



Big Data Aplicado

Profesor/a: José Manuel González Rodríguez

ACTIVIDAD EVALUABLE 1.2 (PROCESO ETL)
02/11/2024

Índice

1.- Introducción al Proceso ETL	4
1.1.- Descripción del Objetivo y Contexto de la Actividad	4
1.2.- Explicación Breve del Proceso ETL y su Relevancia	4
2.- Carga y Exploración del Conjunto de Datos	5
2.1.- Carga del Dataset (imports-85.data)	5
2.2.- Asignación de Nombres de Variables	6
2.3.- Exploración Inicial: Estructura, Filas y Columnas	7
3.- Limpieza de Datos	8
3.1.- Detección de Valores Perdidos	8
3.2.- Reporte de Valores Nulos por Columna	9
4.- Imputación de Valores Perdidos	9
4.1.- Imputación en la Columna precio	10
4.2.- Imputación en rpm-máxima y caballos-fuerza	10
4.3.- Imputación en Otras Columnas	11
5.- Filtrado de Datos	12
5.1.- Filtrado de Columnas con Valores Perdidos	12
5.2.- Filtrado por Altura de Coches	13
6.- Discretización	14
6.1.- Discretización de Tamaño-motor	14
6.2.- Discretización de distancia-ejes	15
6.3.- Discretización de peso-vacío	15
7.- Valores Anómalos	16
7.1.- Detección de Valores Anómalos	17
7.2.- Análisis de Valores Anómalos	18
8.- Exportación del Conjunto de Datos Transformado	19
8.1.- Exportación a CSV	19
8.2.- Validación de la Exportación	20
9.- Conclusiones	20
9.1.- Importancia del Proceso ETL	21
9.2.- Efectividad de las Técnicas de Limpieza	21
9.3.- Detección de Valores Anómalos	21
9.4.- Preparación para el Análisis Futuro	21

10.- Bibliografía	22
11.- Opinión Personal	22
11.1.- Aprendizajes sobre el Proceso ETL	23
11.2.- Reflexión sobre el Uso de R	23
11.3.- Relevancia en el Mundo Actual	23
11.4.- Futuras Mejoras	23
12.- Esquema resumido	24
13.- Anexo	24

1.- Introducción al Proceso ETL

1.1.- Descripción del Objetivo y Contexto de la Actividad

En esta actividad, el objetivo principal es simular un proceso de Extracción, Transformación y Carga de datos (ETL, por sus siglas en inglés) utilizando el lenguaje de programación R. A partir de un conjunto de datos de automóviles (archivo [imports-85.data](#)), se realizan varias operaciones de limpieza y preparación de los datos para obtener un archivo final en formato CSV, que esté listo para su análisis y aplicación en modelos de Machine Learning o reportes.

Este conjunto de datos de automóviles no contiene nombres de variables, lo que implica un paso inicial de configuración manual de columnas. A través de este proceso, se aplicarán técnicas de filtrado, imputación de valores, discretización y detección de valores anómalos. Estas operaciones permitirán transformar el dataset en un formato más manejable y completo, donde los datos sean consistentes y estén listos para su análisis.

1.2.- Explicación Breve del Proceso ETL y su Relevancia

El proceso ETL es un componente esencial en cualquier flujo de datos, especialmente en entornos de Big Data y Ciencia de Datos. Consiste en tres etapas fundamentales:

- **Extracción (Extract):** En esta fase, los datos se obtienen de diferentes fuentes. Estos pueden provenir de bases de datos, archivos de texto, APIs, u otros sistemas. En nuestro caso, se extraen del archivo [imports-85.data](#).
- **Transformación (Transform):** Aquí, los datos son limpiados y transformados para adaptarse a los requisitos del análisis. Este paso incluye la detección y corrección de valores nulos, la normalización de datos, la conversión de tipos de datos, y cualquier otro proceso de preparación. La transformación es crucial, ya que asegura que los datos sean precisos, completos y consistentes, minimizando posibles errores en el análisis posterior.
- **Carga (Load):** En la etapa final, los datos transformados se almacenan en un destino final, como una base de datos o un archivo, en un formato adecuado para su uso. Para esta actividad, el resultado se exportará a un archivo CSV.

El proceso ETL es relevante porque garantiza que los datos sean precisos y completos antes de ser analizados. Esta preparación inicial permite que las herramientas de análisis, como los modelos de Machine Learning, produzcan resultados fiables y precisos, reduciendo errores y ahorrando tiempo en etapas posteriores del proyecto.

2.- Carga y Exploración del Conjunto de Datos

2.1.- Carga del Dataset ([imports-85.data](#))

El primer paso en el proceso ETL es la **extracción** de datos, donde el conjunto de datos `imports-85.data` se carga en un DataFrame en R. Este archivo contiene únicamente los valores de las variables, sin nombres de columnas, por lo que es necesario asignarlos manualmente. Los nombres de las variables son cruciales, ya que permiten identificar cada atributo y manipularlos durante el proceso de transformación. Este conjunto de datos abarca información detallada sobre diversos aspectos de los automóviles, como su precio, tamaño del motor, potencia, entre otros.

Para cargar el archivo, se usa la función `read.table()` en R, que permite leer archivos de datos delimitados. Los datos son delimitados por comas, por lo que el parámetro `sep` se especifica como `" , "`. Además, al carecer de nombres de columnas, se debe definir manualmente un vector con los nombres de cada variable.

The screenshot displays the RStudio interface. The main editor window shows a data frame named 'datos' with 205 observations and 26 variables. The variables are labeled V1 through V26. The console window shows the following R code being executed:

```
> datos <- read.table("D:/automobile/imports-85.data", sep = ",", header = FALSE, na.strings = "?")
> view(datos)
> view(datos)
> |
```

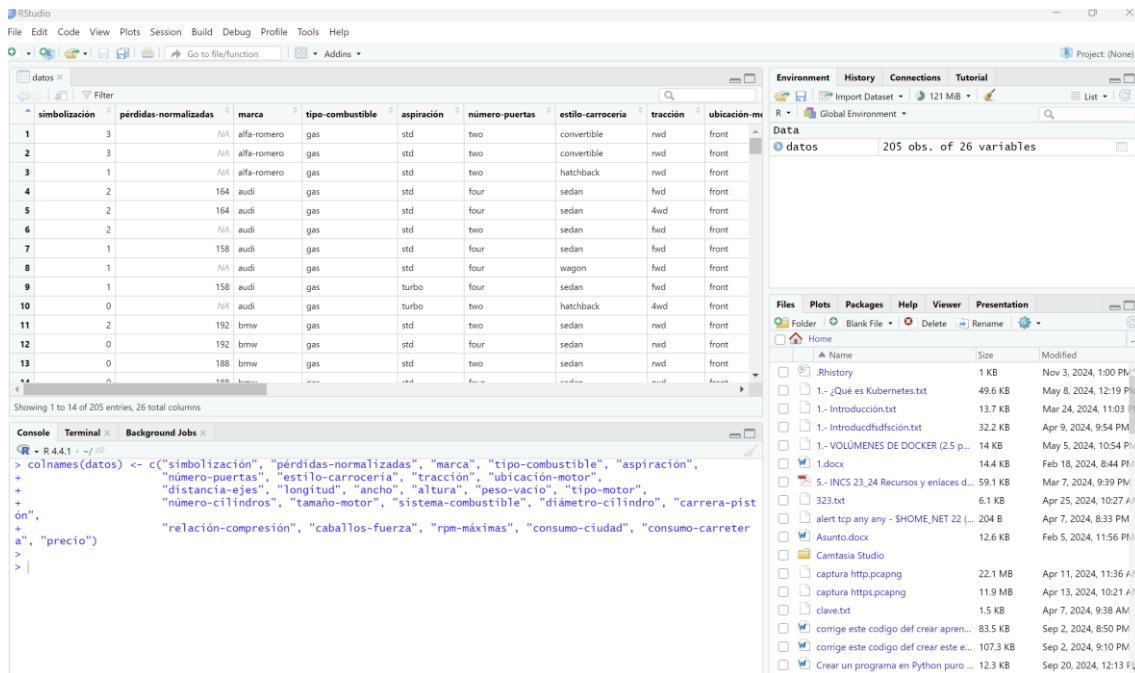
The Environment pane on the right shows the 'datos' object with 205 observations and 26 variables. The Files pane on the bottom right shows the file explorer with various files and folders.

```
datos <- read.table("D:/automobile/imports-85.data", sep = ",", header = FALSE, na.strings = "?")
```

Este código carga un archivo CSV llamado "imports-85.data" desde la carpeta "automobile" en el disco D en un data frame llamado `datos`, especificando que los valores están separados por comas, no hay encabezados y que los signos de interrogación representan valores faltantes.

2.2.- Asignación de Nombres de Variables

La asignación de nombres de variables se realiza para mejorar la legibilidad y manejo del conjunto de datos. A continuación, se listan los nombres de las variables que se incluyen en el archivo `imports-85.data`, asignados en el mismo orden en que aparecen en el archivo:



The screenshot shows the RStudio interface. The Environment pane on the right displays the 'datos' data frame with 205 observations and 26 variables. The Files pane on the right shows a list of files in the 'Home' directory. The Console pane at the bottom shows the command to assign column names to the 'datos' data frame.

```
colnames(datos) <- c("simbolización", "pérdidas-normalizadas", "marca", "tipo-combustible", "aspiración",  
+ "número-puertas", "estilo-carrocería", "tracción", "ubicación-motor",  
+ "distancia-ejes", "longitud", "ancho", "altura", "peso-vacio", "tipo-motor",  
+ "número-cilindros", "tamaño-motor", "sistema-combustible", "diámetro-cilindro", "carrera-pistón",  
+ "relación-compresión", "caballos-fuerza", "rpm-máximas", "consumo-ciudad", "consumo-carretera", "precio")
```

```
colnames(datos) <- c("simbolización", "pérdidas-normalizadas", "marca", "tipo-combustible", "aspiración",  
+ "número-puertas", "estilo-carrocería", "tracción", "ubicación-motor",  
+ "distancia-ejes", "longitud", "ancho", "altura", "peso-vacio", "tipo-motor",  
+ "número-cilindros", "tamaño-motor", "sistema-combustible", "diámetro-cilindro", "carrera-pistón",  
+ "relación-compresión", "caballos-fuerza", "rpm-máximas", "consumo-ciudad", "consumo-carretera", "precio")
```

Este código asigna nombres a las columnas del data frame 'datos', proporcionando etiquetas descriptivas para cada variable del conjunto de datos de automóviles.

2.3.- Exploración Inicial: Estructura, Filas y Columnas

Una vez cargado el conjunto de datos, se realiza una exploración inicial para conocer su estructura, tipos de datos, y dimensiones. Este paso es fundamental para entender qué transformaciones adicionales podrían requerirse y para identificar los tipos de datos con los que se está trabajando.

1.- Visualización de las Primeras Filas: Para observar una muestra inicial del conjunto de datos y familiarizarse con la disposición de los valores, se emplea la función `head(datos)`, que muestra las primeras filas del DataFrame.

```
Console Terminal Background Jobs
R 4.4.1 ~ /
> head(datos)
  simbolización pérdidas-normalizadas marca tipo-combustible aspiración número-puertas estilo-carrocería tracción
1             3                    NA alfa-romero          gas          std             two      convertible rwd
2             3                    NA alfa-romero          gas          std             two      convertible rwd
3             1                    NA alfa-romero          gas          std             two      hatchback  rwd
4             2                   164      audi          gas          std             four      sedan      fwd
5             2                   164      audi          gas          std             four      sedan      fwd
6             2                    NA      audi          gas          std             two      sedan      fwd

  ubicación-motor distancia-ejes longitud ancho altura peso-vacio tipo-motor número-cilindros tamaño-motor sistema-combustible
1      front      88.6      168.8  64.1  48.8      2548      dohc          four          130      mpfi
2      front      88.6      168.8  64.1  48.8      2548      dohc          four          130      mpfi
3      front      94.5      171.2  65.5  52.4      2823      ohcv          six          152      mpfi
4      front      99.8      176.6  66.2  54.3      2337      ohc          four          109      mpfi
5      front      99.4      176.6  66.4  54.3      2824      ohc          five          136      mpfi
6      front      99.8      177.3  66.3  53.1      2507      ohc          five          136      mpfi

  diámetro-cilindro carrera-pistón relación-compresión caballos-fuerza rpm-máximas consumo-ciudad consumo-carretera precio
1             3.47             2.68              9.0              111          5000          21          27      13495
2             3.47             2.68              9.0              111          5000          21          27      16500
3             2.68             3.47              9.0              154          5000          19          26      16500
4             3.19             3.40              10.0             102          5500          24          30      13950
5             3.19             3.40              8.0              115          5500          18          22      17450
6             3.19             3.40              8.5              110          5500          19          25      15250
>
```

2.- Verificación de la Estructura: La función `str(datos)` permite revisar los tipos de datos y el número de filas y columnas, lo cual es útil para determinar si alguna columna necesita ser convertida a otro tipo de dato.

```
Console Terminal Background Jobs
R 4.4.1 ~ /
> str(datos)
'data.frame': 205 obs. of 26 variables:
 $ simbolización      : int  3 3 1 2 2 2 1 1 1 0 ...
 $ pérdidas-normalizadas: int  NA NA NA 164 164 NA 158 NA 158 NA ...
 $ marca              : chr   "alfa-romero" "alfa-romero" "alfa-romero" "audi" ...
 $ tipo-combustible    : chr   "gas" "gas" "gas" "gas" ...
 $ aspiración          : chr   "std" "std" "std" "std" ...
 $ número-puertas      : chr   "two" "two" "two" "four" ...
 $ estilo-carrocería   : chr   "convertible" "convertible" "hatchback" "sedan" ...
 $ tracción            : chr   "rwd" "rwd" "rwd" "fwd" ...
 $ ubicación-motor     : chr   "front" "front" "front" "front" ...
 $ distancia-ejes      : num   88.6 88.6 94.5 99.8 99.4 ...
 $ longitud            : num   169 169 171 177 177 ...
 $ ancho               : num   64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 67.9 ...
 $ altura              : num   48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 ...
 $ peso-vacio          : int    2548 2548 2823 2337 2824 2507 2844 2954 3086 3053 ...
 $ tipo-motor          : chr    "dohc" "dohc" "ohcv" "ohc" ...
 $ número-cilindros    : chr    "four" "four" "six" "four" ...
 $ tamaño-motor        : int    130 130 152 109 136 136 136 136 131 131 ...
 $ sistema-combustible : chr    "mpfi" "mpfi" "mpfi" "mpfi" ...
 $ diámetro-cilindro   : num    3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.13 ...
 $ carrera-pistón      : num    2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 3.4 ...
 $ relación-compresión : num    9 9 10 8 8.5 8.5 8.5 8.3 7 ...
 $ caballos-fuerza     : int    111 111 154 102 115 110 110 110 140 160 ...
 $ rpm-máximas         : int    5000 5000 5000 5500 5500 5500 5500 5500 5500 ...
 $ consumo-ciudad      : int    21 21 19 24 18 19 19 19 17 16 ...
 $ consumo-carretera   : int    27 27 26 30 22 25 25 25 20 22 ...
 $ precio              : int   13495 16500 16500 13950 17450 15250 17710 18920 23875 NA ...
>
```

3.- Dimensiones del Dataset: Con `dim(datos)`, se obtiene el número de filas y columnas del conjunto de datos, lo que permite conocer el tamaño total del archivo cargado.

```
Console Terminal Background Jobs
R - R 4.4.1 - ~/
> dim(datos)
[1] 205 26
> |
```

Esta exploración proporciona una comprensión preliminar del dataset, lo cual es esencial antes de aplicar cualquier técnica de transformación.

3.- Limpieza de Datos

3.1.- Detección de Valores Perdidos

La limpieza de datos es una etapa clave dentro del proceso ETL, ya que permite asegurar la calidad y consistencia de los datos antes de analizarlos. Un paso esencial en esta fase es la **detección de valores perdidos** o valores nulos, que representan datos ausentes en las observaciones. Los valores perdidos pueden ocurrir por distintas razones, como errores en la captura de datos o valores desconocidos en el momento de recolección. Detectarlos y tratarlos adecuadamente evita problemas en etapas posteriores del análisis.

Para identificar los valores perdidos en cada columna del DataFrame en R, se puede usar la función `is.na()` junto con `colSums()` para contar el número de valores nulos por columna. Este método permite obtener un resumen detallado de cuántos valores faltan en cada variable, lo que facilita el diseño de estrategias específicas de imputación o eliminación.

```
Console Terminal Background Jobs
R - R 4.4.1 - ~/
> colSums(is.na(datos))
simbolización pérdidas-normalizadas marca tipo-combustible aspiración
0 41 0 0 0
número-puertas estilo-carrocería tracción ubicación-motor distancia-ejes
2 0 0 0 0
longitud ancho altura peso-vacio tipo-motor
0 0 0 0 0
número-cilindros tamaño-motor sistema-combustible diámetro-cilindro carrera-pistón
0 0 0 4 4
relación-compresión caballos-fuerza rpm-máximas consumo-ciudad consumo-carretera
0 2 2 0 0
precio
4
```

Este código muestra el número de valores nulos para cada columna del conjunto de `datos`. En el contexto de este dataset, es importante realizar este análisis de manera detallada, ya que algunas columnas pueden tener un número considerable de datos faltantes, lo

que afecta la precisión del análisis y de cualquier modelo de Machine Learning que utilice estos datos.

3.2.- Reporte de Valores Nulos por Columna

El reporte de valores nulos facilita la toma de decisiones para la imputación o eliminación de filas y columnas. En el caso de este conjunto de datos, el reporte ayuda a identificar qué columnas requieren una estrategia de imputación específica, como reemplazar valores perdidos con la media, la mediana, o un valor constante.

Un ejemplo de salida podría ser el siguiente:

Campo	Valores_Perdidos
pérdidas-normalizadas	41
diámetro-cilindro	4
carrera-pistón	4
precio	4
número-puertas	2
caballos-fuerza	2
rpm-máximas	2

Este reporte permite visualizar las columnas que necesitan ser tratadas y ofrece una base para decidir la técnica de imputación más adecuada, que será realizada en la siguiente sección.

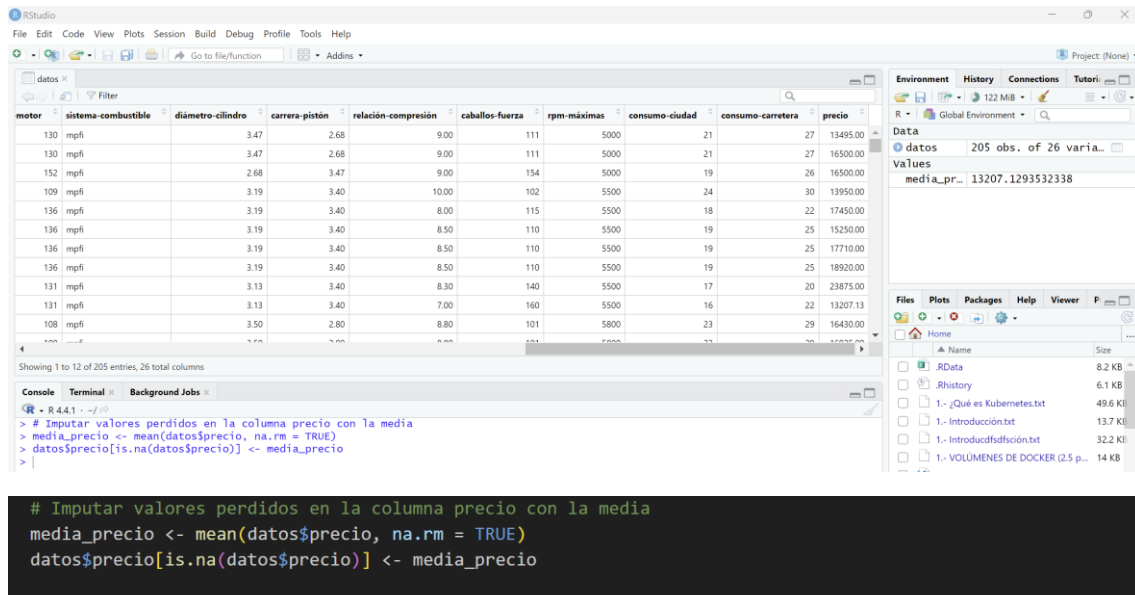
Este desarrollo proporciona una visión clara sobre cómo identificar y reportar los valores nulos en el conjunto de datos antes de aplicar técnicas de imputación específicas.

4.- Imputación de Valores Perdidos

La imputación de valores perdidos es una técnica utilizada para sustituir valores nulos o faltantes en el conjunto de datos, asegurando así la completitud del dataset y mejorando su calidad para el análisis. Para esta actividad, se aplican distintas estrategias de imputación en función de las características y necesidades de cada columna.

4.1.- Imputación en la Columna `precio`

Para la columna `precio`, que representa el costo del automóvil, se utilizará la media aritmética de los valores existentes para reemplazar los valores perdidos. Esta técnica es útil en el caso de variables numéricas, ya que mantiene el valor promedio del conjunto de datos sin afectar en exceso su distribución.



The screenshot shows the RStudio interface. The main window displays a data frame with columns: `motor`, `sistema-combustible`, `diámetro-cilindro`, `carrera-pistón`, `relación-compresión`, `caballos-fuerza`, `rpm-máximas`, `consumo-ciudad`, `consumo-carretera`, and `precio`. The `precio` column contains several NA values. The console shows the following R code:

```
# Imputar valores perdidos en la columna precio con la media
media_precio <- mean(datos$precio, na.rm = TRUE)
datos$precio[is.na(datos$precio)] <- media_precio
```

The Environment pane on the right shows the `datos` object with 205 observations and 26 variables. The Files pane at the bottom right shows the project structure.

En este código, primero se calcula la media de la columna `precio`, omitiendo los valores nulos (`na.rm = TRUE`), y luego se sustituyen los valores faltantes con dicha media.

4.2.- Imputación en `rpm-máxima` y `caballos-fuerza`

Para las columnas `rpm-máxima` y `caballos-fuerza`, se utilizan valores constantes, de 5000 y 120 respectivamente. Esta estrategia es útil cuando los valores específicos son representativos o apropiados en función de los datos o el contexto. En este caso, estos valores permiten mantener la consistencia de los datos en lugar de eliminar las observaciones incompletas.

The screenshot shows the RStudio interface. The Environment pane on the right shows a data frame 'datos' with 205 observations and 26 variables. The 'Values' section for 'datos' shows 'media_pr...' with the value 13207.1293532338. The Console pane at the bottom shows the following R code:

```
# Imputar valores perdidos en rpm-máximas y caballos-fuerza
> datos$`rpm-máximas`[is.na(datos$`rpm-máximas`)] <- 5000
> datos$`caballos-fuerza`[is.na(datos$`caballos-fuerza`)] <- 120
```

Below the console, a code block shows the same code in a dark background:

```
# Imputar valores perdidos en rpm-máximas y caballos-fuerza
datos$`rpm-máximas`[is.na(datos$`rpm-máximas`)] <- 5000
datos$`caballos-fuerza`[is.na(datos$`caballos-fuerza`)] <- 120
```

Este bloque de código establece el valor de 5000 en los valores perdidos de `rpm-máximas` y 120 en `caballos-fuerza`.

4.3.- Imputación en Otras Columnas

Para el resto de columnas con valores perdidos, excluyendo `diámetro-cilindro` y `carrera-pistón`, se puede emplear un método de imputación visto en clase, como la **mediana** o algún **algoritmo de imputación** (por ejemplo, el método K-Nearest Neighbors o KNN). La mediana es una técnica robusta en presencia de valores atípicos, ya que no se ve afectada por estos.

A continuación, se realiza una imputación usando la mediana para las columnas restantes con valores nulos.

The screenshot shows the RStudio interface. The Environment pane on the right shows the 'datos' data frame with 205 observations and 26 variables. The 'Values' section for 'datos' shows 'media_pr...' with the value 13207.1293532338. The Console pane at the bottom shows the following R code:

```
# Imputación con la mediana para otras columnas con valores perdidos (excluyendo diámetro-cilindro y carrera-pistón)
> columnas_a_imputar <- c("pérdidas-normalizadas", "número-puertas") # incluir otras columnas con valores nulos si es necesario
> for (col in columnas_a_imputar) {
+   mediana <- median(datos[[col]], na.rm = TRUE)
+   datos[[col]][is.na(datos[[col]])] <- mediana
+ }
>
```

```
# Imputación con la mediana para otras columnas con valores perdidos (excluyendo diámetro-cilindro y carrera-pistón)
columnas_a_imputar <- c("pérdidas-normalizadas", "número-puertas") # incluir otras columnas con valores nulos si es necesario
for (col in columnas_a_imputar) {
  mediana <- median(datos[[col]], na.rm = TRUE)
  datos[[col]][is.na(datos[[col]])] <- mediana
}
```

En este código, se recorre cada columna en `columnas_a_imputar` y se reemplazan los valores nulos con la mediana correspondiente.

Con estas técnicas de imputación, el conjunto de datos se completa y se garantiza que no haya valores nulos en las columnas de interés. Esto facilita la continuidad del proceso ETL, permitiendo pasar a la fase de filtrado sin perder información relevante.

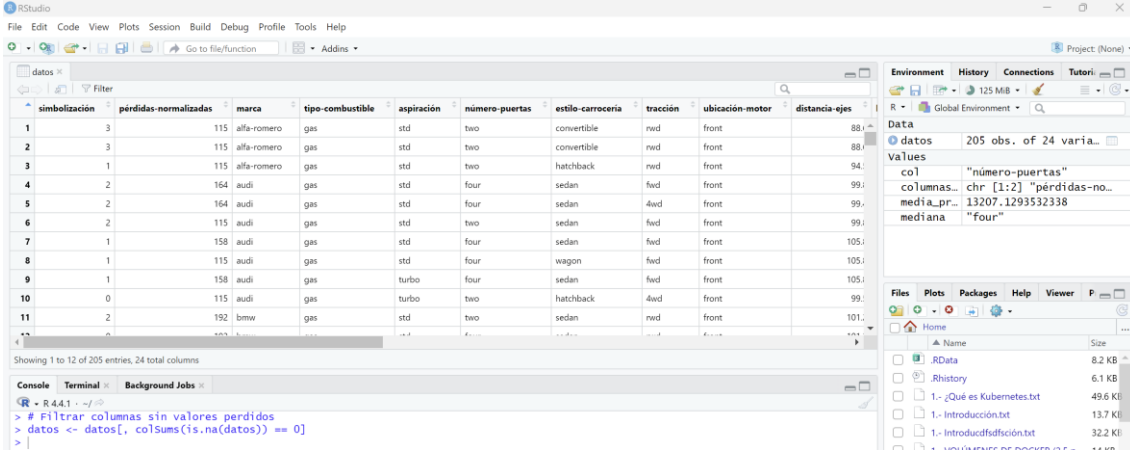
5.- Filtrado de Datos

El filtrado de datos es una técnica importante dentro de la etapa de transformación en el proceso ETL. Este proceso implica eliminar filas o columnas según criterios específicos para obtener un conjunto de datos más limpio y representativo. En esta actividad, se aplicarán dos tipos de filtrado: (1) filtrado de columnas que contienen valores perdidos y (2) filtrado de filas según una condición de altura de los automóviles.

5.1.- Filtrado de Columnas con Valores Perdidos

El primer paso es eliminar aquellas columnas que aún contengan valores perdidos, lo cual asegura que el conjunto de datos final esté completo en cuanto a información. Este paso es especialmente útil cuando la cantidad de valores perdidos es alta y no se justifica realizar imputaciones.

En R, se puede identificar y eliminar columnas con valores perdidos utilizando la función `colSums()` junto con `is.na()`, y seleccionando sólo las columnas donde el número de valores nulos es cero.



The screenshot shows the RStudio interface. The Environment pane on the right shows a data frame 'datos' with 205 observations and 24 variables. The console shows the following R code:

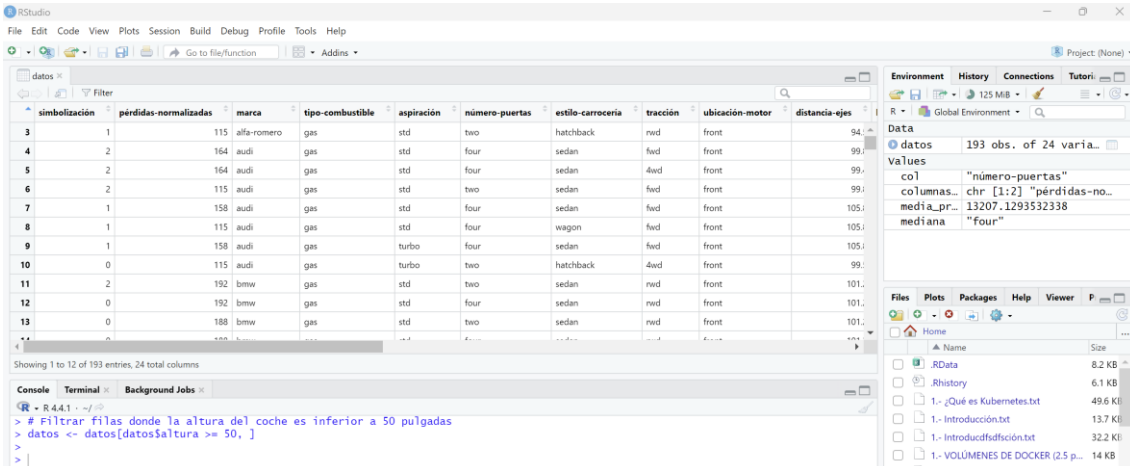
```
# Filtrar columnas sin valores perdidos
datos <- datos[, colSums(is.na(datos)) == 0]
```

The data frame 'datos' has the following columns: simbolización, pérdidas-normalizadas, marca, tipo-combustible, aspiración, número-puertas, estilo-carrocería, tracción, ubicación-motor, and distancia-ejes.

Este código selecciona únicamente las columnas que no tienen valores nulos, manteniendo así un conjunto de datos limpio.

5.2.- Filtrado por Altura de Coches

El siguiente paso en el filtrado consiste en eliminar aquellos registros donde la altura del automóvil sea inferior a 50 pulgadas. Esto puede ser útil para excluir vehículos con características fuera del rango esperado, lo cual podría representar datos atípicos o errores en el registro.



The screenshot shows the RStudio interface. The Environment pane on the right shows a data frame 'datos' with 193 observations and 24 variables. The console shows the following R code:

```
# Filtrar filas donde la altura del coche es inferior a 50 pulgadas
datos <- datos[datos$altura >= 50, ]
```

The data frame 'datos' has the following columns: simbolización, pérdidas-normalizadas, marca, tipo-combustible, aspiración, número-puertas, estilo-carrocería, tracción, ubicación-motor, and distancia-ejes.

En este código, se seleccionan únicamente las filas donde el valor de la columna `altura` es mayor o igual a 50, eliminando las observaciones que no cumplen con esta condición.

Con estos pasos de filtrado, se obtiene un conjunto de datos más limpio y adecuado para análisis posteriores. Este proceso asegura que sólo se incluyan datos completos y que cumplen con ciertos criterios, mejorando la calidad del conjunto final para su exportación y análisis.

6.- Discretización

La discretización es el proceso de convertir variables continuas en variables categóricas dividiéndolas en intervalos o rangos. Esto facilita el análisis de ciertos datos y es útil en algoritmos de aprendizaje automático que funcionan mejor con variables categóricas. En esta actividad, discretizaremos tres variables: `tamaño-motor`, `distancia-ejes`, y `peso-vacio`.

6.1.- Discretización de `Tamaño-motor`

Para la variable `tamaño-motor` (tamaño del motor), se puede dividir en categorías como "pequeño", "mediano" y "grande". Esto permite clasificar los motores según su tamaño y hacer comparaciones de manera más sencilla.

En R, se utiliza la función `cut()` para dividir los valores en intervalos.

The screenshot shows the RStudio interface. The 'Environment' pane on the right shows the 'datos' dataset with 193 observations and 25 variables. The 'Data' pane shows the first few rows of the dataset. The 'Console' pane at the bottom shows the R code used for discretization:

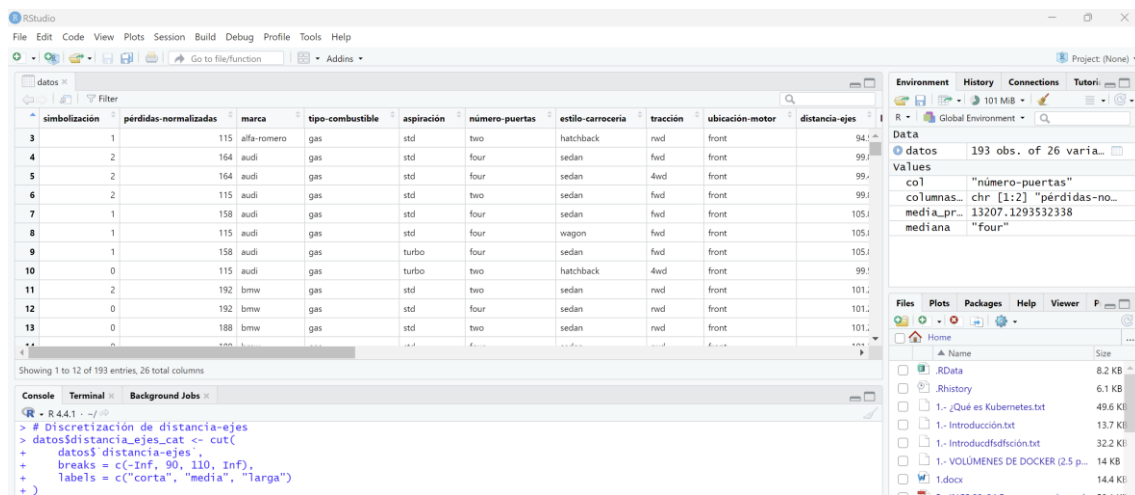
```
# Discretización de tamaño-motor
datos$tamaño_motor_cat <- cut(
  datos$tamaño_motor,
  breaks = c(-Inf, 100, 200, Inf),
  labels = c("pequeño", "mediano", "grande")
)
```

```
# Discretización de tamaño-motor
datos$tamaño_motor_cat <- cut(
  datos$tamaño_motor,
  breaks = c(-Inf, 100, 200, Inf),
  labels = c("pequeño", "mediano", "grande")
)
```

Aquí, los valores de `tamaño-motor` menores o iguales a 100 se etiquetan como "pequeño", los que están entre 100 y 200 como "mediano", y los superiores a 200 como "grande".

6.2.- Discretización de `distancia-ejes`

La variable `distancia-ejes` (distancia entre ejes) también se puede discretizar en categorías como "corto", "medio" y "largo". Esto permite agrupar los autos según la distancia entre ejes y observar cómo este factor influye en otras características.



The screenshot shows the RStudio interface. The 'Environment' pane on the right shows the 'datos' dataset with 193 observations and 26 variables. The 'Data' pane shows the structure of the dataset. The 'Console' pane at the bottom shows the following R code being executed:

```
# Discretización de distancia-ejes
datos$distancia_ejes_cat <- cut(
  datos$`distancia-ejes`,
  breaks = c(-Inf, 90, 110, Inf),
  labels = c("corta", "media", "larga")
)
```

```
# Discretización de distancia-ejes
datos$distancia_ejes_cat <- cut(
  datos$`distancia-ejes`,
  breaks = c(-Inf, 90, 110, Inf),
  labels = c("corta", "media", "larga")
)
```

En este código, los valores de `distancia-ejes` menores o iguales a 90 se clasifican como "corta", entre 90 y 110 como "media", y mayores a 110 como "larga".

6.3.- Discretización de `peso-vacío`

Finalmente, para la variable `peso-vacío` (peso en vacío), se pueden establecer categorías como "ligero", "medio" y "pesado", lo cual ayuda a agrupar los vehículos en función de su peso y hacer análisis comparativos.

The screenshot shows the RStudio interface. The main window displays a data table with columns: simbolización, pérdidas-normalizadas, marca, tipo-combustible, aspiración, número-puertas, estilo-carrocería, tracción, ubicación-motor, and distancia-ejes. The table shows 13 rows of data. The console window shows the following R code:

```
# Discretización de peso-vacio
datos$peso_vacio_cat <- cut(
  datos$`peso-vacio`,
  breaks = c(-Inf, 2000, 3000, Inf),
  labels = c("ligero", "medio", "pesado")
)
```

The Environment pane on the right shows the 'datos' object with 193 observations and 27 variables. The Files pane shows the project files.

Aquí, los vehículos con un `peso-vacio` menor o igual a 2000 se etiquetan como "ligero", aquellos entre 2000 y 3000 como "medio", y los superiores a 3000 como "pesado".

Con estos pasos de discretización, las variables continuas `tamaño-motor`, `distancia-ejes` y `peso-vacio` se han transformado en variables categóricas. Esto facilita el análisis al agrupar datos en categorías significativas y permite aplicar ciertos algoritmos de análisis que requieren variables categóricas.

7.- Valores Anómalos

La detección de valores anómalos, o outliers, es crucial en el proceso ETL, ya que permite identificar datos que se desvían significativamente del resto de las observaciones. Estos valores extremos pueden deberse a errores en la recolección de datos, variabilidad natural de los datos o a eventos inusuales. Identificar y analizar estos valores es importante, ya que pueden influir negativamente en el análisis y en la precisión de los modelos predictivos.

En esta actividad, buscaremos valores anómalos en las variables continuas del conjunto de datos que aún no han sido discretizadas.

7.1.- Detección de Valores Anómalos

Uno de los métodos más comunes para identificar outliers es utilizar el **rango intercuartílico** (IQR, por sus siglas en inglés). Este método define los valores anómalos como aquellos que están fuera del rango $[Q1 - 1.5 \cdot IQR, Q3 + 1.5 \cdot IQR]$, donde $Q1$ y $Q3$ son el primer y tercer cuartil de la variable, respectivamente.

A continuación, se muestra cómo implementar esta técnica en R para una columna continua, por ejemplo `precio`.

The screenshot shows the RStudio interface. The Environment pane on the right shows a dataset named 'datos' with 193 observations and 27 variables. The Console pane at the bottom shows the following R code:

```
> # Función para detectar valores anómalos usando el rango intercuartílico (IQR)
> detectar_outliers <- function(x) {
+   Q1 <- quantile(x, 0.25, na.rm = TRUE)
+   Q3 <- quantile(x, 0.75, na.rm = TRUE)
+   IQR <- Q3 - Q1
+   outliers <- x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)
+   return(outliers)
+ }
>
> # Aplicar la función para detectar outliers en cada variable continua
> outliers_precio <- detectar_outliers(datos$precio)
> outliers_tamaño_motor <- detectar_outliers(datos$tamaño-motor)
> outliers_distancia_ejes <- detectar_outliers(datos$distancia-ejes)
> outliers_peso_vacio <- detectar_outliers(datos$peso-vacio)
```

```
# Función para detectar valores anómalos usando el rango intercuartílico (IQR)
detectar_outliers <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  outliers <- x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)
  return(outliers)
}
```

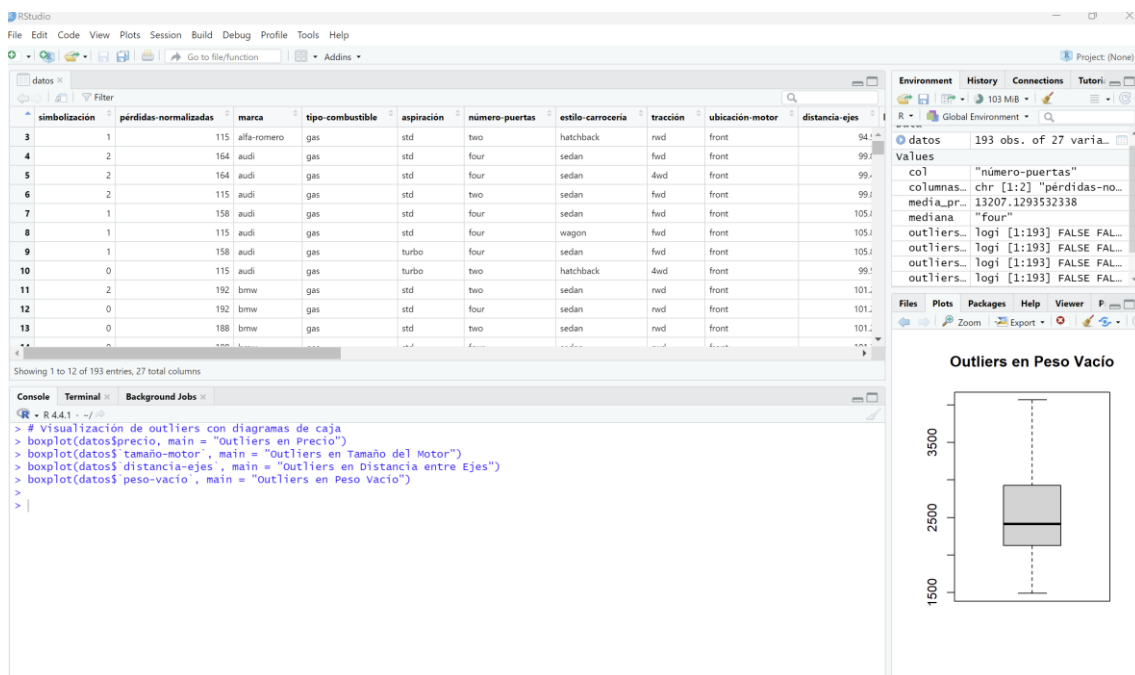
```
# Aplicar la función para detectar outliers en cada variable continua
outliers_precio <- detectar_outliers(datos$precio)
outliers_tamaño_motor <- detectar_outliers(datos$tamaño-motor)
outliers_distancia_ejes <- detectar_outliers(datos$distancia-ejes)
outliers_peso_vacio <- detectar_outliers(datos$peso-vacio)
```

En este código, la función `detectar_outliers` retorna un vector booleano que indica cuáles elementos son outliers en la variable correspondiente. Esta técnica se puede aplicar a cada variable continua para identificar y contar los valores anómalos.

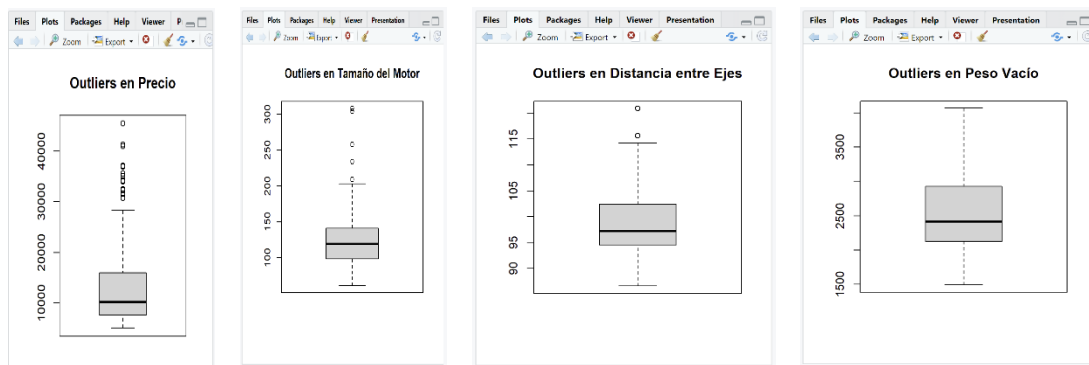
7.2.- Análisis de Valores Anómalos

Una vez identificados, es importante decidir qué hacer con estos valores anómalos. Dependiendo del contexto y del análisis que se quiera realizar, los outliers pueden eliminarse, mantenerse o ajustarse. Para esta actividad, se puede optar por visualizarlos y analizarlos antes de tomar una decisión.

Por ejemplo, podemos visualizar los outliers utilizando diagramas de caja (boxplots) para cada variable continua, lo cual facilita la interpretación gráfica de los valores extremos.



```
# Visualización de outliers con diagramas de caja
boxplot(datos$precio, main = "Outliers en Precio")
boxplot(datos$tamaño-motor, main = "Outliers en Tamaño del Motor")
boxplot(datos$distancia-ejes, main = "Outliers en Distancia entre Ejes")
boxplot(datos$peso-vacio, main = "Outliers en Peso Vacío")
```



Estos boxplots permiten identificar visualmente los outliers en cada variable continua. Basándonos en esta información, se puede decidir si eliminar los valores anómalos o mantenerlos según la relevancia que puedan tener para el análisis final.

Con estos pasos, la identificación de valores anómalos permite un conjunto de datos más controlado, donde los valores extremos han sido revisados y, si es necesario, tratados adecuadamente. Este proceso contribuye a la calidad del conjunto de datos para el análisis posterior.

8.- Exportación del Conjunto de Datos Transformado

La última etapa del proceso ETL es la **exportación** del conjunto de datos transformado. Esta fase consiste en guardar el dataset final en un archivo en formato **CSV** que pueda ser utilizado para análisis posteriores o compartido con otras personas. En este caso, el archivo de salida se guardará con el nombre `automoviles-XXX.csv`, donde `XXX` serán tus iniciales.

Esta exportación incluye todas las transformaciones realizadas previamente, como la imputación de valores perdidos, filtrado, discretización y tratamiento de valores anómalos.

8.1.- Exportación a CSV

En R, la función `write.csv()` permite guardar un `data.frame` en un archivo CSV. Esta función tiene varios argumentos que permiten definir el nombre del archivo, especificar si se incluyen nombres de filas y si el delimitador es una coma, entre otros.

```
# Exportar el conjunto de datos transformado a un archivo CSV
write.csv(datos, "D:/automobile/automoviles_pmga.csv", row.names = FALSE)
```

En este código:

- Se especifica el nombre del archivo como `automoviles-pmga.csv`, donde `XXX` debe ser reemplazado por tus iniciales.
- `row.names = FALSE` indica que no se incluyan nombres de filas en el archivo CSV, manteniéndolo limpio y en un formato estándar.

8.2.- Validación de la Exportación

Es una buena práctica verificar que el archivo se haya exportado correctamente y que contenga los datos transformados. Para ello, puedes leer nuevamente el archivo y observar las primeras filas para confirmar su contenido.

```
# Validación: Cargar el archivo exportado para revisar su contenido
datos_exportados <- read.csv("D:/automobile/automoviles_pmga.csv")
head(datos_exportados)
```

simbolización	pérdidas-normalizadas	marca	tipo-combustible	aspiración	número-puertas	estilo-carroceria	tracción	ubicación-motor
3	1	115	alfa-romero	gas	std	hatchback	rwd	front
4	2	164	audi	gas	std	sedan	fwd	front
5	2	164	audi	gas	std	sedan	4wd	front
6	2	115	audi	gas	std	sedan	fwd	front

```
# Validación: Cargar el archivo exportado para revisar su contenido
datos_exportados <- read.csv("D:/automobile/automoviles_pmga.csv")
head(datos_exportados)
```

Este paso permite asegurar que los datos se exportaron de manera correcta y que todas las transformaciones aplicadas durante el proceso ETL están reflejadas en el archivo final.

Con esta etapa de exportación, el proceso ETL concluye exitosamente. El archivo `automoviles-pmga.csv` ahora contiene un conjunto de datos limpio, transformado y listo para análisis o integración en otros sistemas de procesamiento de datos.

9.- Conclusiones

El proceso de ETL (Extracción, Transformación y Carga) es fundamental en el manejo y análisis de datos, ya que permite preparar y optimizar la información para su uso en diversas aplicaciones. En esta actividad, se ha llevado a cabo un proceso ETL completo utilizando un

conjunto de datos de automóviles, lo que ha permitido obtener varias conclusiones significativas.

9.1.- Importancia del Proceso ETL

La actividad ha demostrado la relevancia del proceso ETL en la limpieza y preparación de datos. La extracción de datos desde el archivo `imports-85.data` y la posterior transformación mediante técnicas de filtrado, imputación y discretización han sido esenciales para garantizar la calidad del conjunto de datos final. Sin un proceso adecuado de ETL, los análisis podrían verse afectados por datos incompletos o erróneos, llevando a conclusiones incorrectas.

9.2.- Efectividad de las Técnicas de Limpieza

Las técnicas implementadas, como la identificación y eliminación de valores perdidos, así como la imputación de datos, resultaron efectivas para asegurar la integridad del conjunto de datos. La discretización de variables continuas a categorías también facilita el análisis y la interpretación de los datos, permitiendo que se realicen comparaciones significativas entre diferentes grupos.

9.3.- Detección de Valores Anómalos

La detección y análisis de valores anómalos es un paso crucial que no debe pasarse por alto. La utilización del rango intercuartílico (IQR) para identificar outliers permitió reconocer posibles errores en los datos o casos extremos que podrían afectar el análisis. Esta práctica refuerza la necesidad de validar y examinar los datos antes de proceder a análisis más complejos.

9.4.- Preparación para el Análisis Futuro

La correcta ejecución del proceso ETL ha dejado un conjunto de datos limpio y estructurado, listo para ser utilizado en análisis posteriores. La exportación del dataset en formato CSV facilita su uso en otras herramientas de análisis de datos y permite compartirlo con otros investigadores o profesionales.

En resumen, esta actividad ha proporcionado una visión clara de cómo el proceso ETL es fundamental para el manejo efectivo de datos. La implementación de estas técnicas no solo mejora la calidad de los datos, sino que también optimiza el tiempo y recursos necesarios para obtener análisis significativos. La capacidad de trabajar con datos limpios y organizados es un recurso invaluable en cualquier proyecto que implique análisis de información.

10.- Bibliografía

La bibliografía incluye las referencias utilizadas para la realización de esta actividad, así como aquellas que proporcionan información adicional sobre el proceso ETL y el manejo de datos en R. Las siguientes fuentes son esenciales para entender los conceptos y técnicas aplicadas:

1. R Core Team (2024). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Disponible en: <https://www.r-project.org/>
2. Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data (2nd ed.). O'Reilly Media. Disponible en: <https://r4ds.had.co.nz/>
3. Wickham, H., François, R., Henry, L., & Müller, K. (2024). dplyr: A Grammar of Data Manipulation. Consultado el 3 de noviembre de 2024, desde <https://dplyr.tidyverse.org>
4. Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., & Woo, K. (2024). ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. Consultado el 3 de noviembre de 2024, desde <https://ggplot2.tidyverse.org>

11.- Opinión Personal

La actividad de realizar un proceso ETL utilizando R ha sido una experiencia altamente enriquecedora y educativa. A través de esta práctica, he podido apreciar la importancia de cada fase del proceso, desde la extracción de datos hasta su carga final en un formato utilizable.

11.1.- Aprendizajes sobre el Proceso ETL

Una de las enseñanzas más valiosas ha sido entender que el ETL no es solo un conjunto de técnicas, sino un enfoque integral para garantizar la calidad de los datos. La capacidad de identificar y manejar valores perdidos y anómalos es crucial en el análisis de datos. He aprendido que la limpieza y transformación de datos son pasos fundamentales que a menudo se pasan por alto, pero que tienen un impacto significativo en la calidad de los resultados de análisis posteriores.

11.2.- Reflexión sobre el Uso de R

El uso de R para este ejercicio ha sido particularmente efectivo, dado que la herramienta proporciona una amplia gama de funciones y paquetes que facilitan la manipulación y análisis de datos. La sintaxis y las funciones específicas de R, como `read.csv()`, `write.csv()`, y el uso de `dplyr` y `ggplot2` (que se podrían incluir en una actividad futura), permiten realizar tareas complejas de manera intuitiva y eficiente. Me ha gustado cómo R se adapta bien al manejo de conjuntos de datos grandes y cómo las visualizaciones pueden proporcionar información valiosa sobre los datos.

11.3.- Relevancia en el Mundo Actual

En el contexto actual, donde la toma de decisiones basada en datos es cada vez más prevalente, la habilidad de realizar un proceso ETL efectivo se vuelve esencial. Las organizaciones dependen de datos precisos y bien estructurados para informar sus estrategias y operaciones. Esta experiencia me ha motivado a seguir aprendiendo y perfeccionando mis habilidades en análisis de datos, ya que considero que será fundamental en mi futuro profesional.

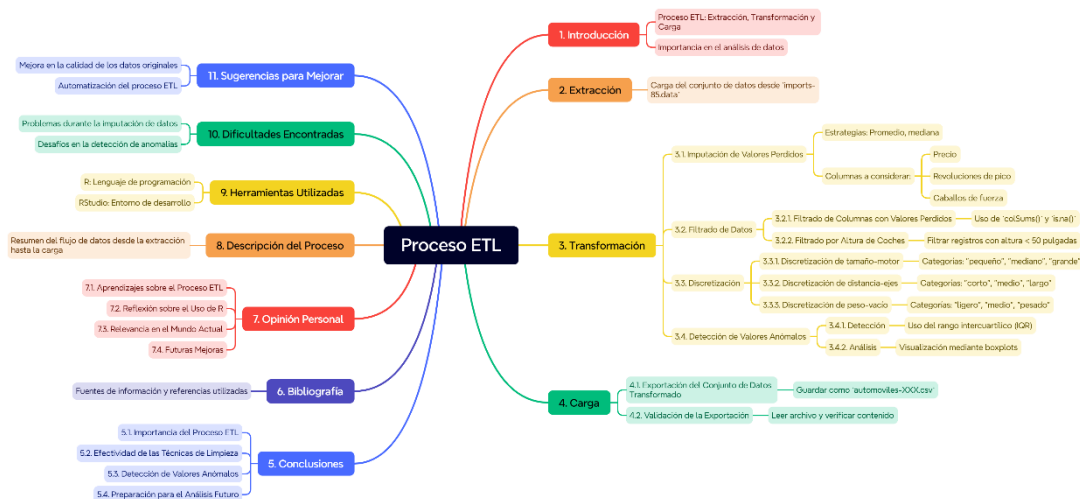
11.4.- Futuras Mejoras

De cara al futuro, me gustaría explorar más a fondo técnicas avanzadas de imputation y análisis de outliers. También estoy interesado en aprender sobre herramientas de visualización más avanzadas que se integren con R, así como en ampliar mis

conocimientos sobre otros lenguajes y herramientas de ETL. Además, trabajar con conjuntos de datos más grandes y complejos será un reto que espero asumir.

En resumen, esta actividad no solo me ha proporcionado habilidades técnicas, sino que también me ha dado una nueva apreciación por la importancia del manejo de datos en el mundo actual. Estoy entusiasmado por continuar mi aprendizaje en este campo y aplicar lo que he aprendido en futuros proyectos.

12.- Esquema resumido



El esquema resumido detalla el proceso ETL, que incluye la extracción, transformación y carga de datos, así como la imputación de valores perdidos, filtrado, discretización, detección de anomalías, y la importancia de este proceso en el análisis de datos, junto con conclusiones y reflexiones personales.

13.- Anexo

1. **Fichero Documentación (Actividad Evaluable 1.2 ETL_Pedro_Manuel_García_Alvarez.pdf)** : Este documento incluye una explicación detallada del proceso de ETL que he realizado. En él se describen los pasos seguidos, las decisiones tomadas y una explicación de cada etapa del análisis de datos.
2. **Fichero de datos corregido (automoviles_pmga.csv)** : Este fichero contiene el conjunto de datos de automóviles después de haber realizado la imputación de valores perdidos, el filtrado de

datos, la discretización de variables y la detección de valores anómalos. Está preparado para un análisis más efectivo y preciso.

3. **Fichero de esquema resumido (PDF) (Proceso ETL_Pedro_Manuel_García_Álvarez.pdf)** : Este documento presenta un esquema resumido que resume visualmente el proceso de ETL, mostrando las etapas, las técnicas utilizadas y las conclusiones. Este recurso facilita la comprensión del flujo del proceso y los resultados obtenidos.

Índice Alfabético

A

abarca	5
adicionales	7
algoritmo	11
altura	12, 13
amplia	23
análisis4, 8, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25	
aprendizaje	14, 24
Aprendizajes	23
aritmética	10
ausentes	8
automático	14
automóviles	4, 24
avanzadas	23
ayuda	9, 15

B

base	4, 9
bases	4
bibliografía	22
Big	4
bloque	11
booleano	17

C

cabo	20
caja	18
calidad	8, 9, 14, 19, 21, 22, 23
campo	24
cantidad	12
capacidad	22, 23
captura	8
cara	23
características	9, 13
Carga	4, 5
caso	9, 10
categorías	14, 15, 16, 21
categorías	14
Ciencia	4
cilindro	11
clara	9, 22
clave	8
código	6, 8, 11, 13
columnas	6, 8, 9, 10, 11, 12, 13
complejas	23
completitud	9
comprensión	8, 25

comunes	17
conclusiones	21, 24
condición	12
configuración	4
conjunto ... 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 16, 19, 20, 21, 23, 24	
consistencia	8, 10
consistentes	4
continua	17
continuas	14, 16, 21
continuidad	12
conversión	4
corrección	4
costo	10
Create	22
CSV	6, 19, 21

D

Data	4, 22
DataFrame	5, 8
Dataset	5
decisiones	9, 23, 24
desarrollo	9
Descripción	4
descriptivas	6
destino	4
detección	4, 8, 16, 21, 24, 25
diagramas	18
dicha	10
disco	6
discretización	4, 14, 19, 21, 25
diseño	8
disposición	7
distancia	15
distintas	8, 9
diversas	20
documento	24, 25

E

efectivas	21
Efectividad	21
ejecución	21
ejercicio	23
ejes	15, 16
Elegant	22
eliminación	9, 21
emplea	7
enfoque	23
enriquecedora	22
enseñanzas	23
Environment	22

errores	4, 5, 8, 13, 16, 21
esenciales	21, 22
específicas	8, 23
Esquema.....	24
esté	4, 12
estén.....	4
estrategia	9, 10
estrategias	8, 9, 23
etapa	4, 8, 12, 19, 20, 24
etapas	4, 5, 8
etiquetas	6
ETL	4, 5, 19, 20, 21, 22, 23, 24
exceso.....	10
existentes	10
experiencia	22, 23
Explicación	4
Exploración	5, 7
exportación	14, 19, 21
Extracción	4

F

factor	15
faltantes	9, 10
fiables	5
Filas	7
flujo	4, 25
formato	4, 19, 21, 22
Foundation	22
fuentes.....	22
fundamentales.....	23
Futuras.....	23

G

gama	23
gráfica.....	18
Grammar	22

H

habilidad.....	23
habilidades.....	23, 24
herramienta	23
herramientas.....	5, 21, 23

I

identificación	19, 21
iguales	15
implementación	22
Importancia	21
imputación	4, 8, 9, 11, 19, 21, 24
inferior	13

información.....	5, 12, 20, 22, 23
integración.....	20
integridad	21
intercuartílico	17, 21
interrogación.....	6
Introducción.....	4
investigadores	21

L

Language.....	22
Learning	4, 9
legibilidad	6
lenguaje	4
lenguajes.....	24
limpieza.....	4, 8, 21, 23
limpio.....	12, 14, 19, 21

M

Machine.....	4, 5, 9
manejo.....	6, 20, 22, 23, 24
manipulación.....	23
manteniéndolo	19
máxima	10
máximas.....	11
mayores	15
mediana	11, 12
mejora	22
menores	15
método.....	8, 11, 17
Model	22
modelo	9
motor.....	14, 15
motores.....	14
muestra.....	7, 8, 17

N

Nearest	11
necesidades	9
Neighbors	11
nombres	4, 5, 6, 19
noviembre	22

O

observaciones	10, 13
operaciones	4
Opinión	22
orden	6
organizaciones	23

P

paquetes.....	23
parámetro.....	5
paso.....	4, 5, 7, 8, 12, 13, 20, 21
peso.....	15
posibles.....	4, 21
posterior.....	21
posteriores.....	5, 8, 19
práctica.....	20, 21
precisión.....	9, 16
preparación.....	4, 5, 21
presencia.....	11
problemas.....	8
procesamiento.....	20
Proceso.....	4, 21, 23
programación.....	4
promedio.....	10
puedan.....	19
puedes.....	20

R

rango.....	13, 17, 21
realización.....	22
recolección.....	16
recurso.....	22, 25
referencias.....	22
Reflexión.....	23
reflexiones.....	24
Reilly.....	22
Relevancia.....	23
relevante.....	5
restantes.....	11
resto.....	11, 16
resumen.....	8
robusta.....	11

S

siglas.....	4, 17
significativas.....	16, 21
siguientes.....	22

sintaxis.....	23
sistemas.....	20
Statistical.....	22
superiores.....	15, 16

T

tamaño.....	5, 8, 14
tareas.....	23
Team.....	22
técnica.....	8, 9, 10, 11, 12, 17
técnicas.....	4, 9, 12, 21, 22, 23, 25
toma.....	9, 23
tomadas.....	24
Transformación.....	4, 20
transformaciones.....	7, 19, 20
tratadas.....	9
tratamiento.....	19

U

Using.....	22
uso.....	20, 21, 23
utilización.....	21

V

Validación.....	20
valiosa.....	23
valiosas.....	23
valor.....	9, 10, 11, 13
valores... 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 23, 24	
variabilidad.....	16
variables.....	5, 6, 10, 14, 16, 21, 25
vector.....	5, 17
Verificación.....	7
visión.....	9, 22
visto.....	11
Visualisations.....	22
Visualización.....	7
visualizaciones.....	23