Imagen creada por IA generativa usando el siguiente prompt: 'Estoy haciendo un Jupyter Notebook de la memoria de un EDA. La cabecera quiero que me quede bonita. ¿Podrías generarme una imagen de un robot escribiendo a máquina de escribir? No muy complejo, con un fondo muy simple, (Luego editada con paint)'

Memoria del EDA: Análisis de la retirada de vehículos por parte de la EMT Madrid (2016-2023): motivos, momento de la retirada y tiempo en el depósito.

ORGANIZACION DEL REPOSITORIO

> El nombre de la carpeta principal es simplemente "EDA". En esta están la presentación en PowerPoint, esta memoria (en PDF y en Jupyter Notebook) y un
> README (en PDF y en formato Markdown). Además, hay 3 carpetas:
>
> - **Dataset:** Aquí se guardan, en diferentes subcarpetas, los archivos descargados y creados durante el desarrollo del EDA.
> - **Imágenes:** Imágenes creadas con inteligencia artificial, descargadas de internet y creadas con librerías de visualización de datos.
> - **Jupyter:** Aquí están los diferentes notebooks con los que se ha realizado el presente estudio.
>
> > Dentro de la carpeta de Jupyter también hay carpetas de imágenes. Estas fueron las que se usaron para ilustrar los notebooks de Jupyter. Desde otra carpeta lejana se hacía complicado poder mostrarlas.

ELECCIÓN DEL TEMA

> En un principio, planeaba realizar un EDA para determinar la vida útil de los vehículos en España, entendida como el tiempo desde la primera matriculación hasta su baja y envío al desguace. Tras buscar sin éxito datasets con la información necesaria, encontré unos archivos que contenían datos suficientes para llevar a cabo un EDA. El tema cambió respecto a la idea inicial; estos archivos proporcionaban información sobre los **vehículos retirados de las vías públicas de Madrid entre 2016 y 2023**. Detallan el tipo de vehículo, fecha y hora de la retirada, depósito al que se envió, grúa que lo retiró, motivo de la retirada y un identificador (sin la matrícula por protección de datos). Estos datos los hallé en la página del **Ministerio para la Transformación Digital y de la Función Pública**, donde se indica que fueron proporcionados por la **EMT (Empresa Municipal de Transportes de Madrid)**, responsable del servicio. Los archivos están disponibles por años, y de cada año hay dos archivos, '**Entradas**' y '**Salidas**'. Estos datasets tienen información sobre la fecha y hora de la retirada

> de la vía pública y entrada al depósito, motivo y lugar de retirada, grúa que lo retiró, tipo de vehículo y cualquier otro dato necesario. En el archivo de salida también se informa del depósito de salida, que no siempre coincide con el depósito al que se llevó el vehículo en un primer momento. Además, se incluyen **tablas auxiliares** con el significado de códigos de motivo de retirada, turno, tipos de vehículos e información sobre los distintos depósitos municipales. Los archivos están disponibles en formato **CSV y XML**.

ELECCIÓN DEL ESTILO DE LA PRESENTACIÓN

> En la presentación de PowerPoint, he incluido varios **personajes** robots generados con el motor de inteligencia artificial de '**OpenAI**' a través de '**Copilot**', en los cuales se puede ver el logotipo de **EMT Madrid**. La **inteligencia artificial** es una herramienta ampliamente utilizada en la actualidad, especialmente en la **ciencia de datos**, por lo que consideré apropiado incorporarla. El público al que va dirigida la presentación se encuentra en formación y se presentará en un contexto académico no laboral, por lo que existe cierta **libertad artística**. Por lo tanto, añadir estos personajes a la presentación puede contribuir a hacerla más amena, **estéticamente atractiva y fácil de seguir**
>
> > **Los prompts usados para generar las imágenes fueron:**
> >
> > - "Genérame una imagen, fondo liso blanco, de un robot señalando con el dedo índice con el brazo a 45 grados, imagen simple, blanco y negro."
> > - "Genérame una imagen, fondo liso blanco, de un robot señalando hacia adelante, que parezca que está dando una orden, imagen simple, blanco y negro."
> > - "Genérame una imagen, fondo liso blanco, de un robot que se está preguntando algo, imagen simple, blanco y negro."
> > - "Genérame una imagen, fondo liso blanco, de un robot que está levantando un dedo, blanco y negro."
> > - "Genérame una imagen, fondo liso blanco, de un robot que está levantando dos dedos, blanco y negro."
> > - "Genérame una imagen, fondo liso blanco, de un robot que está diciendo adiós con la mano, blanco y negro."
> >
> > Usando Microsoft Paint y PowerPoint, edité las imágenes para añadirles el logotipo de EMT, el cual descargué de su página web.

LIMPIEZA Y PREPARACIÓN DE LOS DATOS.

> En general, estaban bastante limpios; apenas requirieron limpieza.

ENTRADAS

> En un Jupyter Notebook llamado **'preparacion_datos_entrada.ipynb**, utilizando la biblioteca **pandas** y la función **'read_csv'**, volqué los archivos en diferentes **DataFrames**, uno por año, llamándolos 'entradas[aquí el año]' para poder hacer una lectura con comodidad. Las columnas son las siguientes:
>
> > - **idVehiculo**: Es un identificador alfanumérico para cada vehículo.
> > - **deposito**: El nombre del depósito al que se llevó el vehículo en cuestión.
> > - **tipoVehiculo**: Es un código numérico que indica el tipo de vehículo en cuestión. El significado del código está indicado en un archivo CSV auxiliar.
> > - **motivo**: Es un código numérico que indica el motivo por el que se retiró el vehículo de la vía pública. El significado del código está indicado en un archivo CSV auxiliar.
> > - **Turno**: Es un código numérico que indica el turno en el que se retiró el vehículo. El significado del código está indicado en un archivo CSV auxiliar.
> > - **Grua**: Es un código alfanumérico que indica la grúa que retiró el vehículo.
> > - **fechaEntrada**: Indica la fecha y hora en que se retiró el vehículo de la vía publica.
> > - **tipoVia**: Indica el tipo de vía desde la cual se retiró el vehículo.
> > - **Via**: Nombre de la vía.
> > - **numero**: Número de la vía.
> > - **cp**: Código postal de la vía.
>
> Los pormenores de la preparación de datos están explicados en el mismo Jupyter Notebook **'preparacion_datos_entrada.ipynb'**, pero aquí está el resumen:
>
> - Al visualizar el DataFrame de cada año, noté que en el año 2016 había una columna llamada 'Unnamed: 11' con solo una entrada válida. Esta columna no existe en ningún otro año, por lo tanto, la eliminé en la celda **[4]**.
> - Después de esto, utilizando el comando 'info()' y visualizando los DataFrames, noté que no todos los años tenían el mismo formato de fechas, ni estaban en formato 'datetime'. Por lo tanto, a partir de la celda **[5]**, cambié el formato de fecha a 'datetime' y además creé una columna solo con la hora.
> - En la celda **[9]**, sustituí los códigos por la información de 'turno' y 'tipoVehículo' utilizando la información de las tablas auxiliares. Además, añadí una columna al DataFrame con un booleano para saber si fue o no en fin de semana.
> - Utilizando datos de la página del ayuntamiento, creé un diccionario con las fechas y nombres de los días festivos de cada año en la celda [11], para poder crear una columna con información sobre si cada vehículo fue retirado en día festivo, y qué festividad, o si no era festivo.
> - En la celda **[16]**, comencé a comprobar si algún identificador de vehículo aparecía repetido. Noté que sí, y tras algunas comprobaciones, le cambié el nombre a uno de ellos.
> - En la celda **[21]**, uní los DataFrames de todos los años en un único DataFrame.
> - En la celda **[28]**, mientras trataba de guardar todos los DataFrames en formato Parquet, noté que en la columna que indica el tipo de vehículos había valores numéricos. Algunas

> Pregunté a ChatGPT cuál era el tipo de archivo más apropiado para guardar dataframes y que mantuviera los tipos de datos. Me aconsejó Parquet. entradas no habían sido sustituidas por sus significados y esos códigos no aparecían en la tabla auxiliar. Les di un valor provisional mientras encontraba su significado.

Después de esto, pude guardar los archivos listos para ser usados.

> > Para averiguar los codigos faltantes en la columna de tipos de vehiculos envie un correo, a varios organismos, del ayuntamiento de madrid, el cual pondré al final.

SALIDAS

> Al igual que para 'Entradas', utilicé la biblioteca **pandas** y la función **'read_csv'** para volcar los archivos en diferentes **DataFrames**, uno por año. Esta vez, los llamé **'salidas[aquí el año]'** para poder hacer una lectura correcta de los mismos. Esto en un jupyter notebook llamado **'preparacion_datos_salida.ipynb**:
>
> Algunas columnas son comunes a los datos de 'Entradas' otras no, las columnas son las siguientes:
>
> - **idVehiculo**: Identificador alfanumérico único para cada vehículo.
> - **tipoVehiculo**: Código numérico que indica el tipo de vehículo. El significado del código está indicado en un archivo CSV auxiliar.
> - **motivo**: Código numérico que indica el motivo por el cual se retiró el vehículo de la vía pública. El significado del código está indicado en un archivo CSV auxiliar.
> - **Turno**: Código numérico que indica el turno en el cual se retiró el vehículo. El significado del código está indicado en un archivo CSV auxiliar.
> - **Grua**: Código alfanumérico que indica la grúa que retiró el vehículo.
> - **tipoVia**: Indica el tipo de vía desde donde se retiró el vehículo.
> - **Via**: Nombre de la vía.
> - **numero**: Número de la vía.
> - **cp**: Código postal de la vía.
> - **depositoEntrada**: Nombre del depósito al que se llevó el vehículo en un primer momento.
> - **fechaEntrada**: Fecha y hora de cuando se retiró el vehículo de la vía pública y entró al depósito.
> - **depositoSalida**: Nombre del depósito del cual salió el vehículo.
> - **fechaSalida**: Fecha y hora de cuando salió el vehículo del depósito.
>
> > > No siempre el deposito al que se lleva el vehiculo en un primer momento y el deposito de salida es el mismo.
>
> Los pormenores de la preparación de datos de Salidas tambien están explicados en el Jupyter Notebook, su nombre es **'preparacion_datos_salida.ipynb'**, pero aquí está el resumen:
>
> - La primera comprobación que hice fue en el año 2016. Comprobé si había una columna llamada 'Unnamed: 11', pero esta vez se llamaba 'Unnamed: 13'. Al igual que en Entradas, tampoco la había en otros años. Eliminé la columna en la celda número **[7]** del Jupyter Notebook.
> - En la celda **[11]**, corregí la fila con idVehiculo repetido. Esta incidencia también estaba presente en Salidas, en los mismos vehículos.

- Al intentar convertir las columnas de fechaEntrada y fechaSalida a formato datetime del dataframe Salidas2016, me dio error. El error indicaba que era en la fila '36911'. Al imprimir dicha fila, vi que la fecha de entrada, el depósito de entrada y la fecha de salida tenían los datos mezclados entre sí; además, faltaba una fecha. Usando el idVehículo, lo comprobé en la tabla de entradas. Si la fecha y/o hora en la tabla de entradas hubieran sido anteriores, la fecha en esta tabla habría sido la de salida, por lo que podríamos haberlo arreglado. Pero era la misma; nos falta un dato. Investigando, volví a leer la tabla, esta vez sin eliminar la columna 'Unnamed: 13'. Ahí estaba la información faltante. Comenté la línea donde se elimina la columna, imprimí con dicha columna, copié la información y volví a eliminarla. En la celda **[14]** arreglé este problema.
- En la fila **[15]**, pasé a formato datetime las columnas con fechas.
- En la celda **[19]**, añadí una columna a los dataframes de todos los años con los días que cada vehículo pasa en el depósito.
- En la celda **[22]**, sustituí los códigos por la información de tipos de vehículos (la incidencia ya se ha comentado) y turnos.
- En la celda **[24]** volque el archivo CSV de motivos en un dataframe, para poder verlo bien y sustituir los codigos por su significado en mis dataframess de Salidas.
- En la celda **[24]** volqué el archivo CSV de motivos en un dataframe para poder verlo bien y sustituir los códigos por su significado en mis dataframes de Salidas.
- En la celda **[29]**, los guardé en formato Parquet en la carpeta 'salidas_validas', la cual a su vez está dentro de la carpeta Dataset.
- La unión vertical de los dataframes de distintos años en uno general la hice en la celda **[31]**.

Con esto, los datos estaban prácticamente listos para verificar si existe relación entre el tiempo que pasa un vehículo en el depósito y el tipo de vehículo del que se trate.

PREPARACIÓN DE TABLAS Y FIGURAS PARA LA INTRODUCCIÓN.

La idea era comenzar la presentación con una **introducción de los datos** y mostrar cómo algunos **hechos históricos**, como la tormenta 'Filomena de 2021', el confinamiento de 2020 y también algo que se suele decir como 'en verano Madrid se queda vacía', pueden verse **reflejados en las gráficas**.

Al igual que en los dos Jupyter anteriores, en este, **'intro.ipynb'**, tras cargar las librerías que necesitaría, cargué los dataframes. Esta vez los cargué no desde el archivo que la EMT pone a disposición de quien los necesite, sino que cargué los **dataframes de entradas** (con la información de los vehículos retirados de las vías públicas) que **había preparado** en el Jupyter Notebook 'preparacion_datos_entrada.ipynb'.

Agrupando por años, en la celda **[6]**, creé una **tabla** para ver cuántos **vehículos se retiraron** de las calles cada **año**. Con esto, ya tenía la cantidad media, el año de máxima retirada y el año de menor retirada de vehículos. Estos datos me parecían **apropiados como introducción.**

Los pormenores de la preparación de la introduccion tambien están explicados en el Jupyter Notebook correspondiente, pero aquí está el resumen:

- En la celda **[9]**, utilizando las bibliotecas matplotlib y seaborn (sns), creé una figura de barras con una línea horizontal que representa la media de vehículos retirados en esos años, para mostrar gráficamente si había diferencias en el número de vehículos retirados según el año.

- En la figura anterior, aunque bastante visual, las diferencias entre los años podían ser más evidentes. Por lo tanto, en la celda **[10]**, utilizando la biblioteca **Matplotlib para trazar la gráfica** y configurar su apariencia, y **Seaborn para eliminar los márgenes adicionales**, creé una gráfica de línea. También incluí una línea horizontal que representa la media. En esta figura, **la diferencia era mucho más visible**. Ambas figuras fueron guardadas en la carpeta '**figura_intro**'.
- Para ver la evolución anual, una vez cargados los dataframes anuales, en la celda **[12]** agrupé por fecha. Como cada fila es un vehículo, al agrupar por día, se obtendría la cantidad de vehículos retirados por día.
- En la celda **[14]**, convertí la columna con la fecha en formato datetime y la establecí como **el índice** del DataFrame.
- En la celda **[16]**, usando las bibliotecas matplotlib y seaborn, creé una figura de líneas. En el **eje X** estarían **los días** del año y en el **eje Y** la cantidad de **vehículos retirados**. La gráfica demuestra lo que sospechaba: la tormenta Filomena, el confinamiento y el verano de Madrid quedan reflejados por los datos. La gran cantidad de datos hace la tabla algo incómoda de ver.
- Al intentar unir los años, no se podía porque no tenían la misma cantidad de índices (días del año). Esto se debía no solo a los años bisiestos, sino también a que en algunos años faltaban datos de ciertos días en los que no se retiraron vehículos (o esos datos no fueron incluidos). **Añadí manualmente el día al CSV** para solucionar este problema.
- Una vez arreglados esos CSV, que estaban en la carpeta "pasos_intermedios" y cargados, los uní en un solo DataFrame. En este DataFrame, **cada columna representa un año y cada fila representa un día del año**, siendo el índice el día del año. Cada entrada en el DataFrame es el número de vehículos retirados de la calle para ese día y ese año. Esto lo hice en la celda **[28]**.
- En la celda **[30]**, utilizando **'slice'**, eliminé el número del día en la fecha, dejando **solo el mes**.
- En la celda **[31]**, compruebo que no haya ningún mes escrito de más de una forma distinta.
- En la celda **[32]**, creé un diccionario para traducir los meses del año y los traduje.
- En la **[33]**, agrupé por meses, sumando las entradas.
- En la **[34]**, lo convertí a tipo entero (eran float).
- Me di cuenta de que los meses salían desordenados, así que en la celda **[35] los ordené**.
- Ahora sí, con los meses traducidos y la retirada de vehículos agrupadas por meses, en la celda **[36]**, creé una figura mucho más **cómoda de leer**, aunque el pico de retiradas de vehículos (que triplica la media) **no era visible**. Guardé la figura en la carpeta 'figura_intro'.
- En las celdas **[37] y [38]** creé una **figura** de **pastel** con el porcentaje de cada tipo de vehículos que ha sido retirado.
- Entre las celdas [40] y [43] calculé la cantidad de vehículos que habían entrado al depósito en las fechas estudiadas y no habían salido, así como la cantidad de vehículos que habían salido de los depósitos y habían entrado antes de 2016.

Los datos y las figuras para la introducción de la presentación en PowerPoint quedaron listos en este punto.

PRIMERA HIPÓTESIS

> En el Jupyter Notebook de nombre **'primera_hipo.ipynb'** se demostró que el motivo de retirada y el momento de este estaban relacionados.
>
> > Se demostró que había relación con ser, o no, fin de semana, ser, o no, festivo, y también con la hora de retirada.
>
> El jupyter esta explicado, pero aqui está un resumen:
> Una vez cargados los dataframes necesarios, utilizando la tabla auxiliar, se cambiaron los códigos de motivo por su significado.
> El motivo es bastante descriptivo, pero en la celda **[7]** escribí un código para, mediante un input, **obtener una explicación de cada motivo.**
> Hay **61 motivos** de retirada de vehículos de la vía pública. Para introducir la hipótesis, se creó en la celda **[10] una figura de pastel con los porcentajes de los motivos de retiradas.**
>
> Vi que **cuatro motivos fueron los responsables de algo más de la mitad de la retirada de vehículos (2016-2023).**
> Teniendo en cuenta que hay 61 motivos distintos, queda clara la importancia de estos cuatro motivos. De hecho, los dos motivos más comunes son responsables de casi el 40% de
> las retiradas. Este es el motivo por el que incluí este gráfico para introducir la primera hipótesis.
>
> **FIN DE SEMANA**
> Dado que **'motivo de retirada'** es una **variable categórica** y **ser o no fin de semana** es una **variable booleana** (también categórica), la prueba de **chi-cuadrado** es apropiada. Para ello, utilicé la biblioteca pandas para manejar y procesar los datos, y también la función chi2_contingency del módulo stats de la biblioteca scipy para realizar la prueba. En este caso, la estadística chi-cuadrado (más de 2000) es considerablemente mayor que los grados de libertad (54) y el valor p es 0. Por lo tanto, **podemos rechazar la hipótesis nula y afirmar que efectivamente están relacionados.**
>
> **FESTIVO**
> Ser o no festivo, y qué festivo sea, son variables categóricas (esta vez no es booleana), pero la prueba del chi-cuadrado sigue siendo una opción válida. **También rechazamos la hipótesis nula**, ya que la estadística chi-cuadrado (más de 4100) es considerablemente mayor que los grados de libertad (702) y el valor p también es 0. Esto se encuentra en la celda **[15]**.
>
> **HORA DEL DÍA**
> En este caso, comparamos una variable categórica, **motivo**, con una numérica, **hora** (donde la hora se representó en minutos, siendo las 00:00 0 y las 23:59 es 1439).
> Por lo tanto, la prueba más apropiada para este caso es la prueba de análisis de varianza, **ANOVA**. El **valor de p**, al igual que en los casos anteriores, fue **0**,
> lo que sugiere una alta significancia, es decir, ambas variables están muy relacionadas. Además, la **estadística F** (**970.6171**) es sustancialmente mayor que 1, lo que
> indica una diferencia significativa entre los motivos de retirada y la hora a la que se retiraron, **podemos rechazar la hipótesis nula y afirmar que efectivamente están relacionados.**
> La columna con la hora (en minutos) se creó en la celda **[16]** y la prueba ANOVA se realizó en la celda **[20]**.

> Después de realizar la prueba de Análisis de Varianza (ANOVA) y encontrar una relación significativa entre los grupos, se realizó la prueba de **Tukey** como análisis **post hoc**. El término 'post hoc' significa 'después de esto' y se refiere a las pruebas o análisis realizados después de la
> prueba de hipótesis inicial para investigar y comprender mejor las diferencias específicas entre los grupos.
>
> Viendo los resultados de la prueba **Tukey**, se observó que si bien no todos los motivos de retirada de vehículos guardaban relación con la hora de retirada, bastantes de ellos sí lo estaban. Por lo tanto, se pudo concluir que, en general, el motivo de retirada estaba relacionado con la hora.

SEGUNDA HIPÓTESIS

En el Jupyter **'segunda_hipo.ipynb'**, se demostró que existe relación entre **el tiempo** que un vehículo pasa en el depósito y el **tipo de vehículo** del que se trate.
Al igual que en los otros notebooks, este también explica sus pasos detalladamente. Sin embargo, aquí presento un resumen:
Lo primero fue cargar los dataframes que había creado en el Jupyter **'preparacion_datos_salida.ipynb'.**
En la celda **[6]**, volqué en una variable la media de tiempo que un vehículo pasa en el depósito. Luego, en las celdas **[7]** y **[8]**, creé un dataframe con una columna del tiempo que de media pasa cada tipo de vehículo en formato datatime y otra en formato float.
Con esto, y usando las bibliotecas **matplotlib** para trazar la figura, **seaborn** para estilizar el gráfico de barras y **format** para redondear los decimales de la media de los días, creé en la celda **[11]** una **gráfica de barras** para ver cuánto pasaba en el depósito, de media, cada tipo de vehículo, y su comparación con la media.
Al igual que con la hora de retirada y el motivo, para este caso también hacimos una prueba ANOVA. Para ello, en la celda **[13]**, preparé un dataframe **solo con las columnas necesarias:** tipoVehiculo, TiempoEnDeposito (días) y Tiempo en depósito (float), y en la celda **[15]** realicé la prueba.
El valor de p es casi 0 (1.2581579809780119e-139), lo que sugiere una alta significancia estadística y el rechazo de la hipótesis nula. Además, la estadística F (95.28800497120615) es mucho mayor que 1, indicando una diferencia significativa en los tiempos de permanencia entre al menos dos tipos de vehículos en el depósito.

> En este caso tambien hice la prueba de **Tukey**. Los resultados de lesta muestraro que hay diferencias significativas en el tiempo que pasan en el depósito entre los diferentes tipos de vehículos. Esto puede sugerir que el tipo de vehículo influye en el tiempo que pasa en el depósito, por lo que podemos decir que la hipótesis se cumple.
>
> - Se encontraron diferencias significativas entre el grupo de **Bicicletas** y todos los otros tipos de vehículos.
> - También hubo diferencias significativas entre algunos otros grupos:
>   **Camión** y **Coche**
>   **Camión** y **Patinetes (VMP)**
>   **Coche** y **Motocicleta**
>   **Motocicleta** y **Patinetes (VMP)**

- No hubo diferencias significativas entre:
  Bicicletas y **Remolque**
  Camión y **Motocarro**
  Camión y **Motocicleta**
  Coche y **Motocarro**
  Coche y **Patinetes (VMP)**
  Motocarro y **Patinetes (VMP)**
  Motocarro y **Vehículo pesado**
  Motocicleta y **Remolque**
  Motocicleta y **Vehículo pesado**
  Patinetes (VMP) y **Vehículo pesado**.

En resumen, el análisis muestra qué hay algunos tipos de vejiculos que tienen **diferencias significativas** en sus **medias** aunque algunos no.

También, al final del Jupyter, imprimí unos gráficos de cajas (boxplot), los cuales eran difíciles de leer debido a la gran cantidad de valores extremos, por lo que no tenía mucho sentido mostrarlos en la presentación. Sin embargo, se observa que, en general, las cajas son muy estrechas, excepto en el caso del remolque, y que la mediana está algo desplazada hacia abajo. También hay muchos valores extremos, lo cual dificulta la lectura de la caja. Teniendo en cuenta la gran cantidad de datos con los que está hecho este estudio, esto es algo normal.

Esto se puede interpretar de la siguiente manera:

- **Mediana desplazada hacia abajo:** La mediana está más cerca de los valores más bajos de los datos en comparación con la media. Esto puede indicar que hay algunos valores extremadamente bajos que están afectando la mediana, pero no afectan tanto a la media.
- **Muchos puntos por encima de la caja:** Esto sugiere una concentración de datos por encima de la mediana. Puede indicar que hay una cantidad significativa de valores altos que están inflando la mediana, pero no están influyendo tanto en la media.
- **Caja estrecha:** Indica que la dispersión de los datos (la variabilidad) dentro del rango intercuartílico es baja. Esto podría ser debido a que la mayoría de los datos se concentran en un rango relativamente estrecho alrededor de la mediana, pero hay algunos valores extremos que están extendiendo los bigotes.

FIGURA FINAL

La figura final es un scatterplot, un gráfico de dispersión, en el que, utilizando las coordenadas de los distintos depósitos (las cuales están en el archivo “300227-0-grua-depositos.csv”) y la cantidad de vehículos recibidos por cada depósito, coloqué los depósitos sobre una imagen georreferenciada de Madrid. la cual esta tomada de aquí, usando el mismo sistema de coordenadas que utiliza Google. Todo está explicado en el jupyter "depositos.ipynb".