

Librerías de R para Interactuar con Excel



Por Antonio Richaud

Introducción

En el análisis de datos, la capacidad de integrar diferentes herramientas puede potenciar significativamente los resultados y la eficiencia del trabajo. Una de las combinaciones más poderosas es el uso de R para el análisis de datos complejos junto con Excel para la presentación y manejo de estos datos. R, con su vasto conjunto de librerías, permite realizar análisis avanzados como regresiones, clusterización, análisis de componentes principales (PCA) y análisis geoespacial. Estas capacidades analíticas se pueden complementar al exportar los resultados a Excel, utilizando librerías específicas de R como `readxl` y `writexl`.

La combinación de R y Excel aprovecha la potencia analítica de R y la familiaridad de Excel. Mientras que R es ideal para manipular y analizar grandes conjuntos de datos, Excel es ampliamente utilizado por su interfaz amigable y sus capacidades para la presentación de datos. Esta sinergia facilita la creación de informes y la toma de decisiones basada en datos, permitiendo a los analistas compartir sus resultados de manera más efectiva con aquellos que quizás no estén tan familiarizados con R.

Además, el uso de R para generar gráficos y su inclusión directa en archivos de Excel mejora la visualización de datos. Los gráficos complejos creados en R pueden ser exportados e integrados en hojas de cálculo de Excel, ofreciendo una solución completa para el análisis y la presentación de datos. Esta guía explorará las principales librerías de R para interactuar con Excel, proporcionando ejemplos prácticos de cómo utilizarlas para mejorar tu flujo de trabajo analítico.

Librería readxl

La librería readxl es una herramienta poderosa y eficiente para leer datos de archivos Excel (.xls y .xlsx) directamente en R. Su principal ventaja es que no requiere la instalación de software adicional, lo que facilita su uso y configuración. Además, readxl está diseñada para ser rápida y eficiente, lo que la convierte en una excelente opción para manejar grandes volúmenes de datos.

Esta librería ofrece una variedad de funciones útiles que permiten importar hojas de cálculo específicas, definir rangos de celdas y manejar diversos tipos de datos. Esto es particularmente útil cuando se trabaja con archivos Excel que contienen múltiples hojas o cuando se necesita extraer solo una parte específica de los datos.

El uso de readxl es ideal para integrar datos de Excel en análisis y proyectos en R, permitiendo una fácil manipulación y análisis de datos directamente desde archivos Excel. Vamos a ver un ejemplo de cómo utilizar readxl para leer un archivo de Excel en R:

```
# Instalación de la librería readxl
install.packages("readxl")

# Carga de la librería
library(readxl)

# Lectura de un archivo Excel
# Asegúrate de proporcionar la ruta correcta al archivo
data <- read_excel("ruta/al/archivo.xlsx")

# Visualización de los datos
print(data)
```

En este ejemplo, primero instalamos la librería utilizando `install.packages("readxl")` si no la tenemos ya instalada. Luego, cargamos la librería en nuestro entorno de R con `library(readxl)`. La función principal para leer archivos Excel es `read_excel()`, donde especificamos la ruta del archivo que queremos leer. Una vez leído el archivo, podemos visualizar los datos utilizando la función `print()`.

Además de la función básica `read_excel()`, `readxl` permite especificar hojas específicas y rangos de celdas, lo que proporciona gran flexibilidad para trabajar con diferentes formatos y estructuras de datos en archivos Excel.

```
# Lectura de una hoja específica
data_sheet <- read_excel("ruta/al/archivo.xlsx", sheet =
"NombreDeLaHoja" )

# Lectura de un rango específico de celdas
data_range <- read_excel("ruta/al/archivo.xlsx", range = "A1:C10" )
```

En estos ejemplos adicionales, mostramos cómo leer una hoja específica de un archivo Excel utilizando el parámetro `sheet`, y cómo leer un rango específico de celdas utilizando el parámetro `range`.

Librería writexl

La librería writexl es una herramienta muy buena para exportar datos de R a archivos Excel (.xlsx) de manera eficiente y sin la necesidad de depender de software externo. Esta librería se destaca por su simplicidad y rapidez, permitiendo que los usuarios escriban data frames en hojas de cálculo de Excel con facilidad.

Entre las funcionalidades que ofrece writexl, se incluyen la capacidad de escribir data frames en hojas de cálculo, personalizar nombres de hojas y aplicar formatos básicos. Esto es especialmente útil cuando necesitas compartir y presentar los resultados de tus análisis en un formato ampliamente accesible y fácil de entender para otros usuarios.

La utilización de writexl en tus proyectos de R facilita la generación de reportes y la colaboración con personas que prefieren trabajar en Excel. A continuación, se presenta un ejemplo práctico de cómo utilizar writexl para exportar un data frame a un archivo de Excel:

```
# Instalación de la librería writexl
install.packages("writexl")

# Carga de la librería
library(writexl)

# Creación de un data frame de ejemplo
data <- data.frame(
  Nombre = c("Ana", "Luis", "Carlos"),
  Edad = c(28, 34, 29),
  Ciudad = c("Puebla", "CDMX", "Guadalajara")
)

# Escritura del data frame a un archivo Excel
write_xlsx(data, "ruta/al/archivo.xlsx")
```

En este ejemplo, comenzamos instalando la librería writexl utilizando `install.packages("writexl")`, si aún no está instalada. Luego, cargamos la librería en nuestro entorno de R con `library(writexl)`. Creamos un data frame de ejemplo llamado `data` que contiene algunos nombres, edades y ciudades. Finalmente, utilizamos la función `write_xlsx()` para escribir el data frame a un archivo de Excel, especificando la ruta donde se guardará el archivo.

Además de las funciones básicas, writexl permite la personalización de las hojas de cálculo. A continuación, se muestra cómo escribir múltiples data frames en diferentes hojas de un mismo archivo de Excel:

```
# Creación de otros data frames de ejemplo
data2 <- data.frame(
  Producto = c("A", "B", "C"),
  Precio = c(10, 15, 8)
)

# Escritura de múltiples data frames en un archivo Excel
write_xlsx(list(Hoja1 = data, Hoja2 = data2), "ruta/al/archivo.xlsx")
```

En este ejemplo, creamos un segundo data frame llamado `data2` y lo escribimos junto con el primer data frame `data` en un solo archivo de Excel, cada uno en su propia hoja. Utilizamos una lista para especificar los nombres de las hojas y los data frames correspondientes.

Librería openxlsx

La librería openxlsx permite crear, leer y modificar archivos Excel (.xlsx) sin necesidad de software externo, ofreciendo una amplia gama de funciones para personalizar hojas de cálculo. Esta librería es ideal para aquellos usuarios que necesitan un control detallado sobre la manipulación y presentación de datos en Excel desde R.

Entre las funcionalidades destacadas de openxlsx se incluyen la creación de múltiples hojas, la aplicación de formatos avanzados, la inclusión de gráficos y la creación de tablas dinámicas. Esto hace que openxlsx sea una herramienta muy poderosa para la generación de reportes y análisis avanzados directamente en Excel. A continuación, te muestro cómo utilizar openxlsx para crear un archivo Excel con múltiples hojas y aplicar algunos formatos básicos:

```
# Instalación de la librería openxlsx
install.packages("openxlsx")

# Carga de la librería
library(openxlsx)

# Creación de un nuevo workbook y una hoja
wb <- createWorkbook()

# Añadir una hoja al workbook
addWorksheet(wb, "Hoja1")

# Escribir datos en la hoja
writeData(wb, sheet = "Hoja1", x = iris)

# Aplicar un formato básico
setColWidths(wb, sheet = "Hoja1", cols = 1:5, widths = "auto")

# Añadir otra hoja con datos diferentes
addWorksheet(wb, "Hoja2")
writeData(wb, sheet = "Hoja2", x = mtcars)

# Guardar el workbook
saveWorkbook(wb, "ruta/al/archivo.xlsx", overwrite = TRUE)
```

En este ejemplo, instalamos la librería `openxlsx` utilizando `install.packages("openxlsx")` si no está instalada. Luego, cargamos la librería en R con `library(openxlsx)`. Creamos un nuevo workbook y añadimos una hoja llamada “Hoja1”. Escribimos el conjunto de datos iris en esta hoja y ajustamos automáticamente el ancho de las columnas. A continuación, añadimos otra hoja llamada “Hoja2” y escribimos el conjunto de datos `mtcars`. Finalmente, guardamos el workbook en la ruta especificada utilizando `saveWorkbook()`.

Además de las funcionalidades básicas, `openxlsx` permite la creación de gráficos y tablas dinámicas. A continuación, un ejemplo de cómo añadir un gráfico:

```
# Añadir una hoja para el gráfico
addWorksheet(wb, "Gráfico")

# Escribir datos
writeData(wb, "Gráfico", iris)

# Crear un gráfico
chart <- createPlot(wb, sheet = "Gráfico", width = 6, height = 3,
startRow = 1, startCol = 6)
chart$addData(x = iris$Sepal.Length, y = iris$Sepal.Width, type =
"scatter", style = "point")

# Guardar el workbook con el gráfico
saveWorkbook(wb, "ruta/al/archivo_con_grafico.xlsx", overwrite = TRUE)
```

En este ejemplo, añadimos una nueva hoja llamada “Gráfico” y escribimos el conjunto de datos iris. Luego, creamos un gráfico de dispersión utilizando la función `createPlot()` y especificamos los datos para los ejes X e Y. Finalmente, guardamos el workbook con el gráfico incluido.

Librería excelR

La librería excelR permite integrar hojas de cálculo interactivas de Excel en aplicaciones Shiny de R. Esta librería es ideal para crear aplicaciones interactivas que combinen el poder de análisis de R con la facilidad de uso de Excel.

Con excelR, puedes editar, formatear y visualizar datos en tiempo real dentro de una interfaz web. Esto es especialmente útil para aplicaciones donde los usuarios necesitan manipular datos de manera interactiva y ver los resultados de sus cambios al instante. Ahí te va cómo utilizar excelR para integrar una hoja de cálculo interactiva en una aplicación Shiny:

```
# Instalación de las librerías necesarias
install.packages("shiny")
install.packages("excelR")

# Carga de las librerías
library(shiny)
library(excelR)

# Definición de la interfaz de usuario
ui <- fluidPage(
  titlePanel("Ejemplo de excelR en Shiny"),
  excelOutput("hoja_excel")
)

# Definición del servidor
server <- function(input, output, session) {
  # Datos iniciales
  datos <- data.frame(
    Nombre = c("Ana", "Luis", "Carlos"),
    Edad = c(28, 34, 29),
    Ciudad = c("Madrid", "Barcelona", "Valencia")
  )

  # Renderización de la hoja de cálculo
  output$hoja_excel <- renderExcel({
    excelTable(data = datos)
  })
}

# Ejecución de la aplicación Shiny
shinyApp(ui, server)
```

En este ejemplo, primero instalamos las librerías shiny y excelR utilizando `install.packages()`. Luego, cargamos ambas librerías en nuestro entorno de R. Definimos la interfaz de usuario (ui) utilizando `fluidPage()` y creamos un panel de título junto con una salida de Excel (`excelOutput`). En la definición del servidor (`server`), creamos un data frame con algunos datos de ejemplo y lo renderizamos en la salida de Excel utilizando `renderExcel()` y `excelTable()`. Finalmente, ejecutamos la aplicación Shiny con `shinyApp(ui, server)`.

excelR proporciona una manera intuitiva de trabajar con datos en una aplicación Shiny, permitiendo que los usuarios interactúen directamente con los datos en una hoja de cálculo similar a Excel.

Librería tidyverse

Es una colección de paquetes en R diseñados para la manipulación y análisis de datos de manera eficiente y coherente. Esta colección incluye algunos de los paquetes más populares y útiles en R, como dplyr, tidyr, ggplot2 y readr. Cada uno de estos paquetes tiene funciones específicas que facilitan el trabajo con datos, desde su limpieza y transformación hasta su visualización y análisis.

Paquetes como dplyr y tidyr son esenciales para la transformación y limpieza de datos. ggplot2 es ampliamente utilizado para la visualización de datos, permitiendo crear gráficos sofisticados y personalizables. readr facilita la importación de datos desde diferentes fuentes, como archivos CSV, lo que permite una integración fluida con otras librerías de R.

Cuando se utilizan en conjunto con librerías de Excel como readxl y writexl, los paquetes del tidyverse permiten transformar, limpiar y visualizar datos antes de exportarlos a Excel. Esto mejora significativamente el flujo de trabajo y la calidad del análisis, proporcionando herramientas robustas para el manejo de datos.

Vamos a ver en la página que sigue cómo utilizar algunos de los paquetes del tidyverse junto con readxl y writexl para trabajar con datos y exportarlos a Excel:

```
# Instalación de los paquetes necesarios
install.packages("tidyverse")
install.packages("readxl")
install.packages("writexl")

# Carga de las librerías
library(tidyverse)
library(readxl)
library(writexl)

# Lectura de un archivo Excel con readxl
data <- read_excel("ruta/al/archivo.xlsx")

# Transformación de datos con dplyr y tidyr
data_clean <- data %>%
  filter(!is.na(VariableInteres)) %>% # Filtrar filas con NA en una
  # variable de interés
  mutate(NuevaVariable = Variable1 * 2) # Crear una nueva variable

# Visualización de datos con ggplot2
ggplot(data_clean, aes(x = Variable1, y = Variable2)) +
  geom_point() +
  theme_minimal()

# Exportación de los datos limpios a un nuevo archivo Excel con writexl
write_xlsx(data_clean, "ruta/al/nuevo_archivo.xlsx")
```

En este ejemplo, instalamos los paquetes necesarios (tidyverse, readxl y writexl) y los cargamos en R. Luego, leemos un archivo Excel utilizando readxl. Usamos dplyr y tidyr para filtrar y transformar los datos, creando un nuevo data frame data_clean. Utilizamos ggplot2 para visualizar los datos limpios con un gráfico de dispersión. Finalmente, exportamos los datos limpios a un nuevo archivo Excel utilizando writexl. :)

Documentación



<https://swechhya.github.io/excelR/>



<https://www.tidyverse.org/packages/>