**ACTIVIDAD LABORATORIO NO.1**

**TRABAJO: LECTURA DE DATOS Y ANÁLISIS DESCRIPTIVO**

**PRESENTADO POR:**

**ALEJANDRO DE MENDOZA**

**PRESENTADO AL PROFESOR:**

**ING ROGERIO ORLANDO BELTRAN CASTRO**

**FUNDACIÓN UNIVERSITARIA INTERNACIONAL DE LA RIOJA**

**BOGOTÁ D.C.**

**11 DE FEBRERO**

**2026**

**TABLA DE CONTENIDO**

[INTRODUCCIÓN 3](#_Toc221745545)

[DESARROLLO ACTIVIDAD 3](#_Toc221745546)

[Arquitectura Inicial de la Implementación 4](#_Toc221745547)

[Exploración inicial del dataset (Script 01) 5](#_Toc221745548)

[Explicación detallada del código 5](#_Toc221745549)

[Verificación de calidad de datos 7](#_Toc221745550)

[Resultado esperado de esta etapa 8](#_Toc221745551)

[Imagen del archivo bike\_sharing\_clean.csv 10](#_Toc221745552)

[Análisis de la Variable Respuesta (Script 02) 11](#_Toc221745553)

[Explicación detallada del código 11](#_Toc221745554)

[Análisis de Correlaciones (Script 03) 19](#_Toc221745555)

[Explicación detallada del código 19](#_Toc221745556)

[Visualizaciones 23](#_Toc221745557)

[Guardado de Resultados 26](#_Toc221745558)

[Imágenes Gráficos Implementados 26](#_Toc221745559)

[Imágenes en Terminal 27](#_Toc221745560)

[Imagen archivo CSV generado 28](#_Toc221745561)

[Resumen 28](#_Toc221745562)

[Análisis de Distribuciones y Visualizaciones (Script 04) 29](#_Toc221745563)

[Explicación detallada del código 29](#_Toc221745564)

[Análisis Estadístico Por Grupos 31](#_Toc221745565)

[Bloque Principal De Visualizaciones (12 gráficos) 33](#_Toc221745566)

[División de Datos - Entrenamiento y Validación (Script 05) 40](#_Toc221745567)

[Explicación detallada del código 41](#_Toc221745568)

[Guardado de información estructurada 47](#_Toc221745569)

[Resumen Técnico Del Script 48](#_Toc221745570)

[Imágenes Gráficos Implementados 49](#_Toc221745571)

[Imagen archivo CSV generado 49](#_Toc221745572)

[Arquitectura Final de la Implementación 49](#_Toc221745573)

[CONCLUSIONES DE LA ACTIVIDAD 51](#_Toc221745574)

[BIBLIOGRAFÍA 52](#_Toc221745575)

[AGRADECIMIENTO 52](#_Toc221745576)

# INTRODUCCIÓN

En el contexto actual de las ciudades inteligentes y la movilidad sostenible, los sistemas de alquiler de bicicletas (bike sharing systems) se han consolidado como una alternativa eficiente, ecológica y accesible para el transporte urbano. Estos sistemas no solo contribuyen a la reducción de emisiones contaminantes y congestión vehicular, sino que también generan grandes volúmenes de datos que permiten analizar patrones de comportamiento, demanda, estacionalidad y factores que influyen en su uso. El análisis de estos datos se convierte, por tanto, en una herramienta fundamental para la toma de decisiones estratégicas y la optimización del servicio.

El presente trabajo desarrolla un análisis descriptivo exhaustivo del Bike Sharing Dataset correspondiente a la ciudad de Washington D.C. durante los años 2011 y 2012. Este dataset, ampliamente utilizado en el ámbito académico y disponible en el repositorio UCI Machine Learning Repository, contiene información diaria sobre el número total de bicicletas alquiladas, así como variables meteorológicas, temporales y de comportamiento de los usuarios.

El objetivo principal de este estudio es explorar, comprender y caracterizar el comportamiento de la variable respuesta cnt, que representa el total de bicicletas alquiladas por día, identificando los factores que influyen significativamente en su variación. A través de un proceso estructurado de análisis exploratorio de datos (EDA, por sus siglas en inglés), se busca detectar patrones estacionales, relaciones entre variables, posibles problemas de multicolinealidad y características estadísticas relevantes que permitan preparar el dataset para futuras etapas de modelización predictiva.

A diferencia de un enfoque tradicional monolítico, en el cual todo el análisis se desarrolla en un único script, este proyecto fue diseñado bajo un enfoque modular y estructurado, dividiendo cada etapa del proceso en scripts independientes. Esta decisión metodológica responde a principios de buenas prácticas en ingeniería de software y ciencia de datos, tales como la separación de responsabilidades, la claridad conceptual, la reproducibilidad paso a paso y la escalabilidad del proyecto. Cada módulo aborda una fase específica del análisis, permitiendo un flujo de trabajo organizado y fácilmente auditable.

El análisis inicia con una exploración inicial del dataset para verificar su calidad, identificar valores faltantes o duplicados y comprender su estructura general. Posteriormente, se estudia en profundidad la variable respuesta mediante estadísticas descriptivas, análisis de distribución, pruebas formales de normalidad y detección de valores atípicos. En una tercera fase, se evalúan las correlaciones entre las variables predictoras y la variable objetivo, utilizando tanto el coeficiente de Pearson como el de Spearman, además de analizar posibles problemas de multicolinealidad entre predictores.

Adicionalmente, se examinan patrones temporales y estacionales, comparaciones interanuales, comportamiento por condición climática y diferencias entre tipos de usuarios (casuales y registrados), lo que permite construir una visión integral del sistema de alquiler. Finalmente, se realiza una división adecuada del dataset en conjuntos de entrenamiento y validación mediante un esquema temporal 80-20, respetando la naturaleza secuencial de los datos y evitando el problema de data leakage, aspecto crítico en problemas de series temporales.

Este estudio no solo permite comprender el comportamiento histórico del sistema de bicicletas compartidas, sino que también establece una base sólida para el desarrollo posterior de modelos predictivos robustos. Al integrar análisis estadístico, visualización de datos y principios de ingeniería de software, el proyecto demuestra la importancia de un enfoque estructurado y metodológicamente riguroso en el análisis de datos reales.

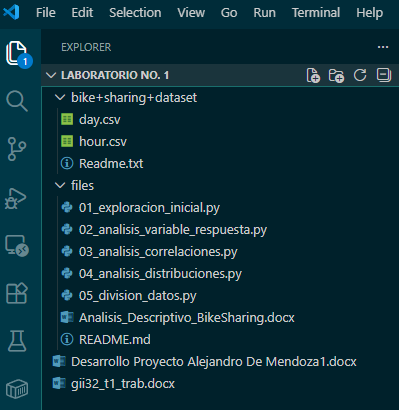
Este trabajo representa una aplicación práctica de técnicas de análisis exploratorio y preparación de datos sobre un problema real de movilidad urbana, destacando la relevancia del análisis estadístico como etapa previa indispensable para cualquier proceso de modelización en ciencia de datos.

# DESARROLLO ACTIVIDAD

A continuación, se presenta el desarrollo paso a paso del análisis descriptivo aplicado al Bike Sharing Dataset, siguiendo un enfoque modular y estructurado. Cada etapa del proceso fue implementada en scripts independientes, lo que permite organizar el trabajo de forma clara, reproducible y progresiva. En las siguientes secciones se documentan los procedimientos realizados, los resultados obtenidos y las principales interpretaciones estadísticas y gráficas derivadas del análisis exploratorio de los datos.

## Arquitectura Inicial de la Implementación

Tal como se indicó anteriormente en la introducción no se decidió ejecutar un desarrollo monolítico para este caso ya que este proyecto fue diseñado bajo un enfoque modular y estructurado, dividiendo cada etapa del proceso en scripts independientes. A continuación, detallo la imagen de la arquitectura inicial, la cual va a ser modificada con la creación de nuevos archivos a medidas que se vayan ejecutando los scripts, esto para dar claridad en el desarrollo de este laboratorio:



Entonces como se puede denotar en la carpeta bike+sharing+dataset ubicada en la raíz del proyecto Laboratorio No. 1, tenemos los datos descargados del link remitido en la actividad los cuales son los siguientes:

* day.csv
* hour.csv
* readme.txt

Por otra parte, tenemos la carpeta de nombre “files”, ubicada de igual manera en la raíz del proyecto Laboratorio No. 1, donde están los scripts desarrollados de mi parte con los siguientes nombres:

* 01\_exploracion\_inicial.py: visión general del conjunto de datos, calidad básica de la información y detección de problemas.
* 02\_analisis\_variable\_respuesta.py: estudio en profundidad la variable respuesta
* 03\_analisis\_correlaciones.py: evaluar la relación existente entre dicha variable y el resto de variables del dataset.
* 04\_analisis\_distribuciones.py: estudiar cómo se distribuye la variable respuesta bajo diferentes condiciones.
* 05\_division\_datos.py: división de los datos en conjuntos de entrenamiento y validación.
* README.md: Archivo con la explicación completa del proyecto de manera técnica.

Finalmente tenemos los archivos ubicados en la raíz del proyecto laboratorio No. 1 que son los siguientes:

* Desarrollo Proyecto Alejandro De Mendoza.docx: Desarrollo de la documentación completa del laboratorio en Word.
* gii32\_t1\_trab.docx: Actividad a desarrollar.

Con lo que se finaliza la explicación de la arquitectura y a continuación se procede a ejecutar la explicación de cada uno de los scripts creados, los archivos nuevos generados a partir de estos y toda su funcionalidad.

## Exploración inicial del dataset (Script 01)

En esta primera etapa se realiza la carga y exploración inicial del Bike Sharing Dataset (archivo day.csv). El objetivo es obtener una visión general del conjunto de datos: conocer su tamaño, el periodo temporal que cubre, los tipos de variables disponibles y validar la calidad básica de la información antes de avanzar hacia análisis estadísticos más profundos.

Esta fase es fundamental porque permite detectar problemas comunes como valores faltantes, registros duplicados o tipos de datos incorrectos, los cuales podrían afectar la interpretación de resultados y la futura construcción de modelos predictivos.

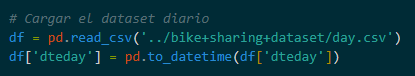
### Explicación detallada del código

#### 1) Importación de librerías

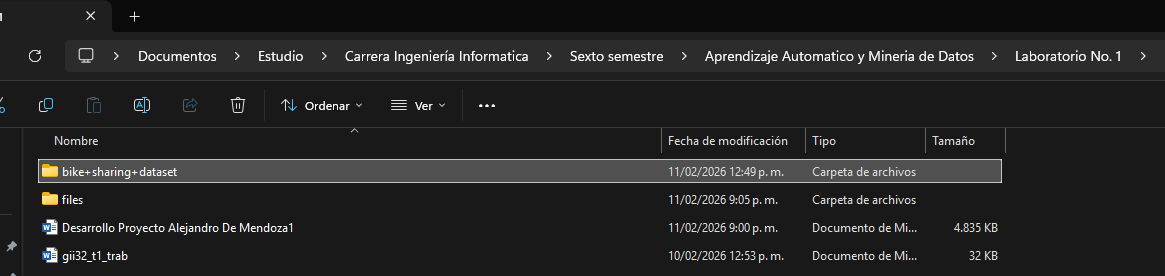


* pandas (pd): se usa para cargar y manipular datos en forma de tabla (DataFrame).

#### 2) Carga del dataset y conversión de fecha



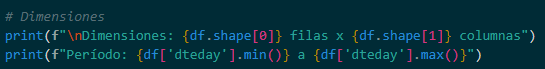
* Se carga el archivo day.csv en un DataFrame llamado df (ojo en este caso se carga directamente desde la carpeta de donde se descargó el dataset del link proporcionado la cual se extrajo y se ubicó dentro de la carpeta de laboratorio No. 1), a continuación, las imágenes explicatorias:





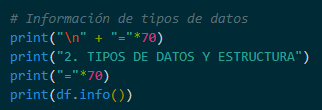
* Se convierte la columna dteday a tipo fecha (datetime), lo cual es clave para:
  + Calcular correctamente el rango de fechas,
  + Ordenar cronológicamente,
  + Hacer análisis temporal posterior.

#### 3) Información general del dataset



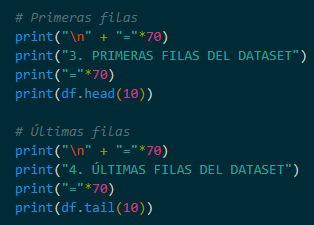
* df.shape devuelve el tamaño del dataset:
* shape[0] = número de filas (registros)
* shape[1] = número de columnas (variables)
* Se imprime el periodo mínimo y máximo usando min() y max() sobre la fecha.

#### 4) Tipos de datos y estructura



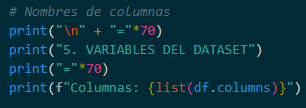
* df.info() muestra:
  + número de columnas y filas,
  + tipo de dato de cada variable (int, float, datetime, etc.),
  + conteo de valores no nulos por columna.
* Esto permite detectar rápidamente columnas con valores faltantes o tipos incorrectos.

#### 5) Visualización rápida: primeras y últimas filas



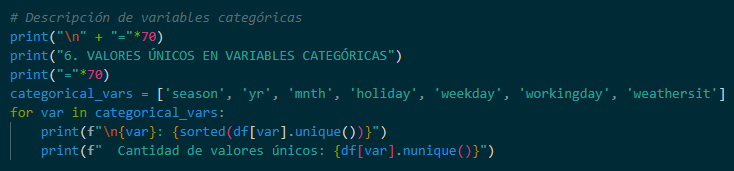
* head(10) muestra las primeras 10 filas.
* tail(10) muestra las últimas 10 filas.
* Esto sirve para confirmar que:
  + El dataset cargó correctamente,
  + Las columnas tienen valores razonables,
  + La fecha se interpreta bien.

#### 6) Listado de columnas



* Se imprime la lista completa de variables para tener claridad de qué información está disponible.

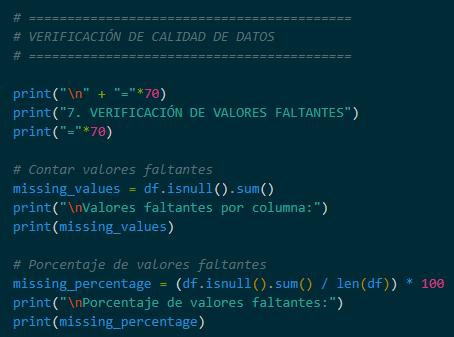
#### 7) Valores únicos en variables categóricas



* Aquí se analiza el contenido de variables que representan categorías o etiquetas:
  + unique() obtiene los valores distintos presentes.
  + nunique() cuenta cuántos valores distintos hay.
* Esto permite verificar consistencia. Por ejemplo:
  + yr debería tener solo [0, 1] (2011, 2012)
  + holiday debería ser [0, 1]
  + season normalmente va de 1 a 4
  + mnth debería ser 1 a 12
* Si aparecieran valores fuera de rango (ej. mnth=13) sería una alerta.

### Verificación de calidad de datos

#### 8) Valores faltantes

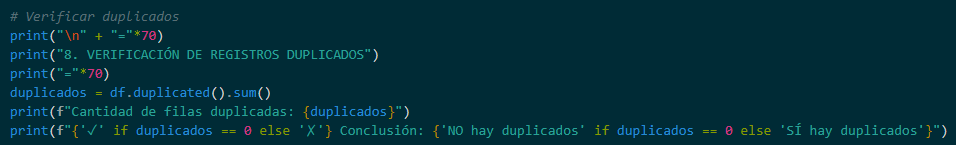


* isnull() detecta celdas vacías (NaN).
* sum() cuenta cuántos faltantes hay por columna.
* Luego se calcula el porcentaje de faltantes dividiendo entre el total de registros.
* Finalmente:



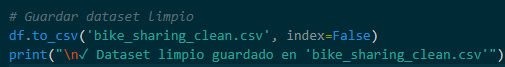
* Si la suma total de faltantes es 0, se imprime una conclusión positiva.

#### 9) Verificación de duplicados



* duplicated() marca filas repetidas.
* sum() cuenta cuántas existen.
* Luego se imprime una conclusión indicando si hay o no duplicados.

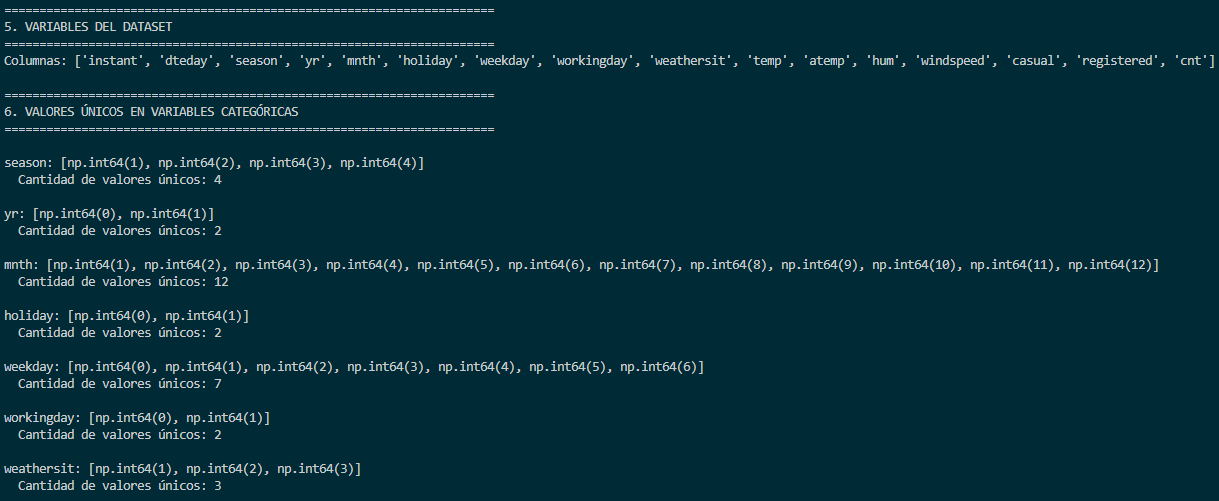
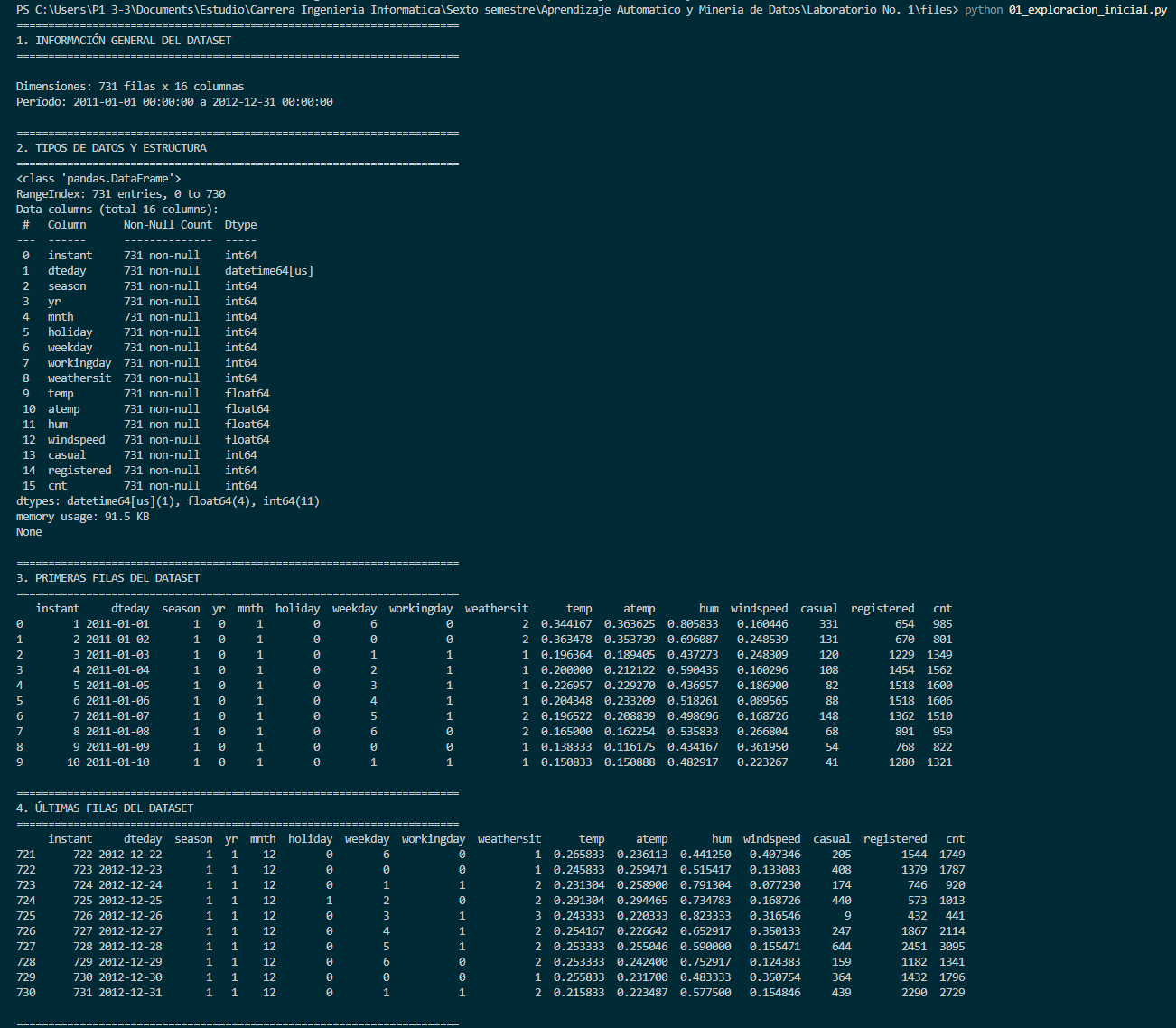
#### 10) Guardado del dataset limpio

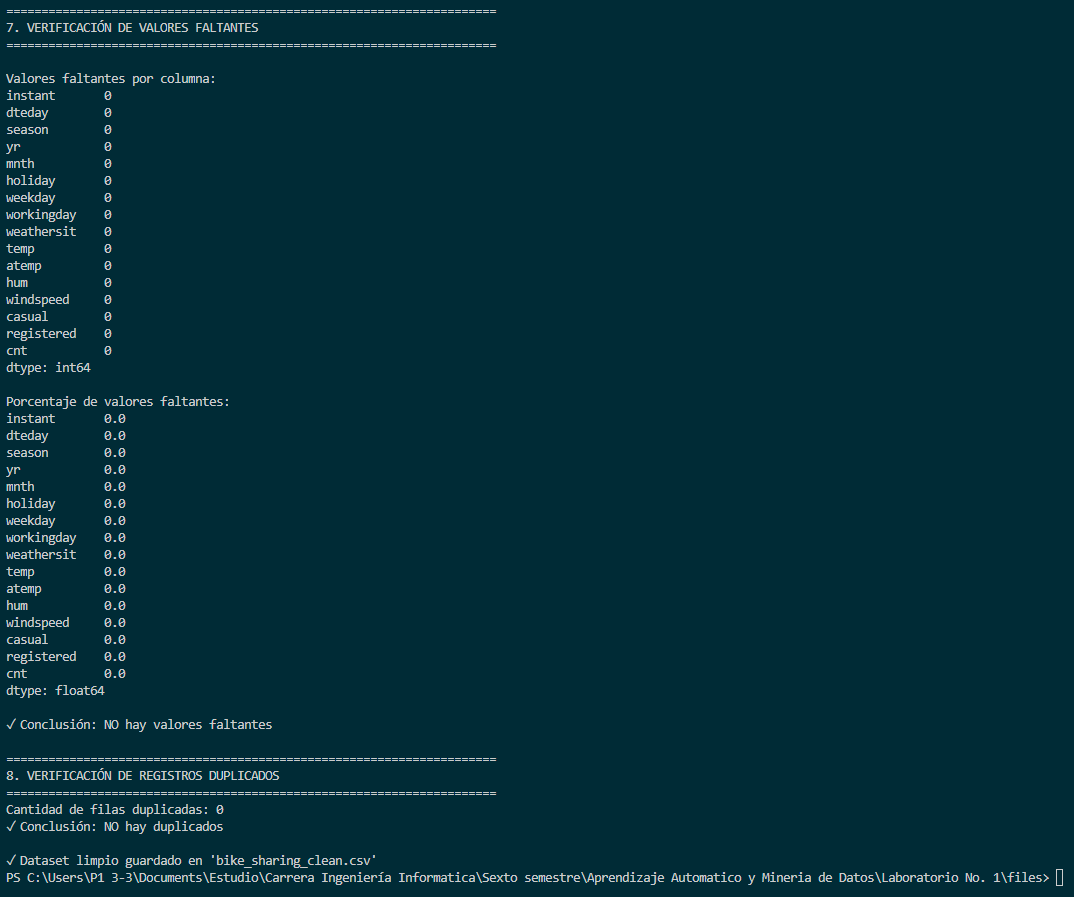


* Se guarda una versión “limpia” del dataset (mismo contenido si no hay errores).
* index=False evita guardar el índice numérico como columna adicional.

### Resultado esperado de esta etapa

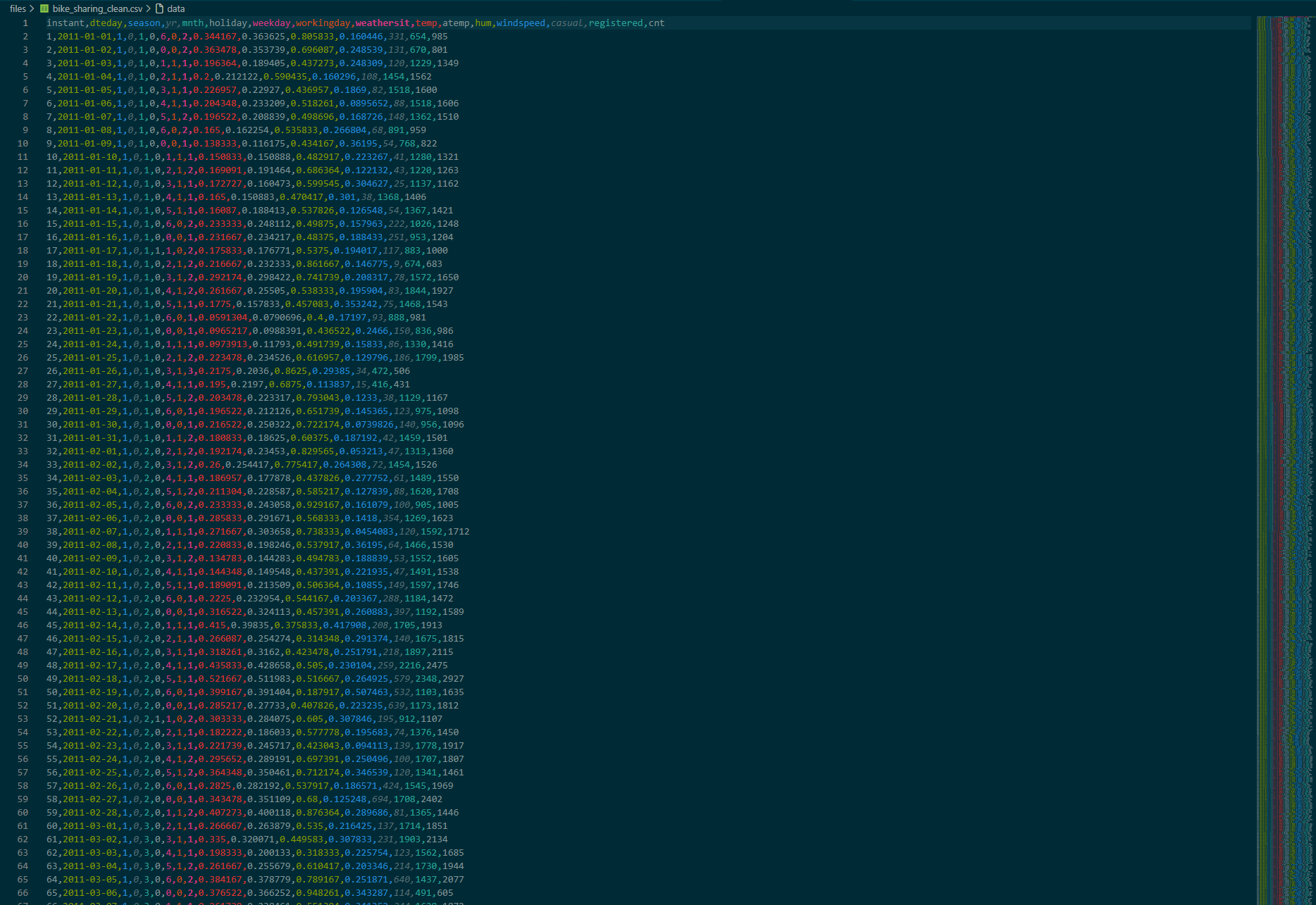
* Al finalizar este script, se debe tener:
  + Confirmación del tamaño y periodo del dataset.
  + Tipos de datos verificados (especialmente la fecha).
  + Revisión básica de categorías.
  + Validación de calidad:
  + Sin valores faltantes
  + Sin duplicados
  + Archivo generado: bike\_sharing\_clean.csv
* Se remiten imágenes de la salida de la consola en Visual Studio Code:





### Imagen del archivo bike\_sharing\_clean.csv

A continuación, la imagen del archivo .csv generado



## Análisis de la Variable Respuesta (Script 02)

Una vez realizada la exploración inicial del dataset y verificada su calidad (ausencia de valores faltantes y duplicados), el siguiente paso fundamental en cualquier proceso de análisis de datos es estudiar en profundidad la variable respuesta.

En este proyecto, la variable objetivo es cnt, que representa el número total de bicicletas alquiladas por día. Al tratarse de la variable que posteriormente será modelada, es imprescindible comprender:

* Su comportamiento estadístico
* Su nivel de dispersión
* Su forma de distribución
* La presencia de valores atípicos
* Su grado de normalidad

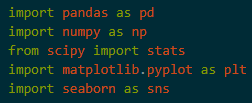
Este análisis permite anticipar posibles transformaciones necesarias, identificar riesgos para la modelización (como asimetría u outliers) y comprender la dinámica real del fenómeno que se desea predecir. Este Script se centra exclusivamente en este análisis univariado profundo, combinando:

* Estadística descriptiva clásica
* Medidas de forma (asimetría y curtosis)
* Tests formales de normalidad
* Detección de outliers
* Visualizaciones complementarias

Este enfoque permite obtener una comprensión integral del comportamiento estadístico de la variable objetivo antes de avanzar hacia análisis bivariados o modelización.

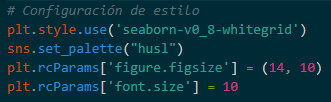
### Explicación detallada del código

#### 1. Importación de Librerías



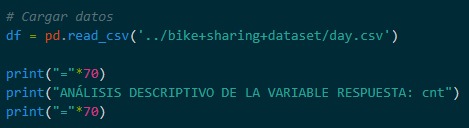
* Se utilizan:
  + pandas → manipulación y análisis de datos.
  + numpy → generación de vectores y operaciones numéricas (sí se usa aquí).
  + scipy.stats → pruebas estadísticas y funciones de probabilidad.
  + matplotlib / seaborn → visualización.
  + seaborn as sns → para manejar el estilo de seaborn con una paleta de colores.

#### 2. Configuración de Estilo Gráfico



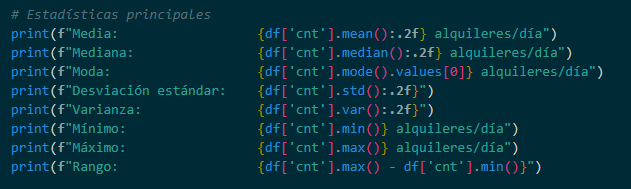
* Se define un estilo visual uniforme para todas las gráficas:
  + Fondo blanco con grilla
  + Paleta profesional
  + Tamaño estándar de figuras
  + Tamaño de fuente consistente
  + Esto mejora la calidad visual del análisis.

#### 3. Carga del Dataset

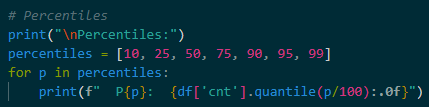


* Se carga el dataset diario original. En este script solo interesa la variable cnt.

#### 4. Estadísticas Descriptivas Básicas



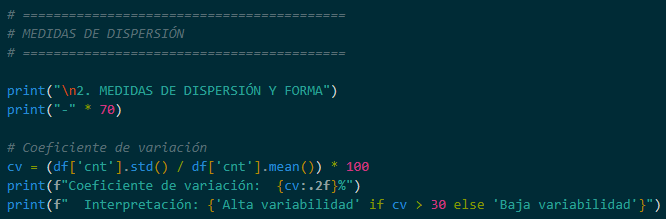
* Se calculan las métricas fundamentales:
  + df['cnt'].mean()
  + df['cnt'].median()
  + df['cnt'].mode()
  + df['cnt'].std()
  + df['cnt'].var()
  + df['cnt'].min()
  + df['cnt'].max()
* Se obtienen:
  + Media
  + Mediana
  + Moda
  + Desviación estándar
  + Varianza
  + Mínimo y máximo
  + Rango
* Esto permite entender:
  + Tendencia central
  + Dispersión
  + Amplitud de los datos
* También se calculan percentiles relevantes:



* + Lo cual permite analizar la distribución por cuantiles (P10, P25, P50, etc.).

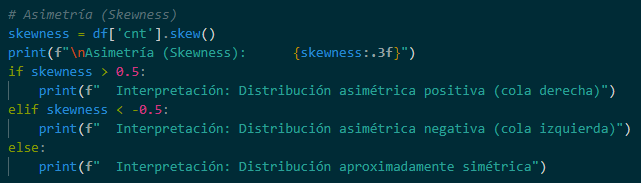
#### 5. Medidas de Dispersión y Forma

##### Coeficiente de Variación



* Mide dispersión relativa respecto a la media.
* Si CV > 30% → alta variabilidad.

##### Asimetría (Skewness)



* 0.5 → asimetría positiva
* < -0.5 → asimetría negativa
* Entre -0.5 y 0.5 → aproximadamente simétrica
* Permite entender si la distribución está sesgada.

##### Curtosis

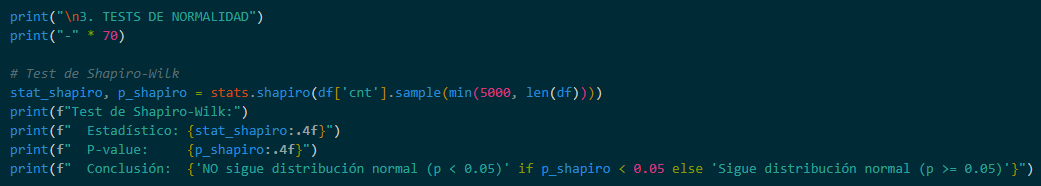


* 0 → leptocúrtica (más puntiaguda)
* < 0 → platicúrtica (más aplanada)
* = 0 → similar a normal
* Describe la concentración de valores en los extremos.

#### 6. Tests de Normalidad

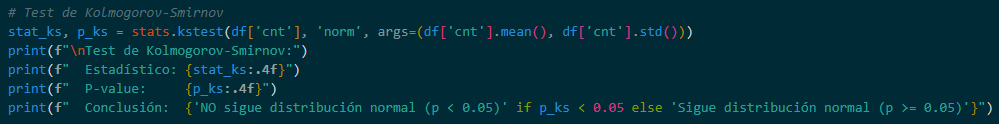
Se aplican las siguientes dos pruebas formales:

##### Shapiro-Wilk



* H₀: Los datos siguen distribución normal.
* Si p < 0.05 → se rechaza normalidad.
* Se usa sample por seguridad cuando hay datasets grandes.

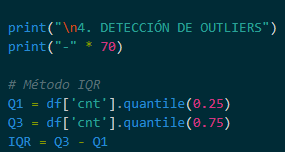
##### Kolmogorov-Smirnov



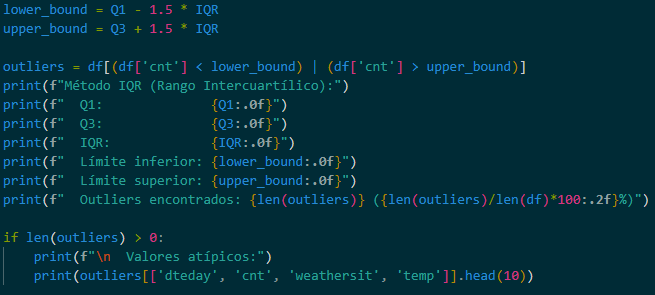
* Compara la distribución empírica contra una normal teórica con la misma media y desviación estándar.

#### 7. Detección de Outliers (Método IQR)

* Se utiliza el método clásico del rango intercuartílico:

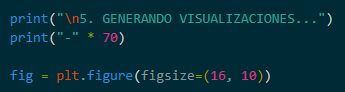


* Límites:



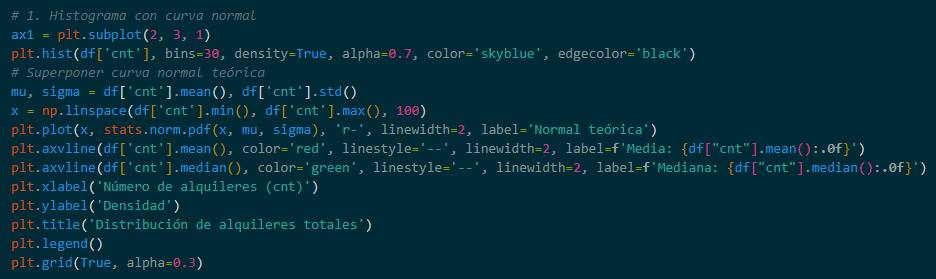
* + Se identifican valores fuera de esos límites.
  + Este método es robusto y no depende de la normalidad.

#### 8. Visualizaciones



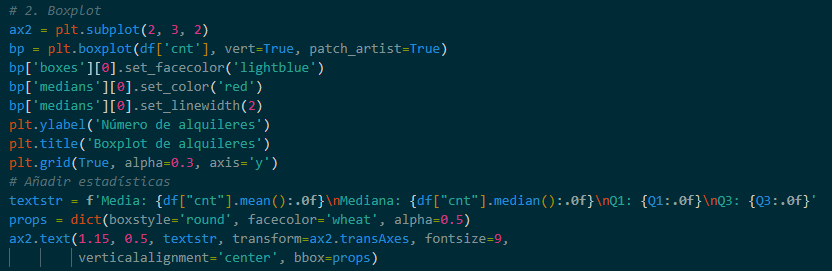
* Se generan 6 gráficos clave:

##### Histograma + curva normal teórica



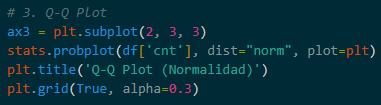
* Permite comparar visualmente con una normal ideal.
* Se superponen media y mediana.

##### Boxplot



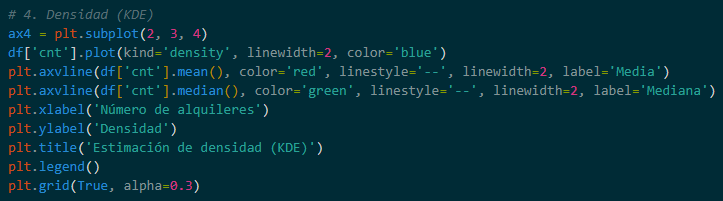
* Resume dispersión.
* Identifica outliers.
* Se añaden estadísticas en un recuadro.

##### Q-Q Plot



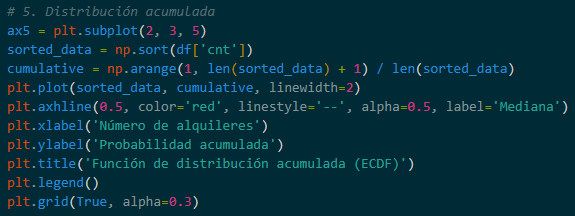
* Compara cuantiles reales vs cuantiles normales.
* Si los puntos siguen la línea → distribución normal.

##### KDE (Densidad)



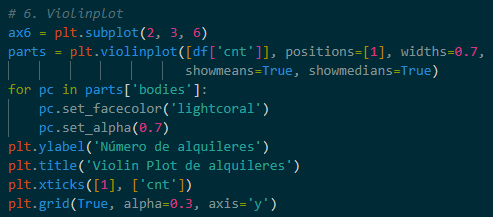
* Estimación suave de la distribución.
* Permite ver forma real de la curva.

##### Función de distribución acumulada (ECDF)



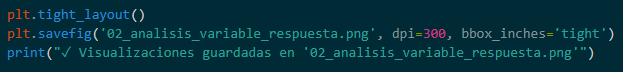
* Muestra probabilidad acumulada.
* Permite interpretar percentiles visualmente.

##### Violin Plot



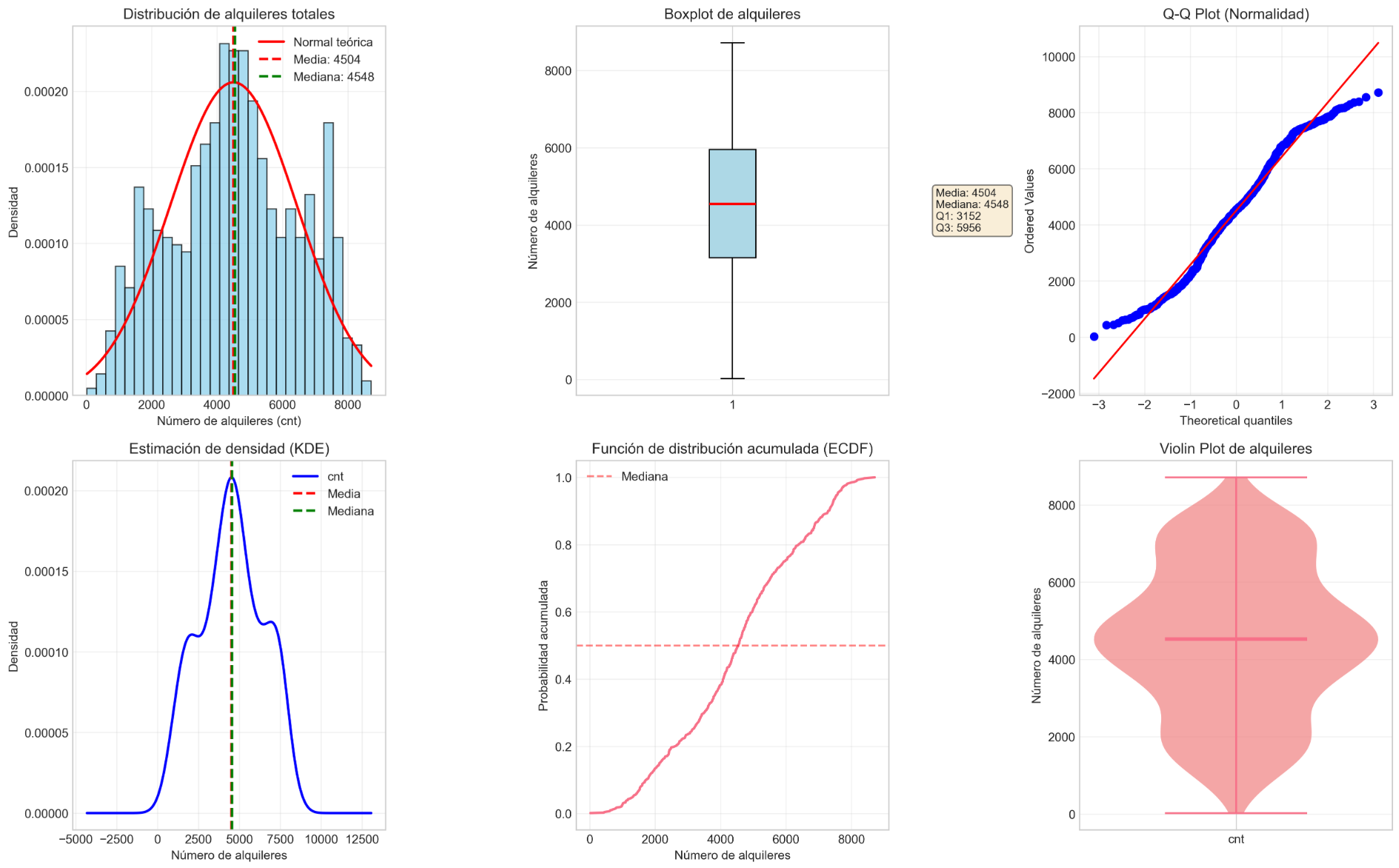
* Combina boxplot + densidad.
* Permite visualizar forma completa de la distribución.

#### 9. Guardado de gráficos



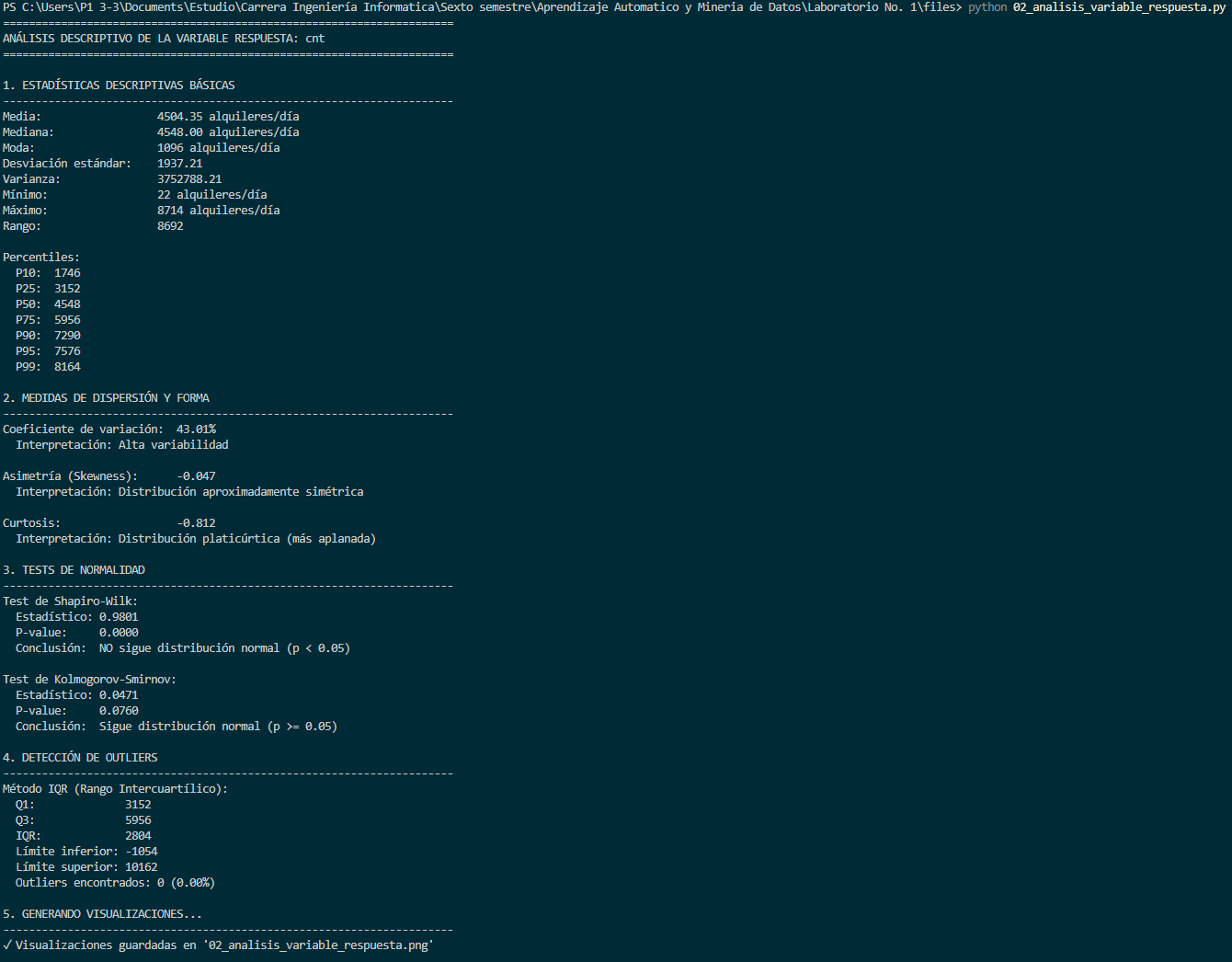
* Todas las gráficas se guardan en: 02\_analisis\_variable\_respuesta.png

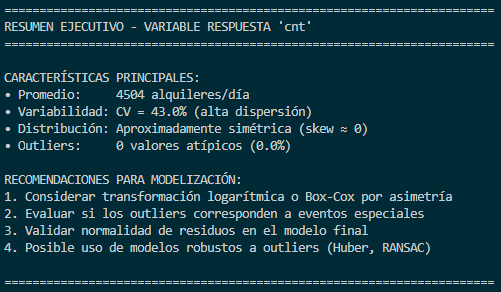
#### 10. Imágenes de graficos implementados



#### 11. Salida de Terminal

A continuación, las imágenes de la salida de la terminal de Visual Studio Code:





#### 12. Resumen

El script finaliza imprimiendo un resumen interpretativo automático:

* Promedio
* Nivel de dispersión
* Tipo de distribución
* Cantidad de outliers
* Recomendaciones para modelización
* Esto facilita la comunicación de resultados.

## Análisis de Correlaciones (Script 03)

Una vez comprendida la distribución y comportamiento estadístico de la variable respuesta cnt, el siguiente paso lógico dentro del análisis exploratorio consiste en evaluar la relación existente entre dicha variable y el resto de variables del dataset.

El análisis de correlaciones permite:

* Identificar qué variables están más asociadas al número de alquileres.
* Determinar la dirección de la relación (positiva o negativa).
* Medir la fuerza de asociación entre variables.
* Detectar posibles problemas de multicolinealidad entre predictores.
* Fundamentar la selección de variables para futuros modelos predictivos.
* En este script se utilizan dos coeficientes de correlación:
* Pearson → mide relación lineal.
* Spearman → mide relación monótona (no paramétrica).

Además, se realiza un análisis de multicolinealidad y se generan visualizaciones para reforzar la interpretación estadística.

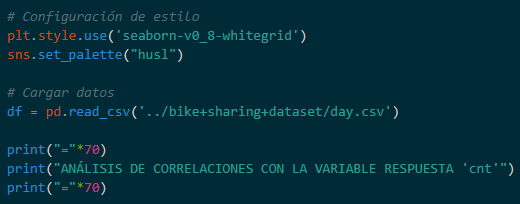
### Explicación detallada del código

#### 1. Importación de librerías



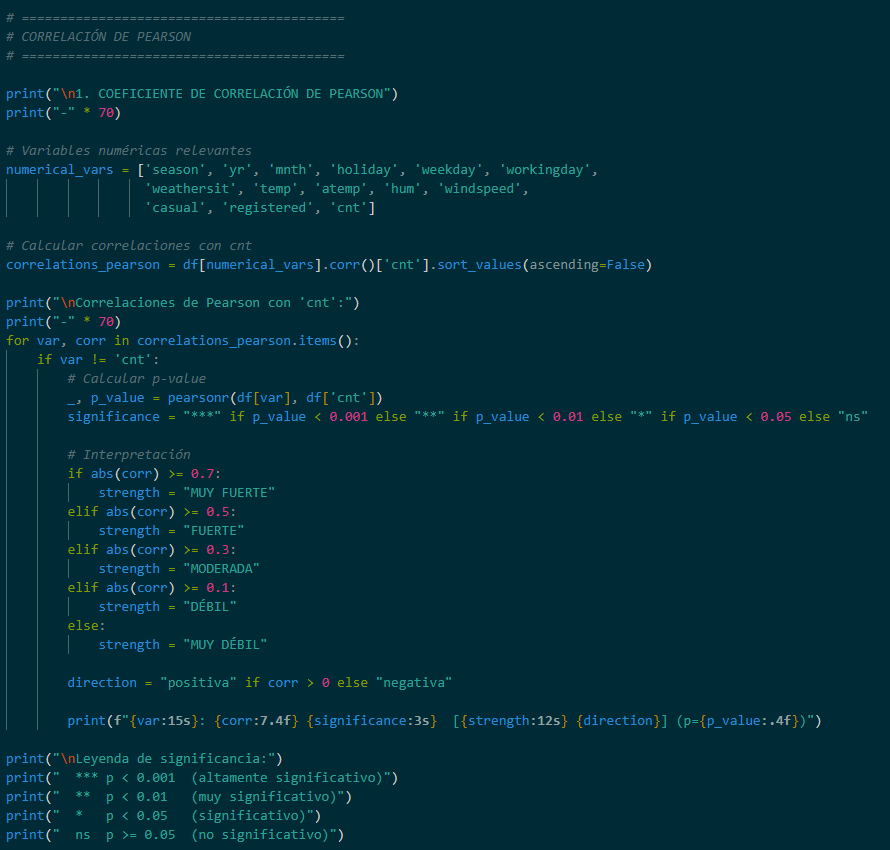
* pandas: Manipulación del dataset.
* Numpy: Se usa para:
  + Crear máscaras triangulares en el heatmap.
  + Usar np.polyfit para regresiones en scatterplots.
  + Manejo de np.nan.
* Matplotlib: Generación de gráficos.
* seaborn: Se usa para:
  + Se usa sns.heatmap
  + Se usa sns.boxplot
* scipy.stats: Se usa para:
  + pearsonr() → correlación + p-value
  + spearmanr() → correlación no paramétrica + p-value

#### 2. Configuración visual y cargue de datos



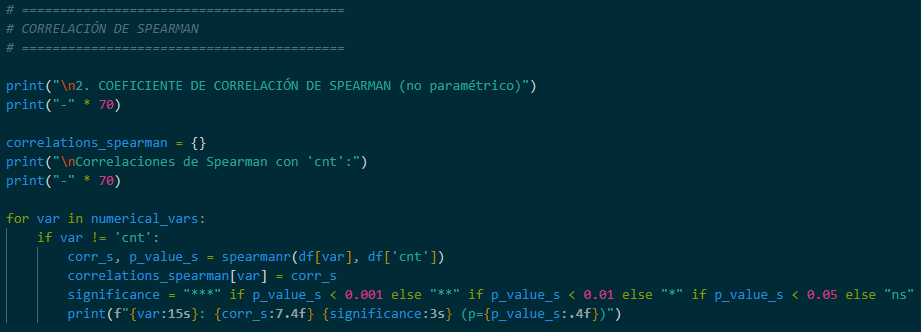
* Define estilo gráfico profesional y consistente.
* Se cargan los datos desde day.csv

#### 3. Cálculo de Correlación de Pearson



* Se seleccionan variables numéricas.
* Se calcula la matriz de correlación.
* Se extrae solo la columna correspondiente a cnt.
* Luego para cada variable:
  + \_, p\_value = pearsonr(df[var], df['cnt'])
    - Se calcula:
      * Coeficiente r
      * p-value (significancia estadística)
      * Después el código clasifica automáticamente:
        + MUY FUERTE
        + FUERTE
        + MODERADA
        + DÉBIL
        + MUY DÉBIL
* Esto automatiza la interpretación estadística.

#### 4. Correlación de Spearman



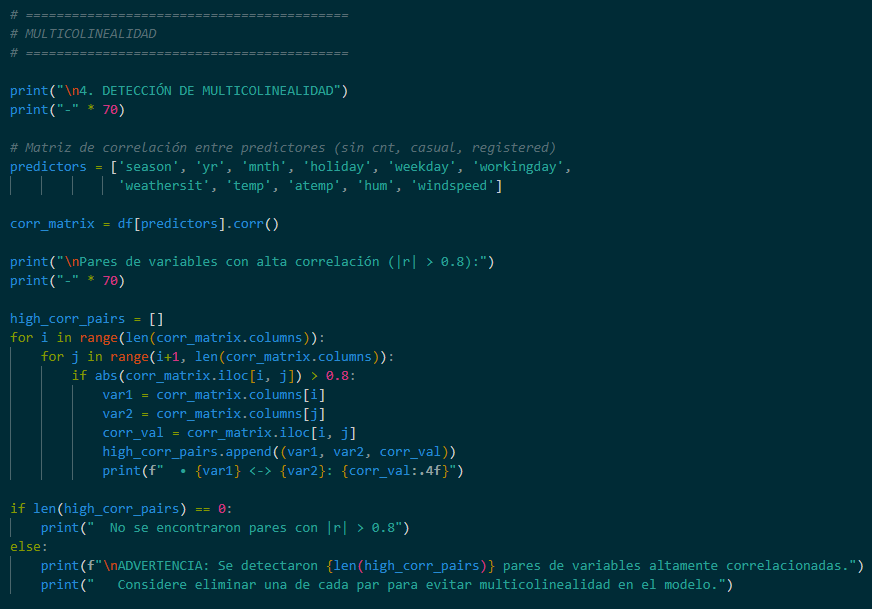
* Se usa cuando:
  + No se asume normalidad.
  + La relación puede no ser lineal.
  + Spearman trabaja con rangos.
* Esto refuerza robustez del análisis.

#### 5. Categorización automática



* El script divide variables en:
  + Fuertes positivas
  + Moderadas positivas
  + Débiles positivas
  + Débiles negativas
  + Moderadas negativas
  + Fuertes negativas
* Esto convierte números en interpretación clara.
* Es una muy buena práctica para documentación.

#### 6. Detección de Multicolinealidad



* Aquí se elimina:
  + cnt
  + casual
  + registered
* Porque:
  + casual + registered = cnt
  + Incluirlas distorsiona el análisis.
* Luego se busca:
  + if abs(corr\_matrix.iloc[i, j]) > 0.8:
* Umbral típico en estadística para colinealidad alta.
* Detecta correctamente:
  + temp ↔ atemp
  + season ↔ mnth
* Esto es clave para evitar modelos inestables.

### Visualizaciones

* A continuación, los respectivos gráficos de visualización:

#### Heatmap de correlación



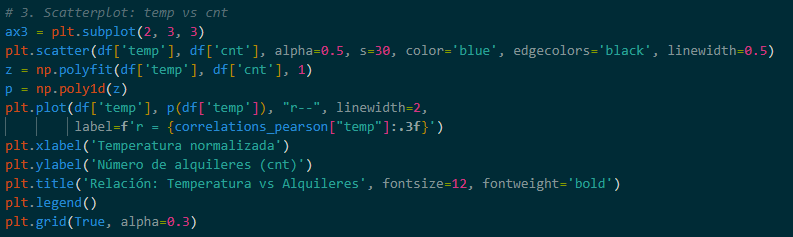
* Visualiza correlaciones entre predictores.
* Se usa máscara triangular para evitar duplicados.
* Escala de color centrada en 0.

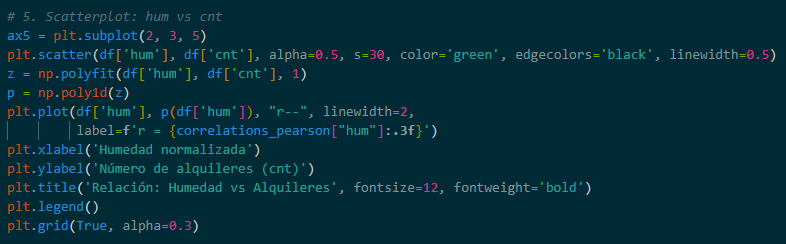
#### Barplot de correlaciones con cnt



* Ordenadas de menor a mayor.
* Colores:
  + Verde → positivo
  + Rojo → negativo
* Incluye líneas de referencia en:
  + ±0.3
  + ±0.5
* Esto facilita interpretación visual.

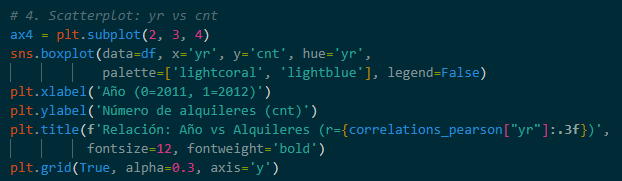
#### Scatterplots con regresión

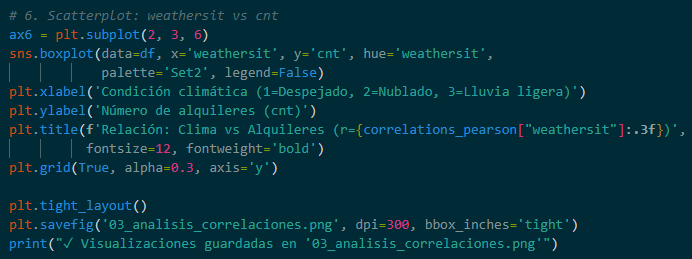




* Para:
  + temp vs cnt
  + hum vs cnt
* Se usa:
  + np.polyfit()
* Para generar línea de tendencia.
* Esto muestra relación lineal visualmente.

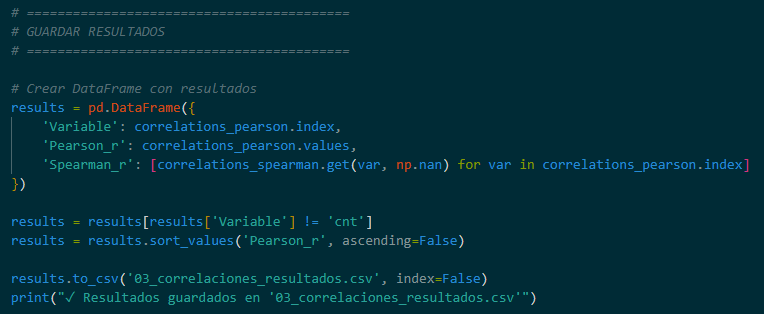
#### Boxplots categóricos





* Para:
  + yr
  + weathersit
  + sns.boxplot()
* Muestran diferencias de distribución entre categorías.

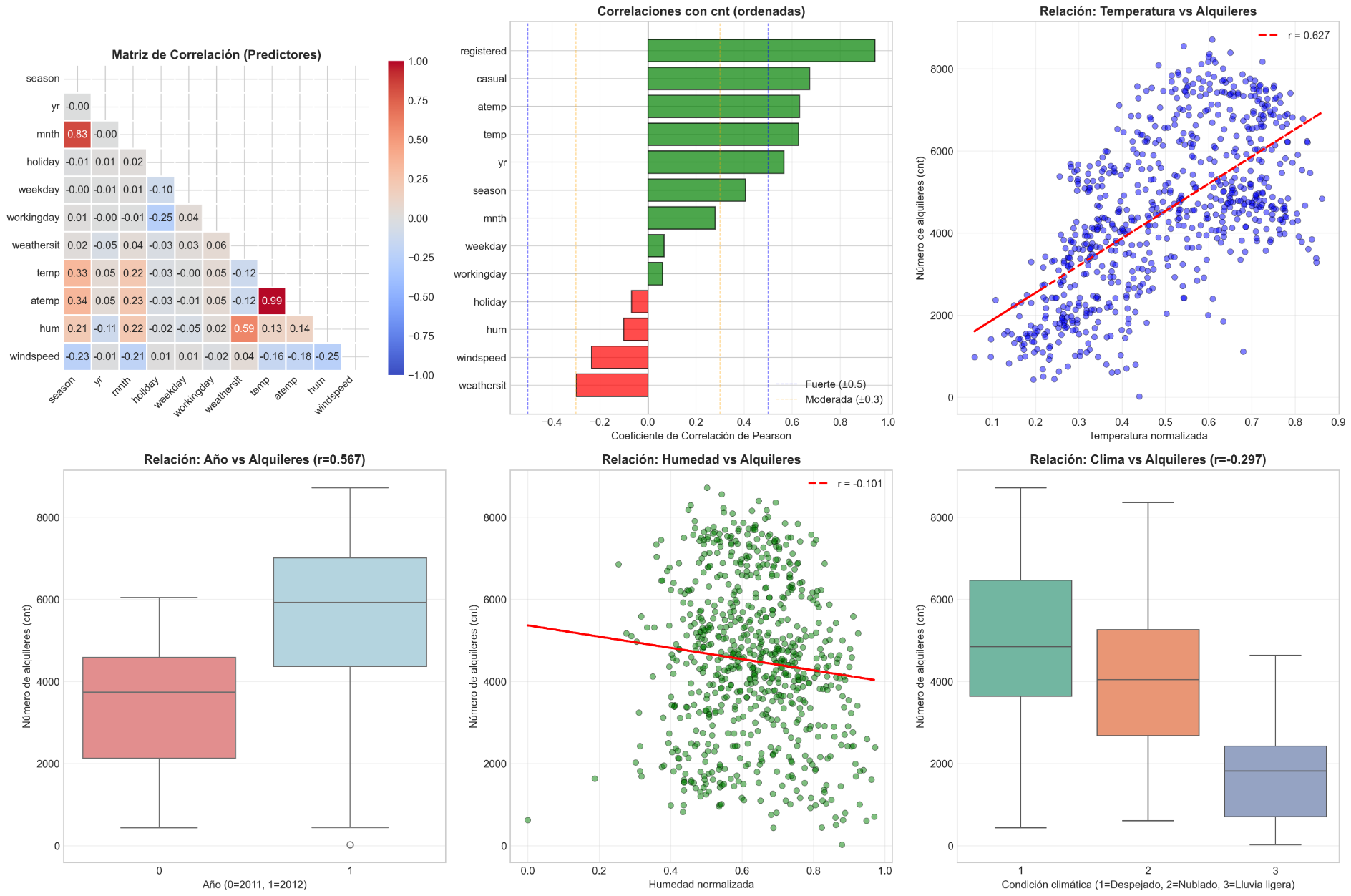
### Guardado de Resultados



* Se crea un DataFrame con:
  + Pearson
  + Spearman
* Y se guarda en:
  + 03\_correlaciones\_resultados.csv
* Esto permite usar resultados en Excel o el Word.

### Imágenes Gráficos Implementados

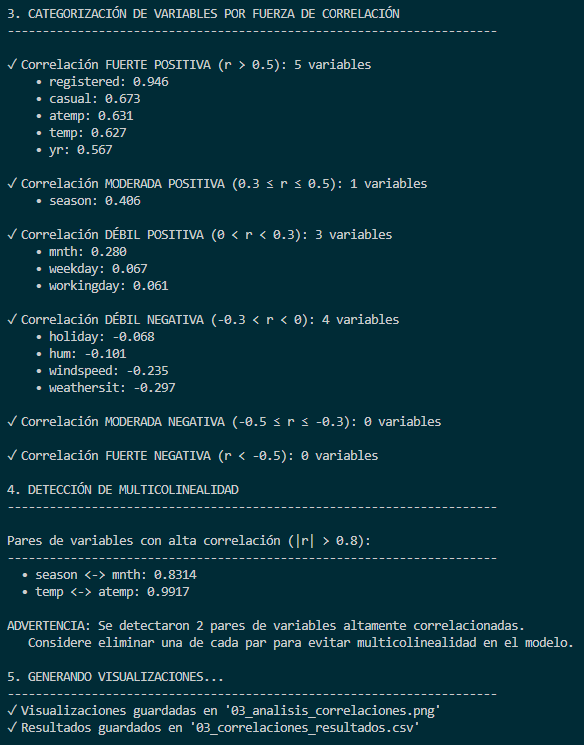
* A continuación, la imagen de los gráficos que se han desarrollado:



### Imágenes en Terminal

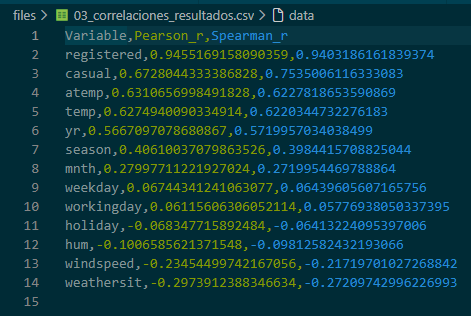
A continuación, las imágenes de la resolución en la consola:



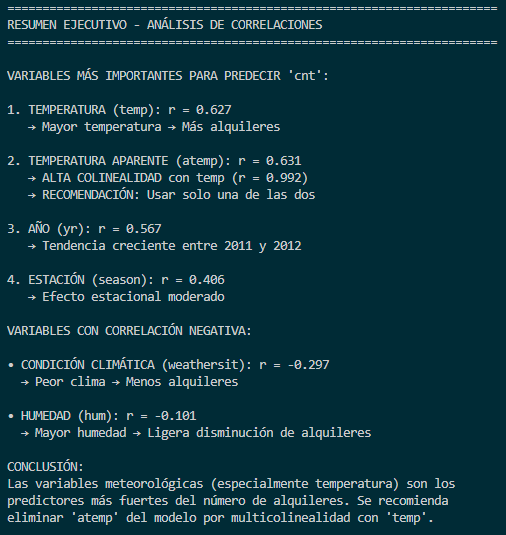


### Imagen archivo CSV generado

A continuación, la imagen del archivo .csv generado:



### Resumen



* El script genera automáticamente:
  + Variables más importantes
  + Advertencia de colinealidad
  + Interpretación económica

## Análisis de Distribuciones y Visualizaciones (Script 04)

El análisis exploratorio no termina únicamente con estadísticas básicas o correlaciones individuales.

Una parte fundamental en cualquier proyecto de minería de datos consiste en estudiar cómo se distribuye la variable respuesta bajo diferentes condiciones, ya que esto permite descubrir patrones más profundos en los datos.

En este caso, el dataset Bike Sharing Dataset contiene información diaria de alquileres de bicicletas en Washington D.C. durante los años 2011 y 2012.

La variable objetivo principal es:

* cnt → total de bicicletas alquiladas por día.

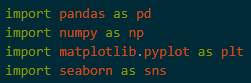
El objetivo de este cuarto script es identificar patrones de comportamiento en la demanda de bicicletas, analizando cómo cambia el número de alquileres según:

* La estación del año
* Las condiciones climáticas
* El día de la semana
* El mes
* La evolución temporal entre años
* El tipo de usuario (casual vs registrado)

Además, este script genera una colección completa de visualizaciones que permiten comprender de forma gráfica el impacto de estas variables en la demanda.

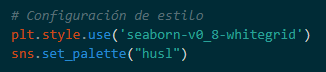
### Explicación detallada del código

#### 1. Importación de librerías



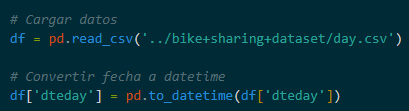
* Aquí se utilizan librerías estándar:
  + pandas → manipulación y agrupación de datos
  + numpy → cálculos numéricos y regresiones lineales (polyfit)
  + matplotlib / seaborn → generación de gráficos

#### 2. Configuración del estilo gráfico



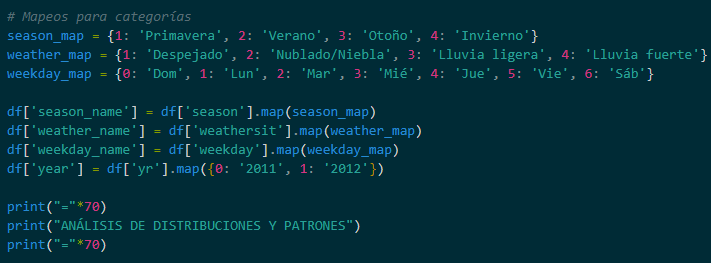
* Esto asegura que:
  + Los gráficos sean más limpios y profesionales
  + Se use una paleta de colores uniforme

#### 3. Carga del dataset y tratamiento de fechas



* Se carga el dataset diario y se convierte la columna dteday en formato de fecha real para poder realizar análisis temporales.

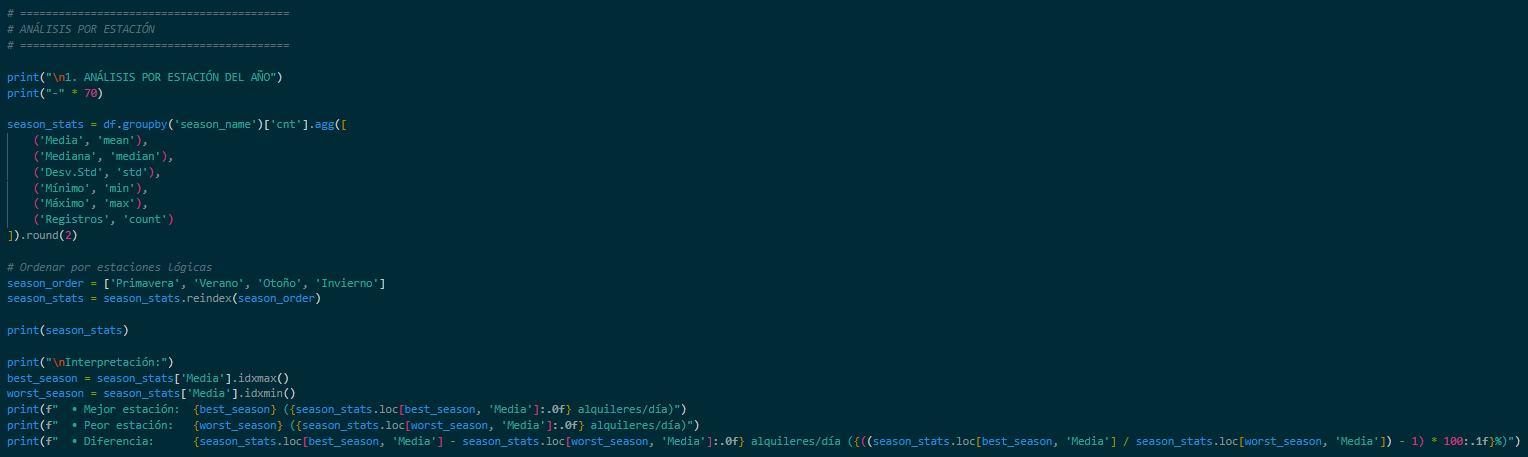
#### 4. Creación de etiquetas categóricas



* El dataset original usa números para representar categorías.
* Se crean mapas para traducirlos a nombres comprensibles.
* Luego se agregan columnas nuevas:
  + df['season\_name'] = df['season'].map(season\_map)
  + df['weather\_name'] = df['weathersit'].map(weather\_map)
  + df['weekday\_name'] = df['weekday'].map(weekday\_map)
  + df['year'] = df['yr'].map({0: '2011', 1: '2012'})
* Esto mejora la interpretación en tablas y gráficos.

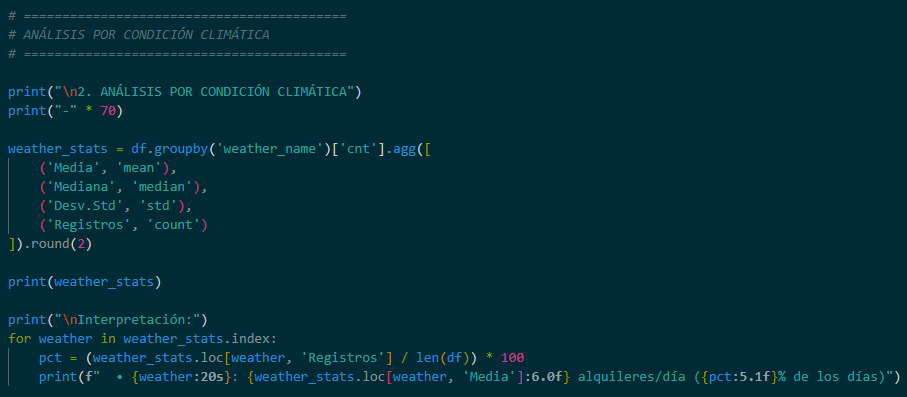
### Análisis Estadístico Por Grupos

#### 5. Análisis por estación del año



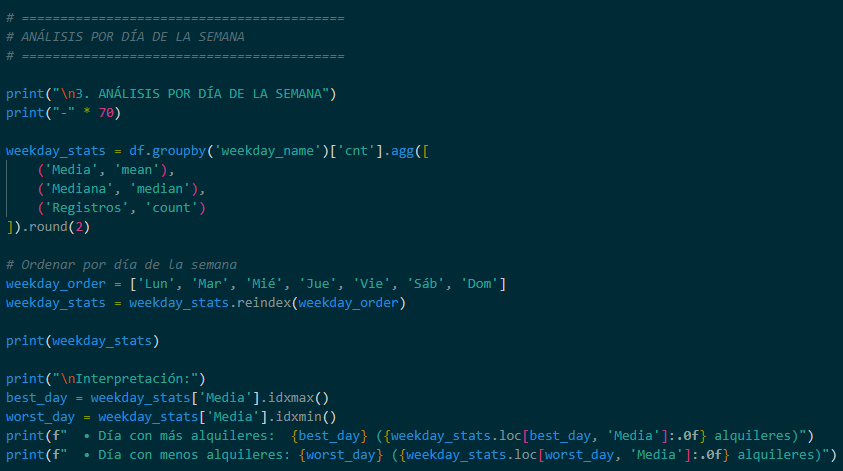
* Aquí se agrupa el dataset por estación y se calculan medidas descriptivas.
* Se identifica:
  + Estación con más demanda
  + Estación con menos demanda
  + Diferencia porcentual entre ambas
  + Esto permite detectar estacionalidad fuerte.

#### 6. Análisis por condición climática



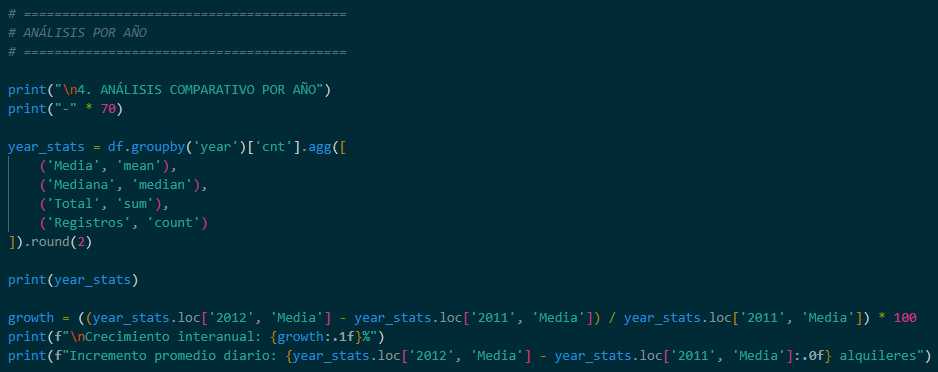
* Este bloque evalúa el impacto del clima sobre la demanda.
* Interpretación típica:
  + Días despejados → más alquileres
  + Lluvia → fuerte reducción de uso

#### 7. Análisis por día de la semana



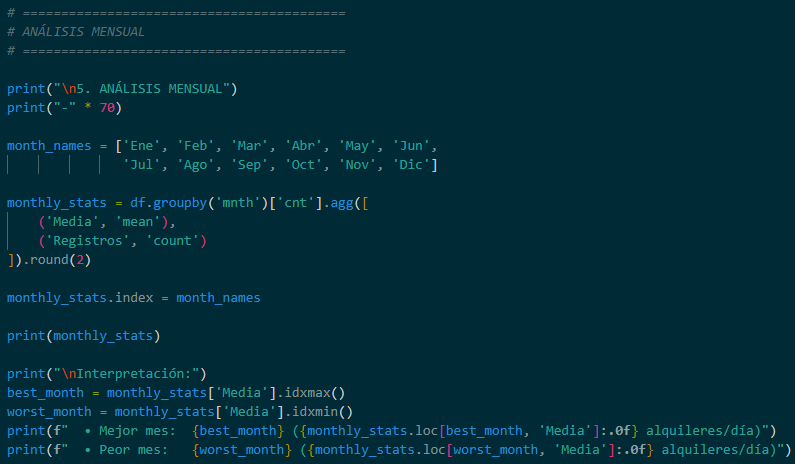
* Este análisis responde preguntas como:
  + ¿Se alquilan más bicicletas entre semana o en fines de semana?
  + ¿Existe un patrón laboral o recreativo?

#### 8. Análisis por año



* Este bloque permite medir el crecimiento del sistema:
  + aumento de la demanda
  + tendencia temporal positiva

#### 9. Análisis mensual



* Se observa qué meses tienen mayor o menor demanda, normalmente ligados a:
  + verano → pico
  + invierno → caída

#### 10. Tipos de usuarios

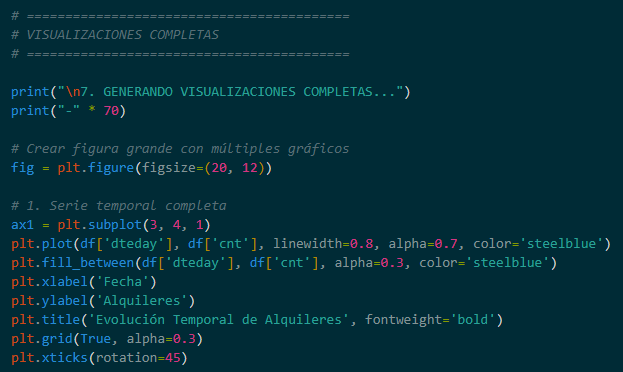


* Aquí se separa la demanda total en:
  + Usuarios casuales (turistas o recreación)
  + Usuarios registrados (uso frecuente)
* También se calcula el ratio registrados/casuales.

### Bloque Principal De Visualizaciones (12 gráficos)

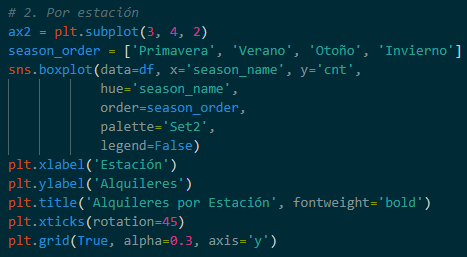
Este script genera un panel completo con:

#### Gráfico 1: Serie temporal



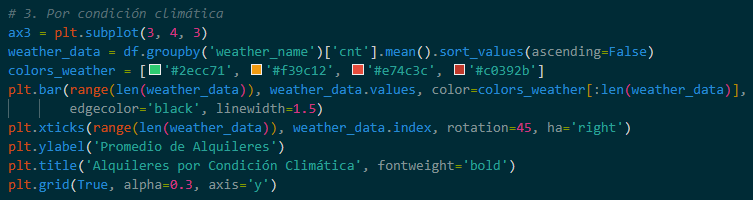
* Muestra la evolución diaria de alquileres y permite detectar tendencias crecientes.

#### Gráfico 2: Boxplot por estación (categórico)



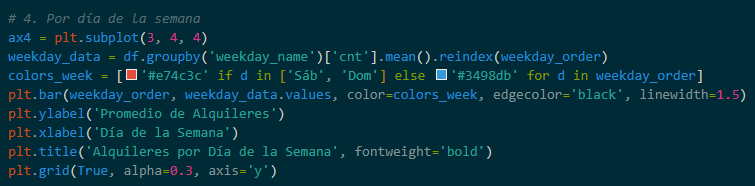
* Boxplot categórico → compara distribución de alquileres entre estaciones.

#### Gráfico 3: Barras por clima



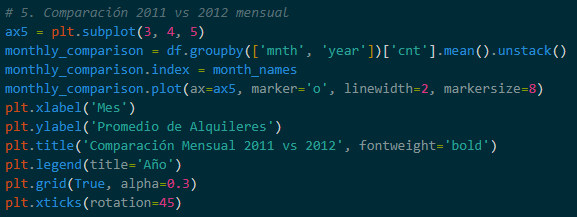
* Resume el promedio de alquileres según el clima.

#### Gráfico 4: Barras por día de semana



* Permite observar comportamiento semanal.

#### Gráfico 5: Comparación mensual por año



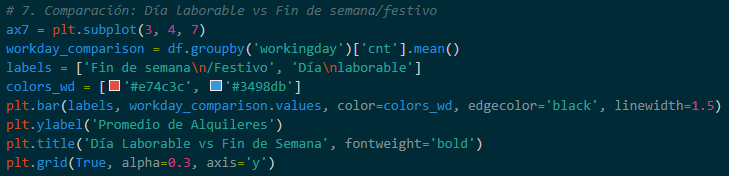
* Detecta si 2012 supera consistentemente a 2011.

#### Gráfico 6: Scatterplot temp vs cnt por estación



* Muestra relación temperatura-demanda diferenciada por estación.

#### Gráfico 7: comparación: Día laborable vs Fin de semana/festivo



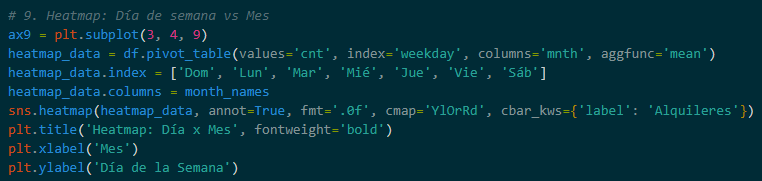
* Compara el promedio de alquileres entre días laborables y no laborables, permitiendo identificar diferencias de uso según la rutina semanal.

#### Gráfico 8: Usuarios casuales vs registrados



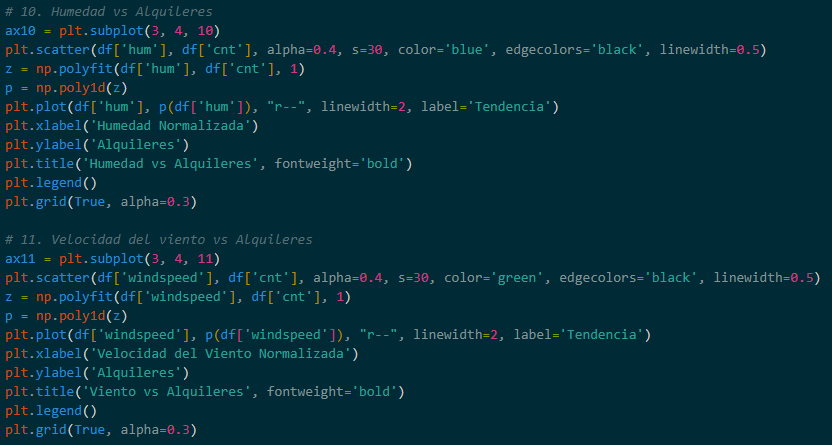
* Compara la contribución promedio de usuarios casuales y registrados. Se observa que los usuarios registrados representan la mayor proporción de alquileres, indicando un uso frecuente y regular del servicio.

#### Gráfico 9: Heatmap día vs mes



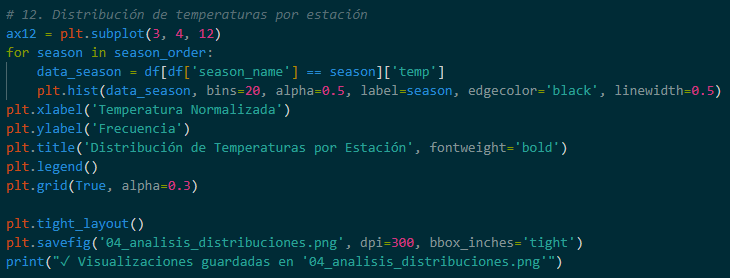
* Identifica combinaciones críticas como:
  + meses pico
  + días con mayor demanda

#### Gráficos 10 y 11: Scatterplots con regresión



* Evalúan impacto de:
  + humedad
  + viento

#### Gráfico 12: Histograma de temperaturas por estación



* Compara cómo cambia el clima en cada estación.

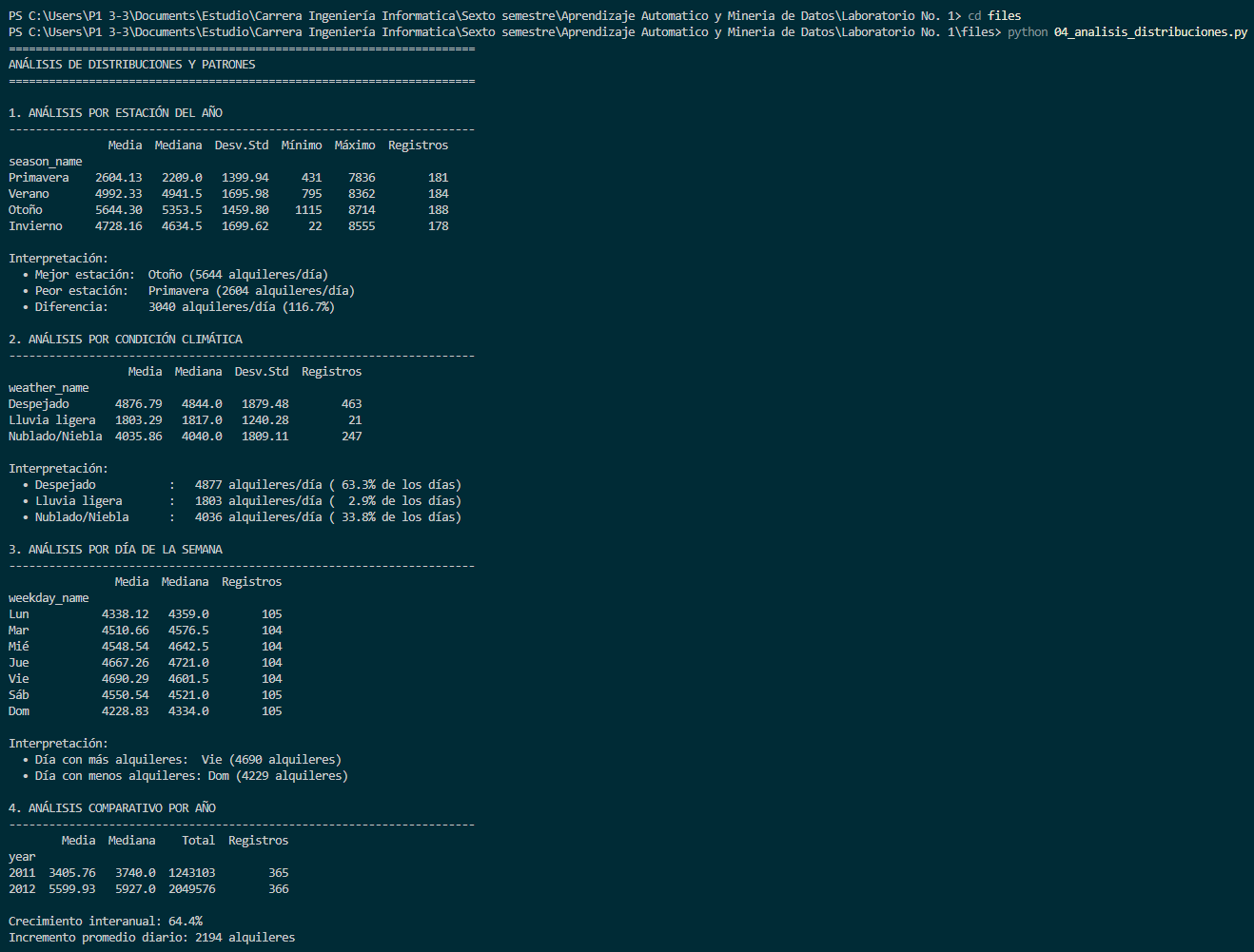
#### Resumen

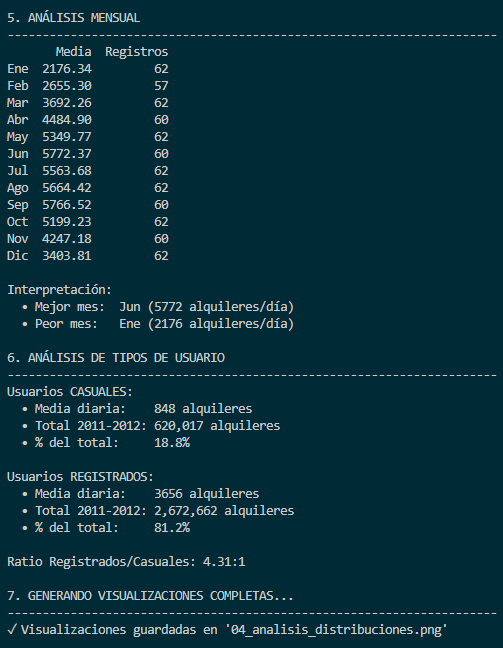


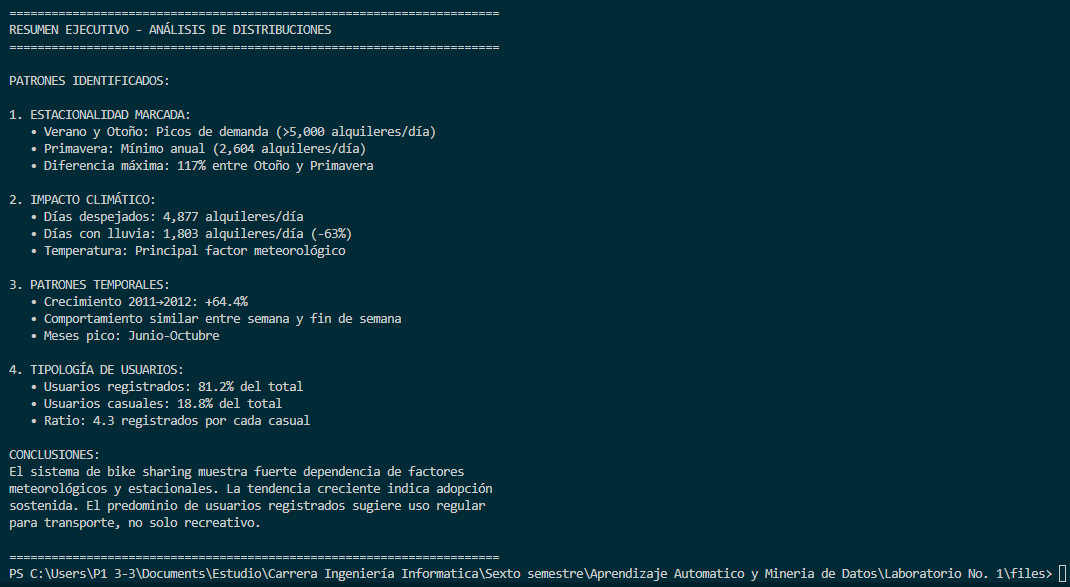
* El script termina imprimiendo conclusiones automáticas:
  + fuerte estacionalidad
  + impacto climático significativo
  + crecimiento anual
  + predominio de usuarios registrados

#### Resultados En Terminal

A continuación, las imágenes del resultado en consola:

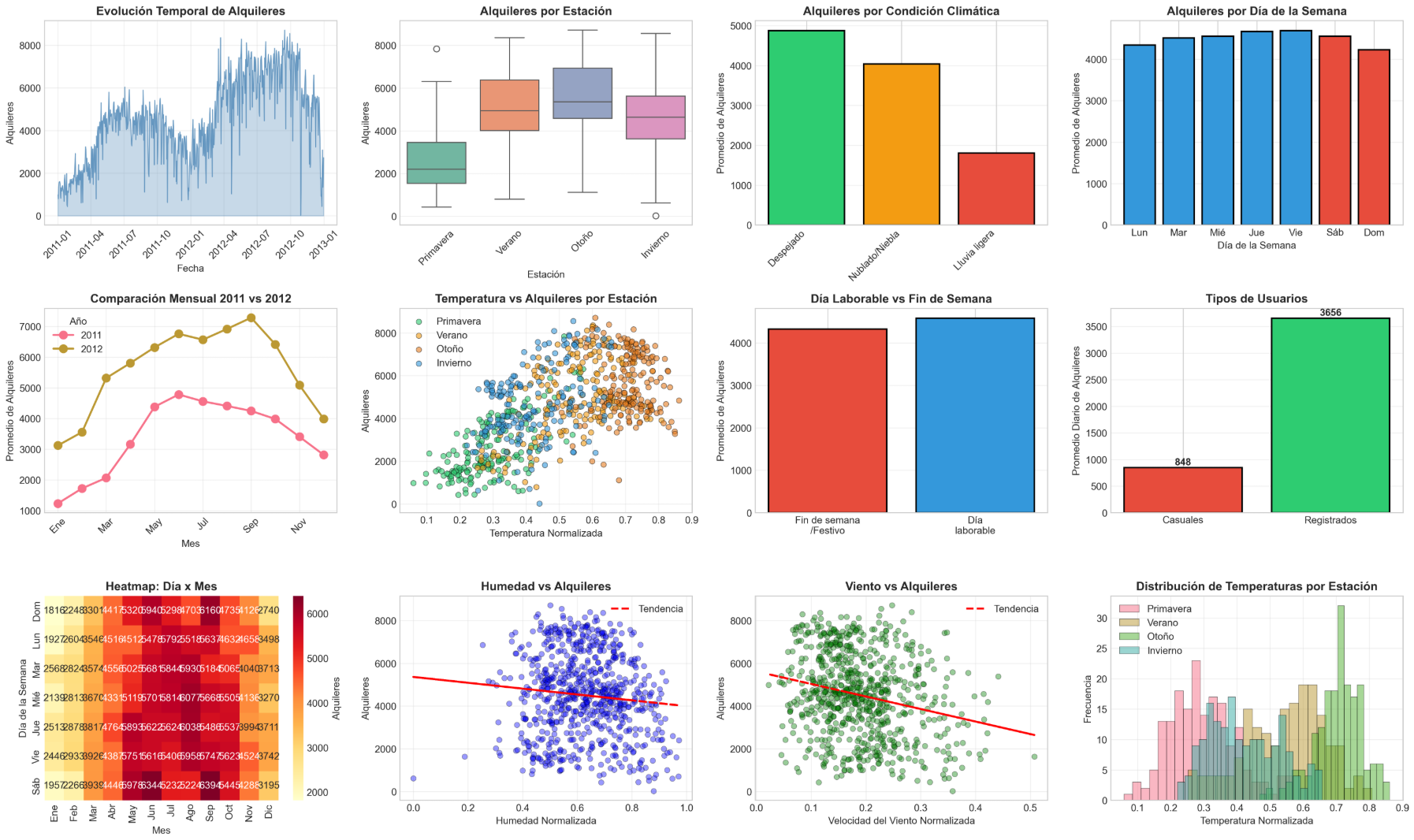






#### Imágenes Gráficos Implementados

A continuación, la imagen de los gráficos implementados



## División de Datos - Entrenamiento y Validación (Script 05)

Una vez realizado el análisis exploratorio y comprendida la estructura del dataset, el siguiente paso fundamental en cualquier proyecto de Machine Learning es la división de los datos en conjuntos de entrenamiento y validación.

Esta etapa es crítica porque:

* Permite entrenar el modelo con una parte de los datos.
* Evalúa su capacidad de generalización con datos no vistos.
* Evita sobreajuste (overfitting).
* Simula el comportamiento real del modelo en producción.

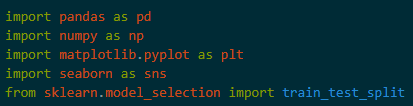
En este caso particular, el dataset representa una serie temporal diaria (2011–2012), lo que implica que el orden cronológico es fundamental. Por ello, se implementan dos estrategias de división:

* División temporal (recomendada)
* División aleatoria (solo comparativa)

El objetivo es demostrar por qué, en problemas temporales, la división aleatoria puede generar data leakage y producir evaluaciones irreales del modelo.

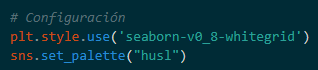
### Explicación detallada del código

#### Importaciones



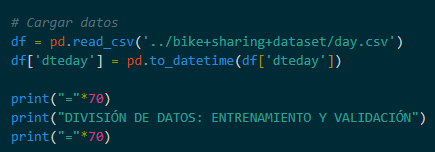
* pandas → manipulación de datos.
* numpy → usado para np.arange en el gráfico 5.
* matplotlib y seaborn → visualizaciones.
* train\_test\_split → división aleatoria.

#### Configuración



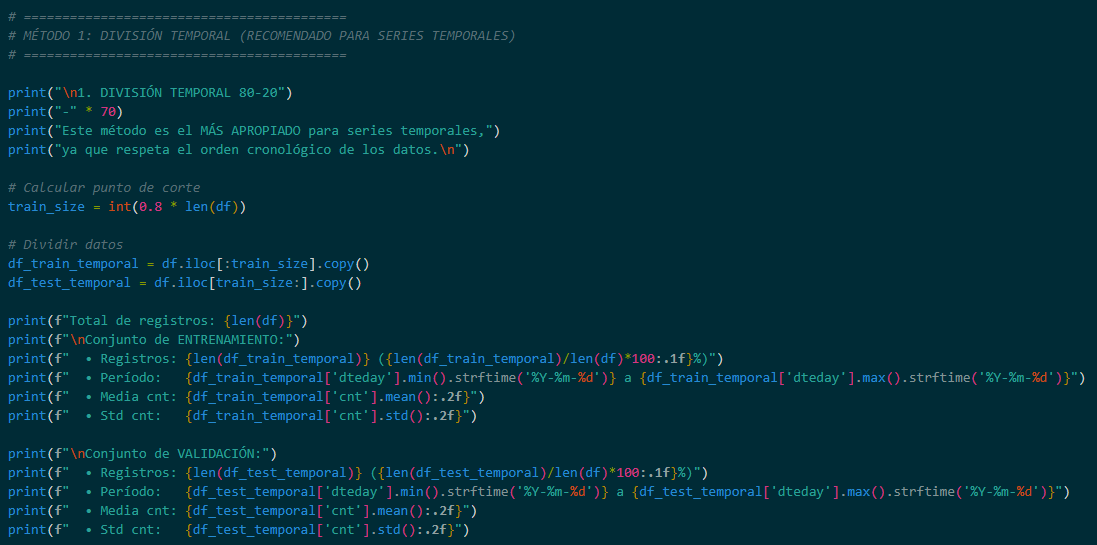
* Aplica un estilo predefinido de Matplotlib basado en Seaborn.

#### Carga de datos



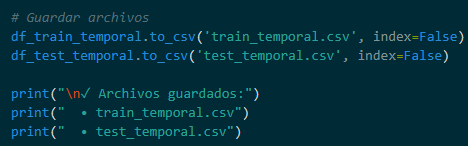
* Se convierte la columna fecha a formato datetime para poder:
  + Ordenar correctamente
  + Mostrar períodos
  + Graficar series temporales

##### Método 1: División Temporal



* Aquí se hace:
  + 80% primeros registros → entrenamiento
  + 20% últimos registros → validación
* IMPORTANTE:
  + Se usa iloc para mantener el orden cronológico.
  + Esto significa:
    - El modelo se entrena con el pasado.
    - Se valida con el futuro.
    - Se evita fuga de información (data leakage).

###### Guardado de archivos



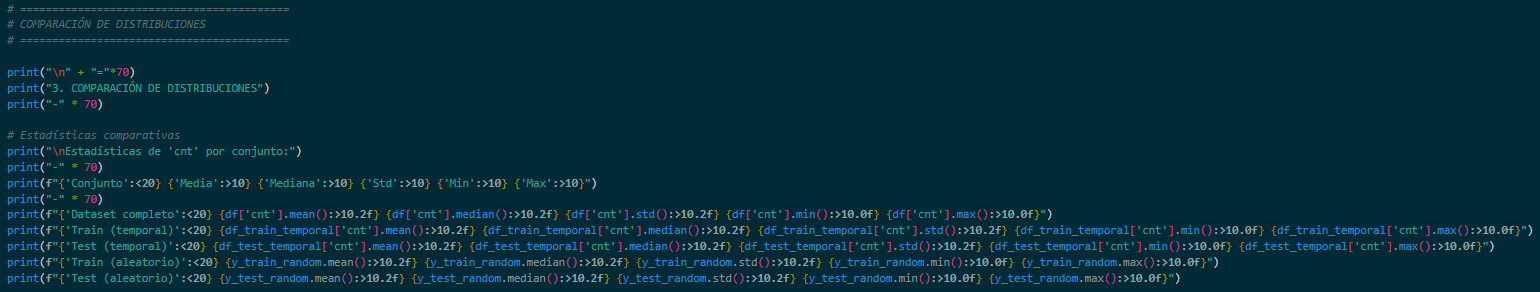
* Se generan los archivos que luego usarás para modelización.
* Esto es una buena práctica porque:
  + Separa claramente análisis de modelado.
  + El dataset queda reproducible.

##### Método 2: División Aleatoria (Alternativa No Recomendada)



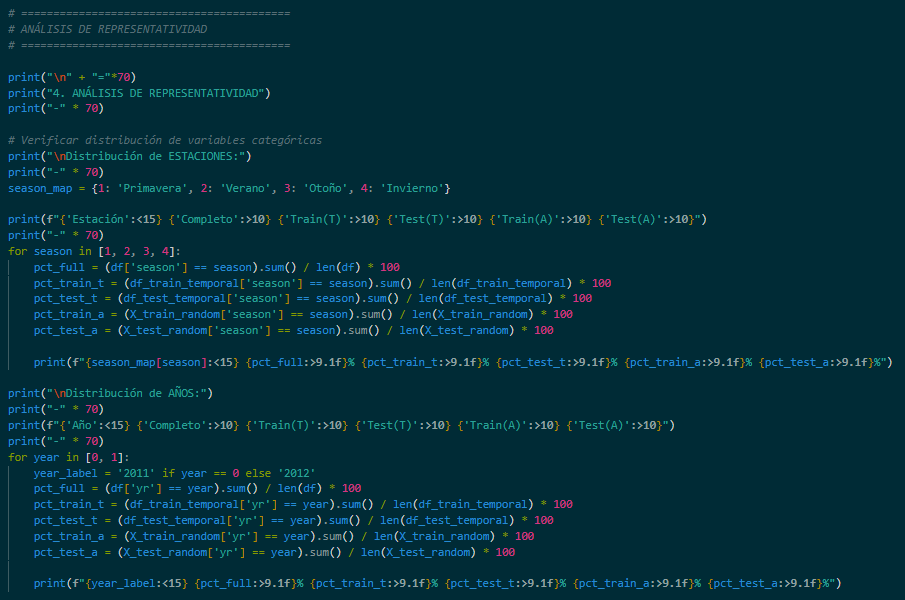
* Aquí ocurre algo peligroso en series temporales:
* Se mezclan datos de 2011 y 2012.
* El modelo puede entrenar con datos del futuro.
* Luego “valida” con datos del pasado.
* Eso es data leakage.
* En un problema real esto produciría métricas artificialmente buenas.

##### Comparación De Distribuciones



* Aquí se compara:
  + Media
  + Mediana
  + Desviación estándar
  + Mínimo
  + Máximo
* Entre:
  + Dataset completo
  + Train temporal
  + Test temporal
  + Train aleatorio
  + Test aleatorio
* Esto demuestra si los conjuntos son representativos.

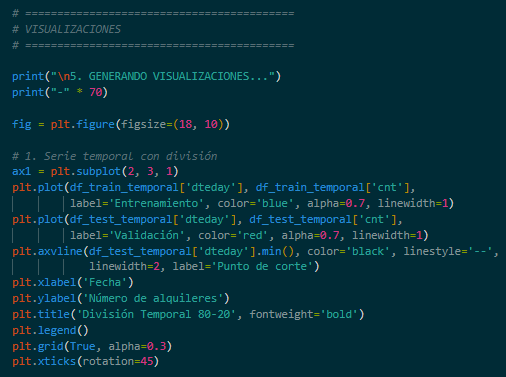
##### Análisis De Representatividad



* Aquí se verifica que:
  + Las estaciones estén balanceadas.
  + Los años estén representados.
* Muy importante en series temporales porque:
  + El conjunto temporal tendrá:
    - Más datos 2011 en train
    - Solo final 2012 en test

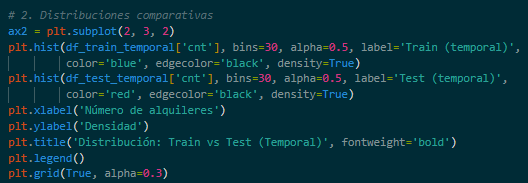
#### Visualizaciones

##### Serie temporal con división



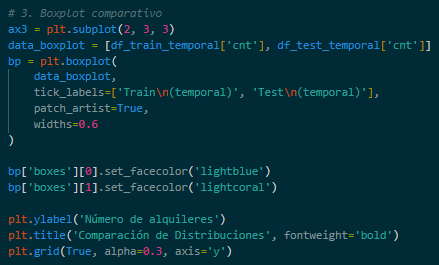
* Muestra:
  + Línea azul → entrenamiento
  + Línea roja → validación
  + Línea vertical → punto de división
* Esto permite ver claramente que el modelo se validará en datos posteriores.

##### Histogramas comparativos



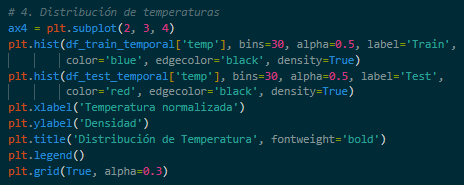
* Compara distribución de cnt entre:
  + Train temporal
  + Test temporal
* Sirve para verificar que no haya cambios drásticos.

##### Boxplot comparativo



* Comparación visual de:
  + Medianas
  + Rango intercuartílico
  + Outliers
* Entre train y test temporal.

##### Distribución de temperatura



* Verifica si la variable meteorológica principal tiene distribución similar en ambos conjuntos.

##### Distribución de estaciones



* Gráfico de barras comparativo.
* Evalúa representatividad categórica.
* Muy importante para evitar sesgo.

##### Evolución mensual por conjunto



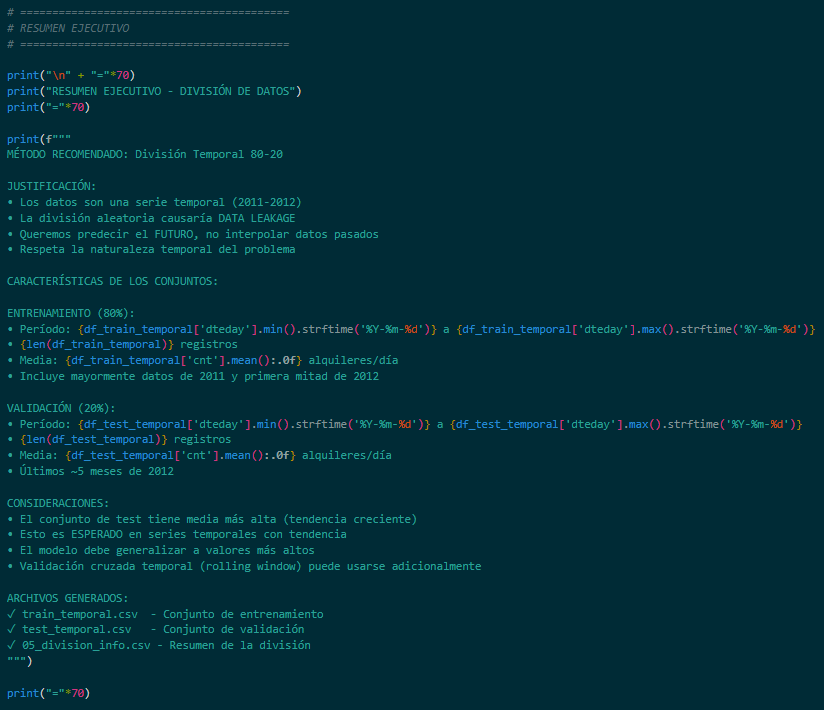
* Compara promedio mensual entre train y test.
* Aquí se ve claramente la tendencia creciente.

### Guardado de información estructurada



* Se genera un archivo resumen profesional con:
  + Método
  + Conjunto
  + Número de registros
  + Media
  + Desviación estándar
* Excelente para documentación formal.

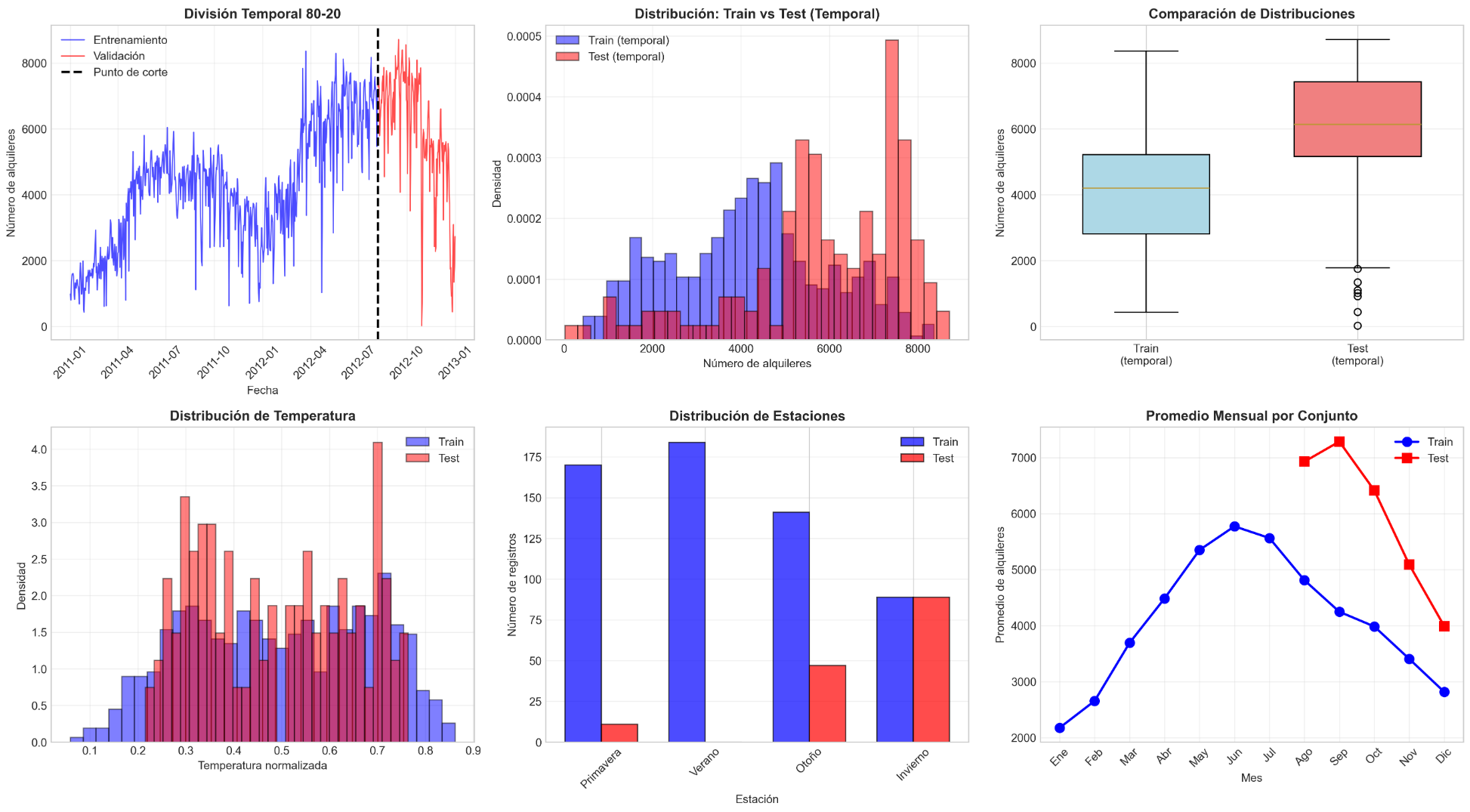
### Resumen Técnico Del Script



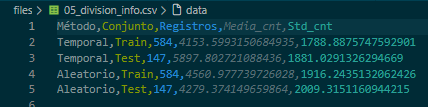
* Implementa división temporal correcta para series temporales.
* Demuestra por qué la división aleatoria no es adecuada.
* Evalúa representatividad estadística.
* Genera visualizaciones comparativas.
* Guarda datasets listos para modelado.

### Imágenes Gráficos Implementados

A continuación, la imagen de los gráficos que se han desarrollado:

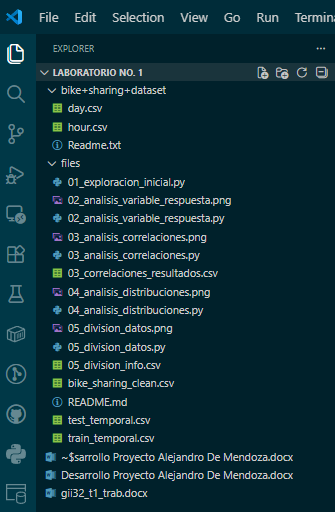


### Imagen archivo CSV generado



## Arquitectura Final de la Implementación

Ahora con base en esta ejecución a continuación doy a conocer la arquitectura final de esta implementación para dar total claridad frente al desarrollo, la imagen es la siguiente:



Entonces como se indicó anteriormente en la carpeta bike+sharing+dataset ubicada en la raíz del proyecto Laboratorio No. 1, tenemos los datos descargados del link remitido en la actividad los cuales son los siguientes:

* day.csv
* hour.csv
* readme.txt

Por otra parte, tenemos la carpeta de nombre “files”, ubicada de igual manera en la raíz del proyecto Laboratorio No. 1, donde están los scripts desarrollados de mi parte con los siguientes nombres:

* 01\_exploracion\_inicial.py: visión general del conjunto de datos, calidad básica de la información y detección de problemas.
* 02\_analisis\_variable\_respuesta.py: estudio en profundidad la variable respuesta
* 03\_analisis\_correlaciones.py: evaluar la relación existente entre dicha variable y el resto de variables del dataset.
* 04\_analisis\_distribuciones.py: estudiar cómo se distribuye la variable respuesta bajo diferentes condiciones.
* 05\_division\_datos.py: división de los datos en conjuntos de entrenamiento y validación.
* README.md: Archivo con la explicación completa del proyecto de manera técnica.
* En esta carpeta con la implementación de los códigos como podemos ver se genero el desarrollo de los siguientes archivos:
  + bike\_sharing\_clean.csv: Dataset limpio generado tras la exploración inicial, sin valores faltantes ni duplicados.
  + 02\_analisis\_variable\_respuesta.png: Gráficos descriptivos de la variable objetivo cnt (histograma, boxplot, Q-Q plot, densidad).
  + 03\_analisis\_correlaciones.png: Visualizaciones de correlaciones entre variables predictoras y cnt, incluyendo heatmap y relaciones clave.
  + 03\_correlaciones\_resultados.csv: Tabla con los coeficientes de correlación Pearson y Spearman para cada variable respecto a cnt.
  + 04\_analisis\_distribuciones.png: Conjunto de gráficos que muestran patrones de alquileres por estación, clima, mes, usuarios y tendencias.
  + 05\_division\_datos.png: Visualización de la división de datos en entrenamiento y validación, comparando distribuciones y períodos.
  + 05\_division\_info.csv: Resumen numérico de la partición train/test (registros, medias y desviaciones estándar por método).
  + test\_temporal.csv: Conjunto de entrenamiento (80%) generado con división temporal, usado para entrenar el modelo con datos del pasado.
  + train\_temporal.csv: Conjunto de validación/prueba (20%) generado con división temporal, usado para evaluar el modelo con datos futuros.

Finalmente tenemos los archivos ubicados en la raíz del proyecto laboratorio No. 1 que son los siguientes:

* Desarrollo Proyecto Alejandro De Mendoza.docx: Desarrollo de la documentación completa del laboratorio en Word.
* gii32\_t1\_trab.docx: Actividad a desarrollar.

Con lo que se finaliza la explicación de la arquitectura final de este desarrollo. Y con este último punto doy por finalizado el desarrollo de este laboratorio.

# CONCLUSIONES DE LA ACTIVIDAD

Para concluir este trabajo, puedo afirmar que el desarrollo del análisis descriptivo del Bike Sharing Dataset no solo permitió comprender el comportamiento de la demanda de bicicletas en Washington D.C. durante el periodo 2011–2012, sino que también evidenció la importancia de estructurar adecuadamente un proceso de análisis de datos. A diferencia de un enfoque monolítico, donde todo el procedimiento se concentra en un único script, la implementación modular utilizada en este proyecto facilitó una construcción progresiva del conocimiento, permitiendo analizar cada componente del problema de forma independiente, organizada y metodológicamente clara. Esta separación por fases; exploración inicial, análisis de la variable respuesta, estudio de correlaciones, análisis de distribuciones y división de datos, aportó trazabilidad, claridad conceptual y mayor control sobre cada decisión tomada. Desde el punto de vista técnico, la calidad estructural del dataset resultó adecuada para el análisis, ya que no se detectaron valores faltantes ni registros duplicados. Esto permitió centrar el esfuerzo en la interpretación estadística y en la identificación de patrones relevantes, en lugar de invertir tiempo significativo en procesos de limpieza compleja. La variable respuesta cnt, que representa el número total de bicicletas alquiladas por día, mostró una distribución aproximadamente simétrica, aunque los tests formales de normalidad indicaron desviaciones estadísticamente significativas. Este hallazgo es coherente con datos reales de comportamiento humano, donde rara vez se cumple la normalidad perfecta, y sugiere que futuras etapas de modelización deberán considerar transformaciones o modelos robustos dependiendo del enfoque seleccionado. El análisis de correlaciones permitió identificar claramente los factores más influyentes sobre la demanda. En particular, la temperatura (temp) y el año (yr) mostraron relaciones fuertes con la variable objetivo, lo que evidencia que tanto los factores meteorológicos como la evolución temporal del sistema influyen significativamente en el número de alquileres. Asimismo, se detectó una colinealidad extremadamente alta entre temp y atemp, lo cual constituye un hallazgo importante desde el punto de vista metodológico, ya que la inclusión simultánea de ambas variables podría generar inestabilidad en modelos lineales. Este tipo de análisis previo fortalece la calidad del modelado posterior, evitando errores conceptuales frecuentes como la redundancia informativa entre predictores.

Por otra parte, el análisis de distribuciones confirmó la existencia de patrones estacionales marcados, con mayor demanda en verano y otoño, así como una reducción considerable en condiciones climáticas adversas. También se evidenció un crecimiento significativo entre 2011 y 2012, lo que sugiere una adopción progresiva del sistema de bicicletas compartidas. Este crecimiento implica que el fenómeno no es estático, sino dinámico, y que cualquier modelo predictivo debe contemplar la tendencia temporal como componente fundamental.

Finalmente, la decisión de aplicar una división temporal 80–20 para los conjuntos de entrenamiento y validación demuestra una comprensión adecuada de la naturaleza del problema. Al tratarse de una serie temporal, la partición aleatoria podría generar data leakage y producir resultados artificialmente optimistas. En cambio, la división cronológica respeta la estructura real del fenómeno y simula un escenario auténtico de predicción futura, fortaleciendo la validez del proceso de evaluación.

En síntesis, esta actividad no solo permitió aplicar técnicas estadísticas y de análisis exploratorio de datos, sino que también reforzó la importancia de la metodología, la organización del código y la interpretación crítica de los resultados. Más allá de los números y gráficos obtenidos, el proyecto evidencia cómo un análisis estructurado y bien fundamentado constituye la base indispensable para desarrollar modelos predictivos sólidos, confiables y coherentes con la realidad del problema estudiado.

# BIBLIOGRAFÍA

A continuación, la bibliografía implementada en este desarrollo:

* Tema 1. Introducción a los procesos de descubrimiento del conocimiento. Aprendizaje Automático y Minería de Datos (COLGII) - PER 15746 - Enero 2026.
* Tema 2. Conceptos básicos de aprendizaje automático. Aprendizaje Automático y Minería de Datos (COLGII) - PER 15746 - Enero 2026.
* Tema 3. Exploración y preprocesamiento de datos. Aprendizaje Automático y Minería de Datos (COLGII) - PER 15746 - Enero 2026.
* Clases virtuales con el profesor Ing. Rogerio Orlando Beltrán Castro.
* Fanaee-T, H., & Gama, J. (2013). Event labeling combining ensemble detectors and background knowledge. Progress in Artificial Intelligence, 2(2–3), 113–127. Springer Berlin Heidelberg. https://doi.org/10.1007/s13748-013-0040-3
* James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: With applications in Python (2nd ed.). Springer.
* Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). Introduction to linear regression analysis (6th ed.). Wiley.
* Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). Applied linear statistical models (5th ed.). McGraw-Hill Irwin.
* Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). Biometrika, 52(3–4), 591–611.
* Pearson, K. (1896). Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. Philosophical Transactions of the Royal Society of London, 187, 253–318.
* Spearman, C. (1904). The proof and measurement of association between two things. The American Journal of Psychology, 15(1), 72–101.
* Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning (2nd ed.). Springer.
* Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

# AGRADECIMIENTO

Finalmente, deseo expresar mi más sincero agradecimiento al profesor Ing. Rogerio Orlando Beltrán Castro, por los conocimientos, orientación y acompañamiento brindados durante el desarrollo de esta actividad. Sus explicaciones y aportes en el área de aprendizaje automático y minería de datos fueron fundamentales para comprender la importancia del análisis exploratorio, el preprocesamiento y la correcta preparación de los datos como base para cualquier proceso de modelización. Gracias a los conceptos compartidos en clase, fue posible realizar este trabajo de manera estructurada, aplicando buenas prácticas tanto en el análisis estadístico como en la organización modular del código. Sin duda, esta experiencia fortaleció significativamente mi formación académica y mi entendimiento de los procesos de descubrimiento del conocimiento en ciencia de datos.

**¡¡¡Mil gracias profesor!!!**

**Respetuosamente,**

**Alejandro De Mendoza**