



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Análisis de datos de movilidad en la ciudad de València.

Trabajo Fin de Máster

Máster Universitario en Gestión de la Información

Autor: Victoria Bulavina
Tutor: César Ferri Ramírez
2018/2019

“Hey, Timon, ever wonder what those sparkly dots are up there?” Pumba, The Lion King

Resumen

El presente trabajo tiene como objetivo analizar los datos del tráfico de la ciudad de Valencia para mostrar la congestión vial de los coches y las bicicletas durante varias temporadas del año, ver las zonas más ocupadas de la ciudad de Valencia y horas punta. Para ello se estudiarán los datos abiertos del Ayuntamiento de Valencia. La parte práctica ha consistido en la creación de una aplicación para analizar los datos, mostrándolos en un mapa interactivo, programado y utilizado por R Shiny, para dar los consejos a los usuarios de vías de un mejor uso del transporte.

Palabras clave: data análisis, tráfico Valencia, coches, bicicletas, R Shiny.

Abstract

This final thesis has as a target analyze Open Data of the traffic of Valencia city to show the road congestion with cars and bicycles during different times of the year, show the most occupied parts of the city and rush hours. To accomplish the target were studied the Open Data from Town Hall of Valencia. The practical part consisted creation of an application for data analysis, showed on an interactive map, created and programmed in R Shiny to give advices for the better usage of the transport.

Keywords: data analysis, Valencia traffic, cars, bicycles, R Shiny.

Tabla de contenidos

1. Introducción	11
2. Data análisis y su uso en vida diaria	13
Análisis de datos	13
3. Herramientas de data análisis	14
4. R en data análisis	17
5. Estado del tráfico en Comunidad Valenciana	18
6. Proceso de creación de la aplicación	20
Herramientas usadas	20
Origen de datos	20
Introducción a Shiny	21
Introducción a Leaflet	21
Paso 1: Crear la aplicación básica Shiny	21
Paso 2: Cargar los datos y prepararlos para el mapeo	22
Paso 3: Crear el mapa básico	26
Paso 4: Mapear los datos con el Leaflet	28
7. Análisis de resultados	32
Análisis de los registros	32
8. Solución de problemas	37
9. Conclusión	39
10. Bibliografía	41

Tabla de ilustraciones

Ilustración 1. R	14
Ilustración 2. Tableau	14
Ilustración 3. Python.....	15
Ilustración 4. SAS.....	15
Ilustración 5. Excel	15
Ilustración 6. Agencia municipal de la bicicleta.....	19
Ilustración 7. Agencia municipal de la bicicleta.....	19
Ilustración 9. Datos originales	21
Ilustración 10. Creación de la aplicación.....	22
Ilustración 11. Librerías.....	22
Ilustración 12. Python unión.....	23
Ilustración 13. Cmd ejecución	23
Ilustración 14. Limpieza de datos de los coches	24
Ilustración 15. Conversión de coordenadas	25
Ilustración 16. Guardar .csv.....	25
Ilustración 17. Datos finales	26
Ilustración 18. setView code	27
Ilustración 19. Output básico.....	28
Ilustración 20. ui. R	29
Ilustración 21. server.R.....	29
Ilustración 22. global.R	30
Ilustración 23. process.R	30
Ilustración 24. Aplicación final	31
Ilustración 25. 5.04 bicicletas	33
Ilustración 26. 5.04 coches	33
Ilustración 27. 18.04 bicicletas	34
Ilustración 28. 18.04 coches	34
Ilustración 29. 20.04 bicicletas	35
Ilustración 30. 20.04 coches	35



1. Introducción

En los últimos años, aproximadamente a partir del año 2009, la aplicación de técnicas relacionadas con Big Data han revolucionado el mundo. Las empresas empezaron hacer los análisis de datos que tenían para mejorar sus negocios, para lanzar nuevos productos y mucho más.

Todos nosotros perfectamente entendemos, que hoy en día no se puede imaginar ninguna empresa, ningún negocio, ni estudio sin datos, ni sin análisis de los mismos. El análisis de datos nos ayuda a mejorar todos los ámbitos de la vida. Cada conjunto de datos distinto es específico para un propósito: mejorar el rendimiento, predecir, estudiar el objeto y otros. Cada conjunto de datos es procesado y analizado. Dependiendo del tipo, cantidad y propósito de la adquisición de datos, se elige un método de procesamiento y análisis.

El problema del análisis de los flujos de tráfico de la ciudad es relevante por varias razones:

1. El aumento en el número de automóviles, que afecta la congestión del tráfico.
2. Un aumento en el número de accidentes y la posibilidad de que ocurran.
3. Costos excesivos de dinero para la reconstrucción o construcción de secciones adicionales de carreteras tanto para automóviles como para bicicletas.
4. Optimización de flujos de tráfico.

Específicamente uno los objetivos de este trabajo era analizar datos del tráfico de la ciudad de Valencia para mostrar a los usuarios de las vías, como, cuando y de que manera hacer una mejor utilización de las vías. Para cumplir el objetivo he sacado dos tipos de datos a tiempo real del Ayuntamiento de Valencia: los datos del tráfico de los coches y los datos del tráfico de las bicicletas. Este documento aborda el problema de optimizar el flujo de tráfico de la ciudad usando R Shiny mostrando resultados en el mapa interactivo de la ciudad de Valencia.

En el trabajo de optimización de flujos de tráfico, se tendrán en cuenta los datos con los siguientes parámetros: geolocalización, intensidad, tiempo, fechas y fines de semana. Como resultado de este análisis obtendremos recomendaciones reales para mejorar el flujo del tráfico de la ciudad.

Basando en lo que está expuesto, la estructura de este trabajo se compone de seis capítulos:

El segundo capítulo tiene una pequeña introducción a data análisis, su impacto y uso en la vida diaria y los métodos más utilizados en data análisis.

El tercer capítulo habla sobre las herramientas de data análisis.

El cuarto capítulo presenta una breve explicación de cómo se usa R en ciencia de datos y como simplifica el manejo de datos masivos para data análisis.



El quinto capítulo muestra el estado del tráfico de la ciudad de Valencia: qué tendencias tiene y los problemas que existen actualmente.

El sexto capítulo contiene el proceso de creación de la aplicación: los códigos ejecutados, formato de datos usados y ejemplos de muestra.

El séptimo capítulo presenta el análisis de los resultados obtenidos.

El octavo capítulo cuenta los problemas, que he tenido durante el proceso de creación de la aplicación y como los he resuelto.

Última parte son conclusiones y resumen de los resultados obtenidos, analizando los objetivos, descritos en la introducción.

2. Data análisis y su uso en vida diaria

"Data Analytics" es una herramienta importante para obtener inteligencia empresarial y proporcionar respuestas individuales a los clientes. A veces abreviado como "Analytics", se está volviendo cada vez más importante para organizaciones de todos los tamaños. La práctica de "Data Analytics" ha evolucionado y se ha expandido gradualmente con el tiempo, proporcionando muchos beneficios, como el análisis de pronósticos para los sectores financiero o inmobiliario. Por lo tanto, es una herramienta en la que podemos confiar para mejorar nuestros resultados.

El campo de análisis de datos está conectado con nuestro mundo e influye en el de una manera sin precedentes. Millones, incluso miles de millones de computadoras, teléfonos inteligentes y tabletas están recopilando datos en estos momentos, aprender a analizar estos datos es una forma de aprender a mejorar nuestras respuestas.

Análisis de datos

¿Qué es el análisis de datos? Prácticamente, es una comparación, de un mínimo, dos objetos de datos. Podría ser fácilmente dos proyectos en el trabajo, dos escuelas, dos recetas, dos vehículos o dos vacaciones, realmente cualquier cosa que nos interese. Por otro lado, puede comparar el mismo caso, pero teniendo en cuenta sus datos anteriores, es decir, datos históricos.

Aquí hay un ejemplo de cómo el análisis de datos impacta en nuestra vida diaria. Hay una historia de un estadista que decidió verificar la dosis del medicamento para su tía, que tenía diabetes. ¿Qué pasó? Él comenzó a recopilar la información de su glucómetro y realizó un seguimiento de sus niveles de glucosa, que ya estaba midiendo.

Después de recopilar suficientes datos para analizar, comparó los resultados con los valores aceptables, y parecía alto. El punto principal, desde la perspectiva del análisis de datos, es que recopiló los datos y los comparó para ver qué significaba. Con su nueva información, el estudiante y su tía entraron en el cambio en la medicación de la tía.

En este ejemplo, vemos claramente cómo el análisis de los datos y el resultado de su análisis podrían afectar la vida de cada ser humano. Pero no es el único ejemplo. En mi trabajo, veremos cómo afecta el análisis estadístico a la situación del tráfico en Valencia. Quién sabe, tal vez tendrá el mismo impacto en los ciudadanos, como un ejemplo de medición médica.



3. Herramientas de data análisis

Hoy en día tenemos miles de herramientas que nos ayudan a hacer un análisis de datos. Entre ellas podemos encontrar herramientas gratuitas o versiones de pago. Pero ¿cómo podemos encontrar la mejor, si hay miles de ellos? ¡Sencillo! ¡Solo búscalo en Google! Después de poner en la barra de búsqueda simples "herramientas de análisis de datos para la investigación", no pensé mucho en hacer clic en el primer enlace para ver la información. El sitio web prometió hablar sobre las 10 mejores herramientas de análisis de datos (TOP 10 Data Analytics tools, n.d.). Echemos un vistazo a algunos de ellos:

1. R



Ilustración 1. R

Es un lenguaje de programación que se creó específicamente para los estadistas con el fin de realizar mejores trabajos relacionados con las estadísticas. R proporciona una amplia variedad de técnicas estadísticas y gráficas, y es altamente extensible. También es de código abierto, lo que lo hace más deseable para los usuarios.

2. Tableau



Ilustración 2. Tableau

Esta herramienta también está relacionada con la visualización de datos y el análisis de datos. Los productos de Tableau consultan bases de datos relacionales, cubos de procesamiento analítico en línea, bases de datos en la nube y hojas de cálculo para generar visualizaciones de datos de tipo gráfico. Los productos también pueden extraer, almacenar y recuperar datos de un motor de datos en memoria. Ofrece una versión de prueba gratuita con una pequeña cantidad de funciones incluidas, que podría ser suficiente para realizar un pequeño trabajo o análisis, pero si desea profundizar más, desafortunadamente tiene que pagar por el uso.

3. Python



Ilustración 3. Python

Python es otro lenguaje de programación. Entre R se utiliza para el análisis de datos y siempre muestra el alto rendimiento. Además de R también es de código abierto y fácil de aprender. Entre los usuarios del existen diferencias de opinión entre las ventajas del uso de R y Python, dividiéndolos en dos grupos.

4. SAS



Ilustración 4. SAS

SAS es un paquete de software estadístico desarrollado por SAS Institute para análisis avanzado, análisis multivariado, inteligencia empresarial, gestión de datos y análisis predictivo (SAS , n.d.).

5. Excel



Ilustración 5. Excel

Excel es una herramienta analítica básica, popular y ampliamente utilizada en casi todas las industrias. Todos estamos familiarizados con Excel. Hay una gran cantidad de trabajos y herramientas estadísticas que puede usar para el rendimiento. Es fácil de usar y útil en tareas tales como limpieza de datos, transformación de datos y conversión de datos.

¿Por qué elegí presentar estas herramientas aquí? Son herramientas muy potentes y de alto valor que se utilizan en todo el mundo. Además de eso, para completar mi tarea y obtener los resultados necesarios con las conclusiones, utilicé tres de estas herramientas, para ser exactos:

- R

Análisis de datos de movilidad en la ciudad de València.

- Python
- Excel.

4. R en data análisis

R es un lenguaje diseñado específicamente para el análisis estadístico y la reconfiguración de datos. Es muy fácil trabajar con R, no solo porque no es un lenguaje de programación complicado, sino también porque R es de código abierto, y no tendrá problemas para encontrar respuestas a sus dudas y preguntas que pueda encontrar durante el código de desarrollo. Además, absolutamente todos pueden crear un paquete para R, así como editar o inspirarse. R es fácil de usar, por lo que los estadistas que no tienen experiencia en programación usan R. Solo con un par de clicks puede ejecutar varias líneas de código y proporcionar una idea de cómo funciona. Esta sencillez le da a R ventaja, por lo que es una opción ideal para proyectos de ciencia de datos.

R es un procesador de datos – se utiliza en el proceso de limpieza y recopilación de datos. Esta es una parte muy importante y costosa en ciencia de datos. R tiene una extensa biblioteca de herramientas para trabajar y manipular bases de datos.



5. Estado del tráfico en Comunidad Valenciana

Si alguna vez os habéis fijado en flujo de coches en Valencia, habréis notado, que cada año la cantidad de los coches estaba aumentando. La gente preferiría vivir en los pueblos, fuera de la ciudad, por varias razones, como económicas, tranquilidad y comodidad. Pero para vivir fuera de la capital, la mayoría de las veces necesitas tener coche. Por esto al llegar algunas veces a Valencia por la mañana, habéis notado la cantidad de coches, que están entrando. Todos sabemos que debido al tráfico vamos a pasar por un atasco y que si no salimos con tiempo podemos llegar tarde al trabajo, universidad, podemos perder las reuniones importantes y etc. Muchas veces lo mismo pasa al volver a casa. Estas horas de entrada y salida se llaman horas puntas y tenemos las soluciones para evitar los atascos, pero las soluciones no siempre son razonables, entre ellas, por ejemplo, salir antes de casa o salir después de la hora punta, lo mismo para la salida. En la mayoría de casos, la gente suele quedarse más horas en trabajo, para evitar el atasco. Si cogemos un ejemplo real, según las noticias de 2017 la cantidad de vehículos ha crecido un 0,7%. (Roig, 2017) “Más vehículos matriculados en Valencia, en concreto 8.300 respecto a lo que ocurrió en 2017.” (Moreno, 2019)

Aquí claramente nos vemos, que la cantidad de los vehículos dentro de la ciudad crece cada vez más. Estoy segura, que estáis de acuerdo, que las soluciones no son las mejores del mundo, por esto el gobierno de Valencia empezó a construir los carriles bici. La construcción de los carriles bici no tiene solo un contexto. Por ejemplo, los usuarios de bicicletas no solo libran la mitad de los carriles para grandes vehículos, pero también aumentan la cantidad de la gente, que está practicando deporte y costumbres de vida sana y activa. Si vemos los datos reales (Ilustración 7), que nos podemos encontrar en la Plaza de toros en Valencia, se ve la cantidad de las bicis en la ciudad: anuales, diarias, el aumento desde el año 2017.



Il·lustració 7. Agencia municipal de la bicicleta



Il·lustració 6. Agencia municipal de la bicicleta

Si comparamos estos datos del día 31 de agosto con los datos de una semana antes resultó, que desde el día 22 de agosto (Ilustración 8) la cantidad de las bicicletas ha crecido en 35 410 bicicletas. Cada vez podemos ver el crecimiento de la cantidad de carriles bici fuera y dentro de la ciudad, que nos demuestra la existencia del problema de la alta cantidad de vehículos y demuestra una de las soluciones aplicadas actualmente.

6. Proceso de creación de la aplicación

En este capítulo me gustaría mostrar el proceso y la estructura de la aplicación. A continuación, encontrará explicaciones sobre el formato y el contenido de los datos. Los pasos que seguí para obtener datos claros y compatibles. Además, pongo los pasos que conducen a la creación de la aplicación y al resultado final con todas las explicaciones de código y comentarios sobre los detalles y el proceso de ejecución.

Herramientas usadas

Para completar mi tarea, utilicé tres herramientas de análisis de datos, mencionadas en el capítulo anterior:

- R
Para el trabajo principal y la creación de la aplicación.
- Python
Para el gran trabajo de unificación de datos.
- Excel
Para la pequeña transformación de datos y conversiones y para la prueba de datos.

Origen de datos

Dado que mi trabajo se basa en los datos abiertos de Valencia, sería una tontería no aprovechar una oportunidad y no probar el portal de datos abiertos (Valencia, n.d.), proporcionado por el Ayuntamiento de Valencia. Tienen una gran variedad de datos, para cualquier tipo de análisis o simplemente por curiosidad de sus usuarios. Los datos proporcionados están disponibles en diferentes formatos, como: .wfs, .csv, .json, .snp, .gml, .wms, .kml, .kmz y más.

En el menú desplegable de tráfico y transportes, elegí descargar datos de carril bici e intensidad de tráfico por tramos, ambos descargados en formato .csv. El marco de datos original se ve así:

	modified	angulo	intensidad	punto_medida	Y	X	uri
1	2019-04-30T23:56:00+02:00	177	111	1162	726162.4	4373904	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
2	2019-04-30T23:56:00+02:00	239	55	1163	725877.2	4373375	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
3	2019-04-30T23:56:00+02:00	160	42	1334	726183.6	4373697	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
4	2019-04-30T23:56:00+02:00	120	167	2261	726633.3	4370401	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
5	2019-04-30T23:56:00+02:00	161	42	3131	729111.7	4371359	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
6	2019-04-30T23:56:00+02:00	236	167	3134	729643.2	4370568	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
7	2019-04-30T23:56:00+02:00	160	126	4113	724414.4	4369743	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
8	2019-04-30T23:41:00+02:00	177	55	1162	726162.4	4373904	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
9	2019-04-30T23:41:00+02:00	239	0	1163	725877.2	4373375	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
10	2019-04-30T23:41:00+02:00	160	84	1334	726183.6	4373697	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
11	2019-05-01T00:11:00+02:00	177	55	1162	726162.4	4373904	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
12	2019-05-01T00:11:00+02:00	239	55	1163	725877.2	4373375	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
13	2019-05-01T00:11:00+02:00	160	84	1334	726183.6	4373697	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
14	2019-05-01T00:11:00+02:00	120	42	2261	726633.3	4370401	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
15	2019-05-01T00:11:00+02:00	161	126	3131	729111.7	4371359	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
16	2019-05-01T00:11:00+02:00	236	126	3134	729643.2	4370568	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...
17	2019-05-01T00:11:00+02:00	160	126	4113	724414.4	4369743	http://apigobiernoabiertod.valencia.es/apirtod/datos/int...

Ilustración 9. Datos originales

Aquí podemos ver varias columnas, como la fecha en que se modificaron los datos, el ángulo desde donde se capturaron, la intensidad del tráfico, las coordenadas UTM y el enlace que lo lleva a descargar el archivo mencionado uno por uno.

Introducción a Shiny

Shiny es el paquete de R código abierto, que ayuda a crear aplicaciones interactivas rápidamente usando la sintaxis de R. Es fácil de usar, tiene gran variedad de videotutoriales y guías. Para crear una pizarra de Shiny no hace falta conocer ni HTML, ni CSS, ni JavaScript. Por esto Shiny es muy útil para novatos, que quieren hacer alguna aplicación sin conocimientos profundos en la parte visual de desarrollo web. Solo con Shiny se puede hacer una aplicación bien diseñada con toda la información necesaria para la visualización de datos.

Introducción a Leaflet

Leaflet es una famosa open-source librería para crear mapas interactivos escritos en JavaScript. Este paquete de R hace que sea fácil de integrar y controlar mapas Leaflet en R. Además, Leaflet tiene integración con Shiny, que nos sirve para nuestra aplicación de mapa interactivo. La función `leaflet()` devuelve un widget de mapa de Leaflet, que almacena una lista de objetos que se pueden modificar o actualizar luego.

Paso 1: Crear la aplicación básica Shiny

Es posible estructurar su aplicación de dos maneras. Puede tener un archivo **app.R** que tenga todos sus componentes ui y la lógica del servidor, o puede crear tres archivos separados: **ui.R**, **server.R** y **global.R**. La forma más conveniente con RStudio es que no es necesario para crear cada archivo por sí mismo, sólo puede marcar la opción necesaria y empezar a crear su aplicación:



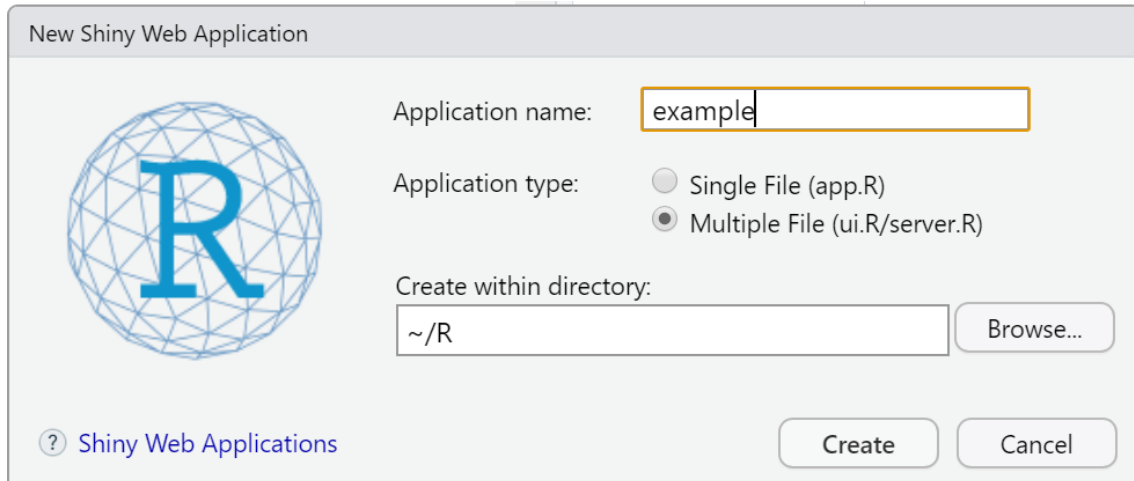


Ilustración 10. Creación de la aplicación

Elegí la ruta de tres archivos, solo porque me gusta que todo esté estructurado y organizado. Cuando la interfaz de usuario y el servidor están en diferentes archivos, hace que sea más fácil de depurar.

Entonces, creé un nuevo proyecto en RStudio, creé nuestros tres archivos separados y los guardé en nuestro directorio de proyectos. Aquí están los nombres de los archivos:

- ui.R
- server.R
- global.R

Normalmente, cuando se ejecuta la aplicación, global.R se ejecuta primero y carga todo en ese archivo al entorno de sesión global R. Realmente creo que es más fácil hacer todo lo que hay que hacer antes de que la aplicación se inicie en global.R. Entonces, en el archivo global.R, cargue las siguientes librerías requeridas:

```
library(shiny)
library(leaflet)
```

Ilustración 11. Librerías

En el archivo server.R, cree la función de mi servidor que se llamará una vez para cada sesión. Y en el fichero ui.R se crea una interfaz de usuario que tiene un panel de título y un diseño de barra lateral, que incluye un panel de barra lateral y un panel principal.

Paso 2: Cargar los datos y prepararlos para el mapeo

Después de completar el primer paso, avancé al segundo. El propósito del segundo paso era preparar conjuntos de datos y, si fuera necesario, realizar algunas limpiezas.

En primer lugar, cuando abrí la carpeta con los archivos descargados, noté que no se trata de un archivo que contenga todos los datos, sino de varios archivos, cada uno de los cuales tenía datos según la fecha. Luego me di cuenta de que tenía que unir todos los archivos en uno para poder compilar mi aplicación correctamente. Como tenía miles de archivos, era imposible unirlos manualmente ni en línea. Por ejemplo, en el caso de los datos de la bicicleta, solo tenía 3084 mini archivos para unir, pero para los automóviles, tenía 52532 mini archivos para unir. Pensé que la forma más fácil y rápida de lograrlo sería creando un pequeño programa en Python:

```
1  import os
2  import re
3  import csv
4
5  files = os.listdir()
6  res = []
7
8  for i in files:
9      with open(i, "r") as fileR:
10         res += fileR.readlines()[1:]
11
12  with open("result.csv", "w") as fileS:
13     fileS.writelines(res)
```

Ilustración 12. Python unión

¿Cómo funciona? Solo necesita colocar el archivo .py en la carpeta de los archivos que desea fusionar y ejecutar el comando `py` en la línea de comando:

```
set PYTHONPATH=%PYTHONPATH%;C:\Users\vbulavina\AppData\Local\Programs\Python\Python37
echo %PATH%
cacls c:\Users\vbulavina\Downloads\filesb\home\files\traffic_vlc /t /e /g Everyone:f
py scriptSum.py
```

Ilustración 13. Cmd ejecución

En primer lugar, creé una ruta para Python y configuré la ruta. Luego le di permiso a la carpeta y ejecuté un comando `py` para fusionar múltiples archivos. Me llevó solo una hora unir todos los archivos y recibir el resultado deseado. Entonces, finalmente obtuve dos archivos, uno para datos de automóviles, otro para bicicletas.



Después de cargar ambos archivos en el entorno R, me sorprendió la cantidad de observaciones para cada uno de los archivos. Por ejemplo, los datos de los automóviles tenían 63 971 331 observaciones en total, ya que para las bicicletas el archivo contenía solo 21 368 líneas. Eso me hizo pensar que debería limpiar conjuntos de datos y completar trabajos de combinar & unir en ellos. Aquí están las líneas de código que me ayudaron a realizar la tarea:

```
#cars data upload and cleaning
result <- read.csv("~/R/traffic/result.csv", header=FALSE)
result2 <- result[complete.cases(result),]
names(result2)[names(result2) == "v1"] <- "lat"
names(result2)[names(result2) == "v2"] <- "lon"
names(result2)[names(result2) == "v4"] <- "date"
names(result2)[names(result2) == "v7"] <- "intensity"
result2 <- result2[result2$date %in% c("2019-04-03", "2019-04-04", "2019-04-04", "2019-04-05", "2019-04-06",
result2$v3 <- NULL
result2$v5 <- NULL
result2$v6 <- NULL
```

Ilustración 14. Limpieza de datos de los coches

Después de cargar el archivo, mi primer paso fue deshacerme de los registros de NA en mi marco de datos. Para cumplir con esta tarea, ejecuté una función `complete.cases()` y obtuve 39 743 471 líneas de salida limpia. Luego, después de ejecutar la función `names()`, cambia el nombre de los nombres de columna para la futura comodidad. El siguiente paso fue filtrar los datos de acuerdo con las fechas necesarias. Como necesitaba datos de bicicletas para que coincidieran con los datos de los automóviles por fecha, extraje un conjunto de datos limpio para tener registros de las fechas necesarias, para ser exactos del 3.04.2019 al 6.05.2019. Y el último paso fue eliminar columnas innecesarias.

Después de completar todos los trabajos de combinación y combinación, pasé al siguiente paso. Como parecía ser, ni Shiny, ni Leaflet no son capaces de procesar con coordenadas UTM. Entonces, el primer punto fue convertir las coordenadas UTM en geográficas. Para completar esta misión utilicé la ayuda de dos paquetes R:

- Paquete `ggmap`: una colección de funciones para visualizar datos espaciales y modelos sobre mapas estáticos de varias fuentes en línea (por ejemplo, *Google Maps* y *Stamen Maps*). Incluye herramientas comunes a esas tareas, incluidas las funciones de geolocalización y enrutamiento. (Kahle, 2019)
- Paquete `rgdal`: proporciona enlaces a la biblioteca de abstracción de datos 'geoespaciales' ('GDAL') y acceso a operaciones de proyección/transformación. (Bivand, 2019)

Después de descargar los paquetes mencionados, pasé a escribir el código. Entonces el resultado final de la transformación fue el siguiente:


```

library(ggmap)
library(rgdal)

datos<-read.csv("C:/Users/vbulavina/Downloads/filesb/home/files/traffic_vlc/MyData.csv",sep=";")
### conversion de coordenadas UTM a geograficas

lat<-c()
lon<-c()
for (i in 1:length(another$X))
#for (i in 1:10)
{
pt<-list(lon=another$Y[i],lat=another$X[i])
pt_etr89 <- SpatialPoints(pt, proj4string = CRS("+init=epsg:25830"))
pt_etr89x<-spTransform(pt_etr89, CRS("+init=epsg:4326"))
pt_etr89x_coord<-coordinates(pt_etr89x)
lat<-c(lat,pt_etr89x_coord[2])
lon<-c(lon,pt_etr89x_coord[1])
}
coord<-data.frame(lat,lon)
write.csv(coord,"C:/Users/vbulavina/Documents/R/traffic/newfile2.csv")

```

Ilustración 15. Conversión de coordenadas

El código anterior realiza la conversión de coordenadas UTM a la geográfica, creando una lista de los registros de las columnas X e Y del marco de datos inicial y transformándolos a la latitud y longitud, respectivamente. El último paso para el proceso de transformación fue unir las columnas “Date” e “Intensity” con “Latitude” y “Longitude”, acabo de convertirlo y guardarlo como archivo .csv:

```

newfile2$date <- date$date
newfile2$intensity <- newfile$intensity
write.csv(date,"C:/Users/vbulavina/Documents/R/traffic/coorddate.csv")

```

Ilustración 16. Guardar .csv

Aquí está el resultado final que solía subir a mi mapa interactivo:

	lat	lon	date	intensity
1	39.4850275868745	-0.370305379783685	4/30/2019	111
2	39.4803391640159	-0.373797509518353	4/30/2019	55
3	39.483154932649	-0.37012979649995	4/30/2019	42
4	39.4533732453693	-0.366024955208796	4/30/2019	167
5	39.4613339815259	-0.336920998626582	4/30/2019	42
6	39.4540766306513	-0.331021747855178	4/30/2019	167
7	39.4480294484164	-0.392010138347727	4/30/2019	126
8	39.4850275868745	-0.370305379783685	4/30/2019	55
9	39.4803391640159	-0.373797509518353	4/30/2019	0
10	39.483154932649	-0.37012979649995	4/30/2019	84
11	39.4850275868745	-0.370305379783685	5/1/2019	55
12	39.4803391640159	-0.373797509518353	5/1/2019	55

Ilustración 17. Datos finales

Paso 3: Crear el mapa básico

Después de definir tres archivos comencé con la construcción del mapa. Como mi objetivo era trazar los datos de tráfico de la ciudad de Valencia, configuré el mapa para que la vista inicial sea Valencia.

Para crear un mapa, necesitaba hacer dos cosas:

1. Necesitaba crear un output de ui.R
2. Necesitaba coger este output como input, en server.R y crear el mapa. Para hacer esto, utilicé una función que llama *renderLeaflet()*

Aquí están los métodos de la cadena para crear un mapa de Valencia:

- *leaflet()* crea el widget del mapa
- *addTiles()* agrega los mosaicos predeterminados del mapa de OpenStreet

- `setView()` establece la vista en las coordenadas proporcionadas con el nivel de zoom proporcionado, en este caso Valencia. También hay opciones para crear el mapa sin el método `setView`:
 - `fitBounds()` ajusta la vista en el rectángulo `[lng1, lat1] - [lng2, lat2]`;
 - `clearBounds()` borra el límite, de modo que la vista será determinada automáticamente por el rango de datos de latitud/longitud en las capas del mapa si se proporciona.

Para configurar la vista del mapa en Valencia, primero agregué sus coordenadas al archivo `server.R` y ejecuté la aplicación para guardar la vista de coordenadas para el futuro.

```
library(shiny)
df = readRDS("C:/Users/vbulavina/Documents/R/traffic/sample_data_cars.Rds")

# Define server logic required to draw a histogram
shinyServer(function(input, output) {
  data <- reactive({
    x <- df
  })

  output$mymap <- renderLeaflet({
    df <- data()
    m <- leaflet() %>%
      addTiles() %>%
      setView(lng=-0.376288, lat=39.469907, zoom=11)

    m
  })
})
```

Ilustración 18. setView código



Ilustración 19. Resultado del mapa básico

Después de llegar que Valencia este mostrado en un mapa, avancé hacia mi objetivo principal.

Paso 4: Mapear los datos con el Leaflet

En mi conjunto de datos tengo más de 20 000 de puntos por cada archivo. En teoría, puede tardar mucho en renderizar y trazar todos los puntos en el mapa. Luego, en mi archivo global.R, leí el pequeño conjunto de datos que lleva una cantidad de tiempo mucho más corta sólo para comprobar cómo funciona el código.

Para asignar los puntos, primero, elimine la función `setView()` ya que la vista se configurada automáticamente en Valencia. Luego pasé a la función de Leaflet y agregué el mapa base. La parte importante sucede con el método `addMarkers()`. Acabo de coger Latitud y Longitud ya estructuradas y también agregué un menú desplegable. El menú desplegable aparecerá cuando hagamos click en la opción de fecha e imprimirá la fecha correspondiente y nos mostrará los puntos en el mapa. De esta manera, por ejemplo, podemos verificar los días en que las bicicletas y coches se usan más o menos, incluso si es un fin de semana o el día del trabajo. Para conseguir elección de los días, en tabla de coordenadas cogí la columna de fecha. Además, depende de una fecha en una ventana emergente que se muestra la frecuencia de aparición de automóviles o bicicletas.

Después de los cuatro pasos, la versión final de los archivos se ve así:

- ui.R

```

library(shiny)
library(leaflet)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Traffic situation in Valencia"),
  sidebarLayout( position = "left",

    # Define the sidebar with one input
    sidebarPanel(
      selectInput("date", "Date:",
                 choices= coorddate$date),
      hr(),
      helpText("Dates for the analysis of usage.")
    ),

    # sidebar with a slider input for number of bins
    ui <- fluidPage( mainPanel(leafletoutput("mymap",height = 400)
      )
    )
  )
)))

```

Ilustración 20. ui.R

ui.R nos muestra las opciones de interfaz que he elegido para mi aplicación, como el título y el panel de la barra lateral para mostrar el menú desplegable de elección y, finalmente, el panel principal para la salida del mapa, relativamente.

- server.R

```

library(shiny)

bikeicon <- makeIcon(
  iconUrl = "https://image.flaticon.com/icons/png/512/34/34665.png",
  iconWidth = 20, iconHeight = 30,
  iconAnchorX = 20, iconAnchorY = 40
)

caricon <- makeIcon(
  "car.png",
  iconWidth = 60, iconHeight = 50,
  iconAnchorX = 20, iconAnchorY = 40
)

server <- function(input,output, session){

  output$mymap <- renderLeaflet({

    map3 <- leaflet(sample_data)%>%
      addProviderTiles(providers$CartoDB.Positron) %>%
      addMarkers(data = final2, lat = ~lat,
                lng = ~lon, icon = caricon, popup = paste("Frequency", result2$intensity)) %>%
      addMarkers(data = newfile2,lat = ~lat,
                lng = ~lon, icon = bikeicon, popup = paste("Frequency", coorddate$intensity))

    map3
  })
}

```

Ilustración 21. server.R

En `server.R`, como se mencionó anteriormente, coloqué parámetros importantes, como en la función `addProviderTiles()` elegí un estilo del mapa que quería para la salida, así como para los marcadores en la función `addMarkers()`, personalicé a mi gusto y señalé claramente dónde están los puntos de los autos y dónde están los puntos de las bicicletas.

- `global.R`

```
library(shiny)
library(leaflet)

#example data set
#df <- readRDS("./sample_data.rds")
```

Ilustración 22. global.R

Este archivo contiene las bibliotecas que quisiéramos para la ejecución de la aplicación y un ejemplo utilizado para las pruebas.

- `process.R`

```
#bicycle data
newfile2 <- read.csv("./newfile2.csv", stringsAsFactors = F)
date <- read.csv("./date.csv", stringsAsFactors = F)
names(date)[names(date) == "V1"] <- "date"
newfile2$date <- date$date
write.csv(date, "C:/Users/vbulavina/Documents/R/traffic/coorddate.csv")

#data with the date
coorddate <- read.csv("~/R/traffic/coorddate.csv", header=FALSE)
coorddate = coorddate[-1,]
names(coorddate)[names(coorddate) == "V1"] <- "lat"
names(coorddate)[names(coorddate) == "V2"] <- "lon"
names(coorddate)[names(coorddate) == "V3"] <- "date"

#cars data upload and cleaning
result <- read.csv("~/R/traffic/result.csv", header=FALSE)
result <- result[-c(63971324, 63971325, 63971326, 63971327, 63971328, 63971329, 63971330, 63971331, 63971332), ]
result2 <- result[complete.cases(result),]
names(result2)[names(result2) == "V1"] <- "lat"
names(result2)[names(result2) == "V2"] <- "lon"
names(result2)[names(result2) == "V4"] <- "date"
result2 <- result2[result2$date %in% c("2019-04-03", "2019-04-04", "2019-04-04", "2019-04-05", "2019-04-06", "2019-04-07",

#cars data converting code
pt<-list(lon= c(725660.333, 726029.995, 725881.654, 725202.672, 725582.171, 726433.336, 726366.919, 726105.759, 726345.77,
pt_etrs89 <- SpatialPoints(pt, proj4string = CRS("+init=epsg:25830"))
pt_etrs89x<-spTransform(pt_etrs89, CRS("+init=epsg:4326"))
pt_etrs89x_coord<-coordinates(pt_etrs89x)
final2 <- data.frame(pt_etrs89x_coord)

#frequency calculations
carsfreq <- table(result2)
bicifreq <- table(newfile2)
```

Ilustración 23. process.R

El archivo donde procesé los datos.

Y la versión final del mapa se ve así:

Traffic situation in Valencia

Date:
4/3/2019

Dates for the analysis of usage.



Ilustración 24. Aplicación final

7. Análisis de resultados

Después de crear una aplicación y hacer intentos con los datos reales, apareció la parte más interesante: el análisis de datos. ¿Por qué es la parte más interesante? ¡La respuesta no fue complicada! Porque después de visualizar y analizar datos puede obtener resultados inesperados, que pueden conducir a una toma de decisiones o a predecir lo impredecible.

Teniendo en cuenta la base de la aplicación y los datos de tráfico que tiene, echemos un vistazo y hablamos de las explicaciones.

Para comprender mejor los datos de salida del mapa, hablemos sobre los criterios a los que debemos prestar atención. En primer lugar, los marcadores del mapa son siempre estables, pase lo que pase. Dado que los registros están todo el tiempo en el mismo lugar, sería imposible que el punto se moviera repentinamente por el mapa. Para ser concretos, para los autos tuve 98 registros y para las bicicletas solo 7, no muchos, pero aún así están llenos de información valiosa. La siguiente observación a la que debemos prestar atención es el rango de fechas. Tuve disponibles solo datos del 3.04.2019 al 6.05.2019, por lo que pude comparar solo un mes del uso de la carretera. Para comprender mejor la influencia de la fecha, busqué un registro antiguo de las condiciones climáticas que pudieran ayudar a encontrar algunas explicaciones. Junto con los datos de fecha y la posición de los punteros, otro registro importante para tener en cuenta son los datos de intensidad, que en realidad representan la frecuencia del uso del transporte tanto para automóviles como para bicicletas.

Análisis de los registros

Para comprender mejor la salida de los registros, verifique los registros históricos de las temperaturas climáticas. Después de verificar las temperaturas y las condiciones climáticas durante el período de tiempo mencionado, descubrí algunas fechas interesantes para analizar. Por ejemplo, tomé en cuenta tres fechas principales, que podrían afectar el uso del transporte, tanto en automóviles como en bicicletas. La primera fecha a la que me gustaría prestar atención es el 5 de abril, es viernes, la temperatura era de 19 grados y no hubo cambios climáticos (Accuweather, 2019). Aquí se puede ver el resultado que obtuve:

Traffic situation in Valencia

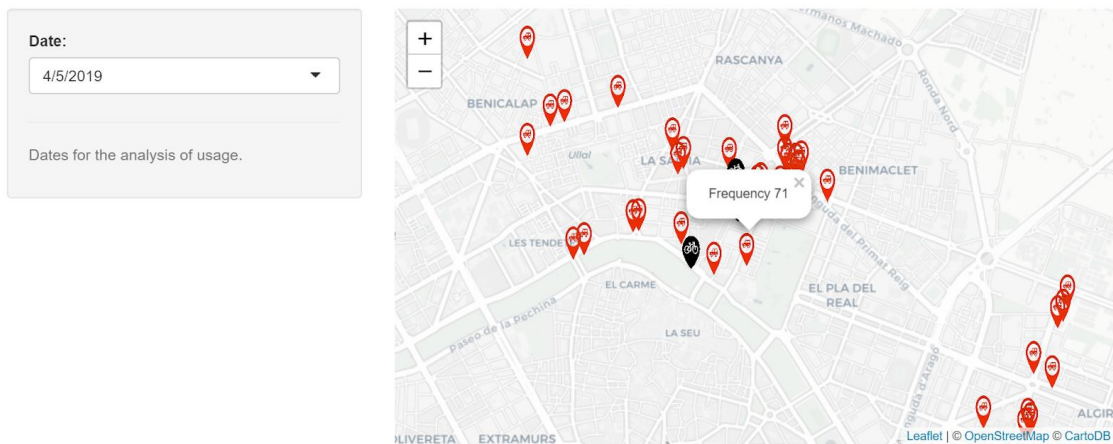


Ilustración 25. 5.04 bicicletas

Traffic situation in Valencia

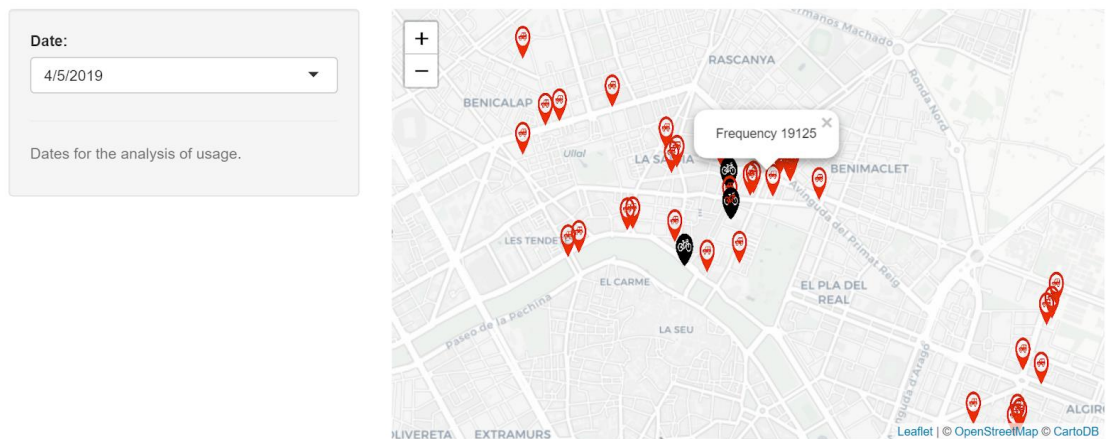


Ilustración 26. 5.04 coches

Ilustración 23 nos muestra el uso de las bicicletas en esta fecha e Ilustración 24 de los automóviles, respectivamente. ¿A qué conclusión nos puede llevar? Una diferencia obvia entre los registros de intensidad entre las bicicletas y los automóviles. Aquí podemos ver que el 5 de abril tenemos 19125 registros de coches y tan solo 71 registros de bicicletas. ¿Qué puede significar esto? Que, sin tener en cuenta las buenas condiciones climáticas, la cantidad de uso de los automóviles es mucho mayor que la de las bicicletas. ¿Qué podría causar tal diferencia? En primer lugar hay que tener en cuenta que es un día entre semana. Al ser viernes las personas se dirigen hacia o desde el trabajo o los estudios u otras actividades que cualquier persona podría hacer el último día laboral de la semana.

Luego, decidí analizar otras fechas interesantes que me llamaron la atención. Observando los datos históricos del clima (Accuweather, 2019), he elegido dos días particulares para el análisis del tráfico. El primero de ellos es el 18 de abril. Día laboral, jueves, pero ya a 15 grados de la temperatura. Dato curioso también, que el 18 de abril fue el último día laboral antes de la

Pascua. Eso podría significar que mucha gente ya podría estar de vacaciones. Aquí está disponible el registro del mapa:

Traffic situation in Valencia

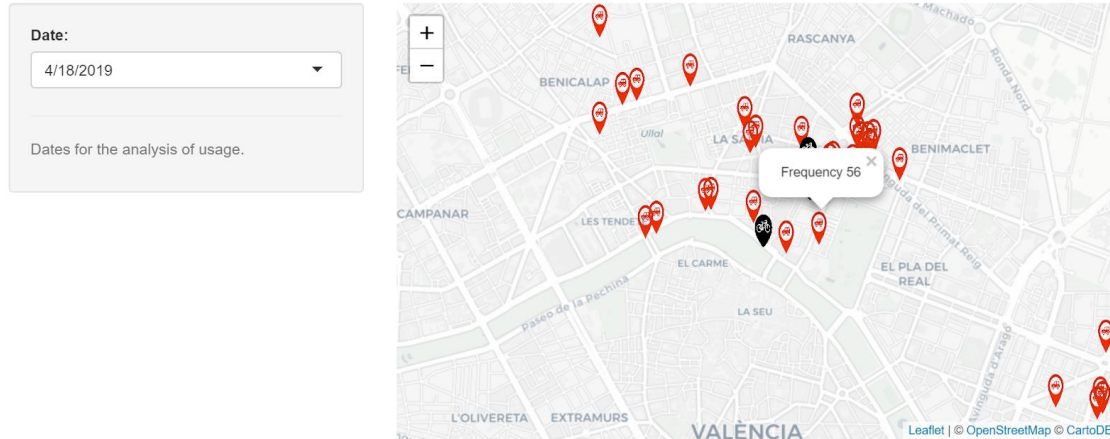


Ilustración 27. 18.04 bicicletas

Traffic situation in Valencia

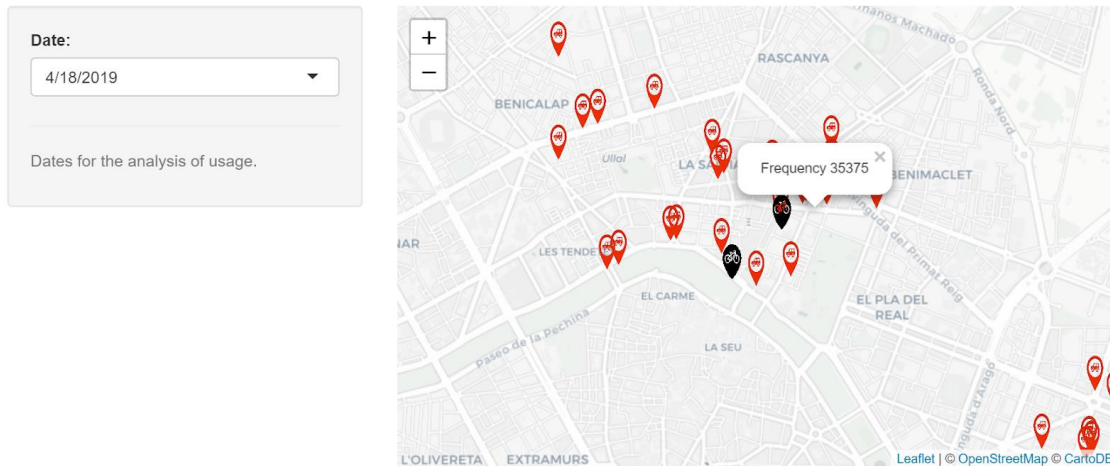


Ilustración 28. 18.04 coches

En cuanto a este resultado, podemos ver que la cantidad de usuarios de automóviles es igual de alta, es decir, 632 veces mayor que los usuarios de bicicletas. ¿Qué podría causar la diferencia? En primer lugar, como se mencionó anteriormente, es el último día hábil antes del período de vacaciones y la gente estaba utilizando todas las oportunidades para realizar sus actividades antes de sumergirse en las vacaciones. Eso podría aumentar fácilmente la cantidad de automóviles y disminuir el uso de la bicicleta. Otro punto para tener en cuenta es que la mayoría de los estudiantes, que son la población más alta que usan las bicicletas también estaban de vacaciones, eso podría causar la disminución en los registros de las bicicletas. La última razón para tener en cuenta es que las condiciones climáticas no eran deseables para los usuarios de

bicicletas. Solo 15 grados seguidos con la lluvia (Vázquez, Levante, 2019). Estas tres razones podrían ser el motivo perfecto de la diferencia del uso del transporte.

Y ahora llegamos al último registro para analizar. El último día que elegí prestar atención fue el 20 de abril. Además de los registros y artículos históricos, el clima de ese sábado no fue el más deseable para un sábado que además coincide con festivo en España. Como descubrí, la temperatura más alta fue de 16 grados, acompañada por fuerte lluvia y tormenta (Vázquez, 2019). Por eso sería curioso analizar los registros de esta fecha. Aquí está disponible el registro del mapa:

Traffic situation in Valencia

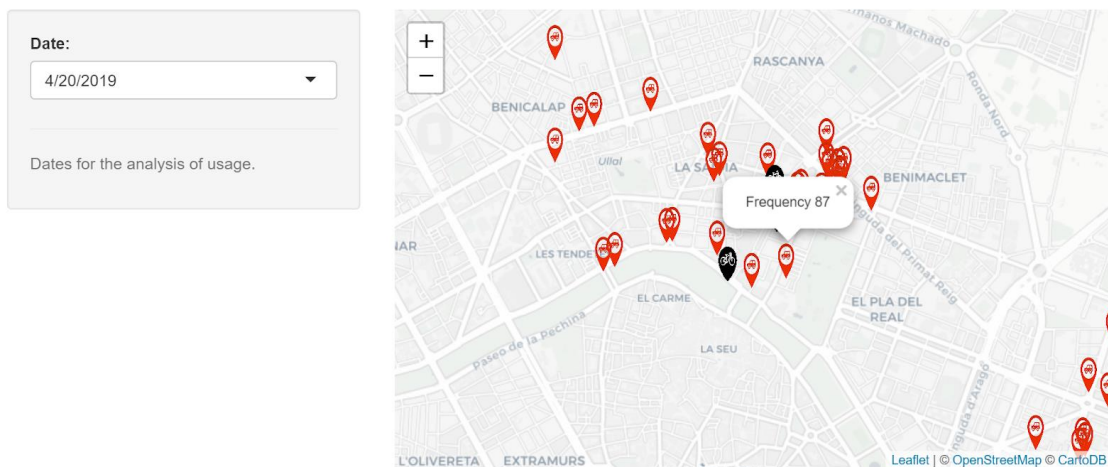


Ilustración 29. 20.04 bicicletas

Traffic situation in Valencia

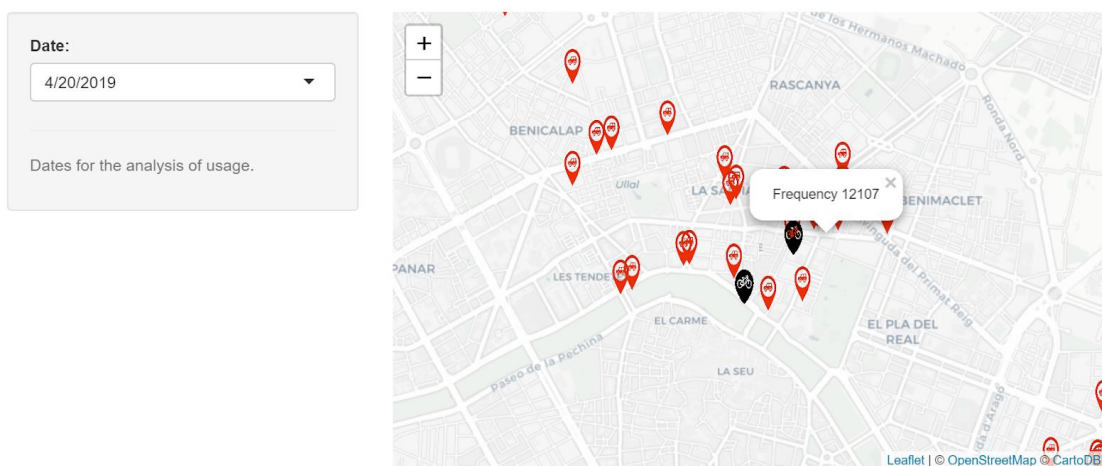


Ilustración 30. 20.04 coches

Al igual que con los registros anteriores, vemos claramente la diferencia entre el uso del transporte. Una vez más, se puede ver una gran diferencia en la cantidad de uso de automóviles

y bicicletas, pero esta vez los registros muestran una disminución increíble para los usuarios de automóviles. ¿Qué podría significar esto? En primer lugar, como se mencionó anteriormente, las condiciones climáticas no invitaban a salir de casa, por lo que muchos usuarios decidieron no utilizar sus coches. Pero el hecho sorprendente, que me hizo pensar profundamente, ¿por qué el registro mostró tantos usuarios de bicicletas en las calles? La única razón lógica podría ser que era festivo de Pascua y la mayoría de las personas que podíamos encontrar en las calles en estos días eran turistas. Como sabemos, los turistas van a hacer turismo sin importar las condiciones climáticas y aprovechan todas las oportunidades para ver la ciudad que visitaron, en nuestro caso, Valencia. A pesar de que el clima no invita a utilizar vehículos como la bicicleta el número de usuarios es bastante alto. Eso solo puede significar, que todavía hay usuarios que prefieren bicicletas en lugar de automóviles dentro de Valencia.

8. Solución de problemas

Mientras estaba creando la aplicación me enfrenté con varios problemas tanto con el contenido de datos como con las especificaciones de la aplicación.

1. Como mencioné anteriormente, el conjunto de datos original era un grupo de pequeños archivos .csv que necesitaba unir para realizar el trabajo, antes de lograr el archivo final tuve que luchar por ello. Todos los intentos que hice para unir los archivos en RStudio fallaron, me enfrenté a diferentes problemas: desde el "archivo no encontrado" hasta las horas de la ejecución del código, pero sin el resultado al final.

Cómo se resolvió: el programa Python, mostrado en la página [23](#), que se colocó en la carpeta de origen y se ejecutó a través de la línea de comando.

2. Cuando cargué un archivo unido en el entorno R, me sorprendió ver el carácter del factor para los registros numéricos. Eso me hizo pensar que el archivo podría tener muchos registros que faltaban o NA, y que no todos los registros coinciden e incluso se están descargando del portal de datos abiertos del Ayuntamiento, podría haber conjuntos de datos incompletos.

Cómo se resolvió: ejecución de la función `complete.cases()`. Me ayudó a deshacerme de muchos valores perdidos y registros de NA por casi el doble de tamaño.

3. Después de unir y limpiar los registros numéricos del conjunto de datos, todavía mostraban el carácter del registros como factor, eso era completamente incompatible con los paquetes `ggmap()` y `rgdal()` para la conversión de coordenadas, porque estos paquetes solo funcionan con registros numéricos.

Cómo se resolvió: ejecución de la función `as.numeric()` con la opción `decimal` para no perder la longitud original de un campo de datos.

4. En algún momento, el código de la conversión de coordenadas fallaba solo con los registros de datos de los automóviles e independientemente mostraba una transformación incorrecta y las coordenadas necesarias mostraban las ubicaciones de Kenia, Sudán o Argelia.



Análisis de datos de movilidad en la ciudad de València.

Cómo se resolvió: después de crear una lista de coordenadas para las columnas X e Y e insertar la lista en el código en lugar de coger datos desde el fichero .csv, de conversión, se resolvió el problema.

9. Conclusión

Este trabajo está dedicado a la creación de la aplicación de un análisis del flujo de datos de los usuarios de automóviles y bicicletas en la ciudad de Valencia. Para su implementación, se encontraron los datos habituales para una tarea similar: las coordenadas de los registros del flujo de automóviles y bicicletas en el territorio de la ciudad, las fechas de los datos registrados, la intensidad del flujo en el número de vehículos por una coordenada específica. Por otro lado, no había información sobre la marca del automóvil, ni información sobre automovilistas, ni sobre ciclistas, solo la trayectoria del movimiento durante un cierto período de tiempo. La tarea consistía en analizar y visualizar datos para determinar la frecuencia de uso de automóviles y bicicletas y pensar en la causa de la intensidad del tráfico. Además, otra tarea consistía en comparar el uso de un vehículo en ciertas condiciones, tales como: condiciones climáticas, día de la semana, días festivos.

Como resultado del trabajo, se creó un modelo en forma de una aplicación interactiva con una demostración de un mapa de la ciudad de Valencia, que permite presentar las trayectorias en un formato conveniente para su posterior análisis. Esta es una estructura que almacena las coordenadas de los puntos de registro.

Dicho modelo permitió que cada día seleccionado en el menú de la aplicación determinara la velocidad de flujo tanto de los automóviles como de las bicicletas. Además, en función de los datos obtenidos, que se muestran en las ventanas emergentes, puede determinar la intensidad del tráfico para un transporte en particular según la fecha.

Después de analizar los datos disponibles, se descubrió que, a pesar de la gran cantidad de usuarios de bicicletas y senderos para bicicletas, la cantidad de usuarios de automóviles es varias veces mayor. También podemos notar que el estado ha abordado este tema de cerca y ha establecido registros de usuarios y bicicletas para carriles para bicicletas y carriles para automóviles en las calles de Valencia para llamar la atención de los residentes y turistas sobre estos datos.

¿Por qué es tan importante prestar atención a esto?

En primer lugar, porque si aumenta el número de usuarios de bicicletas y carriles para bicicletas, el número de usuarios de automóviles disminuirá automáticamente y hará que la ciudad esté menos ocupada.

En segundo lugar, la sobrecarga de la ciudad con los automóviles está directamente relacionada con la contaminación ambiental y las emisiones de escape, que definitivamente empeora la condición del planeta en millones de veces. Por lo tanto, dejar de usar un automóvil y dar preferencia a una bicicleta, de una forma u otra, puede afectar la mejora del medio ambiente.

En tercer lugar, uno puede notar un aumento en la obesidad de la población debido a la inactividad, lo que también provoca el uso de automóviles. Al volver a usar una bicicleta, una persona no solo ayuda a la naturaleza, sino que también se ayuda a sí misma realizando una cierta cantidad de actividad física, previniendo enfermedades tan terribles como la diabetes, la obesidad y la insuficiencia cardíaca.

Gracias al gobierno, que aumenta cada año el número de carriles para bicicletas dentro y fuera de la ciudad, y a las pequeñas empresas que promueven el uso frecuente de bicicletas como Valenbisi.

10. Bibliografía

- Accuweather*. (2019). Recuperado de Accuweather:
<https://www.accuweather.com/es/es/valencia/310683/april-weather/310683>
- Agafonkin, V. (2010-2016). *Rstudio Github*. Recuperado de Rstudio Github:
https://rstudio.github.io/leaflet/map_widget.html
- Bivand, R. (2019, Mayo 29). *Cran.r-project*. Recuperado de Cran.r-project:
<https://cran.r-project.org/web/packages/rgdal/rgdal.pdf>
- Chang, W. (2019, Abril 22). *Cran.r-project*. Recuperado de Cran.r-project:
<https://cran.r-project.org/web/packages/shiny/index.html>
- Cheng, J. (2018, Agosto 27). *Cran.r-project*. Recuperado de Cran.r-project:
<https://cran.r-project.org/web/packages/leaflet/leaflet.pdf>
- Foote, K. D. (2018, Septiembre 25). *Dataversity*. Recuperado de Dataversity:
<https://www.dataversity.net/brief-history-analytics/#>
- Inchttps://shiny.rstudio.com/articles/, R. (2017). *Shiny*. Recuperado de Shiny:
<https://shiny.rstudio.com/articles/>
- Kahle, D. (2019, Febrero 5). *Cran.r-project*. Recuperado de Cran.r-project:
<https://cran.r-project.org/web/packages/ggmap/ggmap.pdf>
- Mohorte. (2018, Noviembre 15). *Magnet*. Recuperado de Magnet:
<https://magnet.xataka.com/en-diez-minutos/valencia-ha-logrado-multiplicar-su-numero-ciclistas-su-receta-simple-poner-carriles-bici>
- Moreno, P. (2019, Agosto 8). *Las Provincias*. Recuperado de Las Provincias:
<https://www.lasprovincias.es/valencia-ciudad/parque-vehiculos-valencia-20190808001752-ntvo.html>
- Python*. (2001-2019). Recuperado de Python: <https://www.python.org/about/>
- Roig, J. B. (2017, Agosto 29). *Levante*. Recuperado de Levante: <https://www.levante-emv.com/valencia/2017/08/30/numero-vehiculos-capital-aumenta-primera/1609476.html>
- R-project*. (n.d.). Recuperado de R-project: <https://www.r-project.org/about.html>
- Sales. (2017, Septiembre 18). *Newgenapps*. Recuperado de Newgenapps:
<https://www.newgenapps.com/blog/6-reasons-why-choose-r-programming-for-data-science-projects>
- SAS . (n.d.). Recuperado de Wikipedia: [https://en.wikipedia.org/wiki/SAS_\(software\)](https://en.wikipedia.org/wiki/SAS_(software))
- Shiny from RStudio*. (2017). Recuperado de Shiny from RStudio:
<https://shiny.rstudio.com/tutorial/written-tutorial/lesson2/>
- TOP 10 Data Analytics tools*. (n.d.). Recuperado de ProSchool:
<https://www.proschoolonline.com/blog/top-10-data-analytics-tools>
- Valencia*. (n.d.). Recuperado de Valencia:
http://www.valencia.es/ayuntamiento/datosabiertos.nsf/fCategoriaVistaAcc_busqueda?readForm&Vista=vCategoriasAccTodas&Categoria=Sin_categoria&lang=1&nivel=2&seccion=1&expand=6&bdorigen=&idapoyo=22ADF97C1FD223B5C1257C55003BD01F
- Vázquez, M. (2019, Abril 16). *Levante*. Recuperado de Levante: <https://www.levante-emv.com/comunitat-valenciana/2019/04/16/tiempo-valencia-gota-fria-semana-santa-2019/1863251.html>

Vázquez, M. (2019, Abril 19). *Levante*. Recuperado de Levante: <https://www.levante-emv.com/comunitat-valenciana/2019/04/19/tiempo-valencia-semana-santa-2019-viernes-santo/1864322.html>

Wikipedia. (n.d.). Recuperado de Wikipedia:
https://en.wikipedia.org/wiki/Tableau_Software

Wyckoff, J. P. (2018, Septiembre 21). *Medium*. Recuperado de Medium:
<https://medium.com/@joyplumeri/how-to-make-interactive-maps-in-r-shiny-brief-tutorial-c2e1ef0447da>