

Guía de DataFrames en R

Larios Ponce Hector Manuel

22-07-2024

Guía de DataFrames en R

En R, los DataFrames son estructuras de datos fundamentales que permiten organizar y manipular datos de manera tabular. Este documento proporciona una introducción básica sobre cómo trabajar con DataFrames en R.

Creación de DataFrames

- **Crear un DataFrame desde cero:**

Para crear un DataFrame nuevo desde cero, puedes usar la función `data.frame()` proporcionando vectores para cada columna:

```
#Ejemplo: Crear un DataFrame de estudiantes # nolint: indentation_linter.
estudiantes <- data.frame(
  nombre = c("Juan", "María", "Pedro"),
  edad = c(22, 24, 21),
  promedio = c(8.5, 7.9, 9.2)
)

# Mostrar el DataFrame creado # nolint: indentation_linter.
estudiantes
```

```
##   nombre edad promedio
## 1   Juan   22      8.5
## 2  María   24      7.9
## 3  Pedro   21      9.2
```

Operaciones básicas con DataFrames

Visualización y exploración de DataFrames

- **Visualizar las primeras filas del DataFrame:**

Utiliza `head()` para ver las primeras filas o `tail()` para ver las últimas.

```
head(estudiantes) # Muestra las primeras 6 filas por defecto
```

```
##  nombre edad promedio
## 1   Juan   22         8.5
## 2  María   24         7.9
## 3  Pedro   21         9.2
```

```
tail(estudiantes, 1) # Muestra las últimas 6 filas por defecto
```

```
##  nombre edad promedio
## 3  Pedro   21         9.2
```

- **Obtener dimensiones del DataFrame:**

Utiliza `dim()` para obtener el número de filas y columnas.

```
dim(estudiantes) # Muestra el número de filas y columnas
```

```
## [1] 3 3
```

- **Nombres de las columnas:**

Utiliza `names()` para obtener los nombres de las columnas.

```
names(estudiantes) # Muestra los nombres de las columnas
```

```
## [1] "nombre" "edad" "promedio"
```

Manipulación de DataFrame

- **Acceso a elementos:**

Puedes acceder a elementos específicos del DataFrame utilizando corchetes `[filas, columnas]` o el nombre de la columna.

```
# Acceder a la edad de María
estudiantes[2, "edad"] # Accede a la fila 2, columna "edad"
```

```
## [1] 24
```

- **Filtrar filas:**

Puedes filtrar filas basadas en condiciones lógicas específicas.

```
# Filtrar estudiantes con promedio mayor a 8.0
estudiantes[estudiantes$promedio > 8.0, ]
```

```
##  nombre edad promedio
## 1   Juan   22         8.5
## 3  Pedro   21         9.2
```

- **Agregar nuevas columnas:**

Puedes agregar nuevas columnas al DataFrame, e incluso asignar valores calculados.

```
# Agregar columna estado (aprobado/reprobado/aprobado)
estudiantes$status <- ifelse(estudiantes$promedio >= 8.0, "ap", "rp")
estudiantes
```

```
##   nombre edad promedio status
## 1   Juan   22      8.5     ap
## 2  María   24      7.9     rp
## 3  Pedro   21      9.2     ap
```

Analisis exploratorio de DataFrame

Resumen estadístico

Utiliza `summary()` para obtener un resumen estadístico de las columnas numéricas.

```
summary(estudiantes)
```

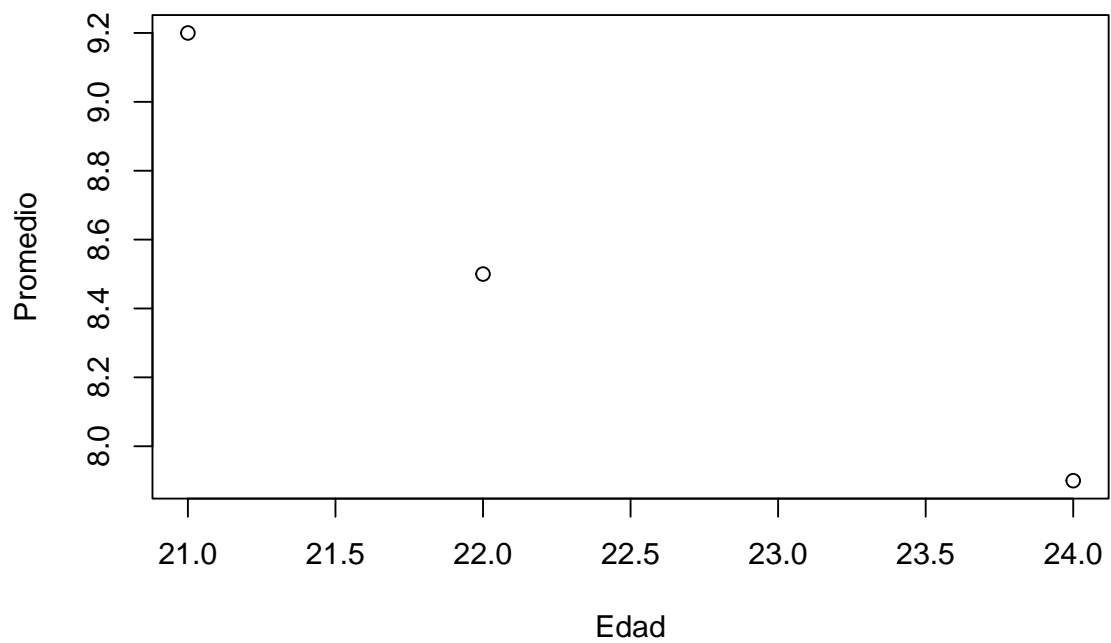
```
##      nombre          edad      promedio      status
## Length:3      Min.    :21.00   Min.    :7.900   Length:3
## Class :character 1st Qu.:21.50   1st Qu.:8.200   Class :character
## Mode  :character Median :22.00   Median :8.500   Mode  :character
##                      Mean    :22.33   Mean    :8.533
##                      3rd Qu.:23.00   3rd Qu.:8.850
##                      Max.    :24.00   Max.    :9.200
```

Gráficos

Puedes crear gráficos para visualizar los datos en el DataFrame.

- **Gráfico de dispersión:** Usa `plot()` para crear gráficos simples.

```
plot(estudiantes$edad, estudiantes$promedio, xlab = "Edad", ylab = "Promedio")
```



- **Gráfico de barras:** Usa `barplot()` para crear gráficos de barras.

```
barplot(  
  estudiantes$promedio,  
  names.arg = estudiantes$nombre,  
  xlab = "Estudiantes",  
  ylab = "Promedio")
```

