntos mínimos.
- Inicialización: se selecciona de manera aleatoria un punto de la base de datos que no ha sido visitado. A partir de él, se verifica si dicho punto cumple con el criterio de minPts, es decir, si el punto incluye dentro de su radio a tantos puntos como puntos mínimos hayan sido establecidos. Si esto es así, dicho punto es considerado como núcleo y se inicia un nuevo clúster a partir de él. Si no lo cumple, el punto es considerado como punto de ruido.
- Expansión del clúster: a partir de este primer punto de núcleo, los puntos del vecindario de épsilon son considerados como parte del clúster si estos a su vez son puntos de núcleo. Esto se repite para el resto de puntos cercanos que son considerados como puntos de núcleo. Si alguno de los puntos de núcleo cumple con el criterio de puntos mínimos, pero uno o varios de ellos no se clasifican como puntos de núcleo, entonces estos serán considerados como ruido.
- Clasificación de los puntos de ruido: los puntos de ruido que están cercanos a un punto de núcleo son añadidos al clúster. Sin embargo, al no ser este punto un punto de núcleo, el clúster no se sigue extendiendo.
- Creación de nuevos clústeres: a partir del resto de puntos de núcleo que no han sido incluidos en el clúster creado, se crea un segundo clúster. Se repiten los procesos anteriores hasta que únicamente queden puntos de ruido sin incluir en ningún clúster.

Las desventajas que presenta este método son varias. Por un lado, hay que seleccionar los valores de los parámetros (épsilon y puntos mínimos) por lo que, si el analista no entiende a la perfección los datos y características, la configuración de los valores elegidos no será la correcta y se obtendrán resultados erróneos.

Además, si el conjunto de datos tiene puntos donde los clústeres que forman son de densidad cambiante, el algoritmo DBSCAN no puede agrupar bien los puntos ya que los parámetros seleccionados son los mismos para todos los clústeres que se crean.

En cuanto a las ventajas, no es necesario elegir el número de clústeres que se quieren formar. Además, la forma del clúster puede ser de cualquier tipo, sin tener ninguna en específico, y hace buen uso de los valores atípicos y del ruido.

Por tanto, este tipo de algoritmos es una buena alternativa cuando el conjunto de datos que se quiere estudiar presenta muchos valores atípicos o cuando las formas de los clústeres no son esféricas.

3. ESTUDIO Y DEPURACIÓN DE LA BASE DE DATOS

El aprendizaje automático se basa en el objetivo de solucionar un problema utilizando para ello la optimización de una función matemática. Para poder alcanzar esta meta, hay que trabajar con una base de datos que pueda ser usada en un contexto matemático.

Es, por tanto, un paso indispensable el conocer en profundidad el conjunto de datos y las variables con las que se van a trabajar, así como llevar a cabo diferentes técnicas de depuración y limpieza de los mismo, con el fin de mejorar la eficiencia de los algoritmos y obtener resultados más sólidos y contundentes.

De hecho, se estima que esta parte del análisis es la más importante y a la que se destina la mayor parte del tiempo (en torno a un 80% - 90% del total).

A continuación, se explicarán los procesos desde la obtención de la base de datos con la que se va a trabajar hasta la preparación de estos para, posteriormente, introducirlos en los algoritmos de los clústeres elegidos.

3.1 Descripción de la muestra

La base de datos que se empleará para llevar a cabo la segmentación ha sido facilitada por una entidad aseguradora española cuya actividad se desarrolla en el ramo de no vida y que inició su actividad comercial en el año 2021. En concreto, los datos facilitados corresponden a pólizas del seguro de auto que comercializa la compañía a fecha 31 de marzo.

El número de pólizas en cartera es de aproximadamente 40.000 pólizas con un total de 2 años de antigüedad. Sin embargo, el número de observaciones con las que se cuentan para realizar el clúster es mayor ya que los datos se han agrupado por número de póliza, pero también por movimiento y año. Es decir, una misma póliza puede tener varios registros si ha habido un cambio de año o movimiento.

Por tanto, la base de datos estará compuesta por 128.144 observaciones y 46 variables que están formadas tanto por variables de tipo numérico, como categórico y de fecha:

1. POLICY_HEADER_CODE: Número de póliza.

2. POLICY_CODE: Número de movimiento de póliza.
3. ANIO: Año contable
4. NP_CARTERA: Estado de la póliza dividiéndose en nueva producción o cartera.
5. TARIFA_PLANA: Contratación de tarifa plana (pago mensualizado sin recargo).
6. MODALIDAD: Modalidad de cada póliza. Las diferentes modalidades se dividen en TE/TL/TA/TRF200/TR300/TR400/TRSF.
7. MODALIDAD_AGRUPADA: Modalidades de cada póliza. Se divide en Terceros/Todo Riesgo.
8. DTO_CAMPAÑA: Descuento aplicado a cada póliza. Se divide en Sin descuento/Campaña 5%/Campaña 10%/Campaña 15%.
9. ASISTENCIA_PREMIUM: Contratación de la opcional Asistencia Premium.
10. PROTECCION_LEGAL_PREMIUM: Contratación de la opcional Defensa Jurídica Premium.
11. AV_MECANICA: Contratación de la opcional Avería Mecánica.
12. VALORACION_+: Contratación de la opcional Valor Nuevo +.
13. INDEMNIZACION_+: Contratación de la opcional Indemnización +.
14. PDC: Contratación de la opcional Protección del Conductor +.
15. NUM_OPC_CONTRATADAS: Número total de opcionales contratadas.
16. AÑOS_CIA_ANT: Total de años que un asegurado ha permanecido en la compañía anterior.
17. AÑOS_ASEGURADO: Número de años que el cliente ha tenido un vehículo asegurado a lo largo de su vida (ya sea el vehículo actual u otro).
18. COND_OC_27: Número de conductores ocasionales con menos de 27 años de antigüedad de carnet de conducir.
19. COND_OC_5: Número de conductores ocasionales con menos de 5 años de antigüedad de carnet de conducir.
20. COND_OC: Número de conductores ocasionales.

21. STDAD_DP: Siniestralidad de Daños Propios ocurrida en la compañía anterior declarada por el cliente.
22. STDAD_RC: Siniestralidad de Responsabilidad Civil ocurrida en la compañía anterior declarada por el cliente.
23. CIA_ANT: Compañía anterior del asegurado.
24. SCORING_SUSCRIP: Puntuación dada al cliente en función de determinadas variables (peor cuanto mayor sea la puntuación).
25. EDAD: Edad del asegurado.
26. ANT_CARNET: Años de antigüedad del carnet del asegurado.
27. PROVINCIA: Provincia de circulación declarada por el cliente.
28. COMBUSTIBLE: Tipo de motor del vehículo.
29. FEC_MATRICULACION: Fecha de matriculación del vehículo.
30. ANT_VEHI: Antigüedad del vehículo.
31. VALOR_VEHI: Valor del vehículo.
32. GARAJE: Tenencia de garaje. Se divide en individual/colectivo (vigilancia/sin vigilancia) o sin garaje.
33. KMS_ANUALES: Kilómetros anuales realizados por el vehículo.
34. PUERTAS: Número de puertas que contiene el vehículo.
35. PLAZAS: Número de plazas que contiene el vehículo.
36. POTENCIA: Potencia del vehículo.
37. USO: Uso que es dado al vehículo. Se divide en particular/profesional.
38. SCORING_ASEGURADOR: Clasificación del cliente según datos bancarios. Se divide en A/B/C/D/E/F/G/Z, siendo esto de mayor a peor y la Z cuando no permite la cesión de sus datos bancarios.
39. SCORING_ASEGURADOR_TRAM: Clasificación del cliente según datos bancarios dividiéndose en favorable/desfavorable.

40. SCORING_ASEGURADOR_TRAM_II: Clasificación del cliente según datos bancarios por tramo. Se divide en AB/C/DEFG/Z.
41. SCORING_ASEGURADOR_TRAM_III: Clasificación del cliente según datos bancarios según ABC/Desfavorable.
42. TARGET_PROPUESTO: Clasificación del cliente en Target o No Target.
43. SEGMENTO_NUEVO_ORDENADO: Segmento al que pertenece el cliente.
44. TERRITORIAL: Ubicación del lugar desde donde se realiza la contratación.
45. TERRITORIAL_TRAMO: contiene la misma información que la anterior, ubicación donde se realiza la contratación, pero creando un tramo. Se divide en CORUÑA/OURENSE/VIGO/LUGO/PONTEVEDRA/SANTIAGO/RESTO.

3.2 Análisis Exploratorio de Datos (EDA)

En primer lugar, hay que cargar la base de datos al software que se va a utilizar para llevar a cabo todos los pasos desde la depuración hasta la segmentación. En este caso, se ha elegido para ello el programa RStudio.

Una vez introducidos los datos en RStudio se procede a establecer el tipo de cada variable, distinguiendo variables de tipo carácter, numéricas, factores y fecha.

A continuación, se realiza un análisis exploratorio de datos con el fin de examinar e indagar en el conjunto de datos y resumir sus principales características, empleando gráficos para ello.

En primer lugar, se estudian las variables categóricas. Para ello se ha recurrido a la función “CrossTable” de la librería “gmodels” y al análisis gráfico “pie” para su visualización.

En segundo lugar, se estudian el resto de variables, es decir, las variables numéricas y de fecha.

Para ello, con la función “summary” y la función “st” del paquete “vtable” se ha hecho un análisis de las medias, desviaciones estándar, valores máximos y mínimos, además de comprobar los valores nulos o missing.

Posteriormente, se ha realizo un análisis visual a través de gráficos de histogramas y diagramas de caja y bigotes con las funciones “hist” y “boxplot”, respectivamente.

Si se atiende a las variables categóricas:

- NP_CARTERA: está compuesta por dos categorías, que son Cartera y Nueva Producción. La primera categoría representa el 68,3% mientras que la segunda es el 31,7%.

Cartera	Nueva Producción
87514	40629
0.683	0.317

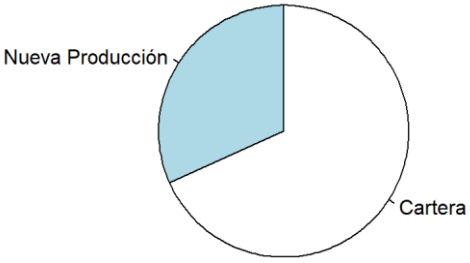


Figura 5. NP_CARTERA. Elaboración Propia.

- TARIFA_PLANA: está compuesta por dos categorías, 1 y 0, la primera indicando que se aplica y la segunda que no. La categoría del 1 representa el 71,3% y la del 0 el 28,7%.

0	1
36765	91378
0.287	0.713

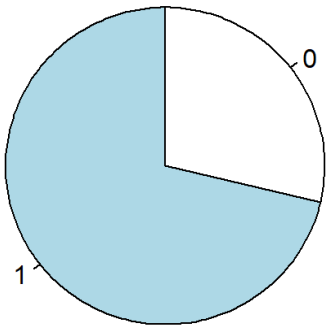


Figura 6. TARIFA_PLANA. Elaboración Propia.

- MODALIDAD: está compuesta por siete categorías que son TE, TA, TL, TRF400, TRF300, TRF200 Y TRSF. La categoría de TE representa únicamente

el 0,3%, TL el 14,1%, TA el 50,9%, TRF400 el 2,1%, TRF300 el 12,2%, TRF200 el 17% y TRSF el 3,5%.

01. TE	02. TL	03. TA	04. TRF400	05. TRF300
401	18017	65183	2735	15634
0.003	0.141	0.509	0.021	0.122

06. TRF200	07. TRSF
21725	4448
0.170	0.035

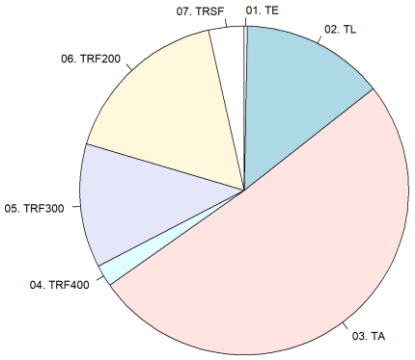


Figura 7. MODALIDAD. Elaboración Propia.

- MODALIDAD_AGRUPADA: está compuesta por dos categorías, Terceros y Todo Riesgo. La categoría de Terceros representa el 65,2% y la de Todo Riesgo el 34,8%.

01. TERCEROS	02. TODO RIESGO
83601	44542
0.652	0.348

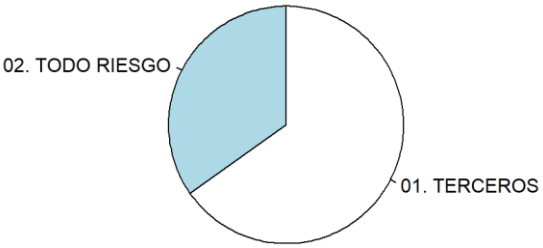


Figura 8. MODALIDAD_AGRUPADA. Elaboración Propia.

- ASISTENCIA_PREMIUM, PDC, AV_MECÁNICA, VALORACION_+, INDEMNIZACION_+, PROTECCION_LEGAL_PREMIUM: todas ellas están compuestas por dos categorías, 0 y 1. En ASISTENCIA_PREMIUM el 0 representa un 82,8% y el 1 un 17,2%, en PDC un 99% y un 1%, en AV_MECÁNICA prácticamente hay un 100% en la categoría 0 (únicamente 22 valores en la categoría 1), en VALORACIÓN_+ hay un 99,8% y un 0,2%, en INDEMNIZACIÓN_+ hay un 85,5% y un 14,5% y en PROTECCIÓN_LEGAL_PREMIUM hay un 98,8% y un 1,2%.

0	1
106146	21997
0.828	0.172

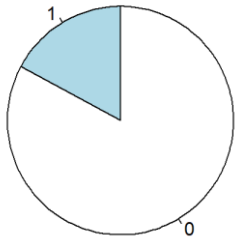


Figura 9. ASISTENCIA_PREMIUM. Elaboración Propia.

0	1
126870	1273
0.990	0.010

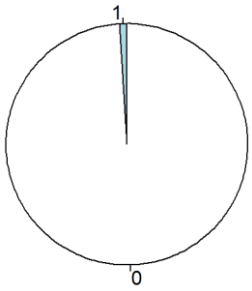


Figura 10. PDC. Elaboración Propia.

0	1
128121	22
1.000	0.000

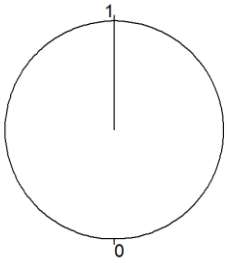


Figura 11. AV_MECÁNICA. Elaboración Propia.

0	1
127829	314
0.998	0.002

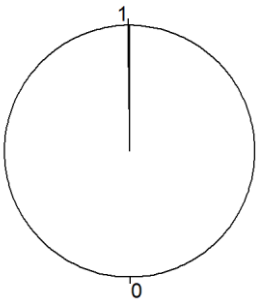


Figura 12. VALORACIÓN_+. Elaboración Propia.

0	1
109619	18524
0.855	0.145

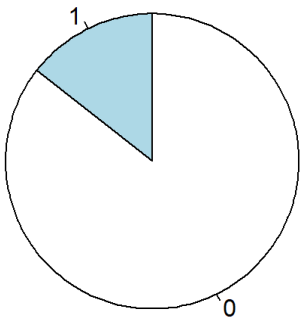


Figura 13. INDEMNIZACIÓN_+. Elaboración Propia.

0	1
126621	1522
0.988	0.012

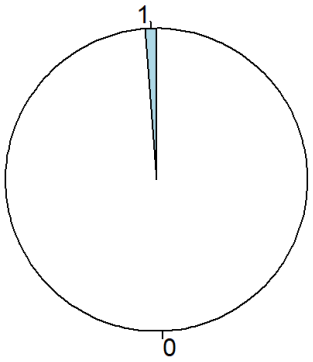


Figura 14. PROTECCION_LEGAL_PREMIUM. Elaboración Propia.

- CIA_ANT: está compuesta por 10 categorías que corresponden a diferentes compañías aseguradoras. La que mayor porcentaje tiene es Otra Compañía con un 41,7%, seguida de AXA y Mapfre con un 14,9% y un 14,5% respectivamente.

ALLIANZ	AXA	GENERALI	LIBERTY	MAPFRE
13645	19129	4643	6062	18593
0.106	0.149	0.036	0.047	0.145

MUTUA MADRILEÑA	OTRA COMPAÑIA	REALE	SEGURCAIXA	VERTI
3576	53394	8140	496	465
0.028	0.417	0.064	0.004	0.004

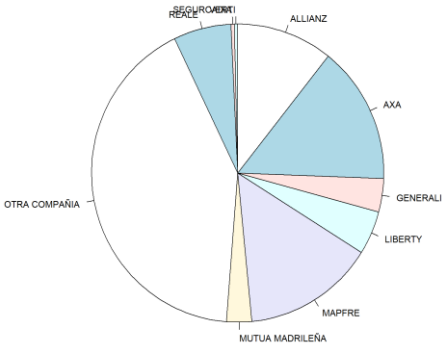


Figura 15. CIA_ANT. Elaboración Propia.

- PROVINCIA: está compuesta por todas las provincias de España. La que mayor porcentaje representa del total es La Coruña con un 35,6% y la que menos en Melilla con 3 observaciones.
- COMBUSTIBLE: está compuesta por 10 categorías, pero las que prevalecen sobre el resto son G (Gasolina) y D (Diésel) que conjuntamente representan un 98% (68,3% es Diésel y 29,7% es Gasolina).

B	D	E	G	L
24	87542	216	38102	284
0.000	0.683	0.002	0.297	0.002

P	R	X	Y	Z
159	4	1807	1	4
0.001	0.000	0.014	0.000	0.000

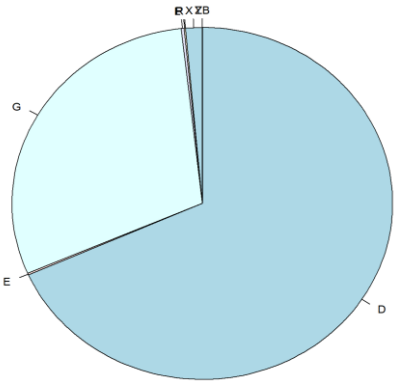


Figura 16. COMBUSTIBLE. Elaboración Propia.

- GARAJE: está compuesta por 4 categorías. Así, Garaje colectivo con vigilancia representa el 3,5%, Garaje colectivo sin vigilancia el 30,9%, Garaje individual el 53,4% y Sin garaje el 12,3%.

Garaje colectivo con vigilancia	Garaje colectivo sin vigilancia	Garaje individual	Sin garaje
4466	39537	68434	15706
0.035	0.309	0.534	0.123

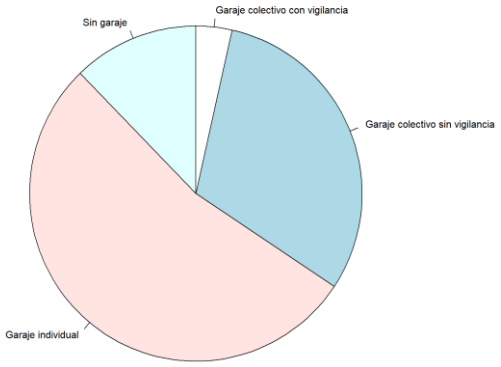


Figura 17. GARAJE. Elaboración Propia.

- KMS_ANUALES: en total hay seis categorías de tramos de kilómetros. La categoría que mayor porcentaje agrupa es la de Hasta 15k que representa casi el 50% del total, en concreto un 49,5%. La que menos es la de Más de 40k que representa un 0,3%.

Hasta 15k	Hasta 20k	Hasta 30k	Hasta 40k	Hasta 5k
63438	18313	4334	604	41106
0.495	0.143	0.034	0.005	0.321

Más de 40k
348
0.003

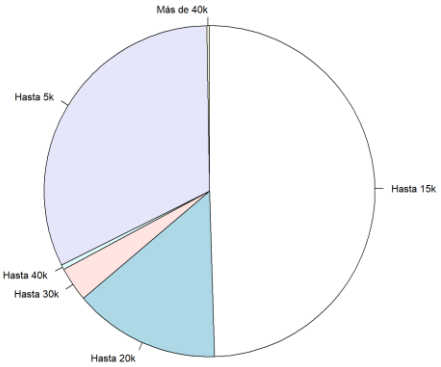


Figura 18. KMS_ANUALES. Elaboración Propia.

- USO: en total hay cuatros categorías, de las cuales Particular diario representa el 65,1%, Particular fin semana el 4,4%, Particular ocasional el 30,3% y Profesional el 0,3%.

Particular diario	Particular fin de semana	Particular ocasional	Profesional
83380	5596	38777	390
0.651	0.044	0.303	0.003

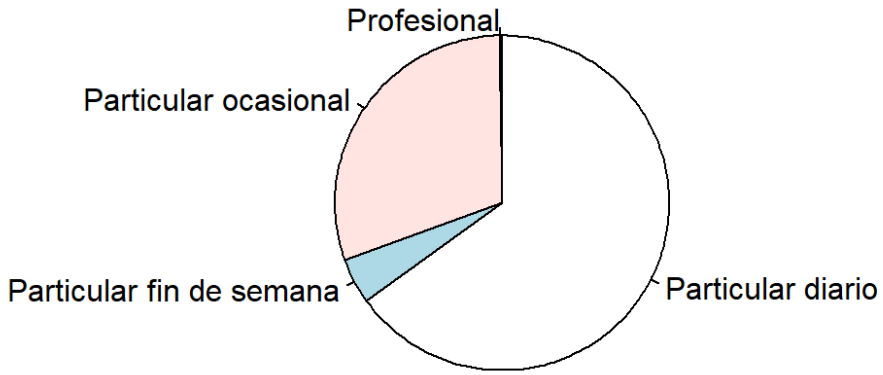


Figura 19. USO. Elaboración Propia.

- SCORING_ASEGURADOR_TRAM, SCORING_ASEGURADOR_TRAM_II, SCORING_ASEGURADOR_TRAM_III, SCORING_ASEGURADOR: el scoring asegurador comprende en total ocho categorías, aunque luego se han creado nuevas variables para tramear dichos grupos. Lo que se puede apreciar es que existen datos faltantes que necesitarán ser tratados más adelante.

1.Favorable - ABC	2.No favorable - DEFGH	3.Sin cesión - Z	4.Sin dato
99572	12339	16106	126
0.777	0.096	0.126	0.001

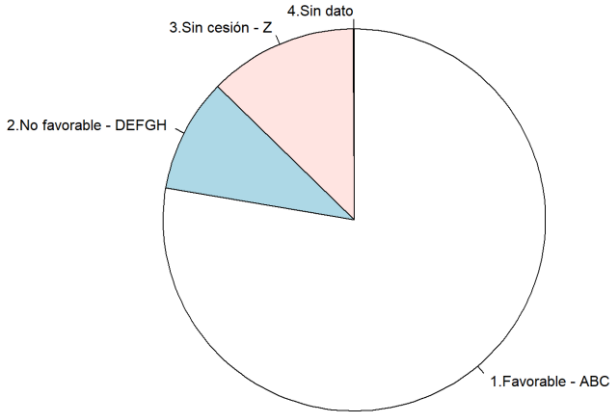


Figura 20. SCORING_ASEGURADOR_TRAM. Elaboración Propia.

AB	C	DEFGH	NA	Z
75290	24282	12339	126	16106
0.588	0.189	0.096	0.001	0.126

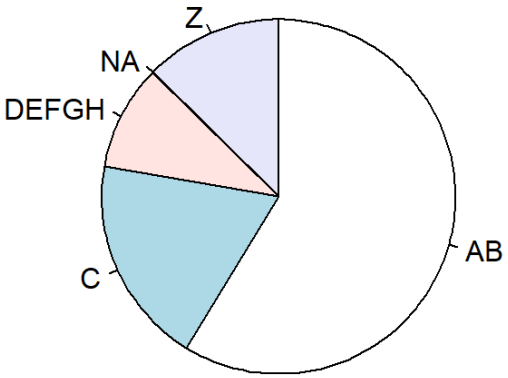
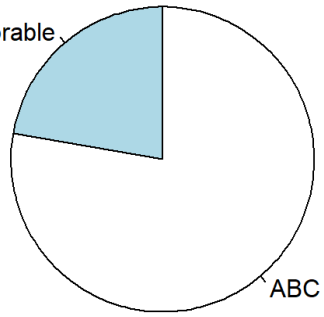


Figura 21. SCORING_ASEGURADOR_TRAM._II. Elaboración Propia.

ABC	Desfavorable
99572	28571
0.777	0.223



A	B	C	D	E	F	G	Z	NA's
48330	26960	24282	4919	1199	5237	984	16106	126

Figura 22. SCORING_ASEGURADOR_TRAM._III. Elaboración Propia.

- TARGET_PROPUESTO: está compuesta por dos categorías, que sea target o que no lo sea. La categoría del target representa un 64,6% del total mientras que el no target un 35,4%.

NO TARGET	TARGET
45306	82837
0.354	0.646

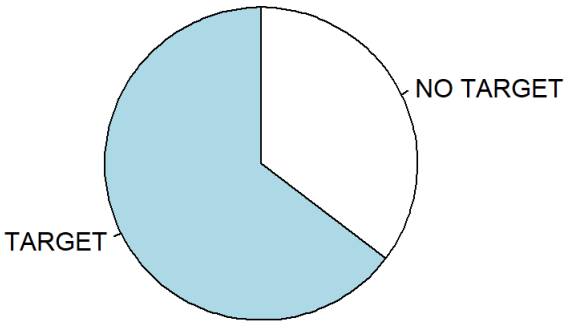


Figura 23. TARGET_PROPUESTO. Elaboración Propia.

- SEGMENTO_NUEVO_ORDENADO: en total hay cuatro tipos de segmentos donde se puede clasificar el cliente. La categoría del Segmento 1 representa el 62,2%, la del Segmento 2 el 15,5%, la del Segmento 3 el 9,6% y la del segmento 4 el 12,7%.

01. SEGMENTO 1	02. SEGMENTO 2	03. SEGMENTO 3	04. SEGMENTO 4
79690	19882	12339	16232
0.622	0.155	0.096	0.127

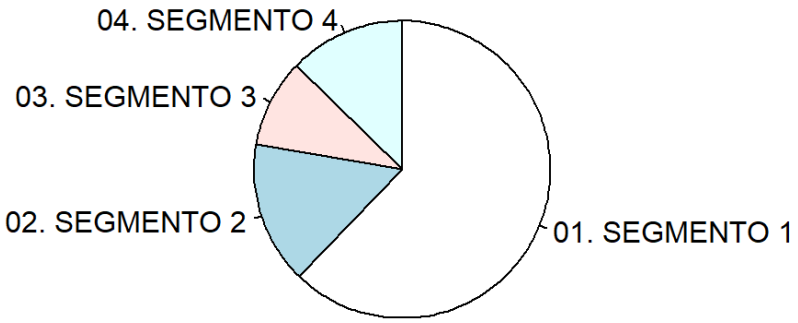


Figura 24. SEGMENTO_NUEVO_ORDEADO. Elaboración Propia.

- TERRITORIAL, TERRITORIAL_TRAMO: ambas variables ofrecen la misma información con la única diferencia de que una la da agrupada y la otra no. En la variable TERRITORIAL_TRAMO existen siete categorías de las cuales la que mayor porcentaje representa del total es A Coruña con un 23%. Por el contrario, Lugo es la menor con un 8,1%.

CORUÑA	LUGO	OURENSE	PONTEVEDRA	RESTO
29423	10377	14797	11867	23443
0.230	0.081	0.115	0.093	0.183

SANTIAGO	VIGO
14833	23403
0.116	0.183

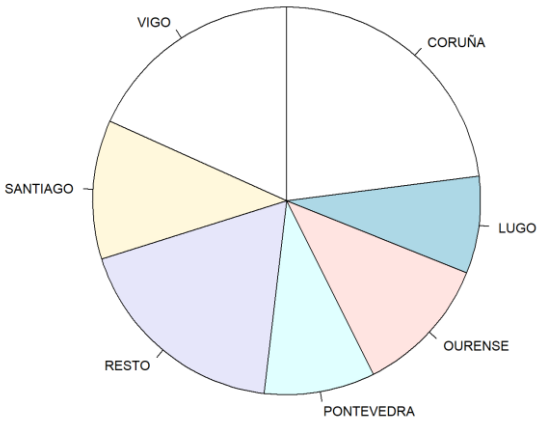


Figura 25. TERRITORIAL_TRAMO. Elaboración Propia.

Si se atiende a las variables numéricas:

- ANIO: la media se encuentra en el 2022, con una desviación típica muy baja de 0,69 siendo el mínimo y el máximo 2021 y 2023, respectivamente. En esta variable se observa que no hay valores atípicos ni extremos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	2022	0.69	2021	2022	2023	2023

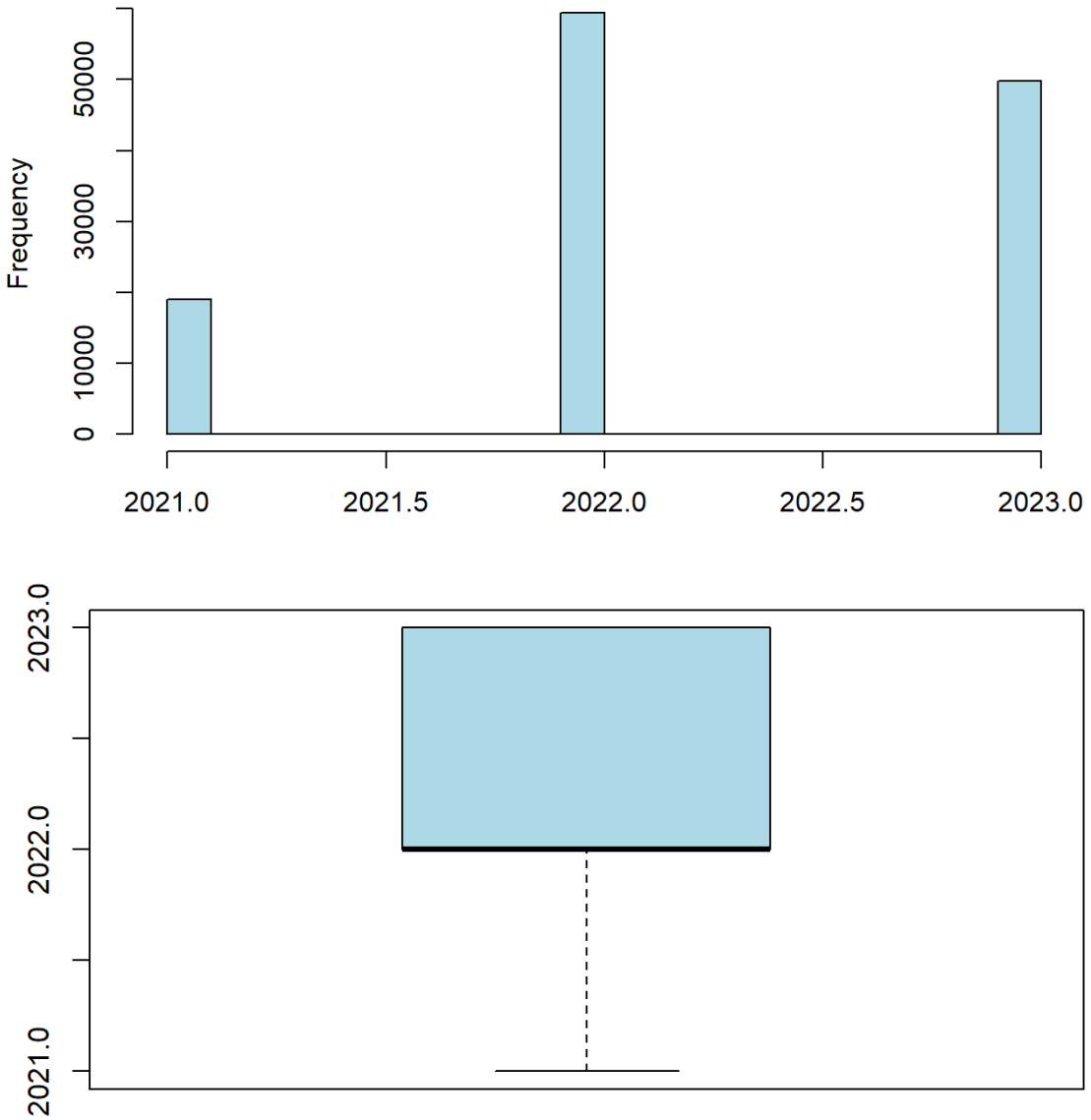
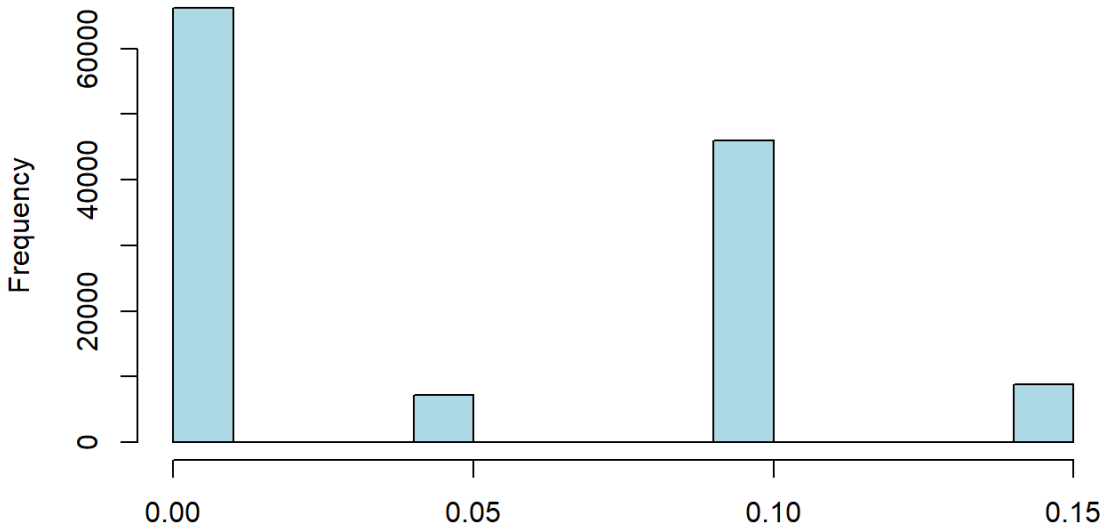


Figura 26. ANIO. Elaboración Propia.

- DTO_CAMPAÑA: la media se encuentra en el 0,049 con una desviación típica muy baja de 0,054. El valor mínimo es 0, es decir, sin aplicar campaña, mientras que el máximo es de 0,15. No existen valores atípicos ni extremos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	0.049	0.054	0	0	0.1	0.15



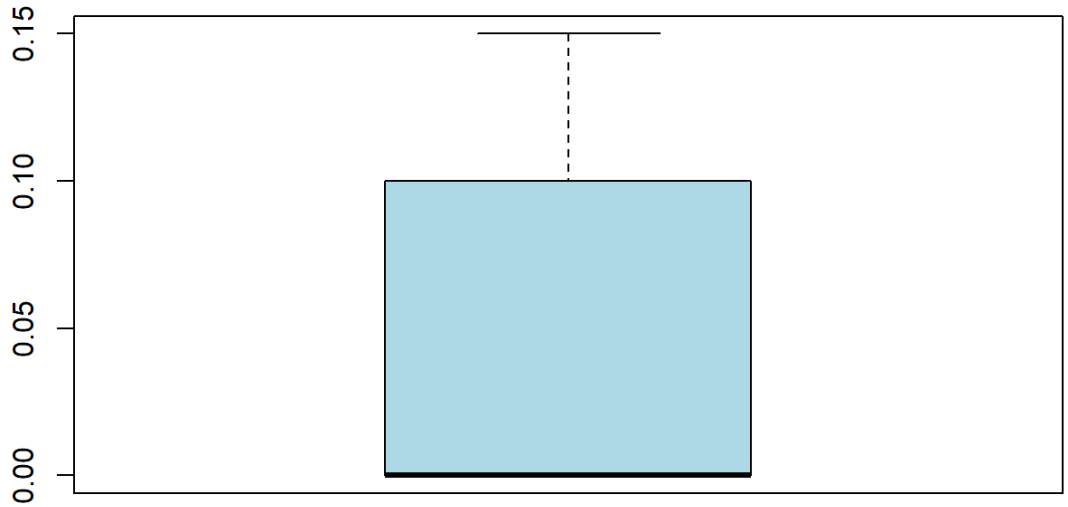


Figura 27.DTO_CAMPAÑA. Elaboración Propia.

-

- NUM_OPC_CONTRATADAS: el valor medio es de 0,33 con una desviación del 0,65. El valor mínimo es 0 y el máximo 4. En este caso. La mayor parte de los casos están concentrados en el 0 y 1 (76,6% - 14,2%). Es por ello por lo que la masa que hay del 2,3 y 4 es muy pequeña.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	0.33	0.65	0	0	0	4

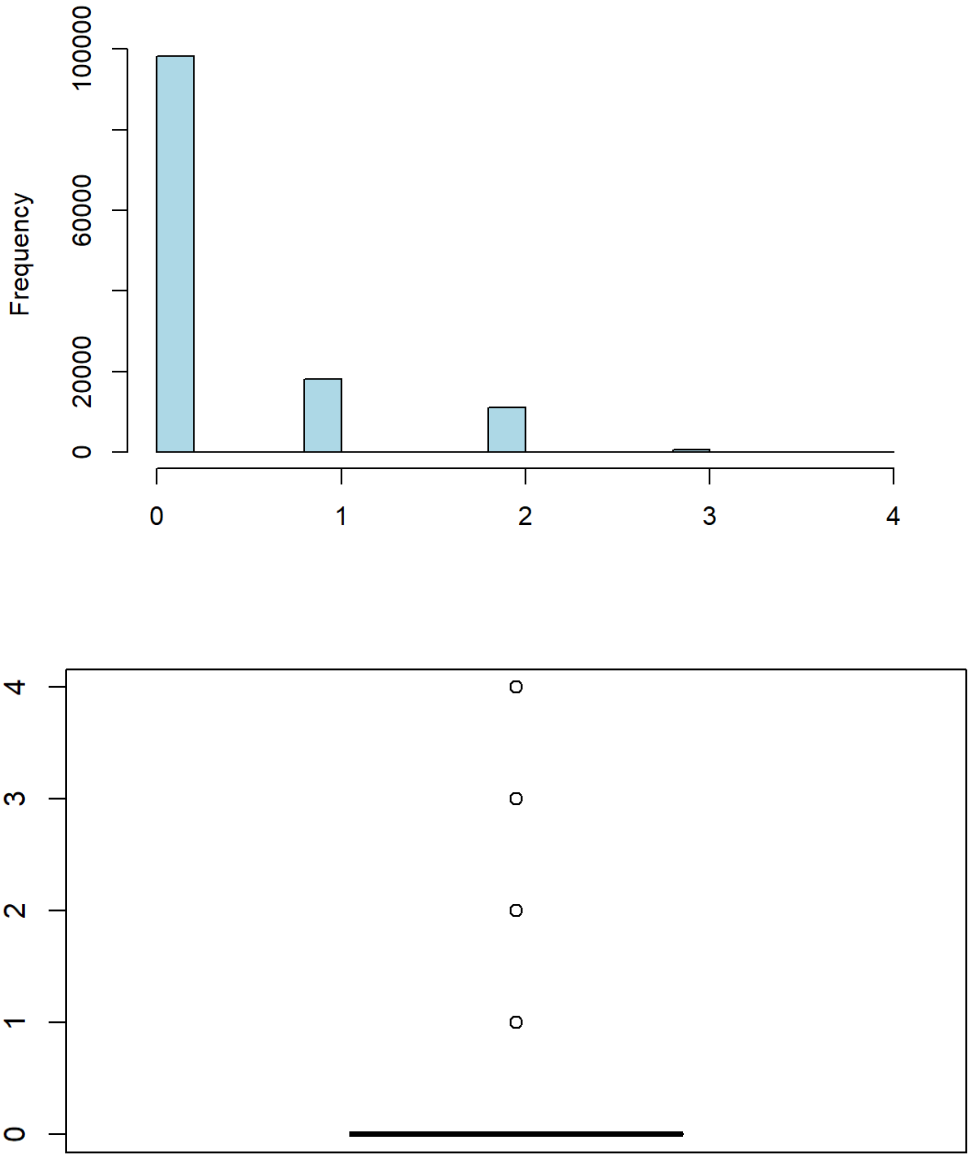


Figura 28.NUM_OPC_CONTRATADAS. Elaboración Propia.

- AÑOS_CIA_ANT: el valor medio es 4 con una desviación del 2,2. El valor mínimo es de 0 mientras que el máximo es de 6. En este caso, no se aprecian ni valores atípicos ni extremos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	4	2.2	0	2	6	6

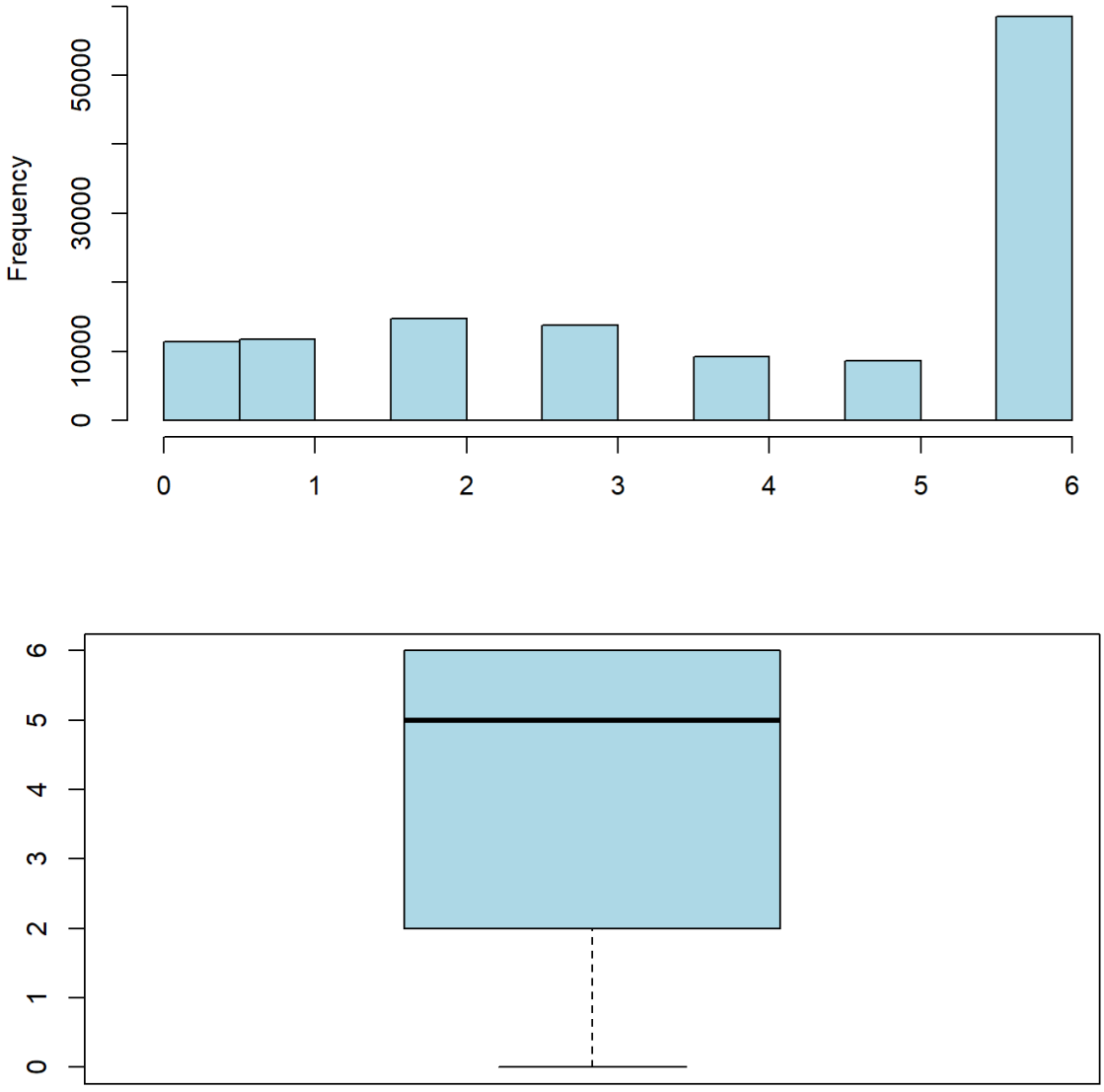


Figura 29. AÑOS_CIA_ANT. Elaboración Propia.

- AÑOS_ASEGURADO: la media se sitúa en un 5,2 con una desviación típica de 1,8. El valor mínimo son 0 años y el máximo de 6, siendo este donde más concentración de datos hay, un 80,5% de los datos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	5.2	1.8	0	6	6	6

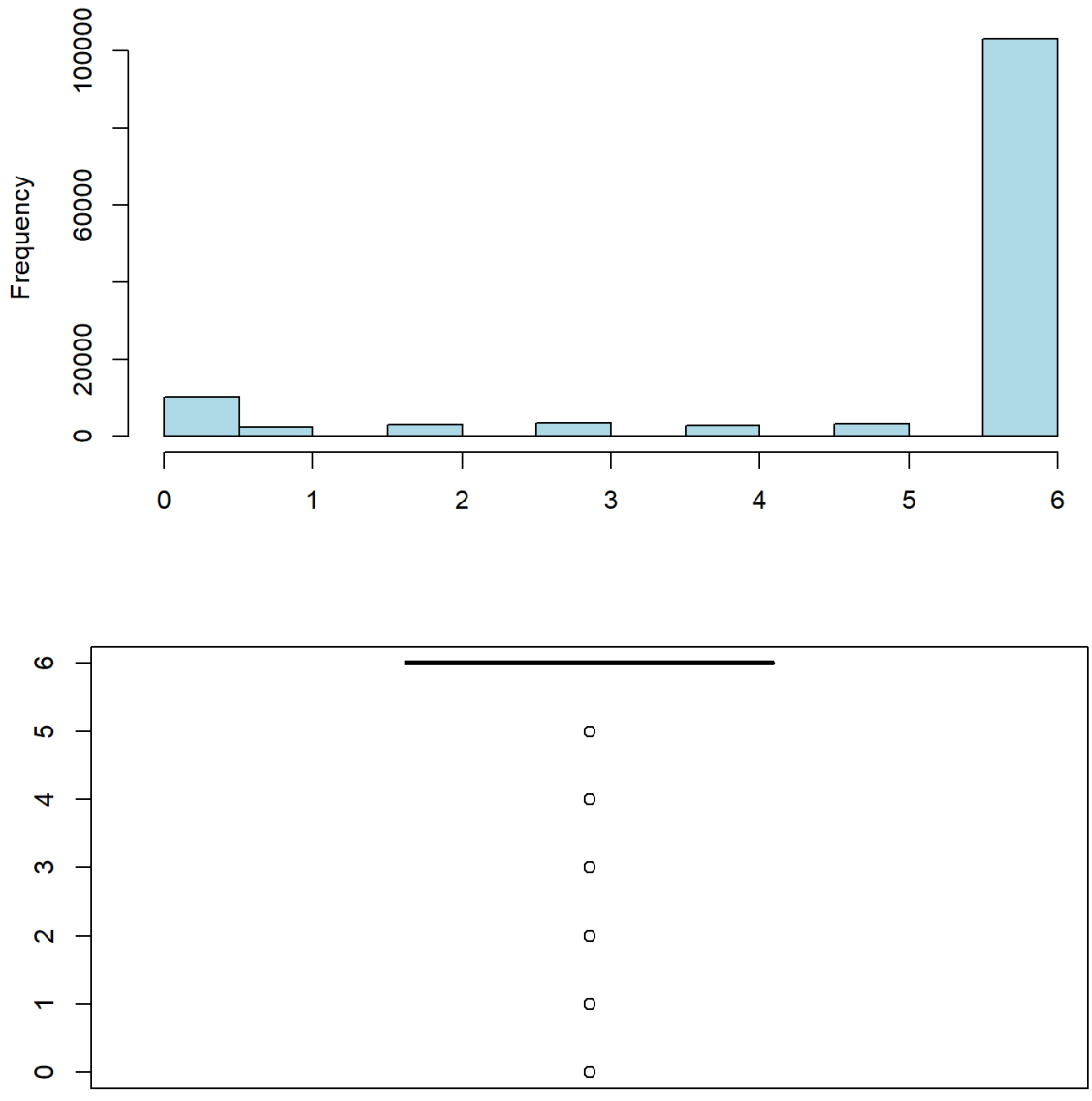


Figura 30. AÑOS_ASEGURADO. Elaboración Propia.

- COND_OC_27: la media se encuentra en 0,033 y la desviación típica es 0,19 con un valor mínimo de 0 y el máximo 4. Hay valores atípicos ya que el 96,8% se concentra en el valor 0.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	0.033	0.19	0	0	0	4

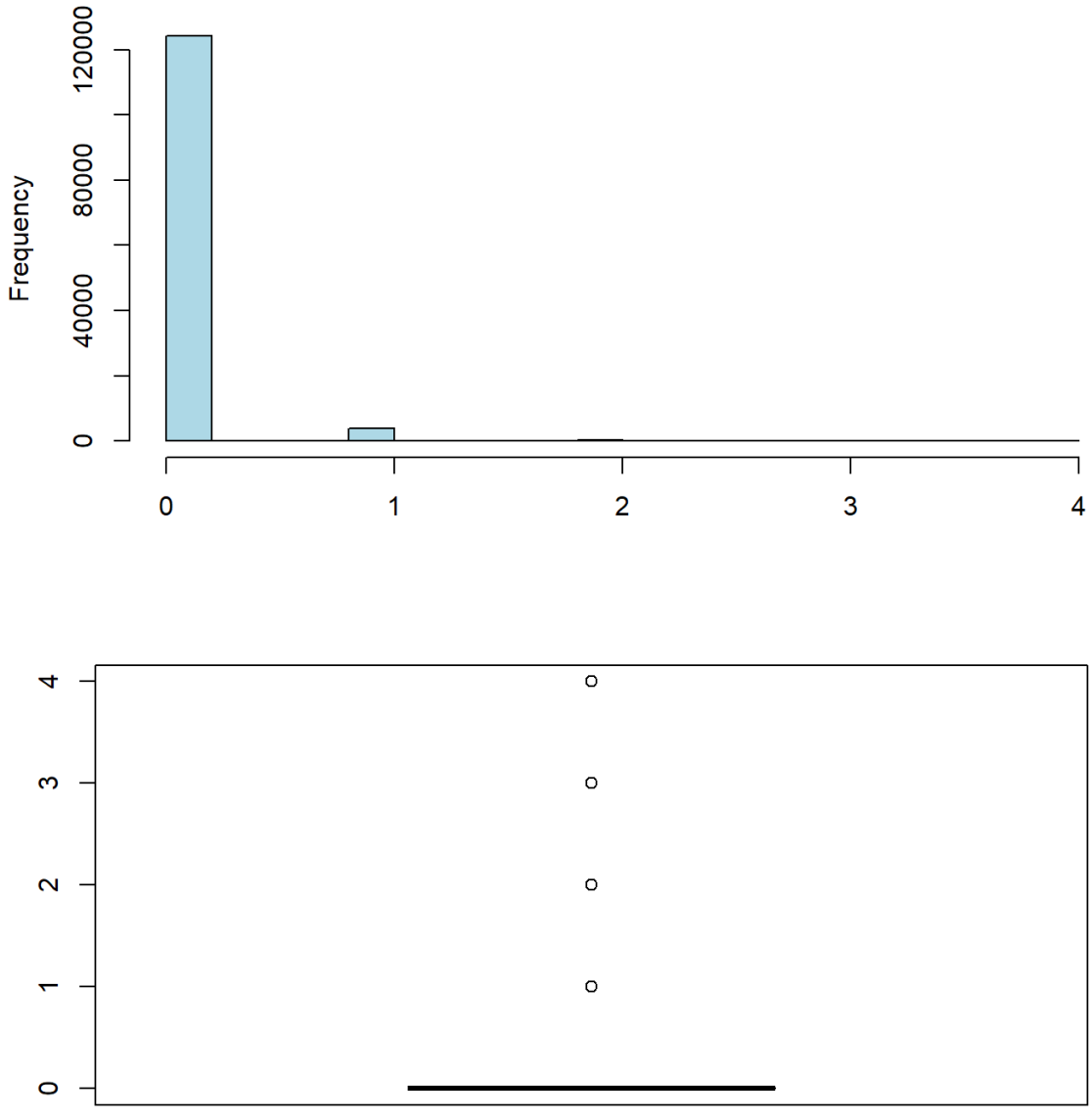


Figura 31. COND_OC_27. Elaboración Propia.

- COND_OC_5: el valor medio es de 0,022 con una desviación típica de 0,15. El valor mínimo es de 0 y el máximo de 2. Al igual que en el caso anterior, existen valores atípicos como consecuencia de que la mayor masa de datos se concentra en el 0 con un 97,8%.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	0.022	0.15	0	0	0	2

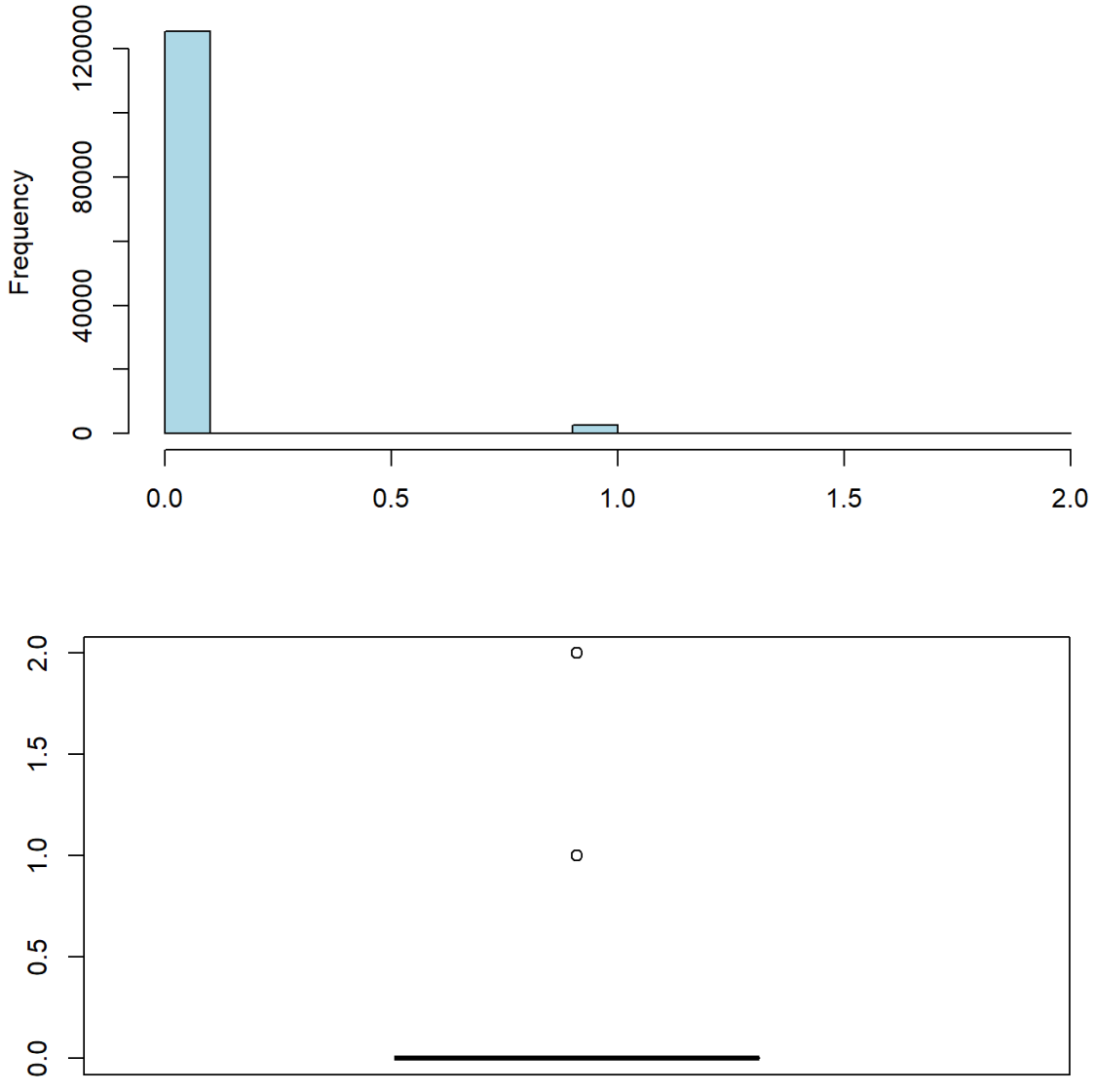
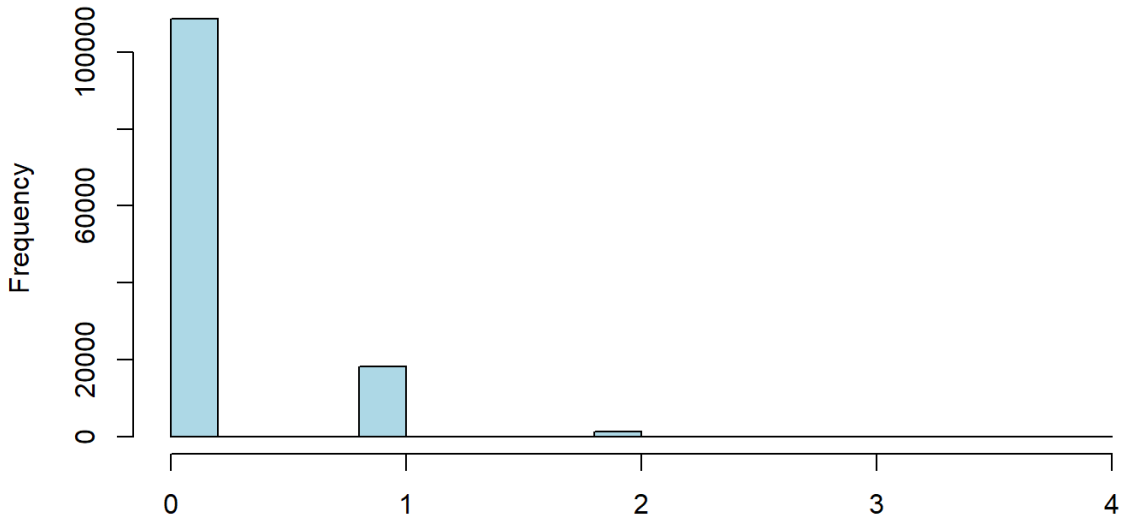


Figura 32. COND_OC_5. Elaboración Propia.

- COND_OC: la media es de un 0,17 con una desviación típica de 0,41 y valor mínimo y máximo de 0 y 4. En este caso existen atípicos ya que los valores se concentran en torno al 0 con un porcentaje de 84,7%

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	0.17	0.41	0	0	0	4



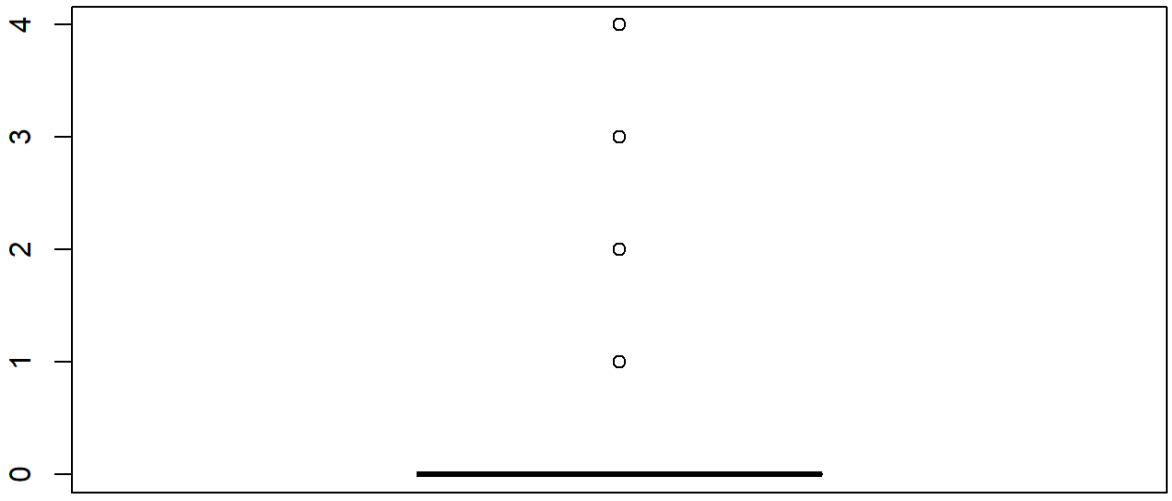


Figura 33. COND_OC. Elaboración Propia.

- STDAD_DP: el valor medio es de 0,032 con una desviación típica de 0,18. Los valores máximos y mínimos son de 0 y 4. Existen también valores atípicos porque un 96,8% de los datos son 0.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	0.032	0.18	0	0	0	4

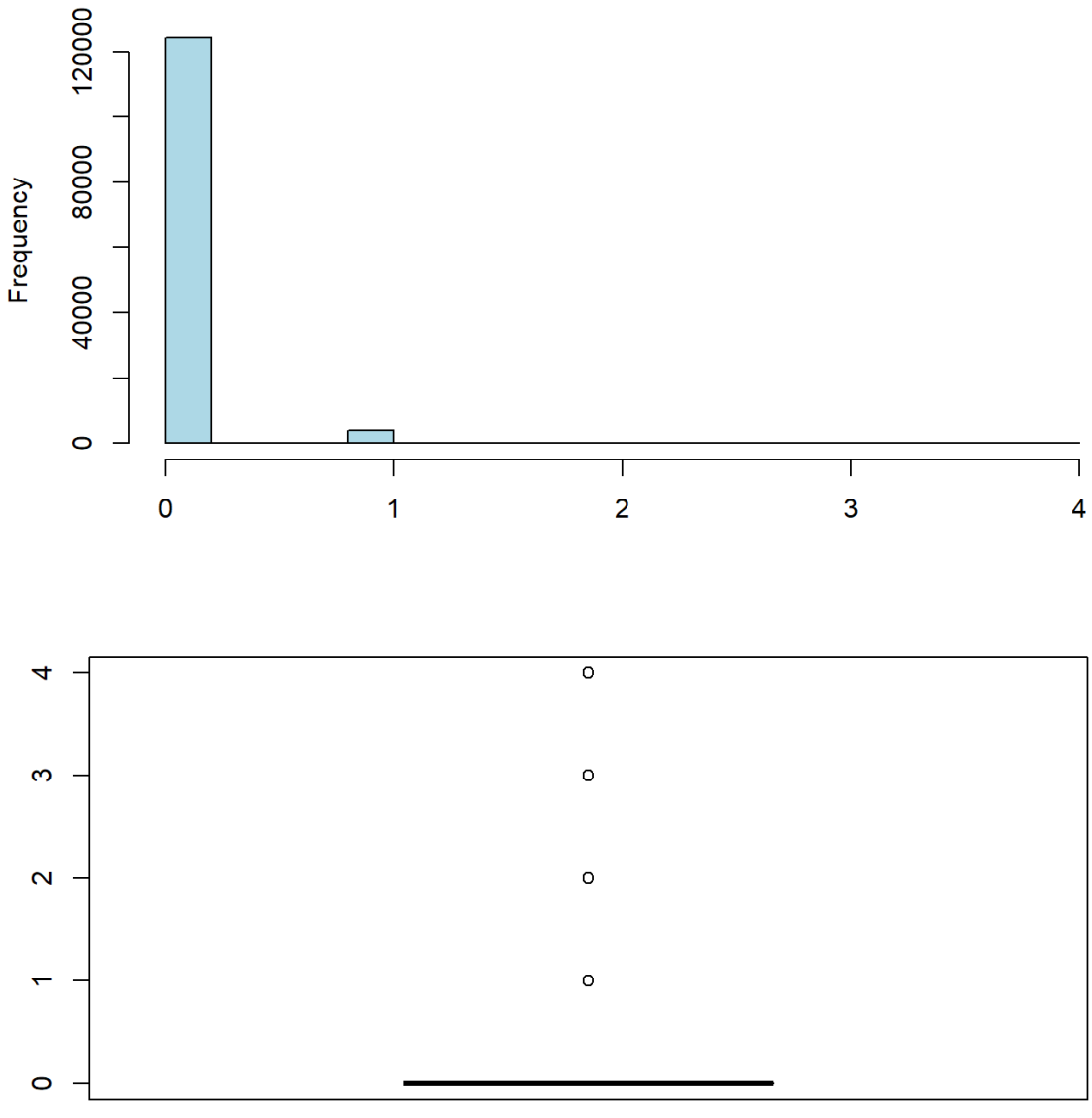
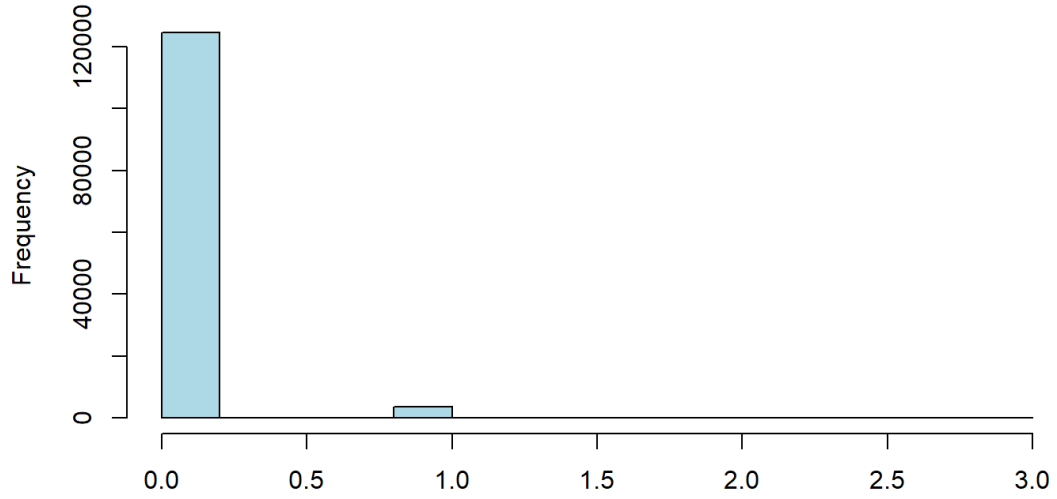


Figura 34. STDAD_DP. Elaboración Propia.

- STDAD_RC: la media se sitúa en 0,028 con una desviación típica de 0,17. El valor mínimo es de 0 y el máximo es de 3. Hay valores atípicos ya que la mayoría de valores se encuentran en torno al 0 y 1 con 97,2% y 2,8%.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	0.028	0.17	0	0	0	3



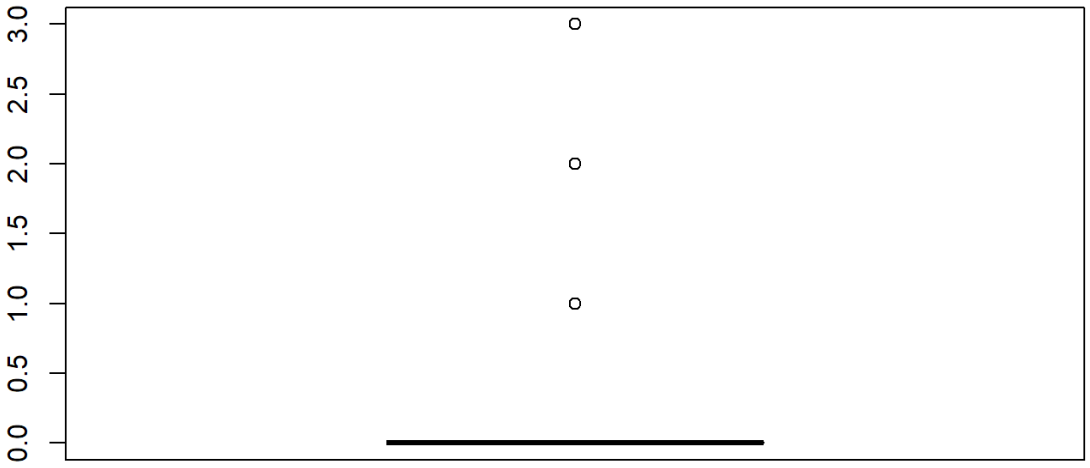
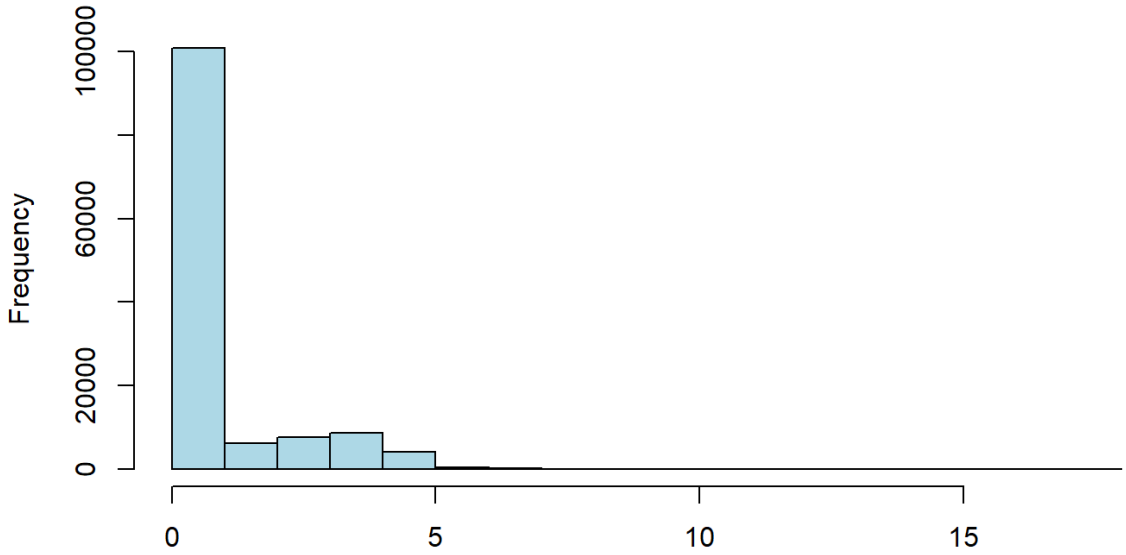


Figura 35. STDAD_RC. Elaboración Propia.

- SCORING_SUSCRIP: el valor medio es de 1 con una desviación típica de 1,5. El mínimo es de 0 y el máximo de 18. En esta variable se observan atípicos con diferentes valores.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	1	1.5	0	0	1	18



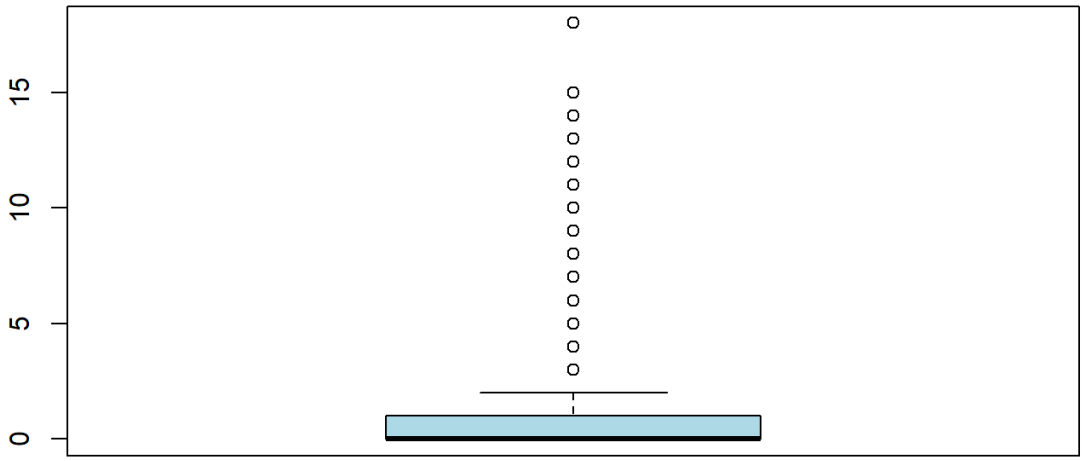


Figura 36. SCORING_SUSCRIP. Elaboración Propia.

- EDAD: la media de edad es de 55 años y una desviación típica de 14, con un mínimo de 20 años y un máximo de 97 años. No se detectan la presencia de valores atípicos ni extremos en la base de datos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	55	14	20	44	66	97

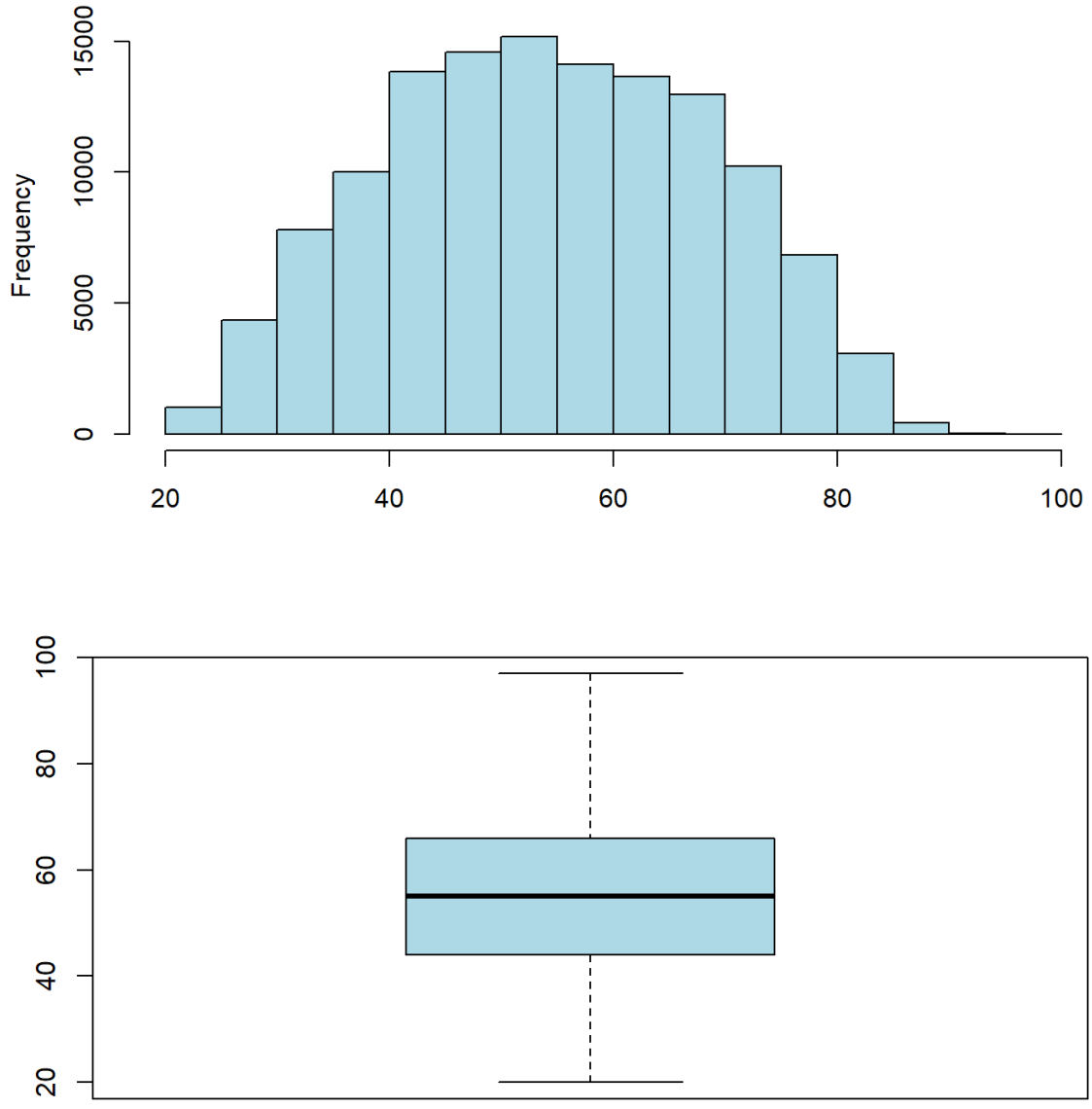


Figura 37. EDAD. Elaboración Propia.

- ANT_CARNET: la media de esta variable es de 30 años y una desviación típica de 14, con un mínimo de 1 y un máximo de 74 años. No se detectan la presencia de valores atípicos ni extremos en la base de datos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	30	14	1	19	42	74

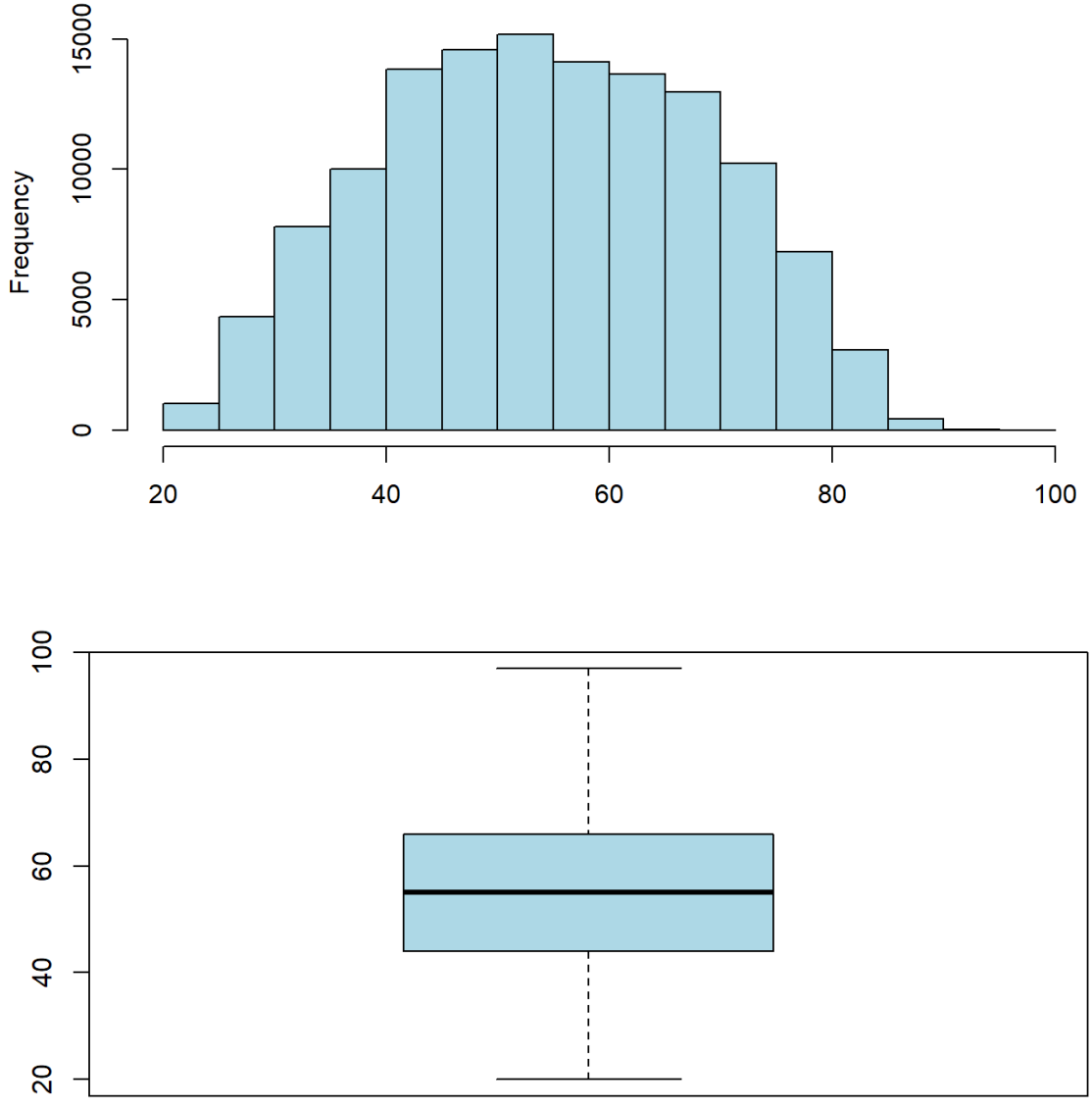


Figura 38. ANT_CARNET. Elaboración Propia.

- ANT_VEHI: el valor medio es 13 con una desviación típica de 7. Los valores mínimos y máximos son 0 y 121. Además, se presencian datos atípicos y un valor extremo.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	13	7	0	7	18	121

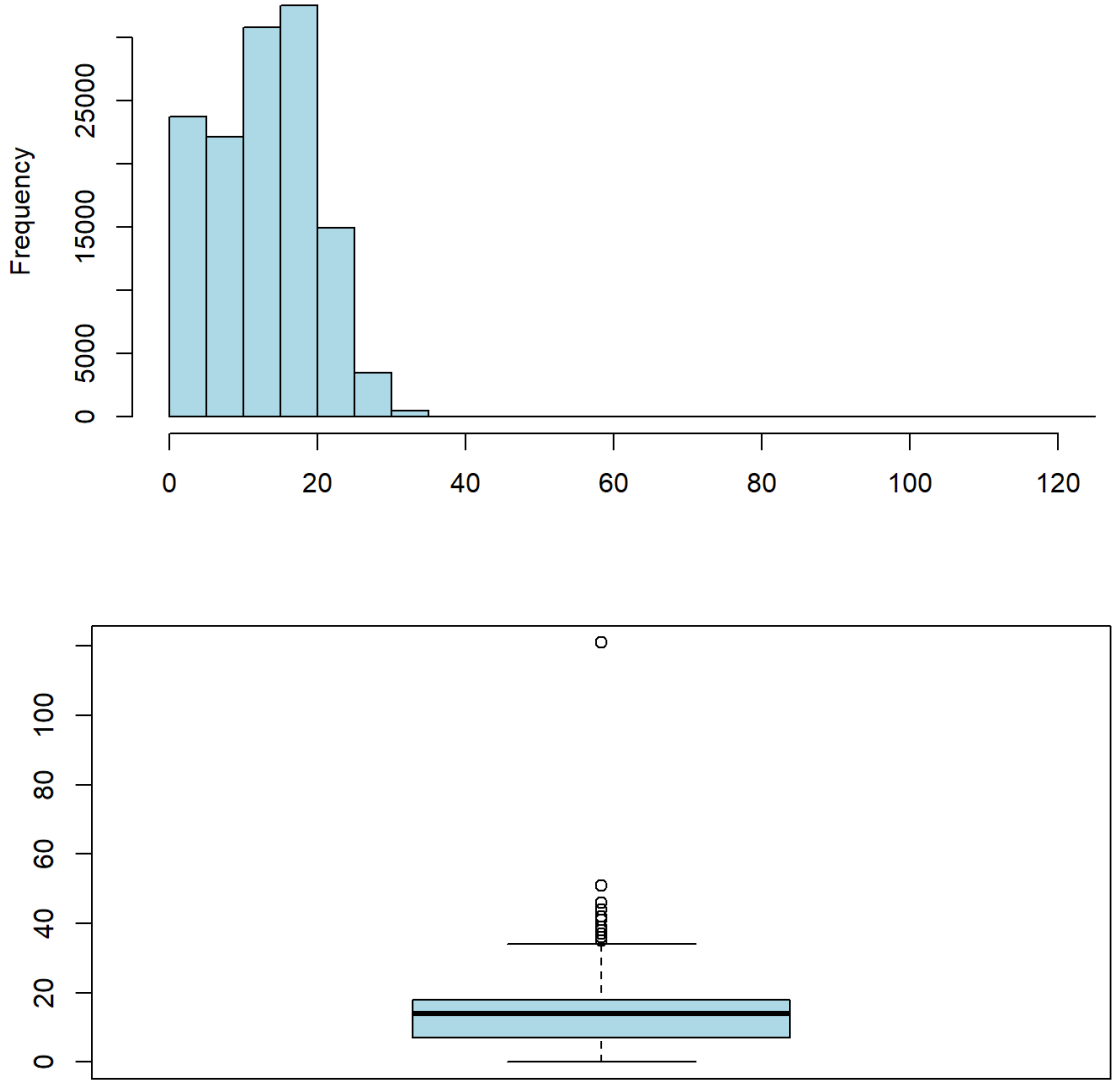
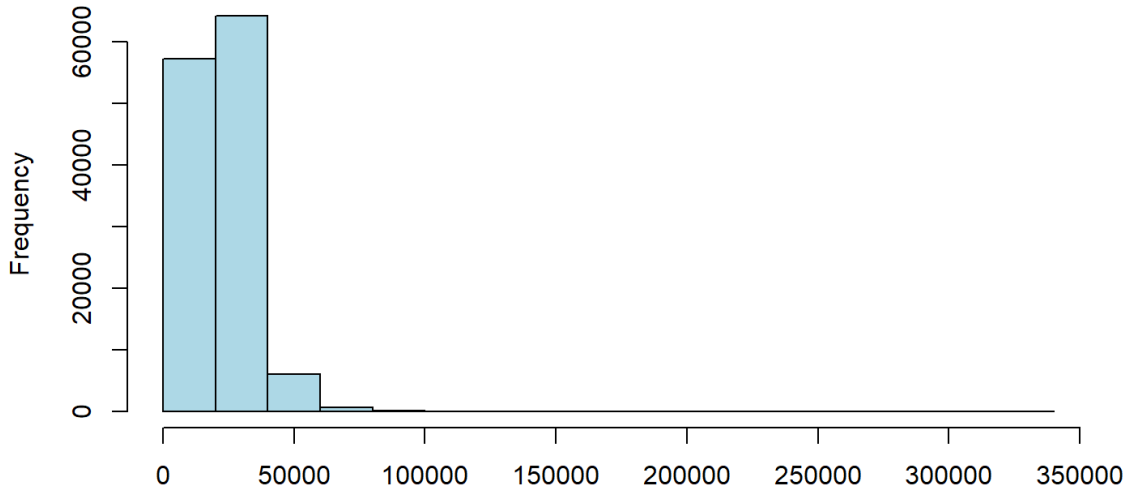


Figura 39. ANT_VEHI. Elaboración Propia.

- VALOR_VEHI: el valor medio es 22.934 con una desviación típica de 9.937. Los valores mínimos y máximos son 1.803 y 324.991. Además, se presencian datos atípicos y un valor extremo.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	22934	9937	1803	16060	27597	324991



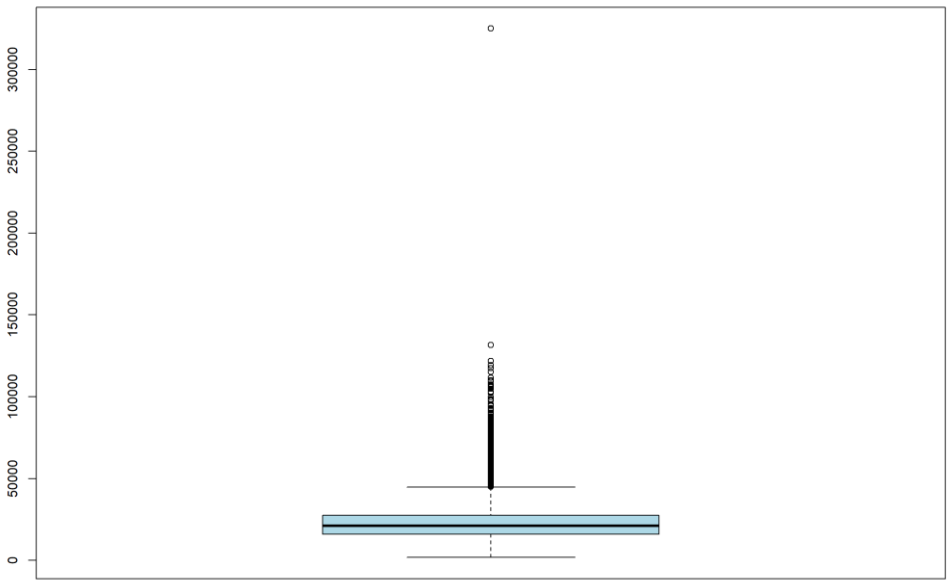
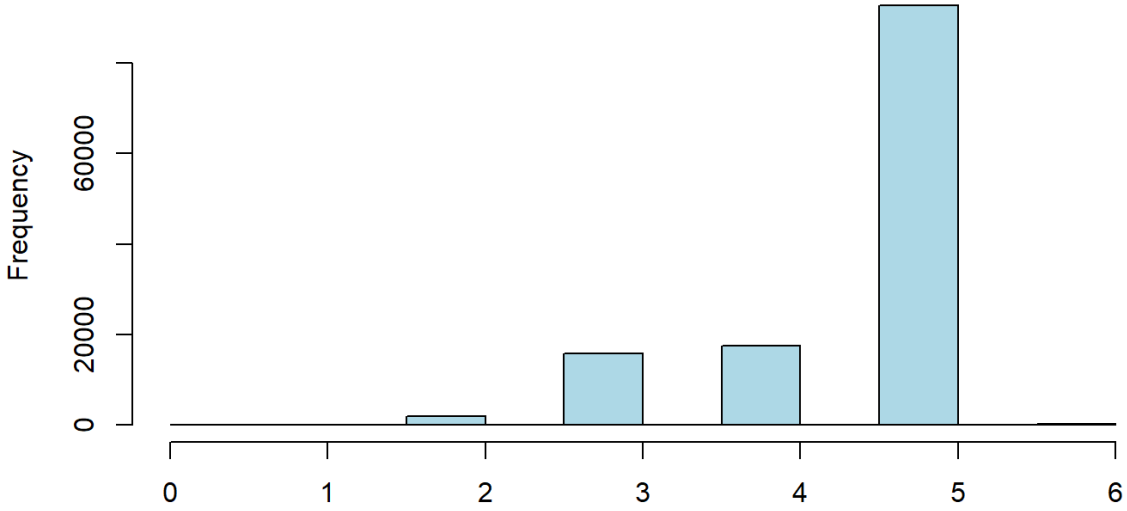


Figura 40. VALOR_VEHI. Elaboración Propia.

- PUERTAS: la media se sitúa en 4,6 con una desviación típica de 0,76 y cuyos valores mínimo y máximo son 0 y 6. Hay valores atípicos ya que la mayor parte de las observaciones se concentran en los valores 3, 4 y 5 (12,3% - 13,7% - 72,3%).

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	4.6	0.76	0	4	5	6



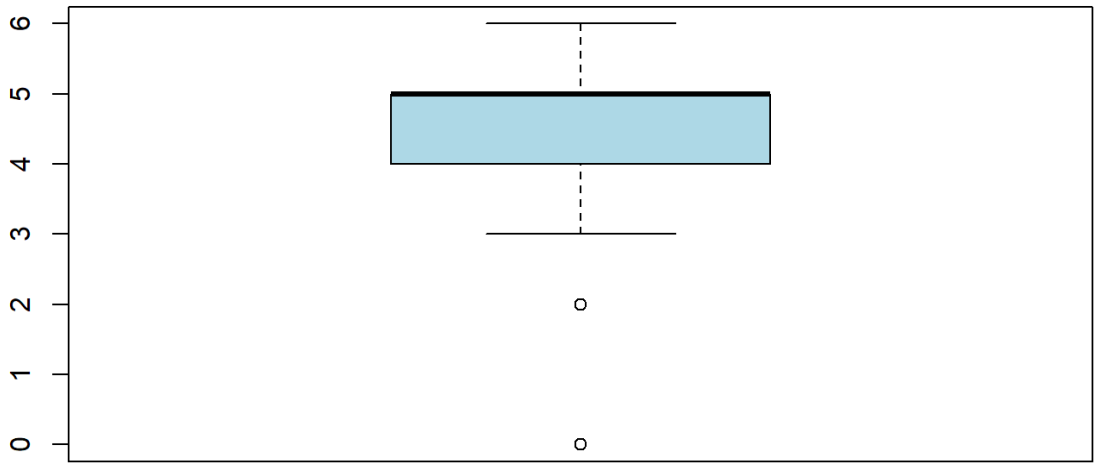


Figura 41. PUERTAS. Elaboración Propia.

- PLAZAS: la media de plazas es de 5 con una desviación del 0,65. El valor mínimo es de 2 y el máximo es de 9. Hay valores atípicos pues la mayor parte de los datos se concentran en 5, en concreto, un 89,1%.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	5	0.65	2	5	5	9

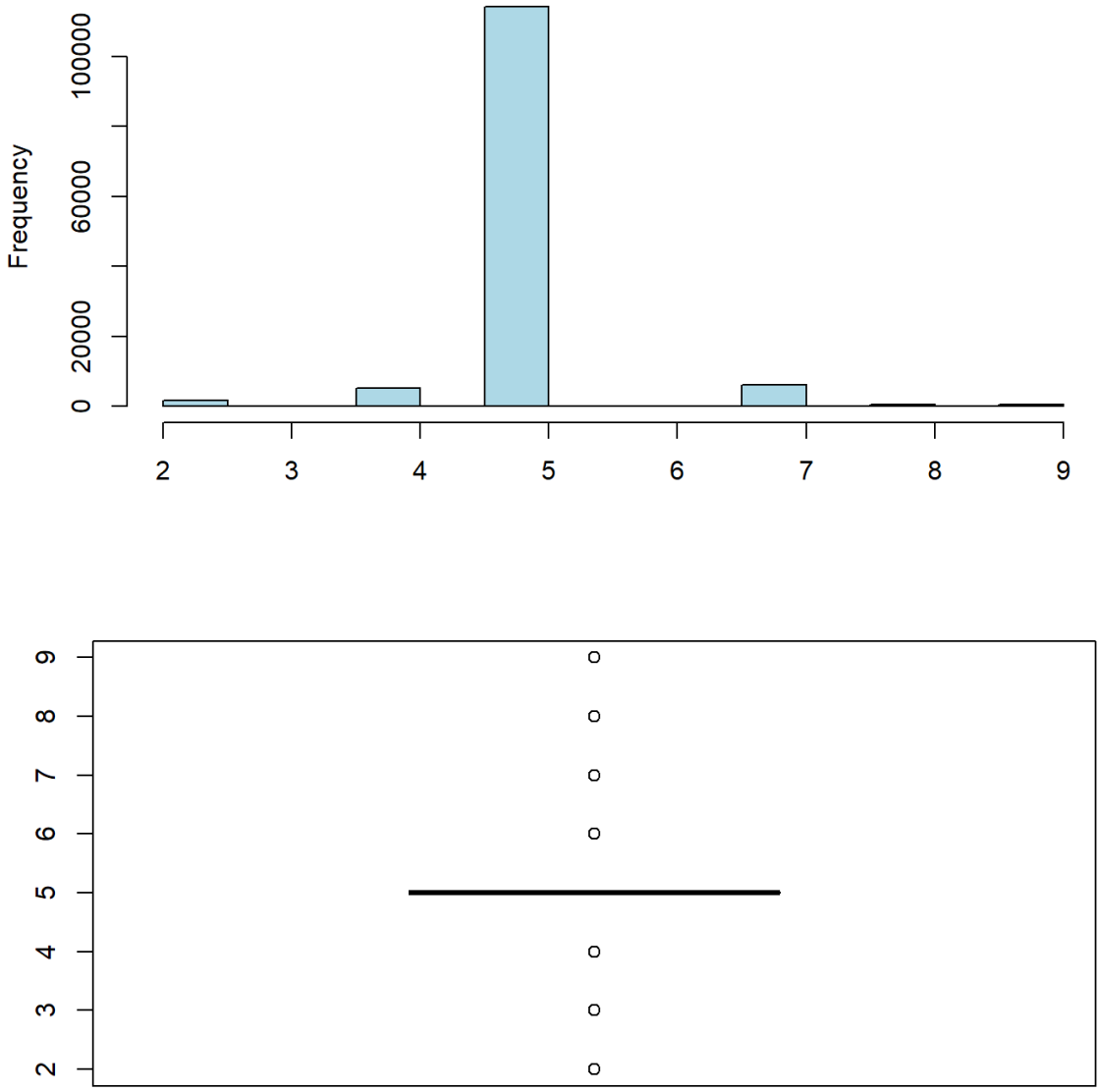
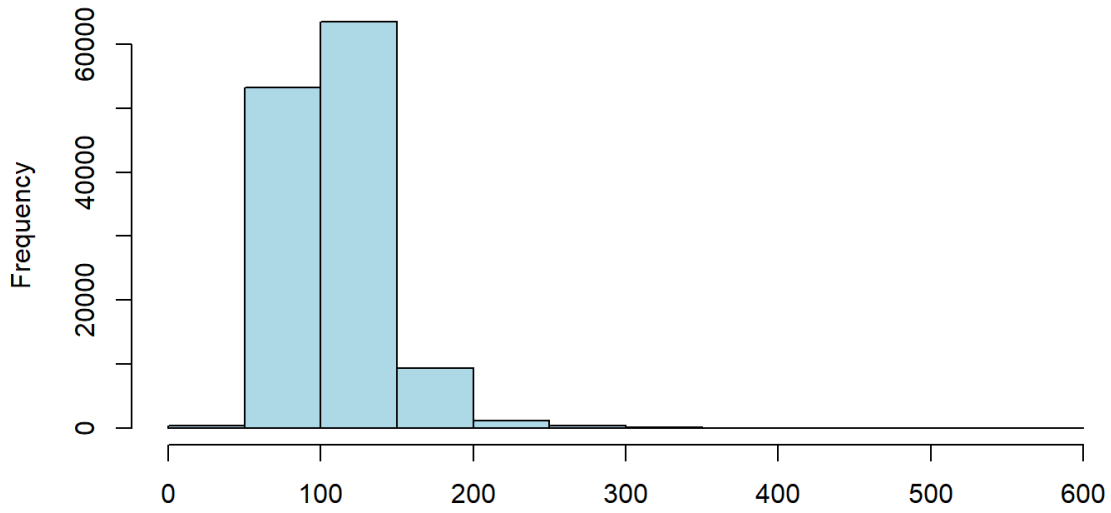


Figura 42. PLAZAS. Elaboración Propia.

- POTENCIA: el valor medio es de 112 y una desviación típica de 35. El valor mínimo es de 5 y el máximo de 551. En esta variable existen tanto valores atípicos como extremos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
128143	112	35	5	90	135	551



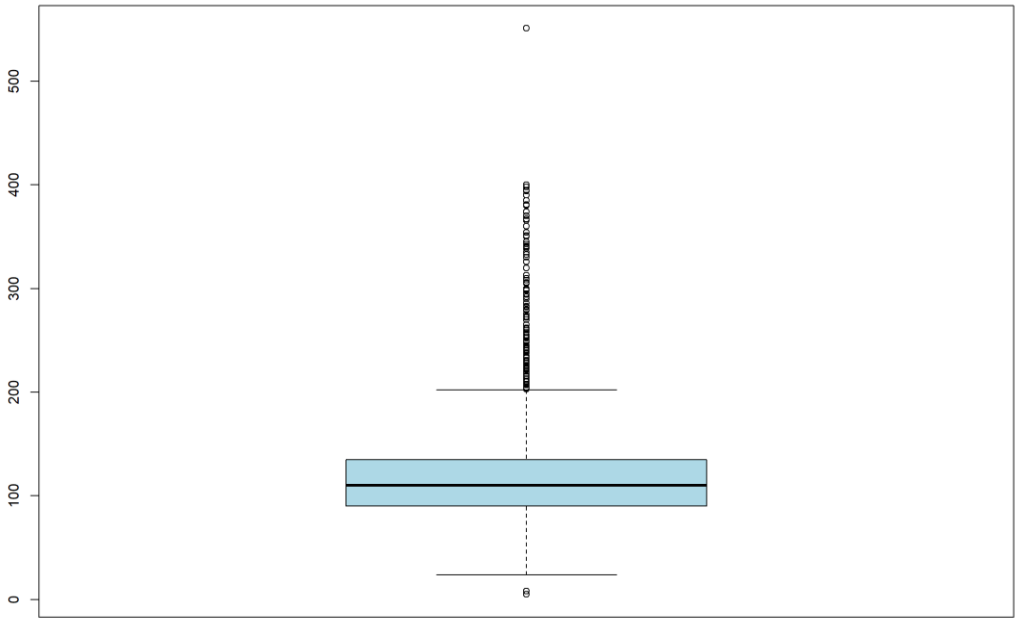


Figura 43. POTENCIA. Elaboración Propia.

- FEC_MATRICULACION: la fecha de matriculación media se sitúa el 28-09-2008, con la mínima el 01-01-1900 y la máxima el 31-01-2023.

```
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
"1900-01-01" "2004-01-05" "2008-01-18" "2008-09-28" "2015-01-08" "2023-01-31"
```

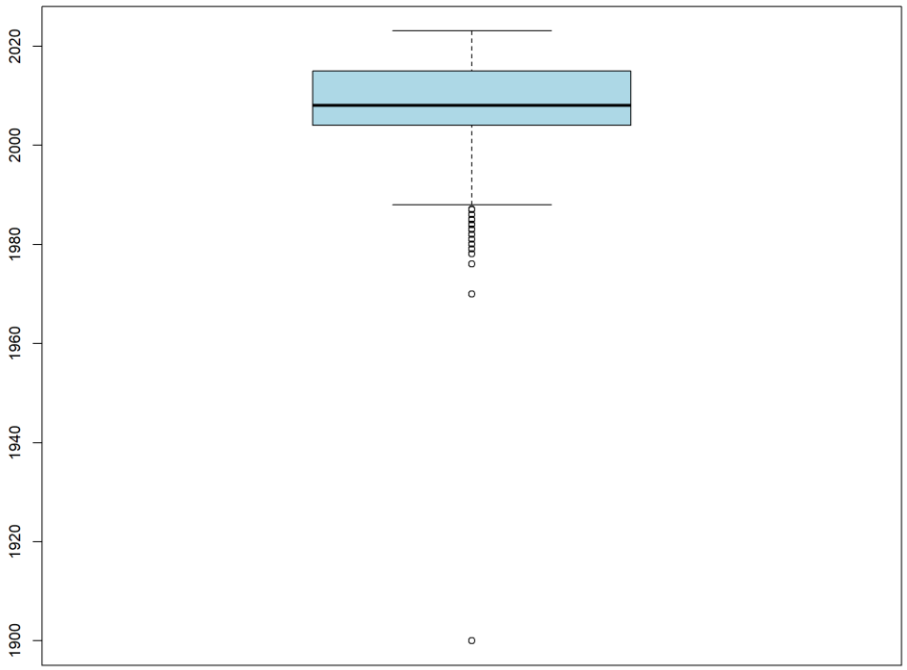


Figura 44. FEC_MATRICULACION. Elaboración Propia.

3.3 Depuración de la base de datos

Una vez que se ha hecho un análisis exhaustivo de todas las variables que conforman la base de datos a utilizar, es necesario llevar a cabo una depuración de los datos de aquellas variables que así lo requieran.

Las variables numéricas que necesitan ser tratadas son AÑOS_CIA_ANT, AÑOS_ASEGURADO, COND_OC, COND_OC_27, COND_OC_5, SCORING_SUSCRIP, ANT_VEHI, VALOR_VEHI, POTENCIA y FEC_MATRICULACION.

Al analizar la base de datos se observó que había incoherencias entre los AÑOS_CIA_ANT y AÑOS_ASEGURADO ya que existían casos donde AÑOS_ASEGURADO era menor que AÑOS_CIA_ANT y donde AÑOS_ASEGURADO era mayor que 0, pero AÑOS_CIA_ANT era igual a 0. En estas situaciones el tratamiento que se llevó a cabo fue establecer para la variable AÑOS_ASEGURADO el mismo valor que AÑOS_CIA_ANT para el primer caso, y para el segundo, establecer para AÑOS_CIA_ANT el valor de AÑOS_ASEGURADO.

Por otro lado, se percibió que también había ciertas incongruencias entre las variables COND_OC, COND_OC_27 y COND_OC_5. Para los casos en que COND_OC_27 era mayor que COND_OC o COND_OC_5 mayor que COND_OC, el valor de la variable se estableció como el mayor valor entre COND_OC_27 y COND_OC_5.

También fue tratada la variable SCORING_SUSCRIP, pues en el análisis exploratorio de datos se observó que existían algunos valores los cuales podían perjudicar al modelo de clustering por considerarse atípicos y extremos. Es por ello que se decidió establecer un criterio para su tratamiento. En caso de que la variable NP_CARTERA fuese igual a "Nueva Producción" y que el SCORING_SUSCRIP fuese mayor que 6, el valor asignado a esta última variable es de 6. Por otro lado, en caso de que NP_CARTERA fuese igual a "Cartera" y que el SCORING_SUSCRIP fuese mayor que 7, el valor de la variable SCORING_SUSCRIP es de 6.

La depuración de las variables ANT_VEHI, VALOR_VEHI, POTENCIA y FEC_MATRICULACION se realizó con el fin de eliminar los valores extremos que se pudieron detectar en el análisis exploratorio. De esta manera, se eliminó la observación que presentaba un valor de 121 en la variable ANT_VEHI. Asimismo, se eliminaron todas aquellas observaciones con VALOR_VEHI mayor de 80.000 y las que tuviesen el valor de POTENCIA de mayor o igual a 400 o menor o igual que 24. Finalmente, la variable FEC_MATRICULACION se trató eliminando todos aquellos puntos por debajo de 01-01-1988.

El resto de variables que también poseían valores atípicos tal y como se visualizaba en el análisis exploratorio, se han decidido por juicio de negocio no modificarlas y no aplicar ningún tipo de depuración.

En cuanto a las variables categóricas que necesitan ser tratadas son SCORING_ASEGURADOR, SCORING_ASEGURADOR_TRAM y SCORING_ASEGURADOR_TRAM_II, MODALIDAD, CIA_ANT, PROVINCIA, COMBUSTIBLE, GARAJE, KMS_ANUALES y USO.

Tal y como se observó en el análisis exploratorio de datos, en las variables SCORING_ASEGURADOR, SCORING_ASEGURADOR_TRAM y SCORING_ASEGURADOR_TRAM_II existían datos faltantes. Es por ello que se decidió que para la variable SCORING_ASEGURADOR todos los valores que no tuviesen datos se sustituían por la categoría "Z", al igual que las variables SCORING_ASEGURADOR_TRAM y SCORING_ASEGURADOR_TRAM_II que se sustituían por "3. Sin cesión - Z" y "Z", respectivamente, aunque finalmente se decide optar por incluir únicamente la variable SCORING_ASEGURADOR_TRAM ya que es la que agrupa por la información que se quiere analizar.

Por otro lado, se ha establecido, por regla general, que cada clase que componen las variables categóricas tiene que representar un 5% o cercano a él individualmente, aunque con ciertas excepciones por decisiones de negocio.

En primer lugar, la variable MODALIDAD ha sido transformada. Se ha agrupado la categoría TE a TL por la poca cantidad de datos que se tienen. El resto de categorías que no alcanzan el 5% se ha decidido no transformarlas por juicio experto de negocio.

Después, se ha transformado la variable CIA_ANT, en concreto, la categoría VERTI, SEGURCAIXA y GENERALI ya que los porcentajes sobre el total eran muy por debajo de un 5%. De esta manera, VERTI se ha agrupado con la categoría de MUTUA MADRILEÑA (esto se debe a que ambas no pertenecen al sistema SINCO) y SEGURCAIXA y GENERALI se ha agrupado a OTRA COMPAÑÍA. El porcentaje de la categoría MUTUA MADRILEÑA también se encuentra por debajo del 5%, pero se decide mantenerla ya que es una categoría en la que se tiene especial interés para su análisis.

Posteriormente, se ha creado una nueva agrupación con menos categorías que las que integraban la variable PROVINCIA. Tras la transformación, las categorías son "NORTE", "OESTE", "RESTO", "Coruña, A", "Lugo", "Ourense", "Pontevedra".

La variable COMBUSTIBLE se ha transformado creando únicamente tres categorías, estas son, “D”, “G” Y “RESTO” ya que en esta última se han integrado “B”, “E”, “L”, “P”, “X”, “Y” y “Z”. Aunque esta nueva categoría no alcance el 5% se decide utilizar en el estudio porque se quiere analizar en la segmentación.

En la variable GARAJE se ha pasado de tener cuatro categorías a tres ya que se han agrupado “Garaje colectivo con vigilancia” y “Garaje colectivo sin vigilancia” en “Garaje Colectivo”.

La variable KMS_ANUALES al haber poca cantidad de datos en las categorías “Hasta 30k”, “Hasta 40k” y “Más de 40k” se han agrupado con el tipo “Hasta 20k” quedando en total 3 categorías.

Por último, en la variable USO se establecen tres categorías en lugar de cuatro ya que “Profesional” y “Particular fin de semana” se sustituyen por la categoría “Resto”.

Si se analizan las variables que se han creado se tiene que:

- AÑOS_CIA_ANT: la única diferencia que se ha producido es que la media ha pasado de 4 a 4,1 y la desviación típica de 2,2 a 2,1.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
127816	4.1	2.1	0	2	6	6

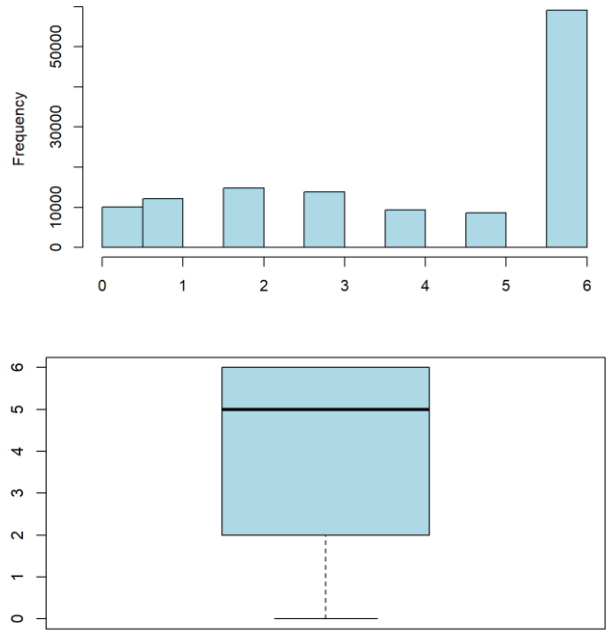


Figura 45. AÑOS_CIA_ANT DEPURADO. Elaboración Propia.

- AÑOS_ASEGURADO: no se han producido cambios apreciables.
- COND_OC, COND_OC_27 y COND_OC_5: no se han producido cambios apreciables.
- SCORING_SUSCRIP: la desviación típica ha pasado del 1,5 al 1,4. Además, el valor máximo ha pasado de ser 18 a 7, por lo que se han reducido los valores atípicos y extremos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
127816	1	1.4	0	0	1	7

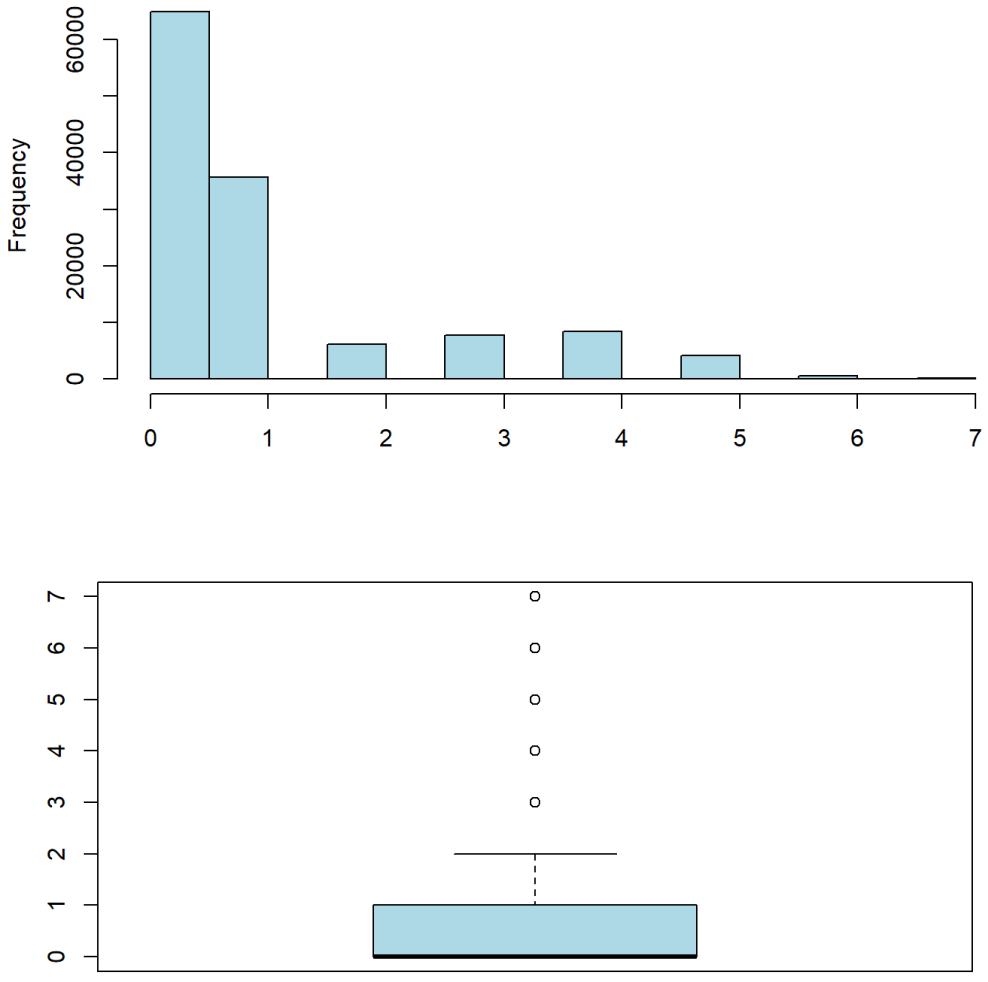
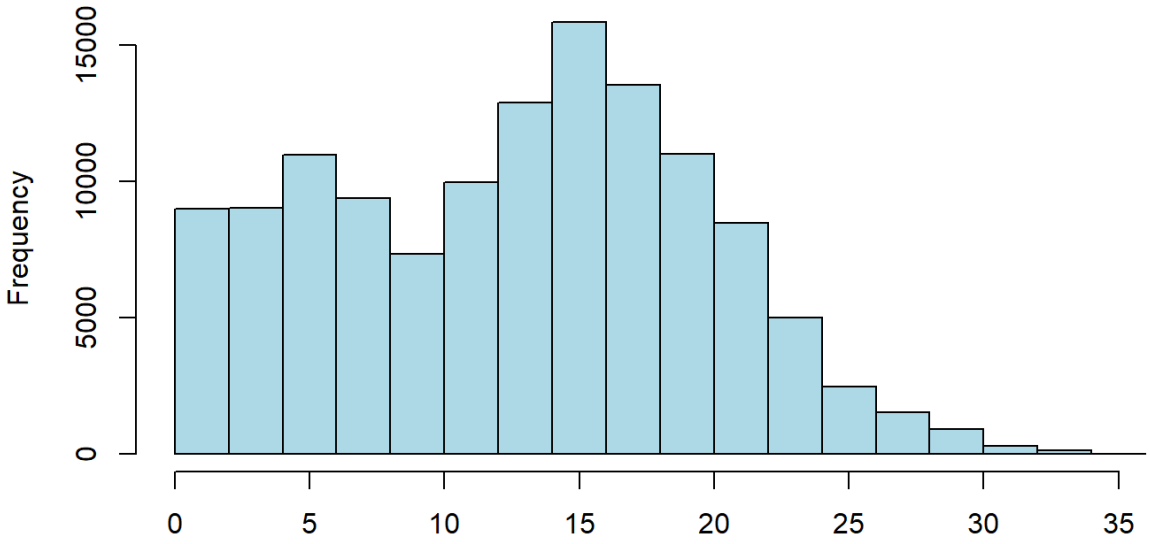


Figura 46. SCORING_SUSCRIP DEPURADO. Elaboración Propia.

- ANT_VEHI: la mayor diferencia que se ha producido ha sido que el valor máximo es ahora de 35 cuando, en la base de datos sin depurar era de 121. Se han eliminado la mayor parte de los valores atípicos y extremos.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
127816	13	7	0	7	18	35



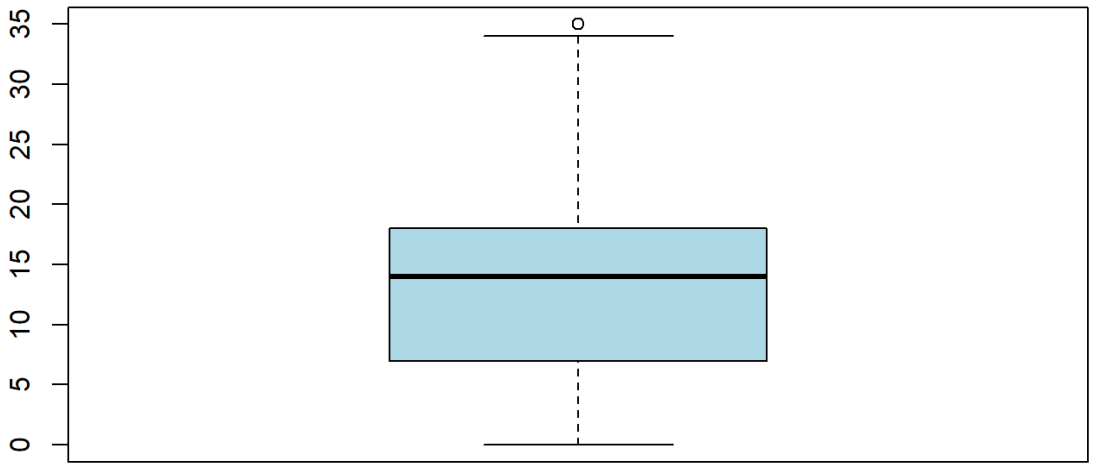


Figura 47. ANT_VEHI DEPURADO. Elaboración Propia.

- VALOR_VEHI: la media ha pasado de 22.934 a 22.831 y la desviación típica de 9.937 a 9.433. El máximo ha disminuido de 324.991 a 79.981. A su vez, valores atípicos también han descendido.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
127816	22831	9433	1803	16060	27550	79981

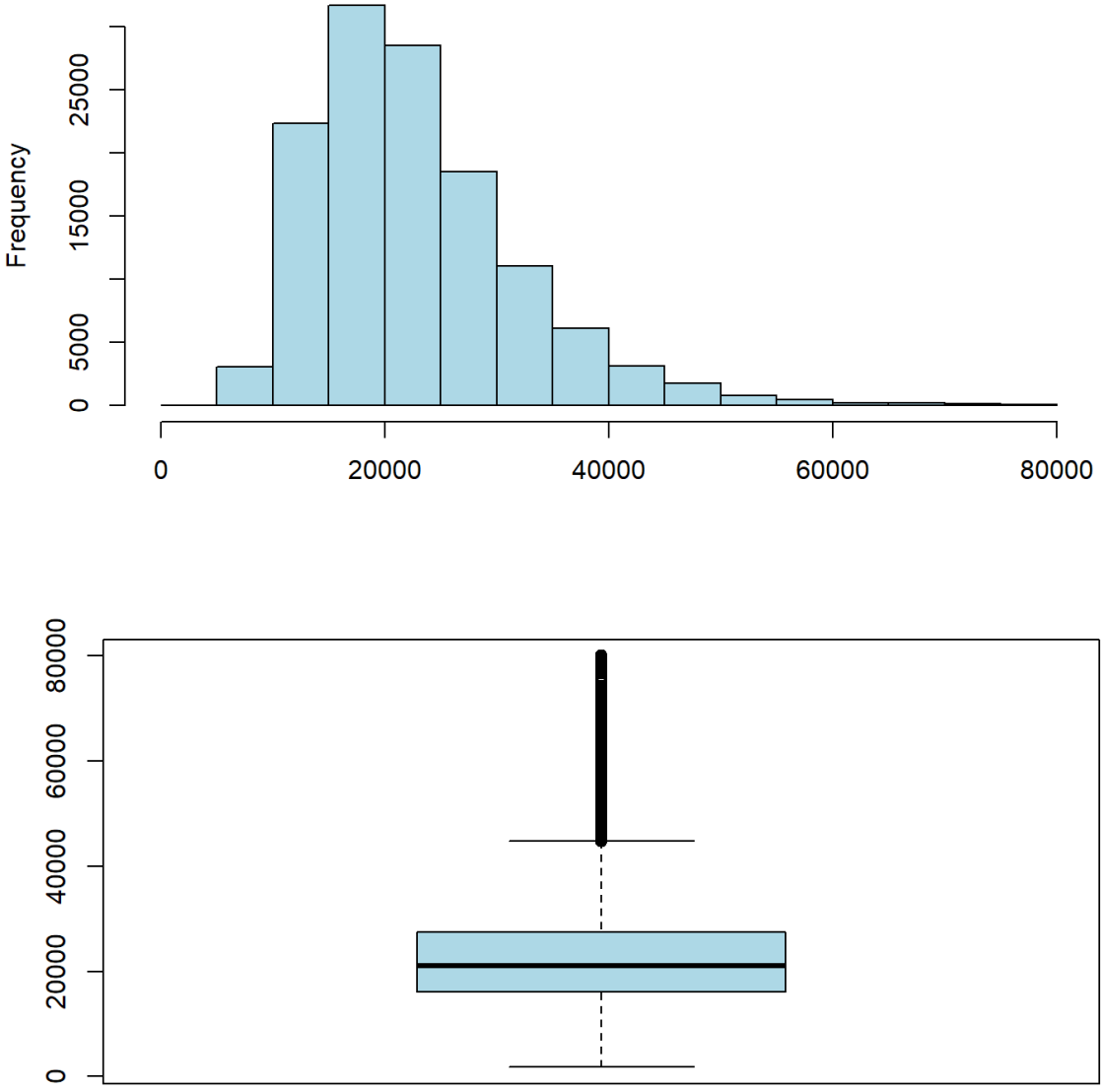
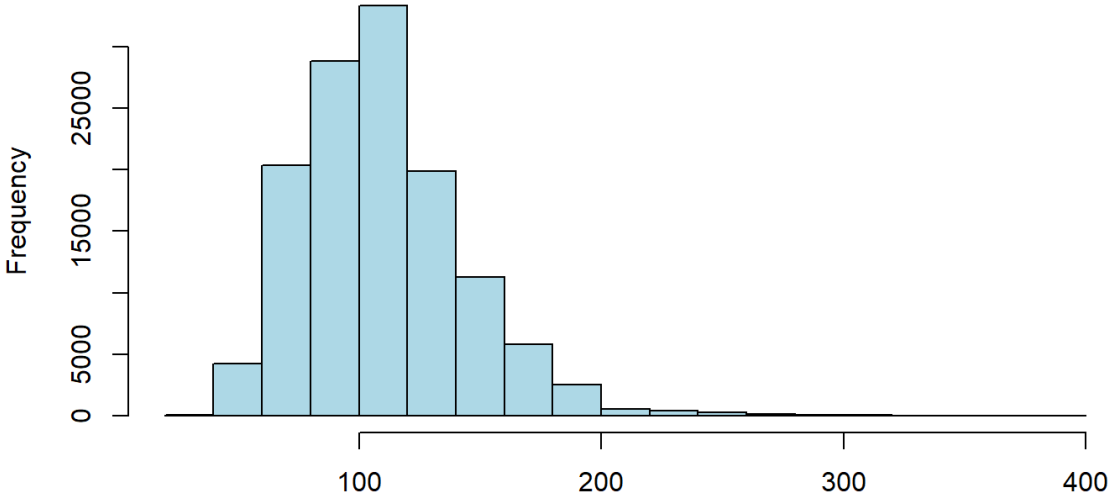


Figura 48. VALOR_VEHI DEPURADO. Elaboración Propia.

- POTENCIA: el valor medio ha disminuido de 112 a 111, la desviación típica de 35 a 34 y el máximo de 551 a 381. Por el contrario, el valor mínimo ha aumentado de 5 a 34.

N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
127816	111	34	34	90	133	381



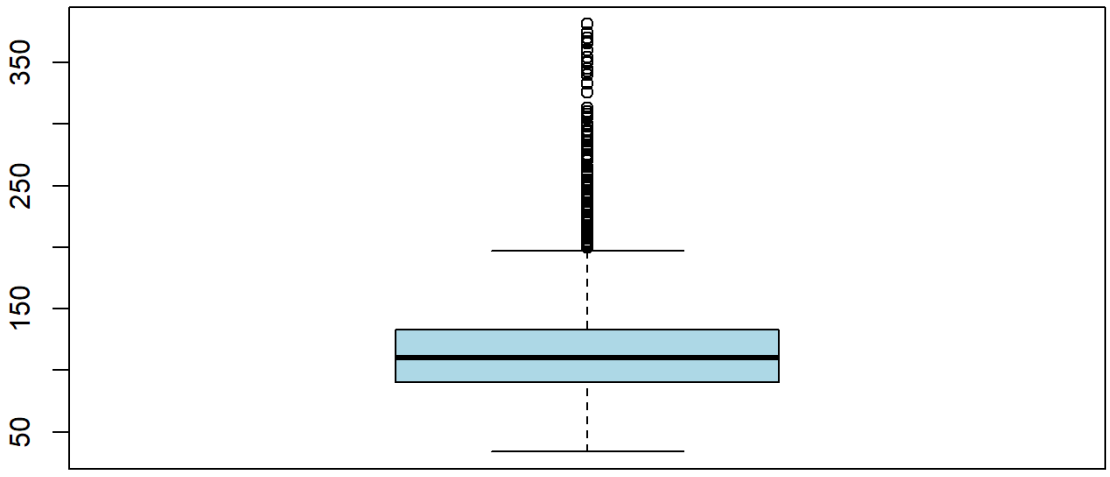


Figura 49. POTENCIA DEPURADO. Elaboración Propia.

-

- FEC_MATRICULACION: lo más destacable es que se han eliminado todo tipo de datos atípicos y extremos. Además, el valor mínimo ha cambiado de 01-01-1900 a 02-01-1988.

```
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
"1988-01-02" "2004-01-05" "2008-01-18" "2008-10-03" "2015-01-07" "2023-01-31"
```

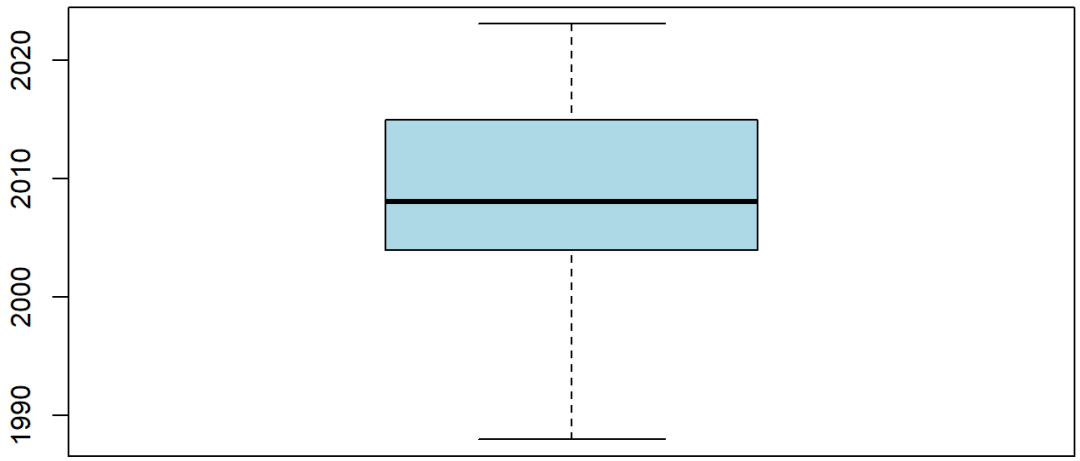


Figura 50. FEC_MATRICULACION DEPURADO. Elaboración Propia.

- SCORING_ASEGURADOR_TRAM: se pasan de tres a cuatro categorías.

1.Favorable - ABC	2.No favorable - DEFGH	3.Sin cesión - Z
99321	12310	16185
0.777	0.096	0.127

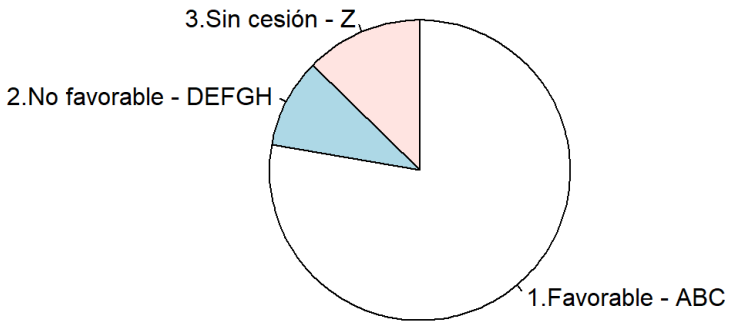


Figura 51. SCORING_ASEGURADOR_TRAM DEPURADO. Elaboración Propia.

- MODALIDAD: tras la agrupación de TE a TL el porcentaje sobre el total de este último ha pasado de 14,1% al 14,4%.

02. TL	03. TA	04. TRF400	05. TRF300	06. TRF200
18360	65065	2697	15579	21694
0.144	0.509	0.021	0.122	0.170

07. TRSF
4421
0.035

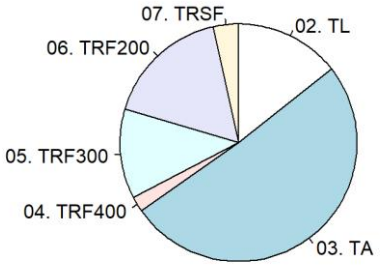


Figura 52. MODALIDAD DEPURADO. Elaboración Propia.

- CIA_ANT: tras la depuración, MUTUA MADRILEÑA representa el 3,2%, un aumento del 0,4% tras la inclusión de VERTI a la misma. Por otro lado, OTRA COMPAÑÍA ha transformado su porcentaje del 41,7% anteriormente a 45,7% tras la incorporación de SEGURCAIXA y GENERALI.

ALLIANZ	AXA	LIBERTY	MAPFRE	MUTUA MADRILEÑA
13601	19075	6052	18530	4030
0.106	0.149	0.047	0.145	0.032

OTRA COMPAÑIA	REALE
58398	8130
0.457	0.064

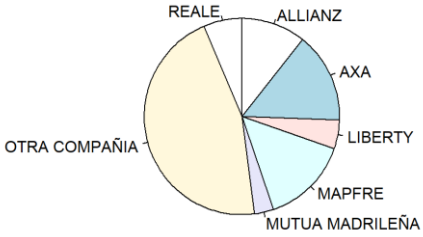


Figura 53. CIA_ANT DEPURADO. Elaboración Propia.

- PROVINCIA: tras la nueva agrupación, la categoría con mayor porcentaje es "Coruña, A" con un 35,6% y la que menos "OESTE" con un 5,3%.

Coruña, A	Lugo	NORTE	OESTE	Ourense
45535	10556	7116	6811	14955
0.356	0.083	0.056	0.053	0.117

Pontevedra	RESTO
35714	7129
0.279	0.056

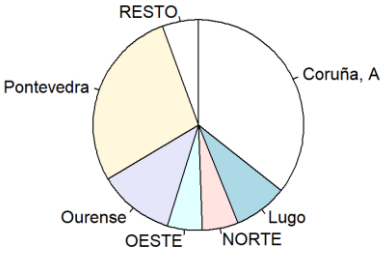


Figura 54. PROVICIA DEPURADO. Elaboración Propia.

- COMBUSTIBLE: se ha creado una nueva clase que es Resto, la cual representa, aproximadamente, el 2% del total.

D	G	RESTO
87362	37969	2485
0.683	0.297	0.019

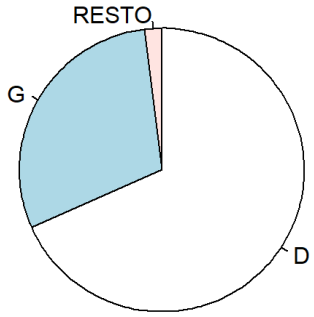


Figura 55. COMBUSTIBLE DEPURADO. Elaboración Propia.

- GARAJE: se ha creado la categoría "Garaje colectivo" que representa la suma de las categorías "Garaje colectivo con vigilancia" y "Garaje colectivo sin vigilancia" de antes de la depuración.

Garaje colectivo	Garaje individual	Sin garaje
43921	68193	15702
0.344	0.534	0.123

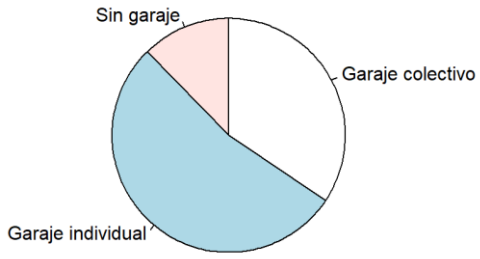


Figura 56. GARAJE DEPURADO. Elaboración Propia.

- KMS_ANUALES: sólo integra tres tipos de categorías, sobrepasando todas el 5% que se ha establecido como mínimo por norma general.

Hasta 15k	Hasta 20k	Hasta 5k
63302	23538	40976
0.495	0.184	0.321

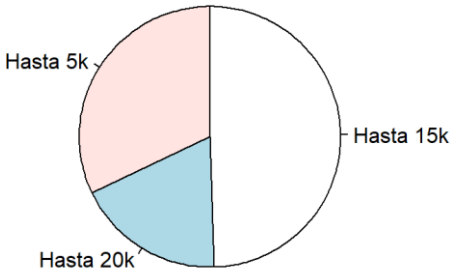


Figura 57. KMS_ANUALES DEPURADO. Elaboración Propia.

- USO: tras la suma de las categorías “Profesional” y “Particular fin de semana” la categoría “Resto” contiene el 4,7% de los datos, dato cercano al 5% elegido.

Particular diario	Particular ocasional	Resto
83204	38639	5973
0.651	0.302	0.047

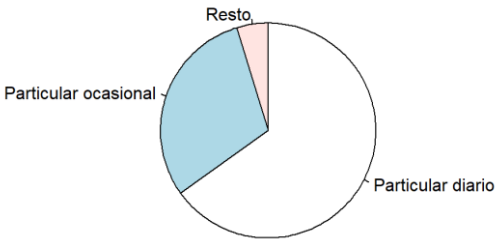


Figura 58. USO DEPURADO. Elaboración Propia.

Una vez que se ha llevado a cabo la agrupación de variables, se ha procedido a ver qué variables son informativas y cuáles no, es decir, aquellas que se pueden eliminar del estudio de segmentación porque no aportan información.

Para ello, se ha empleado la librería “caret” y la función “nearZeroVar”, estableciendo el parámetro uniqueCut en 10.

Aquellas variables que no son informativas son PROTECCION_LEGAL_PREMIUM, AV_MECANICA, VALORACION_+, PDC, COND_OC_27, COND_OC_5, STDAD_DP y STDAD_RC por lo que se decide eliminarlas como variables a utilizar en la segmentación.

3.4 Creación de las bases de datos

Con el fin de poder llevar a cabo la segmentación del conjunto de datos, se ha decido crear dos tipos de bases de datos. Por un lado, una que contenga únicamente variables numéricas y otra con variables numéricas, pero también categóricas:

1. BASE DE DATOS NUMÉRICA: en primer lugar, para que todas las variables de la base de datos fuesen de tipo numérico, aquellas variables de naturaleza categórica se transforman en variables binarias (1 o 0). De esta manera, hay que crear variables para NP_CARTERA, MODALIDAD, MODALIDAD_AGRUPADA, CIA_ANT, PROVINCIA, COMBUSTIBLE, KMS_ANUALES, TERRITORIAL_TRAMO, TARGET_PROPUESTO, GARAJE, SCORING_ASEGURADOR_TRAM, USO y SEGMENTO_NUEVO_ORDENADO.

Además, se decide crear la variable NUM_OPC_CONTRATADAS en dummy siendo el valor 1 si se ha contratado alguna opcional y 0 en caso de que no, sustituyendo esta por la variable original. Lo mismo ocurre con COND_OC que se elimina y se crea a partir de esta una variable dummy donde 1 indica cuando hay mínimo 1 conductor ocasional y 0 cuando no lo hay.

Asimismo, aunque STDAD_DP Y STDAD_RC no se van a incluir en la segmentación, se ha creado una variable dummy donde 1 indica que se ha producido algún tipo de siniestralidad y 0 cuando no.

Una vez creadas las variables binarias, se ha procedido al análisis bivariante de las mismas. En él se ha observado que existen correlaciones muy altas entre ciertas variables.

Se elige el límite de 0,85, por lo que, para el par de variables que tengan ese valor o por encima, se eliminará la variable menos relevante.

En este sentido, se elimina la variable SEGMENTO_NUEVO_ORDENADO y FEC_MATRICULACION.

Por otro lado, se observa que se repite información entre las variables PROVINCIA y TERRITORIAL_TRAMO por lo que, se establece mantener la característica TERRITORIAL ya que en este caso no sido necesario transformar los datos y se van a hacer uso de ellos tal y como se recibieron en la base original.

Además, se encuentran valores altos de correlación entre VALOR_VEHI y POTENCIA. Como el valor más objetivo en este caso es el de POTENCIA y la variable VALOR_VEHI es una estimación se decide eliminar esta última variable.

Existe una correlación negativa cercana a menos uno entre las variables COMBUSTIBLE_GASOLINA_DUMMY y COMBUSTIBLE_DIESEL_DUMMY. Al haber más proporción de COMBUSTIBLE_DIESEL_DUMMY se elige incluir esta última.

Se elimina la variable USO_OCASIONAL_DUMMY, pues la correlación con USO_DIARIO_DUMMY es prácticamente -1. La decisión de mantener USO_OCASIONAL_DUMMY se basa en que tiene mayor proporción con respecto a la otra variable.

2. BASE DE DATOS MIXTO: incluye todas las variables categóricas y todas las variables numéricas. Para este conjunto de datos también se realizan algunas transformaciones específicas.

En primer lugar, al igual que en la base de datos numérica, se eliminan todas las variables que no son informativas. Estas son PROTECCIÓN_LEGAL_PREMIUM, AV_MECANICA, VALORACIÓN+, PDC, COND_27, COND_5, STDAD_DP y STDAD_RC.

Tal y como se decidió en la base de datos numérica, se eliminan las variables NUM_OPC_CONTRATADAS y COND_OC y se sustituyen por las variables dummy asociadas a estas columnas que ya se crearon en el conjunto de datos numérico.

También se elimina la variable FEC_MATRICULACIÓN ya que se observa que la información es la misma que la dada por ANT_VEHI. Asimismo, se elimina la variable SEGMENTO_NUEVO_ORDENADO.

Por último, como se ha detectado con el análisis de correlación de la base de datos numérica, al ser la correlación alta entre VALOR_VEHI y POTENCIA, se elimina VALOR_VEHI por el motivo expuesto anteriormente.

4. RESULTADOS DE LA APLICACIÓN

Para llevar a cabo la segmentación de la cartera de autos se optó, teniendo en cuenta la base de datos facilitadas, así como las ventajas e inconvenientes de cada método, los algoritmos K-Means y K-Prototypes.

Se ha decidido no usar los algoritmos jerárquicos ya que no serían los más adecuados porque la dimensión de la base de datos es muy grande y sería muy compleja la visualización y el análisis de los dendrogramas.

También se descartó el algoritmo K-Modes ya que los datos de las bases de datos eran numéricos y categóricos. Asimismo, no se hizo uso del método DBSCAN por posibles puntos en el conjunto de datos donde los clústeres tuviesen densidad cambiante.

Por tanto, como la cantidad de observaciones y variables que componen la base de datos es muy elevada, se utilizó el algoritmo K-Means para la base de datos numérica creada y K-Prototypes para la mixta.

4.1 Con algoritmo K-Means

Para poder hacer la segmentación con el método de K-Means se necesita la base de datos numérica que se ha descrito anteriormente.

Sin embargo, antes de ejecutar el algoritmo es necesario escalar los datos. Escalar los datos es un paso esencial en los métodos de clustering ya que, al calcular las distancias de los diferentes elementos del conjunto de datos, se hacen uso de los valores de las columnas.

Si las escalas de estas son diferentes entre sí, el algoritmo K-Means sólo tendrá en cuenta aquellas características que cuenten con valores mayores. Por ello, hay que escalar los datos antes de ejecutar el método.

Para escalar los datos se utiliza la función “scale”. Con ella, los datos quedan escalados de manera que tengan media 0 y desviación típica 1.

Por otro lado, el método K-Means necesita de la elección del número de clústeres que se quieren crear. Para poder intuir el valor correcto se utiliza el “Método del codo”.

El “Método del codo” se basa en las distancias intra-clúster, es decir, cuanto mayor es el número de clústeres, disminuye la varianza intra-clúster. Cuanto más pequeña es la distancia intra-clúster mejor ya que más compactos serán los clústeres.

Por ello, el “Método del codo” busca el número de clústeres k que haga que un aumento de k, no mejore de manera significativa la distancia media intra-clúster.

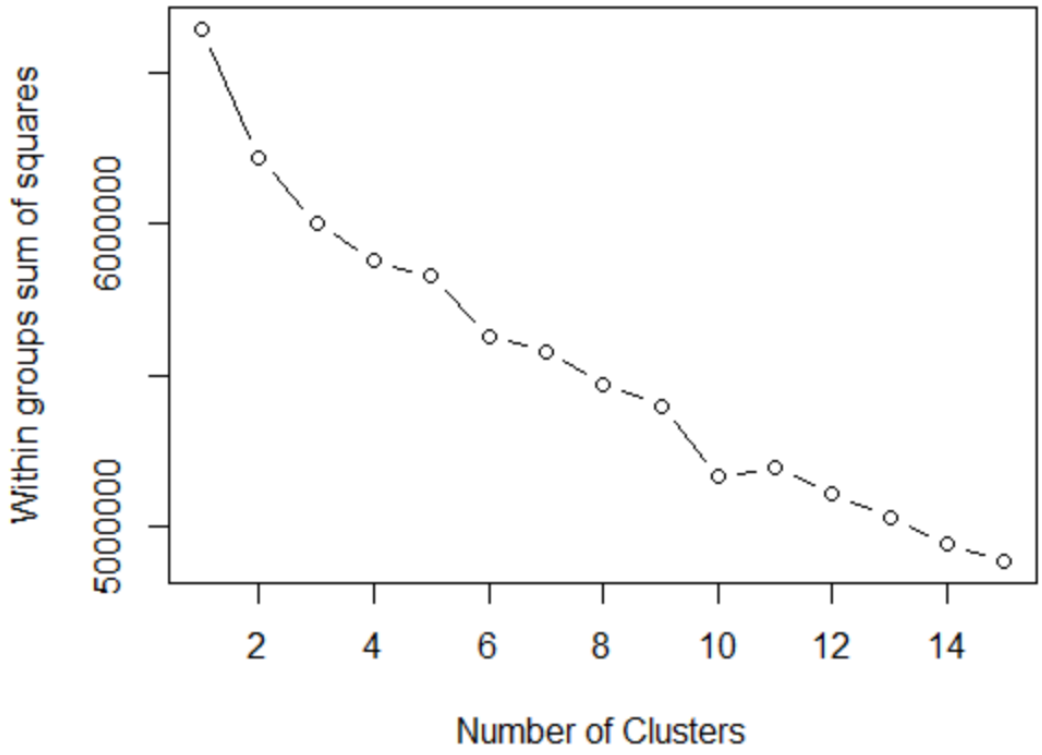


Figura 59. Método del Codo, K-Means. Elaboración Propia.

Tras analizar gráficamente los resultados se eligen como posibles valores de k, esto es, el número de clústeres óptimo, 3, 4 o 5 grupos ya que no se puede concluir ningún valor concreto. Para ello, se ha utilizado un algoritmo que necesita de la función “kmeans” del paquete “Stats” y se ha elegido como distancia la euclídea.

Posteriomente, se aplica para cada nivel k, la misma función “kmeans” utilizada anteriomente, con el fin de asignar a cada observación con uno de los grupos creados. De esta manera, por ejemplo, con k = 3, cada una de las observaciones estará asociada al grupo 1, grupo 2 o grupo 3. Mismo proceso se aplica cuando k = 4 y k = 5.

A continuación, se ha procedido a analizar gráficamente los resultados de los clústeres para las distintas k elegidas, utilizando para ello la función “fviz_cluster” del paquete “factoextra”, habiendo obtenido los siguientes gráficos:

- Para k = 3:

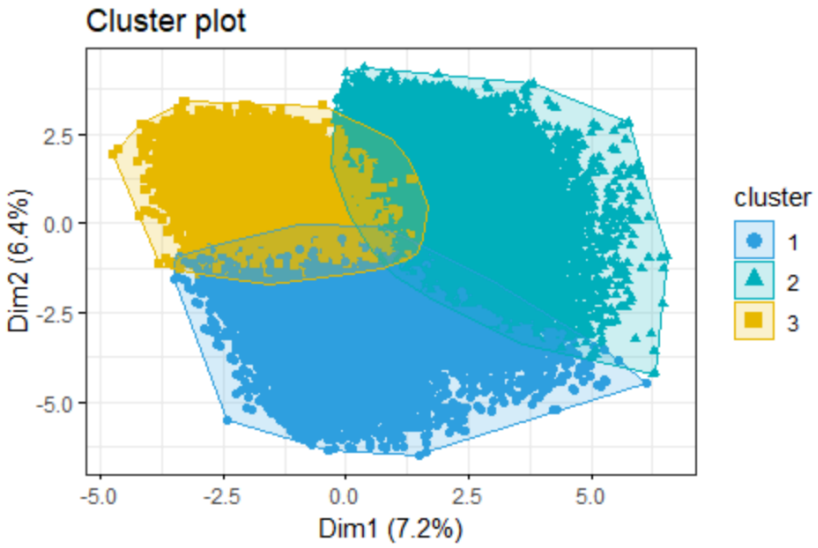


Figura 60. K-Means, k = 3. Elaboración Propia.

- Para k = 4:

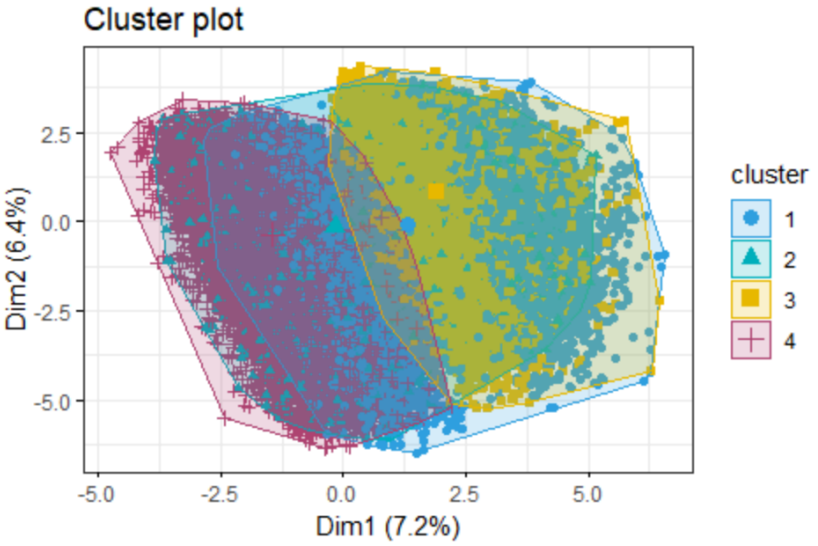


Figura 61. K-Means, k = 4. Elaboración Propia.

- Para k = 5:

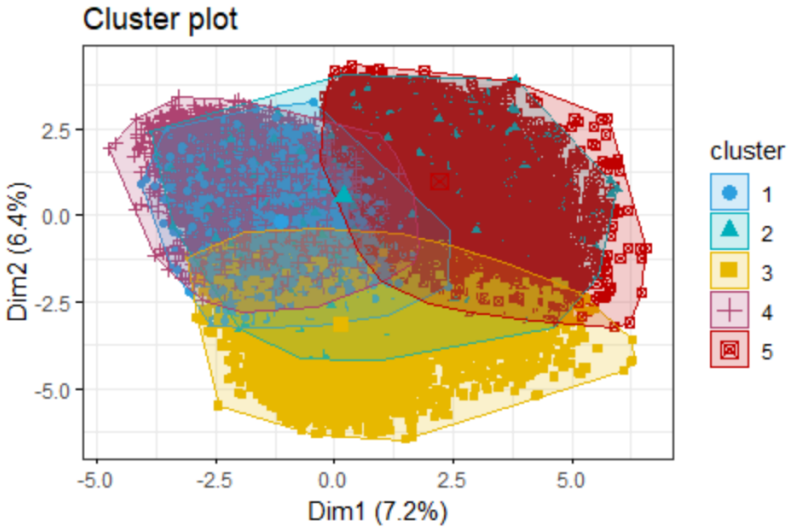


Figura 62. K-Means, k = 5. Elaboración Propia.

Tal y como se puede apreciar, los mejores resultados de la agrupación se dan cuando se crean tres grupos, esto es, cuando k = 3. Con k = 4, no hay una clara diferenciación entre los grupos, pues los clústeres se encuentran superpuestos entre sí. Finalmente, con k = 5, los resultados son muy similares a cuando se crean tres grupos, pero con clústeres que se solapan unos con otros y que no hay una clara separación entre ellos, por lo que dicha agrupación no añade mayor información que con k = 3.

En resumen, cuando la segmentación de la cartera de autos se realiza eligiendo formar tres clústeres (k = 3), se aprecian a simple vista tres grupos claramente diferenciados entre sí. De esta manera, se descartan las agrupaciones creadas cuando el algoritmo de K-Means es ejecutado con k = 4 y k = 5 y se procede a analizar los tres grupos creados cuando k =3.

4.1.1 Análisis de los grupos creados con k = 3

Una vez que ha sido ejecutado el algoritmo K-Means con k = 3, hay que llevar a cabo un análisis de cada grupo con el fin de identificar características representativas de cada grupo. Para realizarlo se ha empleado la función “create_report” con el paquete “DataExplorer”.

A nivel general, los grupos creados están divididos de la siguiente manera, siendo el grupo 3 el que mayor número de observaciones contiene y el 1 el que menos:

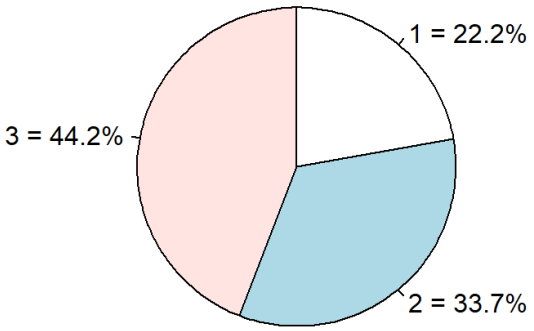


Figura 63. Distribuciones clústeres. Elaboración Propia.

Por tanto, las características propias de cada grupo son:

-Grupo 1: una de las principales características que representan a este grupo son las variables que dependen de la edad. En este sentido, se puede observar que el grupo 1 lo componen principalmente personas con edades no muy elevadas, mucho menores que aquellas que caracterizan al resto de grupos. Es por ello que, la mayor masa de datos de antigüedad de carnet se sitúa entre los 0 y 20 años, al igual que para la variable edad hay mayor volumen de observaciones en el intervalo de 20 a 40 años, para posteriormente ir disminuyendo. La distribución de ambas variables tiene una asimetría en la cola de la derecha.

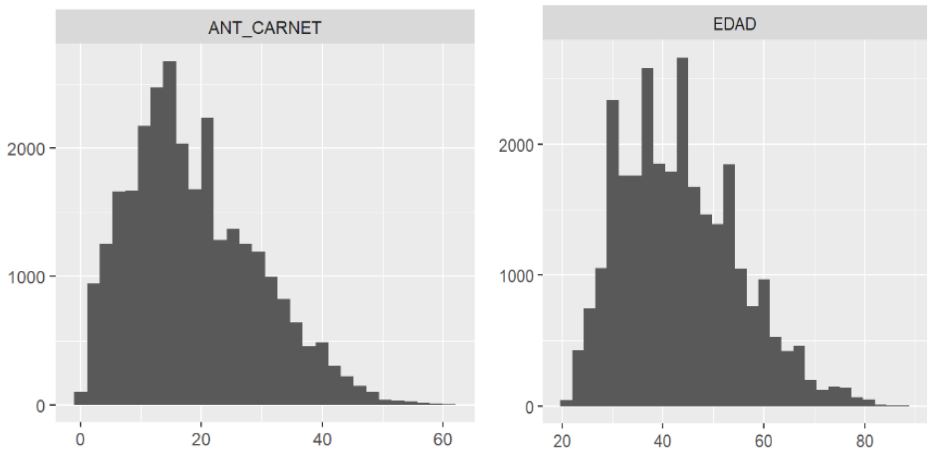


Figura 64. Distribuciones ANT_VEHI y EDAD, K-Means 1. Elaboración Propia.

Otra de las variables a destacar en este grupo son los años asegurados. Mientras que en los grupos 2 y 3 la mayor parte de los puntos se encuentran en el valor seis, en este grupo hay un gran porcentaje de los datos que se encuentra en el valor cero.

Asimismo, el valor más repetido en la variable AÑOS_CIA_ANT es el cero, totalmente contrario al valor más repetido, el seis, en el resto de grupos.

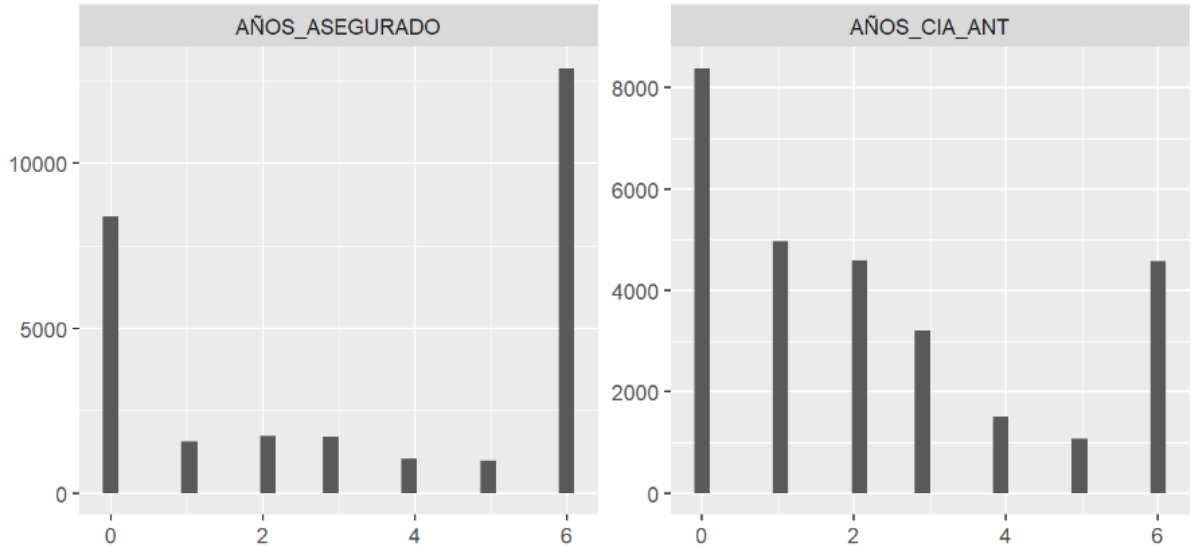


Figura 65. AÑOS_ASEGURADO y AÑOS_CIA_ANT, K-Means 1. Elaboración Propia.

Por otro lado, el valor de DTO_CAMPAÑA que principalmente se aplica a este grupo es de 0,00.

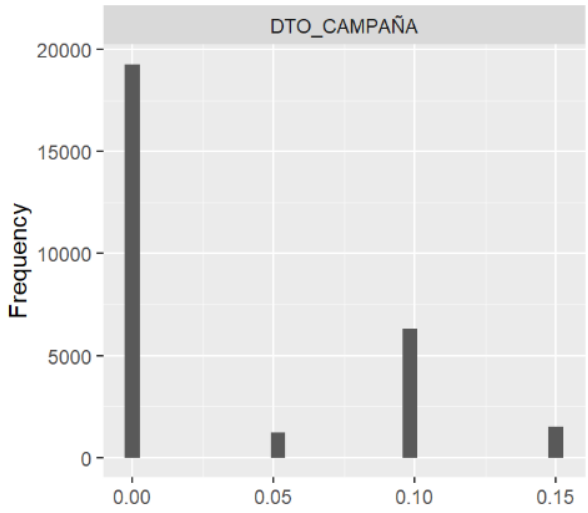


Figura 66. DTO_CAMPAÑA, K-Means 1. Elaboración Propia.

En cuanto a la variable TARIFA_PLANA, hay pocas observaciones en comparación con el resto de clústeres en el valor 0.

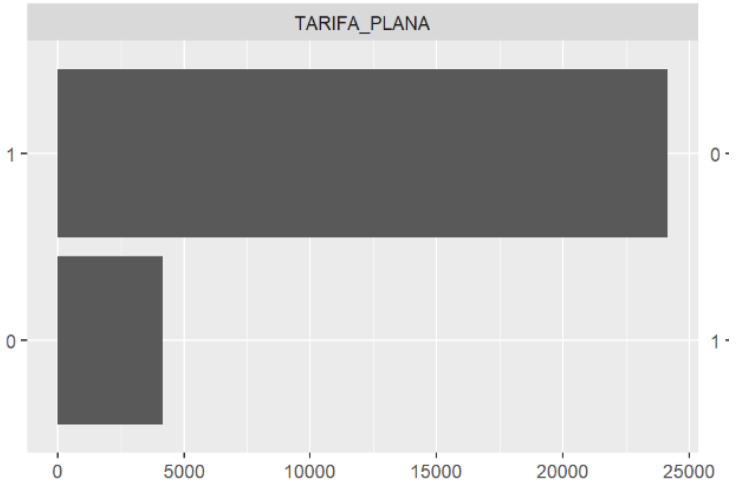


Figura 67. TARIFA_PLANA, K-Means 1. Elaboración Propia.

Otra de las características propias de este grupo es la variable GARAJE_SIN_DUMMY. Una gran parte de las observaciones de este grupo no poseen garaje, porcentaje más alto que en el resto de segmentos.

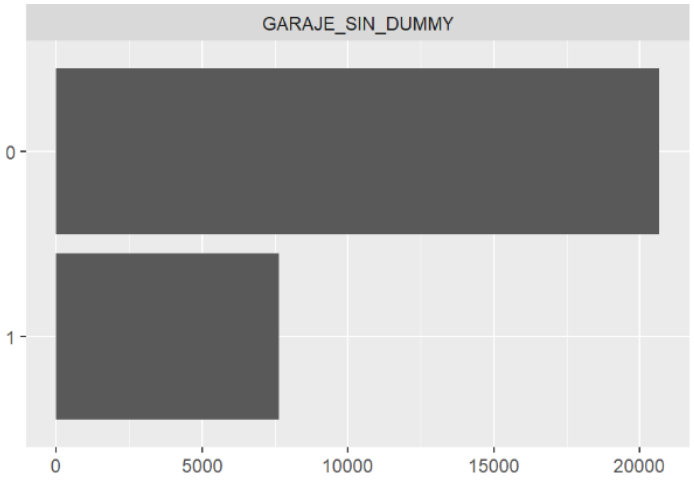


Figura 68. GARAJE_SIN_DUMMY, K-Means 1. Elaboración Propia.

Este grupo se caracteriza porque hay mayor porcentaje de personas que en el resto de clústeres con personas que no consienten la cesión de datos bancarios.

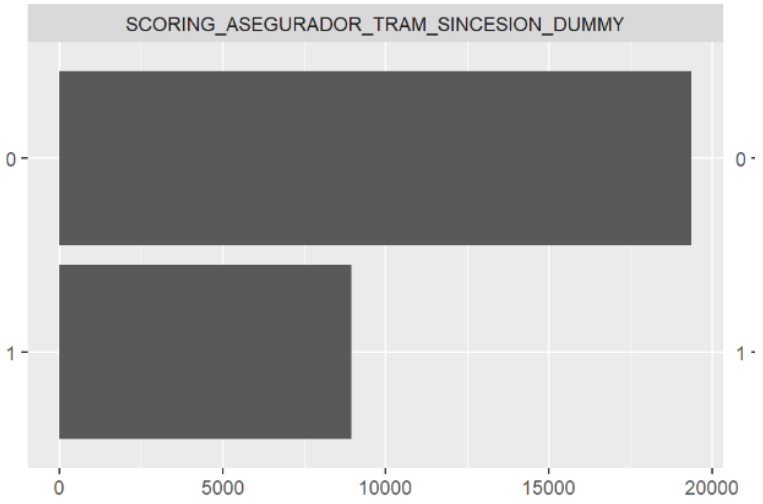


Figura 69. SINCESION_DUMMY, K-Means 1. Elaboración Propia.

Por últmo, si se analiza la variable TARGET_PROPUESTO_DUMMY se puede apreciar que este clúster se caracteriza porque la mayoría de personas calificadas como no target se encuentran en él.

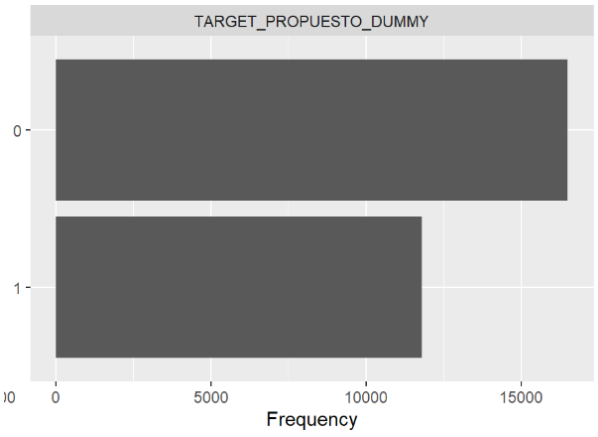


Figura 70.TARGET_PROPUESTO _DUMMY, K-Means 1. Elaboración Propia.

-Grupo 2: al igual que ocurría en el anterior grupo, las variables que dependen de la edad tienen una especial relevancia. En este sentido, la mayor masa de datos de antigüedad de carnet se sitúa entre los 20 a 40 años, al igual que la edad en la que hay mayor volumen de observaciones es el intervalo de 40 a 70 años.

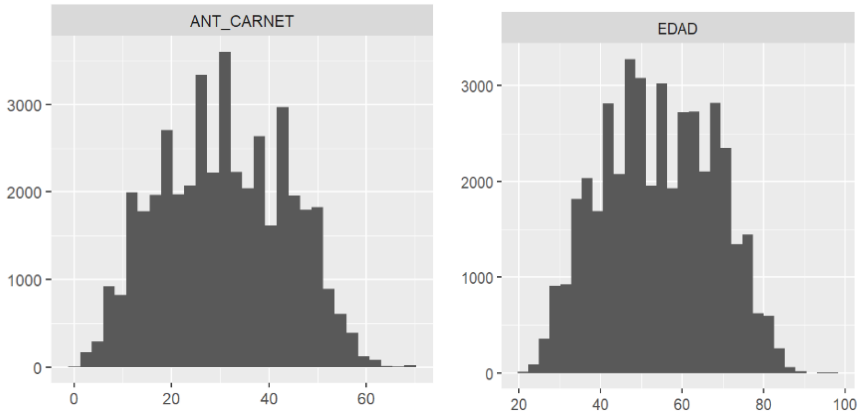


Figura 71. ANT_VEHI y EDAD, K-Means 2. Elaboración Propia.

Por otra parte, el clúster dos se caracteriza porque se aplican más tipos de descuentos que en el resto de grupos. Además, el valor de la variable de SCORING_SUSCRIP es mayor que en el resto de clústeres.

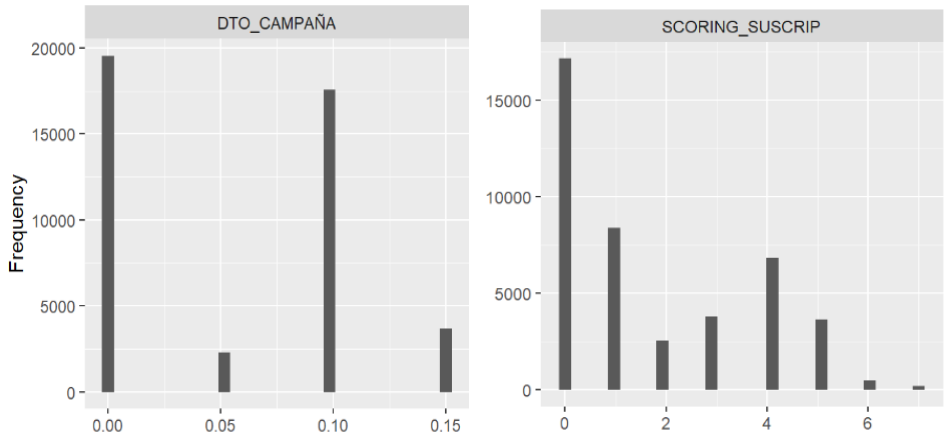


Figura 72. DTO_CAMPAÑA y SCORING_SUSCRIP, K-Means 2. Elaboración Propia.

Tal y como se observa en los gráficos, el segundo grupo prácticamente no tiene porcentaje de observaciones con la modalidad TL o TA.

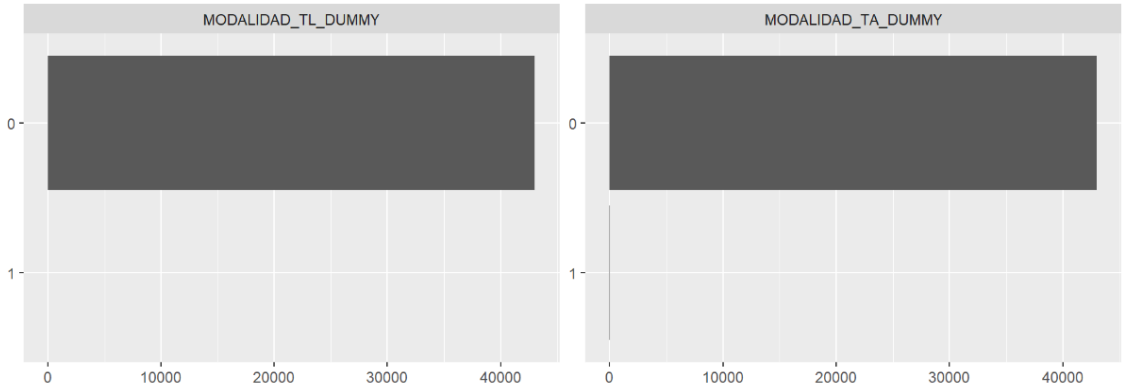


Figura 73. MODALIDAD_TL y MODALIDAD_TA, K-Means 2. Elaboración Propia.

Además, esto se corrobora con la variable MODALIDAD_AGRUPADA_TERCERO_DUMMY. Como se puede observar, el grupo dos prácticamente no tiene observaciones que tengan modalidad de "Tercero" ya que la gran mayoría pertenece a "Todo Riesgo".

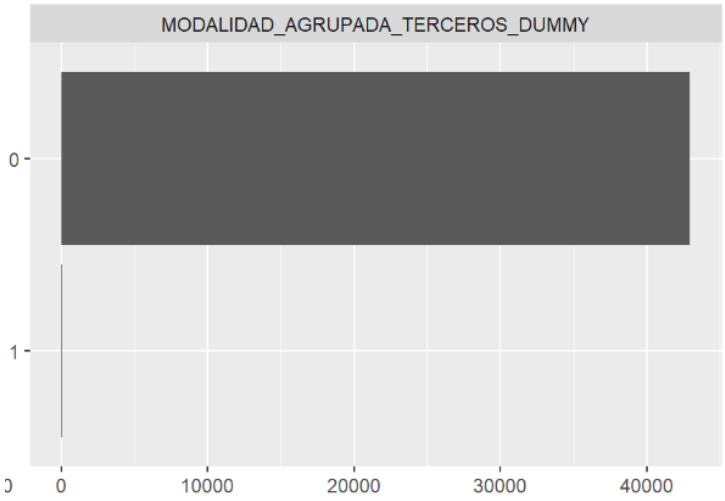


Figura 74. MODALIDAD_AGRUPADA_TERCEROS, K-Means 2. Elaboración Propia.

-Grupo 3: como características propias de este grupo vuelven a ser determinantes aquellas que tienen relación con la edad, al igual que en los otros dos grupos ya analizados.

En este sentido, la mayor masa de datos de antigüedad de carnet se encuentra desde los 30 a los 50 años, al igual que la edad donde hay mayor volumen de observaciones es en el intervalo de 50 a 80 años.

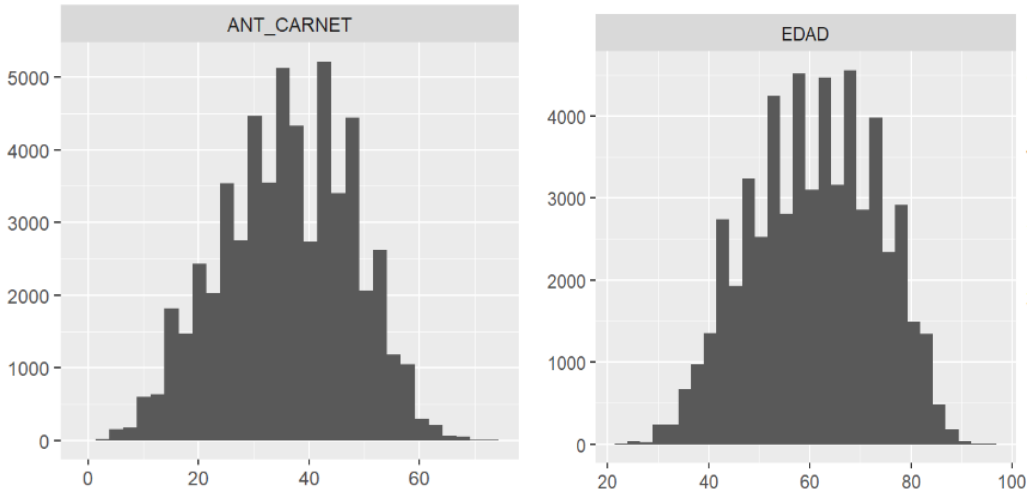


Figura 75. ANT_VEHI y EDAD, K-Means 3. Elaboración Propia.

Asimismo, el grupo tres destaca porque el Scoring de suscripción es muy bajo en comparación con el resto de clústeres creados, sin apenas valores a partir de 1.

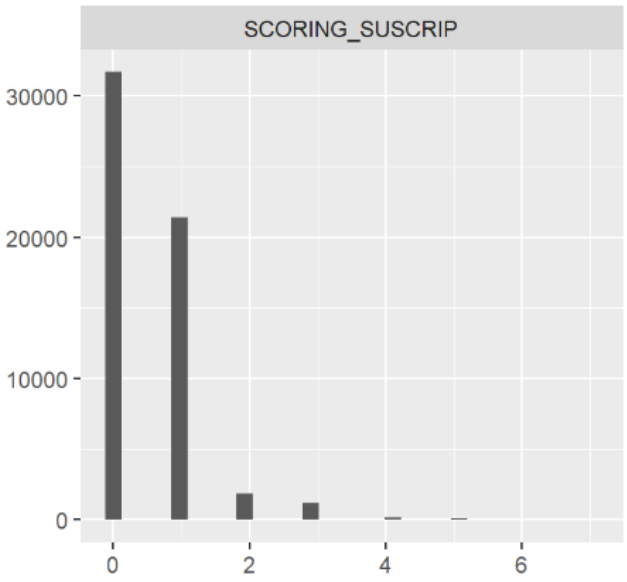


Figura 76. SCORING_SUSCRIP, K-Means 3. Elaboración Propia.

Otra de las características relevantes es aquella referente a la modalidad. Tal y como se puede observar en los gráficos, este grupo se caracteriza por estar formado únicamente por personas cuya modalidad es sólo de “Terceros”, sin haber observaciones referentes a “Todo Riesgo”, tanto con franquicia como sin franquicia.

Figura 77. MODALIDAD, K-Means 3. Elaboración Propia.

En cuanto a los kilometros anuales, este grupo en comparación con el resto, presenta mayor número de datos correspondientes a la categoría de “Hasta 5km anuales”.

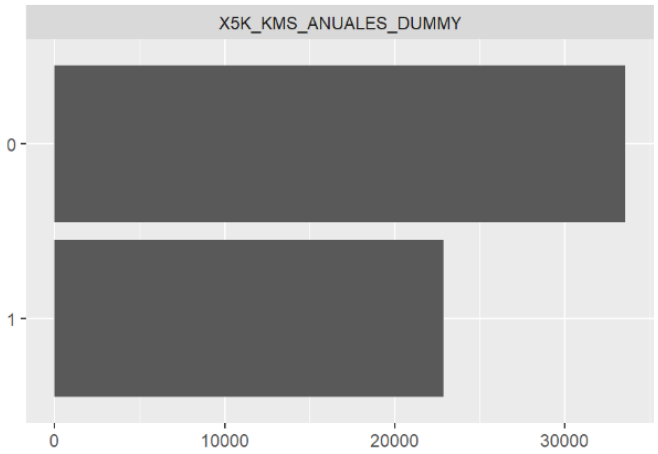


Figura 78. KMS_ANUALES, K-Means 3. Elaboración Propia.

En cuanto al Scoring Asegurador, prácticamente todos los datos que pertenecen a este grupo se consideran que no son tramo no favorable.

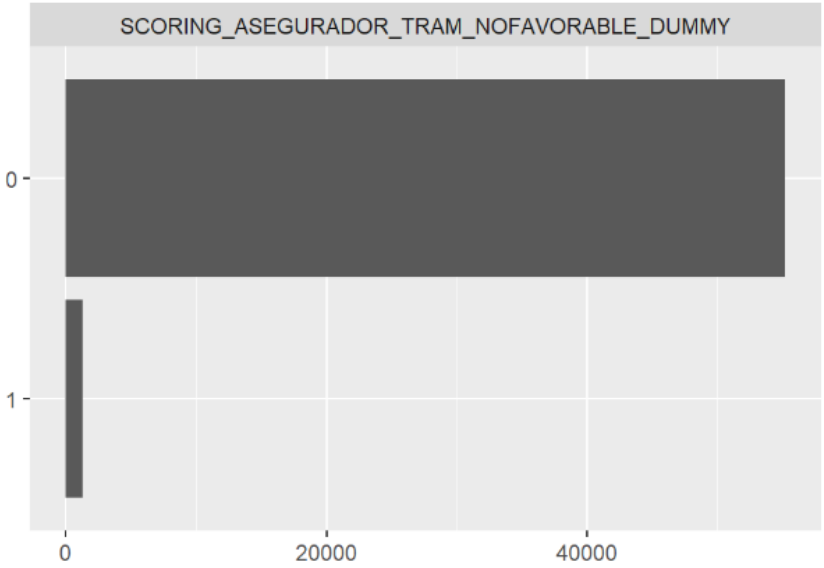


Figura 79. SCORING_ASEGURADOR, K-Means 3. Elaboración Propia.

4.2 Con algoritmo K-Prototypes

Para llevar a cabo la segmentación de la cartera de autos con el algoritmo K-Prototypes es necesario la utilización de la base de datos mixta, esta es, la que contiene datos numéricos y categóricos.

Al igual que con el algoritmo K-Means antes de ejecutar el algoritmo, los datos numéricos han de ser escalados para que todas las variables de esta naturaleza tengan la misma importancia en la segmentación.

El procedimiento para el escalado de los datos es el mismo que para el algoritmo K-Means. Para escalar se utiliza la función “scale”. Con ella, los datos quedan escalados de manera que tengan media 0 y desviación típica 1.

Asimismo, el método K-Prototypes necesita de la elección del número de clústeres que se quieren crear antes de la ejecución. Para ello, se ha elegido, tal y como se hizo con K-Means, el “Método del codo”.

En este caso, el gráfico que se obtiene para el algoritmo K-Prototypes es el siguiente:

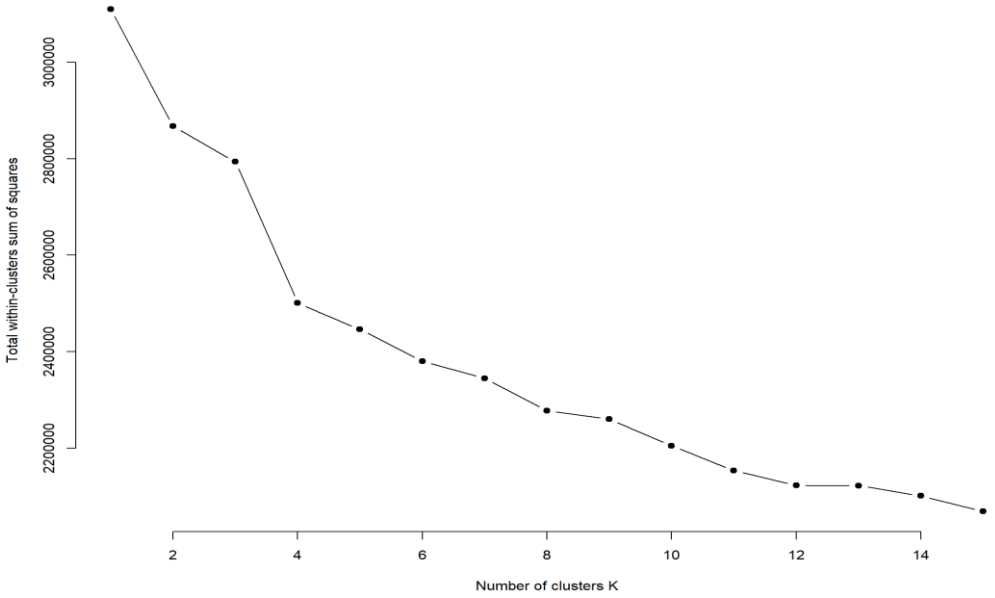


Figura 80. Método del Codo, K-Prototypes. Elaboración Propia.

Una vez obtenido el gráfico del "Método del codo", no se puede concluir ningún valor concreto por lo que se decide ejecutar el algoritmo para crear 3 grupos. Descartamos el valor k= 2 ya que se quiere segmentar en un número mayor de grupos. Para ello, se ha utilizado un algoritmo que necesita de la función "kproto" del paquete "clustMixType" y la distancia euclídea.

Posteriomente, se aplica para cada nivel k, la función "kproto" utilizada anteriomente, con el fin de asignar a cada observación con uno de los grupos creados de la misma manera que con K-Means.

En el método de K-Prototypes no se permite visualizar los grupos creados como ocurría con K-Means por lo que se procede a analizar los diferentes grupos para k =3.

4.2.1 Análisis de los grupos creados con k = 3

De la misma forma que se realizó con el método K-Means, hay que llevar a cabo un análisis de los diferentes grupos que se crean cuando se ejecuta con K-Prototypes

cuando k = 3. Se ha empleado para ello la función "create_report" con el paquete "DataExplorer".

A nivel general, se crean 3 segmentos de los cuales el que mayor número de observaciones contiene es el 3 y el que menos el 1:

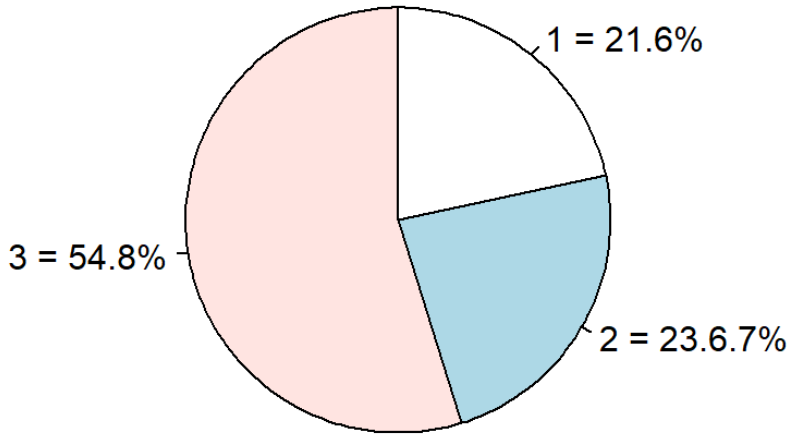


Figura 81. Distribuciones clústeres II. Elaboración Propia.

Las características relevantes de cada grupo son:

-Grupo 1: este grupo se caracteriza porque gran parte de las observaciones corresponden con personas con edades comprendidas entre los 40 y 70 años, con una antigüedad de carnet de entre 10 y 50 años principalmente.

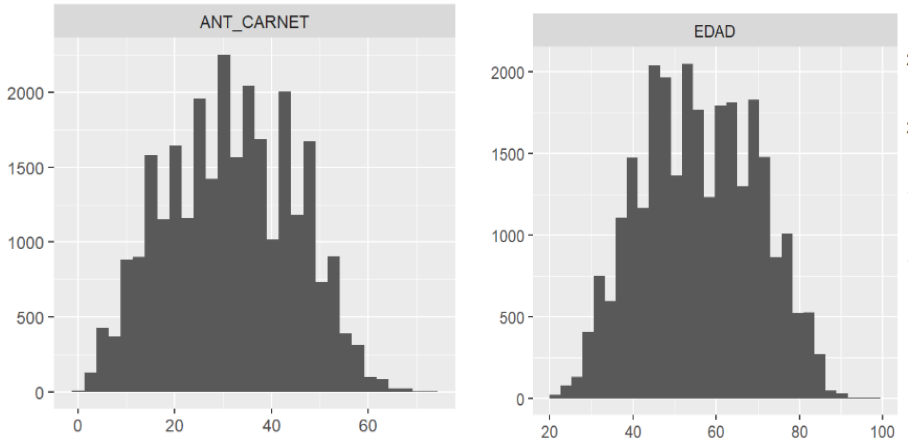


Figura 82. ANT_CARNET y EDAD, KProt 1. Elaboración Propia.

Por otro lado, el grupo 1 destaca por las observaciones a “Todo Riesgo” pues en comparación con los otros dos grupos, el porcentaje que estas representan sobre el total del grupo es mucho mayor que en el resto de grupos. Mismo caso ocurre para la categoría “No Target” que destaca por el mayor volumen de datos que pertenecen a ella.

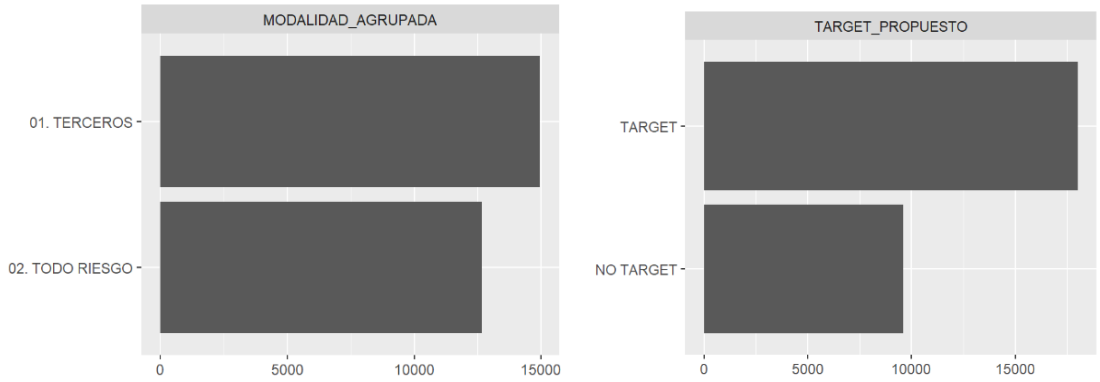


Figura 83. MODALIDAD_AGR y TARGET_PROP, KProt 1. Elaboración Propia.

Por último, las variables que tienen relación con la contratación de opcionales son determinantes para este grupo ya que todas las observaciones tienen al menos contratada una o varias opcionales.

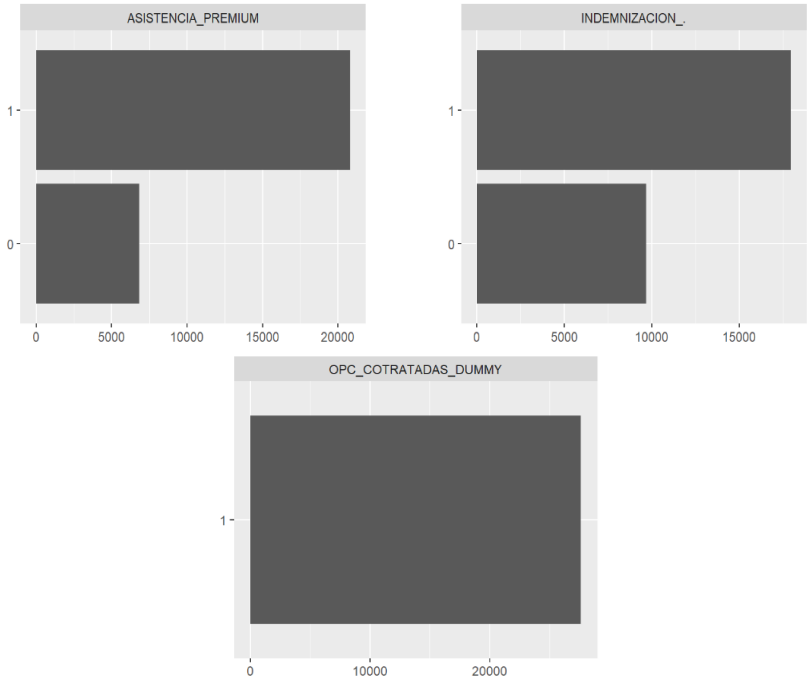


Figura 84. OPCIONALES, KProt 1. Elaboración Propia.

-Grupo 2: al igual que el grupo anterior, una de las variable a tener en cuenta a la hora de analizar este grupo es la edad y la edad de carnet.

Con repecto a la primera, las observaciones tienen mayoritariamente edades entre 20 y 50, años mientras que la antigüedad de carnet corresponde principalmente a un rango entre 0 y 20 años.

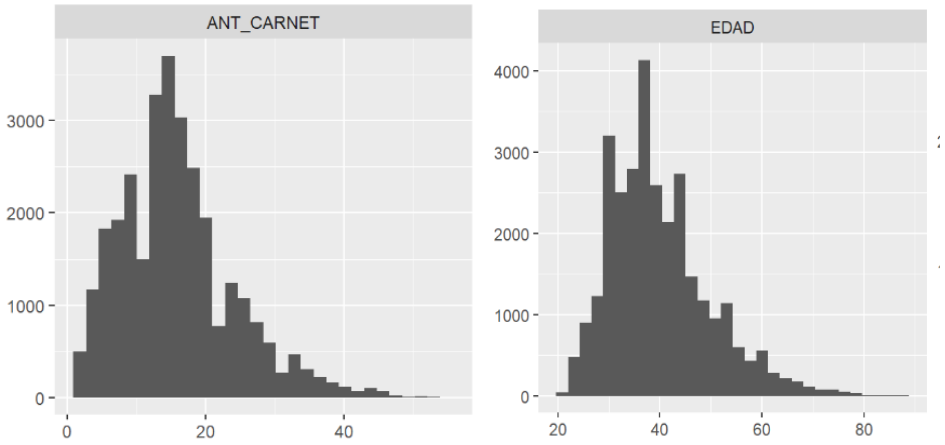


Figura 85. ANT_CARNET y EDAD, KProt 2. Elaboración Propia.

Asimismo, este grupo destaca porque las categorías “Otra compañía” y “Sin garaje” representan mayor porcentaje de datos sobre el total del grupo en comparación con el resto de clústeres.

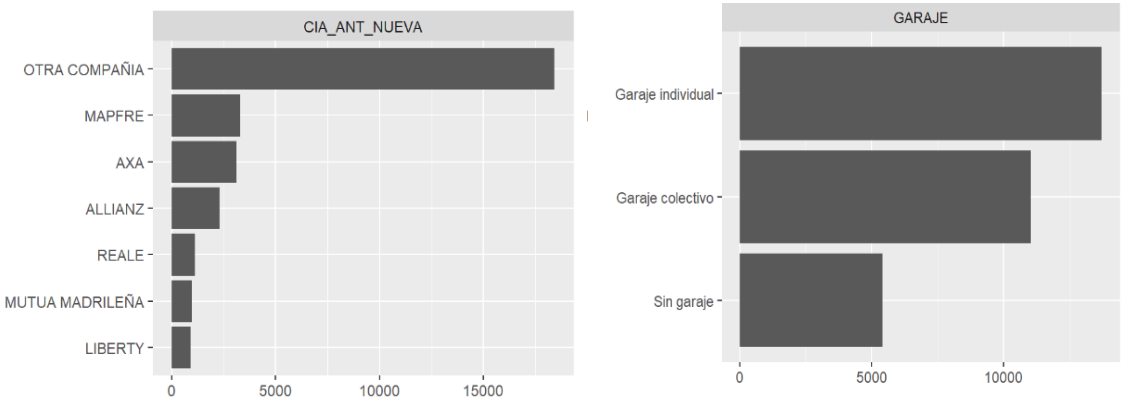


Figura 86. CIA_ANT_NUEVA y GARAJE, KProt 2. Elaboración Propia.

En cuanto al Target, los datos de este grupo pertenecen a la categoría de "No target", contrario a lo que ocurre en el resto de grupos donde principalmente se componen por observaciones que se califican como "Target"

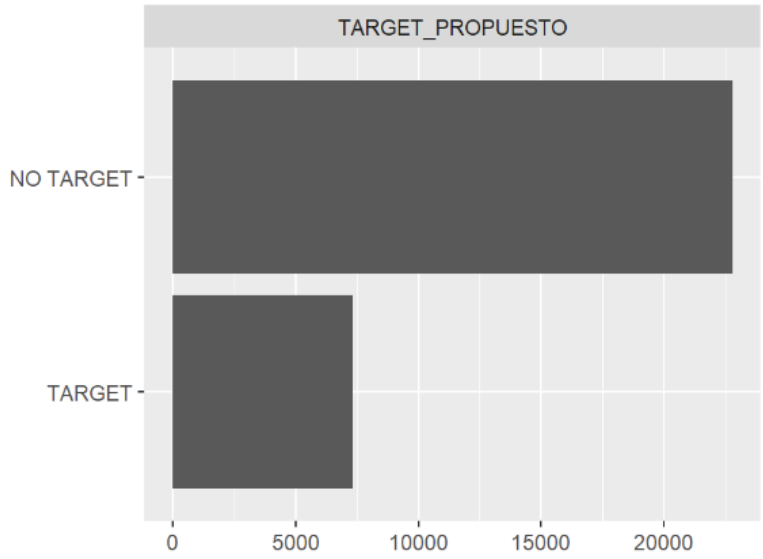


Figura 87. TARGET_PROP, KProt 2. Elaboración Propia.

Finalmente, la variable TERRITORIAL_TRAMO es relevante para la definición del grupo ya que la mayoría de observaciones son catalogadas como "Resto", a diferencia de otros grupos donde la mayor cantidad de datos se encuentran en "A Coruña".

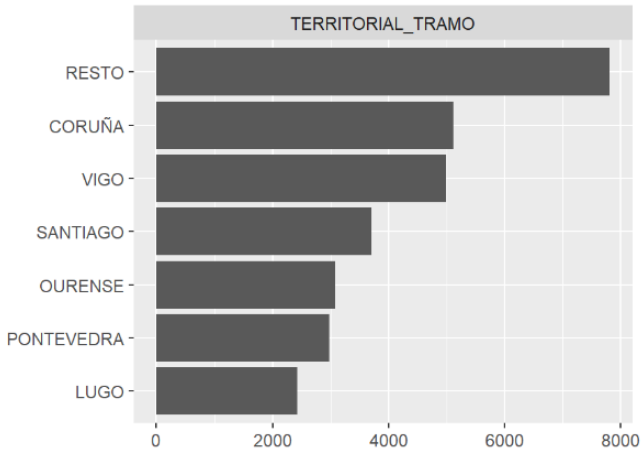


Figura 88. TERRITORIAL_TRAMO, KProt 2. Elaboración Propia.

-Grupo 3: el grupo más mayoritario en cuanto a observaciones se refiere se define porque la variable ANT_CARNET se sitúa en un rango principalmente de entre los 30 a los 50 años y la de EDAD entre los 40 y 80 años.

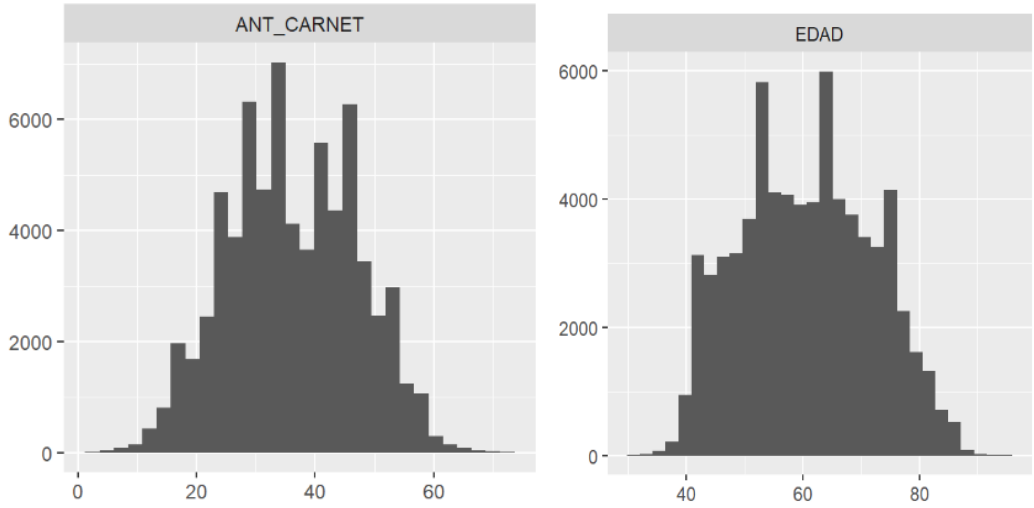


Figura 89. ANT_CARNET y EDAD, KProt 3. Elaboración Propia.

Con respecto a DTO_CAMPAÑA es el grupo donde más se aplica el descuento del 10%. Además, las personas que conforman el clúster tienen en común un Scoring de suscripción muy bajo en comparación con el resto de grupos.

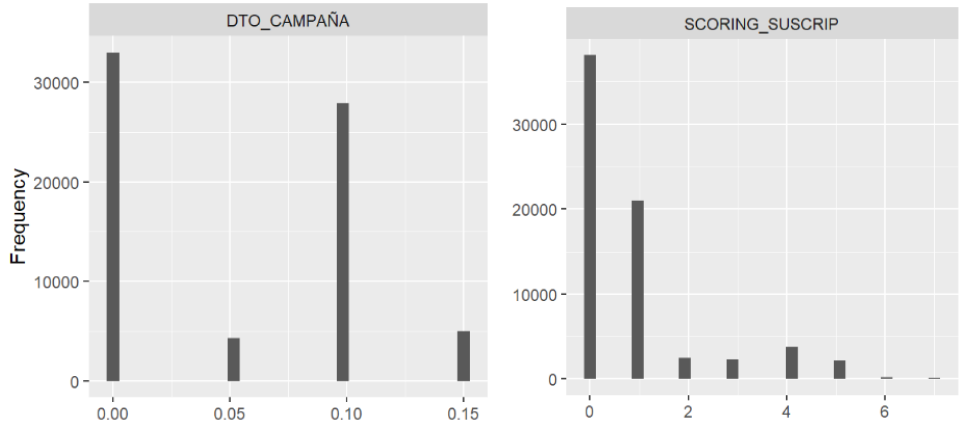


Figura 90. DTO_CAMP y SCOR_SUSC, KProt 3. Elaboración Propia.

También hay que destacar que gran parte del grupo pertenece al tramo favorable, sin haber apenas observaciones en el resto de categorías.

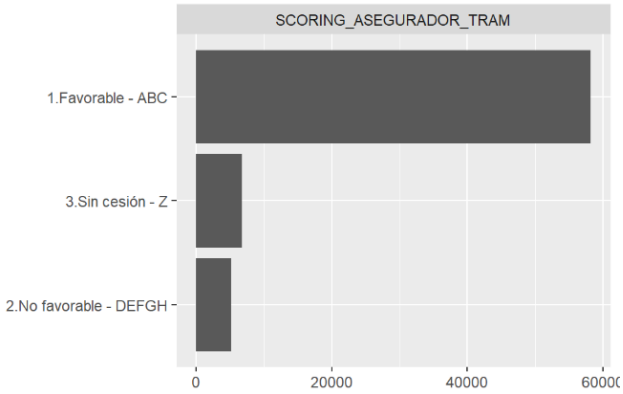


Figura 91. SCOR_ASEG_TRAM, KProt 3. Elaboración Propia.

Por último, cabe destacar que no existen en este clúster observaciones que contengan la variable ASISTENCIA_PREMIUM e INDEMNIZACIÓN_+, por lo que esto a su vez repercute en la variable OPC_CONTRATADAS_DUMMY pues la mayoría es valor 0.

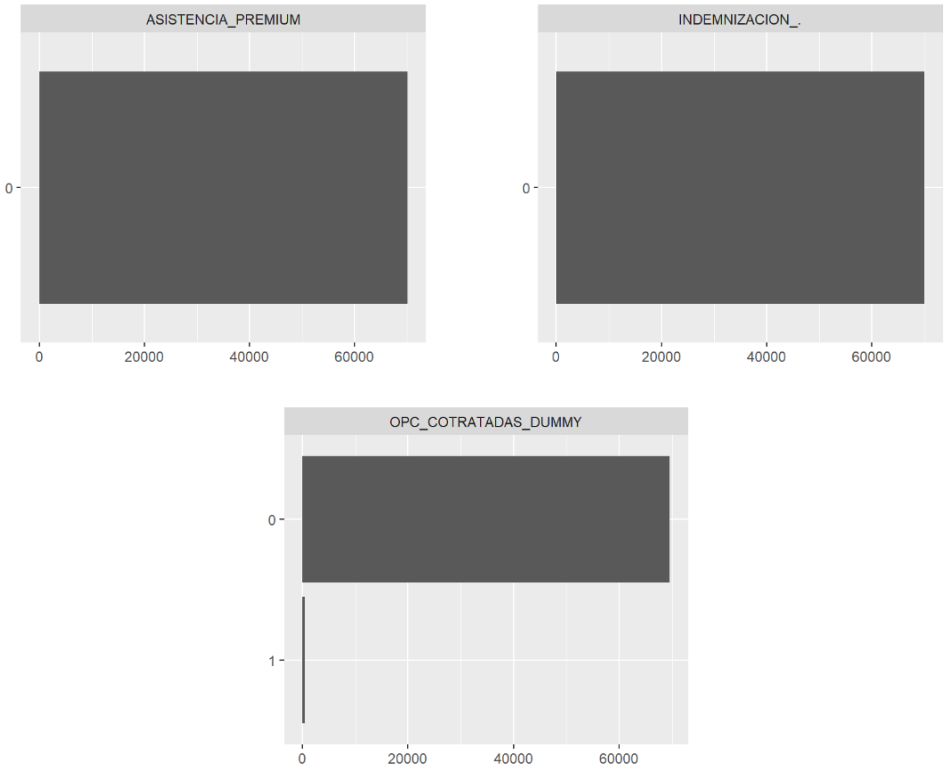


Figura 92. OPCIONALES, KProt 3. Elaboración Propia.

4. CONCLUSIONES

Uno de los aspectos fundamentales a tener en cuenta por las compañías aseguradoras es el entendimiento de los datos con los que trabajan, así como las posibles relaciones que haya entre ellos. El uso de la Inteligencia Artificial y, en especial, el Machine Learning, han agilizado la consecución de dichos objetivos.

Teniendo en cuenta lo anterior, utilizando una base de datos relativa a los asegurados de una compañía de seguros de no vida, en concreto de auto, se han tratado de crear distintos clústeres con el fin de agrupar a clientes similares entre sí empleando para ello técnicas de Machine Learning de aprendizaje no supervisado.

Sin embargo, la base de datos ha requerido de tratamiento y depuración, pues se han podido observar valores faltantes, atípicos y extremos que, en caso de no haberlos tratado, podrían haber distorsionado los resultados obtenidos o, incluso, impedido alcanzar una clasificación coherente.

Por otro lado, también ha sido necesario antes de la ejecución del método de clúster la eliminación de aquellas variables no informativas o aquellas cuyas correlaciones con otras variables eran demasiado elevadas como para incluirlas en el análisis. Gracias a este paso, se ha facilitado la ejecución y el análisis de los grupos creados.

Finalmente, la elección del método de clúster para llevar a cabo la agrupación se ha adecuado a la base de datos utilizada con el fin de obtener resultados fiables. De esta manera, al contemplar dos bases de datos diferentes tras la depuración, se ha optado por el uso del algoritmo K-Means para la base de datos numérica y el algoritmo K-Prototypes para la base de datos mixta.

El resto de algoritmos existentes se han excluido porque la naturaleza de los datos no permitía su uso, la cantidad de datos no permitía un análisis de los grupos sencillo y sólido o no se alcanzaba ninguna conclusión lógica.

Con ambos algoritmos y tras analizar los resultados del Método del codo, el número de clústeres creados han sido tres. Sin embargo, a pesar de que algunas de las características de los grupos eran comunes con los dos métodos, muchas de ellas diferían al utilizar uno u otro. Por tanto, se obtendrán unos resultados diferentes dependiendo del método de clúster elegido.

BIBLIOGRAFÍA

- “Análisis conglomerados cluster. Universidad Autónoma de Madrid”. Estadística.net. Recuperado de: https://www.estadistica.net/Master-Econometria/Analisis_Cluster.pdf?_sm_nck=1

- “Aprendizaje no supervisado”. IBM. Recuperado de: https://www.ibm.com/es-es/topics/unsupervised-learning

- Ayala, J (2020): “Minería de datos. Reducción de dimensionalidad”. RPubs. Recuperado de: https://rpubs.com/JairoAyala/574796

- BBVA. Tecnología (2019): “Machine learning': ¿qué es y cómo funciona?”. Recuperado de: https://www.bbva.com/es/innovacion/machine-learning-que-es-y-como-funciona/

- “Clustering DBSCAN”. Stat Developer. Recuperado de: https://www.statdeveloper.com/clustering-dbscan/

- “Cómo la IA está impactando la industria de los seguros”. Duckcreek. Recuperado de: https://www.duckcreek.es/blog/como-la-ia-impacta-la-industria-de-los-seguros/

- Gutiérrez, R, González, A, Torres, F, & Gallardo, JA. (1994). Técnicas de análisis de datos multivariable: Tratamiento computacional: Universidad de Granada. Recuperado de: https://www.ugr.es/~gallardo/pdf/cluster-3.pdf

- Huang, Z. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery 2, 283–304 (1998).

- International Journal for Research in Applied Science & Engineering Technology, ISSN: 2321-9653; VI June 2020).

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (1st ed.) [PDF]. Springer.

- Kutbay, U. (2018). Partitional clustering. Recent Applications in Data Clustering, 10.

- M. Zamarreño, Virginia (2023): “Inteligencia Artificial para hacer al seguro un poco más humano”. El Economista. Recuperado de:

https://www.eleconomista.es/actualidad/noticias/12180656/03/23/Inteligencia-Artificial-para-hacer-al-seguro-un-poco-mas-humano.html

- MacKay, D. J. (2003). Information theory, inference and learning algorithms. Cambridge university press).

- Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1), 86-97.

- Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS), 42(3), 1-21.

- Villardón, J. L. V. 2007. Introducción al análisis de clúster. Departamento de Estadística, Universidad de Salamanca. 22p.

ANEXOS

```
### CARGAR DATOS DE CARACTERÍSTICAS A R ###
library(readxl)

DATOS_caracteristicas_I = read_excel("DATOS MARZO/DATOS_TFM_MARZO_caracteristicas.xlsx")
DATOS_caracteristicas_FINAL <- as.data.frame(DATOS_caracteristicas_I, stringsAsFactors = TRUE)
str(DATOS_caracteristicas_FINAL)

# ESTABLECER TIPO DE VARIABLE #
DATOS_caracteristicas_FINAL$POLICY_HEADER_CODE=as.character(DATOS_caracteristicas_FINAL$POLICY_HEADER_CODE)
DATOS_caracteristicas_FINAL$POLICY_CODE=as.character(DATOS_caracteristicas_FINAL$POLICY_CODE)
DATOS_caracteristicas_FINAL$NP_CARTERA=as.factor(DATOS_caracteristicas_FINAL$NP_CARTERA)
DATOS_caracteristicas_FINAL$TARIFA_PLANA=as.factor(DATOS_caracteristicas_FINAL$TARIFA_PLANA)
DATOS_caracteristicas_FINAL$MODALIDAD=as.factor(DATOS_caracteristicas_FINAL$MODALIDAD)
DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA=as.factor(DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA)
DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM=as.factor(DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM)
DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM=as.factor(DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM)
DATOS_caracteristicas_FINAL$AV_MECANICA=as.factor(DATOS_caracteristicas_FINAL$AV_MECANICA)
DATOS_caracteristicas_FINAL$`VALORACION_+`=as.factor(DATOS_caracteristicas_FINAL$`VALORACION_+`)
DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`=as.factor(DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`)
DATOS_caracteristicas_FINAL$PDC=as.factor(DATOS_caracteristicas_FINAL$PDC)
DATOS_caracteristicas_FINAL$CIA_ANT=as.factor(DATOS_caracteristicas_FINAL$CIA_ANT)
DATOS_caracteristicas_FINAL$SCORING_SUSCRIP=as.numeric(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP)
DATOS_caracteristicas_FINAL$PROVINCIA=as.factor(DATOS_caracteristicas_FINAL$PROVINCIA)
DATOS_caracteristicas_FINAL$COMBUSTIBLE=as.factor(DATOS_caracteristicas_FINAL$COMBUSTIBLE)
DATOS_caracteristicas_FINAL$FEC_MATRICULACION=as.Date(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)
DATOS_caracteristicas_FINAL$GARAJE=as.factor(DATOS_caracteristicas_FINAL$GARAJE)
DATOS_caracteristicas_FINAL$KMS_ANUALES=as.factor(DATOS_caracteristicas_FINAL$KMS_ANUALES)
DATOS_caracteristicas_FINAL$PUERTAS=as.numeric(DATOS_caracteristicas_FINAL$PUERTAS)
DATOS_caracteristicas_FINAL$USO=as.factor(DATOS_caracteristicas_FINAL$USO)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III)
DATOS_caracteristicas_FINAL$TARGET_PROPUESTO=as.factor(DATOS_caracteristicas_FINAL$TARGET_PROPUESTO)
DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO=as.factor(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO)
DATOS_caracteristicas_FINAL$TERRITORIAL=as.factor(DATOS_caracteristicas_FINAL$TERRITORIAL)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=as.factor(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO)
DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR=as.factor(DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR)

#Ver estructura data frame
str(DATOS_caracteristicas_FINAL)
summary(DATOS_caracteristicas_FINAL)
```

```
### ANÁLISIS EXPLORATORIO DE DATOS (univariante) I ###
# Paquetes a utilizar:
options(scipen = 999)
library(dplyr)
library(ggplot2)
library(readxl)
library(gmodels)
library(Hmisc)
library(ggthemes)

# NP_CARTERA
NP_CARTERA_FREC=CrossTable(DATOS_caracteristicas_FINAL$NP_CARTERA)
max(summary(DATOS_caracteristicas_FINAL$NP_CARTERA))
pie(table(x = DATOS_caracteristicas_FINAL$NP_CARTERA))
# TARIFA_PLANA
TARIFA_PLANA_FREC=CrossTable(DATOS_caracteristicas_FINAL$TARIFA_PLANA)
max(summary(DATOS_caracteristicas_FINAL$TARIFA_PLANA))
pie(table(x = DATOS_caracteristicas_FINAL$TARIFA_PLANA))
# MODALIDAD
MODALIDAD_FREC=CrossTable(DATOS_caracteristicas_FINAL$MODALIDAD)
max(summary(DATOS_caracteristicas_FINAL$MODALIDAD))
pie(table(x = DATOS_caracteristicas_FINAL$MODALIDAD))
# MODALIDAD_AGRUPADA
MODALIDAD_AGRUPADA_FREC=CrossTable(DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA)
max(summary(DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA))
pie(table(x = DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA))
# ASISTENCIA_PREMIUM
ASISTENCIA_PREMIUM_FREC=CrossTable(DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM)
max(summary(DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM))
pie(table(x = DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM))
# PROTECCION_LEGAL_PREMIUM
PROTECCION_LEGAL_PREMIUM_FREC=CrossTable(DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM)
max(summary(DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM))
pie(table(x = DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM))
# AV_MECANICA
AV_MECANICA_FREC=CrossTable(DATOS_caracteristicas_FINAL$AV_MECANICA)
max(summary(DATOS_caracteristicas_FINAL$AV_MECANICA))
pie(table(x = DATOS_caracteristicas_FINAL$AV_MECANICA))
# VALORACION_+
VALORACION_FREC=CrossTable(DATOS_caracteristicas_FINAL$`VALORACION_+`)
max(summary(DATOS_caracteristicas_FINAL$`VALORACION_+`))
pie(table(x = DATOS_caracteristicas_FINAL$`VALORACION_+`))
```

```r
# INDEMNIZACION_+
INDEMNIZACION_FREC=CrossTable(DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`)
max(summary(DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`))
pie(table(x = DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`))
# PDC
PDC_FREC=CrossTable(DATOS_caracteristicas_FINAL$PDC)
max(summary(DATOS_caracteristicas_FINAL$PDC))
pie(table(x = DATOS_caracteristicas_FINAL$PDC))
# CIA_ANT
CIA_ANT_FREC=CrossTable(DATOS_caracteristicas_FINAL$CIA_ANT)
max(summary(DATOS_caracteristicas_FINAL$CIA_ANT))
pie(table(x = DATOS_caracteristicas_FINAL$CIA_ANT))
# PROVINCIA
PROVINCIA_FREC=CrossTable(DATOS_caracteristicas_FINAL$PROVINCIA)
max(summary(DATOS_caracteristicas_FINAL$PROVINCIA))
pie(table(x = DATOS_caracteristicas_FINAL$PROVINCIA))
# COMBUSTIBLE
COMBUSTIBLE_FREC=CrossTable(DATOS_caracteristicas_FINAL$COMBUSTIBLE)
max(summary(DATOS_caracteristicas_FINAL$COMBUSTIBLE))
pie(table(x = DATOS_caracteristicas_FINAL$COMBUSTIBLE))
# GARAJE
GARAJE_FREC=CrossTable(DATOS_caracteristicas_FINAL$GARAJE)
max(summary(DATOS_caracteristicas_FINAL$GARAJE))
pie(table(x = DATOS_caracteristicas_FINAL$GARAJE))
# KMS_ANUALES
KMS_ANUALES_FREC=CrossTable(DATOS_caracteristicas_FINAL$KMS_ANUALES)
max(summary(DATOS_caracteristicas_FINAL$KMS_ANUALES))
pie(table(x = DATOS_caracteristicas_FINAL$KMS_ANUALES))
# USO
USO_FREC=CrossTable(DATOS_caracteristicas_FINAL$USO)
max(summary(DATOS_caracteristicas_FINAL$USO))
pie(table(x = DATOS_caracteristicas_FINAL$USO))
# SCORING_ASEGURADOR
SCORING_ASEGURADOR_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR)
max(summary(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR))
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR))
# SCORING_ASEGURADOR_TRAM
SCORING_ASEGURADOR_TRAM_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM)
max(summary(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM))
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM))
```

```r
# SCORING_ASEGURADOR_TRAM_II
SCORING_ASEGURADOR_TRAM_II_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II)
max(summary(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II))
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II))
# SCORING_ASEGURADOR_TRAM_III
SCORING_ASEGURADOR_TRAM_III_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III)
max(summary(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III))
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III))
# TARGET_PROPUESTO
TARGET_PROPUESTO_FREC=CrossTable(DATOS_caracteristicas_FINAL$TARGET_PROPUESTO)
max(summary(DATOS_caracteristicas_FINAL$TARGET_PROPUESTO))
pie(table(x = DATOS_caracteristicas_FINAL$TARGET_PROPUESTO))
# SEGMENTO_NUEVO_ORDENADO
SEGMENTO_NUEVO_ORDENADO_FREC=CrossTable(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO)
max(summary(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO))
pie(table(x = DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO))
# TERRITORIAL
TERRITORIAL_FREC=CrossTable(DATOS_caracteristicas_FINAL$TERRITORIAL)
max(summary(DATOS_caracteristicas_FINAL$TERRITORIAL))
pie(table(x = DATOS_caracteristicas_FINAL$TERRITORIAL))
# TERRITORIAL_TRAMO
TERRITORIAL_TRAMO_FREC=CrossTable(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO)
max(summary(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO))
pie(table(x = DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO))
#PÓLIZA EN VIGOR
PÓLIZA_VIGOR_FREC=CrossTable(DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR)
max(summary(DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR))
pie(table(x = DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR))
```

```
## Análisis de variables numéricas y fecha##:
# ANIO
ANIO_RESUMEN=summary(DATOS_caracteristicas_FINAL$ANIO)
hist(DATOS_caracteristicas_FINAL$ANIO)
boxplot(DATOS_caracteristicas_FINAL$ANIO)
# DTO_CAMPAÑA
DTO_CAMPAÑA_RESUMEN=summary(DATOS_caracteristicas_FINAL$DTO_CAMPAÑA)
hist(DATOS_caracteristicas_FINAL$DTO_CAMPAÑA)
boxplot(DATOS_caracteristicas_FINAL$DTO_CAMPAÑA)
# NUM_OPC_CONTRATADAS
NUM_OPC_CONTRATADAS_RESUMEN=summary(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)
hist(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)
boxplot(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)
boxplot(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS))
# AÑOS_CIA_ANT
AÑOS_CIA_ANT_RESUMEN=summary(DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT)
hist(DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT)
boxplot(DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT)
# AÑOS_ASEGURADO
AÑOS_ASEGURADO_RESUMEN=summary(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO)
hist(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO)
boxplot(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO)
boxplot(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO))
# COND_OC_27
COND_OC_27_RESUMEN=summary(DATOS_caracteristicas_FINAL$COND_OC_27)
hist(DATOS_caracteristicas_FINAL$COND_OC_27)
boxplot(DATOS_caracteristicas_FINAL$COND_OC_27)
boxplot(DATOS_caracteristicas_FINAL$COND_OC_27)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$COND_OC_27))
# COND_OC_5
COND_OC_5_RESUMEN=summary(DATOS_caracteristicas_FINAL$COND_OC_5)
hist(DATOS_caracteristicas_FINAL$COND_OC_5)
boxplot(DATOS_caracteristicas_FINAL$COND_OC_5)
boxplot(DATOS_caracteristicas_FINAL$COND_OC_5)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$COND_OC_5))
```

```
# COND_OC
COND_OC_RESUMEN=summary(DATOS_caracteristicas_FINAL$COND_OC)
hist(DATOS_caracteristicas_FINAL$COND_OC)
boxplot(DATOS_caracteristicas_FINAL$COND_OC)
boxplot(DATOS_caracteristicas_FINAL$COND_OC)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$COND_OC))
# STDAD_DP
STDAD_DP_RESUMEN=summary(DATOS_caracteristicas_FINAL$STDAD_DP)
hist(DATOS_caracteristicas_FINAL$STDAD_DP)
boxplot(DATOS_caracteristicas_FINAL$STDAD_DP)
boxplot(DATOS_caracteristicas_FINAL$STDAD_DP)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$STDAD_DP))
# STDAD_RC
STDAD_RC_RESUMEN=summary(DATOS_caracteristicas_FINAL$STDAD_RC)
hist(DATOS_caracteristicas_FINAL$STDAD_RC)
boxplot(DATOS_caracteristicas_FINAL$STDAD_RC)
boxplot(DATOS_caracteristicas_FINAL$STDAD_RC)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$STDAD_RC))
# SCORING_SUSCRIP
SCORING_SUSCRIP_RESUMEN=summary(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP)
hist(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP)
boxplot(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP)
boxplot(DATOS_caracteristicas_FINAL$STDAD_RC)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP))
# EDAD
EDAD_RESUMEN=summary(DATOS_caracteristicas_FINAL$EDAD)
hist(DATOS_caracteristicas_FINAL$EDAD)
boxplot(DATOS_caracteristicas_FINAL$EDAD)
# ANT_CARNET
ANT_CARNET_RESUMEN=summary(DATOS_caracteristicas_FINAL$ANT_CARNET)
hist(DATOS_caracteristicas_FINAL$ANT_CARNET)
boxplot(DATOS_caracteristicas_FINAL$ANT_CARNET)
# ANT_VEHI
ANT_VEHI_RESUMEN=summary(DATOS_caracteristicas_FINAL$ANT_VEHI)
hist(DATOS_caracteristicas_FINAL$ANT_VEHI)
boxplot(DATOS_caracteristicas_FINAL$ANT_VEHI)
boxplot(DATOS_caracteristicas_FINAL$ANT_VEHI)$out
length(boxplot(DATOS_caracteristicas_FINAL$ANT_VEHI)$out)
# VALOR_VEHI
VALOR_VEHI_RESUMEN=summary(DATOS_caracteristicas_FINAL$VALOR_VEHI)
hist(DATOS_caracteristicas_FINAL$VALOR_VEHI)
boxplot(DATOS_caracteristicas_FINAL$VALOR_VEHI)
boxplot(DATOS_caracteristicas_FINAL$VALOR_VEHI)$out
```

```
# PUERTAS
PUERTAS_RESUMEN=summary(DATOS_caracteristicas_FINAL$PUERTAS)
hist(DATOS_caracteristicas_FINAL$PUERTAS)
boxplot(DATOS_caracteristicas_FINAL$PUERTAS)
boxplot(DATOS_caracteristicas_FINAL$PUERTAS)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$PUERTAS))
# PLAZAS
PLAZAS_RESUMEN=summary(DATOS_caracteristicas_FINAL$PLAZAS)
hist(DATOS_caracteristicas_FINAL$PLAZAS)
boxplot(DATOS_caracteristicas_FINAL$PLAZAS)
boxplot(DATOS_caracteristicas_FINAL$PLAZAS)$out
CrossTable(as.factor(DATOS_caracteristicas_FINAL$PLAZAS))
# POTENCIA
POTENCIA_RESUMEN=summary(DATOS_caracteristicas_FINAL$POTENCIA)
hist(DATOS_caracteristicas_FINAL$POTENCIA)
boxplot(DATOS_caracteristicas_FINAL$POTENCIA)
boxplot(DATOS_caracteristicas_FINAL$POTENCIA)$out
length(boxplot(DATOS_caracteristicas_FINAL$POTENCIA)$out)
# FEC_MATRICULACION
FEC_MATRICULACION_RESUMEN=summary(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)
boxplot(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)
boxplot(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)$out
length(boxplot(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)$out)
```

```
### VARIABLES INFORMATIVAS ###
library(caret)
near_zero_variance_total_Sin=nearZeroVar(DATOS_caracteristicas_FINAL,uniqueCut = 10, saveMetrics = TRUE)

### DEPURACIÓN DE DATOS ###
# AÑOS_CIA_ANT,AÑOS_ASEGURADO y CIA_ANT
#A: si años_asegurado < años_cia_ant se coge años_cia_ant
for (i in 1:dim(DATOS_caracteristicas_FINAL)[1]) {
  if (DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT[i] > DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO[i]) {
    DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO[i]=DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT[i]
  }
}
#B: si años_asegurado es mayor que 0 y años_cia_ant es 0 se coge los años_asegurado
for (i in 1:dim(DATOS_caracteristicas_FINAL)[1]) {
  if (DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO[i] > 0 & DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT[i]== 0) {
    DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT[i]=DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO[i]
  }
}
# Conductores ocasionales:
for (i in 1:dim(DATOS_caracteristicas_FINAL)[1]) {
  if (DATOS_caracteristicas_FINAL$COND_OC_27[i] >= DATOS_caracteristicas_FINAL$COND_OC[i] | DATOS_caracteristicas_FINAL$COND_OC_5[i]>= DATOS_caracteristicas_FINAL$COND_OC[i]) {
    DATOS_caracteristicas_FINAL$COND_OC[i]=pmax(DATOS_caracteristicas_FINAL$COND_OC_27[i],DATOS_caracteristicas_FINAL$COND_OC_5[i])
  }
}
# Scoring suscripción:
DATOS_caracteristicas_FINAL$SCORING_SUSCRIP[DATOS_caracteristicas_FINAL$NP_CARTERA =="Nueva Producción" & DATOS_caracteristicas_FINAL$SCORING_SUSCRIP>6]=6
DATOS_caracteristicas_FINAL$SCORING_SUSCRIP[DATOS_caracteristicas_FINAL$NP_CARTERA =="Cartera" & DATOS_caracteristicas_FINAL$SCORING_SUSCRIP>7]=6
# Scoring asegurado:
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR[is.na(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR)]='Z'
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM[DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM=='4.Sin dato']='3.Sin cesión - Z'
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II[DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II =='NA']='Z'
# Antigüedad del vehículo
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$ANT_VEHI!=121,]
# Valor del vehículo
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$VALOR_VEHI!=324991.21,]
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$VALOR_VEHI<80000,]
# Potencia
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$POTENCIA!=551,]
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$POTENCIA!=5,]
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$POTENCIA!=8,]
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$POTENCIA!=24,]
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$POTENCIA!=400,]

# Fecha de matriculación
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$FEC_MATRICULACION!="1970-01-01",]
DATOS_caracteristicas_FINAL=DATOS_caracteristicas_FINAL[DATOS_caracteristicas_FINAL$FEC_MATRICULACION>"1988-01-01",]

# Exportar a Excel para comprobar
library(openxlsx)
```

```r
# Importamos el Excel nuevo
library(readxl)
DATOS_caracteristicas = read_excel("excel.xlsx")
DATOS_caracteristicas_FINAL <- as.data.frame(DATOS_caracteristicas, stringsAsFactors = TRUE)
str(DATOS_caracteristicas_FINAL)

DATOS_caracteristicas_FINAL$POLICY_HEADER_CODE=as.character(DATOS_caracteristicas_FINAL$POLICY_HEADER_CODE)
DATOS_caracteristicas_FINAL$POLICY_CODE=as.character(DATOS_caracteristicas_FINAL$POLICY_CODE)
DATOS_caracteristicas_FINAL$NP_CARTERA=as.factor(DATOS_caracteristicas_FINAL$NP_CARTERA)
DATOS_caracteristicas_FINAL$TARIFA_PLANA=as.factor(DATOS_caracteristicas_FINAL$TARIFA_PLANA)
DATOS_caracteristicas_FINAL$MODALIDAD=as.factor(DATOS_caracteristicas_FINAL$MODALIDAD)
DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA=as.factor(DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA)
DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM=as.factor(DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM)
DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM=as.factor(DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM)
DATOS_caracteristicas_FINAL$AV_MECANICA=as.factor(DATOS_caracteristicas_FINAL$AV_MECANICA)
DATOS_caracteristicas_FINAL$`VALORACION_+`=as.factor(DATOS_caracteristicas_FINAL$`VALORACION_+`)
DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`=as.factor(DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`)
DATOS_caracteristicas_FINAL$PDC=as.factor(DATOS_caracteristicas_FINAL$PDC)
DATOS_caracteristicas_FINAL$CIA_ANT=as.factor(DATOS_caracteristicas_FINAL$CIA_ANT)
DATOS_caracteristicas_FINAL$SCORING_SUSCRIP=as.numeric(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP)
DATOS_caracteristicas_FINAL$PROVINCIA=as.factor(DATOS_caracteristicas_FINAL$PROVINCIA)
DATOS_caracteristicas_FINAL$COMBUSTIBLE=as.factor(DATOS_caracteristicas_FINAL$COMBUSTIBLE)
DATOS_caracteristicas_FINAL$FEC_MATRICULACION=as.Date(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)
DATOS_caracteristicas_FINAL$GARAJE=as.factor(DATOS_caracteristicas_FINAL$GARAJE)
DATOS_caracteristicas_FINAL$KMS_ANUALES=as.factor(DATOS_caracteristicas_FINAL$KMS_ANUALES)
DATOS_caracteristicas_FINAL$PUERTAS=as.numeric(DATOS_caracteristicas_FINAL$PUERTAS)
DATOS_caracteristicas_FINAL$USO=as.factor(DATOS_caracteristicas_FINAL$USO)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III=as.factor(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III)
DATOS_caracteristicas_FINAL$TARGET_PROPUESTO=as.factor(DATOS_caracteristicas_FINAL$TARGET_PROPUESTO)
DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO=as.factor(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO)
DATOS_caracteristicas_FINAL$TERRITORIAL=as.factor(DATOS_caracteristicas_FINAL$TERRITORIAL)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=as.factor(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO)
DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR=as.factor(DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR)

summary(DATOS_caracteristicas_FINAL)
str(DATOS_caracteristicas_FINAL)
```

```r
### ANÁLISIS EXPLORATORIO DE DATOS (univariante) II ###
# Paquetes a utilizar:
options(scipen = 999)
library(dplyr)
library(ggplot2)
library(readxl)
library(gmodels)
library(Hmisc)
library(ggthemes)

## Análisis de variables categóricas ##:
# NP_CARTERA
NP_CARTERA_FREC=CrossTable(DATOS_caracteristicas_FINAL$NP_CARTERA)
describe(DATOS_caracteristicas_FINAL$NP_CARTERA)
pie(table(x = DATOS_caracteristicas_FINAL$NP_CARTERA))
# TARIFA_PLANA
TARIFA_PLANA_FREC=CrossTable(DATOS_caracteristicas_FINAL$TARIFA_PLANA)
describe(DATOS_caracteristicas_FINAL$TARIFA_PLANA)
pie(table(x = DATOS_caracteristicas_FINAL$TARIFA_PLANA))
# MODALIDAD
MODALIDAD_FREC=CrossTable(DATOS_caracteristicas_FINAL$MODALIDAD)
describe(DATOS_caracteristicas_FINAL$MODALIDAD)
pie(table(x = DATOS_caracteristicas_FINAL$MODALIDAD))
# MODALIDAD_AGRUPADA
MODALIDAD_AGRUPADA_FREC=CrossTable(DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA)
describe(DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA)
pie(table(x = DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA))
# ASISTENCIA_PREMIUM
ASISTENCIA_PREMIUM_FREC=CrossTable(DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM)
describe(DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM)
pie(table(x = DATOS_caracteristicas_FINAL$ASISTENCIA_PREMIUM))
# PROTECCION_LEGAL_PREMIUM
PROTECCION_LEGAL_PREMIUM_FREC=CrossTable(DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM)
describe(DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM)
pie(table(x = DATOS_caracteristicas_FINAL$PROTECCION_LEGAL_PREMIUM))
# AV_MECANICA
AV_MECANICA_FREC=CrossTable(DATOS_caracteristicas_FINAL$AV_MECANICA)
describe(DATOS_caracteristicas_FINAL$AV_MECANICA)
pie(table(x = DATOS_caracteristicas_FINAL$AV_MECANICA))
# VALORACION_+
VALORACION_FREC=CrossTable(DATOS_caracteristicas_FINAL$`VALORACION_+`)
describe(DATOS_caracteristicas_FINAL$`VALORACION_+`)
pie(table(x = DATOS_caracteristicas_FINAL$`VALORACION_+`))
# INDEMNIZACION_+
INDEMNIZACION_FREC=CrossTable(DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`)
describe(DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`)
pie(table(x = DATOS_caracteristicas_FINAL$`INDEMNIZACION_+`))
```

```
# PDC
PDC_FREC=CrossTable(DATOS_caracteristicas_FINAL$PDC)
describe(DATOS_caracteristicas_FINAL$PDC)
pie(table(x = DATOS_caracteristicas_FINAL$PDC))
# CIA_ANT
CIA_ANT_FREC=CrossTable(DATOS_caracteristicas_FINAL$CIA_ANT)
describe(DATOS_caracteristicas_FINAL$CIA_ANT)
pie(table(x = DATOS_caracteristicas_FINAL$CIA_ANT))
# PROVINCIA
PROVINCIA_FREC=CrossTable(DATOS_caracteristicas_FINAL$PROVINCIA)
describe(DATOS_caracteristicas_FINAL$PROVINCIA)
pie(table(x = DATOS_caracteristicas_FINAL$PROVINCIA))
# COMBUSTIBLE
COMBUSTIBLE_FREC=CrossTable(DATOS_caracteristicas_FINAL$COMBUSTIBLE)
max(summary(DATOS_caracteristicas_FINAL$COMBUSTIBLE))
pie(table(x = DATOS_caracteristicas_FINAL$COMBUSTIBLE))
# GARAJE
GARAJE_FREC=CrossTable(DATOS_caracteristicas_FINAL$GARAJE)
max(summary(DATOS_caracteristicas_FINAL$GARAJE))
pie(table(x = DATOS_caracteristicas_FINAL$GARAJE))
# KMS_ANUALES
KMS_ANUALES_FREC=CrossTable(DATOS_caracteristicas_FINAL$KMS_ANUALES)
describe(DATOS_caracteristicas_FINAL$KMS_ANUALES)
pie(table(x = DATOS_caracteristicas_FINAL$KMS_ANUALES))
# USO
USO_FREC=CrossTable(DATOS_caracteristicas_FINAL$USO)
describe(DATOS_caracteristicas_FINAL$USO)
pie(table(x = DATOS_caracteristicas_FINAL$USO))
# SCORING_ASEGURADOR
SCORING_ASEGURADOR_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR)
describe(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR)
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR))
# SCORING_ASEGURADOR_TRAM
SCORING_ASEGURADOR_TRAM_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM)
describe(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM)
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM), clockwise = TRUE)
# SCORING_ASEGURADOR_TRAM_II
SCORING_ASEGURADOR_TRAM_II_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II)
describe(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II)
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_II))
# SCORING_ASEGURADOR_TRAM_III
SCORING_ASEGURADOR_TRAM_III_FREC=CrossTable(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III)
describe(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III)
pie(table(x = DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_III))
# TARGET_PROPUESTO
TARGET_PROPUESTO_FREC=CrossTable(DATOS_caracteristicas_FINAL$TARGET_PROPUESTO)
describe(DATOS_caracteristicas_FINAL$TARGET_PROPUESTO)
pie(table(x = DATOS_caracteristicas_FINAL$TARGET_PROPUESTO))
```

```
# SEGMENTO_NUEVO_ORDENADO
SEGMENTO_NUEVO_ORDENADO_FREC=CrossTable(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO)
describe(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO)
pie(table(x = DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO))
# TERRITORIAL
TERRITORIAL_FREC=CrossTable(DATOS_caracteristicas_FINAL$TERRITORIAL)
describe(DATOS_caracteristicas_FINAL$TERRITORIAL)
pie(table(x = DATOS_caracteristicas_FINAL$TERRITORIAL))
# TERRITORIAL_TRAMO
TERRITORIAL_TRAMO_FREC=CrossTable(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO)
describe(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO)
pie(table(x = DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO))
#PÓLIZA EN VIGOR
PÓLIZA_VIGOR_FREC=CrossTable(DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR)
describe(DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR)
pie(table(x = DATOS_caracteristicas_FINAL$POLIZA_EN_VIGOR))


## Análisis de variables numéricas y fecha ##:
library(vtable)
st(DATOS_caracteristicas_FINAL[,36:40])
# ANIO
ANIO_RESUMEN=summary(DATOS_caracteristicas_FINAL$ANIO)
hist(DATOS_caracteristicas_FINAL$ANIO)
boxplot(DATOS_caracteristicas_FINAL$ANIO)
# NUM_OPC_CONTRATADAS
NUM_OPC_CONTRATADAS_RESUMEN=summary(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)
describe(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)
hist(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)
boxplot(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS)
# DTO_CAMPAÑA
DTO_CAMPAÑA_RESUMEN=summary(DATOS_caracteristicas_FINAL$DTO_CAMPAÑA)
describe(DATOS_caracteristicas_FINAL$DTO_CAMPAÑA)
hist(DATOS_caracteristicas_FINAL$DTO_CAMPAÑA)
boxplot(DATOS_caracteristicas_FINAL$DTO_CAMPAÑA)
# AÑOS_CIA_ANT
AÑOS_CIA_ANT_RESUMEN=summary(DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT)
describe(DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT)
hist(DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$AÑOS_CIA_ANT, col = "lightblue")
# AÑOS_ASEGURADO
AÑOS_ASEGURADO_RESUMEN=summary(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO)
describe(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO)
hist(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$AÑOS_ASEGURADO, col = "lightblue")
```

```r
# COND_OC_27
COND_OC_27_RESUMEN=summary(DATOS_caracteristicas_FINAL$COND_OC_27)
describe(DATOS_caracteristicas_FINAL$COND_OC_27)
hist(DATOS_caracteristicas_FINAL$COND_OC_27, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$COND_OC_27, col = "lightblue")
# COND_OC_5
COND_OC_5_RESUMEN=summary(DATOS_caracteristicas_FINAL$COND_OC_5)
describe(DATOS_caracteristicas_FINAL$COND_OC_5)
hist(DATOS_caracteristicas_FINAL$COND_OC_5, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$COND_OC_5, col = "lightblue")
# COND_OC
COND_OC_RESUMEN=summary(DATOS_caracteristicas_FINAL$COND_OC)
describe(DATOS_caracteristicas_FINAL$COND_OC)
hist(DATOS_caracteristicas_FINAL$COND_OC, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$COND_OC, col = "lightblue")
# STDAD_DP
summary(DATOS_caracteristicas_FINAL$STDAD_DP)
describe(DATOS_caracteristicas_FINAL$STDAD_DP)
hist(DATOS_caracteristicas_FINAL$STDAD_DP)
boxplot(DATOS_caracteristicas_FINAL$STDAD_DP)
# STDAD_RC
STDAD_RC_RESUMEN=summary(DATOS_caracteristicas_FINAL$STDAD_RC)
describe(DATOS_caracteristicas_FINAL$STDAD_RC)
hist(DATOS_caracteristicas_FINAL$STDAD_RC)
boxplot(DATOS_caracteristicas_FINAL$STDAD_RC)
# SCORING_SUSCRIP
SCORING_SUSCRIP_RESUMEN=summary(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP)
describe(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP)
hist(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$SCORING_SUSCRIP, col = "lightblue")
# EDAD
EDAD_RESUMEN=summary(DATOS_caracteristicas_FINAL$EDAD)
describe(DATOS_caracteristicas_FINAL$EDAD)
hist(DATOS_caracteristicas_FINAL$EDAD)
boxplot(DATOS_caracteristicas_FINAL$EDAD)
# ANT_CARNET
ANT_CARNET_RESUMEN=summary(DATOS_caracteristicas_FINAL$ANT_CARNET)
describe(DATOS_caracteristicas_FINAL$ANT_CARNET)
hist(DATOS_caracteristicas_FINAL$ANT_CARNET)
boxplot(DATOS_caracteristicas_FINAL$ANT_CARNET)
# ANT_VEHI
ANT_VEHI_RESUMEN=summary(DATOS_caracteristicas_FINAL$ANT_VEHI)
describe(DATOS_caracteristicas_FINAL$ANT_VEHI)
hist(DATOS_caracteristicas_FINAL$ANT_VEHI, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$ANT_VEHI, col = "lightblue")
```

```r
# VALOR_VEHI
VALOR_VEHI_RESUMEN=summary(DATOS_caracteristicas_FINAL$VALOR_VEHI)
describe(DATOS_caracteristicas_FINAL$VALOR_VEHI)
hist(DATOS_caracteristicas_FINAL$VALOR_VEHI, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$VALOR_VEHI, col = "lightblue")
# PUERTAS
PUERTAS_RESUMEN=summary(DATOS_caracteristicas_FINAL$PUERTAS)
describe(DATOS_caracteristicas_FINAL$PUERTAS)
hist(DATOS_caracteristicas_FINAL$PUERTAS)
boxplot(DATOS_caracteristicas_FINAL$PUERTAS)
# PLAZAS
PLAZAS_RESUMEN=summary(DATOS_caracteristicas_FINAL$PLAZAS)
describe(DATOS_caracteristicas_FINAL$PLAZAS)
hist(DATOS_caracteristicas_FINAL$PLAZAS)
boxplot(DATOS_caracteristicas_FINAL$PLAZAS)
# POTENCIA
POTENCIA_RESUMEN=summary(DATOS_caracteristicas_FINAL$POTENCIA)
describe(DATOS_caracteristicas_FINAL$POTENCIA)
hist(DATOS_caracteristicas_FINAL$POTENCIA, col = "lightblue")
boxplot(DATOS_caracteristicas_FINAL$POTENCIA, col = "lightblue")
# FEC_MATRICULACION
summary(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)
describe(DATOS_caracteristicas_FINAL$FEC_MATRICULACION)
boxplot(DATOS_caracteristicas_FINAL$FEC_MATRICULACION, col = "lightblue")

### VARIABLES INFORMATIVAS ###
library(caret)
near_zero_variance_total=nearZeroVar(DATOS_caracteristicas_FINAL,uniqueCut = 10, saveMetrics = TRUE)
```

```
### CREACIÓN DE DUMMY ###
#NP_CARTERA:
DATOS_caracteristicas_FINAL$NP_CARTERA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$NP_CARTERA=="Cartera",1,0)
#MODALIDAD:
DATOS_caracteristicas_FINAL=cbind(DATOS_caracteristicas_FINAL, replicate(1,DATOS_caracteristicas_FINAL$MODALIDAD))
colnames(DATOS_caracteristicas_FINAL)[48]="MODALIDAD_NUEVA"
DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA[DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=="01. TE"]="02. TL"
DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=as.factor(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA)
CrossTable(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA)

DATOS_caracteristicas_FINAL$MODALIDAD_TL_DUMMY=ifelse(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=="02. TL",1,0)
DATOS_caracteristicas_FINAL$MODALIDAD_TA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=="03. TA",1,0)
DATOS_caracteristicas_FINAL$MODALIDAD_TRF400_DUMMY=ifelse(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=="04. TRF400",1,0)
DATOS_caracteristicas_FINAL$MODALIDAD_TRF300_DUMMY=ifelse(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=="05. TRF300",1,0)
DATOS_caracteristicas_FINAL$MODALIDAD_TRF200_DUMMY=ifelse(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=="06. TRF200",1,0)
DATOS_caracteristicas_FINAL$MODALIDAD_TRSF_DUMMY=ifelse(DATOS_caracteristicas_FINAL$MODALIDAD_NUEVA=="07. TRSF",1,0)
#MODALIDAD_AGRUPADA:
DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA_TERCEROS_DUMMY=ifelse(DATOS_caracteristicas_FINAL$MODALIDAD_AGRUPADA=="01. TERCEROS",1,0)
#NUM_OPC_CONTRATADAS
DATOS_caracteristicas_FINAL$OPC_CONTRATADAS_DUMMY=ifelse(DATOS_caracteristicas_FINAL$NUM_OPC_CONTRATADAS==0,0,1)
#STDAD_DP Y STDAD_RC
DATOS_caracteristicas_FINAL$SINIESTRALIDAD_DUMMY=ifelse(DATOS_caracteristicas_FINAL$STDAD_DP==0 & DATOS_caracteristicas_FINAL$STDAD_RC==0,0,1)
#CIA_ANT
DATOS_caracteristicas_FINAL=cbind(DATOS_caracteristicas_FINAL, replicate(1,DATOS_caracteristicas_FINAL$CIA_ANT))
colnames(DATOS_caracteristicas_FINAL)[59]="CIA_ANT_NUEVA"
DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA[DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="VERTI"]="MUTUA MADRILEÑA"
DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA[DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="SEGURCAIXA"]="OTRA COMPAÑIA"
DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA[DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="GENERALI"]="OTRA COMPAÑIA"
DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=as.factor(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA)
CrossTable(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA)

DATOS_caracteristicas_FINAL$CIA_ANT_ALLIANZ_DUMMY=ifelse(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="ALLIANZ",1,0)
DATOS_caracteristicas_FINAL$CIA_ANT_AXA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="AXA",1,0)
DATOS_caracteristicas_FINAL$CIA_ANT_LIBERTY_DUMMY=ifelse(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="LIBERTY",1,0)
DATOS_caracteristicas_FINAL$CIA_ANT_MAPFRE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="MAPFRE",1,0)
DATOS_caracteristicas_FINAL$CIA_ANT_MUTUA_MADRILEÑA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="MUTUA MADRILEÑA",1,0)
DATOS_caracteristicas_FINAL$CIA_ANT_OTRA_COMPAÑIA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="OTRA COMPAÑIA",1,0)
DATOS_caracteristicas_FINAL$CIA_ANT_REALE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$CIA_ANT_NUEVA=="REALE",1,0)
```

```
#PROVINCIA
DATOS_caracteristicas_FINAL=cbind(DATOS_caracteristicas_FINAL, replicate(1,DATOS_caracteristicas_FINAL$PROVINCIA))
colnames(DATOS_caracteristicas_FINAL)[67]="PROVINCIA_NUEVA"

DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Araba/Álava"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Asturias"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Ávila"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Bizkaia"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Burgos"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Cantabria"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Gipuzkoa"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="León"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Navarra"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Palencia"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Rioja, La"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Salamanca"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Segovia"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Soria"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Valladolid"]="NORTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Zamora"]="NORTE"

DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Albacete"]="OESTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Badajoz"]="OESTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="cáceres"]="OESTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Ciudad Real"]="OESTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Cuenca"]="OESTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Guadalajara"]="OESTE"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Toledo"]="OESTE"

DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Madrid"]="RESTO"

DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Almería"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Cádiz"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Córdoba"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Granada"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Huelva"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Jaén"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Málaga"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Murcia"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Sevilla"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Melilla"]="RESTO"
```

```
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Alicante/Alacant"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Barcelona"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Castellón/Castelló"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Girona"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Huesca"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Lleida"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Navarra"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Tarragona"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Teruel"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Valencia/València"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Zaragoza"]="RESTO"

DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Balears, Illes"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Palmas, Las"]="RESTO"
DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA[DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Santa Cruz de Tenerife"]="RESTO"

DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=as.factor(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA)
CrossTable(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA)

DATOS_caracteristicas_FINAL$PROVINCIA_NORTE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="NORTE",1,0)
DATOS_caracteristicas_FINAL$PROVINCIA_OESTE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="OESTE",1,0)
DATOS_caracteristicas_FINAL$PROVINCIA_RESTO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="RESTO",1,0)
DATOS_caracteristicas_FINAL$PROVINCIA_CORUNA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Coruña, A",1,0)
DATOS_caracteristicas_FINAL$PROVINCIA_LUGO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Lugo",1,0)
DATOS_caracteristicas_FINAL$PROVINCIA_OURENSE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Ourense",1,0)
DATOS_caracteristicas_FINAL$PROVINCIA_PONTEVEDRA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$PROVINCIA_NUEVA=="Pontevedra",1,0)
#COMBUSTIBLE
DATOS_caracteristicas_FINAL=cbind(DATOS_caracteristicas_FINAL, replicate(1,DATOS_caracteristicas_FINAL$COMBUSTIBLE))
colnames(DATOS_caracteristicas_FINAL)[75]="COMBUSTIBLE_NUEVO"

DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO[DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="B"]="RESTO"
DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO[DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="E"]="RESTO"
DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO[DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="L"]="RESTO"
DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO[DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="P"]="RESTO"
DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO[DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="X"]="RESTO"
DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO[DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="Y"]="RESTO"
DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO[DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="Z"]="RESTO"

DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=as.factor(DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO)
CrossTable(DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO)

DATOS_caracteristicas_FINAL$COMBUSTIBLE_DIESEL_DUMMY=ifelse(DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="D",1,0)
DATOS_caracteristicas_FINAL$COMBUSTIBLE_GASOLINA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="G",1,0)
DATOS_caracteristicas_FINAL$COMBUSTIBLE_RESTO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$COMBUSTIBLE_NUEVO=="RESTO",1,0)
```

```
#GARAJE
DATOS_caracteristicas_FINAL$GARAJE_SIN_DUMMY=ifelse(DATOS_caracteristicas_FINAL$GARAJE=='Sin garaje',1,0)
DATOS_caracteristicas_FINAL$GARAJE_CON_INDIVIDUAL_DUMMY=ifelse(DATOS_caracteristicas_FINAL$GARAJE=='Garaje individual',1,0)
DATOS_caracteristicas_FINAL$GARAJE_CON_COLECTIVO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$GARAJE=='Garaje colectivo con vigilancia' | DATOS_caracteristicas_FINAL$GARAJE=='Garaje colectivo sin vigilancia',1,0)
#KMS_ANUALES
DATOS_caracteristicas_FINAL=cbind(DATOS_caracteristicas_FINAL, replicate(1,DATOS_caracteristicas_FINAL$KMS_ANUALES))
colnames(DATOS_caracteristicas_FINAL)[82]="KMS_ANUALES_NUEVO"

DATOS_caracteristicas_FINAL$KMS_ANUALES_NUEVO[DATOS_caracteristicas_FINAL$KMS_ANUALES_NUEVO=="Hasta 30k"]="Hasta 20k"
DATOS_caracteristicas_FINAL$KMS_ANUALES_NUEVO[DATOS_caracteristicas_FINAL$KMS_ANUALES_NUEVO=="Hasta 40k"]="Hasta 20k"
DATOS_caracteristicas_FINAL$KMS_ANUALES_NUEVO[DATOS_caracteristicas_FINAL$KMS_ANUALES_NUEVO=="Más de 40k"]="Hasta 20k"

DATOS_caracteristicas_FINAL$KMS_ANUALES_5K_DUMMY=ifelse(DATOS_caracteristicas_FINAL$KMS_ANUALES=='Hasta 5k',1,0)
DATOS_caracteristicas_FINAL$KMS_ANUALES_15K_DUMMY=ifelse(DATOS_caracteristicas_FINAL$KMS_ANUALES=='Hasta 15k',1,0)
DATOS_caracteristicas_FINAL$KMS_ANUALES_20K_DUMMY=ifelse(DATOS_caracteristicas_FINAL$KMS_ANUALES=='Hasta 20k',1,0)
#USO
DATOS_caracteristicas_FINAL$USO_OCASIONAL_DUMMY=ifelse(DATOS_caracteristicas_FINAL$USO=='Particular ocasional',1,0)
DATOS_caracteristicas_FINAL$USO_DIARIO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$USO=='Particular diario',1,0)
DATOS_caracteristicas_FINAL$USO_RESTO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$USO=='Particular fin de semana' | DATOS_caracteristicas_FINAL$USO=='Profesional',1,0)
# SCORING_ASEGURADOR_TRAM (ELIMINAMOS EL RESTO DE TRAMOS YA QUE NOS DAN LA MISMA INFORMACION)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_FAVORABLE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM=='1.Favorable - ABC',1,0)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_NOFAVORABLE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM=='2.No favorable - DEFGH',1,0)
DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM_SINCESION_DUMMY=ifelse(DATOS_caracteristicas_FINAL$SCORING_ASEGURADOR_TRAM=='3.Sin cesión - Z',1,0)
# TARGET_PROPUESTO
DATOS_caracteristicas_FINAL$TARGET_PROPUESTO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TARGET_PROPUESTO=='TARGET',1,0)
# SEGMENTO_NUEVO_ORDENADO
DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_S1_DUMMY=ifelse(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO=='01. SEGMENTO 1',1,0)
DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_S2_DUMMY=ifelse(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO=='02. SEGMENTO 2',1,0)
DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_S3_DUMMY=ifelse(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO=='03. SEGMENTO 3',1,0)
DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_S4_DUMMY=ifelse(DATOS_caracteristicas_FINAL$SEGMENTO_NUEVO_ORDENADO=='04. SEGMENTO 4',1,0)
# TERRITORIAL_TRAMO
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO_CORUNA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=='CORUÑA',1,0)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO_LUGO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=='LUGO',1,0)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO_OURENSE_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=='OURENSE',1,0)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO_PONTEVEDRA_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=='PONTEVEDRA',1,0)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO_RESTO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=='RESTO',1,0)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO_SANTIAGO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=='SANTIAGO',1,0)
DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO_VIGO_DUMMY=ifelse(DATOS_caracteristicas_FINAL$TERRITORIAL_TRAMO=='VIGO',1,0)

### VARIABLES INFORMATIVAS ###
library(caret)
near_zero_variance_total_dummy_pre=nearZeroVar(DATOS_caracteristicas_FINAL,uniqueCut = 10, saveMetrics = TRUE)

# Exportar a Excel para comprobar
library(openxlsx)
write.xlsx(DATOS_caracteristicas_FINAL,"excel_final.xlsx")
```

```
### Importamos el Excel nuevo ###
library(readxl)
DATOS_caracteristicas = read_excel("excel_final_kmeans.xlsx")
DATOS_MODELO_FINAL_KMEANS <- as.data.frame(DATOS_caracteristicas, stringsAsFactors = FALSE)
str(DATOS_MODELO_FINAL_KMEANS)

DATOS_MODELO_FINAL_KMEANS$TARIFA_PLANA=as.numeric(DATOS_MODELO_FINAL_KMEANS$TARIFA_PLANA)
DATOS_MODELO_FINAL_KMEANS$ASISTENCIA_PREMIUM=as.numeric(DATOS_MODELO_FINAL_KMEANS$ASISTENCIA_PREMIUM)
DATOS_MODELO_FINAL_KMEANS$`INDEMNIZACION_+`=as.numeric(DATOS_MODELO_FINAL_KMEANS$`INDEMNIZACION_+`)

### VARIABLES INFORMATIVAS ###
library(caret)
near_zero_variance_total_dummy=nearZeroVar(DATOS_MODELO_FINAL_KMEANS,uniqueCut = 5, saveMetrics = TRUE)
```

```
### ANÁLISIS BIVARIANTE II ###
library(readxl)
DATOS_caracteristicas = read_excel("excel_final_kmeans.xlsx")
DATOS_MODELO_FINAL_KMEANS <- as.data.frame(DATOS_caracteristicas, stringsAsFactors = FALSE)
str(DATOS_MODELO_FINAL_KMEANS)

DATOS_MODELO_FINAL_KMEANS$TARIFA_PLANA=as.numeric(DATOS_MODELO_FINAL_KMEANS$TARIFA_PLANA)
DATOS_MODELO_FINAL_KMEANS$ASISTENCIA_PREMIUM=as.numeric(DATOS_MODELO_FINAL_KMEANS$ASISTENCIA_PREMIUM)
DATOS_MODELO_FINAL_KMEANS$`INDEMNIZACION_+`=as.numeric(DATOS_MODELO_FINAL_KMEANS$`INDEMNIZACION_+`)

library(GGally)
library(corrplot)
library(PerformanceAnalytics)
library(ggcorrplot)
library(DescTools)
library(caret)

DATOS_CORR=DATOS_MODELO_FINAL_KMEANS[,-c(1:3)]
str(DATOS_CORR)

CORRELACION=round(cor(DATOS_CORR), 2)
CORRELACION=as.matrix(CORRELACION)

findCorrelation(x = CORRELACION,cutoff = 0.85,verbose = TRUE, names=TRUE)
```

```
### MODELOS DE CLUSTER ###
library(cluster)
library(factoextra)
library(fpc)
library(dbscan)
library(bios2mds)
## K-MEANS ##
# Preparar el set de datos:
#Primer escalamos con estandarización normal:
DATOS_ESCALADOS=scale(DATOS_MODELO_FINAL_KMEANS[,4:dim(DATOS_MODELO_FINAL_KMEANS)[2]])#se escalan los datos (se tipifican/normalizan las variables)
DATOS_ESCALADOS=as.data.frame(DATOS_ESCALADOS)
str(DATOS_ESCALADOS)
#Después escalamos con min max:
library(caret)
DATOS_PARA_ESCALAR=DATOS_MODELO_FINAL[,-c(1:3)]
process <- preProcess(DATOS_PARA_ESCALAR, method=c("range"))
norm_scale <- predict(process, DATOS_PARA_ESCALAR)

#Método del codo
#Para primer escalado:
gc()
set.seed(123)
wss <- (nrow(DATOS_ESCALADOS)-1)*sum(apply(DATOS_ESCALADOS,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(DATOS_ESCALADOS, centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")

#Para segundo escalado:
gc()
set.seed(123)
wss <- (nrow(norm_scale)-1)*sum(apply(norm_scale,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(norm_scale, centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")

# Creación de los clusters:
set.seed(80) #se fija una semilla por el comportamiento aleatorio del k-means
#Cuando k=3
x=kmeans(DATOS_ESCALADOS,centers = 3)
x$cluster
#Cuando k=4
y=kmeans(DATOS_ESCALADOS,centers = 4)
y$cluster
#Cuando k=5
z=kmeans(DATOS_ESCALADOS,centers = 5)
z$cluster
```

```
library("factoextra")

fviz_cluster(x, data = DATOS_ESCALADOS,
             palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
)

#Cuando k=4
y=kmeans(DATOS_ESCALADOS,centers = 4)
y$cluster
y

fviz_cluster(y, data = DATOS_ESCALADOS,
             palette = c("#2E9FDF", "#00AFBB", "#E7B800", "#AE4371"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
)
#Cuando k=5
z=kmeans(DATOS_ESCALADOS,centers = 5)
z$cluster

z

fviz_cluster(z, data = DATOS_ESCALADOS,
             palette = c("#2E9FDF", "#00AFBB", "#E7B800", "#AE4371","#c00000"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
)


DATOS_FINALES_Kmeans_3 <- cbind(DATOS_MODELO_FINAL_KMEANS,x$cluster)
#DATOS_FINALES_Kmeans_4 <- cbind(DATOS_MODELO_FINAL,y$cluster)
#DATOS_FINALES_Kmeans_5 <- cbind(DATOS_MODELO_FINAL,z$cluster)

#########################
options(scipen = 999)
library(dplyr)
library(ggplot2)
library(readxl)
library(gmodels)
library(Hmisc)
library(ggthemes)
DATOS_FINALES_Kmeans_3$`x$cluster`=as.factor(DATOS_FINALES_Kmeans_3$`x$cluster`)
CrossTable(DATOS_FINALES_Kmeans_3$`x$cluster`)
pie(table(DATOS_FINALES_Kmeans_3$`x$cluster`), clockwise = TRUE, labels = c("1 = 22.2%", "2 = 33.7%", "3 = 44.2%"))

#########################
DATOS_FINALES_Kmeans_3_1 <- subset(DATOS_FINALES_Kmeans_3, subset = DATOS_FINALES_Kmeans_3$`x$cluster`==1)
DATOS_FINALES_Kmeans_3_1 <-DATOS_FINALES_Kmeans_3_1[,-c(1,2,3)]

summary(DATOS_FINALES_Kmeans_3_1)
```

```
DATOS_FINALES_Kmeans_3_2 <- subset(DATOS_FINALES_Kmeans_3, subset = DATOS_FINALES_Kmeans_3$`x$cluster`==2)
DATOS_FINALES_Kmeans_3_2 <-DATOS_FINALES_Kmeans_3_2[,-c(1,2,3)]

summary(DATOS_FINALES_Kmeans_3_2)


DATOS_FINALES_Kmeans_3_3 <- subset(DATOS_FINALES_Kmeans_3, subset = DATOS_FINALES_Kmeans_3$`x$cluster`==3)
DATOS_FINALES_Kmeans_3_3 <-DATOS_FINALES_Kmeans_3_3[,-c(1,2,3)]

str(DATOS_FINALES_Kmeans_3_3)

library(DataExplorer)
DataExplorer::create_report(DATOS_FINALES_Kmeans_3_1,output_file = "Resumen primer cluster")
DataExplorer::create_report(DATOS_FINALES_Kmeans_3_2,output_file = "Resumen segundo cluster")
DataExplorer::create_report(DATOS_FINALES_Kmeans_3_3,output_file = "Resumen tercer cluster")
```

```r
### Importamos el Excel nuevo ###
library(readxl)
DATOS_caracteristicas_2 = read_excel("excel_final_kprot.xlsx")
DATOS_MODELO_FINAL_KPROT <- as.data.frame(DATOS_caracteristicas_2, stringsAsFactors = TRUE)
str(DATOS_MODELO_FINAL_KPROT)

DATOS_MODELO_FINAL_KPROT$NP_CARTERA=as.factor(DATOS_MODELO_FINAL_KPROT$NP_CARTERA)
DATOS_MODELO_FINAL_KPROT$TARIFA_PLANA=as.numeric(DATOS_MODELO_FINAL_KPROT$TARIFA_PLANA)
DATOS_MODELO_FINAL_KPROT$MODALIDAD_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$MODALIDAD_NUEVA)
DATOS_MODELO_FINAL_KPROT$MODALIDAD_AGRUPADA=as.factor(DATOS_MODELO_FINAL_KPROT$MODALIDAD_AGRUPADA)
DATOS_MODELO_FINAL_KPROT$ASISTENCIA_PREMIUM=as.numeric(DATOS_MODELO_FINAL_KPROT$ASISTENCIA_PREMIUM)
DATOS_MODELO_FINAL_KPROT$`INDEMNIZACION_+`=as.numeric(DATOS_MODELO_FINAL_KPROT$`INDEMNIZACION_+`)
DATOS_MODELO_FINAL_KPROT$CIA_ANT_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$CIA_ANT_NUEVA)
DATOS_MODELO_FINAL_KPROT$PROVINCIA_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$PROVINCIA_NUEVA)
DATOS_MODELO_FINAL_KPROT$COMBUSTIBLE_NUEVO=as.factor(DATOS_MODELO_FINAL_KPROT$COMBUSTIBLE_NUEVO)
DATOS_MODELO_FINAL_KPROT$GARAJE=as.factor(DATOS_MODELO_FINAL_KPROT$GARAJE)
DATOS_MODELO_FINAL_KPROT$KMS_ANUALES_NUEVO=as.factor(DATOS_MODELO_FINAL_KPROT$KMS_ANUALES_NUEVO)
DATOS_MODELO_FINAL_KPROT$USO_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$USO_NUEVA)
DATOS_MODELO_FINAL_KPROT$SCORING_ASEGURADOR_TRAM=as.factor(DATOS_MODELO_FINAL_KPROT$SCORING_ASEGURADOR_TRAM)
DATOS_MODELO_FINAL_KPROT$TARGET_PROPUESTO=as.factor(DATOS_MODELO_FINAL_KPROT$TARGET_PROPUESTO)
DATOS_MODELO_FINAL_KPROT$TERRITORIAL_TRAMO=as.factor(DATOS_MODELO_FINAL_KPROT$TERRITORIAL_TRAMO)

str(DATOS_MODELO_FINAL_KPROT)

## K-PROTOTYPES ##
library(clustMixType)
#Método del codo
data <- DATOS_MODELO_FINAL_KPROT
  # Elbow Method for finding the optimal number of clusters
  set.seed(123)
# Compute and plot wss for k = 2 to k = 15.
k.max <- 15
data <- na.omit(data) # to remove the rows with NA's
wss <- sapply(1:k.max,
              function(k){kproto(data, k)$tot.withinss})
wss
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")



library(clustMixType)
set.seed(7)
kproto(x = DATOS_caracteristicas_FINAL,k = 4)
kproto(x = DATOS_caracteristicas_FINAL,k = 4)$cluster
```

```
### Importamos el Excel nuevo ###
library(readxl)
DATOS_caracteristicas_2 = read_excel("excel_final_kprot.xlsx")
DATOS_MODELO_FINAL_KPROT <- as.data.frame(DATOS_caracteristicas_2, stringsAsFactors = TRUE)
str(DATOS_MODELO_FINAL_KPROT)

DATOS_MODELO_FINAL_KPROT$NP_CARTERA=as.factor(DATOS_MODELO_FINAL_KPROT$NP_CARTERA)
DATOS_MODELO_FINAL_KPROT$TARIFA_PLANA=as.numeric(DATOS_MODELO_FINAL_KPROT$TARIFA_PLANA)
DATOS_MODELO_FINAL_KPROT$MODALIDAD_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$MODALIDAD_NUEVA)
DATOS_MODELO_FINAL_KPROT$MODALIDAD_AGRUPADA=as.factor(DATOS_MODELO_FINAL_KPROT$MODALIDAD_AGRUPADA)
DATOS_MODELO_FINAL_KPROT$ASISTENCIA_PREMIUM=as.numeric(DATOS_MODELO_FINAL_KPROT$ASISTENCIA_PREMIUM)
DATOS_MODELO_FINAL_KPROT$`INDEMNIZACION_+`=as.numeric(DATOS_MODELO_FINAL_KPROT$`INDEMNIZACION_+`)
DATOS_MODELO_FINAL_KPROT$CIA_ANT_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$CIA_ANT_NUEVA)
DATOS_MODELO_FINAL_KPROT$PROVINCIA_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$PROVINCIA_NUEVA)
DATOS_MODELO_FINAL_KPROT$COMBUSTIBLE_NUEVO=as.factor(DATOS_MODELO_FINAL_KPROT$COMBUSTIBLE_NUEVO)
DATOS_MODELO_FINAL_KPROT$GARAJE=as.factor(DATOS_MODELO_FINAL_KPROT$GARAJE)
DATOS_MODELO_FINAL_KPROT$KMS_ANUALES_NUEVO=as.factor(DATOS_MODELO_FINAL_KPROT$KMS_ANUALES_NUEVO)
DATOS_MODELO_FINAL_KPROT$USO_NUEVA=as.factor(DATOS_MODELO_FINAL_KPROT$USO_NUEVA)
DATOS_MODELO_FINAL_KPROT$SCORING_ASEGURADOR_TRAM=as.factor(DATOS_MODELO_FINAL_KPROT$SCORING_ASEGURADOR_TRAM)
DATOS_MODELO_FINAL_KPROT$TARGET_PROPUESTO=as.factor(DATOS_MODELO_FINAL_KPROT$TARGET_PROPUESTO)
DATOS_MODELO_FINAL_KPROT$TERRITORIAL_TRAMO=as.factor(DATOS_MODELO_FINAL_KPROT$TERRITORIAL_TRAMO)

str(DATOS_MODELO_FINAL_KPROT)


library(readxl)
library(dplyr)
df_7=DATOS_MODELO_FINAL_KPROT
DATOS_caracteristicas_FINAL = df_7
str(df_7)



DATOS_caracteristicas_FINAL <- DATOS_caracteristicas_FINAL %>% mutate(across(where(is.numeric), scale))



DATOS_MODELO_FINAL_KPROT <- as.data.frame(DATOS_caracteristicas_FINAL, stringsAsFactors = TRUE)
str(DATOS_MODELO_FINAL_KPROT)
DATOS_MODELO_FINAL_KPROT

str(DATOS_MODELO_FINAL_KPROT)
```

```
## K-PROTOTYPES ##
library(clustMixType)
#Método del codo
data <- DATOS_MODELO_FINAL_KPROT
  # Elbow Method for finding the optimal number of clusters
  set.seed(123)
# Compute and plot wss for k = 2 to k = 15.
k.max <- 15
data <- na.omit(data) # to remove the rows with NA's
wss <- sapply(1:k.max,
              function(k){kproto(data, k)$tot.withinss})
wss
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")


library(clustMixType)
set.seed(7)
v<-kproto(x = DATOS_MODELO_FINAL_KPROT,k = 3)


DATOS_FINALES_Kproto_3 <- cbind(df_7,v$cluster)

#########################
DATOS_FINALES_Kproto_3_1 <- subset(DATOS_FINALES_Kproto_3, subset = DATOS_FINALES_Kproto_3$`v$cluster`==1)
DATOS_FINALES_Kproto_3_1 <-DATOS_FINALES_Kproto_3_1[,-c(1,2,3)]

summary(DATOS_FINALES_Kproto_3_1)

DATOS_FINALES_Kproto_3_2 <- subset(DATOS_FINALES_Kproto_3, subset = DATOS_FINALES_Kproto_3$`v$cluster`==2)
DATOS_FINALES_Kproto_3_2 <-DATOS_FINALES_Kproto_3_2[,-c(1,2,3)]

summary(DATOS_FINALES_Kproto_3_2)


DATOS_FINALES_Kproto_3_3 <- subset(DATOS_FINALES_Kproto_3, subset = DATOS_FINALES_Kproto_3$`v$cluster`==3)
DATOS_FINALES_Kproto_3_3 <-DATOS_FINALES_Kproto_3_3[,-c(1,2,3)]

summary(DATOS_FINALES_Kproto_3_3)

library(DataExplorer)
DataExplorer::create_report(DATOS_FINALES_Kproto_3_1,output_file = "Resumen primer cluster Kproto")
DataExplorer::create_report(DATOS_FINALES_Kproto_3_2,output_file = "Resumen segundo cluster Kproto")
DataExplorer::create_report(DATOS_FINALES_Kproto_3_3,output_file = "Resumen tercer cluster Kproto")

DATOS_FINALES_Kproto_3$`v$cluster`=as.factor(DATOS_FINALES_Kproto_3$`v$cluster`)
CrossTable(DATOS_FINALES_Kproto_3$`v$cluster`)
pie(table(DATOS_FINALES_Kproto_3$`v$cluster`), clockwise = TRUE, labels = c("1 = 21.6%", "2 = 23.6.7%", "3 = 54.8%"))
```

```
#######################################

library(clustMixType)
set.seed(7)
w<-kproto(x = DATOS_MODELO_FINAL_KPROT,k = 4)


DATOS_FINALES_Kproto_4 <- cbind(df_7,w$cluster)

contar <- 0

Real <-as.factor(df_3$SEGMENTO_NUEVO_ORDENADO)
Real <- as.numeric(Real)

for (i in 1:127816) {
  if (Real[i] == w$cluster[i]) {
    contar <- contar+1
  }
}

Iguales <- contar/127816*100
```