Trabajo Fin de Máster

Gestión y análisis de datos, construcción de modelos predictivos y herramientas de visualización aplicados al Business Intelligence

Verónica Fernández Vidal

Máster en Técnicas Estadísticas

Curso 2022-2023

Propuesta de Trabajo Fin de Máster

Título en galego: Xestión e análise de datos, construción de modelos preditivos e ferramentas de visualización aplicadas ao Business Intelligence

Título en español: Gestión y análisis de datos, construcción de modelos predictivos y herramientas de visualización aplicados al Business Intelligence

English title: Data management and analysis, construction of predictive models and visualization tools applied to Business Intelligence

Modalidad: Modalidad B

Autor/a: Verónica Fernández Vidal, Universidad de Santiago de Compostela

Director/a: José Antonio Vilar Fernández, Universidad de A Coruña

Tutor/a: Abilio Ferreira, Cofrico

Breve resumen del trabajo:

Los objetivos principales del trabajo fin de máster son:

- La gestión y análisis de la base de datos interna de la empresa (Big Data).
- Realización de consultas de bases de datos SQL.
- Construcción de un modelo predictivo para tiempos empleados en la realización de un trabajo.
- Uso de herramientas de visualización aplicados al Business Intelligence.

Recomendaciones:

Otras observaciones:

Don José Antonio Vilar Fernández, Catedrático de la Universidad de A Coruña, don Abilio Ferreira, departamento de informática de Cofrico informan que el Trabajo Fin de Máster titulado

Gestión y análisis de datos, construcción de modelos predictivos y herramientas de visualización aplicados al Business Intelligence

fue realizado bajo su dirección por doña Verónica Fernández Vidal para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En Santiago de Compostela, a 5 de junio de 2023.

VILAR FERNANDEZ JOSE ANTONIO - 32639957J

Firmado digitalmente por VILAR FERNANDEZ JOSE ANTONIO - 32639957J
Fecha: 2023.06.03 22:15:33 +02'00'

El director:
Don José Antonio Vilar Fernández

FERREIRA ARES, ABILIO (FIRMA)

Firmado digitalmente por FERREIRA ARES, ABILIO (FIRMA)
Fecha: 2023.06.04 18:29:39 +02'00'

El tutor:
Don Abilio Ferreira

La autora:
Doña Verónica Fernández Vidal

Declaración responsable. Para dar cumplimiento a la Ley 3/2022, de 24 de febrero, de convivencia universitaria, referente al plagio en el Trabajo Fin de Máster (Artículo 11, Disposición 2978 del BOE núm. 48 de 2022), **la autora declara** que el Trabajo Fin de Máster presentado es un documento original en el que se han tenido en cuenta las siguientes consideraciones relativas al uso de material de apoyo desarrollado por otros/as autores/as:

- Todas las fuentes usadas para la elaboración de este trabajo han sido citadas convenientemente (libros, artículos, apuntes de profesorado, páginas web, programas,...)
- Cualquier contenido copiado o traducido textualmente se ha puesto entre comillas, citando su procedencia.
- Se ha hecho constar explícitamente cuando un capítulo, sección, demostración,...sea una adaptación casi literal de alguna fuente existente.

Y, acepta que, si se demostrara lo contrario, se le apliquen las medidas disciplinarias que correspondan.

Agradecimientos

En primer lugar, agradecer a Cofrico la oportunidad brindada de realizar con ellos las prácticas para el Trabajo de Fin de Máster. Especialmente, a mi tutor Abilio Ferreira por toda su dedicación día a día y el conocimiento compartido. Me gustaría dar las gracias también a todos mis compañeros por el buen trato recibido.

Por otro lado, agradecer a mi tutor académico José Antonio Vilar Fernández por su toda implicación y su orientación en este proyecto, sin su ayuda el trabajo no sería lo que es.

Finalmente, quería darle las gracias a mis padres, familia y amigos por su paciencia y apoyo durante todos mis años académicos.

Índice general

Resumen xi

Prefacio xiii

1. Herramienta Power BI 1
- 1.1. Análisis exploratorio de la base de datos . . . 1
- 1.2. Esquema de estrella . . . 3
- 1.3. Power BI . . . 5
- 1.4. Ilustración de uso . . . 5

2. Modelos predictivos 11
- 2.1. Exploración preliminar: Análisis Cluster . . . 11
  - 2.1.1. Análisis descriptivo . . . 12
  - 2.1.2. Análisis cluster . . . 13
- 2.2. Modelo predictivo para el tiempo de desplazamiento . . . 17
  - 2.2.1. Análisis descriptivo de las variables . . . 17
  - 2.2.2. Construcción del modelo . . . 20
  - 2.2.3. Validación del modelo . . . 22
  - 2.2.4. Predicciones . . . 25
- 2.3. Modelo predictivo para el tiempo de ejecución . . . 26
  - 2.3.1. Análisis descriptivo de las variables . . . 26
  - 2.3.2. Construcción del modelo . . . 32
  - 2.3.3. Validación del modelo . . . 34
  - 2.3.4. Predicciones . . . 37

3. Modelos en producción 39
- 3.1. API . . . 39
  - 3.1.1. Creación de una API con *plumber* . . . 39
- 3.2. Docker . . . 42
  - 3.2.1. Despliege de las APIs dentro de contenedores Docker . . . 42
  - 3.2.2. Docker Compose . . . 44

4. Conclusiones 45

A. Código 47

Bibliografía 55

Resumen

Resumen en español

Hoy en día, disponer de herramientas adecuadas para poder analizar la gran cantidad de información almacenada en las bases de datos, es uno de los principales problemas presentes en las empresas. Esto les permite extraer conclusiones útiles que les ayudan en la toma de decisiones. Este proyecto fue propuesto por Cofrico con el fin de emplear la información disponible en su base de datos con dos objetivos bien diferenciados. Por un lado, obtener una visión clara de la calidad de sus servicios en función de una serie de indicadores que hacen referencia a mediciones de tiempos. Por este motivo, se hará uso de la herramienta de visualización aplicada al Business Intelligence: Power BI. Por otro, disponer de una herramienta que le permita obtener las predicciones para los tiempos de desplazamiento y los de ejecución de aquellos servicios donde Cofrico tiene particular interés. Con este fin, se construyen dos modelos de regresión con efectos aleatorios y se ponen en producción mediante la creación de APIs y el uso de contenedores Docker.

English abstract

Nowadays, one of the main problems found in businesses is having adequate tools to analyze the large amount of information stored in databases. This allows them to draw useful conclusions that help them in decision making. This project was proposed by Cofrico in order to use the information avaliable in its database towards two very well differentiated objectives. On the one hand, obtaining a clear image of the quality of its services, based on a series of indicators time measurement related. For this reason, the visualization tool Power BI, applied to Business Intelligence will be used. On the other, having a tool that allows to obtain predictions of movement and execution times for the services Cofrico is more interested in. To this end, two random effects regression models are built and put into production by creating APIs and using Docker containers.

Prefacio

El análisis Big Data y el uso de aplicaciones dedicadas al Business Intelligence son herramientas que están creciendo cada vez más en el ámbito empresarial. La recopilación de grandes cantidades de datos y la necesidad de aprovecharlos para poder obtener conclusiones útiles, es la base en la que radica la importancia de estas herramientas. Un uso adecuado de estas técnicas permite a las empresas, entre otros sectores, tomar mejores decisiones y aprender de los datos existentes para mejorar en el futuro. De esta forma, pueden tener un mejor control y desempeño de sus servicios.

En la presente memoria se abordarán distintos problemas relacionados con estas técnicas. Este trabajo fue propuesto con una doble motivación. Por un lado, obtener una visión clara de la calidad de los servicios de la empresa. Por otro, disponer de una herramienta de ayuda para diseñar una planificación lo más idónea posible, permitiendo una distribución eficiente del tiempo disponible para desenvolver sus servicios.

Este proyecto de Trabajo Fin de Máster fue presentado por Cofrico, empresa que nace en 1985 en Burela (Lugo), con el objetivo de poder ofrecer sus servicios:

- Refrigeración y climatización a nivel industrial y comercial.
- Mantenimiento industrial: Predictivo, preventivo y correctivo.

Su factor diferencial radica en aplicar la tecnología en la industria mediante la refrigeración ecológica con CO2 y NH3. La empresa cuenta con distintas certificaciones y premios: ISO 9001 y Pyme Innovadora.

En Cofrico, se colabora con la ingeniería en el detalle del proyecto y definición de la instalación. Se instala y ejecuta la solución óptima definida, asegurando una máxima profesionalidad y experiencia del equipo instalador. Además, se responsabilizan del buen funcionamiento e incremento de la vida útil de las instalaciones. Sus principales valores son:

- Innovación: 2 patentes de ahorro de energía, sistema propio de eficiencia energética y mantenimiento 4.0, sello Pyme Innovadora e inversión en I+D+i.
- Eficiencia energética: Tiene un sistema propio Gradhoc para la monitorización y control de las instalaciones, mantenimiento predictivo y ahorro energético.
- Compromiso con el medio ambiente: Son pioneros en utilizar refrigeración con CO2, reutilizando este gas presente en la naturaleza.
- Calidad de servicio: Mejoran continuamente sus procesos productivos para alcanzar objetivos de calidad y satisfacción cada vez más exigentes.

Actualmente, Grupo Cofrico se estructura en un edificio central, diez delegaciones y ocho puntos de servicio en toda España, con un equipo de más de 200 profesionales con formación y acreditada

experiencia en cada una de sus áreas de especialización.

En base a la doble motivación antes mencionada, el trabajo se plantea para alcanzar dos objetivos principales. En primer lugar, el empleo de la herramienta Power BI [20, 25] para que la empresa disponga de una vía útil para analizar, de forma clara y sistemática, la calidad de sus servicios. Más específicamente, el interés de Cofrico radicaba en poder observar que está ocurriendo en la realidad con una serie de indicadores relacionados con mediciones de tiempos en el servicio. Asimismo, para poder acceder a la base de datos, se habrán de emplear consultas SQL.

Otro objetivo importante del trabajo es proporcionar a la empresa de una herramienta que le permita disponer de predicciones para los tiempos de desplazamiento y los de ejecución de aquellos servicios donde la empresa tiene particular interés. Esto permitirá a Cofrico desenvolver una planificación más idónea de sus servicios y mantener un mejor control de los tiempos empleados. Con este fin, se ajustan dos modelos de regresión, para los cuales inicialmente se realiza una exploración preliminar mediante análisis cluster [23, 38, 50]. Finalmente, por la naturaleza de la información, se consideran modelos de regresión con efectos aleatorios [3, 15, 40] para la construcción de los modelos predictivos finales.

A la luz de estos objetivos, la memoria se estructura en cuatro capítulos. En el primer capítulo, se realiza un análisis exploratorio previo de la base de datos de la empresa. No obstante, este capítulo está dedicado fundamentalmente a la herramienta Power BI. En particular, se muestra un ejemplo de uso a partir de uno de los informes realizados por requerimiento de la empresa. En el segundo capítulo, se desarrollan los modelos de regresión de efectos aleatorios propuestos para obtener las predicciones del tiempo de desplazamiento y de ejecución. Se documenta la construcción de ambos modelos predictivos y su validación. En el tercer capítulo, se presentan las herramientas necesarias para poner los modelos creados en producción. De esta forma se faculta a la empresa para realizar un uso más eficiente de los mismo. Para cada modelo se crea una API, la cual será posteriormente montada en un contenedor Docker. Finalmente, se incluye un cuarto capítulo dedicado a establecer las principales conclusiones del trabajo desarrollado.

Capítulo 1

Herramienta de visualización aplicada al Business Intelligence: Power BI

En este capítulo se aborda la creación de una herramienta diseñada para que Cofrico pueda examinar y hacer un seguimiento de la calidad de sus servicios. Por consiguiente, se centra en el desarrollo de la herramienta Power BI, la cual resulta atrayente para la empresa debido a su utilidad a la hora de mostrar, de forma clara y precisa, que está ocurriendo con los datos. El interés de la empresa radica en analizar una serie de indicadores relacionados con las mediciones de distintos tiempos empleados en los servicios. Para poder facilitar el acceso a los datos se creará previamente una base de datos local siguiendo el formato de esquema de estrella.

El capítulo se estructura de forma que, en primer lugar, se realiza un análisis exploratorio de la base de datos. El objetivo final de este análisis es conocer las variables más interesantes a la hora de abordar los problemas planteados en este proyecto. Posteriormente, se describe qué se entiende por un esquema de estrella y se documenta la motivación de su construcción. Se introduce la herramienta Power BI y, finalmente, se muestra un ejemplo de uso.

1.1. Análisis exploratorio de la base de datos

En esta sección se realiza un pequeño análisis exploratorio de la base de datos de la empresa. Por un lado, se comentan las variables más destacadas a la hora de crear los distintos reportes en Power BI. Por otro, se introducen las variables, a priori más relevantes, empleadas en la construcción de los dos modelos predictivos.

La variable Cod_solicitud engloba los códigos que representan a las distintas solicitudes, las cuales hacen referencia a una petición del cliente para realizar una cierta reparación, mantenimiento, etc. La creación de las mismas es el punto de inicio del proceso de asistencia de mantenimiento de la empresa. Por otro lado, la variable Tipo especifica una clasificación de las solicitudes y está constituida por 7 categorías: preventivo (001), mantenimiento (002), obras (003), garantía (004), reclamación (005), gestión interna (006) y mascarillas y otros_covid (007). En este primer capítulo, se documenta la realización de una serie de informes basados en diferentes medidas de tiempos empleados en los siguientes servicios de mantenimiento:

- Mantenimiento preventivo: Tipo de mantenimiento que sigue una planificación según las revisio-

nes de preventivo que tenga contratadas el cliente. Tiene como finalidad evitar o prevenir fallos en las instalaciones o los equipos. Estas solicitudes están asociadas a un contrato.

- Mantenimiento correctivo: Mantenimiento que corrige los errores que se dan en un equipo o instalación. Es necesaria la intervención de un técnico para resolver la avería y devolverla a su función inicial. Además, estás solicitudes no están asociadas a un pedido de venta, contrato o proyecto.

- Reparaciones bajo presupuesto: La presencia de posibles anomalías en las instalaciones durante la ejecución de los servicios de mantenimiento correctivo o preventivo pueden requerir la elaboración de un presupuesto. Normalmente, estas son reparaciones de cierta envergadura que el cliente requiere de un presupuesto para tomar la decisión de acometer la reparación o no. Son solicitudes asociadas a un pedido de venta clasificadas como obras.

- Presupuestos Plataformas de Ofertas ('POF'): Son presupuestos vinculados a los mantenimientos correctivos o preventivos. No es necesaria la intervención del Departamento de Oficina Técnica dado que estos presupuestos no exigen diseño.

Además, los distintos indicadores de tiempos fueron analizados en función de las siguientes variables categóricas:

- Planta: Representa a cada una de las sedes de la empresa. Se mencionó en el prefacio que Cofrico cuenta con 10 delegaciones. No obstante, esta variable está constituida por 14 códigos distintos, dado que también tiene en cuenta ciertos puntos de servicio.

- Cod_centro: Es cada una de las unidades/locales que pertenecen a los clientes. Técnicamente hablando, hace referencia a una dirección de entrega. Está constituida por más de 6000 códigos distintos.

- Condiciones_tarifa: Indica el tipo de cada instalación técnica. Está constituida por 6 códigos distintos: Frío comercial (01), frío industrial (02), clima (03), hostelería (04), multioficio (05) y ninguna (99).

Por otro lado, se requiere la construcción de un modelo predictivo para el tiempo de desplazamiento, tiempo empleado por un técnico para llegar al centro que solicitó asistencia, y para el de ejecución, tiempo invertido en la reparación de una cierta avería. En ambos casos, cada registro hace referencia a una solicitud, ya que la idea es poder disponer de una estimación del tiempo necesario para dar respuesta a las mismas. Asimismo, como se detallará en el Capítulo 2, en un principio se dispone de las variables cualitativas: Planta, Cod_centro, Día_semana (representa el día de la semana en que se recibió la solicitud), Mes, Técnico y Lin_negocio. Técnico es una variable nominal constituida por más de 100 códigos diferentes y Lin_negocio representa un eje analítico en la base de datos de la empresa y se usa para analizar analíticamente las 30 líneas de negocio existentes. Además, la mayoría de las categorías de esta variable constituyen una combinación de las variables Tipo y Condiciones_tarifa.

Cofrico está particularmente interesado en obtener predicciones para los tiempo de desplazamiento y de ejecución de las solicitudes de mantenimiento correctivo (Tipo = '002'). La construcción de los respectivos modelos se detallará en el Capítulo 2. Para obtener la predicción del tiempo de desplazamiento se emplean las variables Cod_centro y km, la cual representa los kilómetros recorridos desde el origen del que parte el técnico hasta el respectivo centro. Asimismo, para el modelo predictivo del tiempo de ejecución, se tienen en cuenta las variables cualitativas:

- Condiciones_tarifa: En este caso solo es relevante considerar: Frío comercial (01), frío industrial (02), clima (03) y hostelería (04). Además, dentro de las solicitudes de mantenimiento correctivo, esta variable hace referencia a lo mismo que Lin_negocio.
- Num_tecnicos: Variable categórica que indica el número de técnicos que acuden a la solicitud y se divide en '1', '2' y '2+'.
- Clasificación_centro_tecnico y Clasificación_tarifa_tecnico: Dos variables que permiten una clasificación de los técnicos. La primera lo hace en función del conocimiento que tiene acerca del centro y la segunda en función de la condición de tarifa. Ambas variables son categóricas y se dividen en 'Alto', 'Bajo' y 'Medio'. Estas variables fueron calculadas en función del número de veces que cada técnico acudía a un determinado centro o condición de tarifa.
- Equipo: Variable que refleja una clasificación de los distintos activos que se pueden intervenir. Se tienen datos de 96 equipos distintos, muchos de ellos con un número bastante pequeño de observaciones. Esta variable será representada en la actual base de datos por Tipo_Equipo.
- Tipo_equipo: Hace referencia a una clasificación más general de los equipos. Se tienen datos de un total de 12 categorías distintas. Esta variable será representada en la actual base de datos por Tipo_Servicio.
- Tipo_averia: Clasificación de las posibles averías que pueden tener los activos. Se tienen datos de 49 categorías, las cuales no presentan un reparto equitativo de las solicitudes, llegando a existir grupos una única observación.

Hasta el momento, estas últimas tres variables no eran obligatorias en las solicitudes disponibles en la base de datos actual. Por este motivo, tras la introducción de las mismas en la construcción de este modelo predictivo, se emplean las solicitudes existentes en la anterior base de datos de Cofrico. No obstante, se realizan las modificaciones necesarias en las distintas variables para poder emplearlas de acuerdo a su clasificación actual.

1.2. Esquema de estrella

Un esquema de estrella [34, 42] es un tipo de esquema de base de datos, ampliamente adoptado por los almacenes de datos relacionales, que consiste en una tabla central de hechos de gran tamaño y una o más tablas de dimensiones a su alrededor. Las tablas de hechos pueden almacenar observaciones o eventos. Estas contienen columnas de clave de dimensiones relacionadas con las tablas de dimensiones y columnas de medidas numéricas. Las columnas de clave de dimensiones determinan la dimensionalidad de una tabla de hechos, mientras que los valores de clave de dimensiones determinan la granularidad de una tabla de hechos. Por otro lado, las tablas de dimensiones describen entidades empresariales. Una tabla de dimensiones contiene una columna (o columnas) de clave que actúa como identificador único y columnas descriptivas. Normalmente, las tablas de dimensiones contienen un número relativamente pequeño de filas. Por el contrario, las tablas de hechos pueden contener un gran número de filas y seguir creciendo con el tiempo.

Una de las ventajas de emplear un modelado en esquema de estrella es que se logran mejores tiempos de respuesta al realizar las diferentes consultas y es más sencillo de comprender la estructura y los alcances del mismo. No obstante, uno de los principales problemas es la rigidez de su diseño que no permite el descubrimiento de nuevas relaciones.

La relevancia de estos esquemas de estrella a la hora de desarrollar modelos en Power BI, motivó la creación de una base de datos siguiendo este diseño a partir de la ya existente en la empresa. La tabla

de hechos contiene columnas de clave de dimensiones y columnas de medidas numéricas relacionadas con tiempos de servicio. Además, se crearon 5 tablas de dimensiones: Plantas, Centros, Condiciones de tarifa, meses y días. De esta forma, se puede disponer de los datos necesarios para la realización de los distintos informes requeridos por la empresa de forma rápida y cómoda. Asimismo, nos permite disponer de los datos necesarios para la construcción de los modelos predictivos, que se documentarán en el Capítulo 2, empleando consultas mucho más simples.

En la Figura 1.1 se pueden apreciar las distintas tablas que conforman la base de datos con diseño de esquema de estrella. Se observa la tabla de hechos, situada en el centro, donde se alojan las distintas medidas numéricas relacionadas con tiempos de servicio necesarias para la realización de los reportes, las variables relevantes en la construcción de los modelos predictivos y todas las claves primarias de las distintas tablas de dimensiones situadas alrededor de esta.

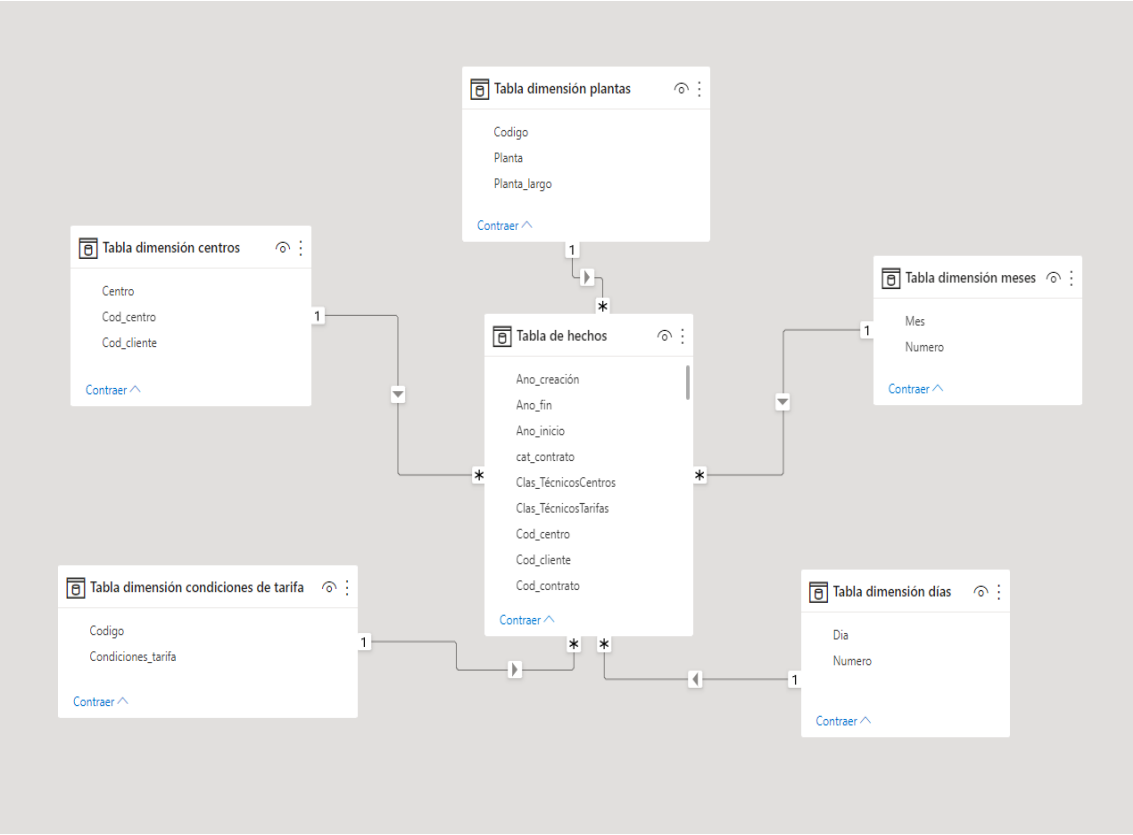


Figura 1.1: Base de datos con modelado en esquema de estrella

Para la construcción de esta base de datos se emplearon consultas SQL desarrolladas en 'Azure Data Studio' [35], una herramienta de base de datos multiplataforma para profesionales de datos que usan plataformas de datos locales y en la nube en Windows, macOS y Linux.

1.3. Power BI

Power BI [20, 25] es una potente herramienta de Business Intelligence creada por Microsoft, que sirve para la gestión y el análisis de datos necesarios en cada momento, proporcionando un informe con visualizaciones interactivas para los usuarios finales. La finalidad principal de esta herramienta es el diseño de un reporte con el que se puede optimizar el proceso de toma de decisiones en los negocios, es decir, proporciona una ventaja competitiva que permite aumentar la productividad sin necesidad de programación. Además, Power BI permite integrar un gran volumen de datos y conectarlos desde distintos orígenes. Todo esto hace que sea una de las herramientas de Business Intelligence más empleada entre una inmensa cantidad de grandes empresas.

En particular, Cofrico comenzó con el uso de esta herramienta debido a que alguno de sus trabajadores ya empleaban su versión gratuita Power BI desktop. Además, su facilidad de uso y su apoyo en la toma de decisiones hizo de ella una herramienta clave en esta empresa.

El interés de la empresa radica en analizar una serie de medidas relacionadas con distintos tiempos empleados en sus servicios. Con este fin, Cofrico proporcionó una lista de indicadores que miden la eficiencia del servicio. Esta lista estaba dividida en mantenimiento correctivo, reparaciones bajo presupuesto, mantenimiento preventivo y elaboración de presupuestos 'POF'. Dentro de cada uno de ellos se encontraban los distintos parámetros a controlar para obtener los tiempos que se querían analizar. Además, los indicadores estaban clasificados de acuerdo a su nivel de prioridad, alta o no. Por consiguiente, el objetivo final es mostrarle a Cofrico la calidad de sus servicios empleando la plataforma de business intelligence, Power BI. En definitiva, la idea principal de la realización de estos reportes no es otra que mostrar a la empresa qué está ocurriendo en la realidad con los datos indicados.

Por otro lado, como ya se comentó en la sección anterior, el diseño de esquema de estrella es muy importante para desarrollar modelos de Power BI optimizados para el rendimiento y la facilidad de uso. Dado que, para esta herramienta un modelo bien diseñado es aquel que proporciona tablas para filtrar y agrupar y tablas para resumir, y todo esto se ajusta bien a los principios de los esquemas de estrella:

- Las tablas de dimensiones admiten el filtrado y la agrupación.
- Las tablas de hechos admiten el resumen.

Estos argumentos justifican haber realizado el trabajo previo de construir la base de datos con modelado de esquema de estrella. De esta forma, para la realización de los distintos informes requeridos por la empresa, se obtienen los datos directamente mediante consultas SQL a esta base de datos desde Power BI.

1.4. Ilustración de uso

En esta última sección se documenta detalladamente uno de los informes creados empleando la herramienta Power BI. Como se mencionó, los informes están etiquetados indicando si son considerados o no de prioridad alta. Para ilustrar un ejemplo de uso de esta herramienta de visualización aplicada al Business Intelligence, se seleccionó uno de los informes etiquetado con prioridad alta. Concretamente, se seleccionó el informe donde se pretende mostrar el indicador del tiempo de respuesta en solicitudes de mantenimiento correctivo. Este indicador engloba el tiempo desde que se recepciona un aviso, es decir, desde que se recibe la solicitud de asistencia de un cliente, hasta que el técnico se presenta en el centro para reparar la avería. Es decir, los parámetros a controlar son la fecha y hora de creación del aviso y la fecha y hora de entrada del primer albarán, donde un albarán se define como un documento donde figura el trabajo realizado junto con la fecha/hora de entrada y fecha/hora de salida.

El interés de la empresa en este informe radica en poder analizar el tiempo de respuesta. Es decir, poder observar el tiempo que se tarda en acudir a un aviso y así poder documentarse acerca de lo eficientes que son a la hora de prestar estos servicios a sus clientes. Por otro lado, también resulta de interés para Cofrico poder observar la evolución de este indicador de forma global, por solicitudes, plantas, condiciones de tarifa y centros. Este mismo esquema de análisis se extendió a otros reportes para mostrar los restantes indicadores requeridos.

El primer paso para la realización de cualquiera de los informes es la introducción de los datos en la herramienta Power BI que, como ya se mencionó en la Sección 1.3, se realizó mediante consultas SQL a la base de datos creada. Para ello se emplea Power Query, una herramienta integrada en Power BI necesaria para la extracción, carga y transformación de datos. Power Query incorpora una interfaz gráfica para obtener muchos orígenes de datos diferentes y un editor independiente para transformarlos y posteriormente cargarlos al modelo de datos. De esta forma, se introducen las variables de la tabla de hechos necesarias en cada reporte que, para el informe considerado como ejemplo para la ilustración del objetivo final de emplear esta herramienta, son el código de solicitud, la planta, el código de centro, la fecha/hora de creación del aviso, la fecha/hora de inicio (entrada del primer albarán), la condición de tarifa y el tiempo de respuesta expresado en horas y días. Finalmente, se incorporan las tablas de dimensiones de centros, plantas, condiciones de tarifa y días. Tras la introducción de todos los datos necesarios en Power BI, el siguiente paso es administrar las relaciones entre las distintas tablas. Una vez se tienen estos pasos completados, se puede proceder con la realización de los respectivos informes.

La interfaz de Power BI integra todos los elementos para la creación de un informe, desde las relaciones entre tablas, las tablas importadas de Power Query, las fórmulas DAX, etc. Destacar que el lenguaje de expresiones de análisis de datos (DAX) no es un lenguaje de programación, sino un lenguaje de fórmulas, donde estas son muy similares a las de Excel. Para realizar una fórmula DAX se debe escribir el nombre de la variable a crear y un signo de igual seguido con la expresión, los argumentos y los valores necesarios. Estas fórmulas se utilizaron para obtener medidas necesarias como, por ejemplo, la media de los distintos indicadores.

Se procede con el modelado de datos que es el proceso de segmentación, análisis y definición de datos que se han recopilado de la empresa y que se quieren reflejar de forma visual. Un buen modelo de datos es indispensable a la hora de obtener un análisis más concluyente para la empresa. De esta forma, para la realización de estos reportes es importante tener claro que se pretende mostrar para poder elegir las visualizaciones adecuadas, ya que es necesario un buen modelo para obtener un buen análisis y extraer mejores conclusiones. A continuación, en la Figura 1.2, se puede ver una de las páginas incluida dentro del informe seleccionado, donde se muestra qué está ocurriendo con este indicador en el último año de forma global. En particular, se observa la evolución del tiempo de respuesta relativa al último año por meses y días de la semana. Asimismo, se puede ver el número de solicitudes creadas y el tiempo medio de respuesta dado en horas y días. De esta forma la empresa dispone de un gráfico útil sobre la evolución de la media de este indicador, observando además los días y meses en los que se sobrepasa la media global.

En la Figura 1.3 se observa la segunda página de este informe, donde se muestra qué está ocurriendo por solicitud. La empresa tiene así información de aquellas solicitudes que presentan un tiempo de respuesta demasiado elevado.

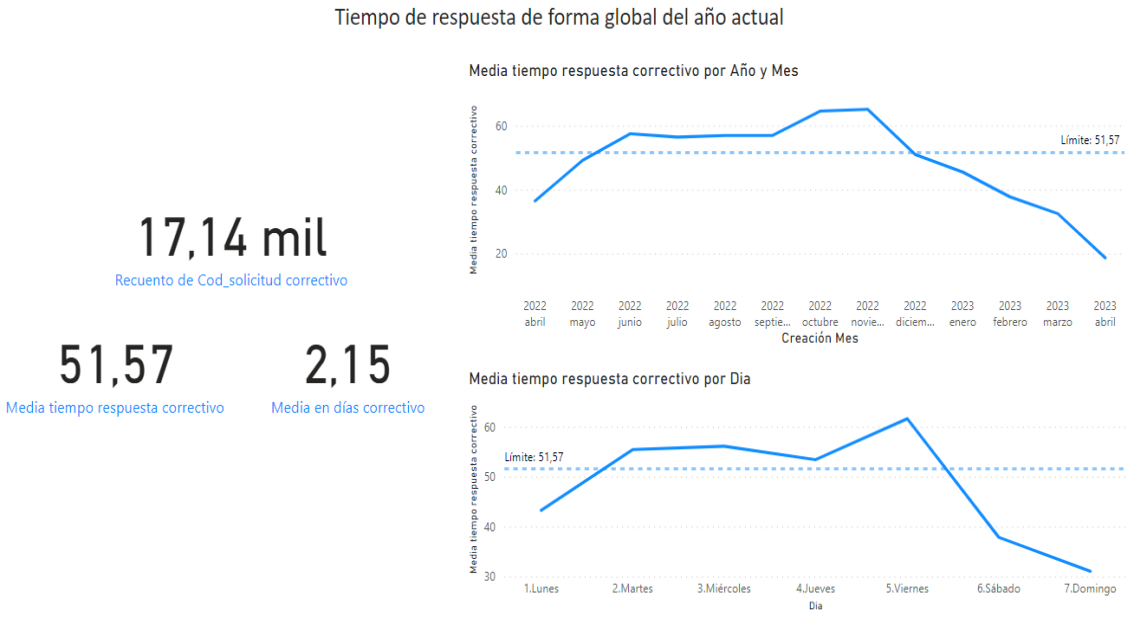
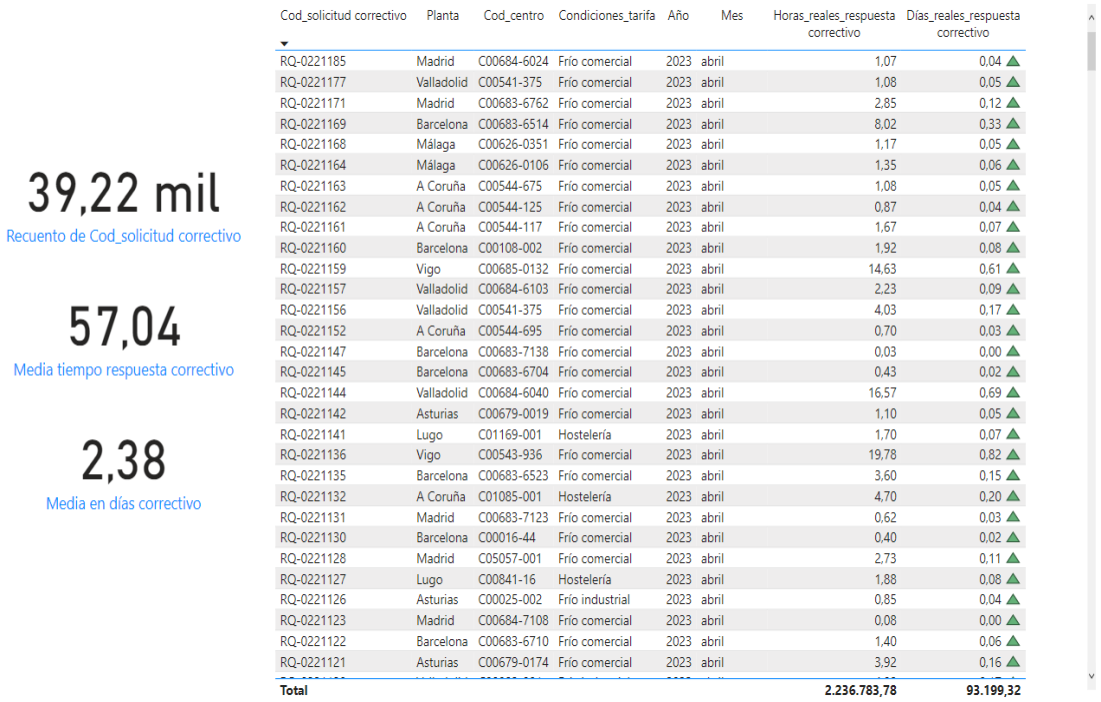


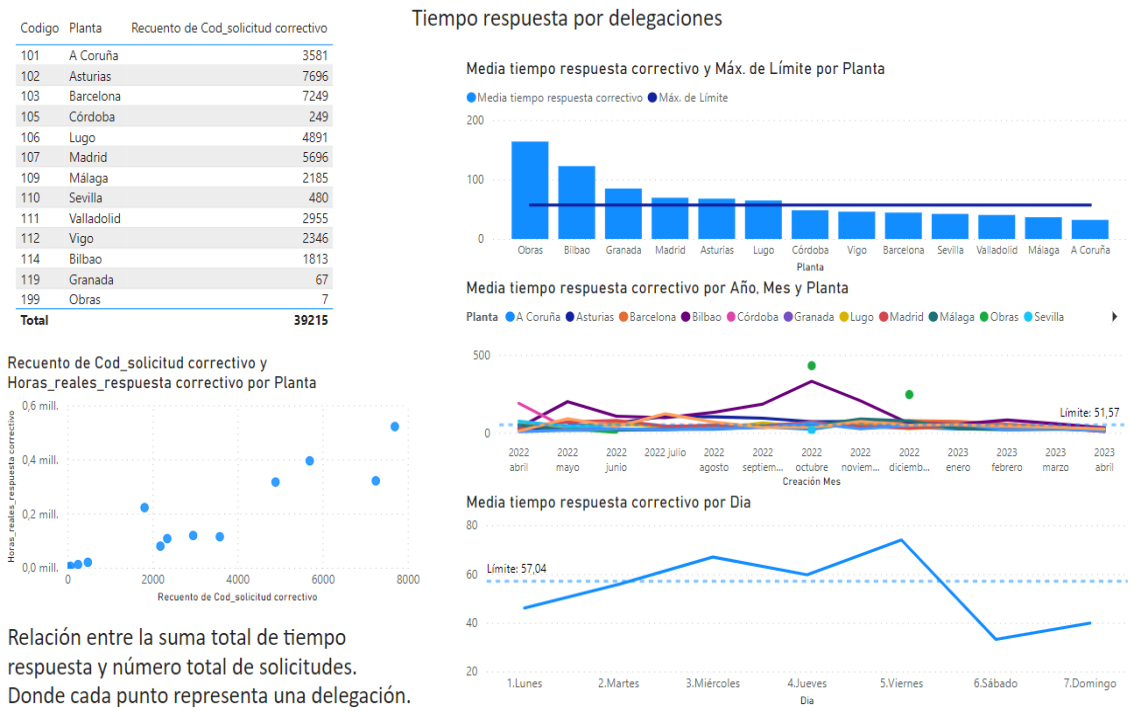
Figura 1.2: Página del informe seleccionado de forma global



Cod_solicitud correctivo	Planta	Cod_centro	Condiciones_tarifa	Año	Mes	Horas_reales_respuesta correctivo	Días_reales_respuesta correctivo
RQ-0221185	Madrid	C00684-6024	Frío comercial	2023	abril	1,07	0,04
RQ-0221177	Valladolid	C00541-375	Frío comercial	2023	abril	1,08	0,05
RQ-0221171	Madrid	C00683-6762	Frío comercial	2023	abril	2,85	0,12
RQ-0221169	Barcelona	C00683-6514	Frío comercial	2023	abril	8,02	0,33
RQ-0221168	Málaga	C00626-0351	Frío comercial	2023	abril	1,17	0,05
RQ-0221164	Málaga	C00626-0106	Frío comercial	2023	abril	1,35	0,06
RQ-0221163	A Coruña	C00544-675	Frío comercial	2023	abril	1,08	0,05
RQ-0221162	A Coruña	C00544-125	Frío comercial	2023	abril	0,87	0,04
RQ-0221161	A Coruña	C00544-117	Frío comercial	2023	abril	1,67	0,07
RQ-0221160	Barcelona	C00108-002	Frío comercial	2023	abril	1,92	0,08
RQ-0221159	Vigo	C00685-0132	Frío comercial	2023	abril	14,63	0,61
RQ-0221157	Valladolid	C00684-6103	Frío comercial	2023	abril	2,23	0,09
RQ-0221156	Valladolid	C00541-375	Frío comercial	2023	abril	4,03	0,17
RQ-0221152	A Coruña	C00544-695	Frío comercial	2023	abril	0,70	0,03
RQ-0221147	Barcelona	C00683-7138	Frío comercial	2023	abril	0,03	0,00
RQ-0221145	Barcelona	C00683-6704	Frío comercial	2023	abril	0,43	0,02
RQ-0221144	Valladolid	C00684-6040	Frío comercial	2023	abril	16,57	0,69
RQ-0221142	Asturias	C00679-0019	Frío comercial	2023	abril	1,10	0,05
RQ-0221141	Lugo	C01169-001	Hostelería	2023	abril	1,70	0,07
RQ-0221136	Vigo	C00543-936	Frío comercial	2023	abril	19,78	0,82
RQ-0221135	Barcelona	C00683-6523	Frío comercial	2023	abril	3,60	0,15
RQ-0221132	A Coruña	C01085-001	Hostelería	2023	abril	4,70	0,20
RQ-0221131	Madrid	C00683-7123	Frío comercial	2023	abril	0,62	0,03
RQ-0221130	Barcelona	C00016-44	Frío comercial	2023	abril	0,40	0,02
RQ-0221128	Madrid	C05057-001	Frío comercial	2023	abril	2,73	0,11
RQ-0221127	Lugo	C00841-16	Hostelería	2023	abril	1,88	0,08
RQ-0221126	Asturias	C00025-002	Frío industrial	2023	abril	0,85	0,04
RQ-0221123	Madrid	C00684-7108	Frío comercial	2023	abril	0,08	0,00
RQ-0221122	Barcelona	C00683-6710	Frío comercial	2023	abril	1,40	0,06
RQ-0221121	Asturias	C00679-0174	Frío comercial	2023	abril	3,92	0,16
Total						**2.236.783,78**	**93.199,32**

Figura 1.3: Página del informe seleccionado por solicitudes

Por otro lado, para observar el comportamiento de este indicador por planta, se optó por un gráfico de barras en el cual se muestra la media del tiempo de respuesta en cada delegación. Además, se realizó un gráfico de líneas en donde se puede ver la evolución del último año por planta y otro con la evolución general por días de la semana. Finalmente, se realizó un gráfico de dispersión para poder observar la relación entre las horas totales de tiempo de respuesta y el número de solicitudes de cada planta. En la Figura 1.4, se incluye una tabla con el número de solicitudes creadas en cada delegación. En suma, esta página facilita información a la empresa de aquellas plantas qué presentan mayor media en tiempo de respuesta y/o una evolución dispar a las demás. Para analizar que ocurre por condición de tarifa se emplearon las mismas visualizaciones que se acaban de mencionar.



Codigo	Planta	Recuento de Cod_solicitud correctivo
101	A Coruña	3581
102	Asturias	7696
103	Barcelona	7249
105	Córdoba	249
106	Lugo	4891
107	Madrid	5696
109	Málaga	2185
110	Sevilla	480
111	Valladolid	2955
112	Vigo	2346
114	Bilbao	1813
119	Granada	67
199	Obras	7
Total		**39215**

Figura 1.4: Página del informe seleccionado por plantas

Finalmente, para poder observar que está ocurriendo por centro, como el número total de estos es bastante elevado, se opta por una tabla y un gráfico de dispersión. La Figura 1.5 muestra la página del informe relativa a esta parte, describiendo el comportamiento de los centros de forma individual y, también, la relación entre las horas totales de tiempo de respuesta y el número de solicitudes para cada centro.

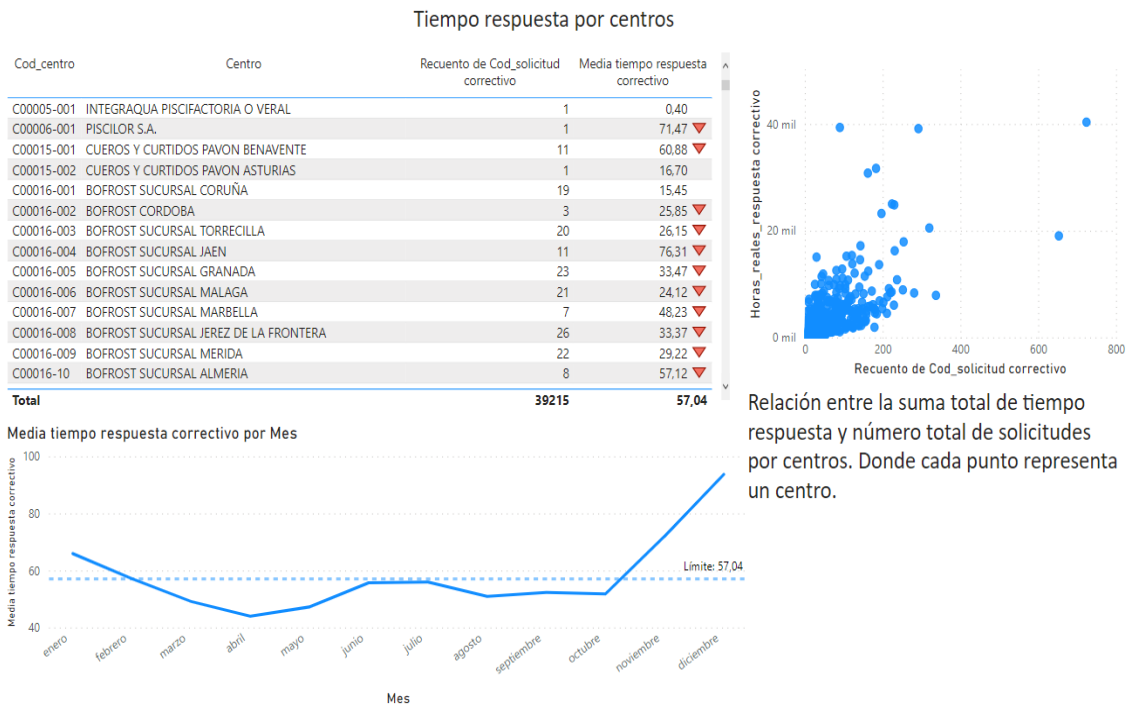


Figura 1.5: Página del informe seleccionado por centros

Capítulo 2

Modelos predictivos

Otro objetivo importante de la empresa es disponer de predicciones para el tiempo de desplazamiento y de ejecución y, de esta forma, desenvolver una planificación más idónea de sus servicios y mantener un mejor control de los tiempos empleados. Este segundo capítulo se centra en documentar como se han construido modelos predictivos para estos tiempos.

En primer lugar se realiza una exploración preliminar encaminada a reducir el rango de las distintas variables categóricas disponibles, alguna de las cuales puede tomar un número extremadamente elevado de valores. Con este fin, se lleva a cabo análisis cluster [23, 38, 50] con cada variable cualitativa en la idea de agrupar categorías para las que el comportamiento en distribución de la respuesta (tiempo de desplazamiento y ejecución) sea similar. A continuación, se emplean modelos de regresión con efectos aleatorios [3, 15, 40] para la construcción de los dos modelos predictivos definitivos.

2.1. Exploración preliminar: Análisis Cluster

En un inicio, se disponía de las variables centro, planta, técnico, línea de negocio, día de la semana y mes. Con el objetivo último de tratar de adquirir alguna información acerca de cuales tienen una mayor capacidad discriminativa en términos de las variables respuesta (tiempo de desplazamiento y tiempo de ejecución), se comenzó realizando un análisis descriptivo del comportamiento de ambos tiempos para los valores de estas variables, incluyendo análisis gráfico uni y bivariante. Además, como ya se ha mencionado, la presencia de variables nominales tomando un número enorme de valores (véase por ejemplo Cod_centro), motivó la búsqueda de subconjuntos de valores para los que la respuesta de interés presente un comportamiento en distribución análogo.

Naturalmente, de inicio no se tiene garantizada la existencia de unos pocos grupos bien establecidos y claramente separables del resto. No obstante, de ser así, se podría reducir sustancialmente el rango de esa variable nominal reemplazando sus valores por las etiquetas de los grupos, facilitando así la construcción de un modelo de regresión. Con esa idea en mente, se realiza un análisis cluster por separado para cada una de las variables cualitativas que presentan un alto número de diferentes categorías.

Por otro lado, lo razonable no es realizar la taxonomía en base simplemente a tiempos medios, sino considerar las distribuciones de tiempos, lo cual resulta mucho más informativo. En otros términos, el análisis cluster se fundamenta en evaluar similitud de acuerdo a cómo de parecidas son las distribuciones de la respuesta (tiempo de desplazamiento o tiempo de ejecución). De esta forma, se pretende segmentar el dominio de cada una de estas variables nominales de modo que la distribución del tiempo de respuesta

es parecida para valores de un mismo grupo y sustancialmente diferente para valores de distintos grupos. De observarse la existencia de grupos bien diferenciados, la partición encontrada permitiría identificar qué variables tiene un mayor potencial discriminativo y conocer los niveles de la variable que engloban las diferencias.

2.1.1. Análisis descriptivo

Como fase inicial, para tratar de adquirir alguna información acerca de qué variables tienen una mayor capacidad discriminativa, se realiza un análisis descriptivo de manera separada para las dos variables respuesta en función de cada una de las explicativas. Esto también puede ayudar a la hora de identificar posibles grupos dentro de las variables. En la Figura 2.1, se presentan diagramas de cajas del tiempo de ejecución para los valores de cada una de las variables nominales. Los diagramas de cajas correspondientes al tiempo de desplazamiento se muestran en la Figura 2.2.

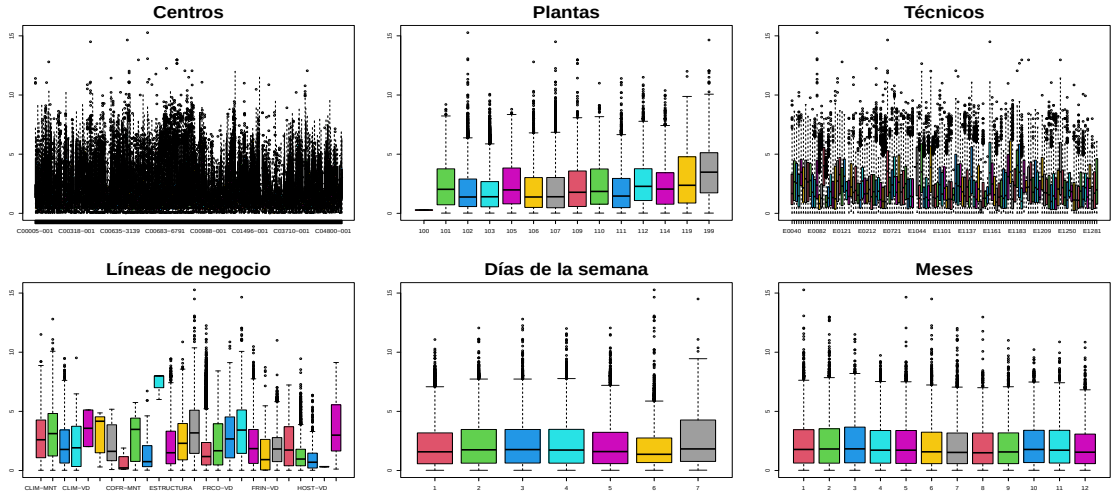


Figura 2.1: Diagramas de cajas del tiempo de ejecución

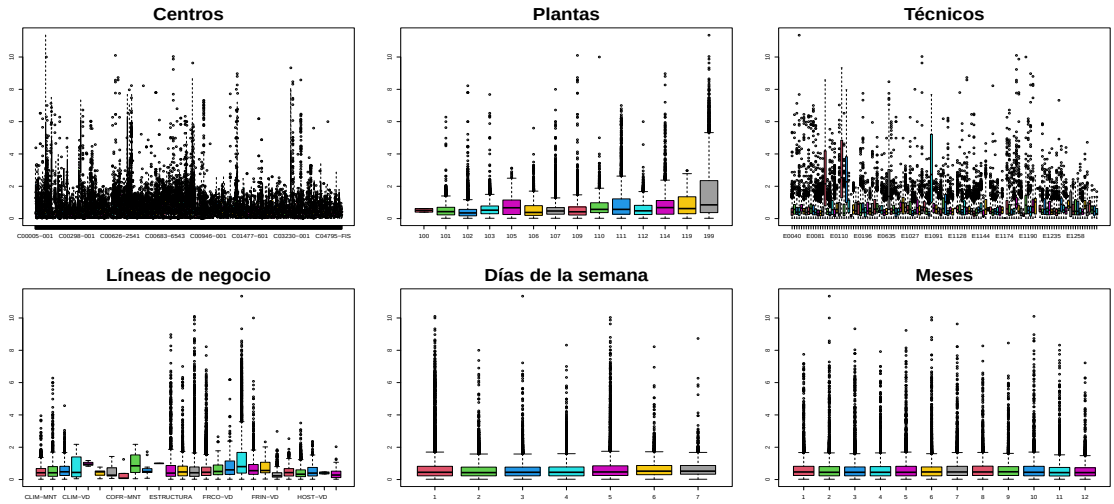


Figura 2.2: Diagramas de cajas del tiempo de desplazamiento

Además de la visualización gráfica, se realizaron también pruebas analíticas para poder concluir, con una significación estadística, si existe o no igualdad de medias. Específicamente, se realizaron pruebas de análisis de la varianza (*anova*) para cada respuesta y con cada explicativa. En los casos donde el test *anova* resultó estadísticamente significativo, es decir, donde se rechazó la igualdad de medias, se realizaron pruebas comparando medias dos a dos empleando diferentes correcciones por rango múltiple (en particular Tukey y Bonferroni). Las mencionadas comparaciones se realizaron para todas las variables excepto para líneas de negocio, técnico y centro, por presentar un número excesivamente elevado de categorías.

Como se observa en las gráficas, no se aprecia la presencia de grupos claramente diferenciados. Por otro lado, en los diagramas de cajas para los valores de las variables planta, días de la semana y meses, no se observa una gran capacidad discriminativa en términos de ninguna de las dos variables respuesta. A pesar de que el *anova* resultó estadísticamente significativo en todas ellas, las pruebas de rango múltiple presentaron p-valores muy elevados para la mayoría de comparaciones en todas las variables examinadas, concluyendo de ahí que estas variables no reportan información de valor para discriminar en media.

2.1.2. Análisis cluster

El siguiente paso consistió en realizar análisis cluster sobre las variables líneas de negocio, técnico y centro, con el fin de poder identificar grupos de categorías para los que la respuesta en estudio (bien tiempo de desplazamiento bien tiempo de ejecución) se distribuye de forma similar. El cluster se ejecuta conforme a dos diferentes criterios de disimilaridad, uno basado en la distancia de Cramér-von Mises [16, 51] y otro basado en la distancia de Kullback-Leibler [31]. Se eligieron estos dos criterios porque la relevancia de este análisis cluster radica en la comparación de distribuciones.

Metodología

Se ha empleado un algoritmo cluster jerarquizado aglomerativo, de tal modo que es preciso partir de una matriz de distancias dos a dos entre objetos y emplear entonces, de manera secuencial, un criterio de vinculación entre grupos. Como se pretende evaluar la discrepancia entre dos categorías en términos de discrepancia entre las distribuciones de probabilidad de la respuesta de interés bajo esas dos categorías, se acude a métricas que permitan comparar dos distribuciones.

Sean $\{X_1, ..., X_m\}$ e $\{Y_1, ..., Y_n\}$ dos muestras de tamaños m y n, $f_{X,m}$ y $f_{Y,n}$ denotan sus funciones de densidades y $F_{X,m}$ y $F_{Y,n}$ las respectivas distribuciones empíricas. Por un lado, se considera el estadístico de la prueba de Cramér-von Mises para chequear la igualdad de distribuciones de dos muestras independientes de longitudes arbitrarias, que toma la forma:

$$\mathrm{Q}^2_{m,n} = \frac{mn}{m+n}\left\{\sum_{i=1}^{m}\left(F_{X,m}(X_i) - F_{Y,n}(X_i)\right)^2 + \sum_{j=1}^{n}\left(F_{X,m}(Y_j) - F_{Y,n}(Y_j)\right)^2\right\},$$

Para el cómputo del estadístico $Q^2_{m,n}$, se emplea la función *cvm_test()* del paquete *twosamples* [21].

Alternativamente, se construyó la matriz de distancias dos a dos considerando la divergencia de Kullback-Leibler. Sean $p(\cdot)$ y $q(\cdot)$ dos densidades arbitrarias de variables arbitrarias respectivas P y Q. Se define la distancia de Kullback-Leibler como:

$$D(P,Q) = \frac{1}{2}\left[d(p,q) + d(q,p)\right] \quad \text{donde } d(p,q) = \sum_{j=1}^{M} p_j \log\left(\frac{p_j}{q_j}\right),$$

siendo p_j y q_j valores respectivos de las densidades p y q sobre una malla de tamaño M de la unión de dominios de ambas densidades.

En nuestro caso, se evalúa $d(\hat{f}_{X,m}, \hat{f}_{Y,n})$, donde $\hat{f}_{X,m}$ y $\hat{f}_{Y,n}$ denotan las estimaciones no paramétricas de tipo kernel de $f_{X,m}$ y $f_{Y,n}$, respectivamente. Para ello, se emplea la función *KLD()* del paquete *LaplacesDemon* [10].

Una vez se dispone de una matriz cuadrada de distancias, se emplean hasta tres criterios diferentes de vinculación entre grupos:

- Método del mínimo: Consiste en definir la distancia entre dos grupos como la menor de las distancias entre elementos de ambos grupos. (*method= 'single'*).
- Método del máximo: Consiste en definir la distancia entre dos grupos como la mayor distancia entre sus individuos (*method='complete'*).
- Método del promedio: Consiste en definir la distancia entre dos grupos como el promedio de las distancias entre sus individuos (*method='average'*).

Para determinar el número apropiado de clusters subyacentes se emplean distintos criterios propuestos en la literatura [9, 24, 57]. Específicamente, se han considerado los tres criterios que siguen y que consisten en maximizar un índice determinado.

- Índice Dunn: Cociente entre la distancia mínima de dos observaciones que no pertenecen a un mismo cluster y la distancia máxima de dos observaciones de un mismo cluster. Un mayor valor implica una menor varianza dentro del cluster y mayor varianza entre clusters. Por tanto, el objetivo es máximizar el índice Dunn.
- Ancho de silueta promedio: Se calcula utilizando la distancia media dentro del cluster y la distancia mínima entre los clusters. Se define de la siguiente manera:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}\right),$$

  donde $a(i)$ representa la distancia promedio del objeto i al resto de objetos dentro del cluster y $b(i)$ la distancia mínima del objeto i al resto de objetos en otros clusters. Su valor se mueve entre -1 y 1, siendo valores próximos a 1 un indicativo de que la observación se ha asignado al cluster correcto.
- Índice de Calinski y Harabasz: Se basa en la comparación de la relación ponderada entre la suma de los cuadrados y la suma de los cuadrados dentro del clúster. Se define como:

$$\frac{B(K)(N-K)}{W(K)(K-1)}, \text{ donde } B(K)=\sum_{k=1}^{K} a_k||\overline{X_k}-\overline{X}||^2 \text{ y } W(K)=\sum_{k=1}^{K}\sum_{C(j)=k}||X_j-\overline{X_k}||^2,$$

  K es el número correspondiente de clusters, $B(K)$ es la divergencia entre-clusters, $W(K)$ es la divergencia intra-clusters y n el número de muestras. Las soluciones cluster con valores más grandes del índice corresponden a soluciones 'mejores'.

Una vez se obtiene la solución cluster para cada variable, se pueden considerar como semejantes aquellas categorías que pertenecen al mismo cluster. No obstante, para comprobar la eficacia del análisis cluster realizado, se procedió a aplicar un análisis de la varianza dentro de cada uno de los grupos resultantes.

Esta forma de proceder es manejable para las variables líneas de negocio y técnicos, pero de realizarlo para la variable centro se llegaría a una matriz de dimensiones tan elevadas que hacen intratable computacionalmente la ejecución del algoritmo cluster, ya que el número de distintos códigos de centros era mucho mayor. Por tanto, fue necesario emplear un algoritmo cluster que trabaje con submuestras, como es el caso del algoritmo partitivo CLARA [17, 27], una variante del algoritmo PAM para muestras de gran dimensión. PAM (Partitioning around medoids) [18, 27] es un algoritmo cluster partitivo que fragmenta los datos en K grupos en un proceso secuencial diseñado para minimizar las distancias de los datos de cada cluster a uno de los datos muestrales que se toma como representante o centroide (el medoide del grupo). A diferencia del popular algoritmo K-medias, los centroides son datos muestrales y es factible emplear diferentes métricas. El algoritmo CLARA considera una pequeña muestra de los datos con tamaño fijo y aplica el algoritmo PAM para generar un conjunto óptimo de medoides para la muestra. La calidad de los medoides resultantes se mide por la diferencia media entre cada observación del conjunto de datos completo y el medoide del cluster. Clara repite el proceso de selección de muestra y creación de clusters un número predeterminado de veces para minimizar el sesgo de muestreo. La función existente en R para llevar a cabo el algoritmo CLARA no permitía introducir los datos de la forma de la cual se disponían. Por consiguiente, fue necesaria la previa programación del mismo, que se realizó conforme a los siguientes pasos:

1. Dividir aleatoriamente el conjunto de datos original en varios subconjuntos con tamaño fijo.
2. Calcular el algoritmo PAM para cada subconjunto y elegir los correspondientes K medoides. Asignar cada observación del conjunto de datos al medoide más cercano.
3. Calcular la suma de las diferencias de las observaciones a su medoide más cercano.
4. Conservar el subconjunto de datos para el que la suma es mínima.

En la Figura 2.3 se muestra un ejemplo de agrupación de los centros empleando el algoritmo CLARA y la métrica de Kullback-Leibler. Las densidades estimadas para el tiempo de desplazamiento con trazo de igual color corresponden a centros en el mismo cluster, en tanto que diferentes colores identifican centros de distintas grupos. En este caso se muestra la solución alcanzada con 4 clusters.

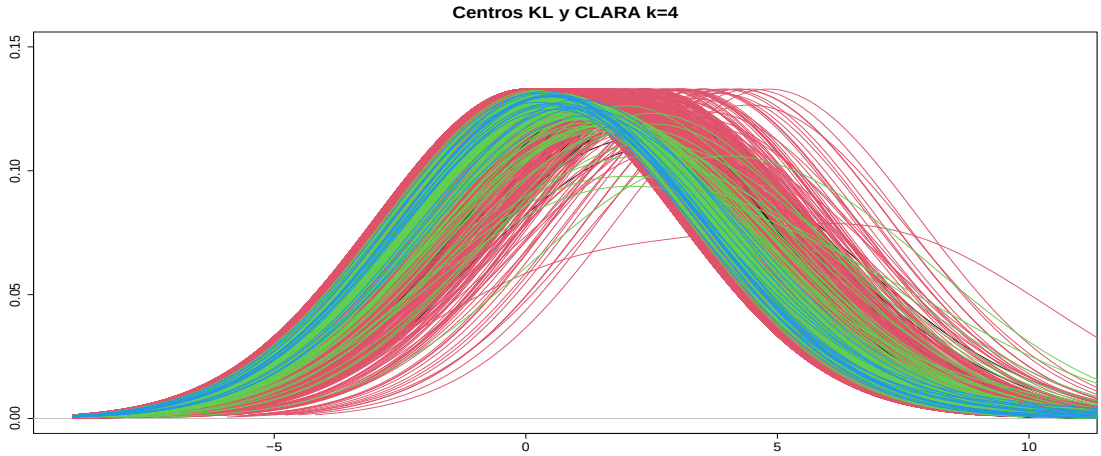


Figura 2.3: Representación de los grupos obtenidos para los centros

Resultados y conclusiones

Se observó que diversos enfoques conducían a diferentes resultados. Por un lado, los criterios de selección del número de clusters proporcionaron distintos valores. En la Figura 2.4, se puede observar

un ejemplo de estos resultados. En la práctica se probaron diferentes números de agrupaciones y se emplearon tres criterios de vinculación entre grupos. En la Figura 2.5, se observa un ejemplo de agrupación considerando dos clusters para los técnicos cuando la variable respuesta es el tiempo de desplazamiento.

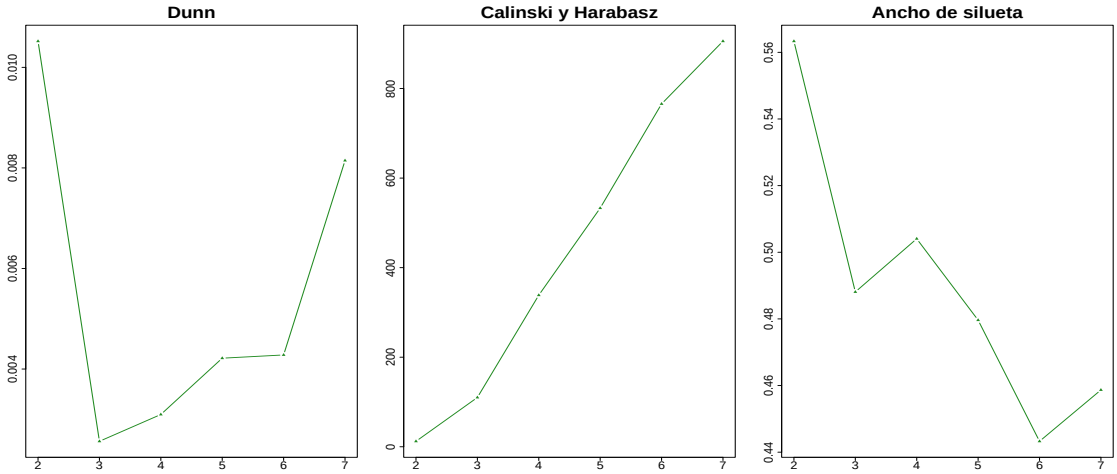


Figura 2.4: Representación valores de los distintos índices para la matriz de distancias correspondiente a la variable técnicos y para la respuesta tiempo de desplazamiento

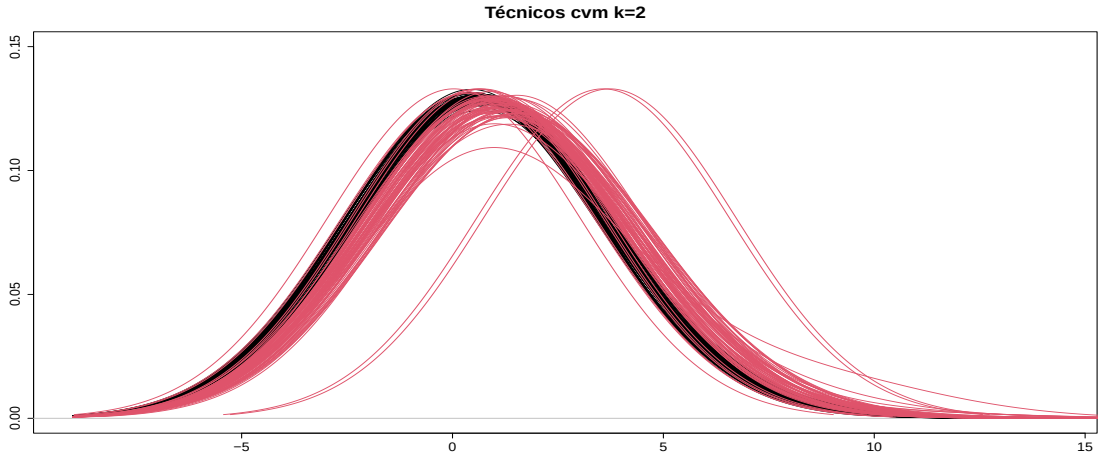


Figura 2.5: Representación de los grupos obtenidos para la variable técnicos y para la respuesta tiempo de desplazamiento

El resultado era fuertemente dependiente de la métrica considerada, llegando a diferentes agrupaciones cuando se ha empleado Cramér-von Mises o Kullback-Leibler. Con el fin de apoyar una u otra solución, se trató de evaluar el nivel de compacidad de los grupos empleando la prueba de distribución libre de Kruskal-Wallis, como alternativa a la prueba *anova* (dependiente de la hipótesis de normalidad no necesariamente satisfecha). Desafortunadamente, en la mayoría de los clusters, la prueba Kruskal-Wallis resultó significativa ($p < 0.01$), delatando así baja compacidad de los clusters encontrados. A resultas de todo ello, se concluyó que, de forma análoga al análisis de medias, el análisis cluster no

resultó una herramienta útil para identificar grupos claramente homogéneos.

En conclusión, el análisis descriptivo permitió observar que parte de las variables nominales disponibles no aportaban información de valor para discriminar en media, siendo únicamente relevantes las variables centro, técnicos y líneas de negocio. La realización del análisis cluster descartó la idea de reducir el rango de las distintas variables discretas, ya que no fue posible identificar grupos claramente homogéneos dentro de las mismas. Por consiguiente, se decidió enfocar el problema mediante un modelo de regresión de efectos aleatorios. Naturalmente, en la construcción de estos modelos de regresión se tuvieron en consideración además aquellas variables, continuas o no, que aportaban información relevante. Concretamente, para la predicción del tiempo de desplazamiento se añadió la variable kilómetros y para el tiempo de ejecución una clasificación de las averías y de los equipos a intervenir. Además, en este momento se descartó la consideración de la variable técnico, dado que la empresa quería disponer de las predicciones antes de realizar la selección del mismo.

2.2. Modelo predictivo para el tiempo de desplazamiento

En esta sección, se documenta detalladamente la construcción del modelo predictivo para el tiempo de desplazamiento. Tras un análisis descriptivo de las variables a priori relevantes para predecir el tiempo de desplazamiento, se procede con la construcción del modelo de regresión empleando efectos aleatorios y se chequea la validación del modelo final. Por último, se obtienen las predicciones para satisfacer los requerimientos de la empresa.

Tras la exploración preliminar, resultaron relevantes para modelizar el tiempo de desplazamiento las variables: kilómetros recorridos desde el origen hasta el centro correspondiente y el código identificando a este último. Como ya se mencionó en la Sección 1.1, la variable centro es una variable nominal con un número extremadamente elevado de categorías diferentes, lo que justifica ser tratada como una covariable de tipo factor de efectos aleatorios. En otros términos, se asume que el efecto sobre el tiempo de desplazamiento debido al centro no es determinista sino aleatorio. En definitiva, se plantea un modelo constituido por una variable explicativa continua de nivel 1, los kilómetros, y una variable de nivel 2, los centros.

2.2.1. Análisis descriptivo de las variables

Inicialmente se parte de un conjunto de datos con 18533 observaciones. No obstante, fue necesario realizar una previa limpieza de los datos para considerar aquellos que realmente reflejan solicitudes de mantenimiento correctivo. En definitiva, no se consideran las solicitudes que tienen asociado un pedido de venta, contrato o proyecto. Asimismo, se eliminan aquellas solicitudes cuyo cociente entre los kilómetros recorridos y el tiempo empleado fue menor de 40 o mayor de 120. Tras este proceso de filtrado, el conjunto de datos se reduce a 10264 observaciones.

Resulta muy oportuno enfatizar que una exploración preliminar de la variable cualitativa Cod_centro permitió observar que la variable ya filtrada presenta 1303 códigos diferentes y que para muchos de ellos existe una única observación.

Análisis univariante

Se comenzó realizando un análisis descriptivo de la variable respuesta, es decir, del tiempo de desplazamiento. La Figura 2.6 muestra un histograma y un diagrama de cajas para esta variable.

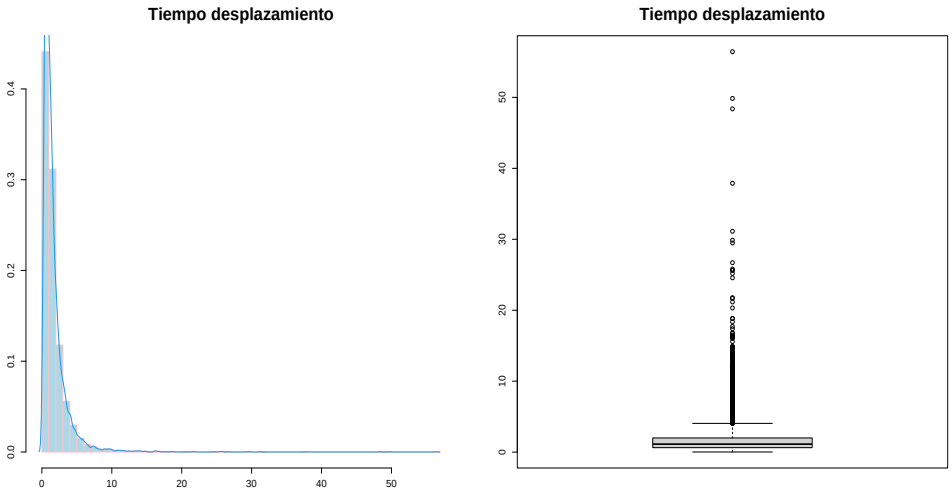


Figura 2.6: Representaciones de la distribución del tiempo de desplazamiento

Se aprecia, en ambas representaciones, una clara asimetría en la distribución de la variable. Se procede a aplicar la transformación logarítmica en orden a simetrizar la distribución tal y como se muestra en la Figura 2.7.

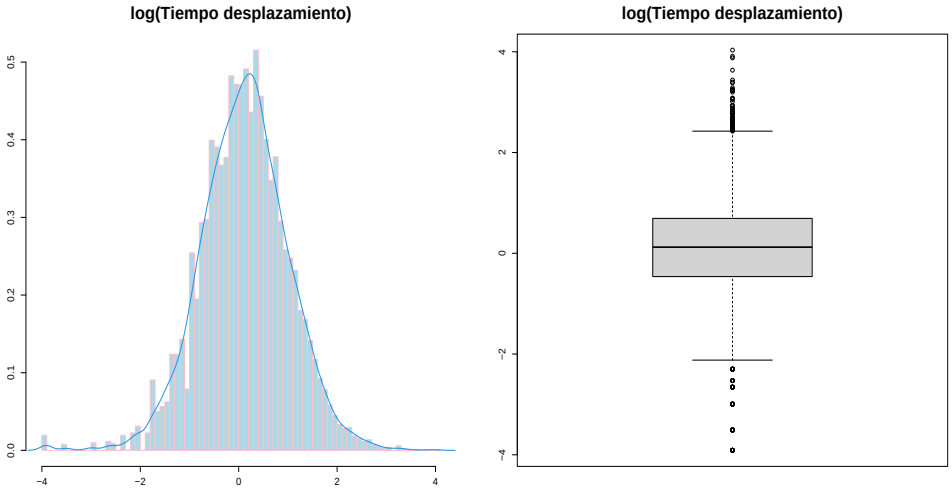


Figura 2.7: Representaciones de la distribución del logaritmo del tiempo de desplazamiento

La Figura 2.8 muestra las mismas representaciones para la variable explicativa kilómetros. De nuevo, se aprecia una clara asimetría y se procede a aplicar logaritmos.

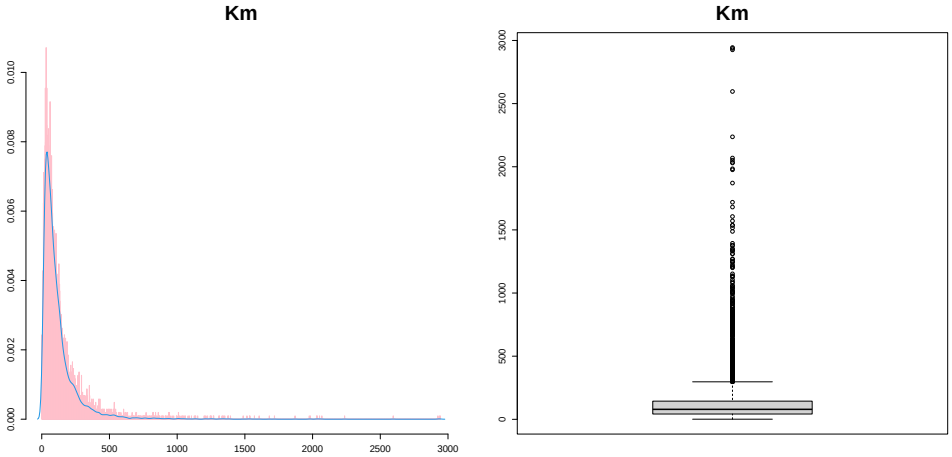


Figura 2.8: Representaciones de la distribución de la variable kilómetros

Análisis bivariante

En este apartado se muestran diferentes gráficos bivariantes, en los cuales se enfrentan las distintas variables explicativas y la respuesta. En un principio, se representa un diagrama de dispersión entre la transformación de la variable explicativa continua y la de la respuesta, es decir, entre el logaritmo de los kilómetros recorridos y el logaritmo del tiempo de desplazamiento, el cual se puede ver en la Figura 2.9. En él se aprecia una evidente relación lineal.

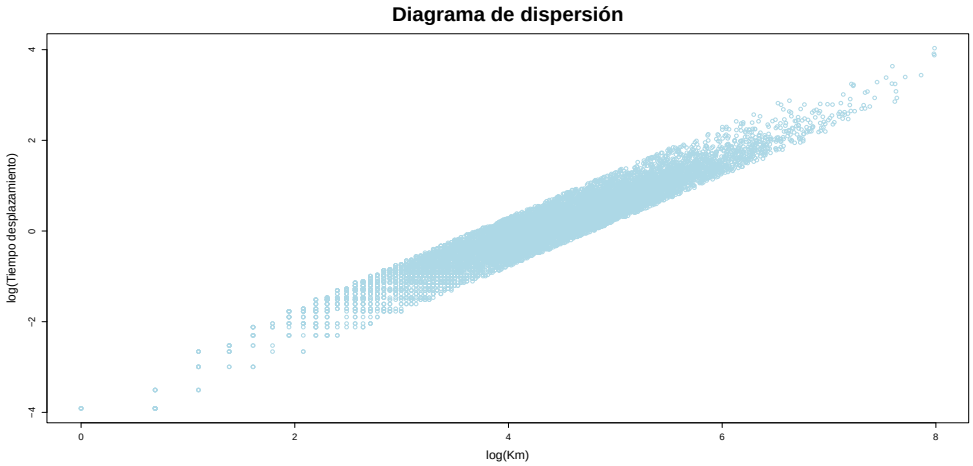


Figura 2.9: Diagrama de dispersión del logaritmo del tiempo de desplazamiento frente al logaritmo de la variable kilómetros

A continuación, se obtienen diagramas de cajas y estimaciones tipo kernel de las densidades de probabilidad (empleando la función *density*) para el logaritmo de la variable respuesta Horas_reales_desplazamiento según cada valor de la variable nominal Cod_centro. No obstante, el excesivo número de distintos códigos de centros (1303) impide apreciar con claridad las representaciones. Para hacer más nítidas e informativas las mismas, se consideran los 11 centros con mayor número de solicitudes. Estas representaciones se pueden apreciar en la Figura 2.10, donde se observa que los tiempos de desplaza-

miento no son homogéneos por centros, ya que se aprecian diferencias en sus medias y variabilidad.

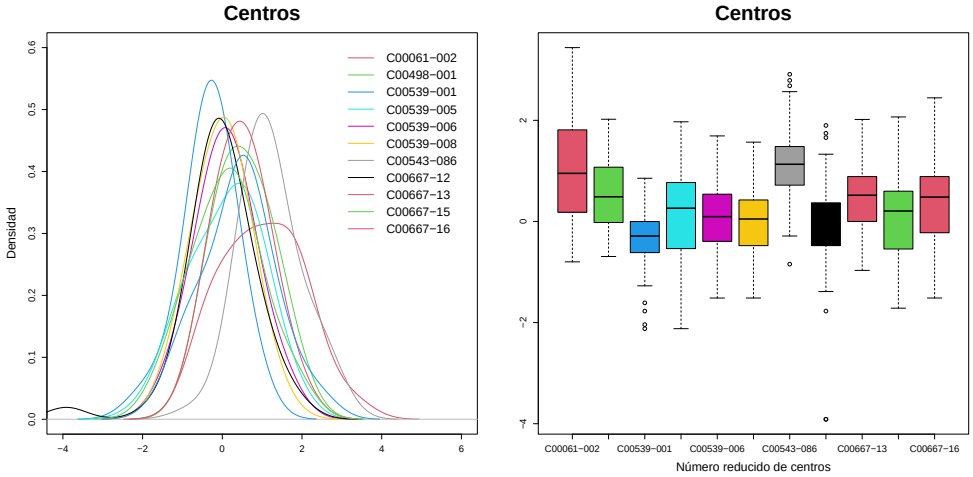


Figura 2.10: Representaciones de las distribuciones del logaritmo del tiempo de desplazamiento para los 11 centros con mayor número de solicitudes

2.2.2. Construcción del modelo

Dado que la variable cualitativa Cod_centro presenta características como las que siguen:

- Un número elevado de más de 1300 categorías, la mayoría de las cuales registran un único dato.
- Los tiempos de desplazamiento presentan variabilidad no nula a través de estas categorías.
- No ha sido factible agrupar las categorías en clusters homogéneos caracterizados por un único modelo de distribución del tiempo de desplazamiento.

Parece justificado considerar la necesidad de construir un modelo de análisis de la varianza con efectos aleatorios.

De acuerdo con los argumentos previos, se consideran dos potenciales modelos lineales de regresión de efectos aleatorios [15] con una variable explicativa continua y una categórica que se formulan como sigue.

- Modelo con intercepto aleatorio [M1]:

$$Y_{ij} = \beta_0 + \beta_1 X_{ij} + u_{0j} + \epsilon_{ij}, \;\; j = 1, \ldots, 1303, \; i = 1, \ldots, n_j,$$

  donde $u_{0j} \sim N(0, \sigma_u^2)$ recoge el efecto aleatorio sobre la respuesta del centro, Y representa el logaritmo del tiempo de desplazamiento y X la variable explicativa de nivel 1, log(km). Además, los interceptos aleatorios de los distintos centros son independientes entre sí e independientes de los errores de primer nivel.

- Modelo con intercepto y pendiente aleatorios [M2]:

$$Y_{ij} = \beta_0 + \beta_1 X_{ij} + u_{1j} X_{ij} + u_{0j} + \epsilon_{ij}, \;\; j = 1, \ldots, 1303, \; i = 1, \ldots, n_j,$$

  donde $u_{0j} \sim N(0, \sigma_{u_0}^2)$, $u_{1j} \sim N(0, \sigma_{u_1}^2)$, Y representa el logaritmo del tiempo de desplazamiento y X representa la variable explicativa de nivel 1, log(km). Además, los interceptos y pendientes aleatorios de los distintos centros son independientes entre sí e independientes de los errores de primer nivel.

Conviene enfatizar que, al objeto de ajustar los modelos M1 y M2 en R se dispone de los paquetes *lme4* [5] y *nlme* [39]. En primer lugar se chequea la necesidad de considerar un modelo con efectos aleatorios empleando máxima verosimilitud restringida para poder realizar una comparación anidada, lo que en programación supone el empleo de *gls()* en lugar de *ls()*. El test resultó significativo, arrojando un p-valor inferior a 0.0001, lo que nos lleva a concluir el interés de considerar un modelo de efectos aleatorios. Se chequeó entonces si el modelo lineal menos parsimonioso, M2, conduce a un incremento significativo de verosimilitud respecto de aquella alcanzada con M1. Los resultados obtenidos con la función *anova()* fuero de nuevo significativos, lo que nos lleva a decantarnos por M2, un modelo lineal con intercepto y pendiente aleatorias.

Tras realizar el ajuste del modelo, se obtuvieron los siguientes resultados:

Parámetro	Estimación	Error típico	p-valor	IC
β_0	-3.789	0.015	$< 2e^{-16}$	$(-3.82, -3.76)$
β_1	0.898	0.003	$< 2e^{-16}$	$(0.89, 0.9)$

Cuadro 2.1: Estimaciones de los efectos fijos y intervalos de confianza

Parámetro	Varianza	Desviación típica	IC para σ
Intercepto	$\sigma^2_{u_0} = 0.033$	$\sigma_{u_0} = 0.181$	$(0.14, 0.22)$
Pendiente	$\sigma^2_{u_1} = 0.002$	$\sigma_{u_1} = 0.044$	$(0.04, 0.05)$
Residual	$\sigma^2_{\epsilon} = 0.062$	$\sigma_{\epsilon} = 0.248$	$0.245, 0.252)$

Cuadro 2.2: Estimaciones y intervalos de confianza de las componentes de la varianza del modelo

En el Cuadro 2.1 se pueden ver las estimaciones puntuales para el intercepto y la pendiente media asociada a la variable explicativa continua. Se observa que ambos parámetros son significativamente distintos de cero. En el Cuadro 2.2 se introducen las estimaciones de las desviaciones típicas del efecto aleatorio del intercepto, de la pendiente y del error.

En estos modelos es importante tener una medida de la proporción total de varianza atribuible a las diferencias entre grupos, es decir, distinguir claramente la variabilidad entre- e intra-grupos. Esto se puede obtener construyendo el coeficiente de partición de la varianza (VPC) [15], que viene dado por:

$$\text{VPC} = \frac{\sigma^2_{u_0} + \sigma^2_{u_1}}{\sigma^2_{u_0} + \sigma^2_{u_1} + \sigma^2_{\epsilon}}$$

En el caso que nos ocupa, un 36 % de la variabilidad se debe a diferencias entre los centros.

2.2.3. Validación del modelo

Se discute en este apartado el grado de cumplimiento de las hipótesis estructurales del modelo ajustado en orden a validarlo, al tiempo que se evalúa su capacidad predictiva. Para la validación del modelo se usa el paquete *HLMdiag* [32] disponible en R. En primer lugar, se obtienen los residuos de nivel 1 y se consideran tanto los obtenidos con ajuste individual en cada grupo (LS) como los obtenidos teniendo en cuenta las predicciones de los efectos aleatorios (EB).

Se procedió a chequear la hipótesis de normalidad gráfica y analíticamente. La Figura 2.11 muestra histogramas de los residuos, junto a los mejores ajustes paramétricos de la familia de Gaussianas y ajustes no paramétricos. Se aprecia claramente la unimodalidad y la simetría. A pesar de que la prueba analítica de Lilliefors resultó significativa, la desviación de la normal no parece excesivamente preocupante. Nótese que se dispone de un tamaño muestral muy elevado, lo que hace particularmente sensibles a las pruebas de bondad de ajuste para detectar desviaciones menores de la nula (i.e. devolver p-valores muy pequeños). Por otro lado, la simetría y unimodalidad mostrado por el gráfico y el hecho en sí de emplear una muestra muy grande valida el proceso de estimación aún en presencia de pequeñas desviaciones de la normalidad.

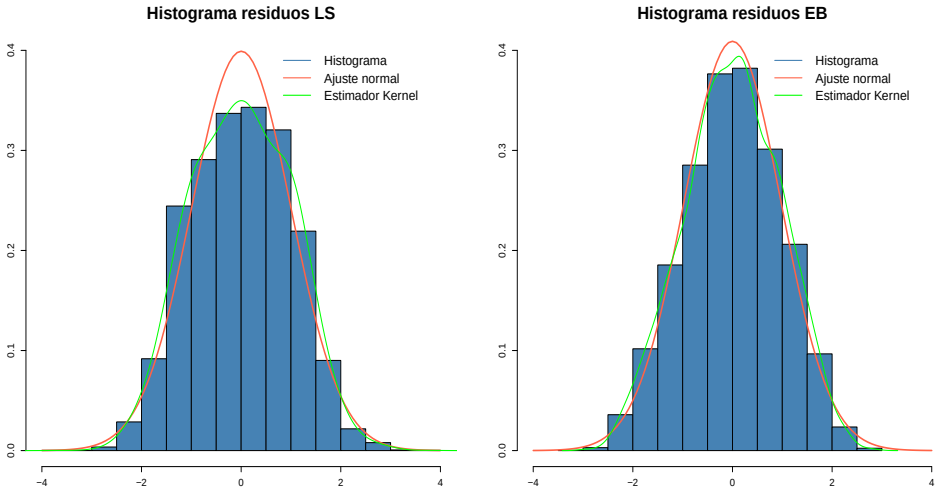


Figura 2.11: Ajuste paramétrico y no paramétrico a la densidad de probabilidad de los residuos de nivel 1

En orden a examinar la hipótesis de linealidad, la Figura 2.12 muestra un diagrama de dispersión de los residuos frente a la variable explicativa log(km) y un ajuste no paramétrico por regresión local lineal. La nube de puntos no sugiere patrón alguno tras los residuos, más allá de lo que acontece para valores muy bajos de log(km) que, de hecho, justifican el cambio significativo de tendencia en esa zona. Sin embargo, para la enorme mayoría de residuos obtenidos sobre la parte relevante del rango de log(km) no se aprecia tendencia. Cabe subrayar que se reajustó el modelo introduciendo monomios de hasta orden tres sin obtener mejoras en el ajuste ni incrementar su capacidad predictiva.

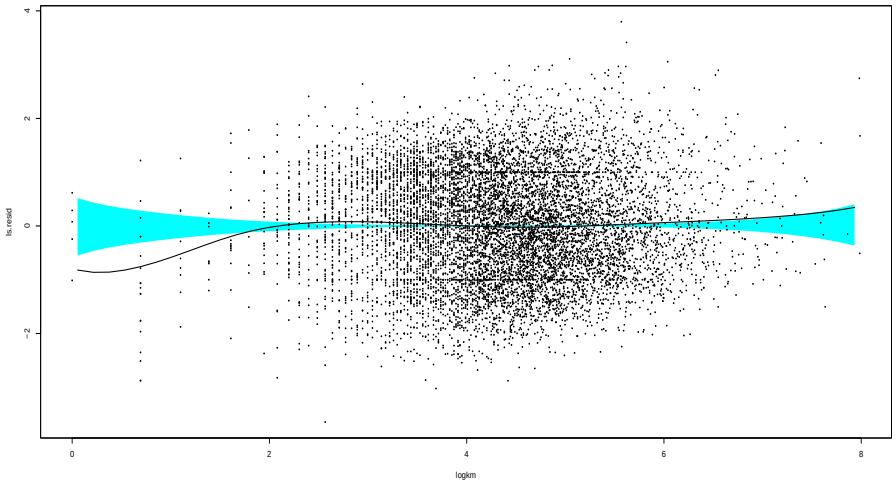


Figura 2.12: Diagrama de dispersión de los residuos frente a la variable explicativa y un ajuste no paramétrico de regresión local lineal

Por último, se comprueba la hipótesis de homocedasticidad a través de un diagrama de cajas de los residuos frente a la variable que induce el efecto grupo, el cual puede verse en la Figura 2.13.

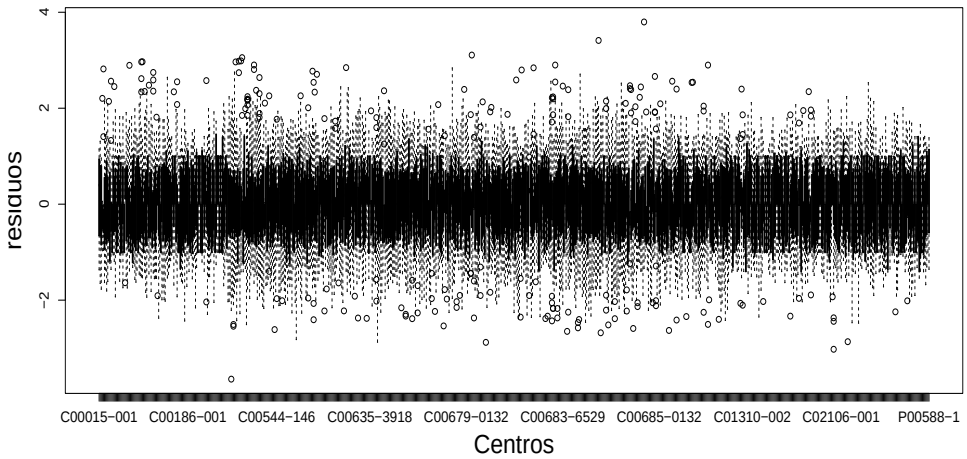


Figura 2.13: Diagrama de cajas de los residuos frente a la variable Cod_centro

El análisis de los residuos del nivel 2 condujo a resultados similares (ver Figura 2.14).

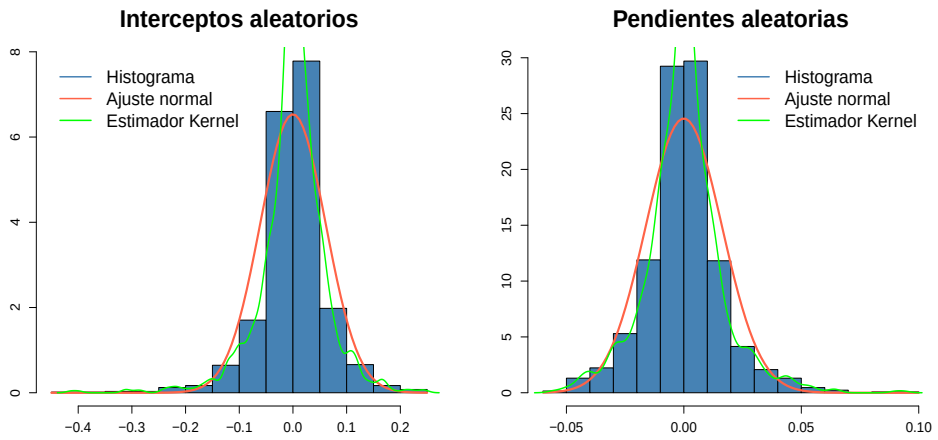


Figura 2.14: Ajuste paramétrico y no paramétrico de la densidad de probabilidad de los residuos de nivel 2

Se chequeó la hipótesis de incorrelación (independencia en contexto de normalidad) tanto para los residuos de nivel 1 como para los de nivel 2. Se aplicó un test de rachas empleando la función *RunsTest()* disponible en el paquete *DescTools* [46] o la función *runs.test()* disponible en *randtests* [11]. Ambas funciones proporcionan prácticamente los mismos p-valores y en todos los casos se obtiene un resultado no estadísticamente significativo, es decir, no se rechaza la hipótesis nula de incorrelación. Por otro lado, también debe haber incorrelación entre los errores de nivel 1 y nivel 2. No obstante, teniendo en cuenta como fue recogida la muestra, no hay nada que haga dudar de la ausencia de la misma.

Por último se realiza un análisis de influencia empleando la función *hlm_influence()*, que permite identificar observaciones influyentes a nivel individual y a nivel de grupo. Inicialmente, se calculan las observaciones influyentes a nivel individual y se representan las mismas en función del umbral interno (3∗IQR) que proporciona la función *dotplot_diag()*. De esta forma, se puede observar que muchas de las observaciones superan ese umbral. No obstante, si se considera un umbral de 0.5, vemos que ninguna de ellas lo supera. Si se procede con la eliminación de algunas de las observaciones más influyentes, se puede ver que no se produce mejora en el ajuste ni en el comportamiento de los residuos. Por otro lado, en los modelos de efectos aleatorios es más interesante ver los grupos influyentes. No obstante, si se prueba a eliminar alguno de ellos, tampoco se produce ninguna mejora. En conclusión, se considera el conjunto de datos completo.

El siguiente objetivo era obviamente determinar la capacidad predictiva del modelo ajustado. Nos hemos centrado en el coeficiente de determinación que, en los modelos lineales de efectos aleatorios, puede evaluarse en dos direcciones [36]:

- R^2 marginal: se relaciona con la variabilidad explicada por los factores fijos.
- R^2 condicional: se relaciona con la variabilidad explicada por factores fijos y aleatorios.

Es decir, para un modelo lineal mixto se puede definir el R^2 marginal como:

$$R^2_m = \frac{\sigma^2_f}{\sigma^2_f + \sigma^2_{u_0} + \sigma^2_{u_1} + \sigma^2_e},$$

donde σ^2_f es la varianza calculada a partir de las componentes de efectos fijos del modelo mixto. Por otro lado, el R^2 condicional, que se puede interpretar como la varianza explicada por todo el modelo,

se puede definir como:

$$R_c^2 = \frac{\sigma_f^2 + \sigma_{u_0}^2 + \sigma_{u_1}^2}{\sigma_f^2 + \sigma_{u_0}^2 + \sigma_{u_1}^2 + \sigma_e^2}.$$

Se obtienen ambos valores empleando la función *r.squaredGLMM* disponible en la librería *MuMIn* [4]. Resultando un valor próximo a 1, concretamente el R^2 marginal es 0.915497 y el R^2 condicional 0.9235503.

2.2.4. Predicciones

Uno de los principales objetivos de este trabajo es obtener las predicciones una vez construido el modelo. Para la empresa es tan relevante disponer de un intervalo como del valor concreto de predicción. Por tanto, en esta sección se presentan las predicciones, intervalos de confianza para las medias ajustadas e intervalos de predicción [29].

La función *predict()* para modelos de efectos aleatorios permite obtener directamente las predicciones a partir del modelo ajustando tomando o no los efectos aleatorios en consideración (lo que habilita la posibilidad de generar predicciones también para centros nuevos no incluidos en la base de datos).

Para obtener intervalos de confianza para las medias ajustadas se emplea *bootMer()*, función que reestima el modelo usando muestras aleatorias de los datos. Se construyen intervalos bootstrap donde se considera la fuente de variación de la varianza residual, la incertidumbre en los coeficientes fijos y la incertidumbre en los parámetros de varianza de los factores de agrupación. Asimismo, también se dispone de la función *predictInterval()* disponible en el paquete *merTools* [28], la cual incorpora toda la incertidumbre de la varianza residual y la de los coeficientes fijos y parte de la incertidumbre en los parámetros de varianza de los factores de agrupación, pero los parámetros de varianza en sí se tratan como fijos.

En la Figura 2.15, se observan los intervalos de confianza al 95 % para el tiempo medio de desplazamiento obtenidos con ambas funciones para los valores de las explicativas de las 45 primeras filas del conjunto de datos empleado.

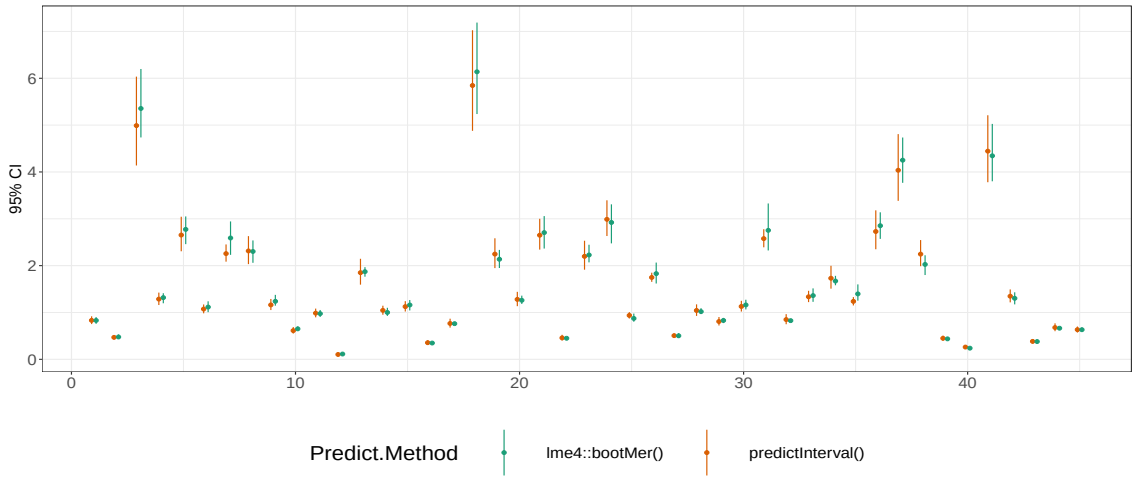


Figura 2.15: Representaciones de los intervalos de confianza al 95 % empleando las funciones *bootMer()* y *predictInterval()*

Si se compara la función *predictInterval()* con la función *lme4::bootMer()*, se tiene que esta última es más lenta computacionalmente, debido a que para cada remuestra bootstrap se reestiman las componentes de los efectos aleatorios. No obstante, la función *predictInterval()* considera la varianza de los efectos aleatorios fija, lo que se traduce en que los intervalos son demasiado conservadores. Por este motivo, los intervalos de confianza presentados a la empresa son los obtenidos con la función *bootMer()*.

Finalmente, la función *predictInterval()* permite obtener también intervalos de predicción. En la Figura 2.16, se observa la representación de los mismos obtenidos para las 45 primeras filas del conjunto de datos.

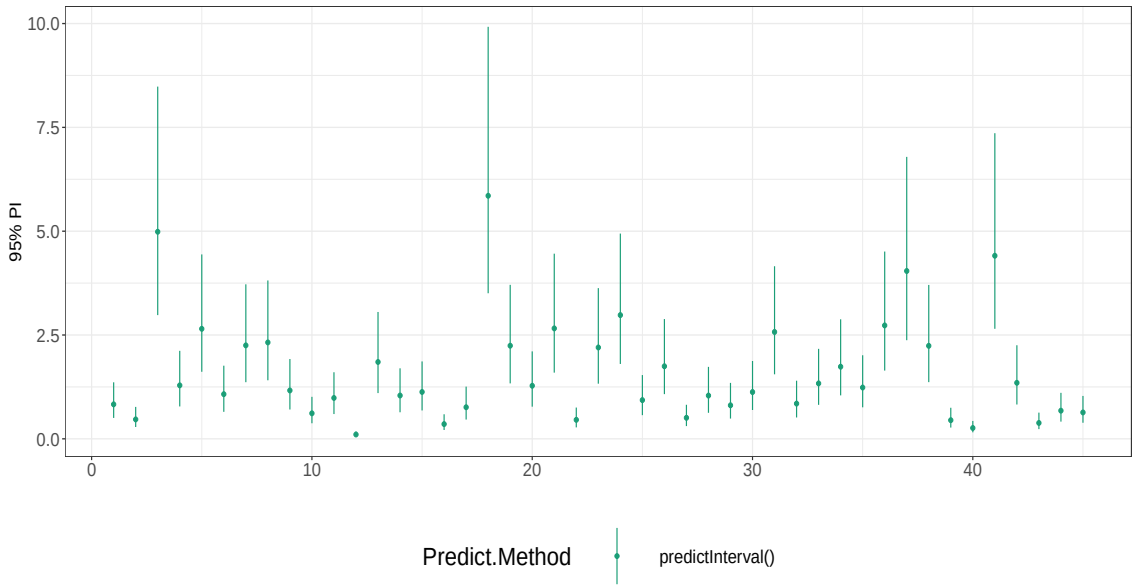


Figura 2.16: Representaciones de los intervalos de predicción al 95 % empleando la función *predictInterval()*

2.3. Modelo predictivo para el tiempo de ejecución

En esta sección, se documenta detalladamente la construcción del modelo predictivo para el tiempo de ejecución. Tras un análisis descriptivo de las variables a priori relevantes para predecir el tiempo de ejecución, se procede con la construcción del modelo de regresión empleando efectos aleatorios y se documenta la validación del modelo definitivo. Por último, se obtienen las predicciones para satisfacer los requerimientos de la empresa.

Para modelizar el tiempo de ejecución se consideran las variables: Condiciones_tarifa, Num_tecnicos, Clasificación_centro_tecnico, Clasificación_tarifa_tecnico, Equipo, Tipo_equipo y Tipo_averia. De esta forma, se dispone de cuatro variables cualitativas de nivel 1 y de tres variables factor de nivel 2.

2.3.1. Análisis descriptivo de las variables

Como en el caso de los tiempos de desplazamiento, fue preceptivo filtrar los datos para considerar solo aquellos que realmente reflejan solicitudes de mantenimiento correctivo. Con este fin, no se tendrán en cuenta las solicitudes que tenga asociado un pedido de venta, contrato o proyecto. Se consideran

aquellas que fueron intervenidas en un único día o en menos de 3 albaranes. Se eliminan los tiempo inferiores a 10 minutos, debido a que estos son considerados errores en la base de datos. Solo se tienen en cuenta las solicitudes clasificadas con una condición de tarifa que sea frío comercial, frío industrial, clima o hostelería. Finalmente, se eliminan aquellas cuyo tipo de avería está clasificada en un tipo de equipo que no procede. Tras este proceso de filtrado, se trabaja con un conjunto de datos con 8890 observaciones.

Como ya se introdujo en la Sección 1.1, la variable explicativa condiciones de tarifa presenta cuatro códigos diferentes ('01', '02', '03' y '04'), las variables Clasificación_centro_tecnico y Clasificación_tarifa_tecnico se dividen ambas en 'Alto', 'Bajo' y 'Medio', finalmente, la variable Num_tecnicos se agrupa en '1', '2' y '2+'. Por otro lado, para las tres variables de nivel 2, se observa que la variable Tipo_equipo presenta 12 categorías distintas, Tipo_averia tiene 49 códigos diferentes y hay 96 códigos de equipo. Se aprecia también que estas dos última variables presentan categorías con una única observación.

Análisis univariante

Se realiza un análisis descriptivo de la variable respuesta que, en este caso, es el tiempo de ejecución. La Figura 2.17 muestra un histograma y un diagrama de cajas para esta variable.

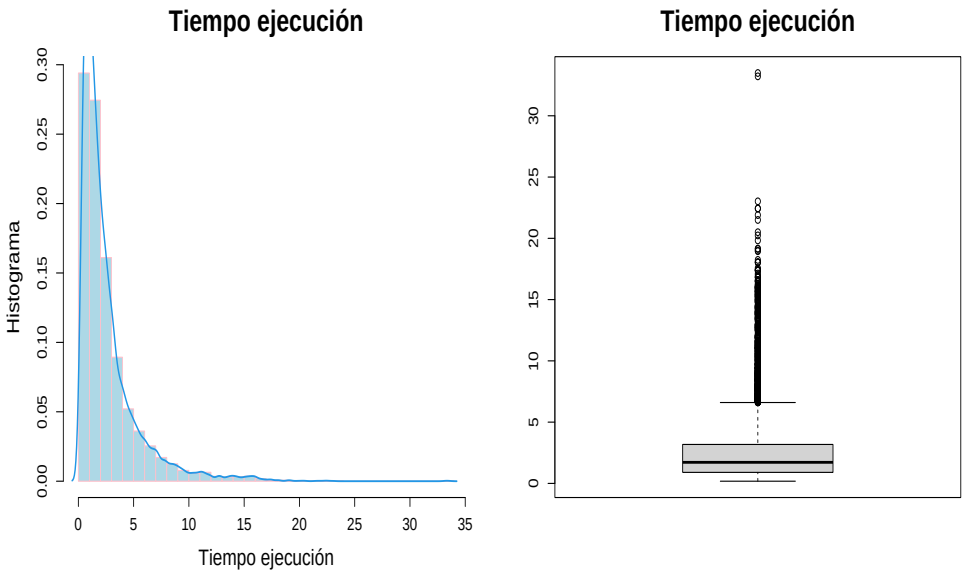


Figura 2.17: Representaciones de la distribución del tiempo de ejecución

Se aprecia, en ambas representaciones, una clara asimetría de la distribución de la variable. Se procede a aplicar la transformación logarítmica en orden a simetrizar la distribución tal y como se muestra en la Figura 2.18.

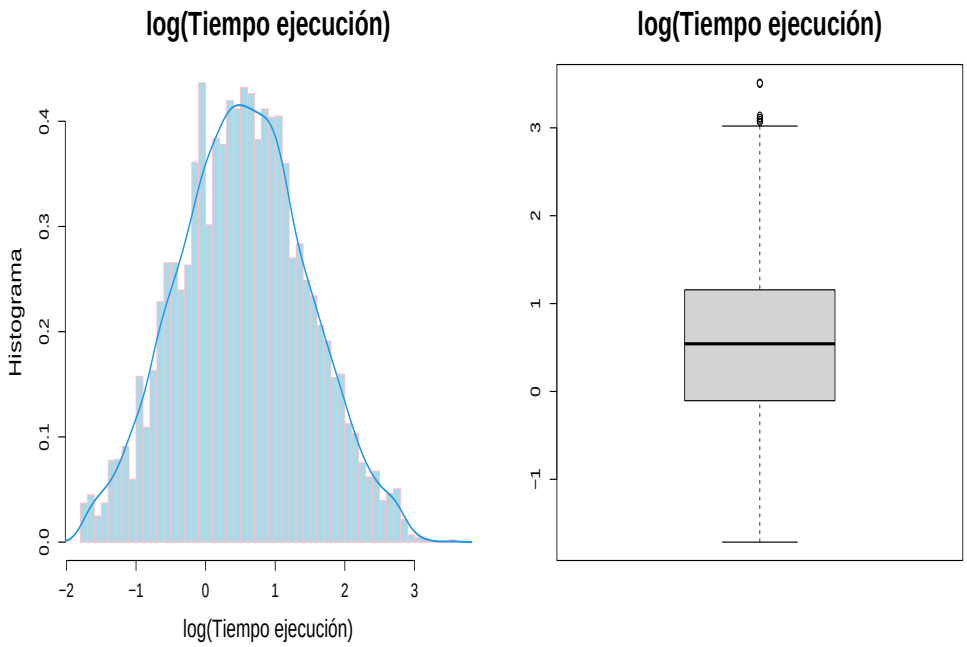


Figura 2.18: Representaciones de la distribución del logaritmo del tiempo de ejecución

Análisis bivariante

En este apartado se muestran diferentes gráficos, en los cuales se enfrentan las distintas variables explicativas y la respuesta (logaritmo del tiempo de ejecución). Para todas ellas, se obtienen diagramas de cajas y estimaciones tipo kernel de la densidad de probabilidad estimada del logaritmo del tiempo de ejecución para cada valor de las diferentes variables explicativas categóricas.

En un principio, se realizan la representaciones relativas a las variables explicativas de nivel 1. El comportamiento en distribución para las cuatro condiciones de tarifa se muestra en la Figura 2.19, apreciándose menor dispersión y un menor tiempo medio de ejecución para la condición de tarifa 'hostelería', identificada por el código '04'. Los resultados según la clasificación de los técnicos por centro y por condición de tarifa se proporcionan en las Figuras 2.20 y 2.21, respectivamente. En el primer caso se observa una progresiva leve disminución del tiempo medio de ejecución al aumentar el nivel de conocimiento del centro, en tanto que no parecen existir diferencias, ni en media ni en dispersión, según el grado de conocimiento de la condición de tarifa que presentan los técnicos (particularmente al comparar las categorías 'Ato' y 'Bajo'). La última de las posibles variables fijas del modelo se encuentra representada en la Figura 2.22, donde se aprecia una disminución progresiva importante en los tiempos medios de ejecución cuando se reduce el número de técnicos trabajando. Esto resulta contra-intuitivo a lo esperado, pero la necesidad de enviar más técnicos puede explicarse por la complejidad de la avería, disponer de uno de apoyo, etc.

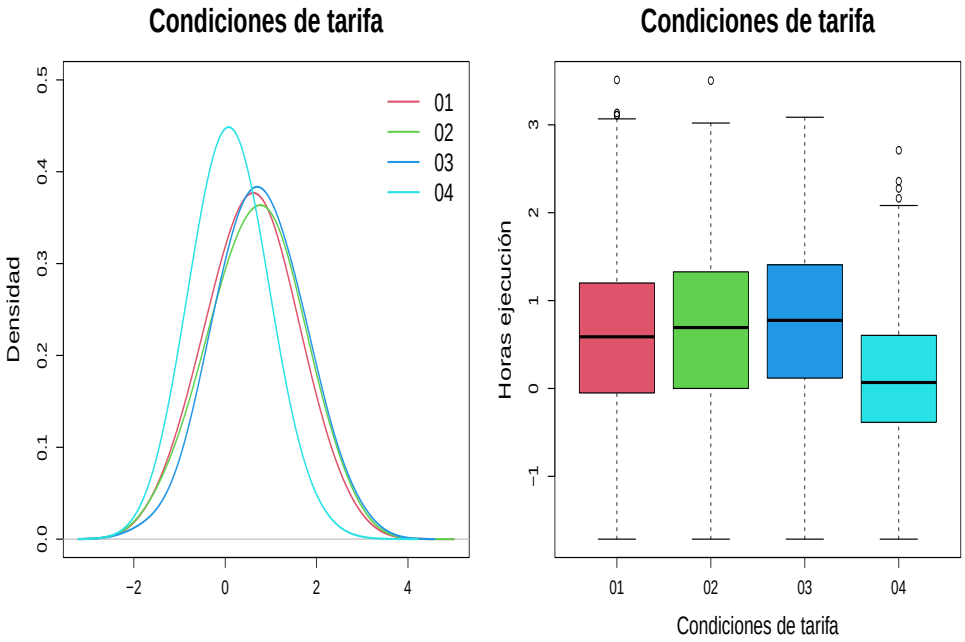


Figura 2.19: Distribuciones estimadas del logaritmo del tiempo de ejecución según cada condición de tarifa.

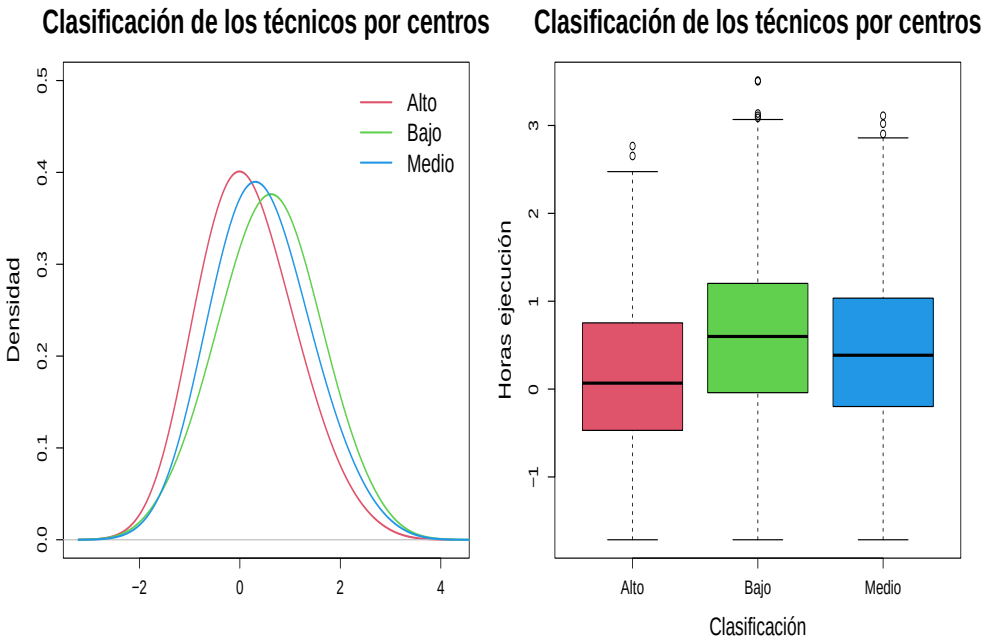


Figura 2.20: Distribuciones estimadas del logaritmo del tiempo de ejecución según cada valor de la variable que clasifica a los técnicos por el conocimiento del centro

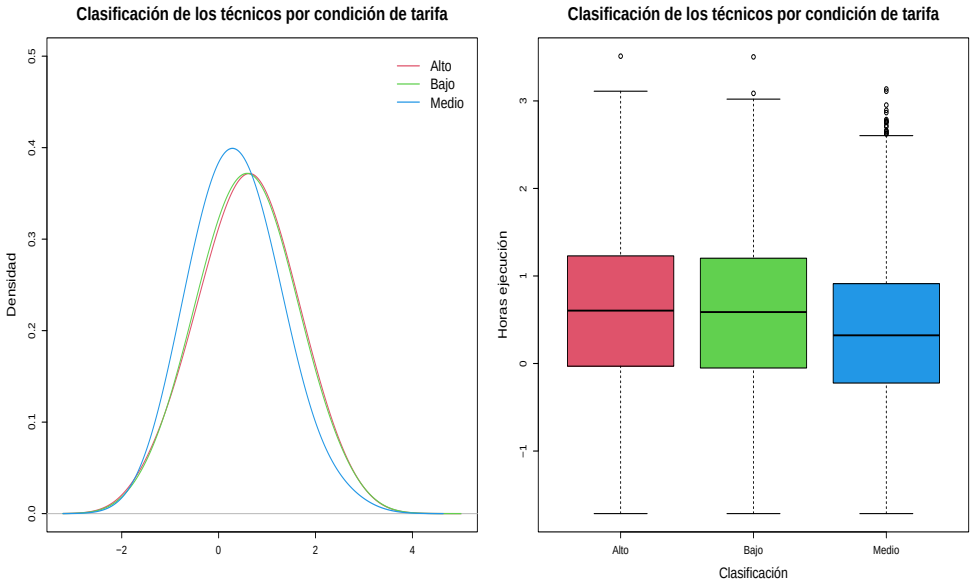


Figura 2.21: Distribuciones estimadas del logaritmo del tiempo de ejecución según cada valor de la variable que clasifica a los técnicos por el conocimiento de la condición de tarifa

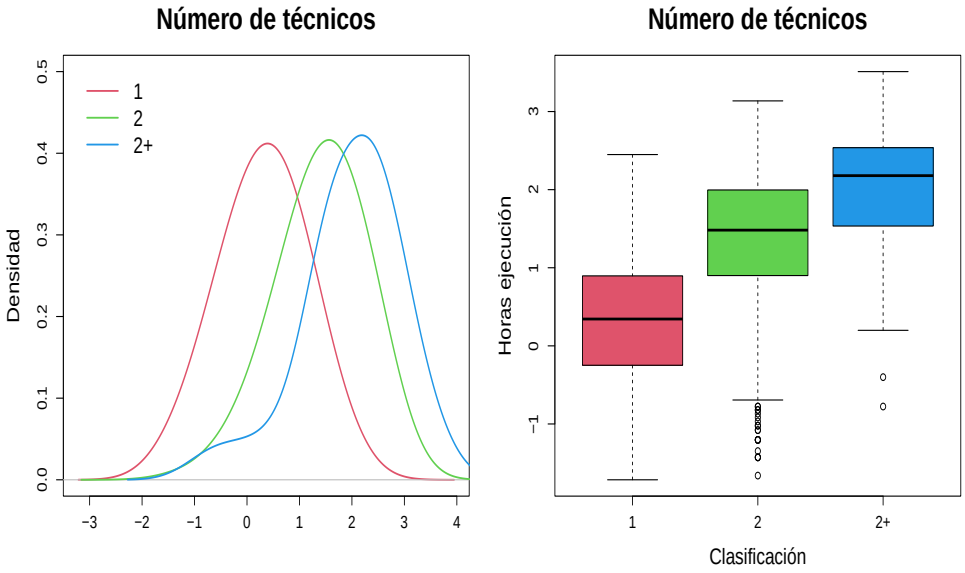


Figura 2.22: Distribuciones estimadas del logaritmo del tiempo de ejecución según cada valor de la variable que indica el número de técnicos

Por otro lado, se realizan las representaciones relativas a las variables de efectos aleatorios. En la Figura 2.23, se pueden observar las relativas a la variable Tipo_equipo, donde existen diferencias en media y varianza. Finalmente, como ya se mencionó, las variables Equipo y Tipo_averia tiene un número bastante elevado de distintos códigos, lo que dificulta la claridad de la representación. Por ello, para ambas variables se representan solo los 11 códigos con mayor número de solicitudes asociadas.

Estas representaciones pueden verse en las Figuras 2.24 y 2.25, observándose claramente que estas variables tampoco son homogéneas.

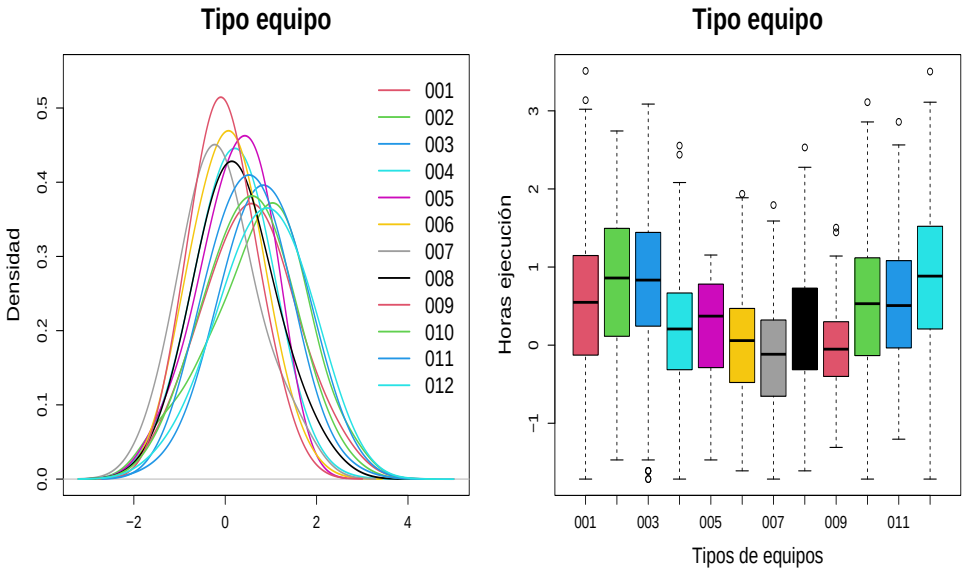


Figura 2.23: Representaciones de las distribuciones del logaritmo del tiempo de ejecución para las categorías de la variable tipos de equipo

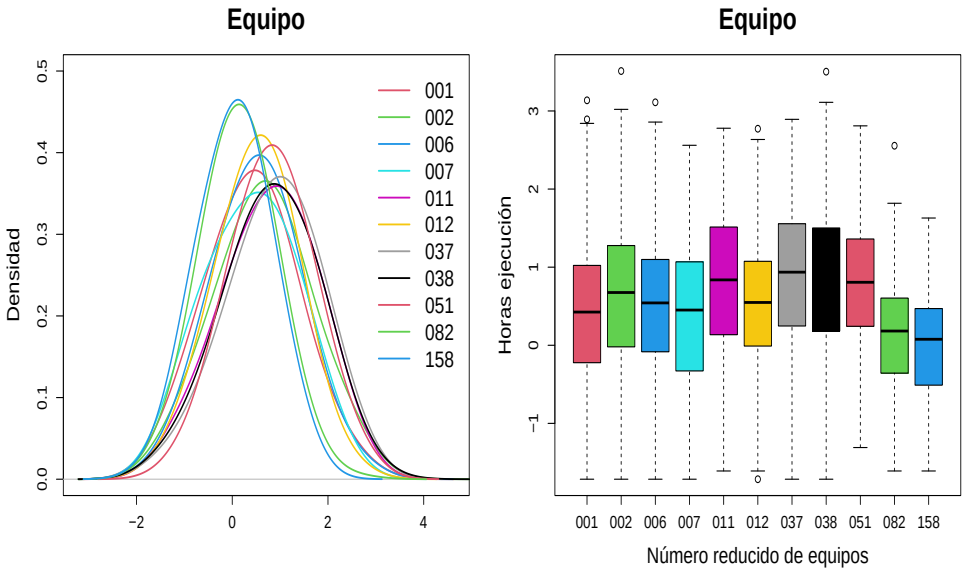


Figura 2.24: Representaciones de las distribuciones del logaritmo del tiempo de ejecución para los 11 códigos de equipo con mayor número de solicitudes

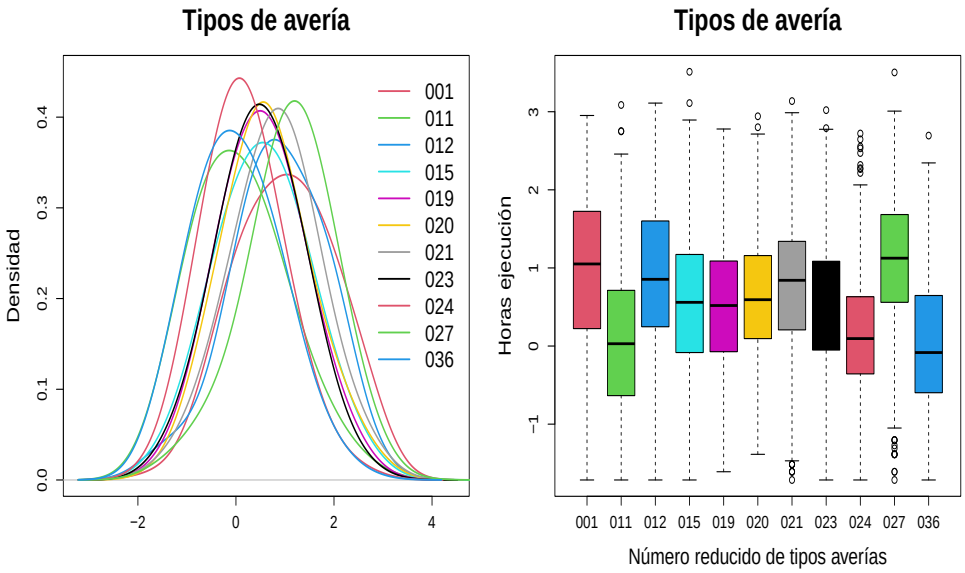


Figura 2.25: Representaciones de las distribuciones del logaritmo del tiempo de ejecución para los 11 códigos de tipos de avería con mayor número de solicitudes

2.3.2. Construcción del modelo

En esta sección, se documenta la construcción del modelo de regresión para tratar de explicar la conducta en media del tiempo de ejecución a partir de las explicativas disponibles. Como en el caso del tiempo de desplazamiento, hay razones para introducir en el modelo las variables Equipo, Tipo_equipo y Tipo_averia como de efectos aleatorios dado que las tres presentan características como:

- Un número bastante elevado de categorías y con un único registro en muchas de ellas.
- Los tiempos de ejecución presentan variabilidad no nula a través de estas categorías.
- De acuerdo con los gráficos en las figuras previas, las distribuciones del tiempo de ejecución condicionado a los diferentes valores de estas variables no parecen mostrar la presencia de grupos homogéneos claramente diferenciables.

Concluimos así la necesidad de que estas variables se incorporen al modelo como una variable factor de efectos aleatorios.

Se plantea un modelo de regresión de efectos mixtos con intercepto aleatorio y siete variables categóricas, cuatro de efectos fijos y tres de efectos aleatorios, que puede formularse como sigue.

$$Y_{hjklr(snm)} = \beta_0 + \sum_{j=2}^{3} \alpha_j I(C_c = j) + \sum_{k=2}^{3} \gamma_k I(C_t = k) + \sum_{l=2}^{3} \delta_l I(N = l) + \sum_{r=2}^{4} \eta_r I(T = r) + u_s + v_n + w_m + \epsilon_h,$$

donde $h = 1, \ldots, r$, con r el número de réplicas, Y representa el logaritmo del tiempo de ejecución, C_c y C_t las variables Clasificación_centro_tecnico y Clasificación_tarifa_tecnico, N refleja Num_tecnicos y T representa Condiciones_tarifa. Las realizaciones $u_s \sim N(0, \sigma_u^2)$, con $s = 1, \ldots, 12$, $v_n \sim N(0, \sigma_v^2)$, con $n = 1, \ldots, 49$ y $w_m \sim N(0, \sigma_w^2)$, con $m = 1, \ldots, 96$, recogen los efectos aleatorios sobre la respuesta inducidos por las variables Equipo, Tipo_equipo y Tipo_averia, respectivamente. Los interceptos aleatorios de los distintos grupos son independientes entre sí e independientes de los errores de primer nivel.

Para chequear la necesidad de considerar un modelo con efectos aleatorios, se realiza un test de razón de verosimilitudes entre el modelo propuesto y aquel que resulta de eliminar la parte aleatoria. Se obtine un p-valor inferior a 0.01 concluyendo el interés en mantener el modelo propuesto. A continuación se aplica el procedimiento secuencial 'backward' para la selección de variables a incluir en el modelo. Se parte de un modelo completo que incorpora todos los efectos que razonablemente pueden llegar a influir en la variable respuesta, y se van eliminando términos con algún criterio, hasta que se llega a la conclusión de que no procede suprimir ningún término más. El criterio empleado es el AIC, es decir, una variable sale del modelo si decrece el AIC del modelo ajustado sin ella. En primer lugar, se chequean las variables que se introducirán en la parte fija y aleatoria del modelo, sin tener en cuenta posibles pendientes aleatorias. De esta forma, se llega a la conclusión de que no es relevante incluir la variable Clasificación_tarifa_tecnico, ni tampoco las categorías '02' y '03' de Condiciones_tarifa. Finalmente, se contrasta la posible introducción de pendientes aleatorias en el modelo. No obstante, como se obtuvieron resultados estadísticamente significativos en todos los casos, no se considera ninguna pendiente aleatoria.

El modelo resultante resulta ser:

$$Y_{hjklr(snm)} = \beta_0 + \sum_{j=2}^{3} \alpha_j I(C_c = j) + \sum_{l=2}^{3} \delta_l I(N = l) + \eta + u_s + v_n + w_m + \epsilon_h,$$

donde los diferentes términos representan los efectos fijos y aleatorios mencionados en el modelo completo formulado con anterioridad.

Tras realizar el ajuste del modelo, se obtuvieron los siguientes resultados:

Parámetro	Estimación	Error típico	p-valor	IC
β_0	0.113	0.068	0.0996	$(-0.022, 0.246)$
α_2	0.121	0.038	0.0016	$(0.046, 0.197)$
α_3	0.141	0.041	0.0005	$(0.061, 0.221)$
δ_2	0.981	0.021	$< 2e^{-16}$	$(0.94, 1.021)$
δ_3	1.504	0.116	$< 2e^{-16}$	$(1.277, 1.731)$
η	-0.189	0.035	$6.21e^{-8}$	$(-0.26, -0.119)$

Cuadro 2.3: Estimaciones de los efectos fijos y intervalos de confianza

Parámetro	Varianza	Desviación típica	IC para σ
Equipo	$\sigma_u^2 = 0.014$	$\sigma_u = 0.118$	$(0.079, 0.169)$
Tipo_averia	$\sigma_v^2 = 0.059$	$\sigma_v = 0.242$	$(0.186, 0.316)$
Tipo_equipo	$\sigma_w^2 = 0.01217$	$\sigma_w = 0.1103$	$(0.044, 0.201)$
Residual	$\sigma_\epsilon^2 = 0.569$	$\sigma_\epsilon = 0.754$	$(0.743, 0.765)$

Cuadro 2.4: Estimaciones y intervalos de confianza de las componentes de la varianza del modelo

En el Cuadro 2.3 se proporcionan la estimación puntual del intercepto medio y de los distintos parámetros que reflejan las diferencias en media con respecto a la categoría de referencia para cada una de las variables de nivel 1. Se observa que todos los parámetros son significativamente distintos de cero, exceptuando el intercepto que no lo es al 5 %. En el Cuadro 2.4 se introducen las estimaciones de la desviación estándar y la varianza del efecto aleatorio asociado a cada una de las tres variables factor y la del error.

De forma análoga a como se procedió en la Sección 2.2.2, con objeto de distinguir claramente la variabilidad entre- e intra-grupos, se puede construir el coeficiente de partición de la varianza. En el caso que nos ocupa, se puede obtener una medida de la proporción total de la varianza atribuible a las diferencias entre grupos para cada uno de los efectos aleatorios y para todos ellos juntos. Así, se obtiene para la variable equipo, VPC= 0.0212 para tipo de avería VPC= 0.0898 y para tipo de equipo VPC= 0.0186. De forma que, un 2.1 % de la variabilidad se explica por diferencias entre los equipos, un 9 % por diferencias en los tipos de averías y un 1.9 % por diferencias en los tipos de equipo. Finalmente, se calcula el coeficiente de partición de la varianza para los 3 efectos aleatorios y se obtiene VPC= 0.1297. De forma que, un 13 % de la variabilidad se debe a diferencias entre las categorías de las variables de nivel 2.

2.3.3. Validación del modelo

Se discute en este apartado el grado de cumplimiento de las hipótesis estructurales del modelo ajustado en orden a validarlo, al tiempo que se evalúa su capacidad predictiva. De nuevo se emplea el paquete HLMdiag [32]. Se comienza por los residuos de nivel 1 y se consideran solo los residuos obtenidos mediante ajuste teniendo en cuenta las predicciones de los efectos aleatorios (EB).

Se procedió a chequear la hipótesis de normalidad gráfica y analíticamente. La Figura 2.26 muestra un histograma de los residuos, junto a los mejores ajustes paramétricos de la familia de Gaussianas y ajustes no paramétricos. Se aprecia claramente la unimodalidad y la simetría. A pesar de que la prueba analítica de Lilliefors resultó significativa, la desviación de la normal no parece excesivamente preocupante. Nótese que se dispone de un tamaño muestral muy elevado, lo que hace particularmente sensibles a las pruebas de bondad de ajuste para detectar desviaciones menores de la nula (i.e. devolver p-valores muy pequeños). Por otro lado, la simetría y unimodalidad mostrado por el gráfico y el hecho en sí de emplear una muestra muy grande valida el proceso de estimación aún en presencia de pequeñas desviaciones de la normalidad.

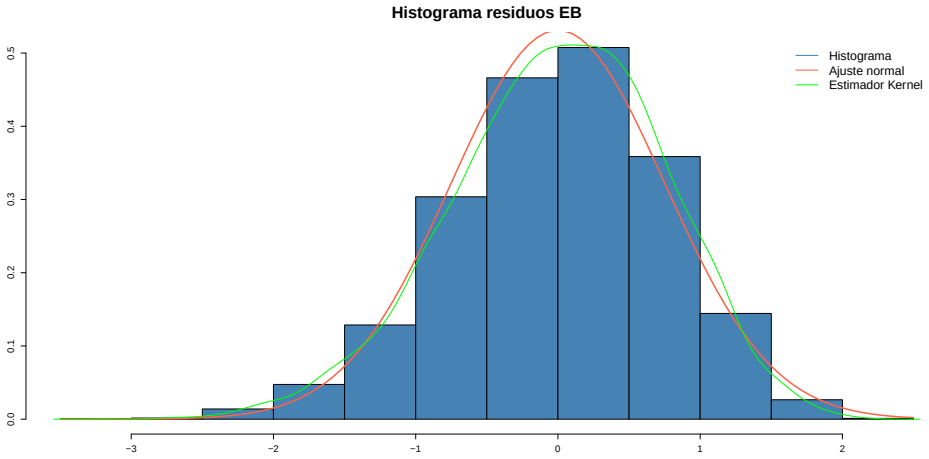


Figura 2.26: Ajuste paramétrico y no paramétrico a la densidad de probabilidad de los residuos de nivel 1

En orden a examinar la hipótesis de linealidad, la Figura 2.27 muestra un diagrama de dispersión de los residuos frente a los valores ajustados y un ajuste no paramétrico de regresión local lineal. La nube de puntos no sugiere patrón alguno tras los residuos y parece razonable aceptar la hipótesis de linealidad. Asimismo, el resultado del test resultante de aplicar la función *sm.regression()* disponible en el paquete *sm* [8] es no estadísticamente significativo.

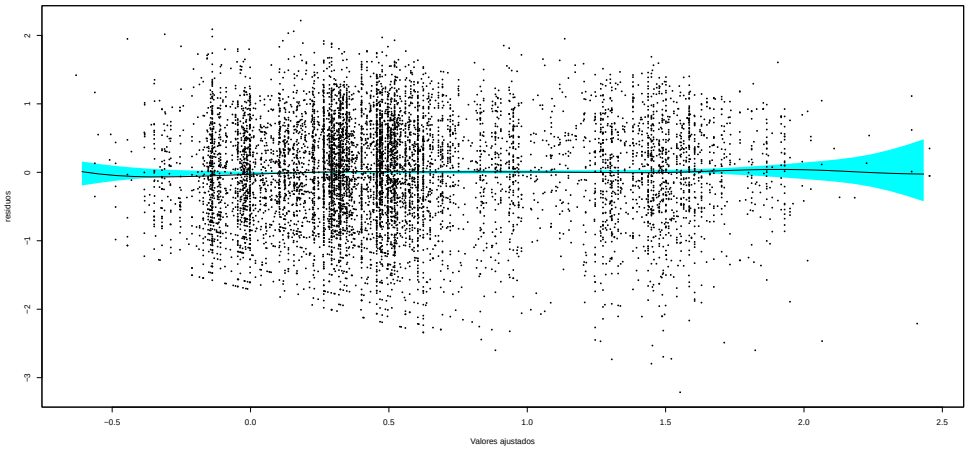


Figura 2.27: Diagrama de dispersión de los residuos frente a los valores ajustados y un ajuste no paramétrico de regresión local lineal

En la Figura 2.28 se muestran diagramas de cajas de los residuos frente a las distintas variables que inducen el efecto grupo al objeto de obtener una intuición acerca de la existencia o no de homocedasticidad. Mientras que para los tipo de equipos no se aprecia heterocedasticidad relevante, para tipo de avería y para equipos existen unas pocas categorías donde el comportamiento en varianza difiere sustancialmente. Cabe subrayar que en estas categorías, por lo general, se dispone de muy pocos registros, lo que impide una valoración precisa del nivel de heterocedasticidad.

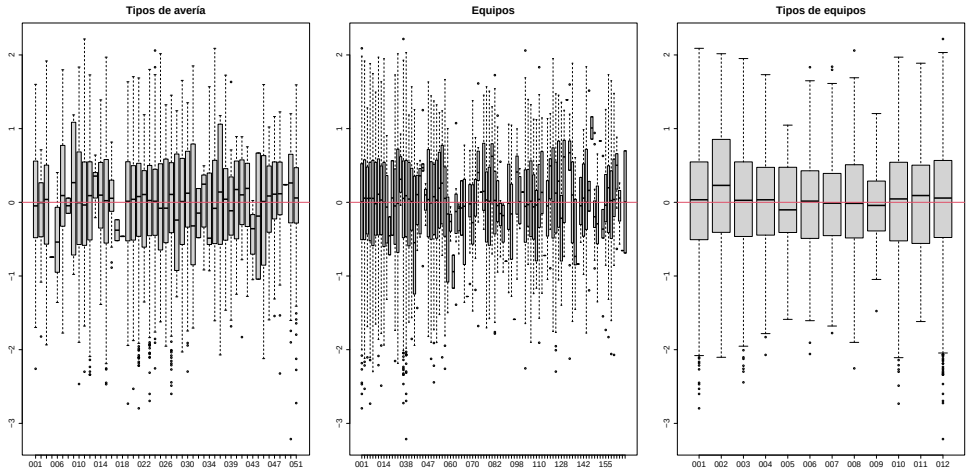


Figura 2.28: Diagrama de cajas de los residuos frente a las distintas variables grupo

A continuación se extraen los residuos de nivel 2 y se contrasta la hipótesis de normalidad en este nivel. Las pruebas de bondad de ajuste a una normal de Lilliefors resultaron en todos los casos no significativas al 5 %. A efectos ilustrativos, la Figura 2.29 muestra un Q-Q plot para las predicciones de cada una de las variables que inducen el efecto grupo.

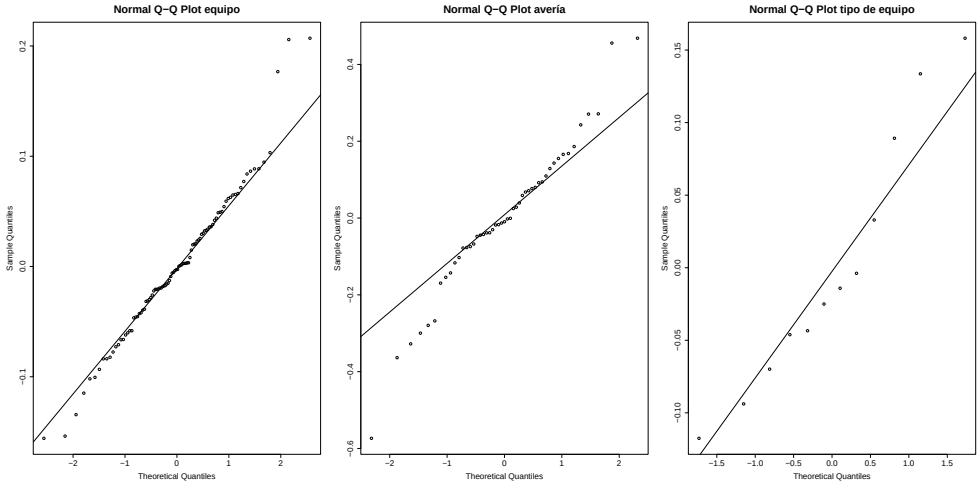


Figura 2.29: Q-Qplot de los residuos de nivel 2

Se chequeó la hipótesis de incorrelación (independencia en contexto de normalidad) tanto para los residuos de nivel 1 como para los de nivel 2. Se aplicó la función *RunsTest* o *runs.test* para realizar un test de rachas. Con ambas funciones se acepta la hipótesis nula.

Por último se realizó un análisis de influencia. Se procedió con la eliminación de observaciones influyentes a nivel individual y a nivel grupo. No obstante, en ningún caso se produce una mejora significativa en el comportamiento de los residuos ni en la capacidad predictiva del modelo.

Para determinar la capacidad predictiva del modelo ajustado, se emplea nuevamente la función *r.squaredGLMM* y se obtiene que el R^2 marginal alcanza un valor de 0.2111456 y el R^2 condicional

0.3134802. Es evidente que la capacidad predictiva, en este caso, es mucho menor que la obtenida para el modelo del tiempo de desplazamiento. Claramente, las variables explicativas disponibles de la base de datos de la empresa no son suficientes para poder predecir con una precisión razonable el tiempo de ejecución. Es necesario disponer de otras variables que sean más informativas a estos efectos. Más aún, el análisis detallado de los datos permitió detectar discordancias en los mismos que pueden ser catalogadas como anomalías de los registros. Por ejemplo, se han observado registros absolutamente dispares dentro de una misma categoría de varias variables explicativas. Por absolutamente dispares se ha de entender que son muy poco razonables y, seguramente, fruto de errores en la introducción de los datos o bien de efectos de variables 'ocultas' (no controladas en la base de datos disponible). No obstante, pese a la baja capacidad predictiva del modelo, la empresa deseaba disponer del modelo ajustado con las variables disponibles.

2.3.4. Predicciones

Como en el caso del tiempo de desplazamiento, Cofrico está muy interesado en disponer de predicciones e intervalos de predicción para los tiempos de ejecución, así como intervalos de confianza para esos tiempos en media. No obstante, en este caso, se optó por presentar a la empresa solamente los intervalos de confianza para los tiempos medios de ejecución. Para ello, se empleó de nuevo la función que usa métodos de remuestreo, bootMer(), que, como ya se mencionó, controla la variabilidad originada por la varianza residual, la incertidumbre en la estimación de los coeficientes fijos y la incertidumbre en la estimación de las componentes de la varianza. La renuncia a proporcionar los intervalos de predicción responde a la baja capacidad predictiva del modelo que resulta en intervalos de predicción altamente imprecisos, con una longitud media de casi nueve horas y, consecuentemente, de interés real nulo.

En la Figura 2.30 se representan los intervalos de confianza al 95 % para el tiempo medio de ejecución para los valores de las explicativas de las 45 primeras observaciones del conjunto de datos.

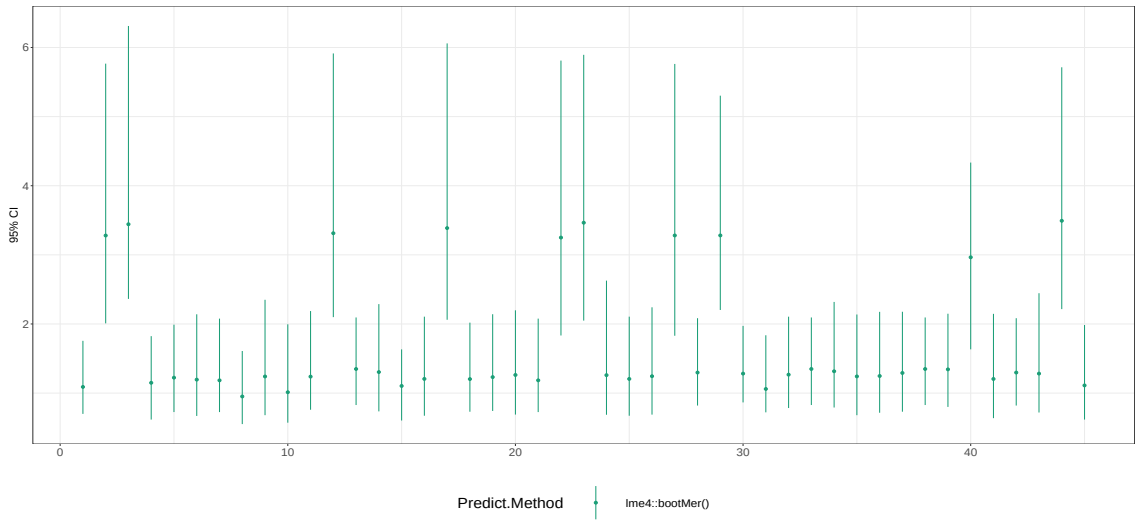


Figura 2.30: Representaciones de los intervalos de confianza al 95 % obtenidos con la función *bootMer()*

Capítulo 3

Modelos en producción

En este capítulo, se describe la herramienta desarrollada para Cofrico al objeto de poder aplicar los dos modelos predictivos creados. Es decir, el objetivo final es poner en producción ambos modelos para permitir a la empresa emplear los resultados obtenidos de forma eficiente.

Con este fin, en primer lugar se emplea el paquete *plumber* para crear una API para cada uno de los modelos. Posteriormente, se llevarán las mismas a contenedores Docker, lo cual permitirá ejecutarlas en cualquier entorno.

3.1. API

Una API [12, 44] o Application Programming Interface, que en español quiere decir Interfaz de Programación de Aplicaciones, es un conjunto de funciones y procedimientos que permite integrar sistemas, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas. Sirven para intercambiar datos entre diferentes tipos de software y así automatizar procedimientos y desarrollar nuevas funcionalidades. Es decir, una API es la herramienta que permite que un software se comunique o interactúe con otro. La importancia de las API radica en que permiten a diferentes programas, dispositivos y aplicaciones trabajar en conjunto.

A continuación, para cada uno de los modelos, se procede con la creación de una API web empleando R. Las API web son un marco común para compartir información a través de una red, más comúnmente a través de HTTP. Como se mencionó, el paquete *plumber* [45] permite realizar esta tarea.

3.1.1. Creación de una API con *plumber*

El objetivo final de esta sección es la creación de una API web con *plumber* para cada modelo [13, 22, 49]. Antes de comenzar, mencionar que un API puede tener uno o varios endpoints, donde un endpoint es el punto de contacto con el servicio. Por otro lado, en estos casos lo más común es emplear los métodos de petición HTTP que definen la acción que se realizará, entre ellos:

- GET: Permite consultar datos.
- POST: Permite crear o enviar datos.
- PUT: Permite actualizar datos.
- DELETE: Permite eliminar datos.

En el caso que nos ocupa, las API serán creadas con un único endpoint y se emplea el método GET.

Para la creación de una API empleando el paquete *plumber* se crea un archivo.R donde se incluye una serie de comentarios introducidos con #*, que definen el comportamiento de la API, y una función que use los parámetros de entrada como argumentos de la misma.

En la parte comentada, se añade el título y descripción de la API, los parámetros de entrada que en este caso son los datos del modelo y, finalmente, el método y el nombre del endpoint. Es relevante mencionar que plumber devuelve datos en formato JSON de forma predeterminada. No obstante, este formato puede ser modificado empleando '@serializer', si así se desea. Como ejemplo, a continuación se pueden ver los comentarios requeridos para la creación de la API que proporcionará las predicciones del tiempo de ejecución:

```
#* @apiTitle API de predicción del tiempo de ejecución
#* @apiDescription
#* Esta api servirá para predecir el tiempo de ejecución
#* dadas las variables número de técnicos, condición de tarifa, tipo de avería, tipo
#* de equipo y tipo de servicio. Además, se proporcionan los resultados para las tres
#* categorías de la variable que clasifica los técnicos en función del conocimiento
#* del centro: Bajo, Medio, Alto
#* @param Tipo_Equipo Código del tipo de equipo
#* @param Tipo_averia Código de la avería
#* @param Tipo_Servicio Código del tipo de servicio
#* @param Num_tecnicos Número de técnicos que acuden
#* @param Condiciones_tarifa Código de la condición de tarifa
#* @get /Predicciones/Solicitudes/Ejecucion
```

Destacar que se introducen las variables de acuerdo a su clasificación en la actual base de datos y luego, en el código, se realiza un cambio de variable para poder emplear el modelo construido con los registros de las solicitudes disponibles en la anterior. Esto se realiza con el fin de poder emplear las resultantes predicciones conforme a las variables disponibles en la base de datos actual.

A continuación, se crea la función que usa los parámetros de entrada como argumentos de la misma, la cual hará uso del modelo creado para devolver las distintas predicciones. Finalmente, para levantar la API es necesario ejecutar las siguientes líneas de código:

```
r <- plumb("API.R")
r$run(port=8001,host="0.0.0.0")
```

Donde 'API.R' es el archivo creado siguiendo los pasos previamente mencionados. La función *plumb* procesa una API de *plumber* y la segunda línea es donde se inicia el servidor. Por último, introduciendo la url http://127.0.0.1:8001/__docs__/ en cualquier navegador web, aparecerá la interface swagger, herramienta que facilita la generación de APIs, la cual permite, de forma sencilla, la ejecución de la función añadida a la API. Puede observarse la misma, para el caso concreto del modelo de tiempo de ejecución, en las Figuras 3.1 y 3.2.

Figura 3.1: Interface swagger para el modelo de tiempo de ejecución

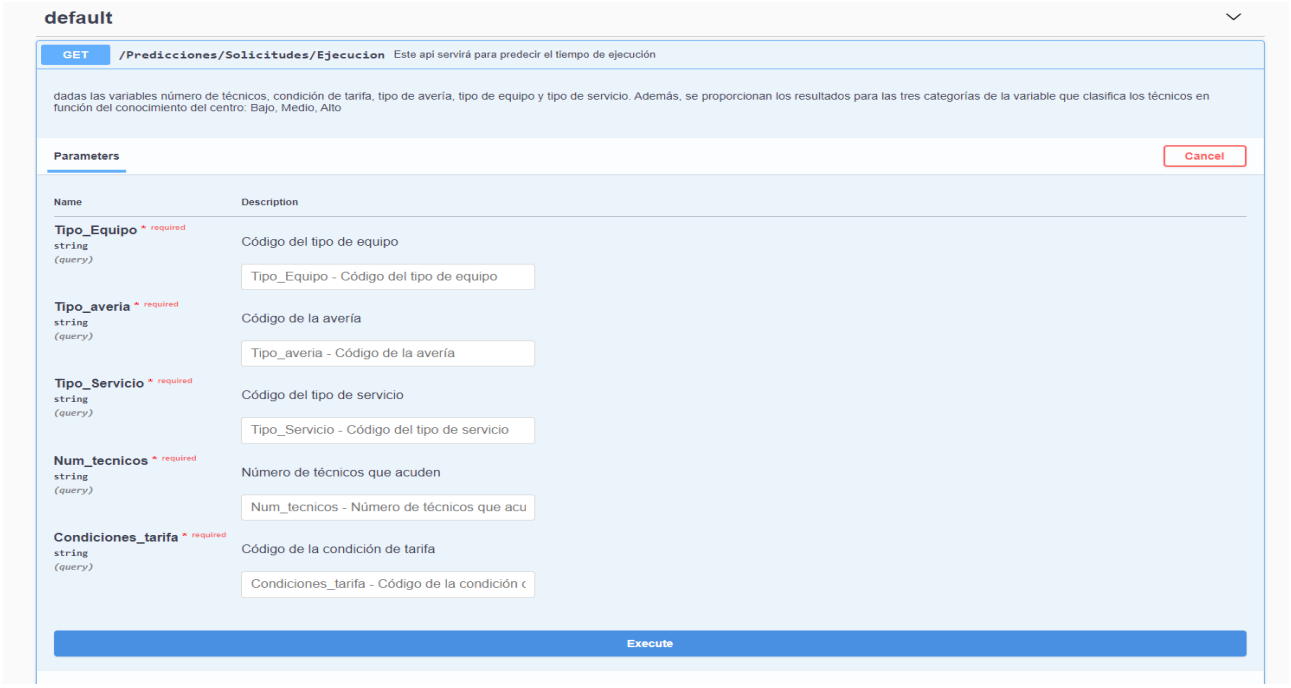


Figura 3.2: Interface swagger para el modelo de tiempo de ejecución

Un problema añadido, que en este caso es necesario tener presente, es el uso de la función *bootMer* que, como ya se mencionó, es bastante costosa computacionalmente. Esto deriva en que es necesario

esperar para poder obtener los resultados. Por otro lado, el código en R es síncrono. Es decir, las funciones se ejecutan de forma secuencial, de forma que si se envían varias solicitudes se tardaría mucho tiempo en recibir los resultados. Para solventar este problema se dispone de los paquetes *future* [6] y *promises* [14], que permiten indicar a R que una función específica debe ejecutarse de forma asíncrona. Además, con el objetivo de reducir el tiempo de espera por los resultados, se emplea el paquete *memoise* [54], que facilita la memorización de funciones y permite almacenar en caché las llamadas a las mismas para que, si se repiten los parámetros, se pueda devolver la salida guardada directamente.

3.2. Docker

Docker [2, 41, 47] es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización a nivel de sistema operativo en Linux. La principal diferencia de Docker con los modelos tradicionales de virtualización es que utiliza los contenedores en vez de las máquinas virtuales.

Los contenedores son un paquete de elementos que permite crear un entorno donde correr aplicaciones independientemente del sistema operativo. Es decir, son unidades estándar de software que empaqueta código y todas sus dependencias de manera que la aplicación se ejecuta rápidamente y de forma fiable bajo múltiples entornos de ejecución.

Una imagen Docker es un paquete ligero, independiente y ejecutable que incluye absolutamente todo lo necesario para poder ejecutar una aplicación. Por lo general, las imágenes se basan en otras ya existentes con configuraciones adicionales. Docker Hub provee una fuente de recursos centralizada para el descubrimiento y distribución de imágenes de contenedores que fomenta la colaboración entre usuarios.

Por otro lado, mencionar que en Docker también existe el término de volúmenes, los cuales ofrecen un almacenamiento persistente completamente gestionado por Docker. Existen distintos tipos: los volúmenes de host, los anónimos y los nombrados.

Teniendo todo esto en consideración, el siguiente paso, una vez se tiene creada una API para cada uno de los modelos, es montar las mismas en Docker. Es decir, se pretende encapsular todo el entorno de trabajo en contenedores Docker. Esto permite asegurar que los archivos se ejecutarán siempre con la misma configuración sobre la que se han hecho todas las pruebas.

3.2.1. Despliege de las APIs dentro de contenedores Docker

La creación de un contenedor [1] siempre empieza con la elaboración de un fichero de texto nombrando Dockerfile, a partir del cual se creará la imagen empleando el comando ‘docker build’, que toma las instrucciones que aparecen en el fichero y las va ejecutando.

Por lo general, un Dockerfile comienza con la especificación de una imagen de partida. A continuación, los principales pasos a seguir son la instalación de los diversos paquetes y capas que puedan ser necesarias para la aplicación, se copian los archivos necesarios dentro de la imagen y, finalmente, se especifica el punto de entrada del contenedor. Para definir un Dockerfile existe una multitud de comandos, entre ellos:

- FROM: Define una imagen base de partida.
- COPY/ADD: No son exactamente los mismo, pero tiene una funcionalidad similar. Permiten copiar ficheros dentro de la imagen del contenedor.

- RUN: Añade capas a la imagen base, instalando aplicaciones, librerías y componentes.
- CMD: Especifica qué comandos se deben ejecutar al iniciarse el contenedor.
- ENTRYPOINT: Es la instrucción 'principal' de un Dockerfile. Especifica el punto de entrada de un contenedor que se quiere que funcione como un ejecutable.
- WORKDIR: Especifica el directorio.

En la Figura 3.3 se puede observar un ejemplo de archivo Dockerfile creado para levantar una de las APIs. Se parte de una imagen inicial con la versión de R necesaria, obtenida del repositorio de Docker Hub. Luego, se instalan los paquetes necesarios, se indica el directorio y se copian todos los scripts y resto de archivos dentro de la imagen. Finalmente, se especifica el punto de entrada.

```
FROM rocker/r-ver:4.2.3

RUN R -e "install.packages('memoise');install.packages('promises');
        install.packages('future');install.packages('fastmap');
        install.packages('plumber');install.packages('jsonlite');
        install.packages('lme4');install.packages('lmerTest');
        install.packages('readxl');install.packages('dplyr')"

WORKDIR /app

COPY ./Source/ .

EXPOSE 8001

ENTRYPOINT ["Rscript", "/app/ejecutar.R"]
```

Figura 3.3: Ejemplo de archivo Dockerfile

Para montar las dos APIs creadas en cotenedores Docker es necesario crear una imagen, siguiendo los pasos previamente comentados para cada una de ellas. Resulta también interesante poder automatizar de esta forma la creación de los modelos. Por consiguiente, se crean 2 imágenes y se levanta un contenedor que crea el modelo y lo guarda en un volumen nombrado, el cual es necesario crear previamente empleando el comando 'docker volume create'. Posteriormente, se corre otro contenedor que usa el modelo guardado en el volumen para crear la API. Claramente, es necesario seguir estos pasos para ambos modelos. El objetivo final no es ejecutar ambos contenedores simultáneamente, de ser el caso lo recomendable sería usar Docker Compose, que se introducirá brevemente en la Sección 3.2.2

Para finalizar, mencionar una serie de pequeños detalles relacionados con los contenedores. Estos se ejecutan con el comando 'docker run'. Además, las operaciones básicas sobre contenedores son: crear, iniciar, parar, mover o eliminar.

3.2.2. Docker Compose

Docker Compose es una herramienta que permite definir y corre aplicaciones multicontenedor dentro de Docker. Esta herramienta te permite configurar los servicios de tu aplicación basándose en el formato de ficheros YAML y después desplegarlos con un sólo comando. Para utilizar Docker Compose hay que seguir los siguientes pasos:

1. Definir el entorno de tu aplicación en un archivo 'Dockerfile'.
2. Definir los servicios de los que se compone tu aplicación en un archivo '.yml'.
3. Ejecutar el comando 'docker compose up'.

Como se mencionó, en este caso no sería necesario el uso de Docker Compose. No obstante, aun cuando la intención no es correr aplicaciones multicontenedor, esta herramienta puede ser de utilidad ya que permite levantar el contenedor con un sencillo comando. En la Figura 3.4 se puede ver un ejemplo de archivo 'docker-compose.yml'. Se puede observar que lo primero que se indica es el nombre del contenedor y de la imagen. Luego, es necesario indicar en que puerto se expone la API y los volúmenes: el que contiene guardado el modelo y el que almacena la memoria caché. Finalmente, en el apartado 'volumes' se indica el nombre de los mismos y que estos fueron creado externamente. Esto te permite levantar el contenedor con el comando 'docker-compose up'. En caso contrario, se tendría que escribir todo lo indicado en el fichero cada vez que se quiere levantar ese contenedor, usando:

'docker run -d -p 8080:8001 –name contenedor-plumber -v modelo-ejecucion:/app/modelo:ro -v cache-ejecucion:/app/Rcache imagen-plumber-ejecucion'

```
services:
  modelo:
    container_name: contenedor-plumber
    image: imagen-plumber-ejecucion
    ports:
       - "8080:8001"
    volumes:
      - cache-ejecucion:/app/Rcache
      - modelo-ejecucion:/app/modelo:ro
volumes:
    cache-ejecucion:
       external: true
       name: cache-ejecucion
    modelo-ejecucion:
       external: true
       name: modelo-ejecucion
```

Figura 3.4: Ejemplo de archivo docker-compose

Capítulo 4

Conclusiones

Este último capítulo pretende enfatizar las contribuciones de este Trabajo de Fin de Máster. Por un lado, se comentan las ventajas atribuidas a la herramienta de visualización aplicada a Business Intelligence: Power BI. Por otro, se resumen las conclusiones obtenidas tras la construcción de los dos modelos predictivos.

Al hacer uso de la herramienta Power BI para la realización de los distintos informes, se observó que esta herramienta permite analizar una gran cantidad de datos de forma clara y sencilla. Además, una vez está terminado el informe, a la vista de los gráficos se pueden extraer conclusiones apropiadas para la ayuda en la toma de decisiones. Se pueden identificar datos atípicos y/o erróneos, lo que faculta su corrección rápidamente. Asimismo, se puede observar la evolución de los distintos indicadores en diferentes períodos temporales y establecer patrones o anomalías, lo que permite a los trabajadores de la empresa actuar en base a esta información. Por otro lado, a partir de los informes se pueden crear paneles que engloben visualizaciones de varios de ellos y, así, tener una visión más general. Por consiguiente, Power BI permite aumentar la productividad de formas rápida, sencilla y sin necesidad de programación. Todos estos argumentos justifican su uso cada vez más extendido, no solo en Cofrico, sino en muchas otras empresas.

El proceso de análisis de los datos y de modelización de los tiempos de desplazamiento y de ejecución permitió extraer conocimiento de la información registrada por la empresa y evaluar su validez para alcanzar niveles de predicción precisos. En primer lugar, el análisis exploratorio preliminar nos llevó a concluir que algunas de las variables registradas por la empresa no resultaba de gran valor para explicar el comportamiento de las variables de interés. Un análisis cluster motivado por la idea de reducir el rango de muchas de las variables categóricas disponibles (algunas tomando un número excesivamente elevado de valores) sin perder información resultó infructuoso.

Se emplearon modelos de regresión con efectos aleatorios para dotar a la empresa de una herramienta que le permita disponer de intervalos para las predicciones de las dos variables respuesta de interés (tiempo de desplazamiento y tiempo de ejecución). Las conclusiones obtenidas a partir de cada uno de los modelos ajustados fueron:

- Tiempo de desplazamiento: La variable explicativa kilómetros refleja muy bien el tiempo a predecir y la variable Cod_centro ayuda añadiendo la variabilidad incorporada por aquellos centros situados en zonas con alta concentración de tráfico, que presentan la necesidad de desplazarse a pie, etc. Esto se tradujo en una alta capacidad predictiva del modelo. Por tanto, en este caso los intervalos obtenidos para las predicciones son altamente precisos.
- Tiempo de ejecución: A diferencia de lo observado con el tiempo de desplazamiento, en este modelo las variables disponibles no resultaron ser tan informativas para predecir de manera

precisa los tiempo de ejecución. El mejor modelo ajustado exhibió una baja capacidad predictiva e intervalos de mayor longitud. Este resultado justifica la necesidad de nuevas variables que añadan mayor información para predecir mejor estos tiempos y, quizás también, por una reorganización en la clasificación de los activos a intervenir. No obstante, se realizó su construcción debido al interés de la empresa en ponerlo en producción.

Por último, un valor añadido del trabajo, de gran utilidad para la empresa, era la construcción de APIs una vez obtenido el ajuste del modelo. Esto permite disponer de las distintas predicciones requeridas sin necesidad de ejecutar ningún script en R.

Apéndice A

Código

En este apéndice se incluye parte del código empleado en R para dar respuesta a los objetivos aquí planteados. Por un lado, se muestra el código relativo a la creación de las dos APIs empleando el paquete *plumber*. Por otro, parte del código usado en la realización del análisis cluster.

En las siguientes líneas de código se observa el archivo.R relativo a la creación de la API para el tiempos de ejecución. Como se mencionó en la Sección 3.1, este archivo debe incluir una parte comentada con #* y una función cuyos argumentos son los parámetros de entrada de la API. Además, se puede observar que primero se memoriza la función empleando el paquete *memoise* y se indica que esa función debe ejecutarse en código asíncrono. En este caso, se muestran solo los intervalos de confianza para las tres categorías de la variable Clasificación_centro_tecnico.

```
#Modelo
#modelo <- lmer(logtiempo ~ Clasificación_centro_tecnico + Num_tecnicos +
                     I(Condiciones_tarifa=="04") + (1|Equipo) + (1|Tipo_averia) +
                     (1|Tipo_equipo),data=datos_ejec)


library(memoise)
library(readxl)
library(dplyr)

dir <- cache_filesystem("Rcache")

calculo <- function(Tipo_Equipo, Tipo_averia, Tipo_Servicio, Num_tecnicos, Condiciones_tarifa){

     codigos <- as.data.frame(read_xlsx("enlace.xlsx"))
     respectivos <- filter(codigos, Tipo_Servicio_lista == as.factor(Tipo_Servicio),
                           Tipo_Equipo_lista == as.factor(Tipo_Equipo))
     Equipo <- respectivos$Equipo
     Tipo_equipo <- respectivos$Tipo_equipo

     load("modelo/modelo.RData")


     data <- data.frame(Equipo = as.factor(Equipo), Tipo_averia = as.factor(Tipo_averia),
       Tipo_equipo = as.factor(Tipo_equipo),
       Clasificación_centro_tecnico = c("Bajo","Medio","Alto"),
```

```
      Num_tecnicos = as.factor(Num_tecnicos),
      Condiciones_tarifa = as.factor(Condiciones_tarifa))


     mySumm <- function(.) {
         predict(., newdata= data, re.form=NULL)
     }

     sumBoot <- function(merBoot) {
        return(
           data.frame(fit = apply(merBoot$t, 2, function(x) as.numeric(quantile(x,
                        probs=.5, na.rm=TRUE))),
             lwr = apply(merBoot$t, 2, function(x) as.numeric(quantile(x,
                         probs=.025, na.rm=TRUE))),
             upr = apply(merBoot$t, 2, function(x) as.numeric(quantile(x,
                         probs=.975, na.rm=TRUE)))
           )
        )
     }


     boot <- bootMer(modelo, mySumm, nsim = 100, seed=2105, use.u=FALSE, type="parametric")

     out <- sumBoot(boot)
}

calculo_mem <- memoise(calculo, cache=dir)

library(promises)
library(future)
library(jsonlite)
library(plumber)
library(lme4)
plan(multicore)

#* @apiTitle API de predicción del tiempo de ejecución
#* @apiDescription
#* Esta api servirá para predecir el tiempo de ejecución
#* dadas las variables número de técnicos, condición de tarifa, tipo de avería,
#* tipo de equipo y tipo de servicio. Además, se proporcionan los resultados
#* para las tres categorías de la variable que clasifica los técnicos en función
#* del conocimiento del centro: Bajo, Medio, Alto
#* @param Tipo_Equipo Código del tipo de equipo
#* @param Tipo_averia Código de la avería
#* @param Tipo_Servicio Código del tipo de servicio
#* @param Num_tecnicos Número de técnicos que acuden
#* @param Condiciones_tarifa Código de la condición de tarifa
#* @get /Predicciones/Solicitudes/Ejecucion



function(Tipo_Equipo, Tipo_averia, Tipo_Servicio, Num_tecnicos, Condiciones_tarifa){
```

```
     e <- environment(calculo_mem)
     future_promise({
          environment(calculo_mem) <- e
          out <- calculo_mem(as.factor(Tipo_Equipo),as.factor(Tipo_averia),
                              as.factor(Tipo_Servicio), as.factor(Num_tecnicos),
                              as.factor(Condiciones_tarifa))


       return(list(Bajo =  list(Predicción = exp(out$fit[1]),
        Intervalo_superior = exp(out$upr[1]),
        Intervalo_inferior = exp(out$lwr[1])),
        Medio = list(Predicción= exp(out$fit[2]),
        Intervalo_superior = exp(out$upr[2]),
        Intervalo_inferior = exp(out$lwr[2])),
        Alto = list(Predicción = exp(out$fit[3]),
        Intervalo_superior = exp(out$upr[3]),
        Intervalo_inferior = exp(out$lwr[3]))))

     })

}
```

A continuación, se observa el archivo.R relativo a la creación de la API para el tiempo de desplazamiento. Se puede observar que este archivo tiene la misma estructura que el anterior. Además, en este caso se muestran tanto los intervalos de confianza como los de predicción.

```
#Modelo
#modelo <- lmer(logtiempo ~ logkm + (1 + logkm |Cod_centro),data=datos_desp)


library(memoise)

dir <- cache_filesystem("Rcache")

calculo.boot <- function(km, Cod_centro){

    logkm <- log(km)

    load("modelo/modelo.RData")

    data <- data.frame(logkm = as.numeric(logkm), Cod_centro = as.factor(Cod_centro))

    set.seed(2105)
    mySumm <- function(.) {
        predict(., newdata= data, re.form=NULL)
    }
    sumBoot <- function(merBoot) {
       return(
         data.frame(fit = apply(merBoot$t, 2, function(x) as.numeric(quantile(x,
                      probs=.5, na.rm=TRUE))),
```

```
            lwr = apply(merBoot$t, 2, function(x) as.numeric(quantile(x,
                      probs=.025, na.rm=TRUE))),
            upr = apply(merBoot$t, 2, function(x) as.numeric(quantile(x,
                      probs=.975, na.rm=TRUE)))
        )
      )
    }

    boot <- bootMer(model_lme4, mySumm, nsim = 100, use.u=FALSE, type="parametric")
    out.boot <- sumBoot(boot)

}

calculo_mem.boot <- memoise(calculo.boot, cache=dir)


calculo.PI <- function(km, Cod_centro){

    logkm <- log(km)

    load("modelo/modelo.RData")

    data <- data.frame(logkm = as.numeric(logkm), Cod_centro = as.factor(Cod_centro))

    out.PI <- predictInterval(model_lme4, newdata= data, level= 0.95, n.sims = 10000,
                 type="linear.prediction", include.resid.var = TRUE, seed=2105)

}

calculo_mem.PI <- memoise(calculo.PI, cache=dir)


library(promises)
library(future)
library(jsonlite)
library(plumber)
library(merTools)
library(lme4)
plan(multicore)

#* @apiTitle API de predicción del tiempo de desplazamiento
#* @apiDescription
#* Esta api servirá para predecir el tiempo de desplazamiento
#* dados los kilómetros y el código de centro
#* @param km kilómetros
#* @param Cod_centro Código de centro
#* @get /Predicciones/Solicitudes/Desplazamiento


function(km, Cod_centro){
```

```
    e.boot <- environment(calculo_mem.boot)
    e.PI <- environment(calculo_mem.PI)
    future_promise({
         environment(calculo_mem.boot) <- e.boot
         environment(calculo_mem.PI) <- e.PI
         out.boot <- calculo_mem.boot(as.numeric(km),as.factor(Cod_centro))
         out.PI <- calculo_mem.PI(as.numeric(km),as.factor(Cod_centro))



       return(list(Intervalos_confianza = list (Predicción = exp(out.boot$fit),
         Intervalo_superior = exp(out.boot$upr),
         Intervalo_inferior = exp(out.boot$lwr)),
         Intervalos_predicción = list(Predicción = exp(out.PI$fit),
         Intervalo_superior = exp(out.PI$upr),
         Intervalo_inferior = exp(out.PI$lwr))))

    })

}
```

Por otro lado, se muestra parte del código usado en la realización del análisis cluster comentado en la Sección 2.1. En primer lugar, se incluye un ejemplo del código relativo a la creación de la matriz de distancias dos a dos necesaria para llevarlo a cabo. La primera matriz se construye empleando la métrica Cramér-von Mises sobre la variable técnico. En la segunda se usa Kullback-Leibler sobre la variable Lin_negocio.

```
library(twosamples)
set.seed(1234)
Tecnicos <- unique(Datos$Técnico)
estT <- matrix(0,nrow = 128,ncol = 128)
for (i in Tecnicos) {
     for (j in Tecnicos) {
         aux1<-total[Datos$Técnico==i,]
         aux2<-tota[Datos$Técnico==j,]
         test <- cvm_test(aux1$TiempoTotal,aux2$TiempoTotal)
         estT[which(Tecnicos==i),which(Tecnicos==j)]=test[1]
   }
}
estT
colnames(estT) <- unique(Datos$Técnico)
rownames(estT) <- unique(Datos$Técnico)


LíneasNegocio <- unique(Datos$Lin_negocio)
library(LaplacesDemon)
set.seed(1234)
est1 <- matrix(0,nrow = 18,ncol = 18)
for (i in LíneasNegocio) {
    for (j in LíneasNegocio) {
        aux1<-total[Datos$Lin_negocio==i,]
        aux2<-total[Datos$LíneLin_negocioaNegocio==j,]
```

```
        test <- KLD((density(aux1$TiempoTotal))$y,(density(aux2$TiempoTotal))$y)
        est1[which(LíneasNegocio==i),which(LíneasNegocio==j)]=test$mean.sum.KLD
    }
}
```

Finalmente, se muestra el bucle creado siguiendo los pasos del algoritmo CLARA para poder realizar el análisis cluster a la variable Cod_centro. No obstante, primero es necesario calcular las submuestras.

```
  total <- Datos
  N<-c(31,30,29,28,27,26,25,24,23,22,21,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1)
  M <- matrix(0,nrow = 56,ncol = 31)
  set.seed(1234)
  for (i in N) {
       objeto <- sample(length(unique(total$Cod_centro)),
                       (length(unique(total$Cod_centro)))/i, replace=FALSE)
       tabla<- table(total$Cod_centro)[objeto]
       M[,which(N==i)] <- names(tabla)
       tabla2<- table(total$Cod_centro)[-objeto]
       total <- filter(total, Cod_centro %in% names(tabla2))
  }
  M #Se obtiene en cada columna una muestra aleatoria de centros

  #Se extraen las 31 submuestras
  submuestras <- list()
  for (k in 1:31) {
     submuestra <- filter(Datos, Cod_centro %in% M[,k])
     submuestras[[k]] <- submuestra

  }
  str(submuestras)



#Se realiza el bucle para las 31 submuestras
library(LaplacesDemon)
library(cluster)
set.seed(123)
distancia = numeric(31)
for (k in 1:31){
   #Se calcula la matriz de distancias dos a dos empleando Kullback-Leiber
   centros <- M[,k]
   est1 <- matrix(0,nrow = 56,ncol = 56)
   for (i in centros) {
     for (j in centros) {
       submuestra <- submuestras[[k]]
       aux1<-submuestra[submuestra$Cod_centro==i,]
       aux2<-submuestra[submuestra$Cod_centro==j,]
       test <- KLD((density(aux1$TiempoTotal,bw=0.4))$y,
                       (density(aux2$TiempoTotal,bw=0.4))$y)
       est1[which(centros==i),which(centros==j)]=test$mean.sum.KLD
     }
   }
```

```
    #Pam y medoids
  d = as.dist(est1)
  medoids <- pam(d, k = 4, diss=TRUE)$medoids

  #Distancia entre todos los centros y los medoids
  centrostotales <- names(table(Datos$Cod_centro))
  medoidsnames <- c(names(table(Datos$Cod_centro)[medoids[1]]),
                    names(table(Datos$Cod_centro)[medoids[2]]),
                    names(table(Datos$Cod_centro)[medoids[3]]),
                    names(table(Datos$Cod_centro)[medoids[4]]))
   est2 <- matrix(0,nrow = 1736,ncol = 4)
   for (i in centrostotales) {
     for (j in medoidsnames) {
        aux1<-Datos[Datos$Cod_centro==i,]
        aux2<-Datos[tDatosotal$Cod_centro==j,]
        test <- KLD((density(aux1$TiempoTotal,bw=0.4))$y,
                        (density(aux2$TiempoTotal,bw=0.4))$y)
        est2[which(centrostotales==i),which(medoidsnames==j)]=test$mean.sum.KLD
      }
    }

  #Se asigna cada centro a un medoids
  min = which(est2 == apply(est2,1,min) , arr.ind = TRUE)
      filas1<-min[min[,2]==1,]
      filas2<-min[min[,2]==2,]
      filas3<-min[min[,2]==3,]
      filas4<-min[min[,2]==4,]

      matriz1<-est2[filas1[,1],]
      matriz2<-est2[filas2[,1],]
      matriz3<-est2[filas3[,1],]
      matriz4<-est2[filas4[,1],]

 #Distancia y distancia total
      dist1 <- apply(matriz1,2,sum)[1]
      dist2 <- apply(matriz2,2,sum)[2]
      dist3 <- apply(matriz3,2,sum)[3]
      dist4 <- apply(matriz4,2,sum)[4]
   distancia[k] <- sum(dist1,dist2,dist3,dist4)

}
```

Bibliografía

[1] Almuiña A (2021). Por qué Docker & R? https://albertoalmuinha.com/es/posts/2020-11-07-docker-and-r/docker_r_es/

[2] Alonso Silva J (2021). Análisis de contenedores Docker y sus implicaciones de seguridad. Universidad politécnica de Madrid.

[3] Baayen RH, Davidson DJ, Bates DM (2008). Mixed-effects modeling with crossed random effects for subjects and items. Journal of Memory and Language 59(4):390-412. https://doi.org/10.1016/j.jml.2007.12.005

[4] Barton K (2023). MuMIn: Multi-Model Inference. R package version 1.47.5. https://CRAN.R-project.org/package=MuMIn

[5] Bates D, Maechler M, Bolker B, Walker S, Bojesen Christensen RH, Singmann H, Bin Dai, Scheipl F, Grothendieck G, Green P, Fox J, Bauer A, Krivitsky P (2023). lme4: Linear Mixed-Effects Models using 'Eigen' and S4. R package version 1.1-32. https://CRAN.R-project.org/package=lme4

[6] Bengtsson H (2023). future: Unified Parallel and Distributed Processing in R for Everyone. R package version 1.32.0. https://CRAN.R-project.org/package=future

[7] Borrajo MI, Saavedra P (2021). Modelos de Regresión. Máster en Técnicas Estadísticas.

[8] Bowman A, Azzalini A (2022). sm: Smoothing Methods for Nonparametric Regression and Density Estimation. R package version 2.2-5.7.1. https://CRAN.R-project.org/package=sm

[9] Bruno C, Videla E, Peña A, Balzarini M (2019). Guía para la construcción de modelos de asociación genómica. Serie Estadística Aplicada. Com. Balzarini M. Brujas. Córdoba, Argentina

[10] Byron Hall B, Martina Hall M, Statisticat, LLC, Brown E, Hermanson R, Charpentier E, Heck D, Laurent S, Gronau QF, Singmann H (2021). LaplacesDemon: Complete Environment for Bayesian Inference. R package version 16.1.6. https://CRAN.R-project.org/package=LaplacesDemon

[11] Caeiro F, Mateus A (2022). randtests: Testing Randomness in R. R package version 1.0.1. https://CRAN.R-project.org/package=randtests

[12] Cámara de Comercio de Bogotá (2019). El mundo conectado por las API. http://hdl.handle.net/11520/22728

[13] Campitelli E (2022). Como crear una API con plumber y hostearla en Heroku. https://eliocamp.github.io/codigo-r/2022/07/plumber-heroku/

[14] Cheng J (2021). promises: Abstractions for Promise-Based Asynchronous Programming. R package version 1.2.0.1. https://CRAN.R-project.org/package=promises

[15] Crujeiras RM (2022). Regresión Generalizada y Modelos Mixtos (Temas 4-5). Máster en Técnicas Estadísticas.

[16] Csorgo S, Faraway J (1996). The exact and asymptotic distributions of cramér-von mises statistics, Journal of the Royal Statistical Society. Series B (Methodological) 58(1):221-234.

[17] DataNovia (2018). CLARA in R: Clustering Large Applications. https://www.datanovia.com/en/lessons/clara-in-r-clustering-large-applications/

[18] DataNovia (2018). K-Medoids in R: Algorithm and Practical Examples. https://www.datanovia.com/en/lessons/k-medoids-in-r-algorithm-and-practical-examples

[19] de Mendiburu F (2021). agricolae: Statistical Procedures for Agricultural Research. R package version 1.3-5. https://CRAN.R-project.org/package=agricolae

[20] Díaz Vásquez RA, Acosta Espinoza JL, Checa Cabrera MA (2022). Power bi como herramienta de apoyo a la toma de decisiones. Universidad y Sociedad, 14(S3):195-207. https://rus.ucf.edu.cu/index.php/rus/article/view/2949

[21] Dowd C (2022). twosamples: Fast Permutation Based Two Sample Tests. R package version 2.0.0. https://CRAN.R-project.org/package=twosamples

[22] Fernández Jauregui A . Cómo poder en producción un modelo de machine learning de R. https://anderfernandez.com/blog/como-poner-en-produccion-un-modelo-de-machine-learning-de-r/

[23] Hardle W, Simar L (2003). Applied Multivariate Statistical Analysis. Method & Data Technologies.

[24] Hennig C (2023). fpc: Flexible Procedures for Clustering. R package version 2.2-10. https://CRAN.R-project.org/package=fpc

[25] Iniesta Ayuso E (2022). Power bi y su utilidad en la gestión empresarial. Trabajo Fin de Grado, Universidad de Valladolid. https://uvadoc.uva.es/handle/10324/56558

[26] Kassambara A, Mundt F (2020). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.7. https://CRAN.R-project.org/package=factoextra

[27] Kaufman L, Rousseeuw P (1990). Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons.

[28] Knowles J, Frederick C, Whitworth A (2023). merTools: Tools for Analyzing Mixed Effect Regression Models. R package version 0.6.1. https://CRAN.R-project.org/package=merTools

[29] Knowles J, Frederick C (2020). Prediction Intervals from merMod Objects. https://cran.r-project.org/web/packages/merTools/vignettes/Using_predictInterval.html

[30] Kuznetsova A, Brockhoff PB, Bojesen Christensen RH, Jensen SP (2020). lmerTest: Tests in Linear Mixed Effects Models. R package version 3.1-3. https://CRAN.R-project.org/package=lmerTest

[31] Landaburu Jiménez E (2002). Las medidas de divergencia en contrastes de bondad de ajuste con ponderación en las clases. Tesis, Universidad Complutense de Madrid.

[32] Loy A, Lowe J, Moran J (2021). HLMdiag: Diagnostic Tools for Hierarchical (Multilevel) Linear Models. R package version 0.5.0. https://CRAN.R-project.org/package=HLMdiag

[33] Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K,Studer M, Roudier P, Gonzalez J, Kozlowski K, Schubert E, Murphy K (2022). cluster: "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al. R package version 2.1.4. https://CRAN.R-project.org/package=cluster

[34] Microsoft (2023). Descripción de un esquema de estrella e importancia para Power BI. https://learn.microsoft.com/es-es/power-bi/guidance/star-schema

[35] Microsoft (2023). ¿Qué es Azure Data Studio? https://learn.microsoft.com/es-es/sql/azure-data-studio/what-is-azure-data-studio?view=sql-server-ver16

[36] Nakagawa S, Schielzeth H (2013). A general and simple method for obtaining R2 from generalized linear mixed-effects models. Methods in Ecology and Evolution, 4(2):133-142. https://doi.org/10.1111/j.2041-210x.2012.00261.x

[37] Ooms J, Temple Lang D, Hilaiel L (2022). jsonlite: A Simple and Robust JSON Parser and Generator for R. R package version 1.8.4. https://CRAN.R-project.org/package=jsonlite

[38] Pateiro B, Sánchez C (2022). Análisis Multivariante. Máster en Técnicas Estadísticas.

[39] Pinheiro J, Bates D, DebRoy S, Sarkar D, EISPACK authors, Heisterkamp S, Willigen BV, Ranke J, R Core Team (2023). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-162. https://CRAN.R-project.org/package=nlme

[40] Pinheiro JC, Bates DM (2000). Mixed-Effects Models in S ans S-PLUS. Statistics and Computing. Springer.

[41] Ponsico Martín P (2017). Tecnología de Contenedores Docker. Trabajo Fin de Grado, Universitat Politècnica de Catalunya. http://hdl.handle.net/2117/113040

[42] Quiroz J (2003). El modelo relacional de bases de datos. Boletín de política informática, 6:53-61.

[43] R Core Team (2023). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

[44] Romero Roldán L (2022). Clasificación y uso de las librerías R que ofrecen datos demográficos. Trabajo Fin de Grado, Universidad de Sevilla. https://hdl.handle.net/11441/142587

[45] Schloerke B, Allen J, Tremblay B, van Dunné F, Vandewoude S (2022). plumber: An API Generator for R. R package version 1.2.1. https://CRAN.R-project.org/package=plumber

[46] Signorell A (2023). DescTools: Tools for Descriptive Statistics. R package version 0.99.48. https://CRAN.R-project.org/package=DescTools

[47] Tirado Zatarain LM (2017). Implementación de Docker en la gestión del entorno de desarrollo. Trabajo Fin de Grado, Universidad politécnica de Sinaloa.

[48] University of Bristol. Center for Multilevel Modelling. https://www.bristol.ac.uk/cmm/learning/multilevel-models/types.html

[49] Vaquerizo R (2018). Crear una RESTful API con R con plumber. Análisis y Decisión. https://analisisydecision.es/crear-una-restful-api-con-r-plumber/

[50] Vicente Villardón JL (2007). Introducción al análisis de clúster. Departamento de Estadística, Universidad de Salamanca.

[51] Vilar JA (2021). Métodos no paramétricos. Máster en Técnicas Estadísticas.

[52] Wickham H, Chang W, Henry L, Pedersen TL, Takahashi K, Wilke C, Woo K, Yutani H, Dunnington D, RStudio (2023). ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. R package version 3.4.2. https://CRAN.R-project.org/package=ggplot2

[53] Wickham H, François R, Henry L, Müller K, Vaughan D (2023). dplyr: A Grammar of Data Manipulation. R package version 1.1.1. https://CRAN.R-project.org/package=dplyr

[54] Wickham H, Hester J, Chang W, Müller K, Cook D, Edmondson M (2021). memoise: 'Memoisation' of Functions. R package version 2.0.1. https://CRAN.R-project.org/package=memoise

[55] Wickham H (2022). plyr: Tools for Splitting, Applying and Combining Data. R package version 1.8.8. https://CRAN.R-project.org/package=plyr

[56] Xie Y (2023). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.42. https://CRAN.R-project.org/package=knitr

[57] Xu Wang, Yusheng Xu (2019). An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. IOP Conference Series: Materials Science and Engineering, 569. https://dx.doi.org/10.1088/1757-899X/569/5/052024