

Introducción a Base R para Manipulación de Datos I

2023-05-13

Importación de datos

En esta sección utilizaremos las letras del Banco Central de la República Dominicana, el archivo tiene los Montos Colocados y Tasa de Rendimiento Promedio Ponderada 2007-2023

Letras del Banco Central: Instrumentos financieros emitidos a plazos de hasta un año, cuya rentabilidad viene dada por la diferencia entre el precio de adquisición y su valor al vencimiento o valor par.

Fuente: https://cdn.bancentral.gov.do/documents/instrumentos-de-inversion/instrumentos-operaciones-de-mercado-abierto/documents/letras_bc_consolidado.xlsx?v=1684020088887{#Link a Letras}

La función `read.table()` en R se utiliza para importar datos tabulares desde un archivo de texto o csv en un dataframe.

```
##Leemos la tabla de letras
letras_bc <- read.table("data/letras_bc_consolidado_clean.csv")
```

Slicing

Slicing es una técnica que se utiliza para extraer subconjuntos de datos en función de las posiciones de las filas y columnas. En R, se puede utilizar la notación `[fila, columna]` para extraer una fila y columna específicas de un dataframe.

La notación es:

```
[fila_inicio:fila_fin, columna_inicio:columna_fin]
```

```
letras_bc[1:3,]
```

```
##   FechadeSubasta FechaLiquidacion MontoSubastado MontoDemandado MontoAdjudicado
## 1    2007-04-03    2007-04-04           400          2281.8           400
## 2    2010-02-17    2010-02-02           300          1088.0           300
## 3    2010-02-24    2010-02-02           500           575.7           500
##   RendimientoPPA
## 1      0.095700
## 2      0.052256
## 3      0.052439
```

```
letras_bc[1:2, c("FechadeSubasta", "FechaLiquidacion", "MontoSubastado", "RendimientoPPA")]
```

```
##   FechadeSubasta FechaLiquidacion MontoSubastado RendimientoPPA
## 1    2007-04-03    2007-04-04           400      0.095700
## 2    2010-02-17    2010-02-02           300      0.052256
```

Filtering

En Base R, la técnica de filtrado de datos se puede realizar utilizando la función `subset()` o mediante la indexación de los datos con expresiones lógicas.

```
##Indexacion de expresiones logicas
letras_monto_subastado_5000 <- letras_bc[letras_bc$MontoSubastado==5000, ]
head(letras_monto_subastado_5000)
```

```
##      FechadeSubasta FechaLiquidacion MontoSubastado MontoDemandado
## 4      2023-04-12      2023-04-04           5000          9955.15
## 78     2008-08-06      2008-08-08           5000          1131.70
## 410    2020-10-07      2020-10-09           5000          4388.80
## 411    2020-10-14      2020-10-16           5000          4252.70
## 412    2020-10-21      2020-10-23           5000         11050.00
## 413    2020-10-28      2020-10-30           5000         13380.00
##      MontoAdjudicado RendimientoPPA
## 4           7951.91      0.12305392
## 78           931.70      0.14527100
## 410          3763.00      0.05498472
## 411          4027.70      0.05511510
## 412          9620.00      0.05483148
## 413         11665.00      0.05661526
```

```
##Uso de subset
letras_monto_subastado_mayor_5000 <- subset(letras_bc, MontoSubastado>5000)
head(letras_monto_subastado_mayor_5000)
```

```
##      FechadeSubasta FechaLiquidacion MontoSubastado MontoDemandado
## 65      2008-05-07      2008-05-09           6000          4763.00
## 442     2021-05-19      2021-05-21          10000         41104.62
## 488     2022-05-18      2022-05-20          10000         12964.58
## 489     2022-05-25      2022-05-27          10000         20566.00
## 490     2022-06-01      2022-06-03          10000          4418.86
##      MontoAdjudicado RendimientoPPA
## 65           4473.00      0.14061000
## 442          22867.50      0.05446814
## 488           1102.13      0.07340886
## 489          16613.60      0.08027351
## 490           3828.86      0.08348543
```

Crear Columnas

En R, se pueden crear nuevas columnas en un dataframe utilizando la notación de corchetes o la función `transform()`. Ambas opciones requieren que se especifique el nombre de la columna que se desea crear, así como una expresión que defina los valores de la columna.

El símbolo `$` en R se utiliza para acceder a una columna específica de un dataframe.

esta es la sintaxis: `dataframe$nombre_columna`

```
letras_monto_subastado_5000$RendimientoPPA2 <- (letras_monto_subastado_5000$RendimientoPPA *100)
head(letras_monto_subastado_5000)
```

```
##      FechadeSubasta FechaLiquidacion MontoSubastado MontoDemandado
## 4      2023-04-12      2023-04-04          5000          9955.15
## 78     2008-08-06     2008-08-08          5000          1131.70
## 410    2020-10-07     2020-10-09          5000          4388.80
## 411    2020-10-14     2020-10-16          5000          4252.70
## 412    2020-10-21     2020-10-23          5000         11050.00
## 413    2020-10-28     2020-10-30          5000         13380.00
##      MontoAdjudicado RendimientoPPA RendimientoPPA2
## 4              7951.91      0.12305392      12.305392
## 78              931.70      0.14527100      14.527100
## 410             3763.00      0.05498472       5.498472
## 411             4027.70      0.05511510       5.511510
## 412             9620.00      0.05483148       5.483148
## 413            11665.00      0.05661526       5.661526
```

```
letras_monto_subastado_5000 <- transform(letras_monto_subastado_5000, MontoSubastadoUS = MontoSubastado)
head(letras_monto_subastado_5000)
```

```
##      FechadeSubasta FechaLiquidacion MontoSubastado MontoDemandado
## 4      2023-04-12      2023-04-04          5000          9955.15
## 78     2008-08-06     2008-08-08          5000          1131.70
## 410    2020-10-07     2020-10-09          5000          4388.80
## 411    2020-10-14     2020-10-16          5000          4252.70
## 412    2020-10-21     2020-10-23          5000         11050.00
## 413    2020-10-28     2020-10-30          5000         13380.00
##      MontoAdjudicado RendimientoPPA RendimientoPPA2 MontoSubastadoUS
## 4              7951.91      0.12305392      12.305392          280000
## 78              931.70      0.14527100      14.527100          280000
## 410             3763.00      0.05498472       5.498472          280000
## 411             4027.70      0.05511510       5.511510          280000
## 412             9620.00      0.05483148       5.483148          280000
## 413            11665.00      0.05661526       5.661526          280000
```

Modificar columnas

Para modificar una columna en un dataframe podemos sobre-escribir la misma columna de la misma manera que creamos columnas.

Cambiar nombres de columnas

La función `colnames()` permite cambiar los nombres de las columnas de un dataframe. Para utilizarla, se debe asignar un vector de nombres de igual longitud que el número de columnas del dataframe a la función `colnames()`.

```
colnames(letras_bc)[1:3] <- c("FechaLiquidacion", "FechaSubasta", "Monto")
head(letras_bc)
```

```
##      FechaLiquidacion FechaSubasta Monto MontoDemandado MontoAdjudicado
## 1      2007-04-03      2007-04-04   400      2281.80      400.00
## 2      2010-02-17      2010-02-02   300      1088.00      300.00
## 3      2010-02-24      2010-02-02   500       575.70      500.00
## 4      2023-04-12      2023-04-04  5000      9955.15     7951.91
## 5      2019-01-01      2019-01-25  2500      3635.00     1515.00
## 6      2019-02-02      2019-02-15  2500      3823.00     2025.00
##      RendimientoPPA
## 1      0.09570000
## 2      0.05225600
## 3      0.05243900
## 4      0.12305392
## 5      0.07269412
## 6      0.07193004
```

La función `names()` permite cambiar los nombres de las columnas de un dataframe, de la misma manera que la función `colnames()`. La única diferencia es que `names()` es más genérica y también se puede utilizar para cambiar los nombres de las filas de una matriz.

```
names(letras_bc)[1:3] <- c("FechaSubasta", "FechaLiquidacion", "MontoSubastado")
head(letras_bc)
```

```
##      FechaSubasta FechaLiquidacion MontoSubastado MontoDemandado MontoAdjudicado
## 1      2007-04-03      2007-04-04         400      2281.80      400.00
## 2      2010-02-17      2010-02-02         300      1088.00      300.00
## 3      2010-02-24      2010-02-02         500       575.70      500.00
## 4      2023-04-12      2023-04-04        5000      9955.15     7951.91
## 5      2019-01-01      2019-01-25        2500      3635.00     1515.00
## 6      2019-02-02      2019-02-15        2500      3823.00     2025.00
##      RendimientoPPA
## 1      0.09570000
## 2      0.05225600
## 3      0.05243900
## 4      0.12305392
## 5      0.07269412
## 6      0.07193004
```