

MAGNA

INSTITUCIÓN DE ESPECIALIZACIÓN PROFESIONAL



HERRAMIENTAS DE VISUALIZACIÓN DE DATOS EN MANTENIMIENTO CON POWER BI Y R

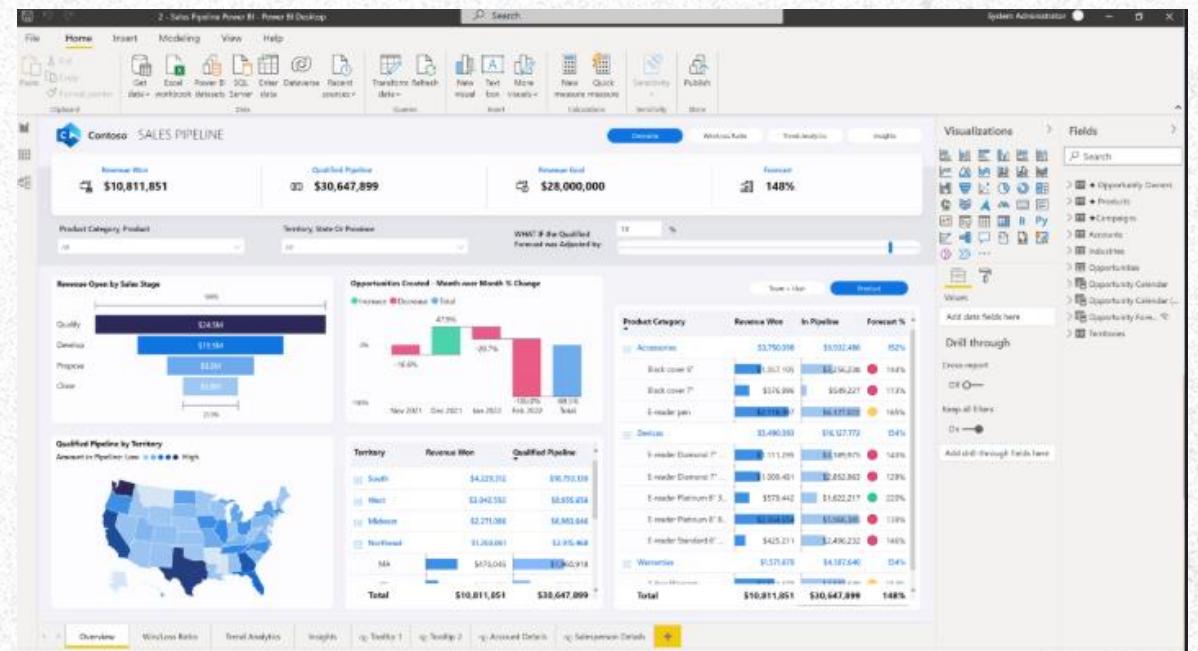
Ing. Breyunner Chávez

Introducción a Power BI

¿Qué es Power BI?

Power BI es una herramienta de análisis de datos y visualización interactiva desarrollada por Microsoft, que permite transformar datos en información accionable mediante gráficos, informes y paneles visuales.

Su propósito es ayudar a los usuarios a tomar decisiones basadas en datos. Esto se logra consolidando información de diversas fuentes, limpiándola, modelándola y presentándola de manera comprensible y visualmente atractiva.



Introducción a Power BI

¿Qué es Power BI?

Características Principales de Power BI

1. Visualización Interactiva:
 - Permite explorar los datos con gráficos dinámicos, mapas, tablas y más.
 - Los usuarios pueden aplicar filtros, segmentar datos por categorías y analizar patrones de manera intuitiva.
2. Conexión con Múltiples Fuentes de Datos:
 - Power BI puede conectarse a más de 100 tipos de fuentes, como: Bases de datos (SQL Server, MySQL, PostgreSQL), Archivos (Excel, CSV, JSON), Servicios en la nube (Azure, Google Analytics, Salesforce), APIs web.
 - Ejemplo en mantenimiento: Combinar datos de sensores de máquinas (IoT) con información financiera para optimizar costos.

Introducción a Power BI

¿Qué es Power BI?

Características Principales de Power BI

3. Análisis Automatizado:

- Ofrece funciones como la detección de anomalías y el análisis de tendencias.
- Ejemplo: Identificar aumento inusual en tiempos de inactividad de máquinas.

4. Actualización en Tiempo Real:

- Los informes pueden actualizarse automáticamente al conectar datos en vivo.
- Ejemplo: Un panel en una planta industrial que muestra la productividad de las máquinas en tiempo real.

5. Facilidad de Uso:

- Diseñada para usuarios sin conocimientos avanzados de programación o análisis de datos.
- La interfaz intuitiva hace que cualquiera pueda trabajar rápidamente.

Introducción a Power BI

¿Qué es Power BI?

Ventajas de Power BI

1. Accesibilidad:
 - Disponible como aplicación de escritorio, servicio web y aplicación móvil.
 - Ejemplo: Consultar un reporte desde el celular mientras estás en una planta de mantenimiento.
2. Integración con Herramientas de Microsoft:
 - Se integra perfectamente con Excel, Teams, SharePoint y OneDrive.
 - Ejemplo: Subir informes de Power BI directamente a un sitio de SharePoint para compartirlos con el equipo.

Introducción a Power BI

¿Qué es Power BI?

Ventajas de Power BI

3. Escalabilidad:

- Se adapta tanto a pequeñas empresas como a grandes organizaciones.
- Ejemplo: Una pequeña empresa puede usar Power BI para analizar inventarios, mientras una multinacional lo usa para monitorear fábricas globales.

4. Colaboración:

- Permite compartir reportes con equipos en tiempo real.
- Ejemplo: El equipo de mantenimiento puede discutir hallazgos directamente en los informes publicados.

¿Qué es Power BI?

Aplicaciones de Power BI en el Ámbito de Mantenimiento

Power BI tiene un papel crucial en el mantenimiento industrial, ya que permite analizar y optimizar las operaciones de las máquinas y equipos. A continuación, se presentan ejemplos concretos:

1. Análisis de Tiempos de Falla

- Objetivo: Determinar el tiempo promedio que transcurre entre fallos consecutivos de un equipo.
- Beneficio: Ayuda a identificar equipos que necesitan mantenimiento preventivo con mayor frecuencia.
- Ejemplo Visual: Un gráfico de líneas que muestre el historial de fallas de una máquina.

¿Qué es Power BI?

2. Costos de Mantenimiento

- Objetivo: Comparar los costos de mantenimiento preventivo y correctivo para tomar decisiones más económicas.
- Beneficio: Identificar estrategias que reduzcan el gasto en mantenimiento.
- Ejemplo Visual: Un gráfico de barras que compare los costos mensuales de mantenimiento por equipo.

3. Frecuencia de Revisiones

- Objetivo: Estudiar patrones de uso y desgaste de equipos para planificar inspecciones más efectivas.
- Beneficio: Aumenta la disponibilidad operativa y evita tiempos de inactividad imprevistos.
- Ejemplo Visual: Un gráfico circular que muestra el porcentaje de inspecciones realizadas por tipo de equipo.

Componentes Principales de Power BI

Power BI está compuesto por varios módulos que facilitan cada etapa del análisis de datos. A continuación, se describen los tres principales:

1. Power Query

- Definición: Es el módulo de Power BI destinado a la extracción, transformación y carga de datos (ETL).
- Funciones principales:
 - ✓ Conectar datos desde múltiples fuentes.
 - ✓ Limpiar datos (eliminar duplicados, llenar valores faltantes).
 - ✓ Transformar datos (dividir columnas, convertir tipos de datos).
- Ejemplo en mantenimiento: Conectar datos de sensores IoT que miden temperaturas y revoluciones de máquinas, limpiarlos y prepararlos para análisis.

Componentes Principales de Power BI

2. Power Pivot

- Definición: Es el módulo que permite modelar los datos, crear relaciones entre tablas y realizar cálculos complejos utilizando el lenguaje DAX (Data Analysis Expressions).
- Funciones principales:
 - ✓ Crear relaciones entre tablas.
 - ✓ Definir columnas calculadas y medidas personalizadas.
 - ✓ Optimizar datos para consultas rápidas.
- Ejemplo en mantenimiento: Relacionar una tabla de equipos con otra de registros de fallos mediante el ID_Equipo. Calcular el tiempo promedio entre fallos (MTBF).

Componentes Principales de Power BI

3. Power View

- Definición: Es el módulo para crear visualizaciones interactivas, como gráficos, tablas dinámicas, mapas, entre otros.
- Funciones principales:
 - ✓ Diseñar paneles y reportes atractivos.
 - ✓ Filtrar datos de manera dinámica.
 - ✓ Compartir informes interactivos con otros usuarios.

¿Qué es la Limpieza de Datos?

La limpieza de datos es el proceso de preparar los datos crudos para análisis eliminando inconsistencias, errores, valores faltantes o duplicados, y transformándolos en un formato adecuado para su uso. Es una etapa fundamental en el análisis de datos, ya que asegura la calidad y confiabilidad de los resultados.

¿Qué es la Limpieza de Datos?

Importancia de la Limpieza de Datos en Mantenimiento

- Precisión del análisis:** En el contexto del mantenimiento, datos erróneos pueden generar análisis incorrectos que afecten las decisiones sobre el mantenimiento preventivo o correctivo. Ejemplo: Un valor erróneo en el tiempo de falla podría llevar a intervalos de mantenimiento incorrectos.
- Ahorro de tiempo y recursos:** Identificar y corregir problemas en los datos desde el inicio evita errores acumulativos en etapas posteriores del análisis. Ejemplo: En una planta industrial, limpiar datos de sensores reduce los falsos positivos en el monitoreo de fallos.
- Consistencia en los reportes:** Asegura que los informes de mantenimiento reflejen la realidad operacional.

Limpieza de Datos en Power Query

Power Query es el módulo de Power BI diseñado específicamente para la limpieza y transformación de datos. A continuación, se detallan las herramientas más relevantes:

1. Eliminación de Duplicados

- ¿Por qué es importante?: Los registros duplicados pueden inflar métricas y distorsionar el análisis. Ejemplo: Un registro duplicado de una falla podría aumentar falsamente la frecuencia de mantenimiento.
- Cómo hacerlo en Power Query: Selecciona la tabla en el Editor de Power Query. Ve a la pestaña Inicio y selecciona Quitar duplicados.
- Demostración práctica sugerida: Muestra una tabla con registros duplicados y aplica la función. Luego, observa cómo se reduce el número de filas.

Limpieza de Datos en Power Query

2. Rellenar Valores Faltantes

- ¿Por qué es importante? Los valores faltantes dificultan los cálculos y pueden generar errores al modelar datos. Ejemplo: Si faltan valores en la columna de costos de reparación, no se podrá calcular el costo promedio por equipo.
- Opciones comunes para llenar valores faltantes: Reemplazar con un valor predeterminado (“0” o “No disponible”). Usar el promedio, la mediana o el valor más cercano. Propagar valores hacia arriba o hacia abajo (útil en series temporales).
- Cómo hacerlo en Power Query: Selecciona la columna con valores faltantes. Ve a la pestaña Transformar y usa la opción Rellenar hacia abajo o Reemplazar valores.
- Demostración práctica sugerida: Muestra una tabla con datos faltantes y aplica diferentes métodos para llenarlos, explicando en qué casos usarías cada opción.

Limpieza de Datos en Power Query

3. Dividir Columnas por Delimitadores

- ¿Por qué es importante?: A menudo, los datos vienen combinados en una sola columna y es necesario separarlos para un análisis adecuado. Ejemplo: Separar una columna con el formato “Equipo - Área” en dos columnas: “Equipo” y “Área”.
- Cómo hacerlo en Power Query: Selecciona la columna a dividir. Ve a la pestaña Transformar y selecciona Dividir columna por delimitador. Elige el delimitador (guion, coma, espacio).
- Demostración práctica sugerida: Muestra una tabla con datos combinados, separa la información y observa el resultado.

Limpieza de Datos en Power Query

4. Transformar Tipos de Datos

- ¿Por qué es importante?: Un tipo de dato incorrecto puede causar errores en cálculos o visualizaciones. Ejemplo: Si una columna de fechas se interpreta como texto, no podrás analizar tendencias temporales.
- Tipos de datos más comunes en Power Query: Texto, Número entero o decimal, Fecha y hora, Booleano (verdadero/falso).
- Cómo hacerlo en Power Query: Selecciona la columna con el tipo de dato incorrecto. En la pestaña Transformar, selecciona Detección de tipo de datos. Cambia el tipo de dato manualmente si es necesario.

Limpieza de Datos en Power Query

Flujo de Limpieza de Datos en Power Query

1. Conectar las Fuentes de Datos: Importa datos de Excel, bases de datos, servicios en la nube, o archivos CSV.
2. Previsualizar y Evaluar la Calidad de los Datos: Usa las barras de calidad en Power Query para identificar valores faltantes, errores o duplicados.
3. Aplicar Transformaciones: Ordena los pasos de limpieza en el panel de consultas para documentar el flujo de trabajo. Las transformaciones realizadas en Power Query son no destructivas, es decir, los datos originales permanecen intactos.
4. Cargar los Datos Limpios en Power BI: Una vez finalizada la limpieza, los datos se cargan en Power BI Desktop para el modelado y la visualización.

Limpieza de Datos en Power Query

Casos de Uso de Limpieza de Datos en Mantenimiento

- Registro de Fallos:** Problema: Los registros contienen duplicados y formatos inconsistentes (fechas en diferentes formatos). Solución: Usar Power Query para eliminar duplicados y unificar el formato de fecha.
- Sensores IoT:** Problema: Los datos de sensores contienen valores faltantes y ruido. Solución: Usar Power Query para llenar los valores faltantes con promedios y filtrar valores atípicos.
- Inventario de Repuestos:** Problema: Las descripciones de piezas están combinadas en una sola columna. Solución: Dividir la columna en “Código de pieza” y “Descripción”.

Modelado de Datos en Power BI

¿Qué es el Modelado de Datos?

El modelado de datos en Power BI es el proceso de estructurar, relacionar y optimizar los datos para facilitar análisis efectivos y visualizaciones rápidas. Implica conectar tablas de diferentes fuentes, definir relaciones lógicas entre ellas y crear cálculos personalizados.

¿Qué es el Modelado de Datos?

Importancia del Modelado de Datos en Mantenimiento

- Eficiencia en consultas:** Un modelo bien estructurado permite realizar análisis complejos y consultas rápidas, sin necesidad de procesar datos en cada visualización. Ejemplo: Relacionar tablas de equipos, componentes y tiempos de mantenimiento permite analizar patrones rápidamente.
- Claridad en los datos:** Ayuda a simplificar conjuntos de datos complejos, creando jerarquías y medidas calculadas que son fáciles de interpretar. Ejemplo: Crear una métrica para "Costo total de reparación por máquina".
- Confiabilidad en las visualizaciones:** Las relaciones bien definidas aseguran que los gráficos y reportes no muestren datos incorrectos o incompletos.

Componentes del Modelado de Datos en Power BI

El modelado de datos en Power BI consta de varios elementos clave:

1. Tablas:

- Las tablas representan los datos importados desde diversas fuentes (Excel, bases de datos, APIs).
- Pueden ser tablas de hechos (contienen datos transaccionales, como registros de mantenimiento) o tablas de dimensiones (contienen descripciones o clasificaciones, como equipos y áreas).
- Ejemplo: Tabla de hechos: "Registros de fallos" (fechas, costos, tiempos de falla). Tabla de dimensiones: "Equipos" (ID de equipo, nombre, ubicación).
- **Demostración práctica sugerida: Muestra cómo cargar varias tablas relacionadas en Power BI y explorar sus estructuras.**

Componentes del Modelado de Datos en Power BI

2. Relaciones

- ¿Qué son las relaciones?: Las relaciones conectan tablas mediante campos comunes, como una clave primaria en una tabla y una clave externa en otra. Esto permite analizar datos combinados de diferentes tablas.
- Tipos de relaciones en Power BI:
 - Uno a muchos: El tipo más común; una fila de la tabla principal puede relacionarse con muchas filas en la secundaria. Ejemplo: Un equipo (tabla principal) puede tener múltiples registros de fallos (tabla secundaria).
 - Uno a uno: Una fila de una tabla se relaciona con exactamente una fila en otra. Ejemplo: Una tabla de "Detalles de equipos" se relaciona uno a uno con una tabla de "Ubicaciones".

Componentes del Modelado de Datos en Power BI

3. Jerarquías

- ¿Qué son las jerarquías?: Son estructuras que permiten organizar los datos en niveles para facilitar el análisis. Ejemplo en mantenimiento: Una jerarquía que va de Área -> Equipo -> Componente -> Fecha de mantenimiento.
- Cómo crear jerarquías en Power BI: En el panel de modelado, selecciona los campos que representan niveles de la jerarquía. Agrúpalos y asigna un nombre significativo a la jerarquía.

Medidas y Columnas Calculadas

1. Medidas

- Definición: Son cálculos dinámicos que se generan en el momento de ejecutar una consulta. Usan el lenguaje DAX (Data Analysis Expressions).
Ejemplo: Calcular el "Tiempo promedio entre fallos (MTBF)".
- Características: No ocupan espacio adicional en el modelo, ya que se calculan sobre la marcha. Se pueden usar en visualizaciones, tablas y gráficos.
- Ejemplo práctico: Crea una medida para calcular el tiempo promedio entre fallos:

$\text{MTBF} = \text{AVERAGE}(\text{'Registros de fallos'}[\text{Tiempo entre fallos}])$

Medidas y Columnas Calculadas

2. Columnas Calculadas

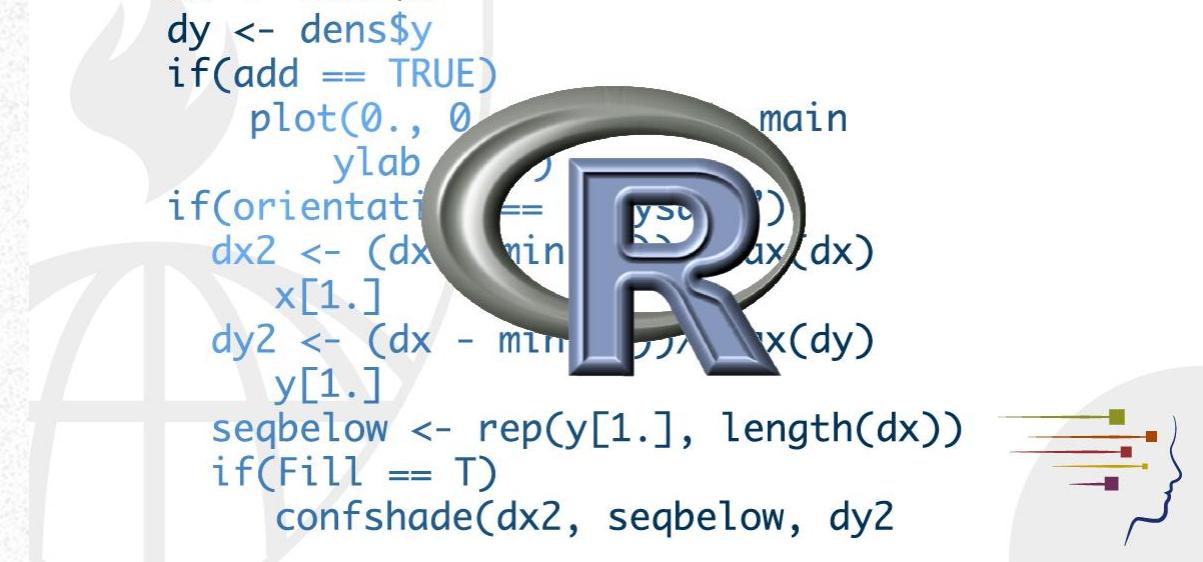
- Definición: Son campos adicionales creados directamente en una tabla mediante cálculos. Ejemplo: Calcular el costo por hora de mantenimiento basado en el costo total y la duración.
- Características: Se almacenan directamente en el modelo, por lo que ocupan espacio adicional. Son útiles para cálculos que deben hacerse antes de cargar los datos en las visualizaciones.
- Ejemplo práctico: Crea una columna calculada para el costo por hora de mantenimiento:

CostoPorHora = 'Registros de fallos'[Costo total] / 'Registros de fallos'[Horas de mantenimiento]

Introducción a la integración de R en Power BI

¿Qué es R?

R es un lenguaje de programación y entorno de software especializado en análisis estadístico y visualización avanzada de datos. Su flexibilidad y potencia lo han hecho una herramienta indispensable en áreas como ciencia de datos, modelado predictivo y generación de gráficos personalizados. Su IDE(entorno de desarrollo integrado) se llama Rstudio, R es el lenguaje y Rstudio la interfaz grafica



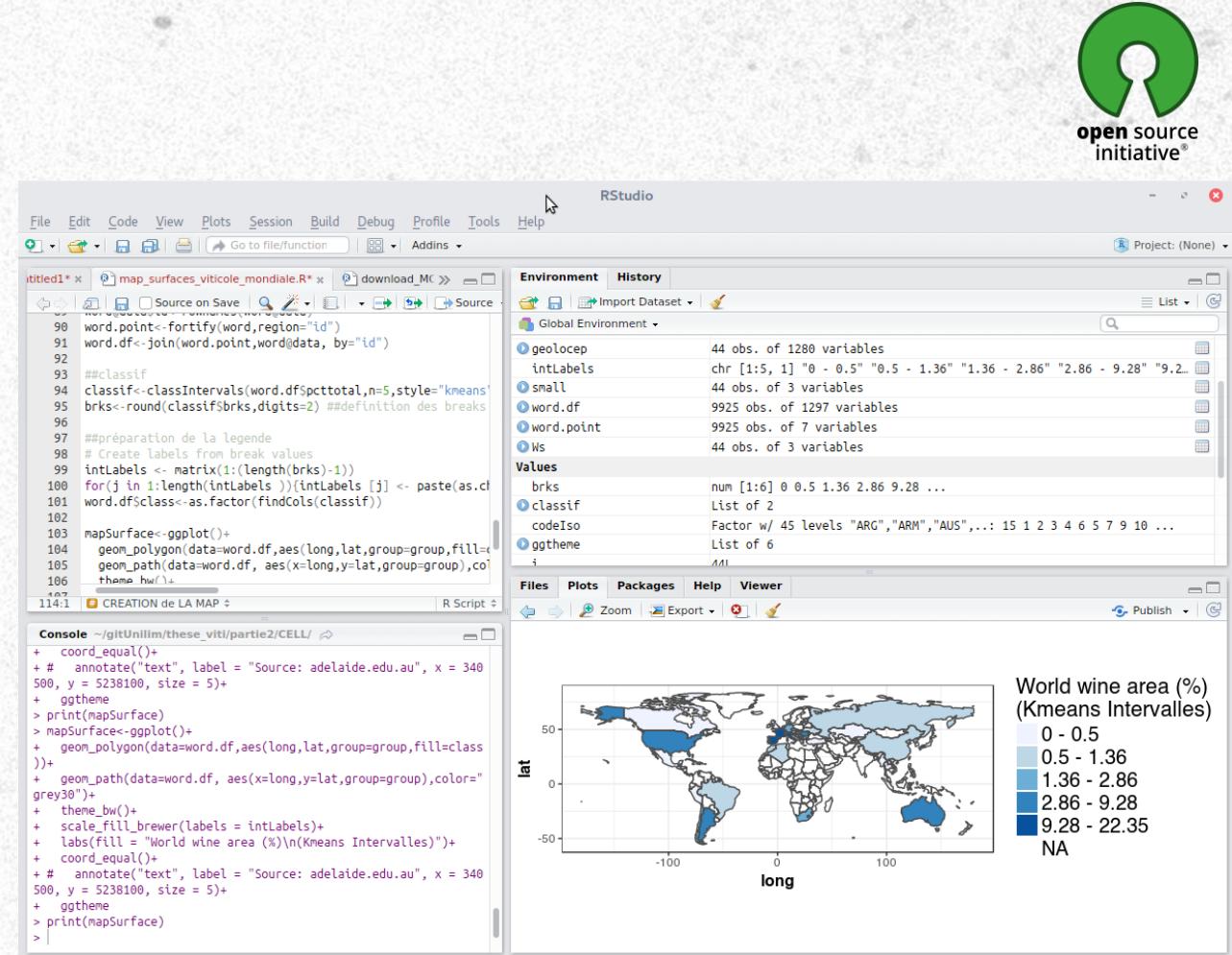
```
dens <- density(data, n = npts)
dx <- dens$x
dy <- dens$y
if(add == TRUE)
  plot(0., 0,
       ylab = "Density")
if(orientation == "vertical")
  dx2 <- (dx - min(dx)) / max(dx)
  x[1.]
  dy2 <- (dx - min(dx)) * max(dy)
  y[1.]
  seqbelow <- rep(y[1.], length(dx))
  if(Fill == T)
    confshade(dx2, seqbelow, dy2)
```

Introducción a la integración de R en Power BI

¿Qué es R?

Características Principales de R

- Lenguaje de código abierto con una extensa comunidad de usuarios.
- Diseñado para realizar cálculos estadísticos complejos y análisis de datos.
- Ofrece múltiples librerías para gráficos avanzados y modelado estadístico.
- Permite construir visualizaciones altamente personalizadas que van más allá de las capacidades estándar de muchas herramientas de BI



Introducción a la integración de R en Power BI

Compatibilidad de Power BI con R

Power BI permite integrar scripts de R para realizar tareas avanzadas que no se pueden lograr con las funcionalidades estándar de la herramienta.

Usos comunes de R en Power BI:

- Limpieza y transformación de datos:** R puede procesar y transformar datos directamente en Power BI usando su capacidad para manipular grandes conjuntos de datos de manera eficiente.
- Creación de gráficos avanzados:** Incluye visualizaciones personalizadas, como gráficos estadísticos complejos o gráficos interactivos que no están disponibles de forma nativa en Power BI.

Requisitos técnicos para la integración de R en Power BI:

- Se necesita tener R instalado en el sistema.
- Es necesario que Power BI reconozca el motor de R instalado para poder ejecutar scripts desde la plataforma.

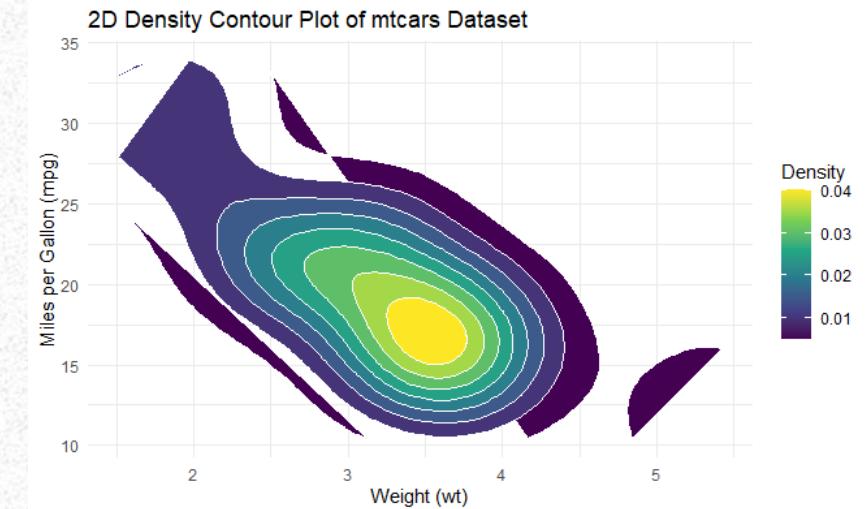
Herramientas de R para Visualización

Librerías principales para visualización en R

R ofrece una gran variedad de librerías para visualización que permiten generar gráficos avanzados. Algunas de las principales librerías incluyen:

1. **ggplot2**: Una de las librerías más populares para gráficos en R. Basada en la gramática de gráficos, permite crear visualizaciones estadísticas altamente personalizables.

Características: Soporte para gráficos como histogramas, gráficos de dispersión, gráficos de líneas y más. Posibilita la creación de capas sobre un gráfico base para añadir elementos personalizados.

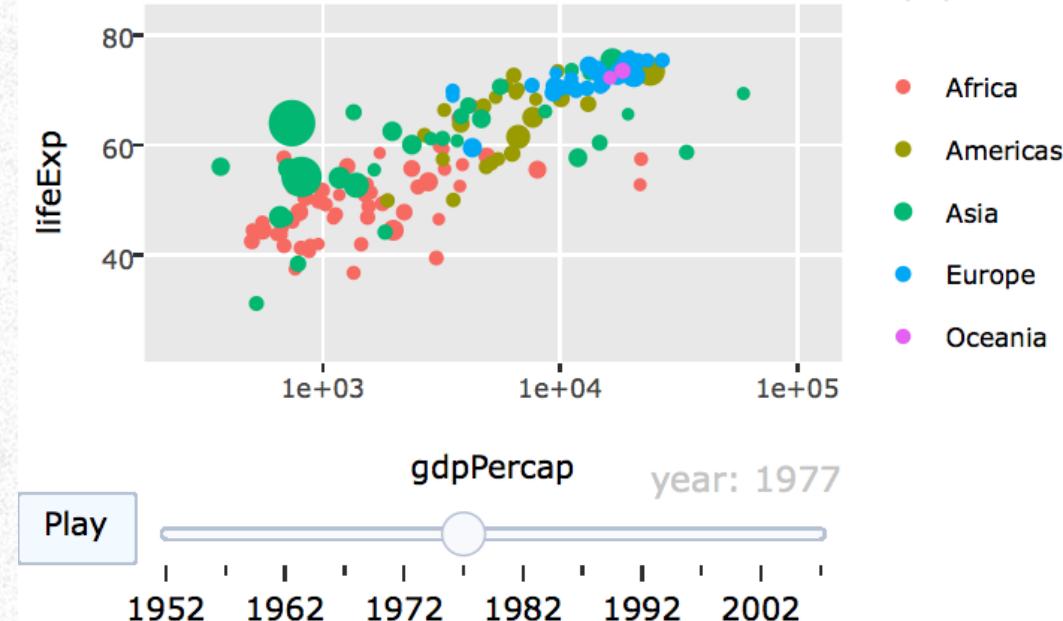


Herramientas de R para Visualización

Librerías principales para visualización en R

2. **plotly**: Ideal para gráficos interactivos que se pueden integrar en dashboards.

Características: Ofrece funcionalidades como zoom, desplazamiento y selecciones dinámicas. Compatible con gráficos 3D, mapas y otros formatos interactivos.



Herramientas de R para Visualización

Librerías principales para visualización en R

3. **gridExtra**: Permite combinar múltiples gráficos en una sola presentación. Muy útil cuando se necesitan paneles comparativos dentro de un visual.
4. **shiny**: Diseñada para construir aplicaciones interactivas que permiten el análisis dinámico de datos. Dentro de Power BI, los scripts de Shiny pueden ser utilizados para generar gráficos que responden a filtros y entradas del usuario.



Configuración en Power BI

Configuración Básica:

Para utilizar R en Power BI, primero debemos realizar una configuración inicial del entorno.

1. Instalación de R, Rstudio y bibliotecas necesarias.

Primero instalamos R, luego Rstudio, abrimos este ultimo y damos click en packages o también podemos instalar desde el editor de codigo

The screenshot shows the RStudio interface with the 'Packages' tab selected. It displays two ways to install R packages:

1ra forma de instalar bibliotecas: A screenshot of the 'Install Packages' dialog. It shows the 'Install from:' dropdown set to 'Repository (CRAN)', the 'Packages' input field containing 'ggplot2', and the 'Install to Library:' dropdown set to 'C:/Users/Fernando/AppData/Local/R/win-library/4.4 [Default]'. The 'Install dependencies' checkbox is checked. At the bottom are 'Install' and 'Cancel' buttons.

2da forma de instalar bibliotecas: A screenshot of the RStudio code editor showing the following R code:

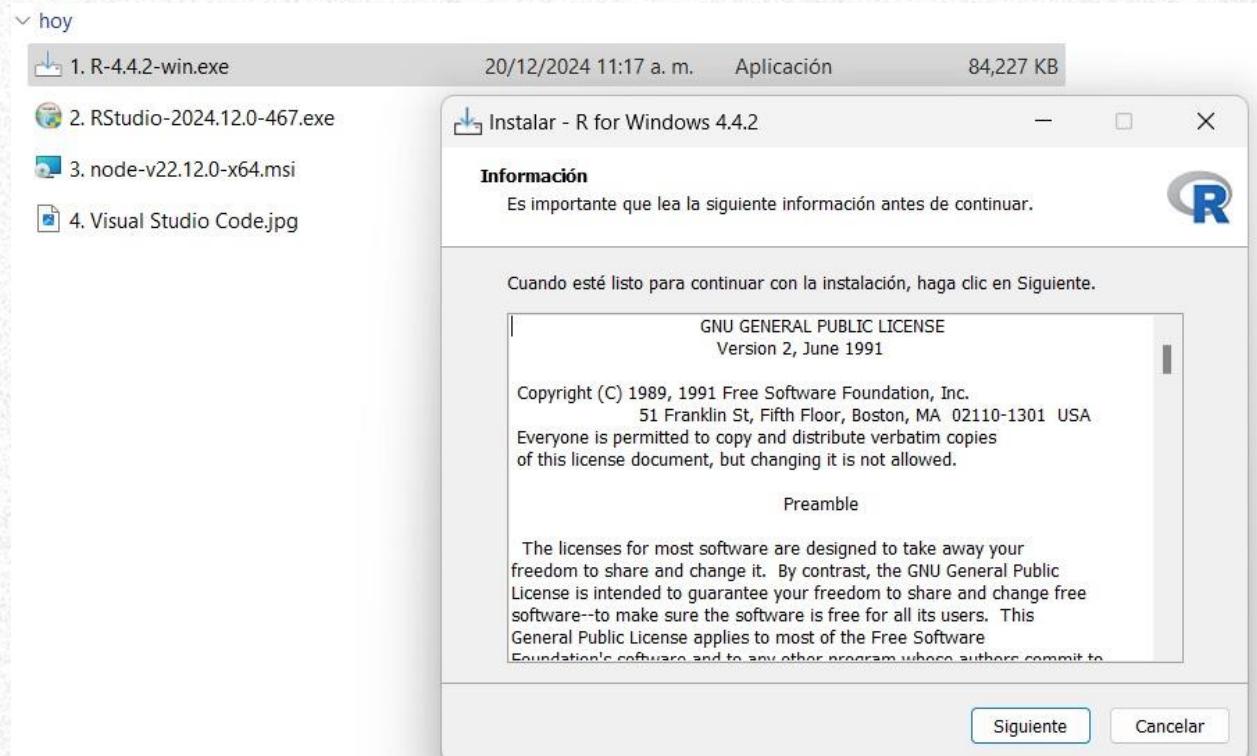
```
install.packages("ggplot2")
install.packages("plotly")
library("ggplot2")
library("plotly")
```

Configuración en Power BI

Configuración Básica:

Para utilizar R en Power BI, primero debemos realizar una configuración inicial del entorno.

1. Instalación del paquete de lenguaje R

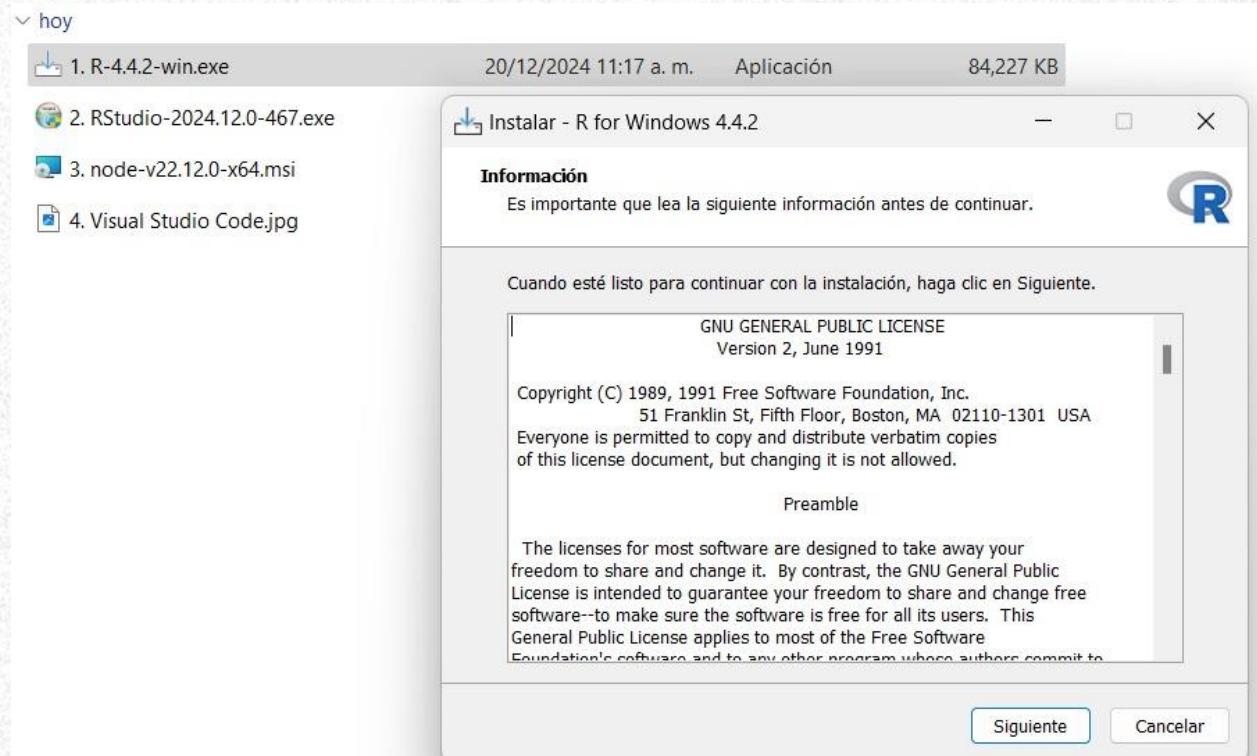


Configuración en Power BI

Configuración Básica:

Para utilizar R en Power BI, primero debemos realizar una configuración inicial del entorno.

1. Instalación del paquete de lenguaje R



Configuración en Power BI

Configuración Básica:

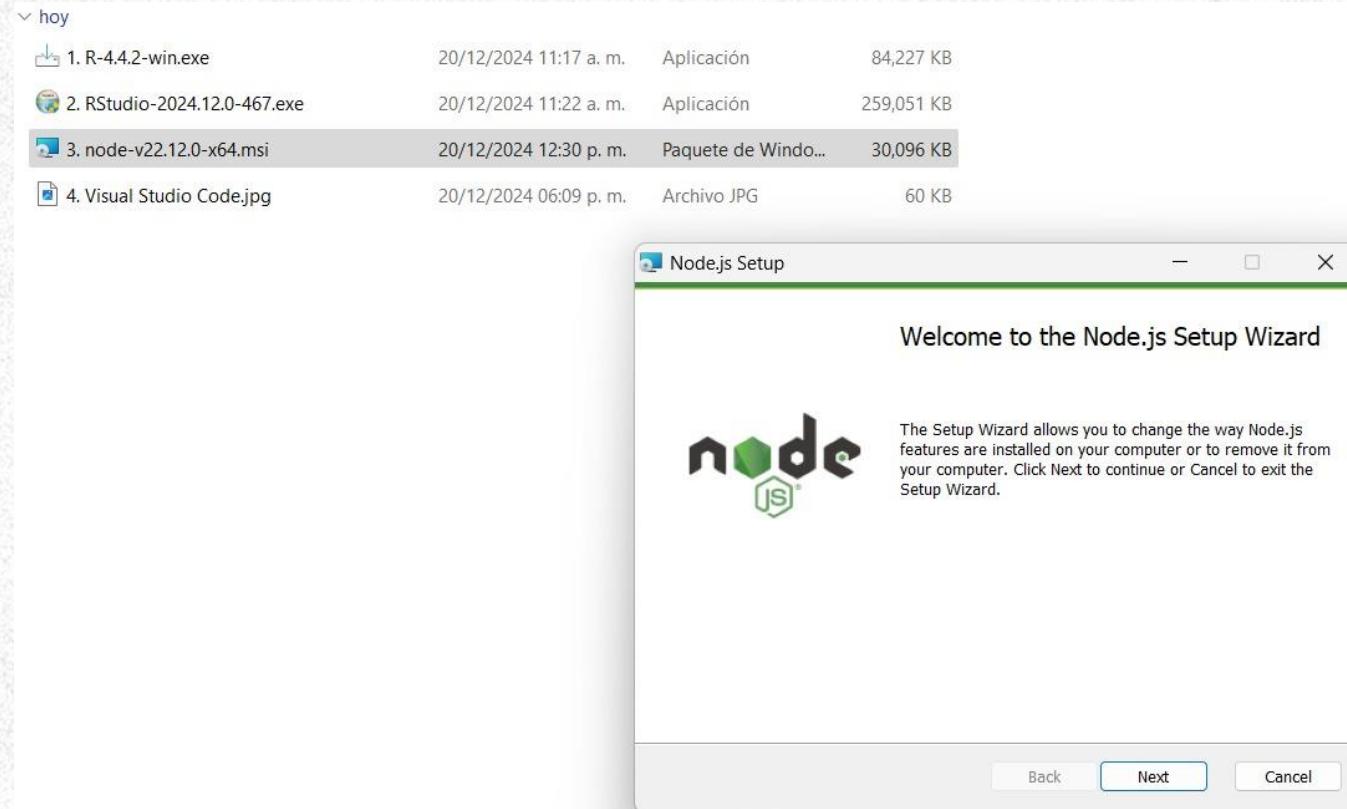
2. Instalación de la interfaz de lenguaje R

▼ hoy		
 1. R-4.4.2-win.exe	20/12/2024 11:17 a. m.	Aplicación
 2. RStudio-2024.12.0-467.exe	20/12/2024 11:22 a. m.	Aplicación
 3. node-v22.12.0-x64.msi	20/12/2024 12:30 p. m.	Paquete de Wind
 4. Visual Studio Code.jpg	20/12/2024 06:09 p. m.	Archivo JPG

Configuración en Power BI

Configuración Básica:

3. Instalamos Node, el cual es un framework de Javascript para usar graficas en Plotly en Power BI



Configuración en Power BI

Configuración Básica:

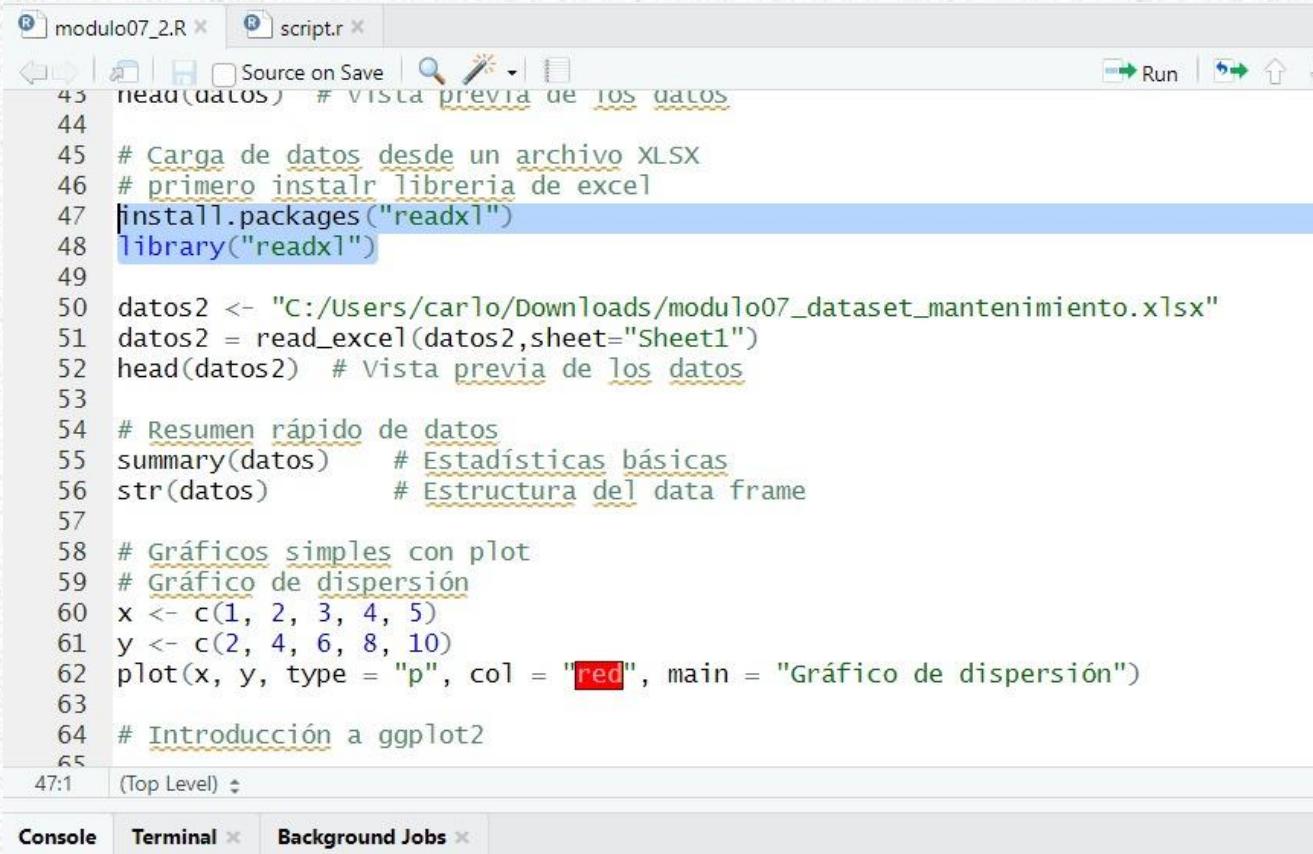
4. Instalamos Visual Studio Code, el cual es un editor de código.



Configuración en Power BI

Configuración Básica:

5. Abrimos Rstudio, instalamos las librerías desde el editor de código



The screenshot shows the RStudio interface with two tabs open: 'modulo07_2.R' and 'script.r'. The 'script.r' tab is active, displaying the following R code:

```
43 head(datos) # Vista previa de los datos
44
45 # Carga de datos desde un archivo XLSX
46 # primero instalar librería de excel
47 install.packages("readxl")
48 library("readxl")
49
50 datos2 <- "C:/Users/carlo/Downloads/modulo07_dataset_mantenimiento.xlsx"
51 datos2 = read_excel(datos2,sheet="Sheet1")
52 head(datos2) # Vista previa de los datos
53
54 # Resumen rápido de datos
55 summary(datos) # Estadísticas básicas
56 str(datos) # Estructura del data frame
57
58 # Gráficos simples con plot
59 # Gráfico de dispersión
60 x <- c(1, 2, 3, 4, 5)
61 y <- c(2, 4, 6, 8, 10)
62 plot(x, y, type = "p", col = "red", main = "Gráfico de dispersión")
63
64 # Introducción a ggplot2
65
```

The line 'install.packages("readxl")' is highlighted with a blue background, indicating it is currently being edited or is the active line.

Configuración en Power BI

Configuración Básica:

Podemos instalar manualmente las librerías, en Package

The screenshot shows the RStudio interface. On the left, the script editor displays R code for loading datasets and performing basic analysis. On the right, a package manager window is open, showing available packages like 'readxl' and 'foreign'.

```
39 datos %>% select(nombre)
40
41 # Carga de datos desde un archivo CSV
42 datos <- read.csv("C:/Users/Fernando/Downloads/dataset_mantenimiento_predictivo.csv")
43 head(datos) # Vista previa de los datos
44
45 # Carga de datos desde un archivo XLSX
46 # primero instalar librería de excel
47 install.packages("readxl")
48 library("readxl")
49
50 datos2 <- "C:/Users/Fernando/Downloads/modulo07_dataset_mantenimiento.xlsx"
51 datos2 = read_excel(datos2,sheet="Sheet1")
52 head(datos2) # Vista previa de los datos
53
54 # Resumen rápido de datos
55 summary(datos) # Estadísticas básicas
56 str(datos) # Estructura del data frame
57
58 # Gráficos simples con plot
59 # Gráfico de dispersión
60 x <- c(1, 2, 3, 4, 5)
```

R Script

Console Terminal Background Jobs

```
R - R4.2.2 - ~/ ~
< es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

> remove.packages("readxl")
Removing package from 'C:/Users/Fernando/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
> datos2 <- "C:/Users/Fernando/Downloads/modulo07_dataset_mantenimiento.xlsx"
> datos2 = read_excel(datos2,sheet="Sheet1")
Error en read_excel(datos2, sheet = "Sheet1"):
  no se pudo encontrar la función "read_excel"
```

Project: (None)

Environment History Connections Tutorial

Values

datos2	"C:/Users/Fernando/Downloads/modulo07_dat...
--------	--

Files Plots Packages Help Viewer Presentation

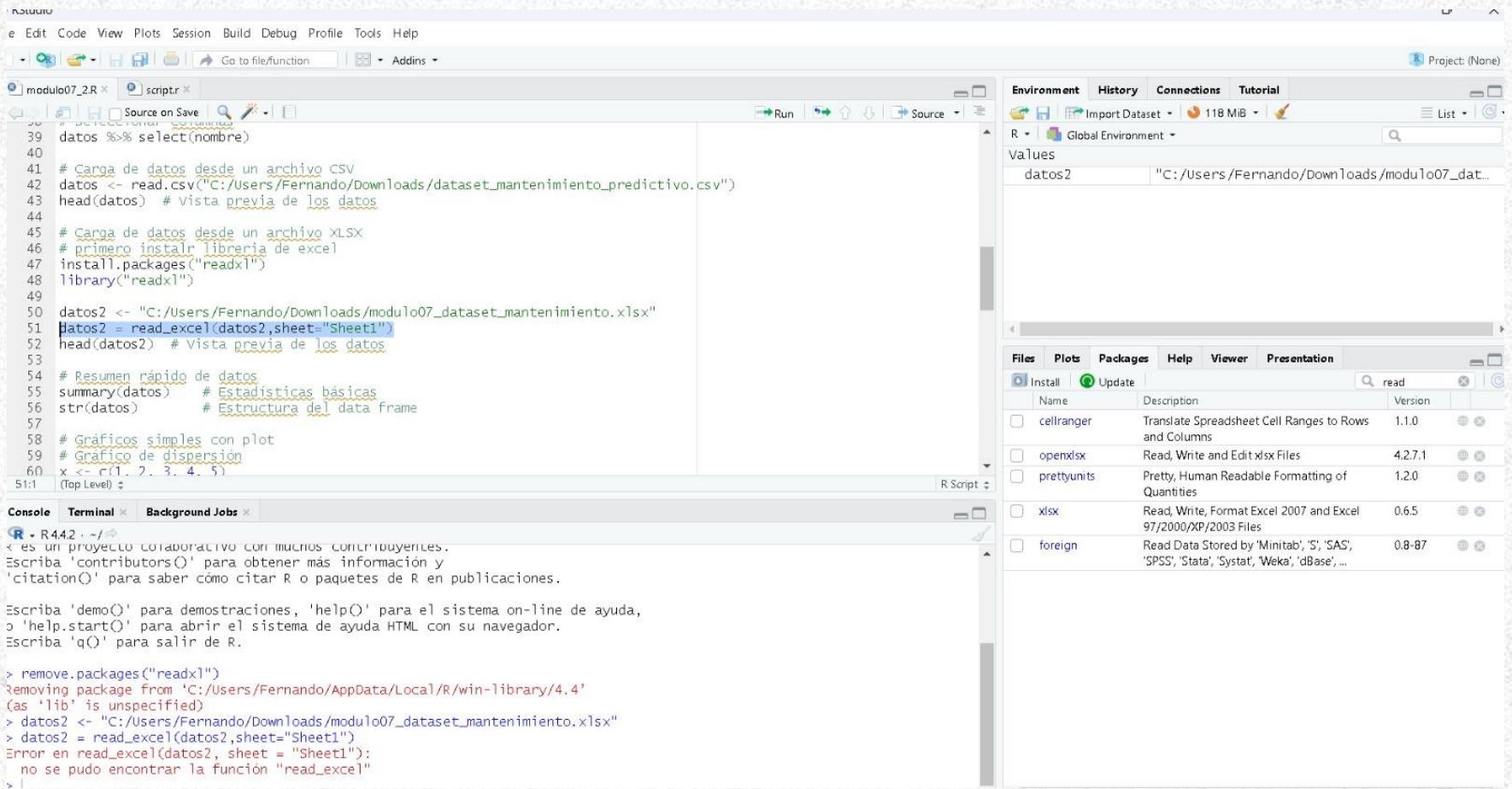
Install Update

Name	Description	Version
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
openxlsx	Read, Write and Edit xlsx Files	4.2.7.1
prettyunits	Pretty, Human Readable Formatting of Quantities	1.2.0
xlsx	Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files	0.6.5
foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Weka', 'dBase', ...	0.8-87

Configuración en Power BI

Configuración Básica:

Podemos instalar manualmente las librerías, en Package



The screenshot shows the RStudio interface. On the left, the script editor displays R code for reading data from CSV and XLSX files. On the right, the package manager window lists several packages available for installation or update.

```
modulo07_2.R
39 datos %>% select(nombre)
40
41 # Carga de datos desde un archivo CSV
42 datos <- read.csv("C:/Users/Fernando/Downloads/dataset_mantenimiento_predictivo.csv")
43 head(datos) # Vista previa de los datos
44
45 # Carga de datos desde un archivo XLSX
46 # primero instalar librería de excel
47 install.packages("readxl")
48 library("readxl")
49
50 datos2 <- "C:/Users/Fernando/Downloads/modulo07_dataset_mantenimiento.xlsx"
51 datos2 = read_excel(datos2,sheet="Sheet1")
52 head(datos2) # Vista previa de los datos
53
54 # Resumen rápido de datos
55 summary(datos) # Estadísticas básicas
56 str(datos) # Estructura del data frame
57
58 # Gráficos simples con plot
59 # Gráfico de dispersión
60 x <- c(1. 2. 3. 4. 5)
51:1 (Top Level) ▾
```

R Script

Console Terminal Background Jobs

```
R - R4.4.2 - ~/R
< es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

> remove.packages("readxl")
Removing package from 'C:/Users/Fernando/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
> datos2 <- "C:/Users/Fernando/Downloads/modulo07_dataset_mantenimiento.xlsx"
> datos2 = read_excel(datos2,sheet="Sheet1")
> Error en read_excel(datos2, sheet = "Sheet1"):
no se pudo encontrar la función "read_excel"
>
```

Project: (None)

Environment History Connections Tutorial

Import Dataset 118 MB

R Global Environment

Values

datos2	"C:/Users/Fernando/Downloads/modulo07_dat...
--------	--

Files Plots Packages Help Viewer Presentation

Install Update

Name	Description	Version
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
openxlsx	Read, Write and Edit xlsx Files	4.2.7.1
prettyunits	Pretty, Human Readable Formatting of Quantities	1.2.0
xlsx	Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files	0.6.5
foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-87

Configuración en Power BI

Configuración Básica:

Podemos instalar manualmente las librerías, en Package

The screenshot shows the RStudio interface with the following details:

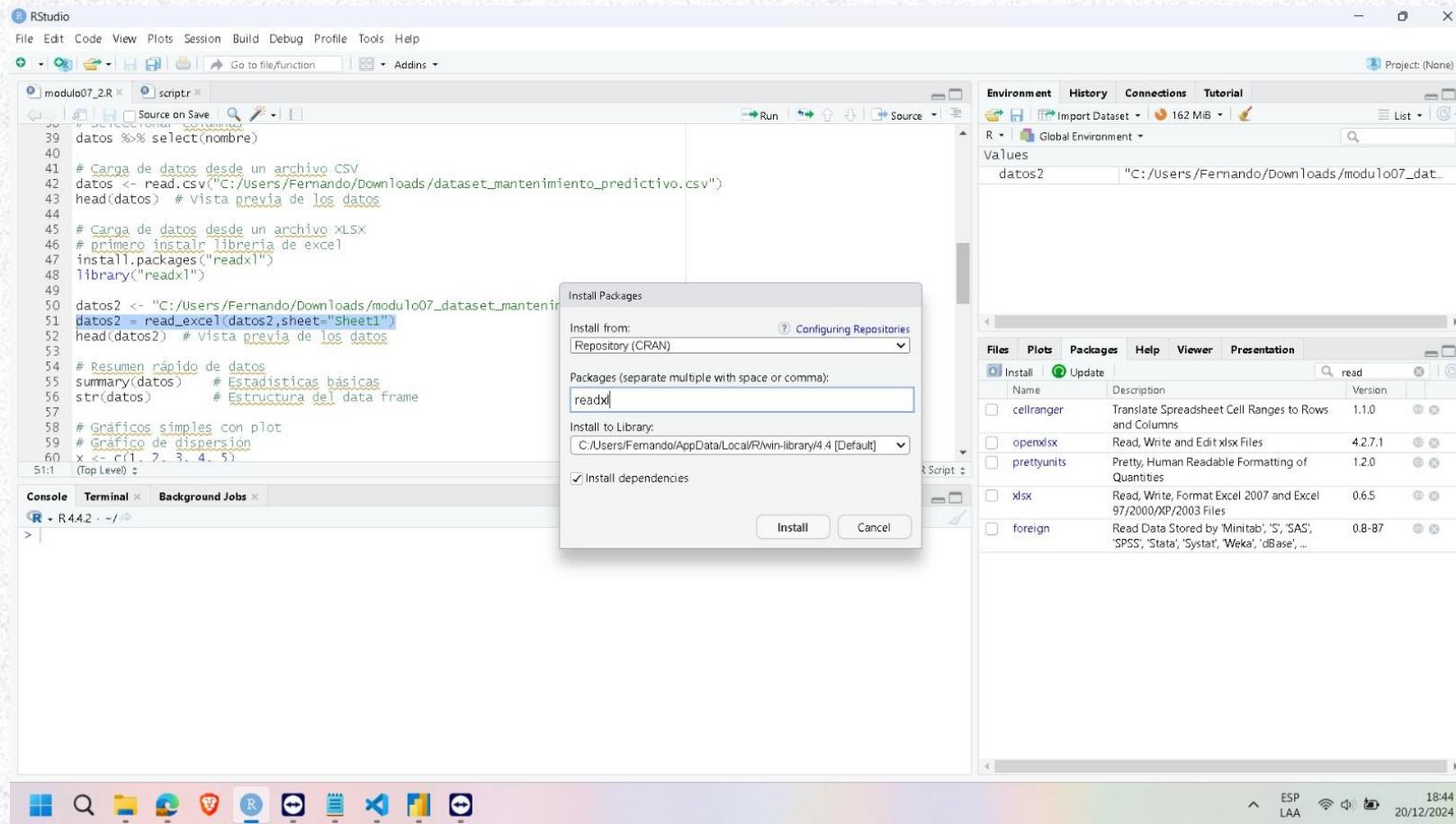
- Code Editor:** The script pane contains R code for reading data from CSV and XLSX files. It includes comments explaining the steps: "# Carga de datos desde un archivo CSV", "# Carga de datos desde un archivo XLSX", and "# Resumen rápido de datos".
- Console:** The console pane shows the R environment setup, including the R version (R 4.4.2) and a note about it being a collaborative project.
- Package Manager:** The right-hand panel shows the "Packages" tab of the RStudio IDE. It lists several packages installed in the global environment:

Name	Description	Version
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
openxlsx	Read, Write and Edit.xlsx Files	4.2.7.1
prettyunits	Pretty, Human Readable Formatting of Quantities	1.2.0
xlsx	Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files	0.6.5
foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-87

Configuración en Power BI

Configuración Básica:

Podemos instalar manualmente las librerías, en Package



Configuración en Power BI

Configuración Básica:

Podemos instalar manualmente las librerías, en Package

The screenshot shows the RStudio interface. On the left, the script editor displays R code for loading data from CSV and Excel files, and creating plots. On the right, a package manager window lists packages available for installation, including 'readxl' which is checked. The bottom status bar shows system information like battery level, signal strength, and date.

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Source on Save Run Source Environment History Connections Tutorial
Project: (None)
modulo07_2_R.R script.r
39 datos %>% select(nombe)
40
41 # Carga de datos desde un archivo CSV
42 datos <- read.csv("C:/users/Fernando/Downloads/dataset_mantenimiento_predictivo.csv")
43 head(datos) # Vista previa de los datos
44
45 # Carga de datos desde un archivo xlsx
46 # primero instalar librería de excel
47 install.packages("readxl")
48 library("readxl")
49
50 datos2 <- "C:/Users/Fernando/Downloads/modulo07_dataset_mantenimiento.xlsx"
51 datos2 = read_xlsx(datos2,sheet="Sheet1")
52 head(datos2) # Vista previa de los datos
53
54 # Resumen rápido de datos
55 summary(datos) # Estadísticas básicas
56 str(datos) # Estructura del data frame
57
58 # Gráficos simples con plot
59 # Gráfico de dispersión
60 x <- c(1, 2, 3, 4, 5)
51:1 (Top Level) ⇩
Console Terminal Background Jobs
R - R4.4.2 - /-
WARNING: RTOOLS IS REQUIRED TO BUILD R PACKAGES BUT IS NOT CURRENTLY INSTALLED. PLEASE DOWNLOAD AND INSTALL THE APPROPRIATE VERSION OF RTOOLS BEFORE PROCEEDING:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Fernando/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
probando la URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/readxl_1.4.3.zip'
Content type 'application/zip' length 1204230 bytes (1.1 MB)
downloaded 1.1 MB

package 'readxl' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/Fernando/AppData/Local/Temp/RtmpCyy5Na/downloaded_packages
> library(readxl)
> |
```

- Ggplot2
- Readxl
- Xml2
- plotly

Configuración en Power BI

Configuración Básica:

6. Configurar Power BI para reconocer R.

The screenshot shows the 'Opciones y configuración' (Options and Configuration) dialog box. The left sidebar has three items: 'Iniciar sesión' (Sign In), 'Opciones y configuración' (Options and Configuration, highlighted in teal), and 'Acerca de' (About). The main area has two sections: 'Opciones' (Options) with a gear icon and 'Configuración de origen de datos' (Data Source Configuration) with a document icon. On the right, under the 'GLOBAL' heading, there is a list of options: 'Carga de datos' (Load Data), 'Editor de Power Query', 'DirectQuery', 'Script de R' (selected, highlighted in teal), 'Creación de scripts de Python', 'Seguridad', 'Privacidad', 'Configuración regional', 'Actualizaciones', and 'Datos de uso'. At the bottom right, there is a section for 'Opciones de script de R' (R script options) with a dropdown menu set to 'C:\Program Files\R\R-4.4.2' and a link to 'Cómo instalar R' (How to install R).

