Grado en Ingeniería Informática

Trabajo de Final de Grado

Análisis del efecto de la COVID-19 sobre la movilidad de visitantes y residentes en destinos turísticos

Autor:
Leonardo David Silva Fialho Soares Monteiro

Supervisor:
Juan Camilo Gómez Esguerra
Tutor académico:
María José Aramburu Cabo

Fecha de lectura: 13 de Julio de 2022
Curso académico 2021/2022

Resumen

Este proyecto se engloba en el área de la ciencia de datos. Su objetivo principal es evaluar el efecto de la COVID-19 en la movilidad con transporte público en entornos turísticos.

Mediante el desarrollo del proyecto se busca proporcionar un flujo de preparación de datos, un modelo adecuado para el análisis de los mismos y una interfaz adecuada para comunicar los resultados.

El proyecto se engloba dentro de otro proyecto mayor, CovMovTur (COVID-19 y movilidades en territorios turísticos: salud física y mental), y busca encontrar herramientas para el análisis de datos de transporte público demostrando por el camino el potencial de la computación en los ámbitos de estudio social.

Palabras clave

Análisis de perfil latente, BigQuery, R, movilidad, Shiny

Keywords

Latente Profile Analysis, BigQuery, R, movility, Shiny

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

1.1.1. Empresa

UBIK Geospatial Solutions existe con el fin de transferir investigación y conocimiento para el desarrollo de soluciones geoespaciales, para ayudar a usuarios y organizaciones a tomar decisiones más inteligentes. Entre los servicios para la integración de información geoespacial que ofrece la empresa, se encuentran los siguientes [1]:

- Desarrollo de aplicaciones móviles basadas en localización (incluyendo GeoGames).
- Desarrollo de aplicaciones Web mapping.
- Infraestructura de Datos Espaciales (IDE).
- Desarrollo de aplicaciones de ciudad inteligente (Smart Campus).
- Integración de datos de medios sociales con otros sistemas de la información,

1.1.2. Proyecto

En este proyecto de final de grado se procederá al análisis de datos de movilidad durante la pandemia. Para ello se seguirá una planificación típica de proyectos de ciencia de datos, la cual se explicará con más detalle en la Sección 2.1 y la Sección 2.2. En esta planificación se abordará desde la preparación de los datos, pasando por la creación del modelo, su interpretación y por último la comunicación de los datos al usuario final.

El proyecto se engloba dentro de otro proyecto mayor, CovMovTur. Este proyecto pretende analizar los efectos de la COVID-19 sobre las movilidades en territorios turísticos, así como el impacto de estos cambios sobre la salud de visitantes y residentes [2]. Para ello el proyecto plantea los siguientes objetivos específicos:

1. Evaluar el efecto dinámico de la COVID-19 sobre la movilidad de visitantes y residentes en destinos turísticos.

2. Identificar los potenciales impactos de la reconfiguración de la movilidad sobre la salud física y mental de visitantes y residentes.

3. Colaborar con las autoridades competentes mediante la provisión de herramientas para una gestión ágil de los impactos de la pandemia sobre los flujos turísticos y de la movilidad cotidiana de los residentes.

1.1.3. Contexto

Este proyecto de final de grado se engloba dentro del primer objetivo. Este pretende evaluar los efectos de la pandemia sobre la movilidad de visitantes y residentes en destinos turísticos.

En la actualidad el proyecto se desarrolla en colaboración con la ATM (Autoritat Territorial de la Mobilitat) del Camp de Tarragona. La ATM proporciona los datos de movilidad del Camp de Tarragona, una zona de interés turístico en las costas mediterráneas. Estos datos se componen de las validaciones (comprobación del pago del servicio de transporte, normalmente por sistemas automatizados, como pueden ser tarjetas inteligentes de transporte) durante los años 2019 y 2020 de viajeros de los servicios de transporte público, tanto por carretera como por ferrocarril, adscritos a la ATM.

Para la ATM, este proyecto supone un salto en la búsqueda de nuevas herramientas que ayuden en la gestión, monitorización y constante mejora del sistema de transporte. Específicamente en este caso, ayudando a comprender las nuevas dinámicas de los viajeros generadas por la COVID-19.

Al finalizar el proyecto, se busca obtener una solución sencilla, accesible, intuitiva y rápida para la visualización de los resultados de analizar e interpretar los datos. Por eso en el proyecto, al final del mismo, se dedica una gran parte del tiempo a la creación de un cuadro de mandos para la visualización y comunicación de los resultados.

1.1.4. Motivación

La motivación principal del proyecto es poder dotar de herramientas especificas para el análisis de datos de transporte público, además de demostrar el potencial que proporciona la computación en este ámbito de estudio social. Esto se ha de tener en cuenta sobre todo dentro del contexto de la crisis provocada por el COVID-19, en el que se engloba el proyecto CovMovTur.

1.2. Alcance del proyecto

1.2.1. Alcance funcional

Desde el punto de vista funcional, el proyecto debe proporcionar información relevante mediante un modelo adecuado y a través de una interfaz (cuadro de mandos) accesible, intuitiva y rápida, centrándose principalmente en aquellos datos con mayor relevancia para el estudio turístico.

1.2.2. Alcance organizativo

Desde el punto de vista organizativo el proyecto deberá proporcionar acceso al usuario (la ATM) a datos de relevancia mediante un entorno web.

1.2.3. Alcance informático

Desde el punto de vista informático, el proyecto deberá proporcionar un flujo de preparación de datos, un modelo y una interfaz que sean capaces de adaptarse a la introducción, comparación y análisis de nuevos datos relevantes, como pueden ser expandir las fechas o territorios de estudio. Para garantizar esta mayor adaptación y flexibilidad, se usará como medio de almacenamiento Google BigQuery, dotando de rapidez y capacidad de cálculo al sistema. En lo relacionado al análisis y la interfaz, se usará el lenguaje de programación R conjuntamente con diversas librerías como pueden ser `mclust`, `ggplot2`, `shiny` y `flexdashboard`.

1.3. Objetivos

El objetivo principal de este proyecto es evaluar el efecto dinámico de la COVID-19 sobre la movilidad de visitantes y residentes en destinos turísticos. Éste se puede desglosar en los siguientes subobjetivos:

- Gestionar el almacenamiento y acceso eficientes a los datos de movilidad.
- Obtener las principales variables que permiten describir los datos de movilidad.
- Buscar y adoptar un método para relacionar y describir las variables de las que dependen los datos.
- Desarrollar una interfaz accesible e intuitiva para la visualización de los resultados del análisis.
- Documentar todo el proceso.

1.4. Descripción del proyecto

Al principio del proyecto solo se dispone de conjuntos de datos en formato CSV sin tratar, que se corresponden con los datos de validación de viajeros de transporte público en el Camp de Tarragona en los años 2019 y 2020.

Se proponen tecnologías de almacenamiento como pueden ser Google BigQuery y tecnologías para análisis de datos como R o Python. Se dispone de ejemplos de sentencias SQL y código en R con relación al tema del análisis. Durante el desarrollo del proyecto parte del código generado se basará estos ejemplos. Todos estos códigos han sido adaptados y han servido como ejemplo o a veces reescritos, para adaptarse al contexto del proyecto.

El proyecto consistirá en desarrollar paso a paso el análisis de los datos, documentando cada paso, e incluirá el desarrollo de un cuadro de mandos para la comunicación de los resultados. Se pretende conseguir una interpretación de los datos del año 2019 para una posterior comparación con los del año 2020.

En la Figura 1.1 podemos observar la arquitectura del sistema a implementar.

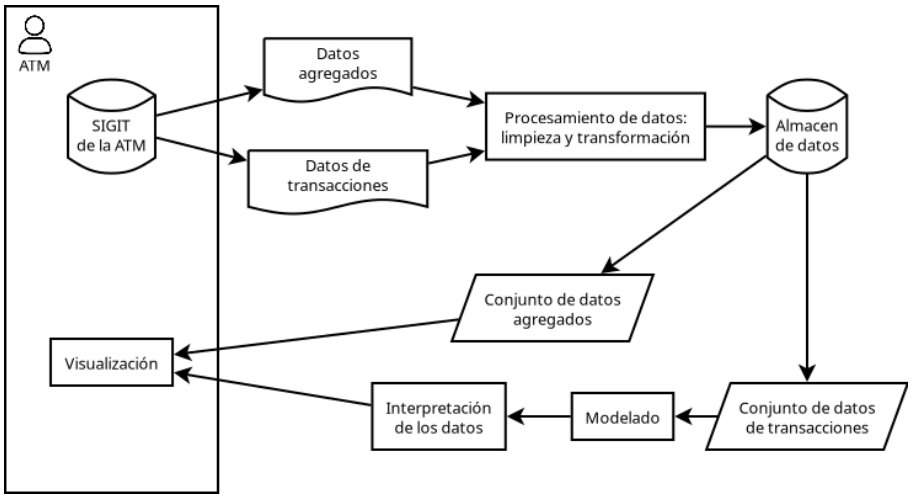


Figura 1.1: Arquitectura general del sistema a implementar.

1.4.1. Tecnologías

En esta sección se describirán las principales tecnologías y las bibliotecas más destacadas que se usarán durante el proyecto.

Google BigQuery

Almacén de datos en la nube de alta escalabilidad. Ejecuta consultas SQL, ademas de contar con un sistema de información geográfica. Será utilizado durante el proyecto para el almacenamiento, limpieza, transformación y agregación de los datos [3].

R

Lenguaje de programación para el análisis estadístico. Es uno de los principales lenguajes utilizado en ciencia de datos. Se usarán principalmente las siguientes bibliotecas:

- **bigrquery**: la API conector de R con Google BigQuery.
- **ggplot2 y bibliotecas afines**: biblioteca muy reconocida para la visualización de gráficos.
- **table1**: se usará para la creación de tablas de estadísticos descriptivos.
- **dplyr**: se usará para la manipulación de datos en R.
- **tidyLPA**: *wrapper* de la biblioteca `mclust` para análisis de perfil latente (LPA, por sus siglas en ingles). La biblioteca `mclust` permite aplicar diferentes modelos de clasificación basados en algoritmos de modelos mezcla.
- **Shiny**: una biblioteca para desarrollo de *web apps* interactivas con R. Se usará en el desarrollo del cuadro de mandos.
- **flexdashboard**: biblioteca para el desarrollo de cuadros de mandos usando R markDown. También se usará en el desarrollo del cuadro de mandos.
- **leaflet**: Leaflet una de las bibliotecas de código de libre más populares de JavaScript. Está biblioteca de R proporciona una manera sencilla de integrar Leaflet dento de R, con características como la integración de eventos de ratón o limites de mapa en aplicaciones web de `shiny`. [4]

RStudio

Es un IDE para el lenguaje de programación R. Se usará a lo largo de todo el proyecto, una vez se hayan subido, transformado, limpiado y combinado los datos en el almacén (Google BigQuery).

Sphinx

Es un generador de documentación escrito en Python. Sphinx convierte archivos reStructuredText principalmente en archivos HTML, aunque también soporta convertir a otros formatos como PDF, EPub, Texinfo y man. Se usará para la creación de la documentación usando el lenguaje markDown. [5]

Contenedores

Principalmente se usará Docker para desplegar los entornos de desarrollo y ejecución del cuadro de mandos. Docker automatiza el despliegue de aplicaciones dentro de contenedores

de software, proporcionando una capa adicional de abstracción y automatización. Una de sus principales ventajas es que permite evitar la sobrecarga de inicio y mantenimiento de máquinas virtuales [6].

1.5. Estructura de la memoria

La memoria está compuesta de 5 capítulos y 2 apéndices. El Capítulo 1 es una introducción al proyecto. El Capítulo 2 describe como se ha planificado y organizado el proyecto, así como el seguimiento temporal del mismo. El Capítulo 3 se centra en los pasos referentes al análisis de los datos. En el Capítulo 4 se habla del diseño del cuadro de mandos. Por último, en el Capítulo 5 se habla de las conclusiones del proyecto. Además se dispone del Apéndice A y del Apéndice B, en los cuales se describen los pasos seguidos para la documentación del proyecto y se presenta el código desarrollado, respectivamente.

Capítulo 2

Planificación del proyecto

2.1. Metodología

Para la realización de este proyecto se ha decidido usar una metodología predictiva como puede ser el desarrollo en cascada. El proyecto se organiza en fases o etapas que deben dejar paso a la siguiente solo una vez terminadas. Esta metodología ha sido elegida principalmente por ser un proyecto realizado por solo una persona y por adecuarse a la estructura general de un proyecto típico de ciencia de datos, como puede verse en la Figura 2.1.

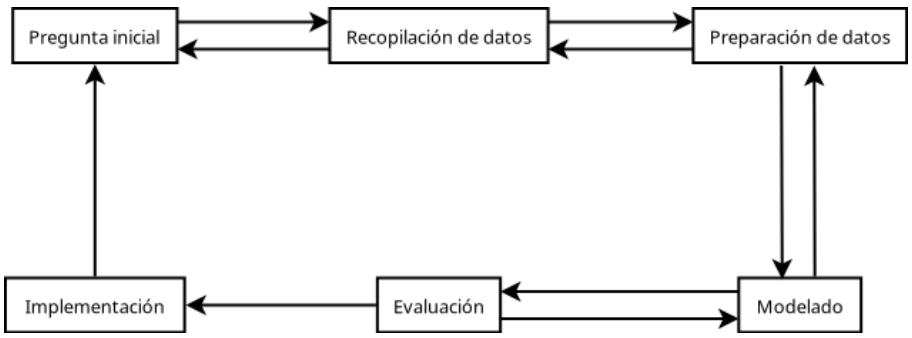


Figura 2.1: Estructura general de un proyecto de ciencia de datos.

Esta estructura se descompone en las siguientes etapas [7]:

1. **Pregunta inicial**: Todo proyecto necesita una comprensión del problema o problemas a solucionar. Los usuarios que necesitan la solución analítica desempeñan el papel más importante en esta etapa, al definir el problema, los objetivos del proyecto y los requisitos de la solución desde una perspectiva más acorde con sus propias necesidades. Cuando el problema se haya establecido claramente, el científico de datos podrá definir el enfoque analítico para resolver el problema. Esta fase implica expresar el problema bajo el contexto de diversas técnicas estadísticas y de aprendizaje automático, para que la organización pueda identificar las más adecuadas para el resultado deseado. El enfoque analítico elegido determina los requisitos de datos. Los métodos analíticos a utilizar requieren de determinados contenidos de datos, formatos y representaciones, orientados por el conocimiento

en el dominio.

2. **Recopilación de datos**: Los científicos de datos identifican y reúnen los recursos de datos disponibles y relevantes para el dominio del problema. Si hay algunas lagunas en la recopilación de datos, es posible que el científico tenga que revisar los requisitos de datos y recopilar más datos o nuevos datos.

3. **Preparación de datos**: Abarca las actividades para construir el conjunto de datos que se utilizará en la siguiente etapa de modelado. Entre las actividades de preparación de datos están la limpieza de datos, combinar datos de múltiples fuentes y transformar los datos en variables más útiles.

4. **Modelado**: En esta etapa se utiliza el conjunto de datos preparado y se pretende desarrollar modelos que resuelvan el problema según el enfoque analítico previamente definido. Este proceso es muy iterativo, lo que provoca cambios en la preparación de datos y en propio modelo. Algunas de las causas de la iteratividad pueden ser que los científicos de datos prueban múltiples algoritmos con sus respectivos parámetros para encontrar el mejor modelo para las variables disponibles o que durante el modelado se extraigan resultados intermedios que informan sobre la adecuación y aplicación del modelo a las variables disponibles.

5. **Evaluación**: El científico de datos evalúa el modelo para comprender su calidad y garantizar que aborda el problema a solucionar de manera adecuada y completa. La evaluación del modelo implica el cálculo de varias medidas de diagnóstico y de otros resultados, como tablas y gráficos, lo que permite al científico de datos interpretar la calidad y la eficacia del modelo en la resolución del problema.

6. **Implementación**: Una vez desarrollado un modelo considerado satisfactorio, se implementa en el entorno de producción o en un entorno de pruebas comparable. Al recopilar los resultados del modelo implementado, los usuarios obtienen información sobre el rendimiento del modelo y su impacto en el entorno en el que se implementó, completando así un círculo de retroalimentación que puede ayudar a mejorar el propio modelo para obtener mejores resultados.

2.2. Planificación

El proyecto se divide en fases siguiendo la estructura general de un proyecto de ciencia de datos, como se describe en la Figura 2.1. A continuación se hace una enumeración de las tareas con la respectiva estimación de su duración:

1. Inicio
    - Pregunta inicial (0,5 días)
    - Aprendizaje
        - BigQuery
            - Leer documentación y realizar ejemplos (1 día)
            - Documentación (2 días)

    - Aprender `bigrquery` (API para R) (1 día)
  - Shiny
    - Leer documentación y realizar ejemplos (1 día)
  - ggplot
    - Leer documentación y realizar ejemplos (1 día)

2. Preparación de datos
   - Carga de datos (1 día)
   - Limpieza, transformación y agregación de datos (12 días)
   - Documentación (1 día)
3. Construcción del modelo
   - Extraer variables características (2 días)
   - Seleccionar variables características (1 día)
   - Documentación (1 día)
4. Ejecutar modelo (1 día)
5. Interpretación
   - Adecuación del modelo (1 día)
   - Significado de los resultados (1 día)
6. Comunicación de los resultados
   - Diseño e implementación de un cuadro de mandos (10 días)

El total de días es de 37,5, teniendo en cuenta una jornada de 8 horas, suman 300 horas en total. El diagrama de Gantt de la Figura 2.2 muestra la planificación temporal de las tareas del proyecto.

2.3. Estimación de recursos y costes del proyecto

2.3.1. Estimación de recursos

Con respecto a la estimación de costes de recursos humanos, tenemos que considerar en primer lugar el rol de científico de datos que asume el trabajador como se muestra en la Tabla 2.1.

Recurso	**Horas**	**Coste**	**Total**
Científico de datos	300	14,50 €/h	4327 €

Tabla 2.1: Costes de recursos humanos.

Los costes de hardware se ven beneficiados por el uso de las herramientas en la nube de Google Cloud, las cuales disponen de un margen de uso gratuito que son suficientes para suplir las

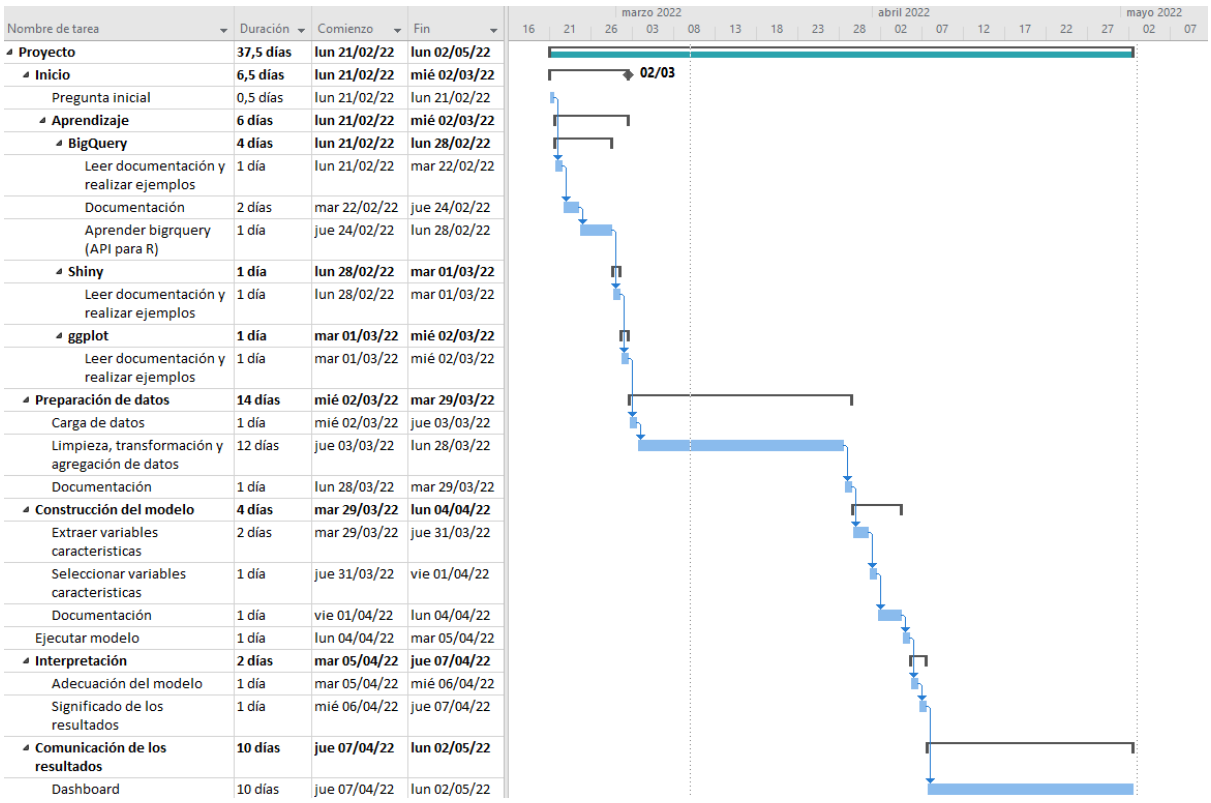


Figura 2.2: Diagrama de Gantt

Recurso	Cantidad	Coste	Total
Portátil HP Zbook G2	2 meses	1200 €	28,57 €
Servidor i7 16GB RAM	2 meses	850 €	20,24 €
		Total	48,81 €

Tabla 2.2: Costes de hardware.

necesidades del proyecto, luego teniendo en cuenta que el ciclo de vida teórico de un dispositivo es de 7 años, el coste total se resume en la Tabla 2.2.

Debido a que durante todo el proyecto solo se utiliza software libre o Google Cloud dentro los margenes de uso gratuitos, se pude considerar que no existe coste alguno.

El resultado final de la estimación de costes se detalla en la Tabla 2.3.

Tipo	Coste
Humanos	4327 €
Hardware	48,81 €
Software	0 €
Total	4375,81 €

Tabla 2.3: Estimación de costes.

ID	Riesgo	Tipo
R01	Falta de experiencia en las tecnologías a usar.	Especifico
R02	Falta de conocimientos teóricos sobre los modelos matemáticos a usar.	Especifico
R03	Mala gestión del tiempo o ausencia.	General
R04	Mala calidad de los datos.	Especifico

Tabla 2.4: Riesgos.

2.4. Análisis de riesgos

En la Tabla 2.4 se listan los principales riesgos identificados. Además, estos pueden clasificarse según si son riesgos que pueden darse en cualquier proyecto o si son específicos de este proyecto. La Tabla 2.5 y la Tabla 2.6 tratan respectivamente el análisis y los planes de prevención o contingencia de riesgos.

2.5. Seguimiento del proyecto

El software de Microsoft Project se ha utilizado en todo momento durante este proyecto para la planificación y el seguimiento. Se ha usado el diagrama de Gantt de la Figura 2.2 como guía temporal y de planificación. Cada quincena este se ha modificado según las necesidades y estado de ejecución del proyecto.

No se ha dado ninguna incidencia que haya condicionado la duración del proyecto. Pero sí que se han dado momentos críticos en los cuales se ha tenido que recurrir a los planes de prevención o contingencia de riesgos. Estos momentos críticos se explican a continuación:

- En la etapa de inicio durante el aprendizaje, se ha necesitado solicitar ayuda a los miembros de la organización para obtener fuentes desde donde partir debido a que la falta de experiencia en algunas herramientas o bibliotecas ha supuesto una falta de conocimiento sobre dónde conseguir fuentes o documentación adecuada para el proyecto.

- Durante la construcción del modelo se ha necesitado comprender conceptos teóricos sobre los criterios para la selección del número de clústeres. Estos son conceptos teóricos y no entran dentro del alcance del proyecto. Pero la formación matemática previa y la realización al mismo tiempo de un trabajo paralelo de temática más teórica sobre el proyecto (fuera del tiempo planificado) han ayudado a la superación de éste y otros temas teóricos dentro del proyecto.

- En la etapa de construcción del cuadro de mandos, algunos errores de integración entre las bibliotecas de `shiny` y `flexdashboard` han supuesto varias horas de pruebas y error para el encaje final del contenido del cuadro de mandos. Debido al largo tiempo planificado para el desarrollo del cuadro de mandos, ésto no ha supuesto un retraso. Después de varias pruebas de diferente contenido o diseño, se ha redirigido parte del cuadro de mandos

ID	Análisis
R01	▪ Descripción: Poca experiencia en el tipo de herramientas, lenguajes y bibliotecas a usar. ▪ Magnitud: Media ▪ Impacto: La curva inicial de aprendizaje puede conllevar retrasos si es más pronunciada de lo previamente planificado.
R02	▪ Descripción: Desconocimiento de los modelos y algoritmos matemáticos que están presentes en el análisis que se realizará. ▪ Magnitud: Media ▪ Impacto: El desconocimiento de los fundamentos teóricos puede llevar a malinterpretar algunos resultados o el cómo puede afectar al análisis tomar ciertas decisiones. La fundamentación teórica no entra dentro del alcance de este proyecto. Una buena comprensión de cada uno de los pasos y los objetivos es fundamental para poder interpretar los resultados y datos intermedios en caso de no llegar a profundizar en este tema.
R03	▪ Descripción: Una mala gestión del tiempo debido a la falta de experiencia o algún retraso causado por una ausencia o baja por parte del único trabajador del proyecto. ▪ Magnitud: Baja ▪ Impacto: El proyecto puede no finalizar en el tiempo previsto, aunque el impacto puede verse reducido por un buen seguimiento de la planificación.
R04	▪ Descripción: Los datos con los que se comienza el análisis presentan una calidad pobre o deficiente de tal manera que no se puede extraer una muestra para el análisis. ▪ Magnitud: Alta ▪ Impacto: El proyecto puede no comenzar debido a que los datos no son suficientes o adecuados para comenzar el análisis.

Tabla 2.5: Análisis de riesgos.

ID	Plan de prevención	Plan de contingencia
R01	Tener en cuenta durante la planificación la curva de aprendizaje de cada herramienta o lenguaje dependiendo de la experiencia previa del trabajador.	Solicitar ayuda a los miembros de la organización en los casos más complicados.
R02	Durante la planificación tener en cuenta las capacidades y experiencia previa del trabajador.	Solicitar ayuda a los miembros de la organización en los casos que generen más dudas o dedicar tiempo extra a comprender un concepto específico.
R03	Monitorizar, seguir y adaptar continuamente la planificación del proyecto según el desarrollo de este.	Si se llega a producir un gran retraso con respecto a la planificación, se deberían replantear aquellas partes del proyecto con menor importancia.
R04	Comunicación entre la ATM y la organización para tener claro cuáles son los objetivos y las necesidades especificas que deben presentar los datos para realizar el análisis. Es decir, plantear de manera correcta cuáles serán los requisitos de datos durante la fase de la pregunta inicial	Replantear los objetivos y el alcance del proyecto para poder realizar un análisis con los datos a nuestro alcance.

Tabla 2.6: Planes de prevención y contingencia de riesgos.

hacia un contenido y diseño mejor integrados, aunque en algunos casos, más estático y previamente construido.

El seguimiento del proyecto durante cada quincena se muestra en Figura 2.3, Figura 2.4, Figura 2.5 y Figura 2.6.

2.6. Conclusiones de la planificación

El seguimiento de este proyecto ha permitido sobreponerse a algunas posibles adversidades como se ha visto en la Sección 2.5. Aun así en ciertos momentos del proyecto se ha notado que una planificación más desarrollada hubiese ayudado durante el mismo. Sobre todo en la etapa de desarrollo del cuadro de mandos, como se da a entender en la Sección 2.5 y la Subsección 4.3.2, una especificación poco concisa del diseño junto con problemas de integración de bibliotecas conllevó una mayor dedicación a resolver problemas durante la implementación del diseño del cuadro de mandos.

En conclusión, aún teniendo en cuenta los problemas surgidos durante la fase de implementación del cuadro de mandos, las desviaciones con respecto a la planificación han sido mínimas y no se ha necesitado de ejecutar acciones correctivas.

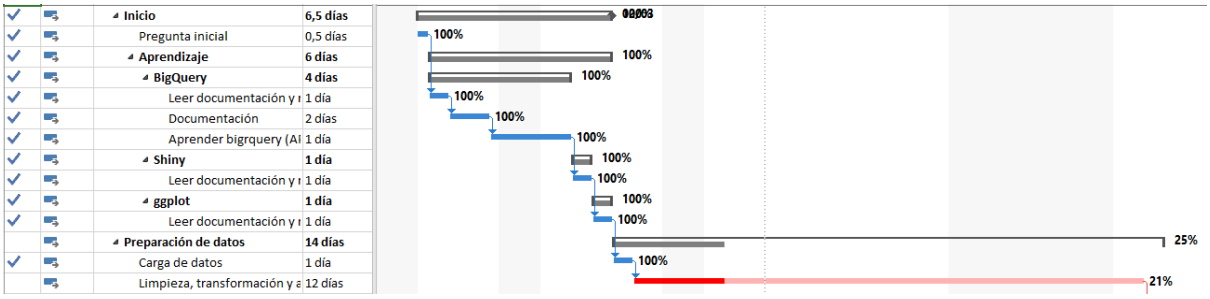


Figura 2.3: Seguimiento de la primera quincena.

Figura 2.4: Seguimiento de la segunda quincena.

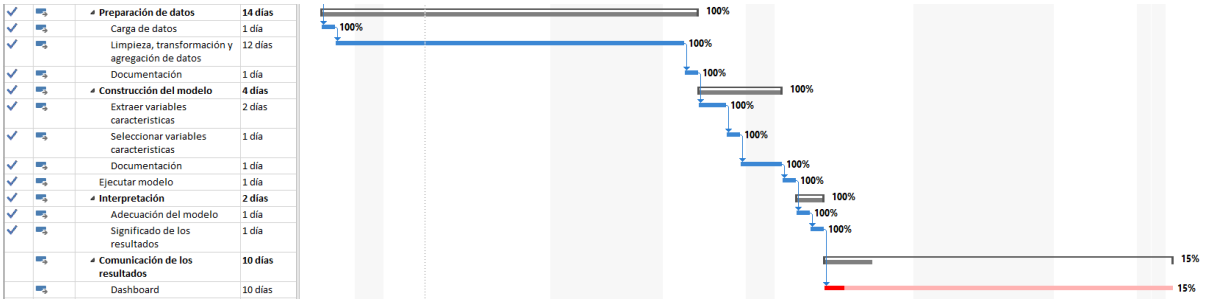


Figura 2.5: Seguimiento de la tercera quincena.

Figura 2.6: Seguimiento de la cuarta quincena.

Capítulo 3

Análisis de los datos

3.1. Introducción

3.1.1. Introducción al análisis

El análisis transcurrirá en diferentes etapas, expuestas anteriormente en la Sección 2.2 sobre la planificación del proyecto. Antes de comenzar a explicar cada una de las etapas vamos a describirlas brevemente:

1. **Pregunta inicial**: se exponen las preguntas iniciales que queremos responder y que se alinean con el objetivo principal del análisis.
2. **Preparación de datos**: se pretende abordar el subobjetivo del proyecto de gestionar el almacenamiento y acceso eficientes a los datos de movilidad, mediante el uso de Google BigQuery.
    - **Análisis de requisitos**: se abordan los requisitos específicos del sistema de almacenamientos de datos esenciales para el desarrollo el proyecto.
    - **Carga de datos**: en este apartado se da una breve explicación de cómo se cargan los datos a Google BigQuery.
    - **Limpieza, transformación y agregación de datos**: en este apartado se describen las principales tareas de transformación y extracción de los datos y las muestras que se usaran en el análisis. Además se describen las consultas SQL utilizadas.
3. **Construcción del modelo**: se pretende abordar el subobjetivo del proyecto de obtener las principales variables que permiten describir los datos de movilidad. Para llevarlo a cabo, se definieron una serie de variables descriptivas, que luego fueron extraídas y seleccionadas por su relevancia.
    - **Extraer variables características**: el propósito de este apartado es extraer un conjunto de datos descriptivos sobre la muestra a analizar. Estos datos descriptivos son las llamadas variables descriptivas, un conjunto de variables estadísticas (como pueden ser medias, frecuencias, desviaciones, etc.) derivadas de los datos originales.

- **Seleccionar variables características**: de todas las variables características seleccionadas en el apartado anterior se pretende verificar si existen redundancias entre ellas, es decir, si se dan grupos de dos o más variables con un significado análogo o estrechamente relacionado. Por ejemplo, dos variables podrían ser la cantidad de días que se usa un recurso y la cantidad de meses que se usa el recurso. Ambas variables están estrechamente relacionadas y pueden ser consideradas redundantes, con saber el número de días que se usa un recurso es fácilmente deducible el número de meses que se usa el mismo recurso con tan solo dividir entre 30.

4. **Ejecutar modelo**: se explica el algoritmo matemático utilizado para analizar la información, además se dan un conjunto de bibliotecas de R para aplicar dicho algoritmo. Con esta sección se aborda otro subobjetivo del proyecto, el buscar y adoptar un método para relacionar y describir las variables de las que dependen los datos.

5. **Interpretación**: una vez ejecutado el modelo, se pretende evaluar e interpretar el significado de los resultados del mismo.

    - **Adecuación del modelo**: en este apartado se pretende comprobar que los resultados obtenidos en el apartado anterior son coherentes. El algoritmo matemático nos devuelve diferentes resultados para los cuales tenemos que utilizar una serie de criterios para calificarlos y conseguir de esta manera un modelo que se adapte de la mejor manera posible a la realidad que estamos describiendo.
    - **Significado de los resultados**: se pretende dar un contexto y sacar información útil a partir de unos resultados que no son más que un conjunto de números y variables.

3.1.2. Área de estudio

El área de estudio es Camp de Tarragona, una región de Cataluña, España, con una población total de 617.504 habitantes en 2018 según el INE (Instituto Nacional de Estadística). La ATM (Autoridad de Movilidad Territorial) del Camp de Tarragona es el consorcio responsable de la gestión de las concesiones de transporte público en esta área y del sistema de información que provee los datos para este proyecto.

La distribución espacial de las actividades económicas y la población en el Camp de Tarragona es desigual, con la mayor parte distribuida a lo largo de la costa. Tarragona y Reus (con poblaciones por encima de los 100.000 habitantes) y los tres municipios costeros de Cambrils, Salou y Vila-seca (con poblaciones por encima de los 20.000 habitantes) constituyen las principales zonas de actividad económica y población. Estos tres municipios costeros constituyen una de las zonas turísticas de más importancia en España, la Costa Daurada, y contienen un 77 % de la capacidad hostelera del Camp de Tarragona. Según el Observatorio de Turismo de Cataluña, estos tres municipios acogieron más de 20,2 millones de pernoctaciones en alojamientos regulados en 2018. El turismo es notablemente estacional. Dos terceras partes de las pernoctaciones tuvieron lugar durante el verano. Además hay que tener en cuenta la presencia de PortAventura (uno de los parques temáticos más grandes de Europa) en los municipios de Vila-seca y Salou, que atraen a más de 3,5 millones de visitantes anualmente según datos del 2015 [8].

3.2. Pregunta inicial

La pregunta o preguntas iniciales de un proyecto de ciencia de datos es la base y punto de partida del proyecto, que se alinea con los objetivos a conseguir y el potencial valor del proyecto. En este proyecto, el principal objetivo es evaluar el efecto dinámico de la COVID-19 sobre la movilidad de visitantes en destinos turísticos. Nuestra pregunta inicial podría ser: *¿Cuál es el impacto de la COVID-19 en el uso del transporte público por parte de visitantes?*

La pregunta inicial es muy amplia y general, es necesario poder comprender las necesidades especificas subyacentes a esta pregunta para poder responder a cada una de ellas.

- Existen visitantes, sobre los cuales se puede obtener información, pero para poder responder a la pregunta inicial es necesario describirlos primero. Para conocer el impacto que ha tenido la pandemia sobre los visitantes, es necesario conocer más sobre estos visitantes. Surgen diferentes preguntas: *¿Cómo se describe a los visitantes? ¿Son todos los visitantes iguales? ¿O se pueden clasificar los visitantes en diferentes perfiles? ¿Cómo sería esta clasificación? ¿Cómo serían los distintos perfiles de los visitantes?*
- Es necesario tener un conocimiento amplio del comportamiento de los visitantes tanto antes como durante y después de la pandemia. También surgen varias preguntas: *¿Existen diferencias de comportamiento entre antes y durante la COVID-19? ¿Como se comparan estas diferencias? ¿Los comportamientos son independientes o existen pautas a través de las cuales se pueden comparar?*

3.2.1. Enfoque analítico

Estas necesidades iniciales llevan a pensar en determinadas soluciones para darles respuesta:

- Los visitantes serán clasificados, por lo cual se necesita encontrar un modelo estadístico que permita clasificar a estos visitantes, aportando una descripción de su comportamiento.
- Será necesario conocer el comportamiento de los visitantes tanto antes como durante la pandemia. Este comportamiento deberá poder ser comparado, por lo cual es necesario poder expresar el comportamiento antes y durante la pandemia bajo los mismos términos. Es necesario encontrar un método que nos permita realizar esta comparación. Por ejemplo, se halla un modelo para clasificar el comportamiento de los visitantes antes de la pandemia y se usa el mismo modelo para clasificarlos durante la pandemia.

3.2.2. Requisitos de datos

De este análisis inicial se deducen un conjunto de requisitos de datos para poder llevar a cabo el proyecto. Al usar datos relativos a transacciones en transporte público, es lógico pensar que se proporcionaran datos relativos a identificadores de viajeros, datos temporales de validación, datos espaciales de inicio o fin del viaje y datos relativos al trayecto o linea de transporte. En especifico:

- **Datos de actividad del viajero**: datos relativos al uso que se realiza del transporte publico. Estos datos en especifico son: identificador de viajero, identificación de transbordos y número de personas con las que viaja.
- **Datos espaciales del viaje**: datos relativos al lugar o lugares geográficos en los cuáles se desarrolla un viaje. Estos datos en especifico son: lugar de inicio o fin de viaje (conjunto de datos geográficos que puede presentar distinto nivel de granulidad), rutas usadas y zonas o regiones de transporte.
- **Datos temporales del viaje**: datos relativos a la fecha y hora en las cuáles se desarrolla un viaje. Estos datos en especifico son: hora del inicio o fin del viaje y fecha de inicio o fin del viaje.

3.3. Recolección de datos

La ATM de Camp de Tarragona dispone de un sistema de gestión de integración tarifaria (SIGIT). Está constituido por un sistema automático de recolección de datos a través de tarjetas electrónicas que se validan al inicio de un trayecto en transporte público y permiten almacenar información relativa a la hora de inicio del trayecto, la localización de la parada donde se inició el trayecto, la identificación de la línea y el tipo de tarifa usada [8].

El conjunto de datos analizados se compone de todas las transacciones realizadas durante el año 2019 y el año 2020. Con un total de 7.688.864 transacciones y 147.528 tarjetas electrónicas diferentes en 2019, y 3.025.481 transacciones y 69.764 tarjetas electrónicas diferentes en 2020. Las transacciones están ordenadas por fecha y hora así como por el código de la tarjeta electrónica y otros detalles sobre la transacción (identificador único de transacción, tipo de transacción, línea, etc.). Estos datos generados por el SIGIT fueron entregados en formato CSV.

3.3.1. Muestreo de datos

La ATM de Tarragona dispone de varias tarifas:

- **T-10**: tarifa multipersonal con 10 transacciones que puede ser recargada.
- **T-10/30**: tarifa unipersonal con 10 transacciones durante 30 días consecutivos.
- **T-50/30**: tarifa unipersonal con 50 transacciones durante 30 días consecutivos.
- **T-70/90**: tarifa multipersonal con 70 transacciones durante 90 días consecutivos. Solo para familias numerosas.
- **T-MES**: tarifa unipersonal ilimitada durante 30 días consecutivos.
- **T-12**: tarifa unipersonal ilimitada para menores de 14 años.
- **Tickets propios**: grupo de tarifas ofrecidas por el operador de una linea de transportes. Estas tarifas no son gestionadas por la ATM.

Si se observan las tarifas y la Figura 3.1, se puede ver que la tarifa T-10 es una tarifa atractiva para los usuarios turísticos. No solo dispone de unas condiciones favorables para su uso turístico, sino que además se puede observar que el incremento de transacciones durante los meses de verano se concentra principalmente en esta tarifa.

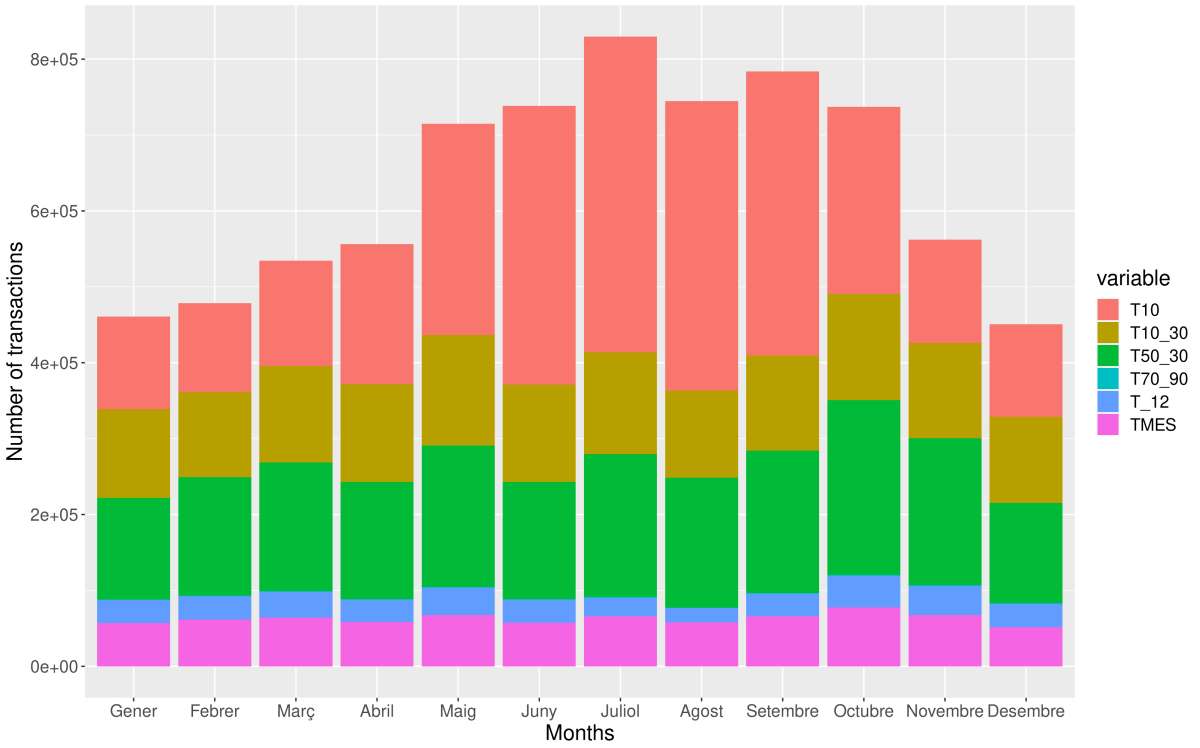


Figura 3.1: Transacciones del año 2019 por mes y tarifa.

Debido a estas razones, para seleccionar una muestra que represente el conjunto de los turistas, se decidió proceder seleccionando:

1. Las transacciones pertenecientes solo a líneas interurbanas.
2. Las transacciones de aquellas tarjetas electrónicas usadas solo durante el verano.
3. Las transacciones de las tarjetas electrónicas con tarifas T-10.

Al realizar esta selección se vislumbra un panorama realmente interesante. Según se puede contemplar en la Tabla 3.1, las tarifas T-10 representan el 93,8 % de las tarjetas y el 89,6 % de las transacciones durante el verano del año 2019.

Tarjetas usadas solo durante el verano	**Tarjetas**	**Transacciones**
Todas las tarifas	36.478	656.577
T-10	34.214	588.709

Tabla 3.1: Tarjetas usadas solo durante el verano del año 2019 y sus transacciones.

Se puede concluir que la muestra a analizar serán las transacciones interurbanas realizadas por tarjetas usadas solo durante el verano y de la tarifa T-10.

3.4. Preparación de los datos

3.4.1. Análisis de requisitos

Requisitos del sistema de almacenamiento de datos

En la Sección 1.3 se presenta uno de los subobjetivos del proyecto relativos al sistema de almacenamiento de datos. Este subobjetivo es gestionar el almacenamiento y acceso eficientes a los datos de movilidad. Para ello se identifican un conjunto de requisitos que debe cumplir:

- Ser capaz de trabajar con grandes cantidades de datos, con énfasis en el rendimiento.
- Ser capaz de trabajar con datos geoespaciales.
- Proporcionar integraciones con lenguajes de importancia a nivel de análisis estadístico, principalmente R o Python.
- Proporcionar herramientas para la limpieza, combinación y transformación de los datos, preferiblemente que sea compatible con SQL.

Como se describió en la Subsección 1.4.1, se ha decidido usar Google BigQuery como sistema de almacenamiento, gestión y procesado de los datos. Esta herramienta cumple perfectamente con los requisitos antes mencionados, además de ser una apuesta de cara al futuro por sus grandes capacidades de escalabilidad e integración con diferentes herramientas o lenguajes.

Requisitos funcionales de la base de datos

Para la consecución del proyecto, la base de datos debe de cumplir un conjunto de requisitos funcionales:

- Las operaciones de limpieza, combinación y transformación de datos deben ser gestionadas mediante consultas SQL para su reutilización.
- Los procesos de limpieza, combinación y transformación deben estar documentados.
- La definición del esquema de las tablas debe seguir un modelo genérico.

3.4.2. Carga de datos

Los datos son proporcionados por la ATM en archivos CSV con un total de 2GB de datos. BigQuery proporciona muchas maneras diferentes de importar datos, pero para este caso en especifico se usará la opción de crear tablas a partir de datos almacenados en Google Drive. Existen varias razones para ello, entre ellas el limite de 200 MB para la carga de archivos CSV locales y los costes adicionales de una instancia de Google Cloud Storage.

3.4.3. Limpieza, combinación y transformación de datos (ETL)

Los datos importados a BigQuery en la sección anterior funcionan como tablas externas. Las tablas externas actúan como tablas estándar de BigQuery. Los metadatos de la tabla, incluido el esquema de la tabla, se almacenan en BigQuery, pero los datos en sí residen en la fuente externa. Este tipo de tablas son usadas principalmente para la realización de procesos ETL durante las operaciones de importación de datos a BigQuery.

Se poseen dos tipos de archivos CSV: archivos con el volcado de todas las transacciones correspondientes a un año y un archivo con datos agregados por parada, mensual y anualmente.

Se generaron consultas SQL para pasar un proceso de limpieza y transformación previo a poder comenzar a trabajar con los datos. A continuación se procederá a explicar estas consultas.

Datos de transacciones anuales

Para cada una de las tablas iniciales (las dos tablas con las transacciones del año 2019 y del año 2020) se han realizado las siguientes transformaciones:

- Usar nombres de columna basados en nombres de campos de GTFS (*General Transit Feed Specification*, una especificación de Google que pretende definir un formato común para horarios de sistemas de transporte publico y datos geográficos asociados) [9].
- Convertir la hora a enteros.
- Convertir la fecha a tipo DATE.
- Crear una columna de tipo DATETIME con la fecha y hora.
- Convertir el id de municipio a tipo INTEGER.
- Crear la columna **urban**: 0 si no es un trayecto urbano y 1 si es un trayecto urbano.
- Arreglar incoherencias en los datos como pueden ser nulos representados por guiones o espacios.

En la Subsección B.2.1 del Anexo B se puede ver el código SQL utilizado para realizar estas transformaciones.

Datos agregados por parada mensual y anualmente

Para la tabla de datos agregados mensual y anualmente por paradas, hemos realizado las siguientes transformaciones:

- Usar nombres de columna basados en nombres de campos de GTFS [9].

- Crear una columna con los meses en tipo INTEGER.
- Borrar filas con datos incorrectos o vacíos.

En la Subsección B.2.2 del Anexo B se puede ver el código SQL utilizado para realizar estas transformaciones.

Extracción de la muestra a analizar

Para la extracción de la muestra (transacciones de tarjetas con tarifa T-10 operativas solo en verano) se siguen un conjunto de pasos descritos a continuación:

1. Se seleccionan las transacciones realizadas con la tarifa T-10.
2. Se seleccionan las transacciones realizadas durante el verano (desde el 21 de Junio hasta el 23 de Septiembre).
3. Se seleccionan las transacciones realizadas durante todo el año exceptuando las fechas de verano (desde el 21 de Junio hasta el 23 de Septiembre).
4. Se realiza la diferencia entre los dos conjuntos anteriores, es decir, transacciones de verano quitando aquellas cuyas tarjetas hayan realizado alguna transacción durante el resto del año. Se usa un LEFT JOIN, seleccionando aquellas uniones que presenten nulos.

En la Subsección B.2.3 del Anexo B se puede ver el código SQL utilizado para realizar esta extracción.

Notas sobre la privacidad

Debido a motivos de privacidad de datos de viajeros, la fuente de datos presenta restricciones de rango en algunas variables. Sobre todo en aquellas temporales y espaciales de los archivos de volcado anuales.

Con respecto a las variables espaciales no se cuentan datos a nivel de parada para cada transacción. Solo se nos proporcionan datos a nivel de municipio para cada transacción, no se cuentan con datos de paradas. Y con respecto a las variables temporales, las horas de las transacciones realizadas entre las 22:00 y 7:00 del día siguiente tienen valores nulos. Solo se nos proporcionan datos a nivel de hora entre las 7:00 y 22:00 de un mismo día y con resolución de 1 hora.

La tarifa T-10 además cuenta con una restricción de uso horario: solo se pueden realizar viajes/transacciones con esta tarifa entre las 6:00 y 23:00. Por lo cual se ha decido implementar una solución para imputar las horas que aparecen como nulas en esta tarifa. Debido principalmente a que no supone ningún esfuerzo extra y que existe una manera sencilla de imputar las horas

de las transacciones restantes para la tarifa T-10, es decir, aquellas transacciones realizadas de 6:00 a 7:00 y de 22:00 a 23:00.

La imputación es posible debido a que se da el siguiente caso: el id de transacción es un entero que asigna valores consecutivos para transacciones consecutivas en el tiempo, si para alguna tarjeta, en un día dado, se produce más de una transacción en distintas franjas horarias, hay que fijarse en si alguna de estas transacciones es nula y así es posible determinar si se produce de 6:00 a 7:00 o de 21:00 a 22:00.

En la Subsección B.2.4 del Anexo B se puede ver el código SQL utilizado para realizar esta imputación.

3.5. Construcción del modelo

3.5.1. Extraer variables características

Usando consultas SQL, como las disponibles en la Subsección B.2.5, se han extraído 26 variables significativas para cada una de las tarjetas, como podemos observar en la Tabla 3.2. Estas han sido calculadas agrupando los datos específicos para cada una de las tarjetas electrónicas durante su periodo en activo en los meses de verano. Cada una de las variables está separada en uno de tres grupos: actividad, tiempo y espacial. Cada tipo depende de si describen patrones de uso, patrones espaciales o patrones temporales de una tarjeta. Esta separación permite realizar tareas de análisis por separado para cada tipo de patrones, obteniendo un mejor comportamiento de las variables y, como consecuencia, unos resultados más útiles y sencillos de interpretar.

Durante este trabajo usaremos solo las variables de actividad para los ejemplos simplificando así la comprensión y evitando repetirnos, exceptuando un solo ejemplo con variables temporales. Todos los pasos seguidos se replicaron de igual manera para cada uno de los tipos de variables durante el proyecto.

3.5.2. Seleccionar variables características

Antes de continuar con el análisis es necesario comprobar la redundancia entre las variables características. Para ello se han calculado las matrices de correlación para seleccionar aquellas variables que se excluirán por no aportar información nueva o relevante. En la Figura 3.2 se puede observar la matriz de correlación de la variables de actividad.

El criterio principal a utilizar para seleccionar aquellas variables a excluir es usar el coeficiente de correlación entre dos variables. Si el coeficiente es muy alto nos quedamos con la variable con un significado más general. Además de este criterio, hay algunas variables que pueden tener poco o ningún sentido a la hora de realizar el análisis (como la variable de `visited_zones`, que presenta en la mayoría de los casos el mismo valor, ya que pocos turistas se dedican a coger líneas de transporte que recorren más de una zona). Algunas de estas últimas son más

Tipo	Nombre	Descripción
	Tarjeta	Variable de agregación (n = 34.214).
Actividad	transactions	Número total de transacciones (n = 588.709).
	avg_transactions	Número medio de transacciones por día.
	transaction_chains	Número de transacciones sin contar transbordos.
	active_period	Número de días entre el primer y último día que la tarjeta fue usada.
	active_days	Número de días que la tarjeta fue usada.
	active_months	Número de meses que la tarjeta fue usada.
	avg_group_size	Número medio de transacciones consecutivas.
	min_group_size	Número mínimo de transacciones consecutivas.
	max_group_size	Número máximo de transacciones consecutivas.
	group_transactions	Número de cadenas de transacciones con más de una transacción.
Time	weekdays_rel	Porcentaje de transacciones entre semana.
	weekends_rel	Porcentaje de transacciones en fin de semana.
	first_half_day_rel	Porcentaje de transacciones en la primera mitad del día (7:00 - 16:00).
	second_half_day_rel	Porcentaje de transacciones en la segunda mitad del día (16:00 - 21:00).
	time_morning_rel	Porcentaje de transacciones durante la mañana (7:00 - 12:00).
	time_midday_rel	Porcentaje de transacciones durante el medio día (12:00 - 16:00).
	time_afternoon_rel	Porcentaje de transacciones durante la tarde (16:00 - 21:00).
	time_night_rel	Porcentaje de transacciones durante la noche (21:00 - 6:00).
Spatial	visited_municipalities	Número de municipios visitados durante todo el periodo activo.
	used_routes	Número de rutas usadas durante todo el periodo activo.
	main_municipality	Porcentaje de transacciones concentrado en el municipio más visitado.
	main_two_municipalities	Porcentaje de transacciones concentrado en los dos municipios más visitados.
	main_three_municiplaities	Porcentaje de transacciones concentrado en los tres municipios más visitado.
	transactions_tarr_reus	Porcentaje de transacciones concentrado en las principales ciudades (Tarragona y Reus) con más de 50.000 habitantes.
	transactions_cgc	Porcentaje de transacciones concentrado en las principales ciudades turísticas (Cambrils, Salou y Vilaseca) entre 20.000 y 50.000 habitantes.
	transactions_urban_municipalities	Porcentaje de transacciones concentrado en las principales ciudades con más de 10.000 habitantes

Tabla 3.2: Variables características y su descripción.

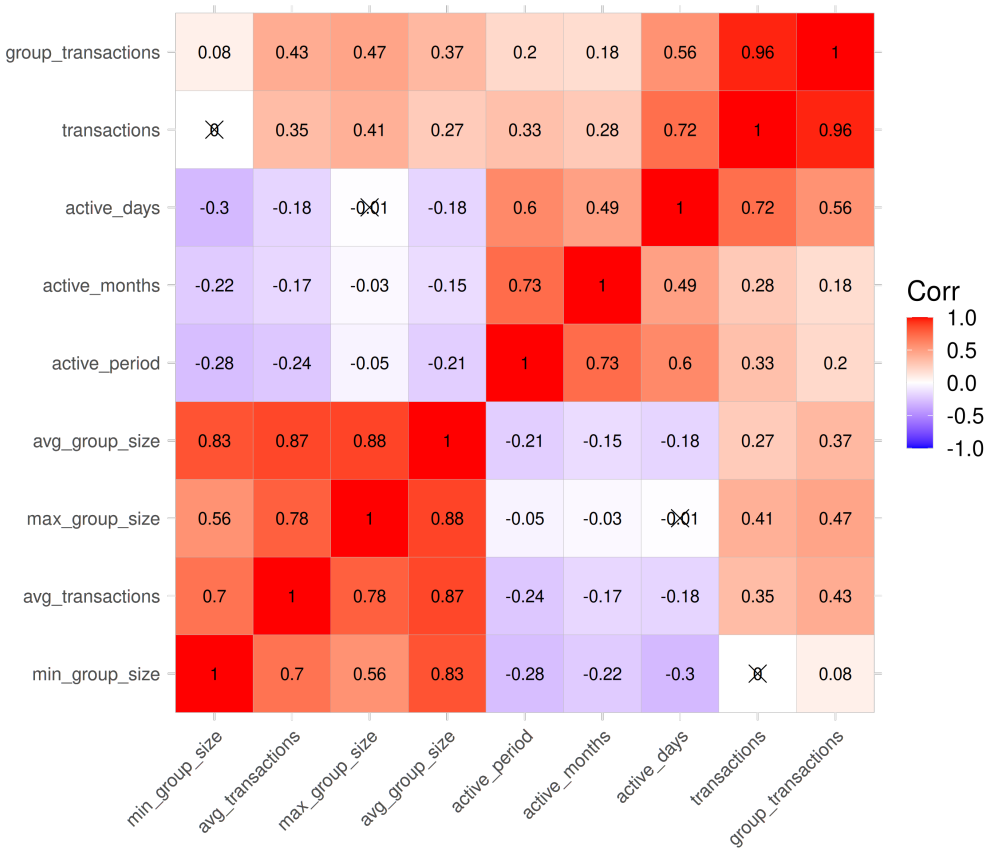


Figura 3.2: Matriz de correlación inicial de actividad.

fácilmente detectables una vez aplicado el modelo, con lo cual esta etapa del análisis puede volver a repetirse iterativamente en conjunto con la siguiente etapa para encontrar resultados más sensatos y fieles a la realidad.

Al final de la selección, con respecto a las variables de actividad, se obtendría algo parecido a la Figura 3.3.

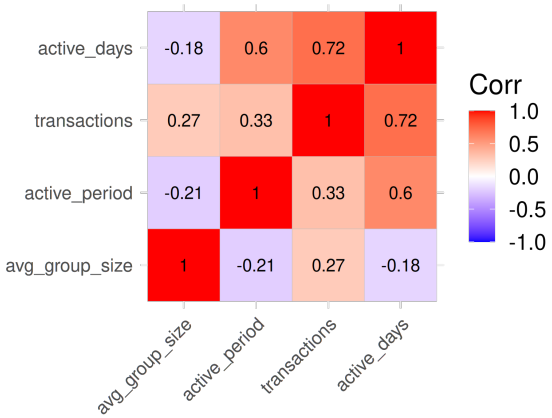


Figura 3.3: Matriz de correlación final de actividad.

Al final de esta etapa las variables seleccionadas en todos los grupos serían:

- **Actividad**: `transactions`,`active_period`, `active_days` y `avg_group_size`.
- **Temporal**: `first_half_day_rel`, `time_morning_rel`, `time_midday_rel` y `time_night_rel`.
- **Espacial**: `transactions_cgc`,`main_two_municipalities`,`main_municipalitiy`,`used_routes`.

3.6. Ejecución del modelo

Se ha utilizado una técnica de clasificación mediante clústeres para identificar los perfiles de los viajeros. Las técnicas de clústeres son algoritmos de agrupamiento de objetos en clases de objetos similares (en este caso las tarjetas electrónicas). La técnica utilizada en este análisis es el análisis de perfil latente (LPA por sus siglas en inglés) un tipo de análisis de variable latente. Este técnica es utilizada sobre todo por la capacidad que ha demostrado en estudios de ciencias sociales y salud [8].

El análisis de variable latente ayuda a trabajar con situaciones en los que algunas variables no son observables. El termino latente significa que esta variable no es medida directamente, en su lugar, se mide de manera indirecta a través de dos o más variables observadas. Es decir, en el caso de este trabajo la clasificación de los perfiles de viajeros es una variable latente que no es observable de manera directa pero sí de manera indirecta a través de las variables características antes descritas, las cuales sí que son observables [10].

El análisis de perfil latente (LPA) es un tipo de análisis de variable latente que se basa en asumir que los datos provienen de una distribución desconocida la cuál proviene de una mezcla de distribuciones más simples. A estas técnicas también se les suele denominar modelos mezcla Gaussianos o normales, pues para determinar la distribución de los datos suelen asumir que estos son una mezcla de una o más clases o clústeres que se pueden describir mediante distribuciones normales [11].

Usando la biblioteca tidyLPA se ha aplicado la técnica de agrupamiento a los datos del año 2019. El resultado son varios modelos correspondientes a los distintos números de posibles clases, con lo cual habrá que elegir el modelo más adecuado para los datos a analizar. Posteriormente, una vez escogidos los modelos que más se adecuan a los datos, se procederá a la clasificación de los datos del año 2020 siguiendo el mismo modelo creado para el año 2019. Esto permitirá comparar la evolución de los datos de un año para otro.

3.7. Interpretación del modelo

3.7.1. Adecuación del modelo

El paso más importante del análisis de perfil latente es determinar el número de clústeres o clases que se adecua mejor a la interpretación de nuestro modelo. La fase de ejecución devuelve

diferentes modelos dependiendo del número de clústeres. Para poder compararlos se utilizan diferente tipos de criterios [11], principalmente se tienen en cuenta los siguientes:

- **BIC o Bayessian Information Criterion**: criterio basado en la verosimilitud del modelo, pero para evitar el sobreajuste del modelo añade un termino de penalización que depende del número de parámetros del modelo. Es el criterio más utilizado.
- **AIC o Akaike Information Criterion**: criterio parecido al BIC pero con un termino de penalización menor que el BIC.
- **Probabilidades posteriores**: media de las probabilidades posteriores de los datos en los diferentes perfiles.

Teniendo en cuenta los criterios anteriores puede darse el caso de un modelo que se adecue perfectamente a los datos, pero a nivel practico no es una obligación seguir dichos criterios. Si para un determinado análisis favorece usar otro modelo que no se adecue tan bien como el señalado por los distintos criterios, se puede usar sin problemas. Estos criterios son principalmente una recomendación o guía a la hora analizar los resultados.

El ejemplo de la Tabla 3.3 muestra la comparación de modelos de agrupamiento según la actividad dependiendo del número de clústeres. Como podemos observar, el modelo con 5 clústeres o perfiles es el más adecuado ya que presenta los valores más bajos tanto de BIC como de AIC. Si se observa el ejemplo Tabla 3.4 que muestra la comparación de modelos de agrupamiento temporal dependiendo del número de clústeres, se puede ver que el modelo con 7 clústeres o perfiles es el más adecuado ya que presenta los valores más bajos tanto de BIC como de AIC. Pero en este caso, se ha seleccionado el modelo con 5 perfiles pues los resultados se adecuan mejor y son más relevantes.

Perfiles	**AIC**	**BIC**	**Entropy**	**Prob. mínima**	**Prob. máxima**
2	805265,1	805374,8	0,9863898	0,9709350	0,9979880
3	789300,6	789452,5	0,9481548	0,8551724	0,9898778
4	780108,2	780302,3	0,9572573	0,8551741	0,9895949
5	762991,7	763228,1	0,9526775	0,8726156	0,9853257

Tabla 3.3: Estadísticos de adecuación del modelo de agrupamiento de actividad del año 2019.

Esta situación aunque parezca extraña tiene su sentido, aunque el AIC y BIC sean criterios que informan de la adecuación del modelo intentando evitar sobreajustes, en algunas situaciones estos se dan pero con un trasfondo más contextual. Por ejemplo, el agrupamiento con 7 perfiles temporales podría haber creado un perfil extra para clasificar a aquellos viajeros que en el agrupamiento con 5 perfiles temporales hubiesen estado agrupados en uno solo. Se puede imaginar que en el agrupamiento con 5 perfiles temporales se posee un perfil que describe a aquellos turistas que viajan durante el periodo del medio día (alrededor de las 12 de la mañana), pero en el de 7 perfiles se crean dos perfiles, uno para los que viajan a principios del mediodía (de 11:00 a 12:00) y otros a final del medio día (de 12:00 a 13:00). Este agrupamiento puede clasificar a los viajeros de una manera bastante más fiel a la realidad, pero contextualmente puede no ser necesario o relevante a la hora de realizar el estudio.

Perfiles	AIC	BIC	Entropy	Prob. mínima	Prob. máxima
2	1194634	1194744	0,9638361	0,9349548	0,9946407
3	1180544	1180696	0,8098629	0,8926959	0,9434097
4	1171230	1171424	0,7628323	0,8074905	0,9549924
5	1165294	1165530	0,8127603	0,8369385	0,9414278
6	1150937	1151216	0,8522300	0,8743252	0,9454467
7	1144529	1144850	0,8540923	0,8472782	0,9748451

Tabla 3.4: Estadísticos de adecuación del modelo de agrupamiento temporal del año 2019.

	A1 Continued (N=494)	A2 Groups (N=407)	A3 Short term (N=27224)	A4 Long term (N=4306)	A5 Sporadic (N=1783)	Total (N=34214)
transactions						
Mean (SD)	65,7 (9,42)	26,9 (9,42)	12,5 (9,42)	40,3 (9,42)	17,7 (9,42)	17,2 (9,42)
Median [Min; Max]	60,5 [19,0; 189]	20,0 [8,00; 119]	10,0 [1,00; 40,0]	39,0 [10,0; 129]	15,0 [2,00; 69,0]	11,0 [1,00; 189]
active_period						
Mean (SD)	47,9 (6,21)	2,64 (6,21)	5,20 (6,21)	13,0 (6,21)	51,4 (6,21)	9,17 (6,21)
Median [Min; Max]	46,0 [11,0; 95,0]	1,00 [1,00; 36,0]	4,00 [1,00; 33,0]	11,0 [3,00; 41,0]	49,0 [28,0; 93,0]	5,00 [1,00; 95,0]
active_days						
Mean (SD)	18,3 (1,82)	1,69 (1,82)	2,71 (1,82)	7,17 (1,82)	6,04 (1,82)	3,66 (1,82)
Median [Min; Max]	16,0 [10,0; 58,0]	1,00 [1,00; 7,00]	3,00 [1,00; 10,0]	7,00 [3,00; 18,0]	5,00 [2,00; 16,0]	3,00 [1,00; 58,0]
avg_group_size						
Mean (SD)	2,13 (1,18)	10,2 (1,18)	2,69 (1,18)	3,10 (1,18)	1,88 (1,18)	2,78 (1,18)
Median [Min; Max]	2,11 [1,00; 5,70]	9,00 [6,67; 30,0]	2,50 [0; 7,00]	2,92 [1,00; 9,15]	1,80 [0; 7,00]	2,50 [0; 30,0]

Tabla 3.5: Resumen de las variables características para cada uno de los perfiles de la clasificación de actividad del año 2019.

3.7.2. Significado de los resultados

Clasificación del año 2019

Los ejemplos de la Figura 3.4 y la Figura 3.5 ilustran el resultado de la clasificación de actividad de los datos del año 2019. Además en la Tabla 3.5 se puede ver un resumen de las variables características para cada uno de los perfiles.

Como se puede observar, se han identificados 5 grupos diferentes:

- **A1 Continuado**: tarjetas que hacen un uso continuado del sistema de transporte público. Se puede observar que está determinado por las medias de las variables `active_period` y `transactions`, las cuales son grandes en comparación con los demás grupos.
- **A2 Grupos**: tarjetas que son usadas por grandes grupos. Se puede observar que está determinado por la media de las variable `avg_group_size`, que es la mayor en comparación.
- **A3 Corto plazo**: tarjetas usadas por turistas de estancias de corto plazo. Se puede observar que está determinado por la media de la variable `active_period`, que es la más pequeña.
- **A4 Largo plazo**: tarjetas usadas por turistas de estancias de largo plazo. Se puede

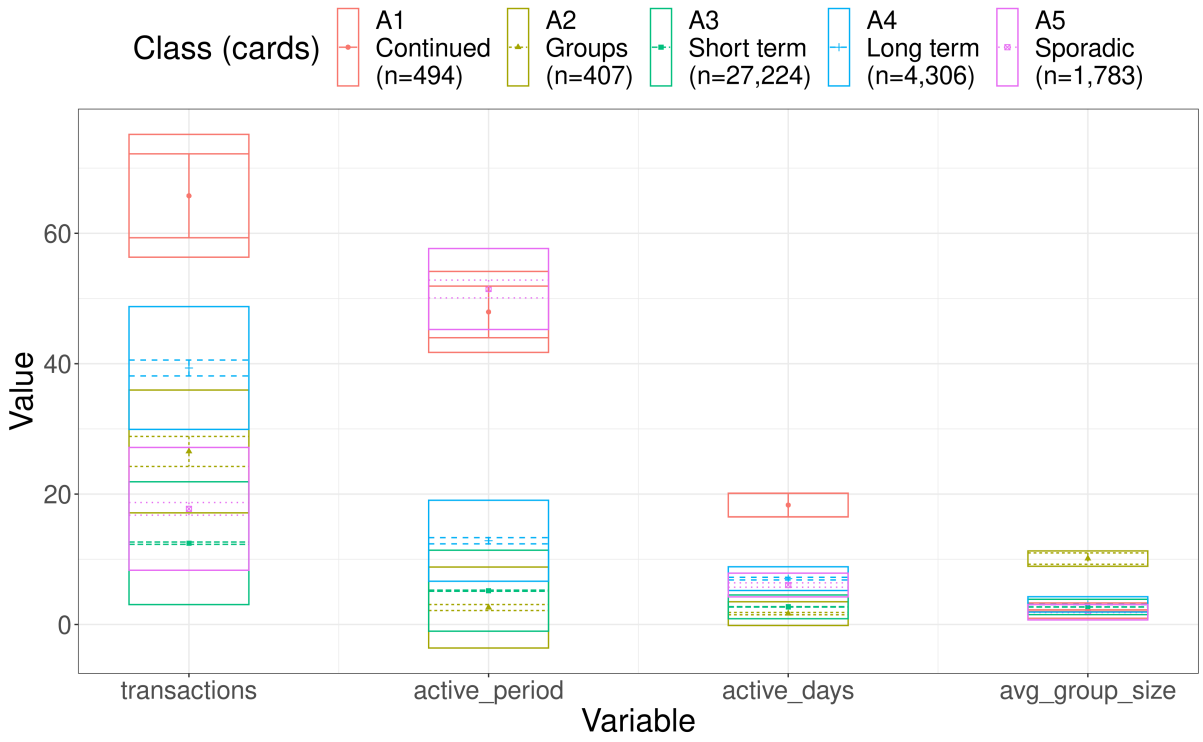


Figura 3.4: Agrupamiento de 5 perfiles de actividad mediante un gráfico de perfiles latentes: las cajas representan la desviación estándar y las barras reflejan el intervalo de confianza.

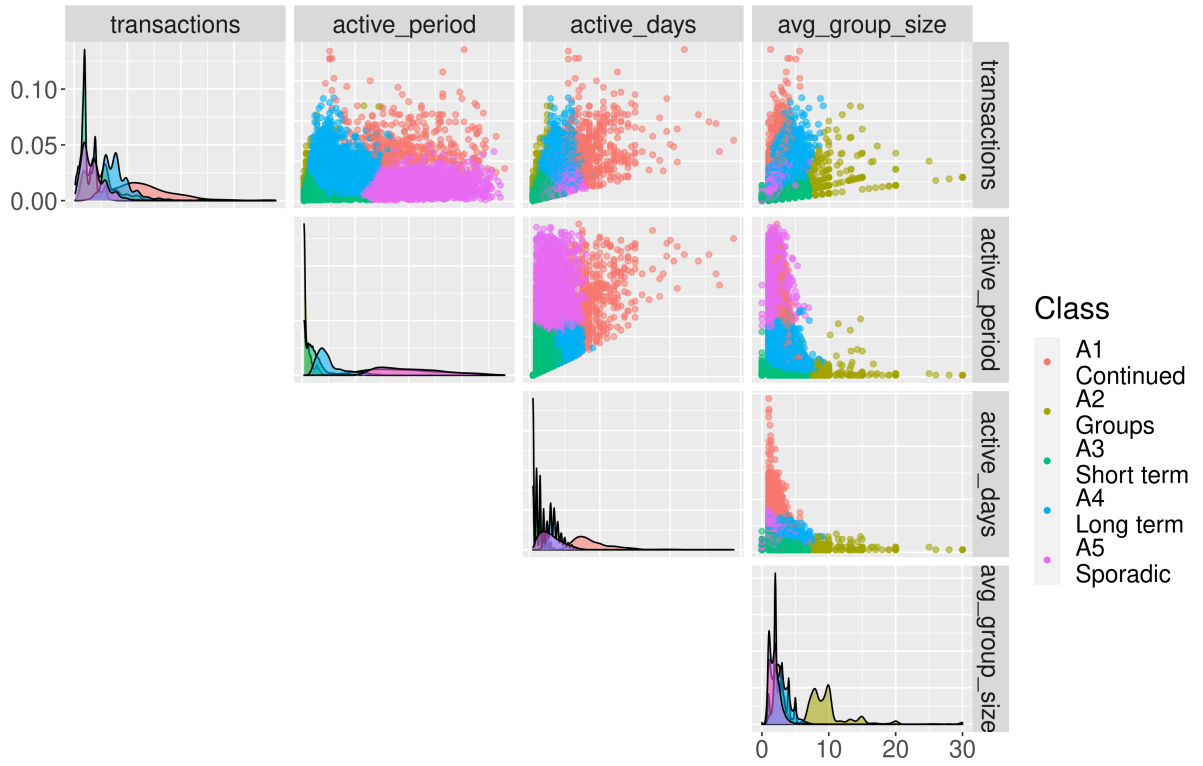


Figura 3.5: Agrupamiento de 5 perfiles de actividad mediante gráfica de dispersión.

observar que está determinado por la media de la variable `active_period`, que comparada con el resto no es de las mayores pero refleja un uso intermedio.

- **A5 Esporadico**: tarjetas con pocas transacciones a lo largo de un gran periodo de tiempo. Se puede observar que está determinado por las medias de las variables `active_period` y `transactions`, las cuales, respectivamente, son la mayor y una de las menores en comparación con el resto.

Clasificación del año 2020

Para el año 2020 se ha aprovechado que la biblioteca de tidyLPA crea un modelo de clasificación de la clase mclust. A los datos del año 2020 se le aplica este sencillo modelo de clasificación obteniéndose el resultado de la Tabla 3.6 con respecto a la clasificación de actividad.

Como se puede observar, se siguen cumpliendo las características de cada uno de los grupos explicados para el año 2019. Además es notable la gran disminución de tarjetas totales en cada uno de los perfiles. En la Figura 3.6 se puede comparar en términos porcentuales el cambio en los diferentes perfiles de un año para otro. Se produce un cambio de los perfiles hacia un perfil más esporádico.

	A1 Continued (N=135)	A2 Groups (N=38)	A3 Short term (N=3197)	A4 Long term (N=206)	A5 Sporadic (N=996)	Total (N=4562)
transactions						
Mean (SD)	65,7 (9,42)	26,9 (9,42)	12,5 (9,42)	40,3 (9,42)	17,7 (9,42)	17,2 (9,42)
active_period						
Mean (SD)	47,9 (6,21)	2,64 (6,21)	5,20 (6,21)	13,0 (6,21)	51,4 (6,21)	9,17 (6,21)
active_days						
Mean (SD)	18,3 (1,82)	1,69 (1,82)	2,71 (1,82)	7,17 (1,82)	6,04 (1,82)	3,66 (1,82)
avg_group_size						
Mean (SD)	2,13 (1,18)	10,2 (1,18)	2,69 (1,18)	3,10 (1,18)	1,88 (1,18)	2,78 (1,18)

Tabla 3.6: Resumen de las variables características para cada uno de los perfiles de la clasificación de actividad del año 2020.

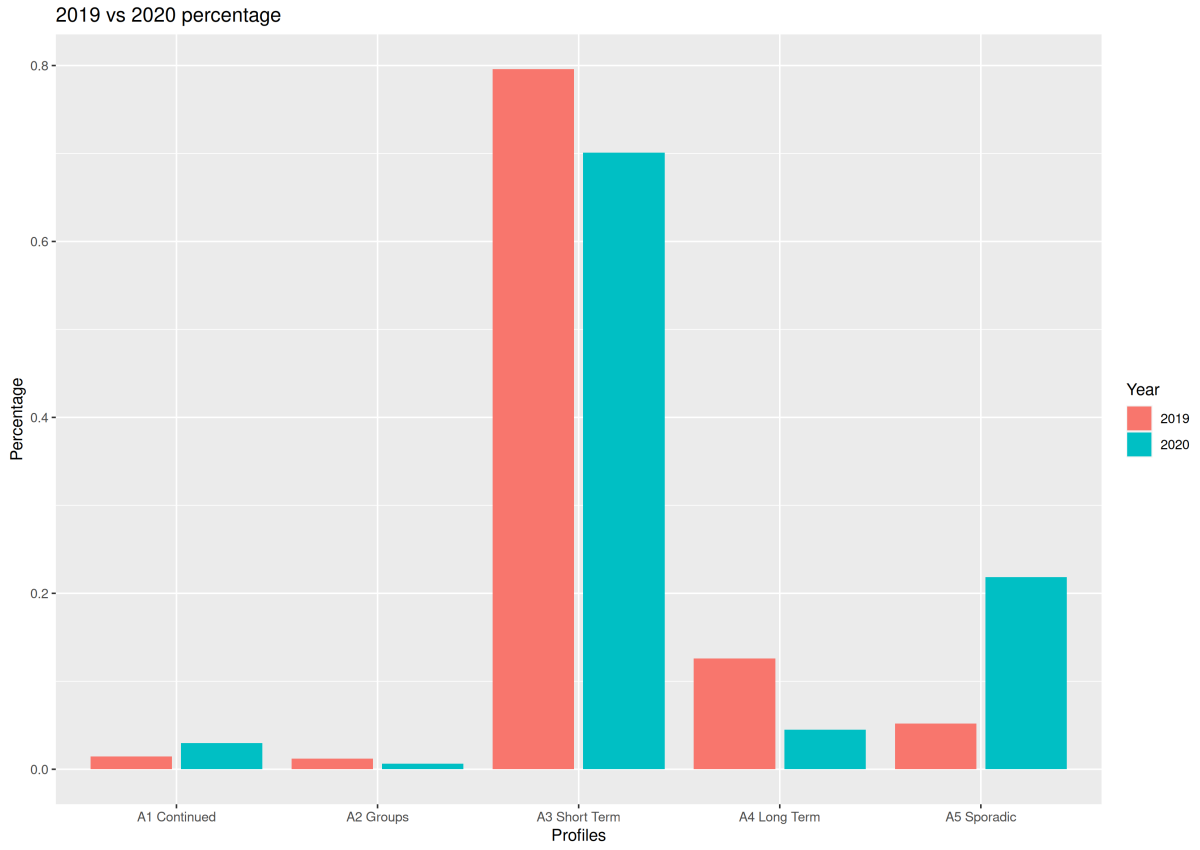


Figura 3.6: Comparación porcentual de 2019 y 2020 de actividad.

Capítulo 4

Comunicación de los resultados

Está sección pretende centrarse en el subobjetivo del proyecto de desarrollar una interfaz accesible e intuitiva para la visualización de los resultados del análisis. Para ello desde el principio del proyecto se ha planteado utilizar un cuadro de mandos, por lo cual se planifica de manera sencilla las fases o etapas de la construcción de un cuadro de mandos, divididas entre:

1. Público objetivo: antes de comenzar a trabajar en el cuadro de mandos hay que saber cuál es el público objetivo, cuáles son sus necesidades y cuáles son sus limitaciones y capacidades.
2. Prototipo: un primer paso que puede ser de utilidad antes de realizar un cuadro de mandos es la realización de uno o más prototipos. Esto ayuda a planificar y previsualizar el diseño antes de implementarlo, es una manera rápida y eficaz de verificar aquellos diseños más adecuados para la comunicación de la información sin necesidad de perder tiempo en el proceso de construcción.
3. Construcción: en esta etapa se lleva acabo el desarrollo del cuadro de mandos teniendo en cuenta lo descrito en los dos apartados anteriores como base, apoyo o punto de partida para la construcción
4. Desplegado: en esta fase se despliega el cuadro de mandos para que este disponible para el usuario final.

4.1. Público objetivo

El público objetivo del cuadro de mandos es la ATM de Camp de Tarragona. Aun así el cuadro de mandos debe estar orientado a poder ser usado en ámbitos divulgativos o académicos, con lo que debe ser lo suficientemente informativo para dar una idea acerca de los resultados sin llegar a ser muy simple o demasiado técnico.

A continuación se describen un conjunto de requisitos funcionales que debe cumplir el cuadro de mandos:

- Permitir buscar los resultados del análisis por mes y por año.
- Poder consultar los resultados interactivamente.
- Sincronizar la visualización de los gráficos mostrados.
- Los clústeres deben poderse comparar por año.
- Poder explorar una muestra de los datos del análisis.
- Describir las vistas y gráficos mostrados.

4.2. Prototipo

En caso de este proyecto se ha decantado por un prototipo en papel para el cuadro de mandos. El prototipo se divide en tres secciones, que se corresponde con tres páginas distintas:

- Una primera página, como muestra la Figura 4.1, con el resumen de estadísticos agregados que sirva para comprobar las principales características del análisis. Esta página incluirá una vista geográfica del área de estudio con marcadores y *overlays*.
- Una segunda página, como muestra la Figura 4.2, compuesta de la información acerca de los diferentes clústeres. Esta página sera menos dinámica que la anterior, estará formada por gráficos estáticos pregenerados y tablas que dan información sobre los diferentes clústeres.
- Una tercera página, como muestra la Figura 4.3, que presentará una muestra del conjunto de datos utilizando una tabla.

4.3. Construcción

La construcción del cuadro de mandos se realiza con la biblioteca `flexdashboard` utilizando su integración con la biblioteca `shiny`.

Durante la fase de construcción, el diseño del cuadro de mandos cambia ligeramente. Las tres páginas pasan a tener el diseño explicado en las siguientes secciones.

4.3.1. Primera página: resumen de estadísticos agregados mensual y anualmente por parada

Como muestra la Figura 4.4, en está página se presenta un conjunto de datos agregados que dan una visión más general y un carácter mas geográfico. Aquí se presentan las transacciones realizadas durante los años 2019 y 2020 agregadas por año, por mes, por parada y por tarifas. Para está página se ha elegido utilizar una estructura clásica de `shiny`, un conjunto de entradas

Figura 4.1: Prototipo de la primera página del cuadro de mandos.

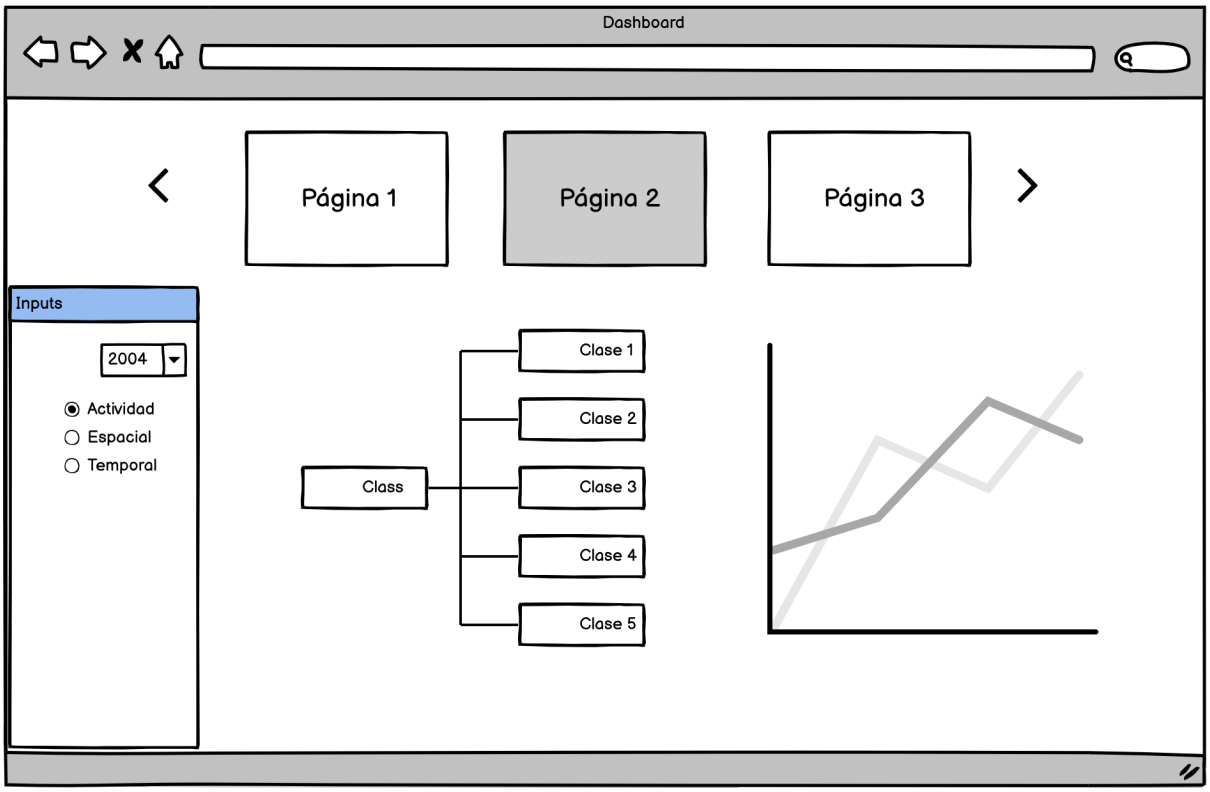


Figura 4.2: Prototipo de la segunda página del cuadro de mandos.

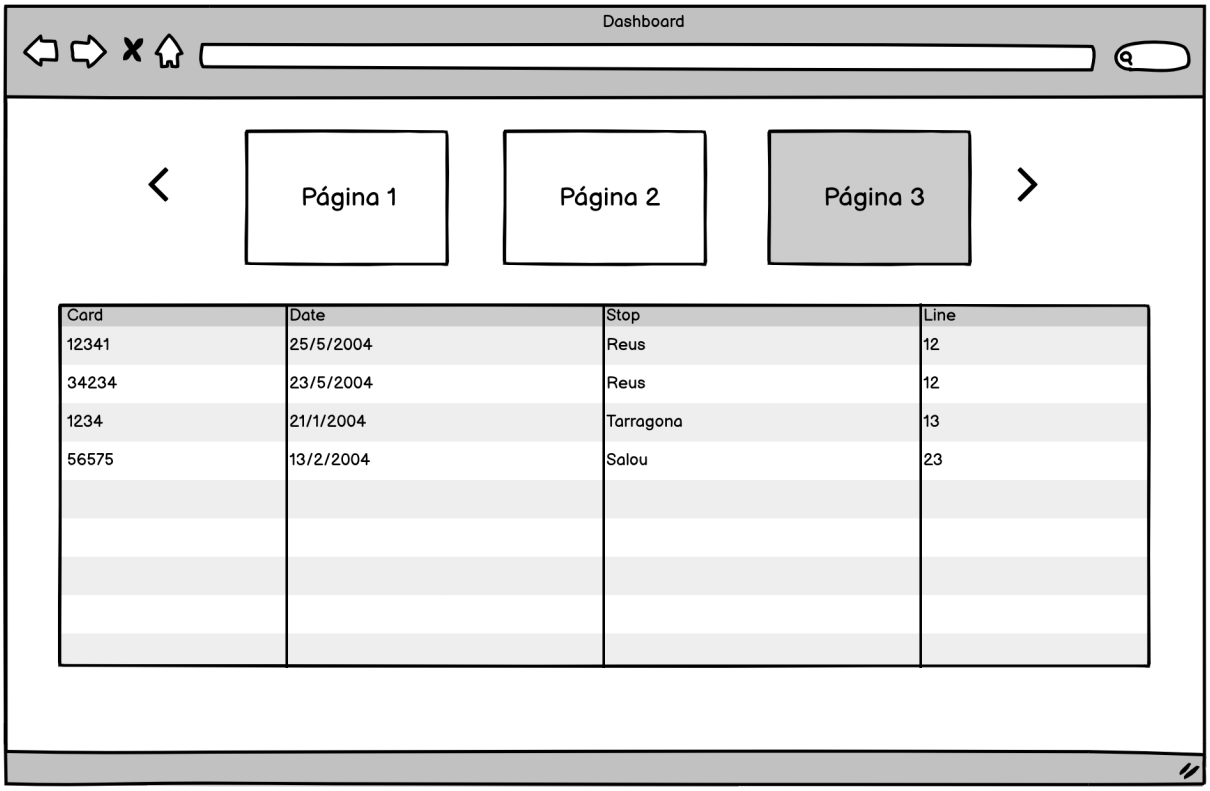


Figura 4.3: Prototipo de la tercera página del cuadro de mandos.

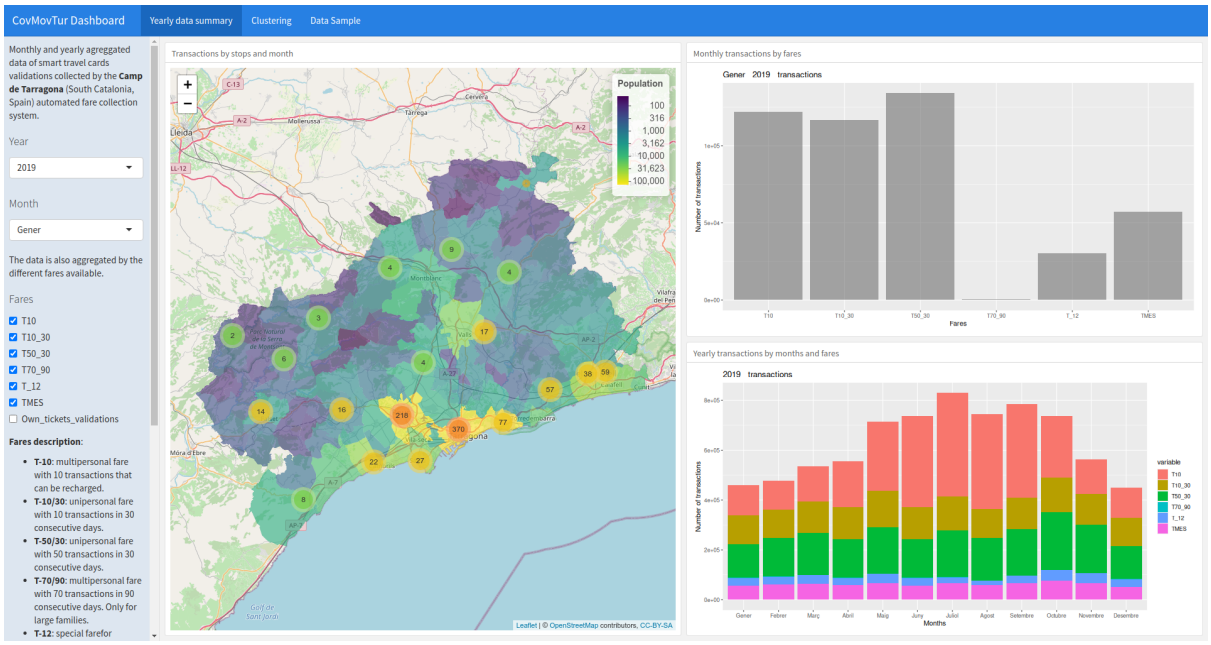


Figura 4.4: Primera página del cuadro de mandos.

de datos relacionadas con un conjunto de salidas mediante el concepto de reactividad: cualquier cambio en las entradas produce una notificación de cambio en las salidas para la reejecución del código. Esta página incluye:

- Una barra lateral en el lado izquierdo que contiene las entradas, como muestra la Figura 4.5a. Estas se dividen en desplegables para la selección del año y del mes, y en cajas de selección para las tarifas.

- Una vista geográfica del área de estudio usando la biblioteca Leaflet, como muestra la Figura 4.5b. Contiene overlays y marcadores que muestran la población de cada municipio y las transacciones por paradas para las tarifas, año y mes seleccionados. Además esta vista geográfica también funciona como una entrada, pues devuelve el conjunto de municipios sobre los que está centrada la vista, pudiendo de esta manera alimentar el segundo elemento gráfico de la página.

- Un gráfico de barras, como muestra la Figura 4.6, en el que se presentan las transacciones del año y del mes seleccionados por paradas, pero con la peculiaridad de que además usa como entrada los municipios en los que está centrada la vista gráfica.

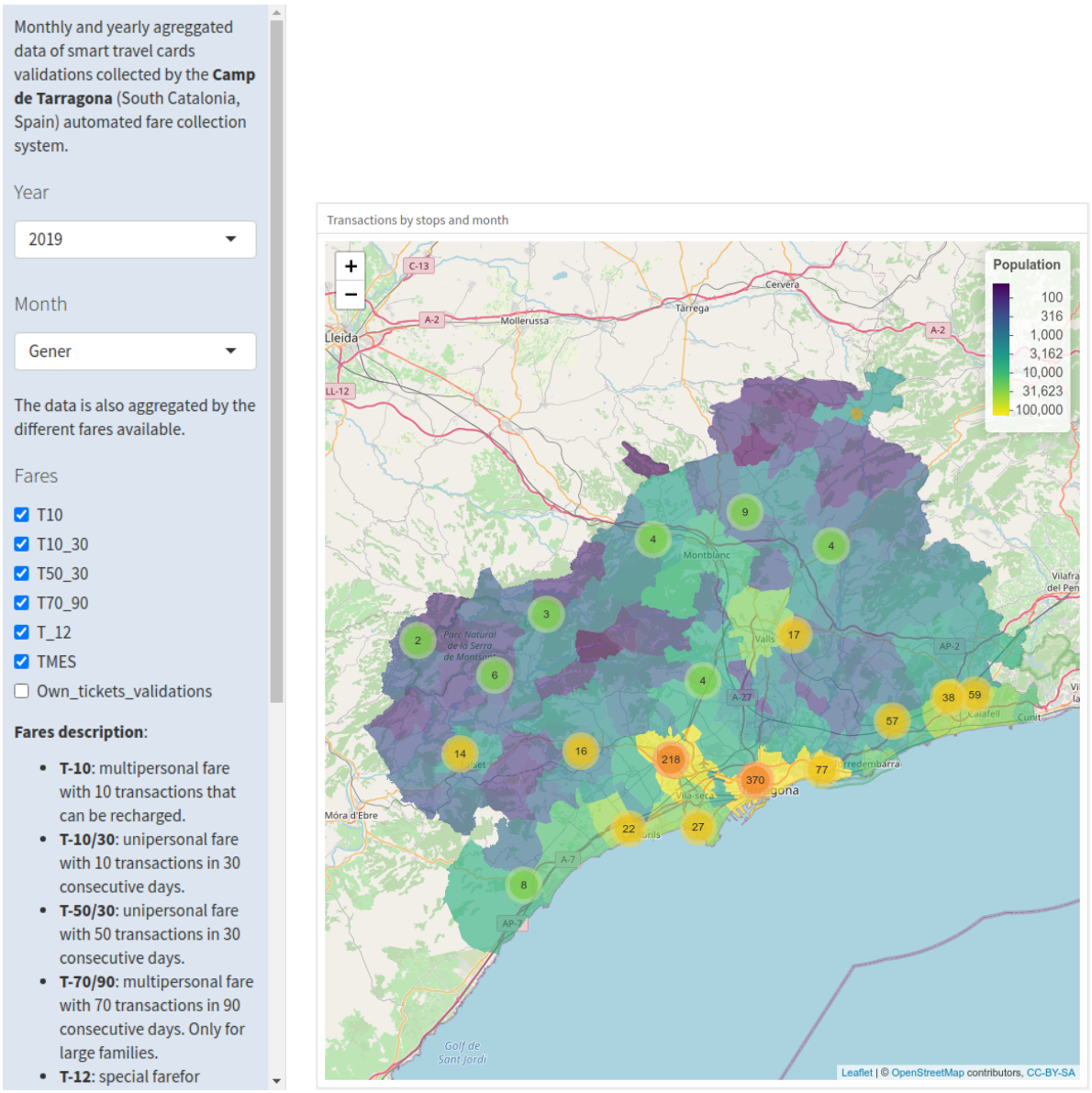
- Una tercer elemento gráfico, como se observa en la Figura 4.7, que muestra las transacciones realizadas en el año seleccionado por mes y por tarifa seleccionada.

4.3.2. Segunda página: resultados del agrupamiento sobre las tarjetas T-10 de verano

Como muestra la Figura 4.8, esta página está compuesta de la información acerca de los diferentes clústeres. En está página se presentan los resultados del análisis separando los diferentes perfiles de actividad, tiempo y espacio. Esta página será menos dinámica que la anterior, estará formada por gráficos estáticos pregenerados y tablas que nos dan información sobre los diferentes clústeres.

Si se observa detenidamente está página, tiene el diseño en diapositivas que se había planteado para el cuadro de mandos al realizar el prototipo. Debido a ciertas restricciones de `flexdashboard` este diseño se tuvo que reimplementar, además de que la página entera ha tenido que evitar el uso de módulos de `shiny` después de varias iteraciones diferentes durante la construcción (este es el problema que se había mencionado en la Sección 2.5 de seguimiento del proyecto). Aquí cabe una reflexión, durante el prototipado no se había especificado de manera concisa el diseño de está página, simplemente se planifico que fuese una página con gráficos incrustados. Con una mejor definición de está página y un mejor conocimiento previo de `flexdashboard`, de sus problemas y limitaciones, se podría haber minimizado el número de iteraciones distintas hasta conseguir un resultado final libre de problemas.

El diseño de esta pagina se divide en conjuntos de dos diapositivas similares para cada uno de los perfiles de actividad, tiempo y espacio. Cada diapositiva dispone de una barra lateral en la derecha que la describe. Las dos diapositivas se dividen en:



(a) Barra lateral. (b) Vista geográfica.

Figura 4.5: Barra lateral y vista geográfica de la primera página del cuadro de mandos.

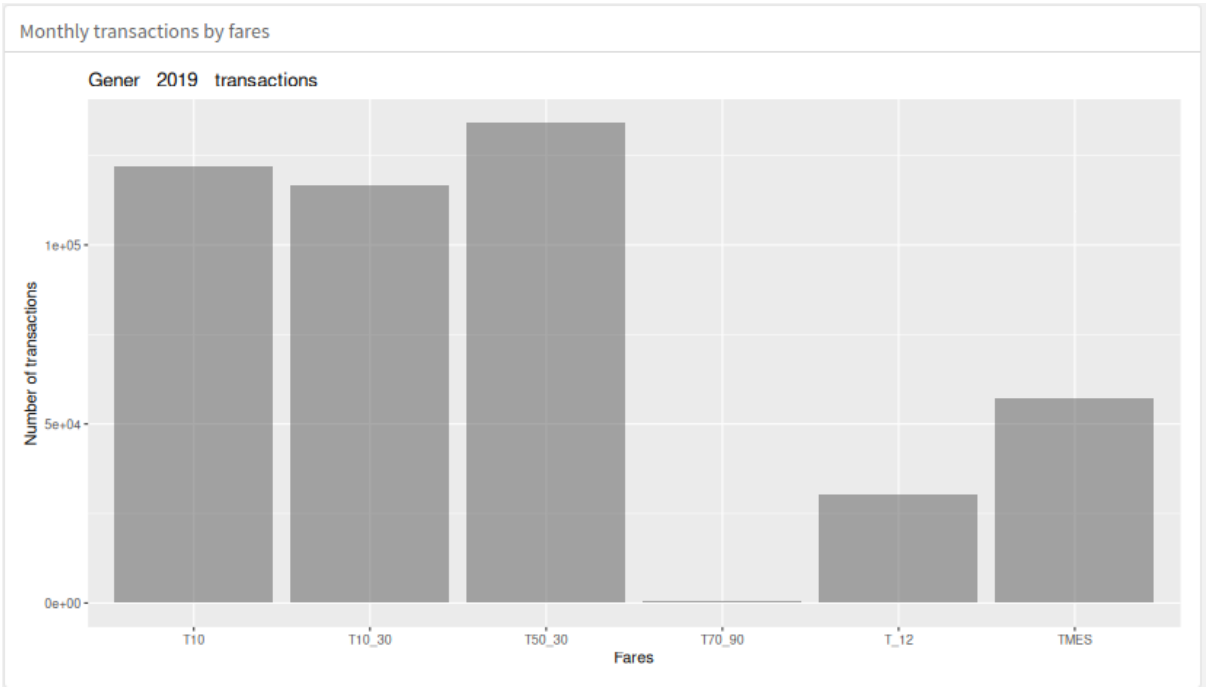


Figura 4.6: Gráfico de barras de las transacciones por tarifa en el mes y año seleccionados. En la primera página del cuadro de mandos.

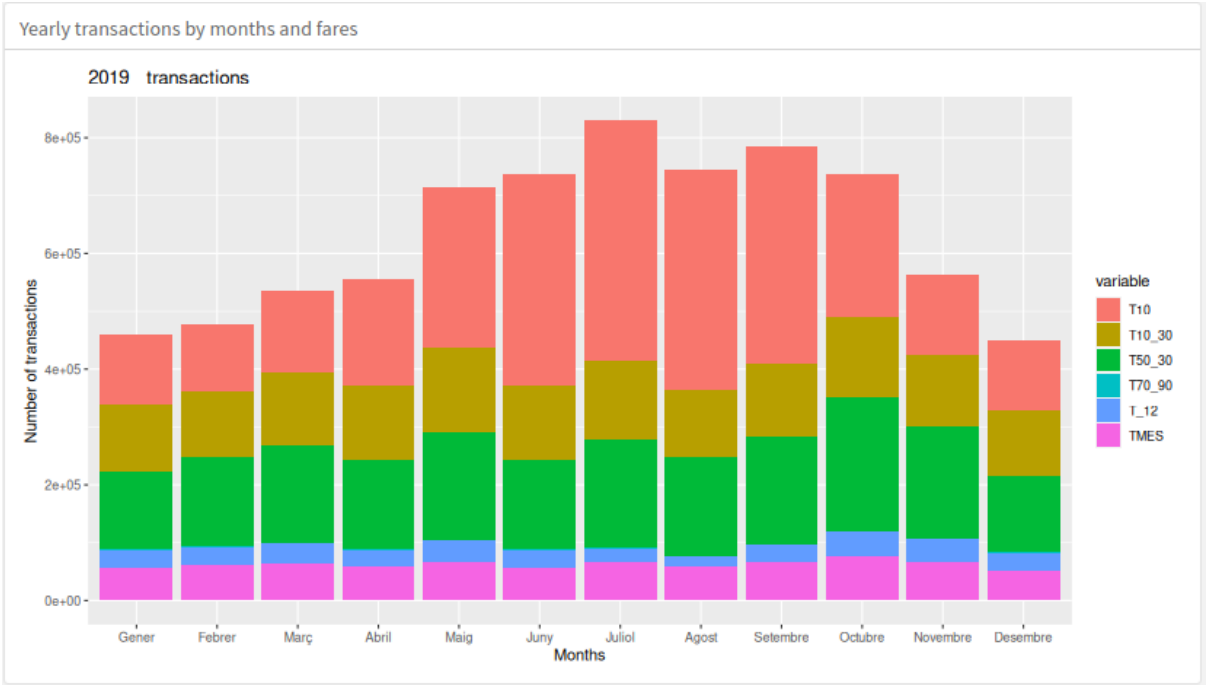


Figura 4.7: Gráfico de barras de las transacciones anuales por mes y tarifa. En la primera página del cuadro de mandos.

- Una primera diapositiva, como muestra la Figura 4.9, con dos gráficos que representan el resultado del agrupamiento: un gráfico de cajas para cada una de las variables características y un gráfico de dispersión en el que cada punto está agrupado en sus respectivas clases y representa la relación por pares entre las variables características. Se ha usando la biblioteca `ggplot2` para la generación de los gráficos.
- Una segunda diapositiva, como muestra la Figura 4.10, con tablas de estadísticos descriptivos en los que se presentan las clases resultantes del agrupamiento para el año 2019 y la clasificación de los datos del 2020 siguiendo el mismo modelo del 2019. Se ha usando la biblioteca `table1` para la generación de las tablas.

4.3.3. Tercera página: muestra del conjunto de datos

Como se muestra en la Figura 4.11, esta página está compuesta por una tabla con una muestra de los datos. También presenta una barra lateral izquierda con información acerca de lo que se muestra en pantalla. La tabla ha sido generada usando la biblioteca `DT` (DataTables), que entre otras muchas opciones, permite filtrar, ordenar y realizar búsquedas en los datos.

4.4. Desplegado

La biblioteca `shiny` distribuye un servidor para poder desplegar las *web apps* que se creen usándola. Este se distribuye a través de un contenedor Docker el cual hay que configurar y construir con las diferentes bibliotecas que vayamos a utilizar. El servidor de `shiny` no solo es capaz de ejecutar *web apps* creadas con esta biblioteca, además es capaz de ejecutar el formato de archivo de R markDown. Esto permite usar `flexdashboard` como biblioteca para la creación de cuadros de mandos que desplegaremos en un contenedor de `shiny`.

4.5. Verificación y test de usuario

Las pruebas, la validación y los tests de usuario se realizaron al acabar junto con el supervisor. Al ser un proyecto centrado en la visualización final de los datos, los test de usuario se realizaron sobre el cuadro de mandos construido. Se comprobaron y corrigieron posteriormente aspectos como la correcta descripción de la información mostrada, la legibilidad de los elementos del cuadro de mandos y la inteligibilidad de las diferentes gráficas mostradas.

Para comprobar la usabilidad, inteligibilidad y cumplimiento de los requisitos del cuadro de mandos se realizó una reunión. En dicha reunión se recibió bastante retroalimentación.

En primer lugar se comentaron aspectos sobre la inteligibilidad de la primera pagina del cuadro de mandos. En una primera versión, las distintas gráficas y vistas tenían propósitos y fines diferentes, no existía un elemento conector entre ellas, dificultando así su entendimiento. Para corregirlo se propuso conectar diferentes elementos con los datos presentados en la vista

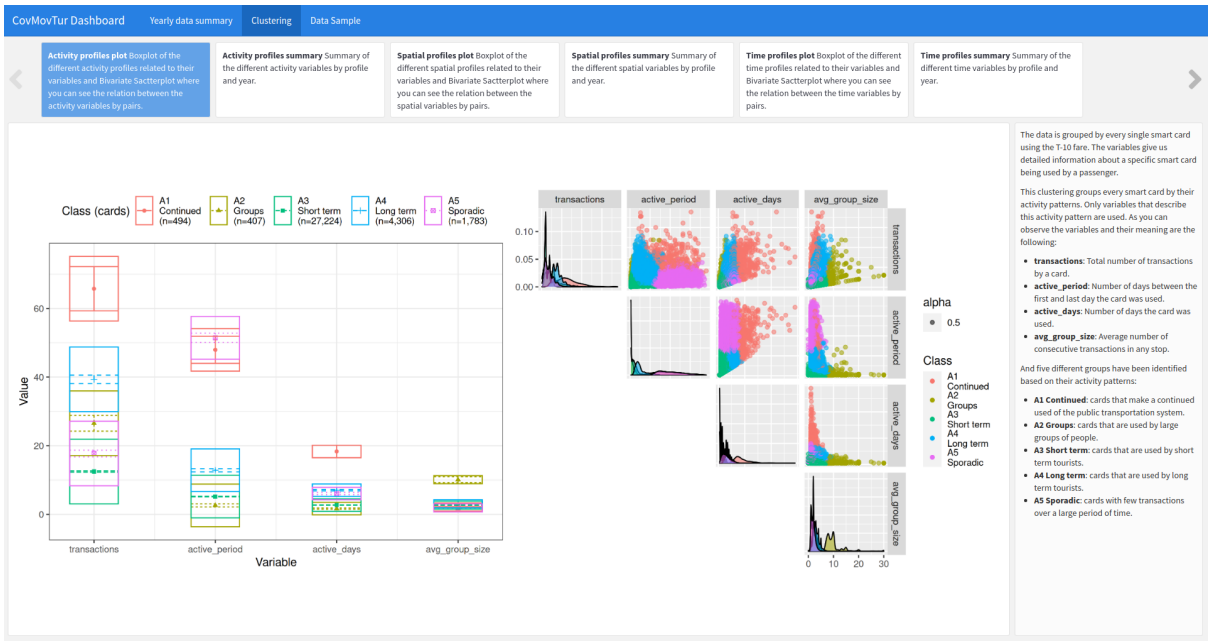


Figura 4.8: Segunda página del cuadro de mandos.

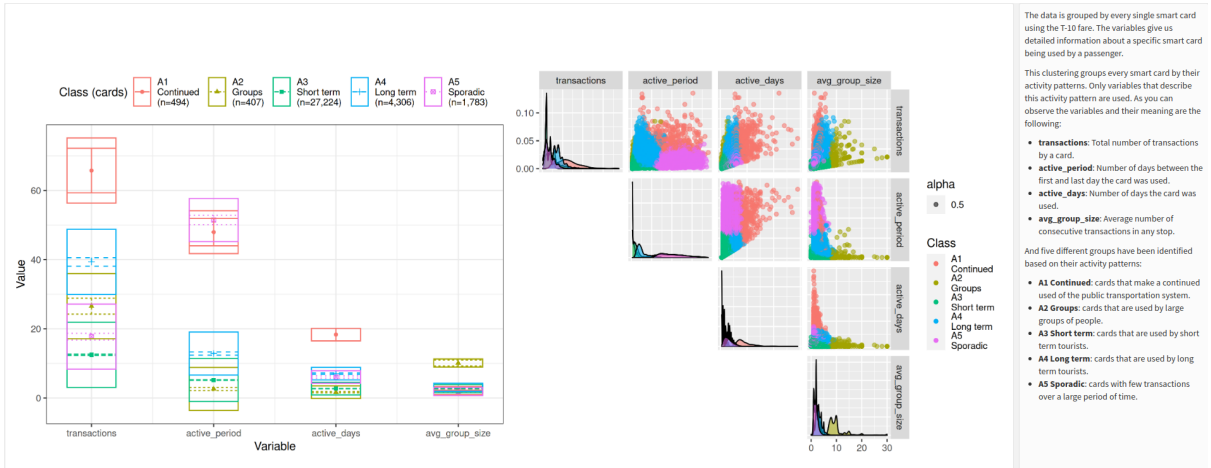
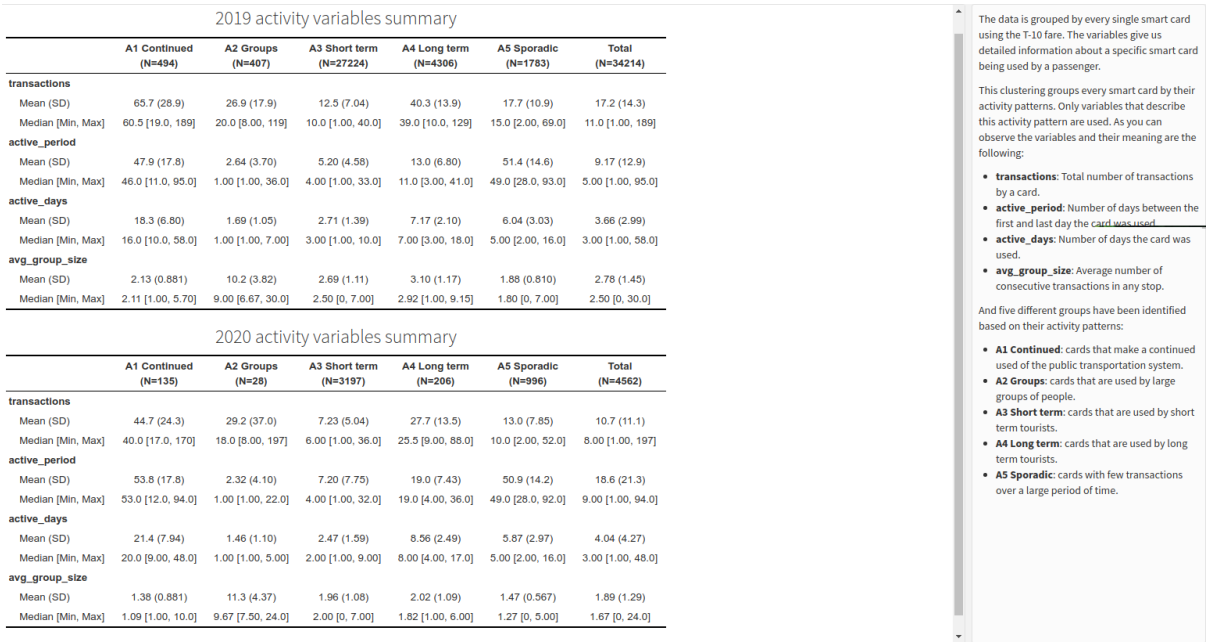


Figura 4.9: Primera diapositiva de la segunda página del cuadro de mandos.

geográfica. Esto se puede comprobar en la sección Subsección 4.3.1, donde se describe el cuadro de mandos con dicho cambio implementado.

En segundo lugar, para versiones futuras, se propuso añadir un tercer gráfico en la primera pagina del cuadro de mandos que proporcione información acerca del número de transacciones diarias durante el mes seleccionado. Además se comento la posibilidad de actualizar el *overlay* de la vista geográfica para que sea más explicito e inteligible.



2019 activity variables summary

	A1 Continued (N=494)	A2 Groups (N=407)	A3 Short term (N=27224)	A4 Long term (N=4306)	A5 Sporadic (N=1783)	Total (N=34214)
transactions						
Mean (SD)	65.7 (28.9)	26.9 (17.9)	12.5 (7.04)	40.3 (13.9)	17.7 (10.9)	17.2 (14.3)
Median [Min, Max]	60.5 [19.0, 189]	20.0 [8.00, 119]	10.0 [1.00, 40.0]	39.0 [10.0, 129]	15.0 [2.00, 69.0]	11.0 [1.00, 189]
active_period						
Mean (SD)	47.9 (17.8)	2.64 (3.70)	5.20 (4.58)	13.0 (6.80)	51.4 (14.6)	9.17 (12.9)
Median [Min, Max]	46.0 [11.0, 95.0]	1.00 [1.00, 36.0]	4.00 [1.00, 33.0]	11.0 [3.00, 41.0]	49.0 [28.0, 93.0]	5.00 [1.00, 95.0]
active_days						
Mean (SD)	18.3 (6.80)	1.69 (1.05)	2.71 (1.39)	7.17 (2.10)	6.04 (3.03)	3.66 (2.99)
Median [Min, Max]	16.0 [10.0, 58.0]	1.00 [1.00, 7.00]	3.00 [1.00, 10.0]	7.00 [3.00, 18.0]	5.00 [2.00, 16.0]	3.00 [1.00, 58.0]
avg_group_size						
Mean (SD)	2.13 (0.881)	10.2 (3.82)	2.69 (1.11)	3.10 (1.17)	1.88 (0.810)	2.78 (1.45)
Median [Min, Max]	2.11 [1.00, 5.70]	9.00 [6.67, 30.0]	2.50 [0, 7.00]	2.92 [1.00, 9.15]	1.80 [0, 7.00]	2.50 [0, 30.0]

2020 activity variables summary

	A1 Continued (N=135)	A2 Groups (N=28)	A3 Short term (N=3197)	A4 Long term (N=206)	A5 Sporadic (N=996)	Total (N=4562)
transactions						
Mean (SD)	44.7 (24.3)	29.2 (37.0)	7.23 (5.04)	27.7 (13.5)	13.0 (7.85)	10.7 (11.1)
Median [Min, Max]	40.0 [17.0, 170]	18.0 [8.00, 197]	6.00 [1.00, 36.0]	25.5 [9.00, 88.0]	10.0 [2.00, 52.0]	8.00 [1.00, 197]
active_period						
Mean (SD)	53.8 (17.8)	2.32 (4.10)	7.20 (7.75)	19.0 (7.43)	50.9 (14.2)	18.6 (21.3)
Median [Min, Max]	53.0 [12.0, 94.0]	1.00 [1.00, 22.0]	4.00 [1.00, 32.0]	19.0 [4.00, 36.0]	49.0 [28.0, 92.0]	9.00 [1.00, 94.0]
active_days						
Mean (SD)	21.4 (7.94)	1.46 (1.10)	2.47 (1.59)	8.56 (2.49)	5.87 (2.97)	4.04 (4.27)
Median [Min, Max]	20.0 [9.00, 48.0]	1.00 [1.00, 5.00]	2.00 [1.00, 9.00]	8.00 [4.00, 17.0]	5.00 [2.00, 16.0]	3.00 [1.00, 48.0]
avg_group_size						
Mean (SD)	1.38 (0.881)	11.3 (4.37)	1.96 (1.08)	2.02 (1.09)	1.47 (0.567)	1.89 (1.29)
Median [Min, Max]	1.09 [1.00, 10.0]	9.67 [7.50, 24.0]	2.00 [0, 7.00]	1.82 [1.00, 6.00]	1.27 [0, 5.00]	1.67 [0, 24.0]

The data is grouped by every single smart card using the T-10 fare. The variables give us detailed information about a specific smart card being used by a passenger.

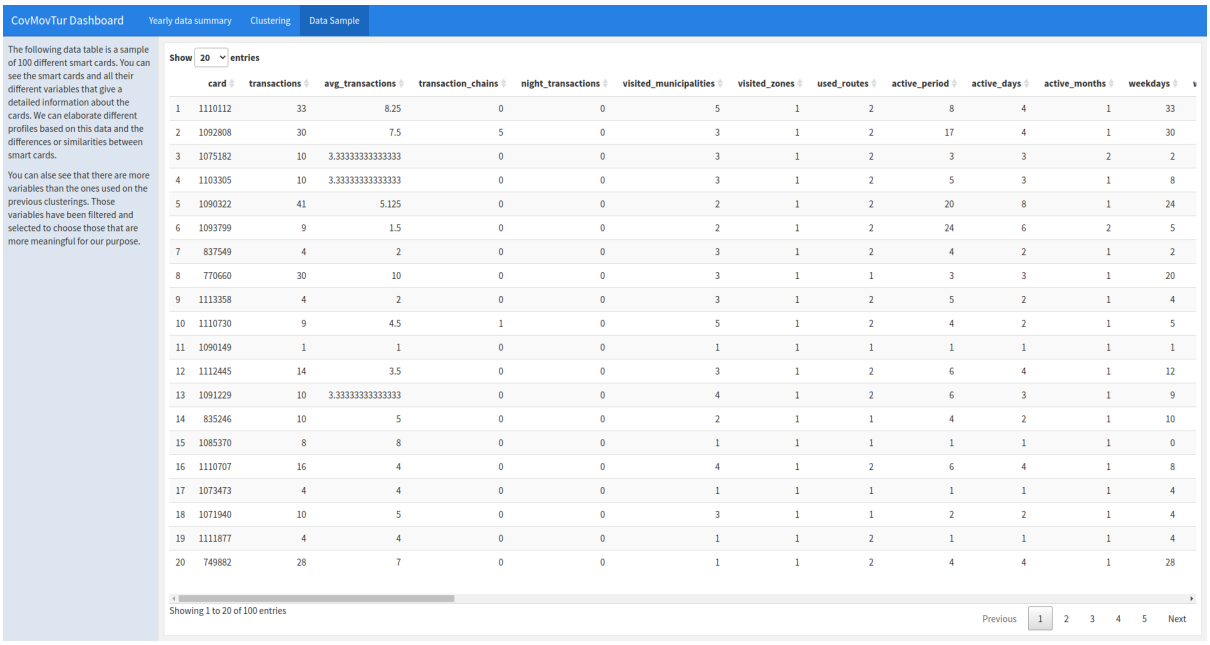
This clustering groups every smart card by their activity patterns. Only variables that describe this activity pattern are used. As you can observe the variables and their meaning are the following:

- **transactions**: Total number of transactions by a card.
- **active_period**: Number of days between the first and last day the card was used.
- **active_days**: Number of days the card was used.
- **avg_group_size**: Average number of consecutive transactions in any stop.

And five different groups have been identified based on their activity patterns:

- **A1 Continued**: cards that make a continued used of the public transportation system.
- **A2 Groups**: cards that are used by large groups of people.
- **A3 Short term**: cards that are used by short term tourists.
- **A4 Long term**: cards that are used by long term tourists.
- **A5 Sporadic**: cards with few transactions over a large period of time.

Figura 4.10: Segunda diapositiva de la segunda página del cuadro de mandos.



CovMovTur Dashboard | Yearly data summary | Clustering | Data Sample

The following data table is a sample of 100 different smart cards. You can see the smart cards and all their different variables that give a detailed information about the cards. We can elaborate different profiles based on this data and the differences or similarities between smart cards.

You can alse see that there are more variables than the ones used on the previous clusterings. Those variables have been filtered and selected to choose those that are more meaningful for our purpose.

Show 20 entries

	card	transactions	avg_transactions	transaction_chains	night_transactions	visited_municipalities	visited_zones	used_routes	active_period	active_days	active_months	weekdays
1	1110112	33	8.25	0	0	5	1	2	8	4	1	33
2	1092808	30	7.5	5	0	3	1	2	17	4	1	30
3	1075182	10	3.33333333333333	0	0	3	1	2	3	3	2	2
4	1103305	10	3.33333333333333	0	0	3	1	2	5	3	1	8
5	1090322	41	5.125	0	0	2	1	2	20	8	1	24
6	1093799	9	1.5	0	0	2	1	2	24	6	2	5
7	837549	4	2	0	0	3	1	2	4	2	1	2
8	770660	30	10	0	0	3	1	1	3	3	1	20
9	1113358	4	2	0	0	3	1	2	5	2	1	4
10	1110730	9	4.5	1	0	5	1	2	4	2	1	5
11	1090149	1	1	0	0	1	1	1	1	1	1	1
12	1112445	14	3.5	0	0	3	1	2	6	4	1	12
13	1091229	10	3.33333333333333	0	0	4	1	2	6	3	1	9
14	835246	10	5	0	0	2	1	1	4	2	1	10
15	1085370	8	8	0	0	1	1	1	1	1	1	0
16	1110707	16	4	0	0	4	1	2	6	4	1	8
17	1073473	4	4	0	0	1	1	1	1	1	1	4
18	1071940	10	5	0	0	3	1	1	2	2	1	4
19	1111877	4	4	0	0	1	1	2	1	1	1	4
20	749882	28	7	0	0	1	1	2	4	4	1	28

Showing 1 to 20 of 100 entries

Previous 1 2 3 4 5 Next

Figura 4.11: Tercera página del cuadro de mandos.

Capítulo 5

Conclusiones

5.1. Resultados del proyecto

Al final del proyecto se han conseguido cumplir los objetivos del mismo. Principalmente se ha encontrado una herramienta eficiente y escalable para el almacenamiento de los datos, se ha encontrado un método y unas variables descriptivas para poder describir y analizar los datos, y por último se ha desarrollado un cuadro de mandos accesible e intuitivo para la visualización de los resultados. Además se ha documentado cada uno de los pasos del mismo.

En cuanto a las herramientas utilizadas surgen diferentes opiniones. Todas ellas tienen grandes facilidades para integrarse conjuntamente y llevar a cabo los objetivos de este proyecto u otros proyectos futuros. El almacenamiento en Google BigQuery cumple con todas nuestras expectativas, además de ofrecernos muchísimas más capacidades aún no exploradas, sin contar con las facilidades de escalabilidad en caso de crecimiento del proyecto. El lenguaje de programación R no solo provee de varias bibliotecas para casi cualquier tipo de análisis estadístico sobre los datos, también provee de bibliotecas para casi cualquier operación de transformación o visualización que se quiera realizar con los datos. Ambas herramientas y las respectivas bibliotecas forman un ecosistema que facilita la construcción de un proyecto de ciencia de datos. Aun así no todas las bibliotecas presentan facilidades de integración, sobre todo aquellas dedicadas a la visualización. Pero también hay que tener en cuenta ejemplos como los de `shiny` o `flexdashboard`, cuya integración es ambigua dependiendo de las respectivas funciones utilizadas.

En resumen, todas las herramientas utilizadas y las respectivas implementaciones se adecuan perfectamente no solo a la arquitectura planteada en Sección 1.4, sino que pueden ofrecer mucho más para futuros proyectos. Todas y cada una de las herramientas y bibliotecas utilizadas se han integrado perfectamente para resolver los objetivos planteados, sin contar con los posibles inconvenientes derivados de algunas bibliotecas de visualización.

5.2. Conclusiones personales

Desde un punto de vista formativo, he tenido que abrir mi mente a la realidad de que por mucho que llegue a saber sobre alguna herramienta o lenguaje siempre va a ver algo nuevo o un nuevo punto de vista por conocer. Mi experiencia con el lenguaje de programación R ha sido agridulce, porque por mucho que avanzaba, y con ello mi satisfacción personal, nunca llegaba a ver el final, siempre había una nueva biblioteca que implicaba un nuevo punto de vista o manera de proceder. También he llegado a comprender la importancia de SQL. Este ha llegado a superar las expectativas que me había generado durante mi periodo de formación.

A nivel personal creo que este proyecto ha cambiado mi perspectiva en muchas áreas relacionadas con la ciencia de datos, veo que ha generado en mi una necesidad de comprensión más teórica al mismo tiempo que me ha hecho darme cuenta de la importancia de los conocimientos prácticos en la materia. Me gustaría continuar trabajando en esta materia, sobre todo explorando nuevos modelos matemáticos, sus aplicaciones y las herramientas necesarias para ello.

Bibliografía

[1] UBIK Geospatial Solutions S.L. http://www.ubikgs.com/. [Consulta: 11 de Marzo de 2022].

[2] Geotec consigue financiación para el proyecto COVMOVTUR dentro de la convocatoria FONDO SUPERA COVID-19. https://www.init.uji.es/covmovtur. [Consulta: 11 de Marzo de 2022].

[3] What is BigQuery? https://cloud.google.com/bigquery/docs/introduction. [Consulta: 6 de Junio de 2022].

[4] Leaflet for R. https://rstudio.github.io/leaflet/. [Consulta: 6 de Junio de 2022].

[5] Sphinx (documentation generator). https://en.wikipedia.org/wiki/Sphinx_(documentation_generator). [Consulta: 6 de Junio de 2022].

[6] Docker (software). https://en.wikipedia.org/wiki/Docker_(software). [Consulta: 6 de Junio de 2022].

[7] Metodología Fundamental para la Ciencia de Datos. https://www.ibm.com/downloads/cas/6RZMKDN8. [Consulta: 18 de Junio de 2022].

[8] Gutiérrez, A., Domènech, A., Zaragozí, B., Miravet, D. Profiling tourists' use of public transport through smart travel card data. *Journal of Transport Geography*, 88:102820, 2020.

[9] General Transit Feed Specification (GTFS). https://developers.google.com/transit/gtfs. [Consulta: 5 de Marzo de 2022].

[10] D. Oberski. *Mixture Models: Latent Profile and Latent Class Analysis*, pages 275–287. Springer International Publishing, 2016.

[11] Henning, C., Meila, M., Murtagh, F., Rocci, R. *Handbook of Cluster Analysis*. Chapman & Hall, 2015.

Anexo A

Documentación del proyecto

A.1. Introducción

Este anexo está dedicado a presentar la documentación del proyecto y su estructura. Se describirán los principales temas revisados durante la fase de aprendizaje. La documentación está escrita usando markDown y la documentación se construye usando la Sphinx, un generador de documentos escrito en Python.

No todos los temas aquí expuestos son igual de relevantes. Algún tema relativo a BigQuery puede no tener gran relevancia o uso para el proyecto, pero han ayudado en el aprendizaje de Google BigQuery. Hay temas como las rutinas que seguramente se hubiesen merecido una mayor atención.

A.2. Organización

- Google BigQuery: se hace una descripción de los principales temas desconocidos de antemano o que se piensa que pueden llegar a ser usados durante el proyecto.
    - Carga de datos: descripción de las distintas formas de cargar los datos a Google BigQuery con sus respectivos pasos. Como muestra la Figura A.1
    - Particiones: descripción de las distintas formas de crear particiones en las tablas con sus respectivos pasos. Como muestra laFigura A.2
    - Clústeres: descripción de las distintas formas de crear clústeres en las tablas con sus respectivos pasos. Como muestra la Figura A.3
    - Arrays y Structs: se presentan dos tipos de datos complejos como son los arrays y los structs. Como muestra la Figura A.4
    - Clausula OVER - funciones analíticas: se describe el funcionamiento de las funciones analíticas, que al contrario que las funciones de agregación computan valores sobre un conjunto de filas y devuelven un resultado para cada fila. Como muestra la Figura A.5

- R: aunque no esté dentro de la planificación, como iniciativa personal, se pretendía documentar el proceso de aprendizaje de las principales bibliotecas de R utilizadas, como se pude observar está sección está comenzada pero no se ha llegado a finalizar.
    - Shiny: una pequeña guía de los principales conceptos de la biblioteca `shiny`. Como muestra la Figura A.6
    - ggplot: una pequeña guía de los principales conceptos de la biblioteca `ggplot2`. Como muestra la Figura A.7
- Proyecto: se presentan los pasos realizados para cargar, transformar y extraer muestras y variables características a partir de los datos.
    - Subida y transformación de datos: los pasos para subir, limpiar, transformar e imputar los datos son explicados en esta sección. Como muestra la Figura A.8
    - Principales consultas: primero se presentan los pasos para extraer las muestras a usar durante el análisis. Al final se presentan las variables características y los pasos para extraerlas de las muestras. Como muestran la Figura A.9 y la Figura A.10

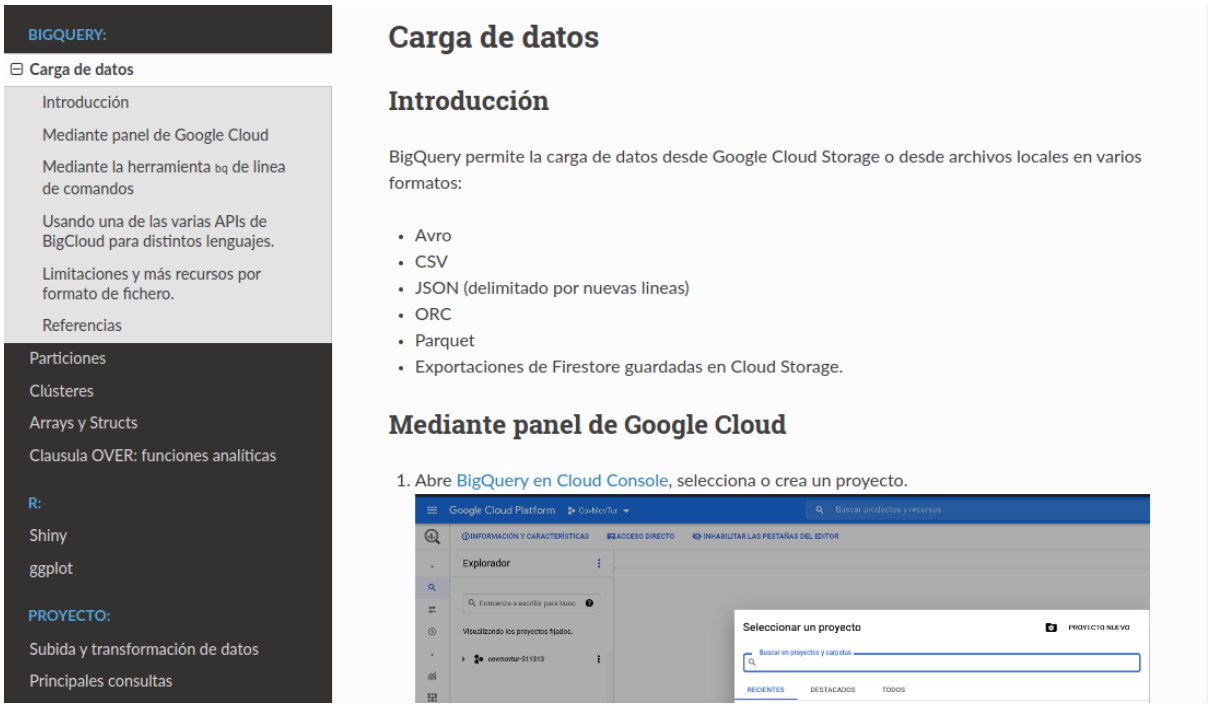


Figura A.1: Sección de carga de datos de la documentación.

Figura A.2: Sección de particiones de la documentación.

Figura A.3: Sección de clústeres de la documentación.

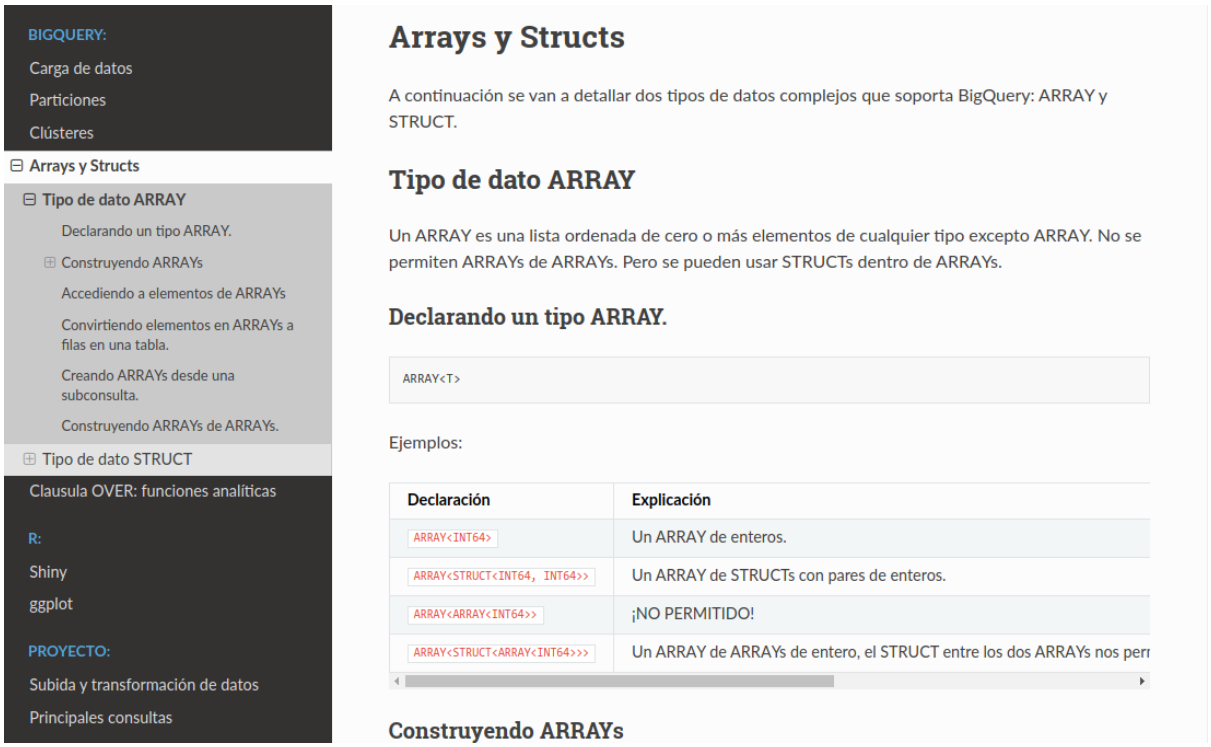


Figura A.4: Sección de arrays y structs de la documentación.

Figura A.5: Sección de funciones analíticas de la documentación.

Figura A.6: Sección de shiny de la documentación.

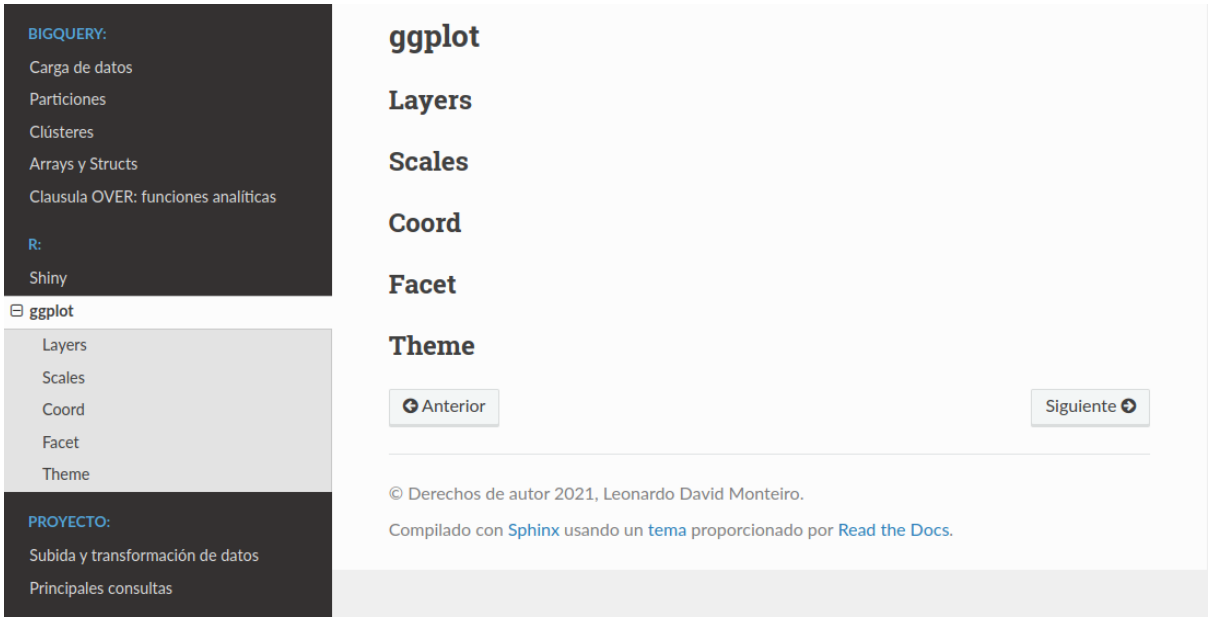


Figura A.7: Sección de ggplot de la documentación.

Figura A.8: Sección de subida y transformación de datos de la documentación.

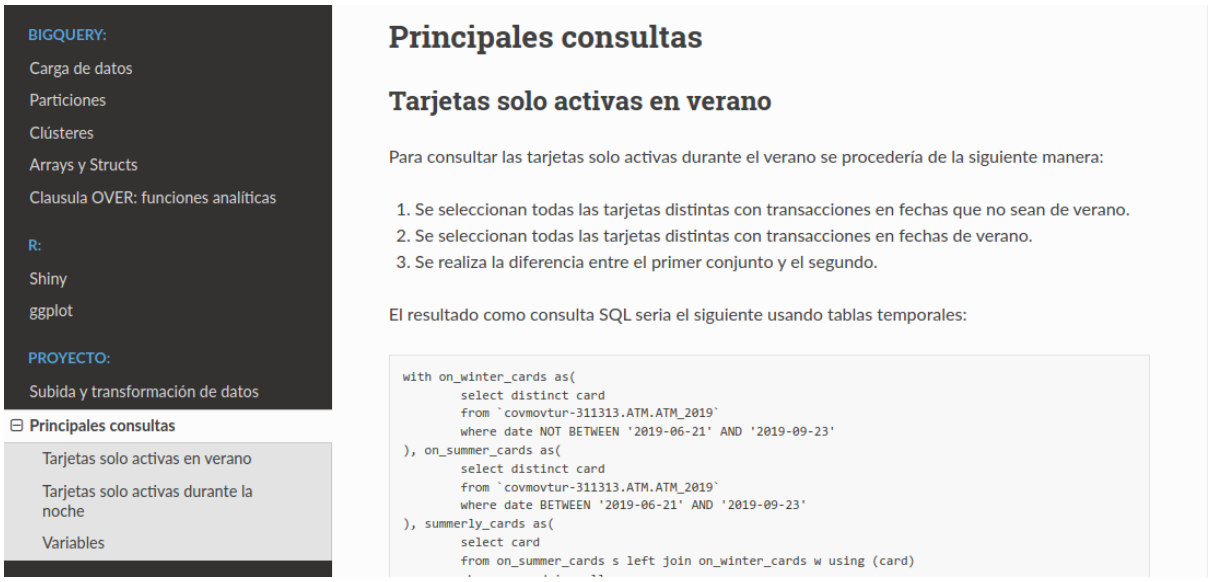


Figura A.9: Sección de principales consultas de la documentación.

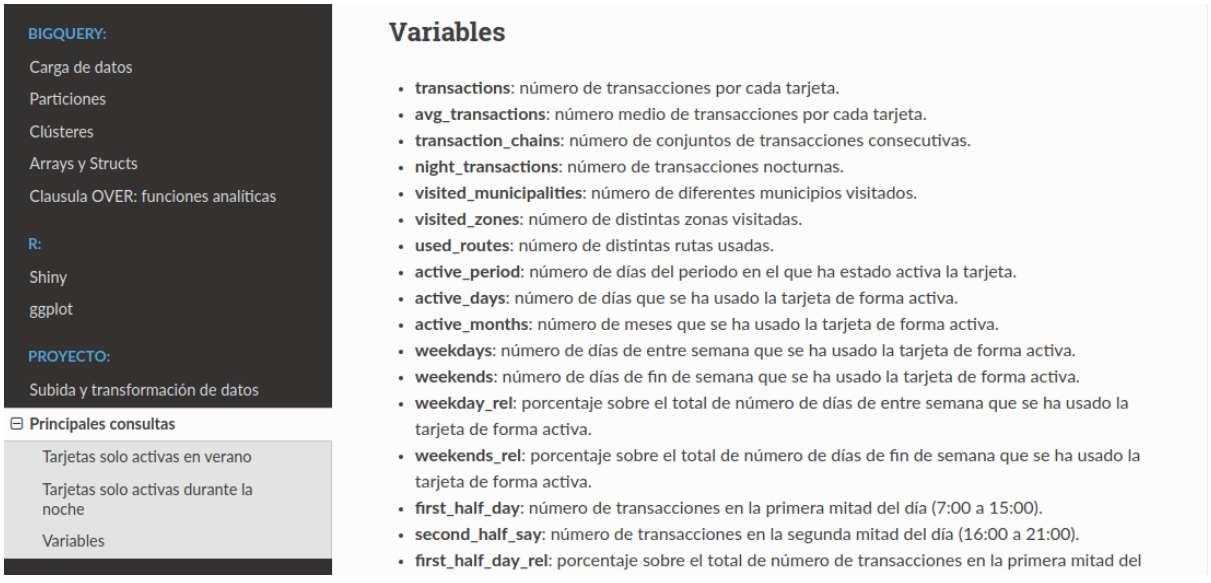


Figura A.10: Sección de principales consultas de la documentación.

Anexo B

Código usado durante el proyecto

B.1. Introducción

En este anexo se pretende presentar el código usado durante este proyecto. Se presentan las principales consultas SQL utilizadas durante el proyecto y las partes más importantes del código en R usado durante el análisis.

B.2. Consultas SQL

Las consultas usan el dialecto de Standard SQL de Google BigQuery.

B.2.1. Datos de transacciones anuales

Consultas para la limpieza y carga de datos de transacciones anuales.

```
CREATE TABLE IF NOT EXISTS ATM.ATM_2020 AS (
    SELECT targeta AS card, viatge AS transaction_id,
    CAST(SUBSTR(data_op,0, 8) AS DATE FORMAT "YYYYMMDD") AS `date`,
    CASE WHEN hora_op="  "
        THEN null
    ELSE CAST(hora_op AS INT64)
    END AS hour,
    CASE WHEN hora_op="  "
        THEN CAST(SUBSTR(data_op,0, 8)
            AS DATETIME FORMAT "YYYYMMDD")
    ELSE CAST(CONCAT(SUBSTR(data_op,0, 8), " ", hora_op)
        AS DATETIME FORMAT "YYYYMMDD HH24")
    END AS date_time,
    tipus_mov AS mov_type, operador AS agency_id, linia AS route_id,
    TRIM(titol) AS fare_id, zones AS zone_id,
    CASE WHEN municipi="     "
        THEN null
```

```
    ELSE CAST(municipi AS INT64)
    END AS municipality_id, comarca AS region_id,
    CASE WHEN operador IN (202, 203, 213, 256, 257)
        THEN 1
    ELSE 0
    END AS urban
FROM `covmovtur-311313.ATM.ATM_2020_RAW`
);
```

B.2.2. Datos agregados por parada mensual y anualmente

Consultas para la limpieza y carga de datos agregados.

```
CREATE TABLE IF NOT EXISTS ATM_AGG.2019_2020_Stops_Month AS (
  SELECT 'Any' AS Year,
  CASE Mes
     WHEN "Gener" THEN "January"
     WHEN "Febrer" THEN "February"
     WHEN "Marc" THEN "March"
     WHEN "Abril" THEN "April"
     WHEN "Maig" THEN "May"
     WHEN "Juny" THEN "June"
     WHEN "Juliol" THEN "July"
     WHEN "Agost" THEN "August"
     WHEN "Setembre" THEN "September"
     WHEN "Setembre" THEN "September"
     WHEN "Octubre" THEN "October"
     WHEN "Novembre" THEN "November"
     WHEN "Desembre" THEN "December"
     END AS Month,
  ATM_AGG.arrayindexof(["January", "February", "March","April","May","June","
    July","August","September","October", "November", "December"], Month) AS
    Month_numeric, Parada AS Stop,
  CASE WHEN Coord_X="-" THEN NULL
     WHEN Coord_X="0" THEN NULL
     ELSE CAST(Coord_X AS INT64)
     END AS x,
  CASE WHEN Coord_Y="-" THEN NULL
     WHEN Coord_Y="0" THEN NULL
     ELSE CAST(Coord_Y AS INT64)
     END AS y,
  CASE WHEN Tipus_parda="-" THEN NULL
     ELSE Tipus_parada
     END AS Stop_type,
  CASE WHEN Municipi="-" THEN NULL
     ELSE Municipi
     END AS Municipality,
  'Validacions Titols Propis' AS Own_tickets_validations, T10, 'T10/30' AS
    T10_30, 'T50/30' AS T50_30, 'T70/90' AS T70_90, 'T-12' AS T_12, TMES
 FROM `covmovtur-311313.ATM_AGG.2019_2020_Stops_Month_RAW`
 );
```

B.2.3. Extracción de la muestra a analizar

```sql
with on_winter_cards as(
  select distinct card
  from `covmovtur-311313.ATM.ATM_2019`
  where date NOT BETWEEN '2019-06-21' AND '2019-09-23'
), on_summer_cards as(
  select distinct card
  from `covmovtur-311313.ATM.ATM_2019`
  where date BETWEEN '2019-06-21' AND '2019-09-23'
), summerly_cards as(
  select card
  from on_summer_cards s left join on_winter_cards w using (card)
  where w.card is null
)

SELECT * FROM `covmovtur-311313.ATM.ATM_2019`
WHERE card IN (SELECT * FROM `summerly_cards`)
```

B.2.4. Imputación de datos

Consultas para la imputación de datos según lo explicado en Sección 3.4.3.

```sql
UPDATE `covmovtur-311313.ATM.ATM_2020` AS c SET hour=22
WHERE fare_id="T-10" AND hour IS null AND transaction_id >
  (SELECT MAX(transaction_id)
  FROM `covmovtur-311313.ATM.ATM_2020`
  WHERE hour IS NOT null AND card=c.card AND date=c.date);
UPDATE `covmovtur-311313.ATM.ATM_2020` AS c SET hour=6
WHERE fare_id="T-10" AND hour IS null AND transaction_id <
  (SELECT MIN(transaction_id)
  FROM `covmovtur-311313.ATM.ATM_2020`
  WHERE hour IS NOT null AND card=c.card AND date=c.date);
UPDATE `covmovtur-311313.ATM.ATM_2020` AS c SET hour=22
WHERE fare_id="T-10" AND hour IS null AND transaction_id IN
  (SELECT transaction_id
  FROM `covmovtur-311313.ATM.ATM_2020`
  WHERE hour=21 and card=c.card and date=c.date);
UPDATE `covmovtur-311313.ATM.ATM_2020` AS c SET hour=6
WHERE fare_id="T-10" AND hour IS null AND transaction_id IN
  (SELECT transaction_id
  FROM `covmovtur-311313.ATM.ATM_2020`
  WHERE hour=7 AND card=c.card AND date=c.date);
```

B.2.5. Extracción de variables características

El objetivo del siguiente código es extraer las diferentes variables características que describen los datos. Para ello se procede de la siguiente manera:

- Se procede a definir diferentes vistas, que ayudaran a simplificar y realizar el trabajo paso a paso:
  - Un conjunto de vistas para las variables de transacciones grupales: `groups_sizes`, `group_variables` y `number_groups`.

- Una vista para aquellas variables que se pueden conseguir mediante agregación: `summarized`.
- Un conjunto de vistas para las variables que realizan un ranking de municipios o paradas: `card_stops_stats`, `card_stops_ranking` y `card_stops_podium`.

- Se realiza una consulta final en la cual se juntan las diferentes variables calculadas en las vistas anteriores.

```
with groups_sizes as(
select DISTINCT card, fare_id, `date`, route_id, municipality_id,
    COUNT(transaction_id) OVER (PARTITION BY date_time, card, fare_id,
    route_id, municipality_id ORDER BY hour RANGE BETWEEN 0 PRECEDING
    AND 1 FOLLOWING) as group_size,
from `covmovtur-311313.ATM.ATM_2020_summeronly_interurban`
where hour IS NOT NULL AND mov_type=0
), group_variables as(
    select card, MAX(group_size) as max_group_size, AVG(group_size) as
    avg_group_size, MIN(group_size) as min_group_size
    from groups_sizes
    group by card
),number_groups as(
    select card, SUM(group_size) as group_transactions
    from groups_sizes
    where group_size>1
    group by card
), summarized as(
    select card,
    count(transaction_id) as transactions,
    count(transaction_id)/count(distinct FORMAT_DATETIME("%d-%m-%Y", `date`))
    as avg_transactions,
    countif(mov_type=1) as transaction_chains,
    countif(hour is null) as night_transactions,
    count(distinct municipality_id) as visited_municipalities,
    count(distinct zone_id) as visited_zones,
    count(distinct route_id) as used_routes,
    DATE_DIFF(MAX(`date`),MIN(`date`), DAY) + 1 as active_period,
    count(distinct FORMAT_DATETIME("%d-%m-%Y", `date`)) as active_days,
    count(distinct FORMAT_DATETIME("%m-%Y", `date`)) as active_months,
    countif(EXTRACT(DAYOFWEEK FROM `date`) NOT IN (1, 7)) as weekdays,
    countif(EXTRACT(DAYOFWEEK FROM `date`) IN (1, 7)) as weekends,
    countif(EXTRACT(DAYOFWEEK FROM `date`) NOT IN (1, 7)) / count(
    transaction_id) as weekdays_rel,
    countif(EXTRACT(DAYOFWEEK FROM `date`) IN (1, 7)) / count(transaction_id)
    as weekends_rel,
    countif(hour IN (7,8,9,10,11,12,13,14,15)) as first_half_day,
    countif(hour IN (16,17,18,19,20,21)) as second_half_day,
    countif(hour IN (7,8,9,10,11,12,13,14,15))/count(transaction_id) as
    first_half_day_rel,
    countif(hour IN (16,17,18,19,20,21))/count(transaction_id) as
    second_half_day_rel,
    countif(hour IN (7,8,9,10,11)) as time_morning,
    countif(hour IN (12,13,14,15,16)) as time_midday,
    countif(hour IN (17,18,19,20,21)) as time_afternoon,
    countif(hour IN (6,22) OR hour is null) as time_night,
    countif(hour IN (7,8,9,10,11))/count(transaction_id) as time_morning_rel,
    countif(hour IN (12,13,14,15,16))/count(transaction_id) as time_midday_rel,
    countif(hour IN (17,18,19,20,21))/count(transaction_id) as
    time_afternoon_rel,
```

```
    countif(hour IN (6,22) OR hour is null)/count(transaction_id) as
    time_night_rel,
    countif(municipality_id IN (43171, 43905,43038)) as transactions_cgc,
    countif(municipality_id IN (43123, 43148)) as transactions_tarragona_reus,
    countif(municipality_id IN (43171, 43905,43038))/count(transaction_id) as
    transactions_cgc_rel,
    countif(municipality_id IN (43123, 43148))/count(transaction_id) as
    transactions_tarragona_reus_rel,
    countif(municipality_id IN (43123, 43148, 43171, 43905, 43038, 43161,
    43092, 43153, 43037, 43051, 43163)) as transactions_urban_municipalities,
    from `covmovtur-311313.ATM.ATM_2020_summeronly_interurban`
    group by card
), card_stop_stats as (
    select card, municipality_id, count(transaction_id) as
    municipality_transactions
    from `covmovtur-311313.ATM.ATM_2020_summeronly_interurban`
    where municipality_id IS NOT NULL
    group by 1, 2
    order by municipality_transactions DESC
), card_stops_ranking as (
    select card, ARRAY_AGG(municipality_id) as municipality_id_array,
    ARRAY_AGG(municipality_transactions) as municipality_transactions_array
    from card_stop_stats
    group by card
), card_stops_podium as (
    select card,
    case when ARRAY_LENGTH(municipality_id_array) = 0
        then 0
    else municipality_transactions_array[offset(0)]
    end as main_municipalitiy,
    case when ARRAY_LENGTH(municipality_id_array) = 0
        then 0
        when ARRAY_LENGTH(municipality_id_array) = 1
        then municipality_transactions_array[offset(0)]
    else municipality_transactions_array[offset(0)]
        + municipality_transactions_array[offset(1)]
    end as main_two_municipalities,
    case when ARRAY_LENGTH(municipality_id_array) = 0
        then 0
        when ARRAY_LENGTH(municipality_id_array) = 1
        then municipality_transactions_array[offset(0)]
        when ARRAY_LENGTH(municipality_id_array) = 2
        then municipality_transactions_array[offset(0)]
            + municipality_transactions_array[offset(1)]
    else municipality_transactions_array[offset(0)]
        + municipality_transactions_array[offset (1)]
        + municipality_transactions_array[offset(2)]
    end as main_three_municipalities
    from card_stops_ranking
)

select *
from summarized left join group_variables using(card)
    left join number_groups using(card) left join card_stops_podium using(card)
order by transactions desc;
```

B.3. Código en R

B.3.1. Selección de variables características

Este código en R dibuja las matrices de correlación para poder seleccionar aquellas variables características relevantes.

```
######################
# FEATURE SELECTION #
######################
# calculate correlation matrices
activity_variable_correlation <-
  round(cor(activity_card_stats[2:length(activity_card_stats)]), 3)

time_variable_correlation <-
  round(cor(time_card_stats[2:length(time_card_stats)]), 3)

spatial_variable_correlation <-
  round(cor(spatial_card_stats[2:length(spatial_card_stats)]), 3)

p.mat_activity <- cor_pmat(activity_card_stats[2:length(activity_card_stats)])
p.mat_time <- cor_pmat(time_card_stats[2:length(time_card_stats)])
p.mat_spatial <- cor_pmat(spatial_card_stats[2:length(spatial_card_stats)])


# Plot Correlograms
activity_corrplot<-ggcorrplot(activity_variable_correlation, hc.order = TRUE,
    lab = TRUE, p.mat = p.mat_activity, sig.level = 0.001)

time_corrplot<-ggcorrplot(time_variable_correlation, hc.order = TRUE, lab =
    TRUE, p.mat = p.mat_time, sig.level = 0.001)

spatial_corrplot<-ggcorrplot(spatial_variable_correlation, hc.order = TRUE, lab
     = TRUE)

corrplot_prow <-
  plot_grid(activity_corrplot + theme(legend.position="none"),
            time_corrplot + theme(legend.position="none"),
            spatial_corrplot + theme(legend.position="none"),
            labels = c("Activity","Time", "Spatial"), ncol = 3, rel_widths = c
    (.85, .85, 1))

# extract a legend that is laid out horizontally
corrplot_legend <-
  get_legend(
    activity_corrplot +
      guides(color = guide_legend(nrow = 1)) +
      theme(legend.position = "top")
  )


corrplot <-
  plot_grid(corrplot_prow,
            ncol = 1, rel_widths = c(0.6, 1), rel_heights = c(.1, 1)) +
  theme(legend.key.size = grid::unit(10, "lines"))

ggsave(filename = "corrplot.png",
       plot = corrplot,
```

```
        height=30, width=40, units='cm',
        dpi = 300)
```

B.3.2. Análisis de perfil latente

Este código en R usa la biblioteca `tidyLPA` para realizar el análisis de perfil latente. Además, permite la comparación de diferentes criterios que permiten realizar la elección del número de componentes del modelo.

```
#######
# LPA #
#######

#Activity cluster (parallel code)
activity_lpa_profiles_full <- mclapply(2:5, function(n) estimate_profiles(
    activity_card_stats[2:length(activity_card_stats)],n_profiles = n, models =
    1), mc.cores = detectCores())

activity_lpa_profiles_full <- simplify2array(activity_lpa_profiles_full)

class(activity_lpa_profiles_full) <- c("tidyLPA","list")

activity_lpa_profiles_comparison <-
  activity_lpa_profiles_full %>%
  compare_solutions(statistics = c("AIC", "AWE", "BIC", "CLC", "KIC"))

#Time cluster (parallel code)
time_lpa_profiles_full <- mclapply(2:7, function(n) estimate_profiles(time_card
    _stats[2:length(time_card_stats)],n_profiles = n, models = 1), mc.cores =
    detectCores())

time_lpa_profiles_full <- simplify2array(time_lpa_profiles_full)

class(time_lpa_profiles_full) <- c("tidyLPA","list")

time_lpa_profiles_comparison <-
  time_lpa_profiles_full %>%
  compare_solutions(statistics = c("AIC", "AWE", "BIC", "CLC", "KIC"))

#Spatial cluster (parallel code)
spatial_lpa_profiles_full <- mclapply(2:7, function(n) estimate_profiles(
    spatial_card_stats[2:length(spatial_card_stats)],n_profiles = n, models = 1)
    , mc.cores = detectCores())

spatial_lpa_profiles_full <- simplify2array(spatial_lpa_profiles_full)

class(spatial_lpa_profiles_full) <- c("tidyLPA","list")

spatial_lpa_profiles_comparison <-
  spatial_lpa_profiles_full %>%
  compare_solutions(statistics = c("AIC", "AWE", "BIC", "CLC", "KIC"))

#Data frame summary of clusters comparison
as.data.frame(activity_lpa_profiles_comparison$fits[c(2,4,6,12,13,14)]) %>%
    kable() %>% save_kable("activity_profiles_table.html")
as.data.frame(time_lpa_profiles_comparison$fits[c(2,4,6,12,13,14)]) %>% kable()
     %>% save_kable("time_profiles_table.html")
```

```
as.data.frame(spatial_lpa_profiles_comparison$fits[c(2,4,6,12,13,14)]) %>% kable() %>% save_kable("spatial_profiles_table.html")

#Select cluster model
time_lpa_profiles <- time_lpa_profiles_full$model_1_class_5
spatial_lpa_profiles <-spatial_lpa_profiles_full$model_1_class_5
activity_lpa_profiles <- activity_lpa_profiles_full$model_1_class_5
```

B.3.3. Interpretación del modelo

Este código en R usa la biblioteca ggplot2 para dibujar las gráficas que representan los diferentes clústeres. Como se puede observar bastante código está comentado, este es para dibujar gráficos en los cuales existan variables con diferentes rangos de valores.

```
############
# Plot LPA #
############

# Format variable and legend labels before for plotting
activity_variable_labels <-
  colnames(activity_card_stats[2:length(activity_card_stats)]) %>%
  str_replace_all(pattern = "_",replacement = "\n")

time_variable_labels <-
  colnames(time_card_stats[2:length(time_card_stats)]) %>%
  str_replace_all(pattern = "_",replacement = "\n")

spatial_variable_labels <-
  colnames(spatial_card_stats[2:length(spatial_card_stats)]) %>%
  str_replace_all(pattern = "_",replacement = "\n")

activity_legend_labels<-
  paste0("A",1:5,"\n",
         c("Continued", "Groups", "Short term","Long term","Sporadic"),
         vapply(1:5,
                FUN = function(x) paste0("\n(n=",format(sum(activity_lpa_profiles$dff$Class==x),big.mark = ','),")"),
                FUN.VALUE = "string")
  )

time_legend_labels<-
  paste0("T",1:5,"\n",
         c("","","","","","",""),
         vapply(1:5,
                FUN = function(x) paste0("\n(n=",format(sum(time_lpa_profiles$dff$Class==x),big.mark = ','),")"),
                FUN.VALUE = "string")
  )

spatial_legend_labels<-
  paste0("S",1:5,"\n",
         c("Coastal","City","Sprawled","Concentrated\ncoastal","Concentrated\ncity"),
         vapply(1:5,
                FUN = function(x) paste0("\n(n=",format(sum(spatial_lpa_profiles$dff$Class==x),big.mark = ','),")"),
                FUN.VALUE = "string")
```

```
  )

activity_legend_labels_unnumbered <-
  paste0("A",1:5,"\n",
         c("Continued", "Groups", "Short term","Long term","Sporadic")
  )

time_legend_labels_unnumbered <-
  paste0("T",1:5,"\n",
         c("","","","","")
  )

spatial_legend_labels_unnumbered <-
  paste0("S",1:5,"\n",
         c("Coastal","City","Sprawled","Concentrated\ncoastal","Concentrated\
    ncity")
  )


# # Plot LPAs
activity_lpa_plot <-
  plot_profiles(activity_lpa_profiles,rawdata = FALSE) +
  scale_shape_discrete(name = "Class (cards)", labels=activity_legend_labels) +
  scale_color_discrete(name = "Class (cards)", labels=activity_legend_labels) +
  scale_linetype_discrete(name = "Class (cards)", labels=activity_legend_labels
    ) +
  theme_bw() +
  theme(legend.position = "top")
# +  scale_x_discrete(labels = activity_variable_labels)
# activity_lpa_plot_a <-
#   plot_profiles(activity_lpa_profiles,rawdata = FALSE,
#                 variables=colnames(activity_card_stats[,c(4,5)])) +
#   scale_shape_discrete(name = "Class (cards)", labels=activity_legend_labels)
    +
#   scale_color_discrete(name = "Class (cards)", labels=activity_legend_labels)
    +
#   scale_linetype_discrete(name = "Class (cards)", labels=activity_legend_
    labels)
# # +  scale_x_discrete(labels = spatial_variable_labels)
#
#
# activity_lpa_plot_b <-
#   plot_profiles(activity_lpa_profiles,rawdata = FALSE,
#                 variables=colnames(activity_card_stats[,c(2,3,6,7,8)])) +
#   scale_shape_discrete(name = "Class (cards)", labels=activity_legend_labels)
    +
#   scale_color_discrete(name = "Class (cards)", labels=activity_legend_labels)
    +
#   scale_linetype_discrete(name = "Class (cards)", labels=activity_legend_
    labels)
# # +  scale_x_discrete(labels = spatial_variable_labels)

# activity_lpa_prow <-
#   plot_grid(activity_lpa_plot_a + theme(legend.position="none"),
#             activity_lpa_plot_b + theme(legend.position="none"),
#             labels = c("A","B"), ncol = 2)
#
# activity_lpa_plot_legend <-
#   get_legend(
#     activity_lpa_plot +
```

```
#       guides(color = guide_legend(nrow = 1)) +
#       theme(legend.position = "top")
#   )
#
# # add the legend underneath the row we made earlier. Give it 10%
# # of the height of one plot (via rel_heights).
# activity_lpa_plot <-
#   plot_grid(activity_lpa_plot_legend, activity_lpa_plot,
#             ncol = 1)

ggsave(filename = "activity_lpa_plot.png",
       plot = activity_lpa_plot,
       height=20, width=32, units='cm',
       dpi = 300)


time_lpa_plot <-
  plot_profiles(time_lpa_profiles,rawdata = FALSE) +
  scale_shape_discrete(name = "Class (cards)", labels=time_legend_labels) +
  scale_color_discrete(name = "Class (cards)", labels=time_legend_labels) +
  scale_linetype_discrete(name = "Class (cards)", labels=time_legend_labels) +
  theme_bw() +
  theme(legend.position = "top")
# +  scale_x_discrete(labels = activity_variable_labels)

ggsave(filename = "time_lpa_plot.png",
       plot =time_lpa_plot,
       height=20, width=32, units='cm',
       dpi = 300)


# spatial_lpa_plot_a <-
#   plot_profiles(spatial_lpa_profiles,rawdata = FALSE,
#                 variables=colnames(spatial_card_stats[,c(2,3,4)])) +
#   scale_shape_discrete(name = "Class\n(cards)", labels=spatial_legend_labels)
    +
#   scale_color_discrete(name = "Class\n(cards)", labels=spatial_legend_labels)
    +
#   scale_linetype_discrete(name = "Class\n(cards)", labels=spatial_legend_
    labels)
# # +  scale_x_discrete(labels = spatial_variable_labels)
#
#
# spatial_lpa_plot_b <-
#   plot_profiles(spatial_lpa_profiles,rawdata = FALSE,
#                 variables=colnames(spatial_card_stats[,c(5)])) +
#   scale_shape_discrete(labels=spatial_legend_labels) +
#   scale_color_discrete(labels=spatial_legend_labels) +
#   scale_linetype_discrete(labels=spatial_legend_labels)
#
# spatial_lpa_prow <-
#   plot_grid(spatial_lpa_plot_a + theme(legend.position="none"),
#             spatial_lpa_plot_b + theme(legend.position="none"),
#             labels = c("A","B"), ncol = 2)
#
# # extract a legend that is laid out horizontally
# spatial_lpa_plot_legend <-
#   get_legend(
#     spatial_lpa_plot_a +
#       guides(color = guide_legend(nrow = 1)) +
```

```
#       theme(legend.position = "top")
#   )
#
# # add the legend underneath the row we made earlier. Give it 10%
# # of the height of one plot (via rel_heights).
# spatial_lpa_plot <-
#   plot_grid(spatial_lpa_plot_legend, spatial_lpa_prow,
#             ncol = 1, rel_heights = c(.2, 1))

spatial_lpa_plot <-
  plot_profiles(spatial_lpa_profiles,rawdata = FALSE) +
  scale_shape_discrete(name = "Class (cards)", labels=spatial_legend_labels) +
  scale_color_discrete(name = "Class (cards)", labels=spatial_legend_labels) +
  scale_linetype_discrete(name = "Class (cards)", labels=spatial_legend_labels)
    +
  theme_bw() +
  theme(legend.position = "top")

ggsave(filename = "spatial_lpa_plot.png",
       plot = spatial_lpa_plot,
       height=20, width=32, units='cm',
       dpi = 300)
```