Tabla de Productos y Precios

A Preprint

Santiago Diez **Juan Rivero** **Agustín Rodriguez** **Tomás Boismene** **Lucas Patiri**

Tomás Cerutti

April 15, 2024

Abstract

Este documento presenta una tabla de productos y precios.

Tutorial básico de R

1 Asignación de variables

1.1 Puedes asignar valores a variables utilizando el operador <- o =

x <- 5 y = 10

2 Operaciones aritméticas

suma <- 3 + 4 resta <- 7 - 2 multiplicacion <- 5 * 2 division <- 10 / 2 potenciacion <- 2^3

3 Gráficos

3.1 R es conocido por su capacidad para generar gráficos de alta calidad.

3.2 Puedes crear gráficos utilizando funciones como plot(), hist(), barplot(), entre otras.

4 Funciones

4.1 R tiene muchas funciones integradas que puedes usar para realizar tareas específicas.

4.2 Por ejemplo, la función mean() calcula la media de un conjunto de números.

numeros <- c(1, 2, 3, 4, 5) media <- mean(numeros) # Imprimir la tabla

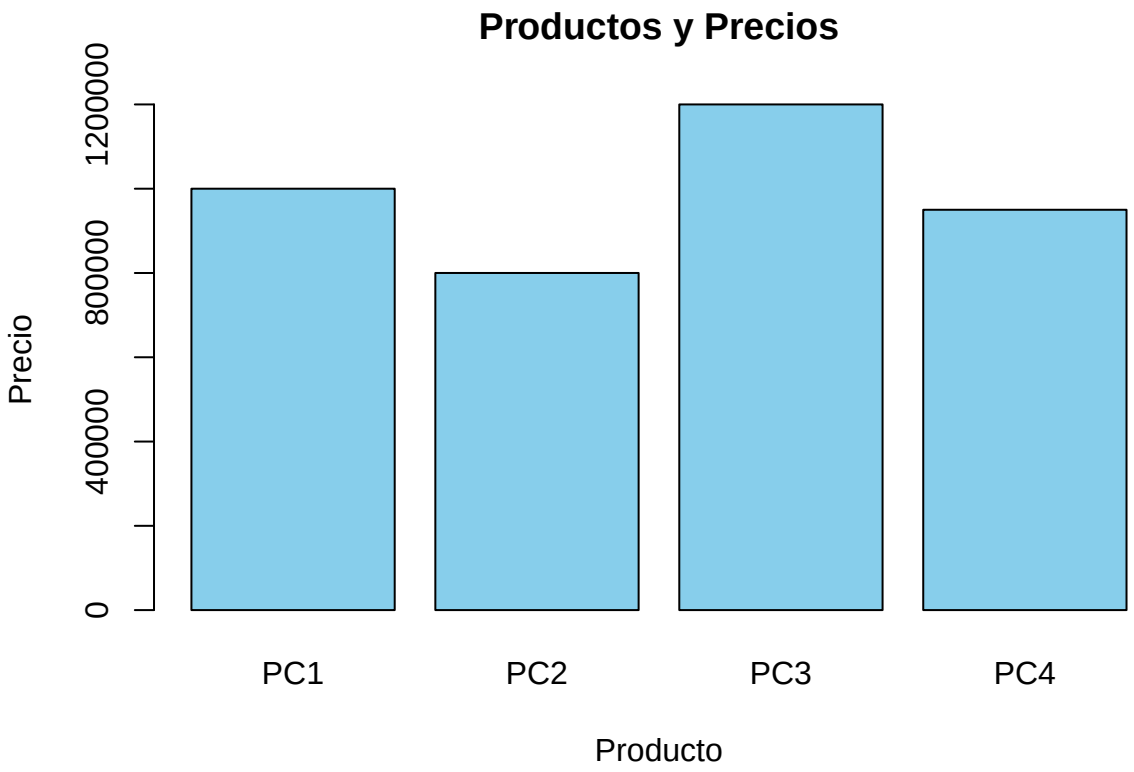

Figure 1: Productos y Precios

```
print(productos)
```

```
##   Producto  Precio
## 1      PC1 1000000
## 2      PC2  800000
## 3      PC3 1200000
## 4      PC4  950000
```

5 Data frames

5.1 Los data frames son estructuras de datos bidimensionales que se usan para almacenar conjuntos de datos.

5.2 Puedes crear data frames utilizando la función data.frame().

edades <- c(25, 30, 35) nombres <- c("Juan", "María", "Pedro") df <- data.frame(Edad = edades, Nombre = nombres)

6 Indexación de vectores y data frames

6.1 Puedes acceder a elementos individuales de un vector o data frame utilizando corchetes [].

7 VECTORES

7.1 Un vector es una estructura de datos que almacena números de doble precisión

```
mi_vector_a <- c(1,12,54,23,12,65,34,12,56,66)
mi_vector_b <- seq(1:10)
mi_vector_a
```

```
##  [1]  1 12 54 23 12 65 34 12 56 66
```

```
mi_vector_b
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

8 Indexación de vectores y data frames

8.1 Puedes acceder a elementos individuales de un vector o data frame utilizando corchetes [].

9 MATRICES

Las matrices se parecen a los vectores, pero tienen filas y columnas. Se alimentan de vectores

```
# Crear una matriz de 10x10 con números enteros aleatorios entre 1 y 100
mi_matriz_a <- matrix(sample(1:100, size = 100, replace = TRUE), nrow = 1

# Imprimir la matriz
print(mi_matriz_a)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]   40   12   92   90   65   44  100   53   63    63
##  [2,]   80   67   27   88   25   59   10   45   45    94
##  [3,]   88   47   56   31   89   62   33    2    7    24
##  [4,]   56   74   71   62   39   53   60   99   97    89
##  [5,]   92   21   56   73   33   55   30   78   54    92
##  [6,]   37   85   38   58   39   52   67   52   23    28
##  [7,]   24   89   97   29   86   45   39   75   23    73
##  [8,]   30   90   40   55   59   89   15   47   69    81
```

```
##  [9,]   29   93   28   25    8   69   19   56   16    70
## [10,]   73   74   86   23   95   17   10   44   81    88
```

Como traer la fila 4 completa?

```
mi_matriz_a[ 4 , ]
```

```
##  [1] 56 74 71 62 39 53 60 99 97 89
```

Como traer la columna 4 completa?

```
mi_matriz_a[  ,4 ]
```

```
##  [1] 90 88 31 62 73 58 29 55 25 23
```

```
start_time <- Sys.time()
mi_matriz_a
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]   40   12   92   90   65   44  100   53   63    63
##  [2,]   80   67   27   88   25   59   10   45   45    94
##  [3,]   88   47   56   31   89   62   33    2    7    24
##  [4,]   56   74   71   62   39   53   60   99   97    89
##  [5,]   92   21   56   73   33   55   30   78   54    92
##  [6,]   37   85   38   58   39   52   67   52   23    28
##  [7,]   24   89   97   29   86   45   39   75   23    73
##  [8,]   30   90   40   55   59   89   15   47   69    81
##  [9,]   29   93   28   25    8   69   19   56   16    70
## [10,]   73   74   86   23   95   17   10   44   81    88
```

```
end_time <- Sys.time()

end_time -start_time
```

```
## Time difference of 0.001534224 secs
```