Minería de Datos para el Análisis de Big Data

Por: Carlos Carreño

ccarrenovi@Gmail.com

Abril,2021

Modulo 2 Introducción al Lenguaje R

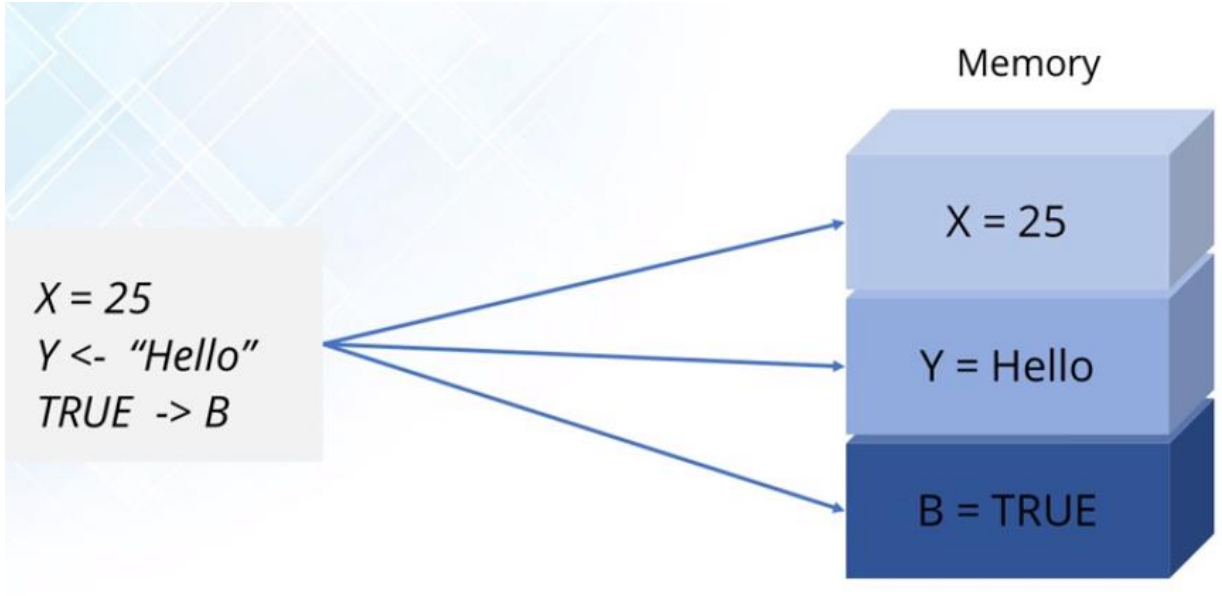
- Introducción
- Variables en R
- Operadores
- Tipos de Datos
- Control de Flujo
- Graficando con R

Introducción

- Porque usar R?
- A parte de ser usado como un lenguaje estadístico, R también puede ser usado como lenguaje de programación para propósito de la analítica.
- R posee herramientas formidables para implementar soluciones de visualización de datos.
- R es simple y fácil de aprender
- R es libre y open source, esto se entiende que uno puede copiar y distribuir libremente el software.

Variables en R

- Cuando crear una variable en R reservas un espacio de memoria para almacenar valores.
- R dinámicamente tipado



Operadores

- Aritméticos
- Asignación
- Relacionales
- Lógicos
- Especiales

Operadores Aritméticos

- + Suma dos operandos o suma unaria
- - resta dos operandos o resta unaria
- * multiplica dos operandos
- / Divide el operando de la izquierda con el de la derecha el resultado es un numero real (**float**)
- ^ Potencia el operando de la izquierda a la potencia de la derecha
- %% Resto o modulo de la división
- %/% El resultado entero de la división

Operadores de Asignación

- = x= algo
- <- x <- algo
- <<- x <<- algo
- -> algo -> x

Operadores relacionales

- > True si el operando de la izquierda es mas grande que el de la derecha
- < True si el operando de la izquierda es menor que el de la derecha
- == True si los operandos son iguales
- != True si los operandos son diferentes
- >= True si el operando de la izquierda es mayor o igual que el de la derecha
- =< True si el operando de la izquierda es menor o igual que el de la derecha

Operadores Lógicos

- & and lógico
- | or lógico
- ! Not o negación

Operadores Especiales

- : Crea una serie de números secuenciales para un vector
- %in% Este operador es usado para identificar si un elemento pertenece a un vector, retorna un boolean.

Tipos de Datos

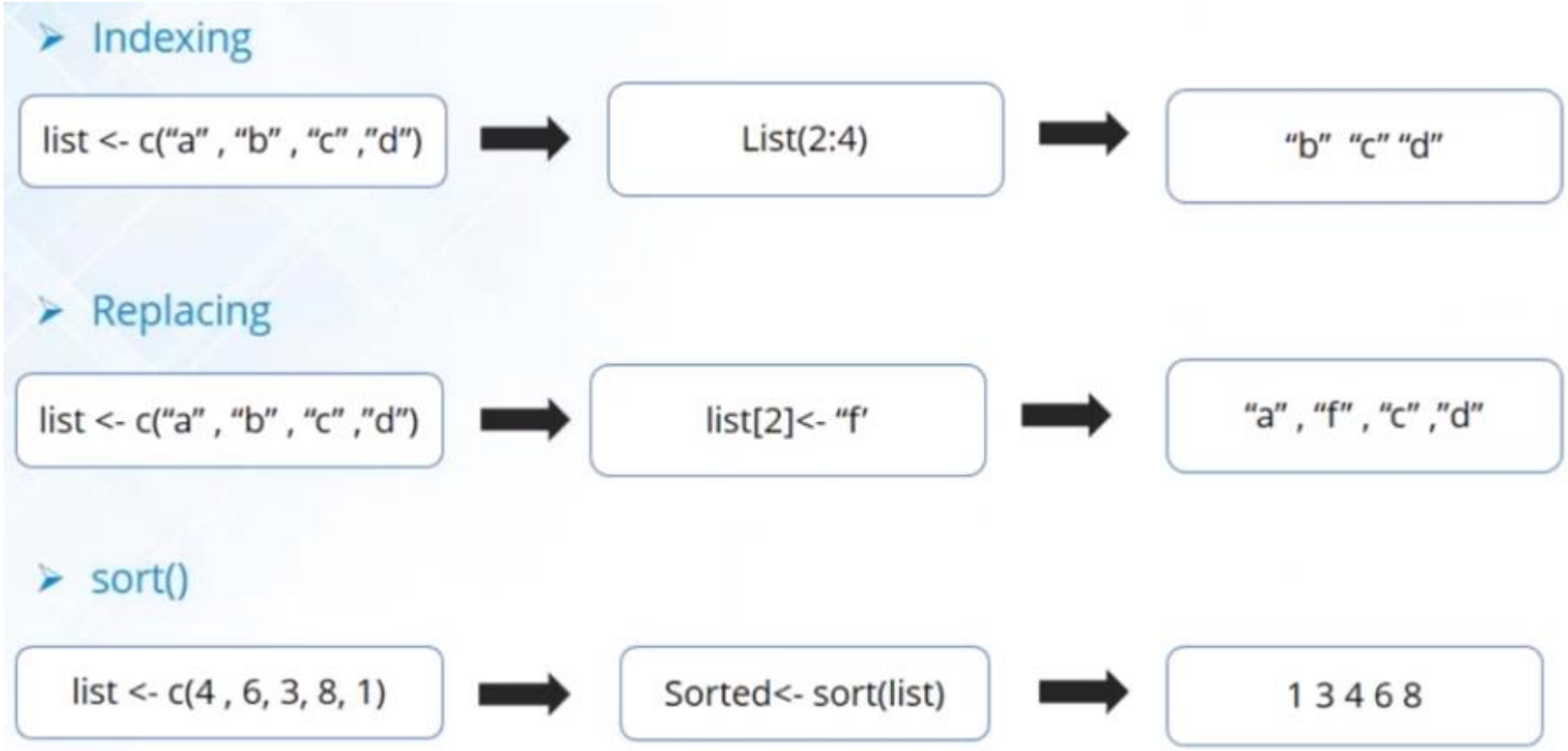
- Vectores
- Listas
- Arrays
- Matrices
- Factores
- Data Frames

Nota: No se requiere declarar la variable de un tipo determinado.

Vectores

- Un vector es una secuencia de datos del mismo tipo de dato básico.
- Ejemplo
- V=c(1,2,3,4,5)

Operaciones con Vectores

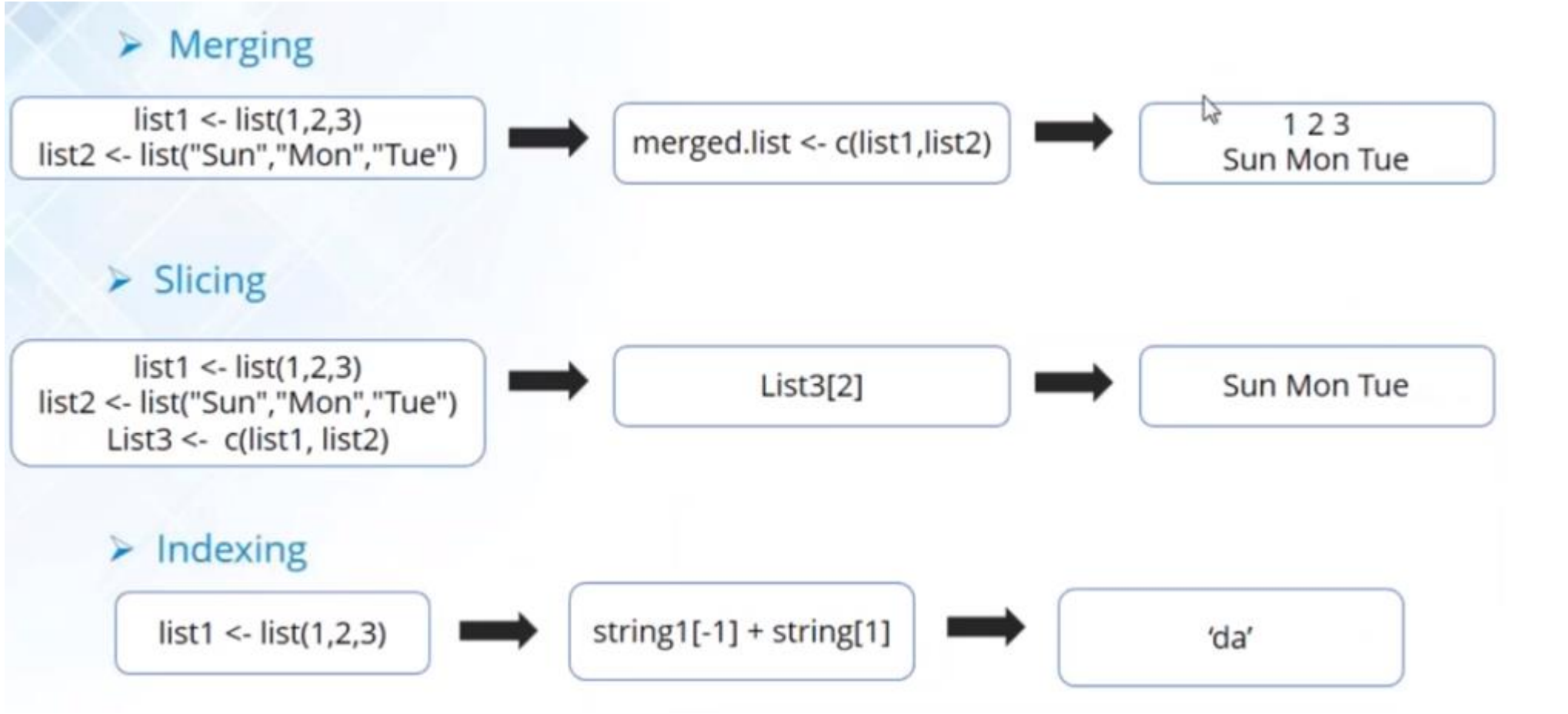


Listas

- Las listas en R son objetos que contienen elementos de diferente tipo como números, cadenas, vectores y otros dentro de ellas.

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc", "dd", "ee")
>x = list(n, s, TRUE)
```

Operaciones con Listas



Arrays

- Los Arrays son objetos de datos en R que pueden almacenar datos en mas de dos simensiones
- El array toma los vectores como entrada y usa los valores del parámetro dim para crear el array.

```
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
result <- array(c(vector1,vector2),dim = c(3,3,2))
```

Matrices

- Las matrices son objetos R en el cual los elementos están dispuestos en dos dimensiones
- Una matriz es creada usando la función matrix()

```
matrix(data, nrow, ncol, byrow, dimnames)
```

- **Nota**: El numero de elementos debe ser un múltiplo del numero de columnas

Factores

- Los factores son objetos de datos que pueden ser usados para categorizar los datos y almacenarlos como niveles o categorías
- Ellos son almacenados como cadenas y enteros
- Son útiles en el análisis de datos y en el modelamiento estadístico

```
data <- c("East","West","East","North","North","East","West","West","East")
factor_data <- factor(data)
```

Data Frame

- Un Data Frame es una tabla de dos dimensiones como la estructura de un array en la cual cada columna contiene valores de una variable y cada fila contiene un conjunto de valores para cada columna.

```
emp_id = c (1:5),
emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
salary = c(623.3,515.2,611.0,729.0,843.25),
emp.data <- data.frame(emp_id, emp_name, salary)
```

Sentencias de Control de Flujo

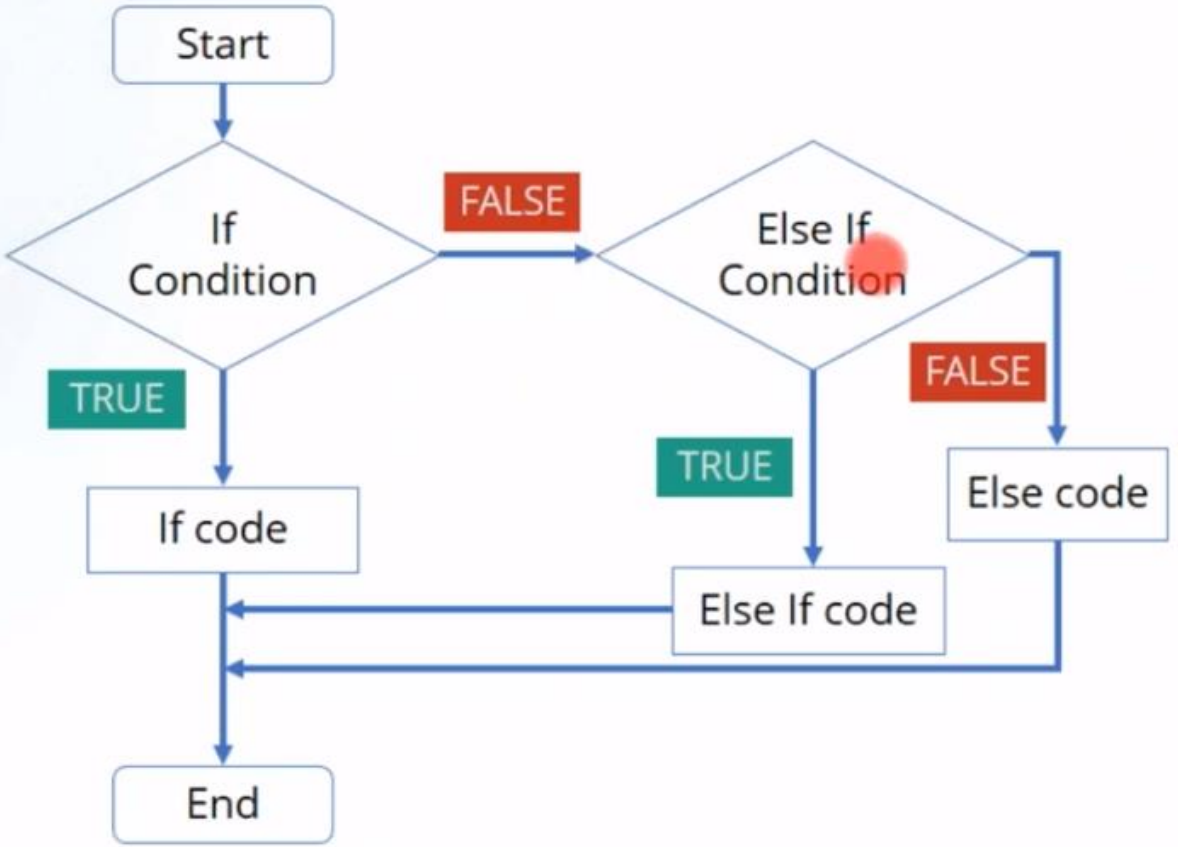
- Las siguientes sentencias permiten control el flujo del programa

Sentencia if .. else

2. if...else statement

Syntax:

```
if (condition 1):
    statements 1 ...
    .
    .
    .
else
    statements n ...
```



Ejemplo: If .. Else

- Ejemplo: sentencia if .. Else

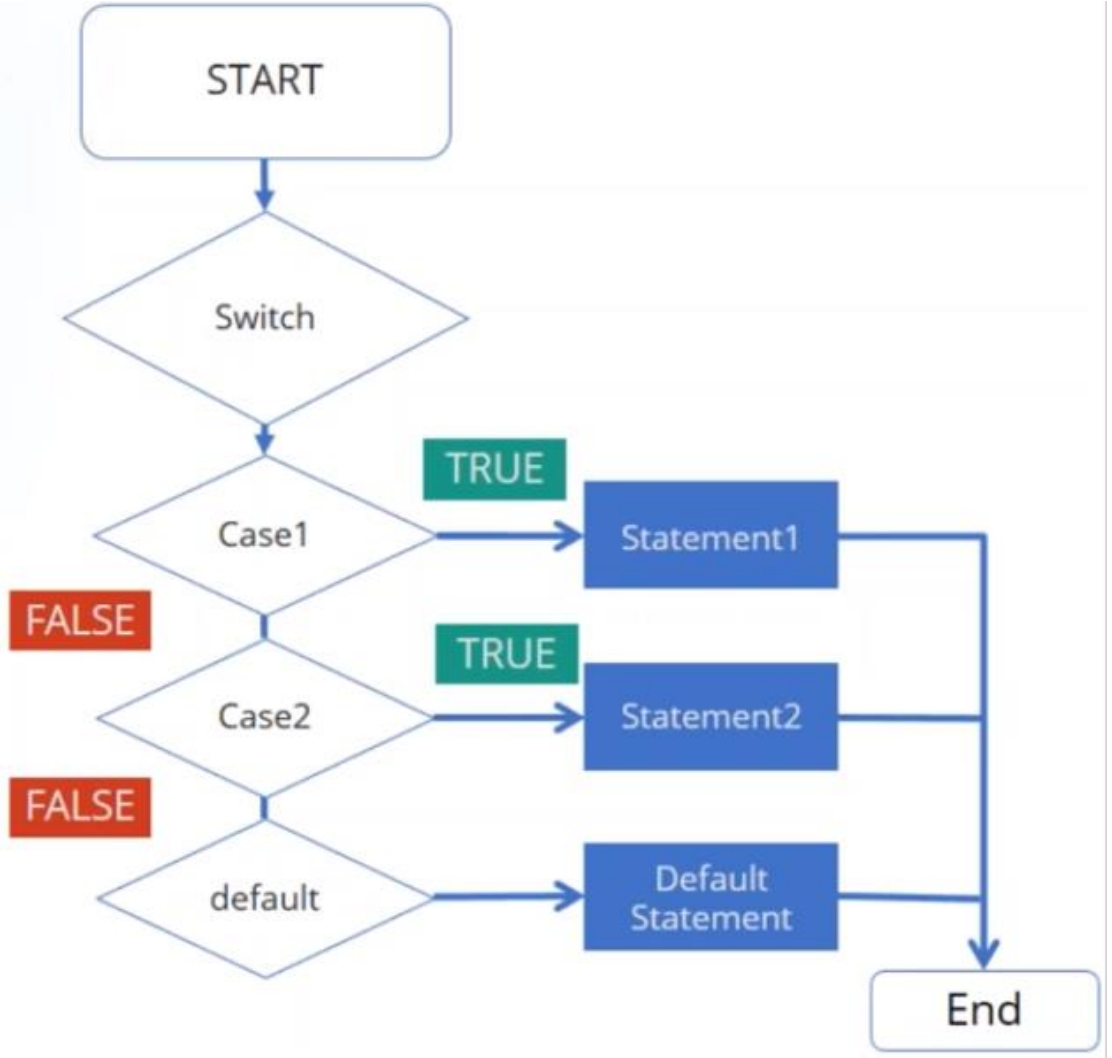
```
1  x = 3
2  y = 5
3
4  if (x > y){
5    print( "x es mayor que y")
6  } else {
7    print ("x es menor que y o son iguales")
8  }
9
10
```

Sentencia Switch

3. Switch statement

Syntax:

```
switch (expression,
value1:  Statement1
value2:  Statement2
.
.
, default Statement
)
```



Ejemplo: Switch

- Ejempplo: Sentencia Switch en R

```r
v = c(150, 200, 300, 350, 400)
option = "mode"

switch(option,"mean" = print(mean(v)),
              "mode" = print(mode(v)),
              "median"  = print(median(v)),
               print("ninguno")
)

```

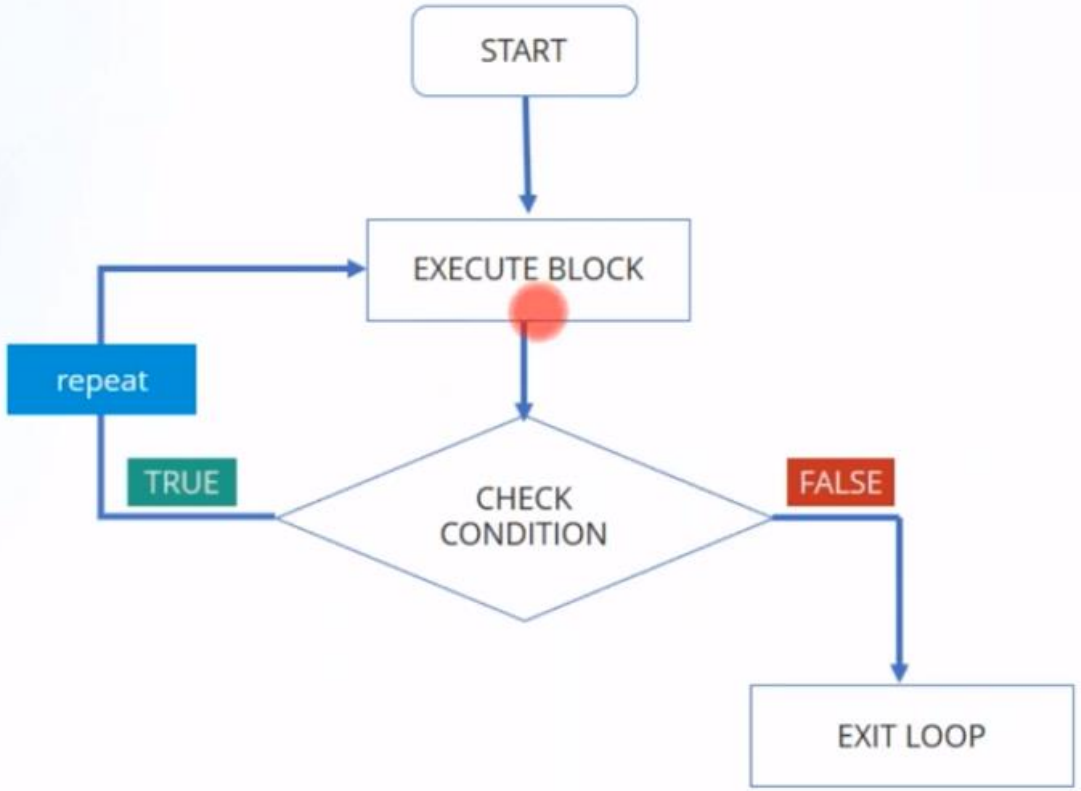
Loops

- Repeat
- While
- For

Sentencia Repeat

4. repeat statement

```
Syntax:
repeat {
   commands
   if(condition) {
      break
   }
}
```



Ejemplo: Repeat

- Ejemplo: Sentencia Repeat

```
1  x = 2
2  repeat{
3    x = x ^ 2
4    print(x)
5    if (x > 64)
6      break
7  }
8
9
```

9:1 (Top Level)

Console Terminal Jobs

~/

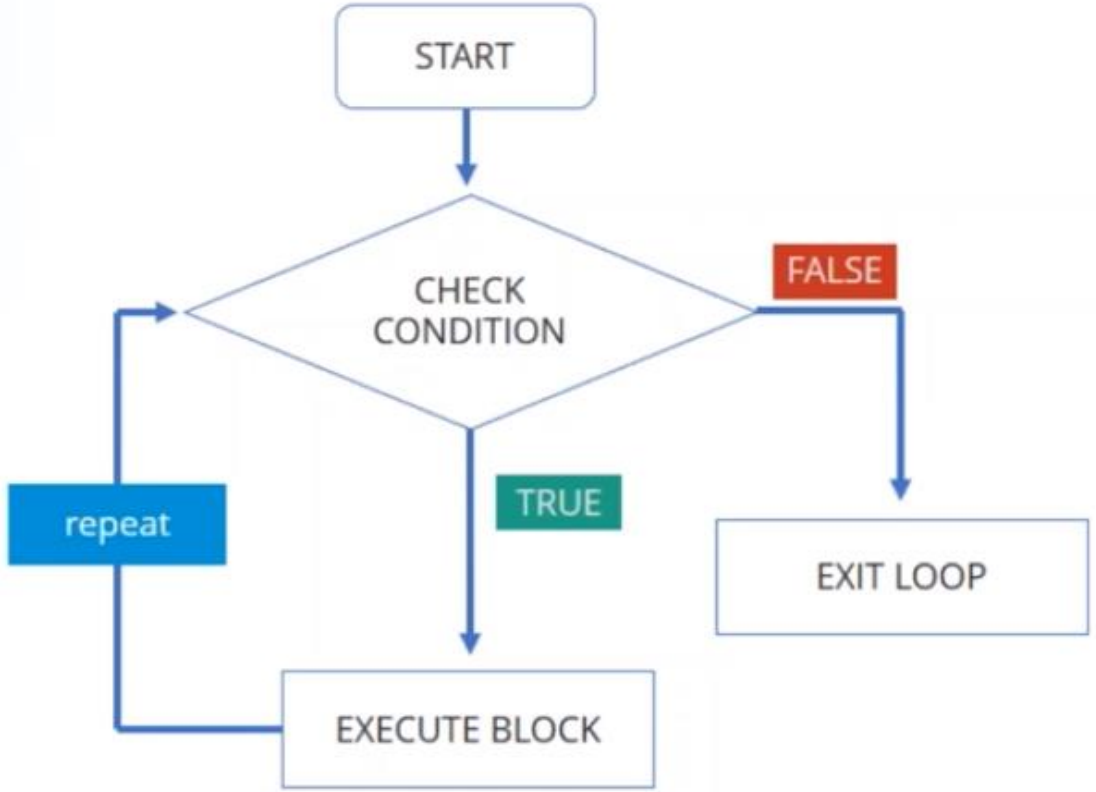
```
> source('~/.active-rstudio-document')
[1] 4
[1] 16
[1] 256
>
```

Sentencia While

5. while statement

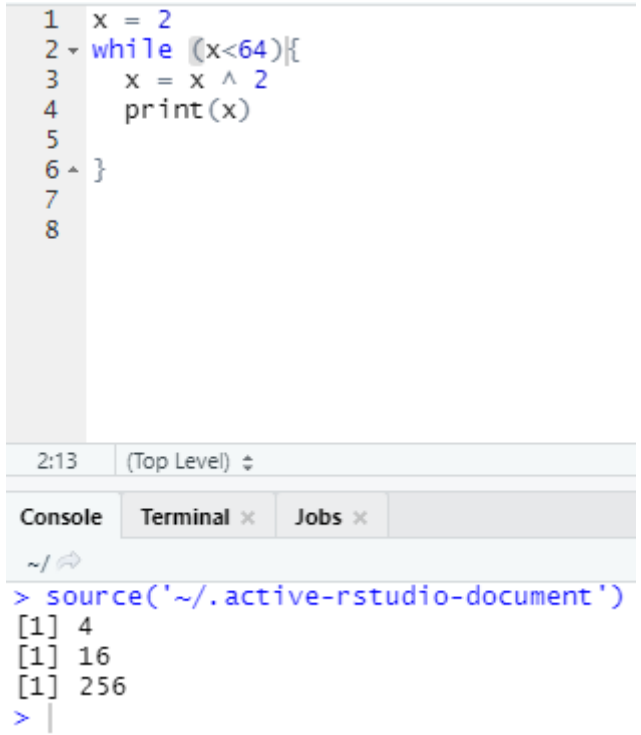
Syntax:

```
while (condition is True)
    {
        statements...
    }
```



Ejemplo: while

- Ejemplo de sentencia while

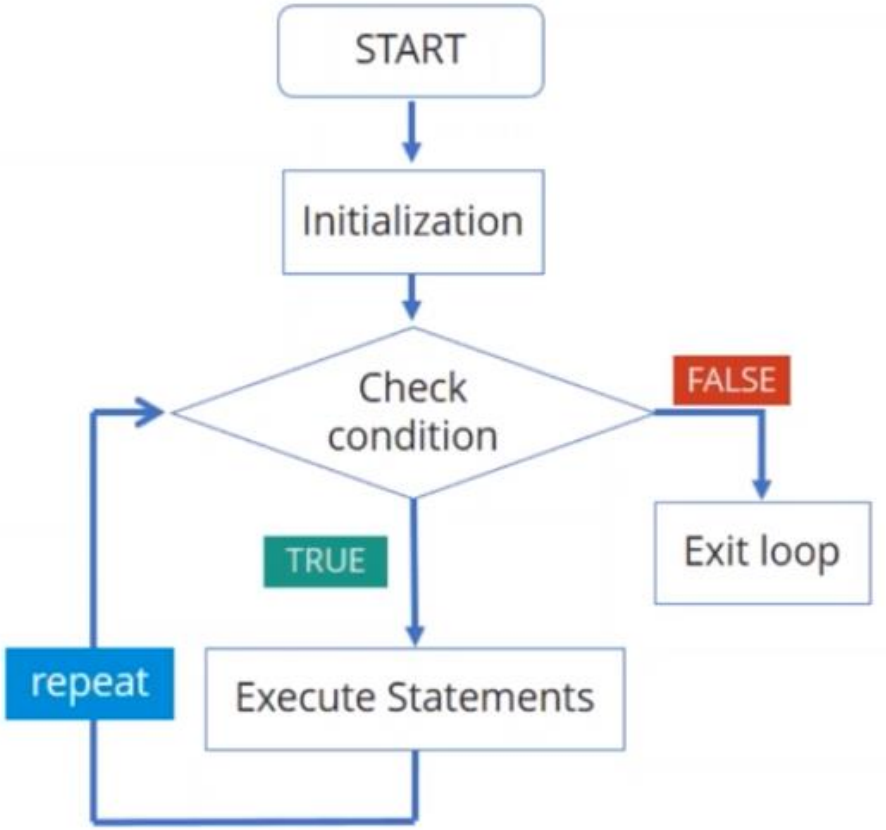


Sentencia For

6. for statement

Syntax:

```
for(value in vector)
{
        statements...
}
```



Ejemplo: For

- Ejemplo: Sentencia For

```
1  v = c(7,19,37,49,65)
2  for(i in v){
3    print(i)
4  }
5
```

5:1 (Top Level)

Console Terminal Jobs

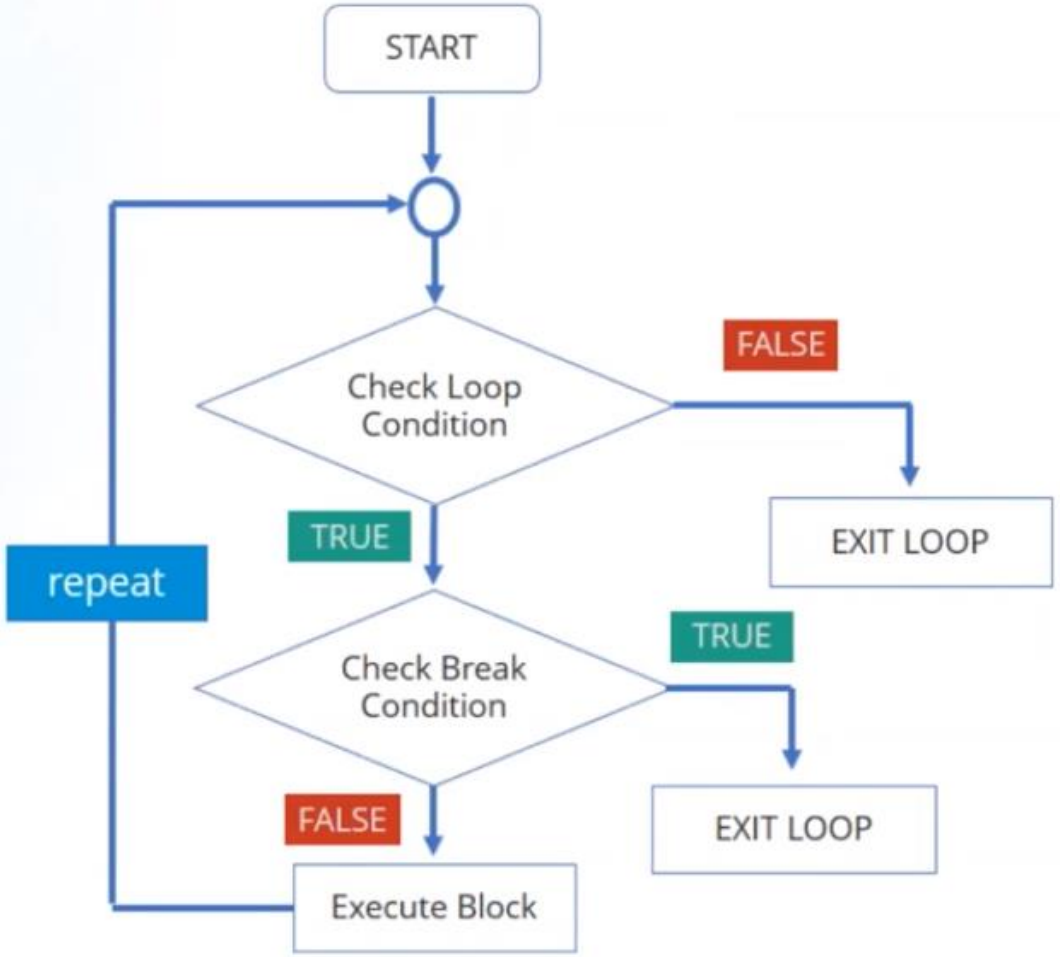
```
~/
> source('~/.active-rstudio-document')
[1] 7
[1] 19
[1] 37
[1] 49
[1] 65
>
```

Break

7. break

Syntax:

```
break;
```

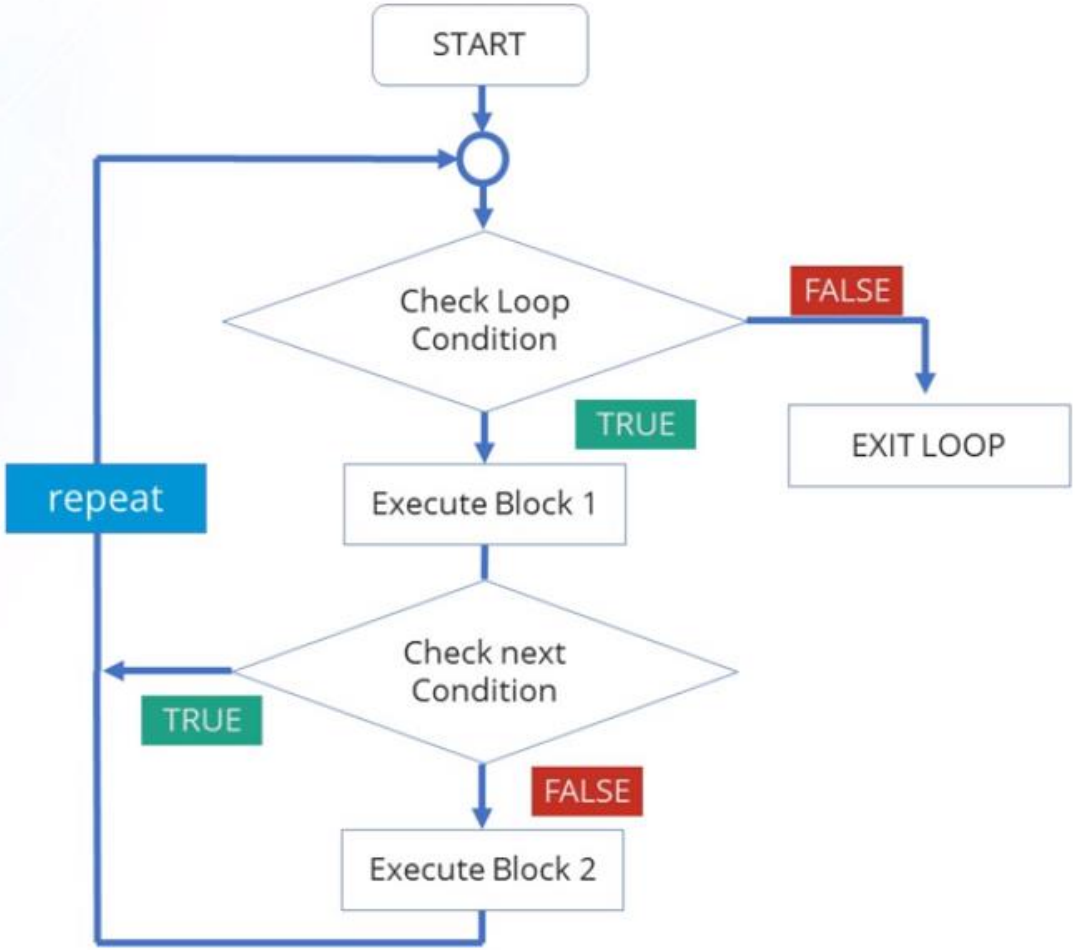


Next

8. next

Syntax:

```
next;
```



Ejemplo Next

- Ejemplo de sentencia Next

```
1  v = c(7,19,37,49,65)
2  for(i in v){
3    if( (i%%7) == 0 )
4      next
5    print(i)
6  }
7
8
```

8:1 (Top Level)

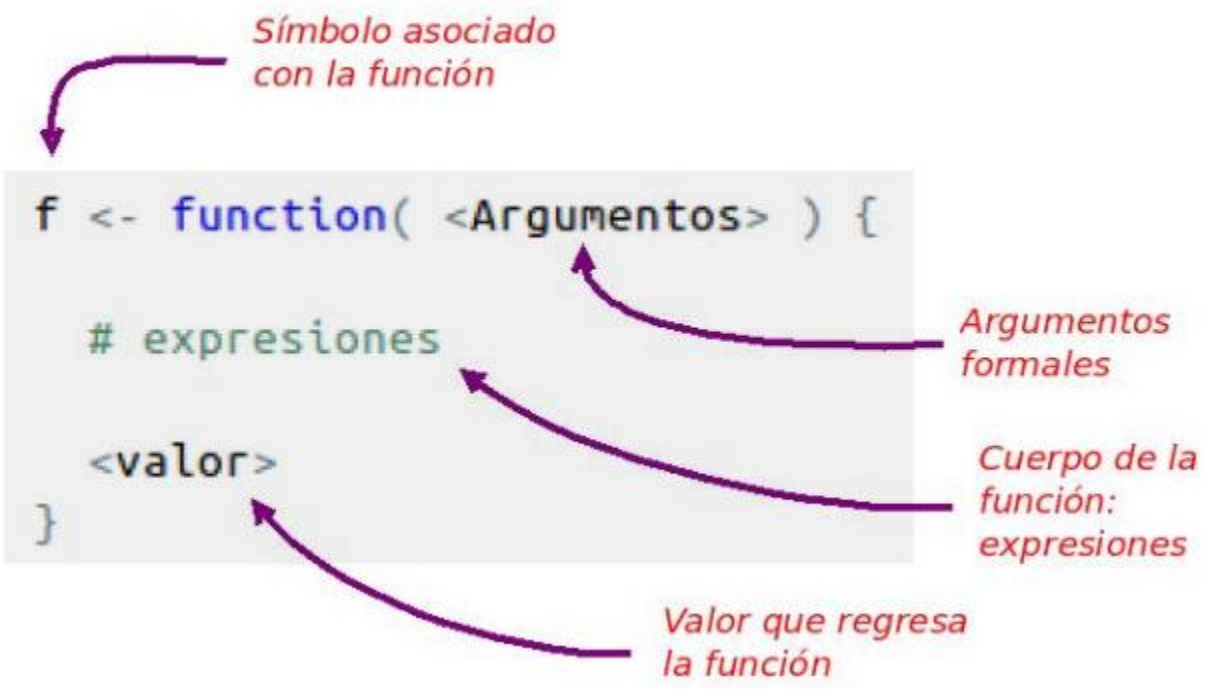
Console Terminal Jobs

~/

```
> source('~/.active-rstudio-document')
[1] 19
[1] 37
[1] 65
>
```

Creación de Funciones

- La sintaxis general para crear una funcion en R es:



Creación de Funciones

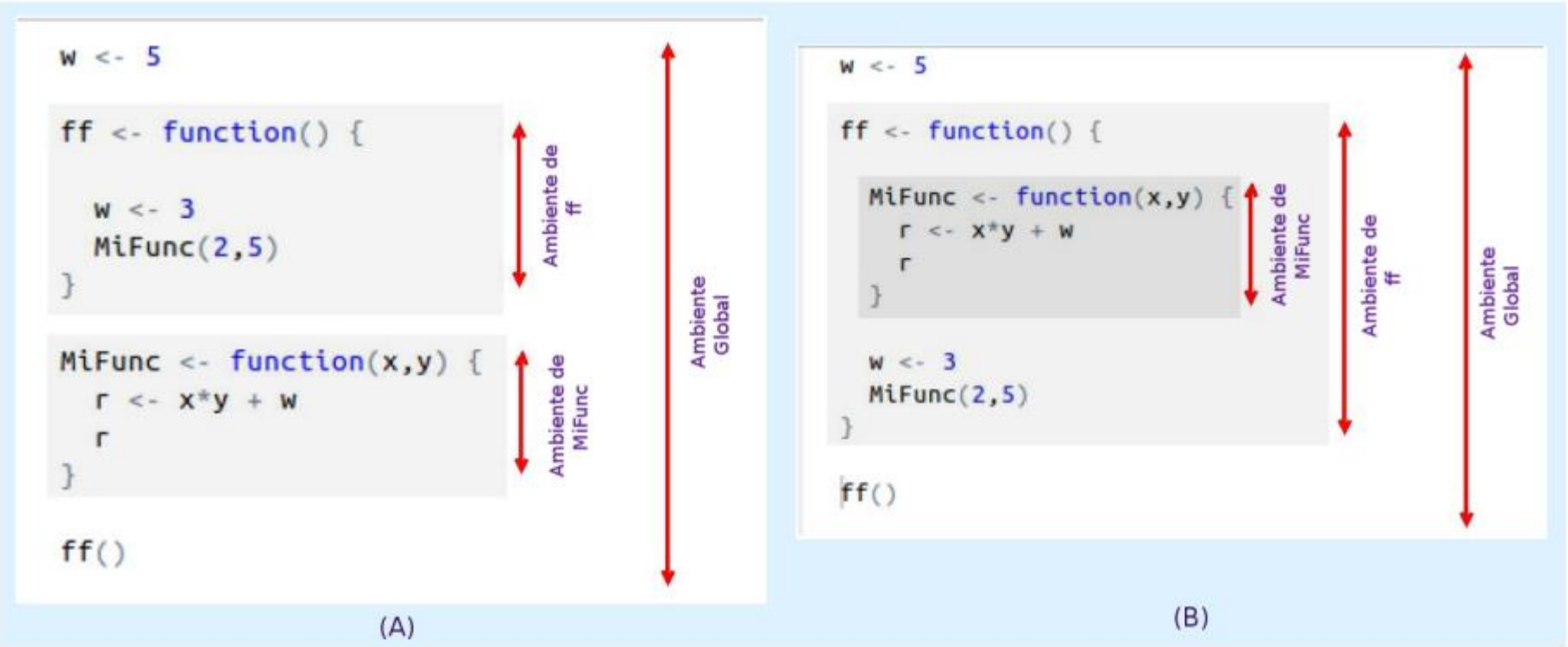
```
> MisFunciones.area <- function(radio,pi=3.1416){
    result <- pi*radio^2
    return (result)
 }
> MisFunciones.area(0.4567)
[1] 0.6552589
>
```

Argumento Especial

- Sirve para transferir un número variable de argumentos a otra función.

```
> s3 <- function(x, y, z=0){
+     r <- x +y +z
+      r
+ }
> s2 <- function(x,y,...){
+       r <- s3(x,y,...)
+       r
+ }
> s2(3,4)
[1] 7
> s2(3,4,9)
[1] 16
```

Alcance de las Variables



Recursividad

- R soporta la recursividad, es decir la posibilidad de una función de llamarse o invocarse a sí misma

```
MiFact <- function(n) {
    if (n==0) return (1) # salida inmediata
    if (n > 0) return (n*MiFact(n-1))
    return (NULL) # caso fallido
}
# Ahora se usa la función con 5 y 8
MiFact(5)
```

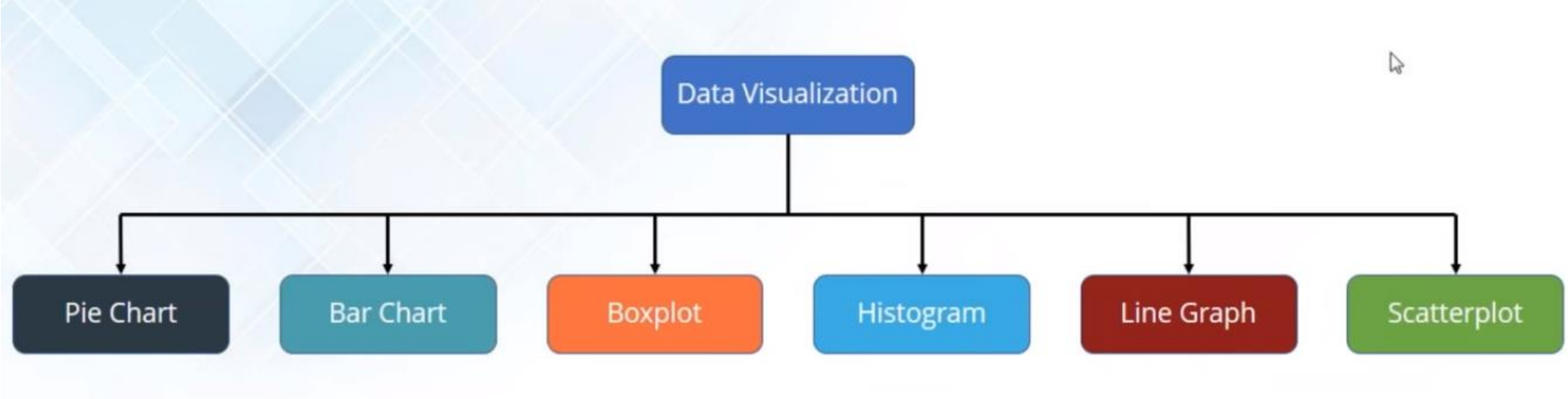
Demo

- Crear una función que dado un numero , determine si el numero es primo.

Nota: Un numero es primo si es divisible solo por el mismo o el 1.

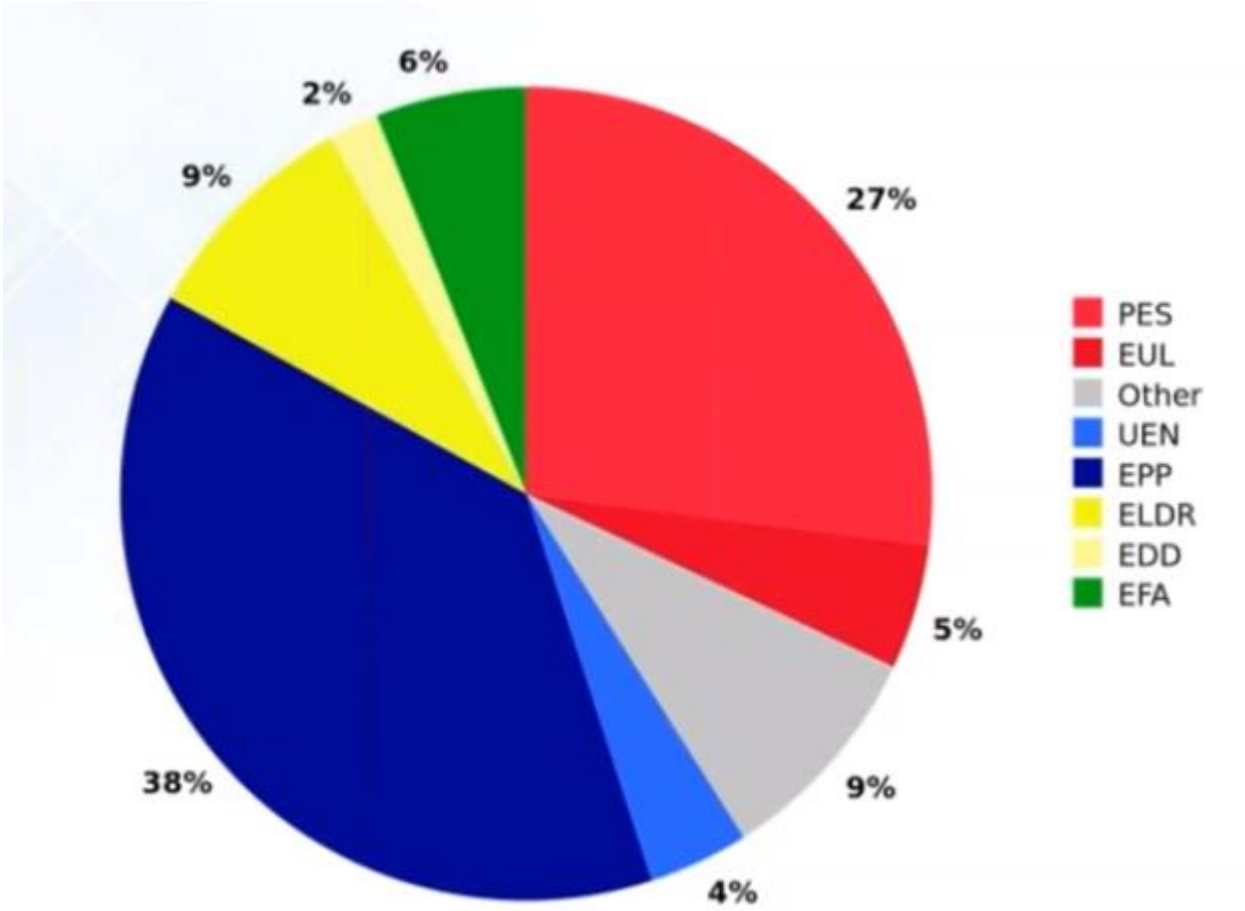
Visualización de Datos

- La visualización de datos es un aspecto importante de la analítica por que ayuda a presentar de manera simple los patrones.



Pie Chart

- Pie chart sirve para comparar las partes de dentro de un todo



Ejemplo: Pie Chart

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Untitled1* | Untitled2*

Source on Save | Run | Source

```
1  v = c(14,34,78,56,98)
2  names = c("Hr","Marketing","Finance","Sales","IT")
3  pie(v,names, col = rainbow(length(v)))
```

3:39 (Top Level) R Script

Console | Terminal | Jobs

~/

```
> source('~/.active-rstudio-document')
>
```

Environment | History | Connections | Tutorial

Import Dataset | List

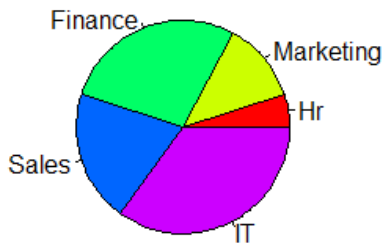
Global Environment

values

i	7
names	chr [1:5] "Hr" "Marketing" "Finance" "Sales" "IT"
option	"mode"
v	num [1:5] 14 34 78 56 98
x	256

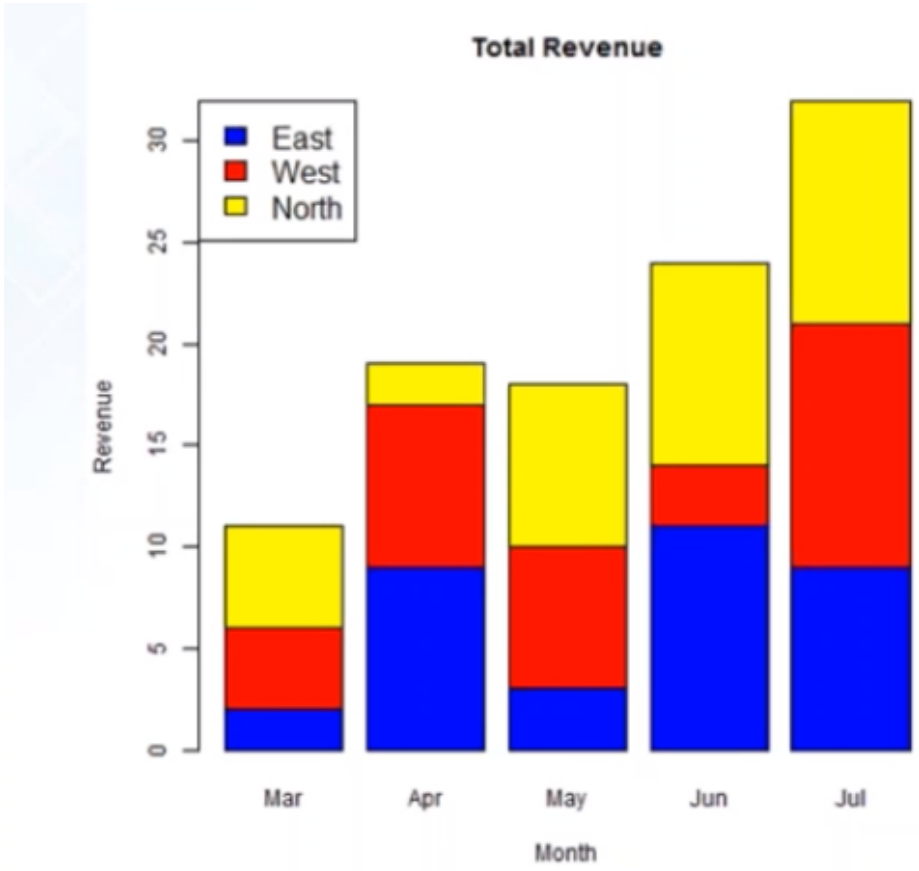
Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish

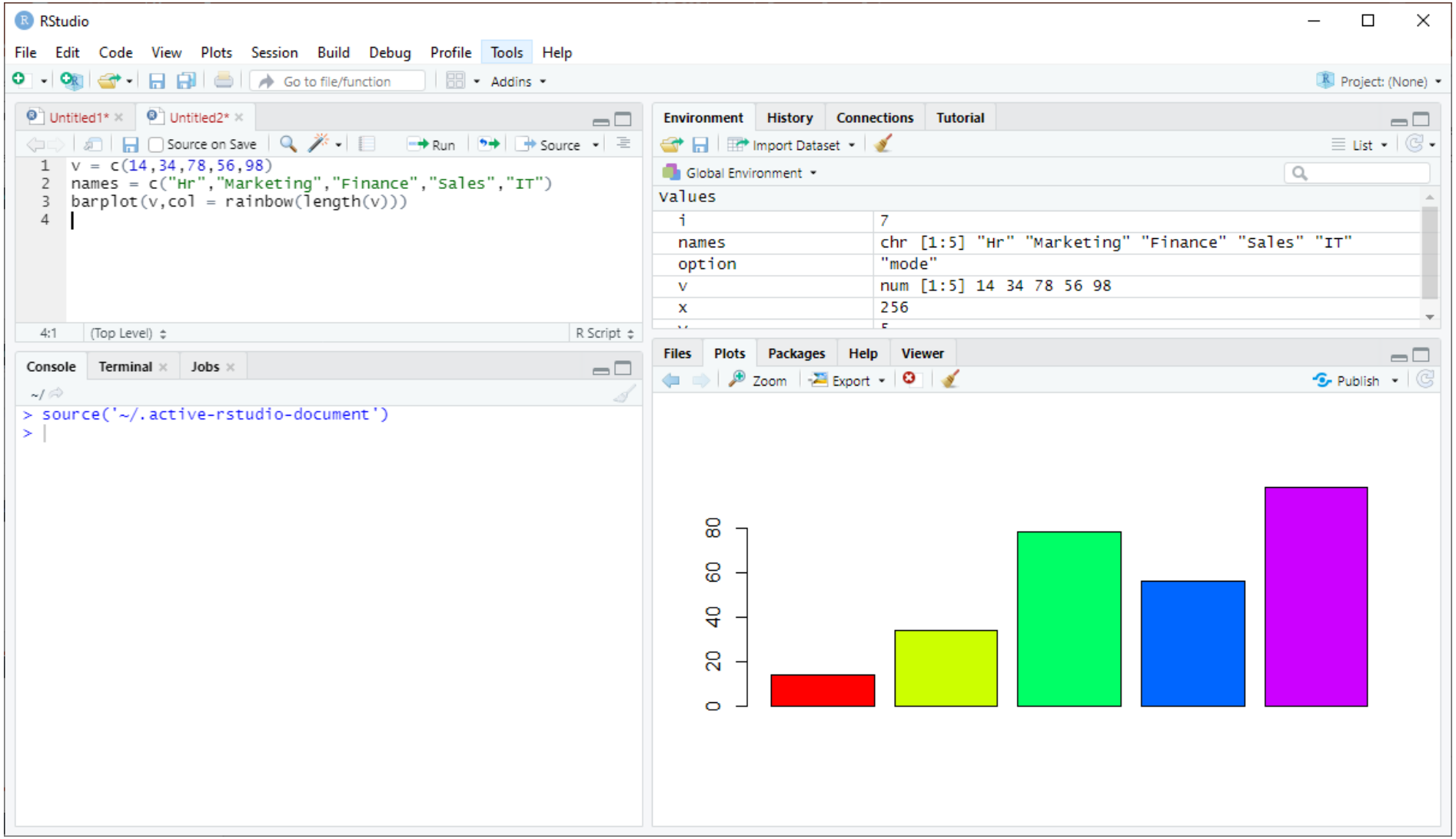


Bar Chart

- Usado para comparar cosas entre diferentes grupos o cambios en el tiempo.

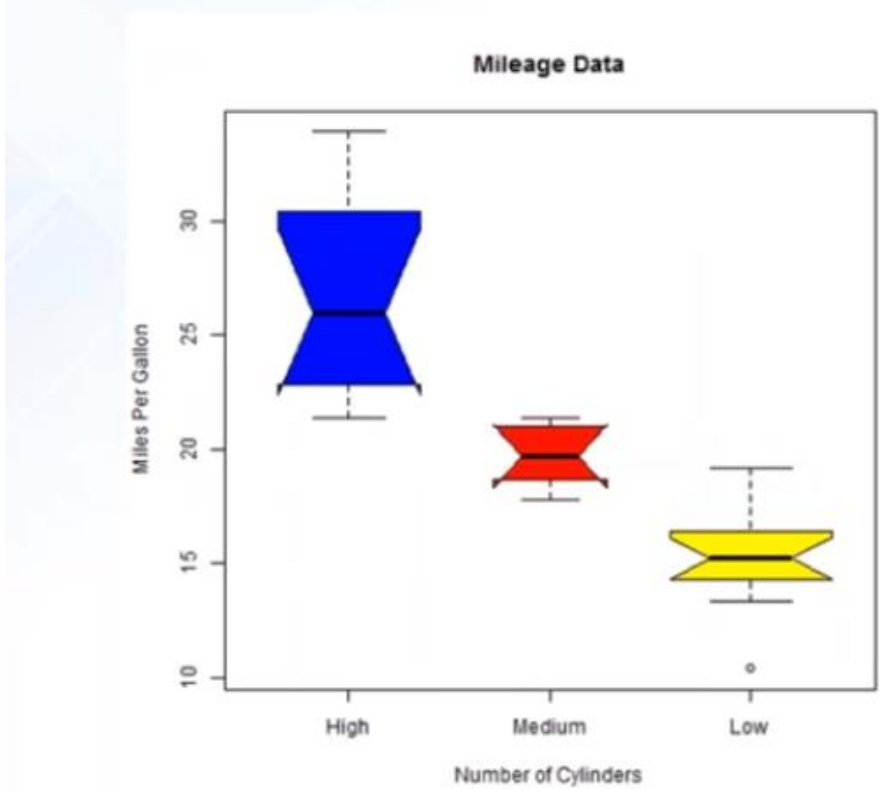


Ejemplo : Bar Chart



Boxplot

- Boxplot son usados para resumir datos desde múltiples fuentes y mostrar los resultados en un solo grafico



Ejemplo: BoxPlot

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Untitled1* Untitled2* Untitled3*

```
1  boxplot(mpg ~ cyl, data = mtcars, xlab="Numero de Cilindros",
2          ylab="Millas por Galon", main = "Datos de las Millas",
3          col = c("red","yellow","blue"))
```

3:38 (Top Level) R Script

Environment History Connections Tutorial

Global Environment

Values

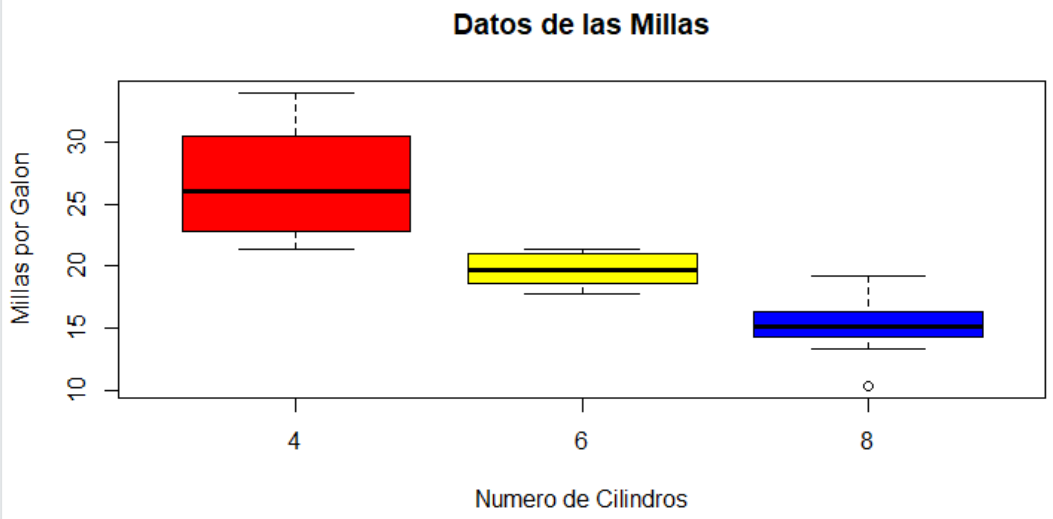
i	7
names	chr [1:5] "Hr" "Marketing" "Finance" "Sales" "IT"
option	"mode"
v	num [1:5] 14 34 78 56 98
x	256
y	5

Console Terminal Jobs

```
> mtcars
                     mpg cyl  disp  hp drat    wt  qsec vs am gear
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3
```

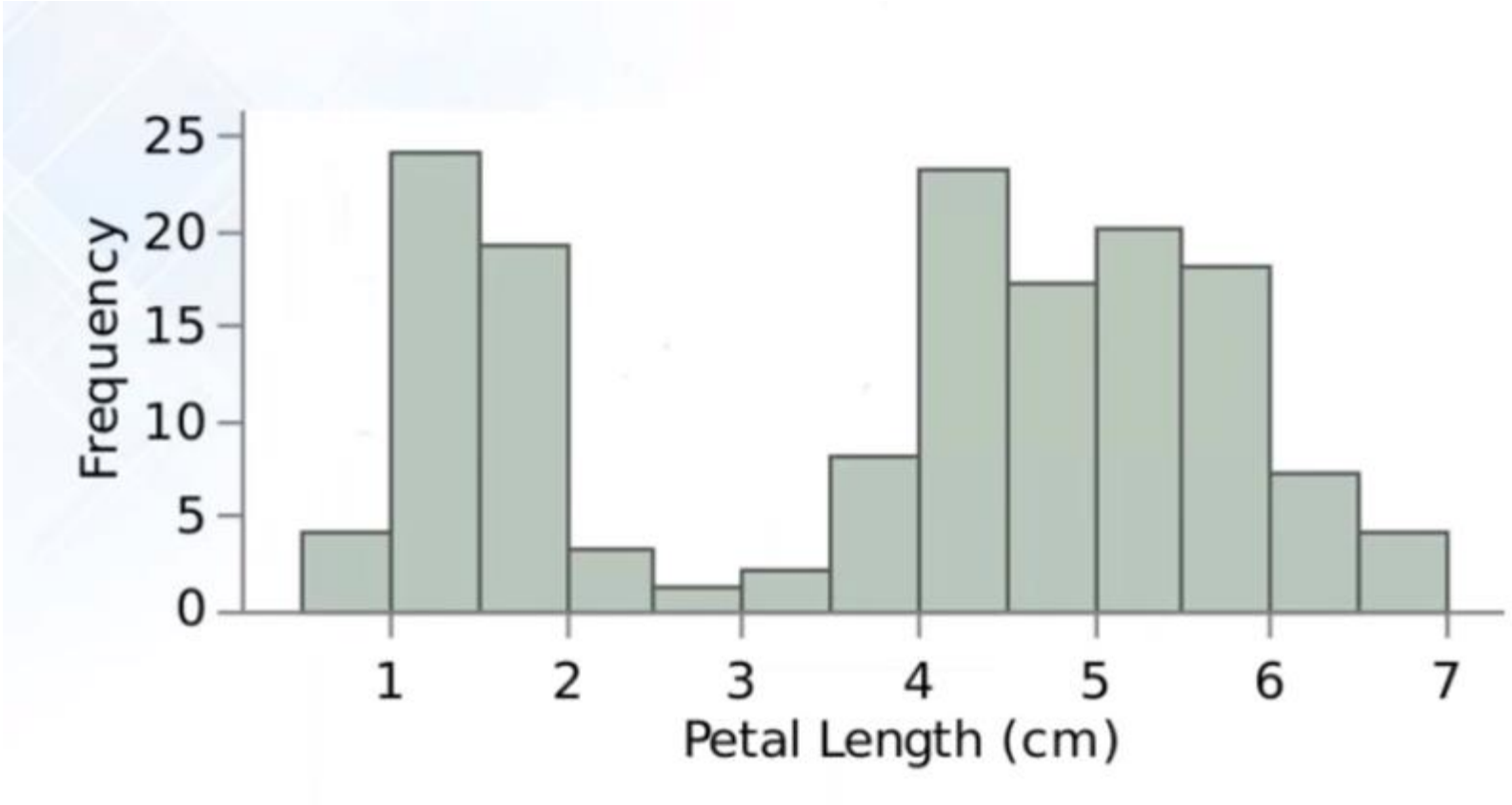
Files Plots Packages Help Viewer

Zoom Export Publish



Histograma

- Los histograma son usados para graficar la frecuencia de ocurrencias en datos continuos que han sido divididos en clases llamados bins



Ejemplo: Histograma

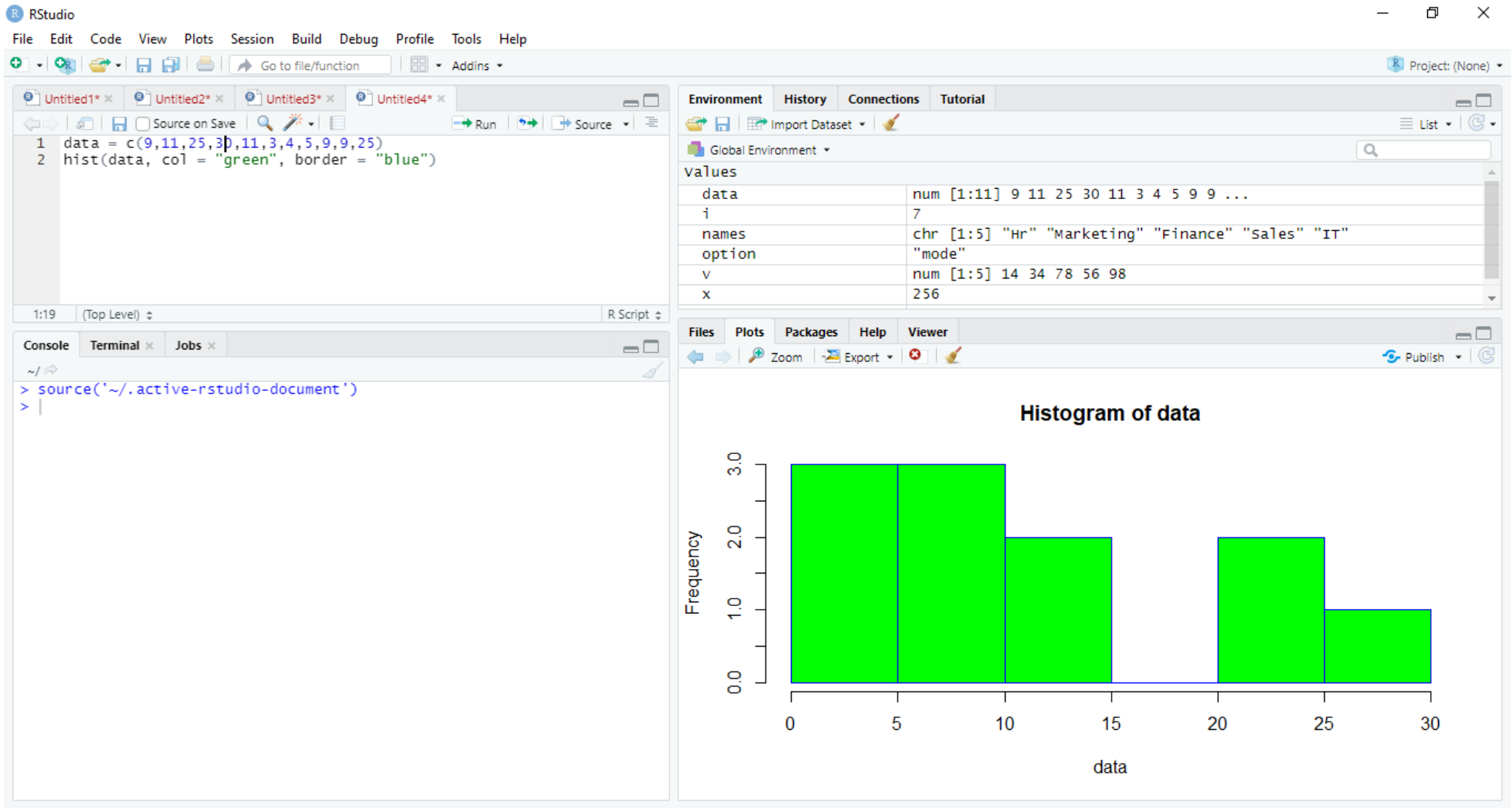
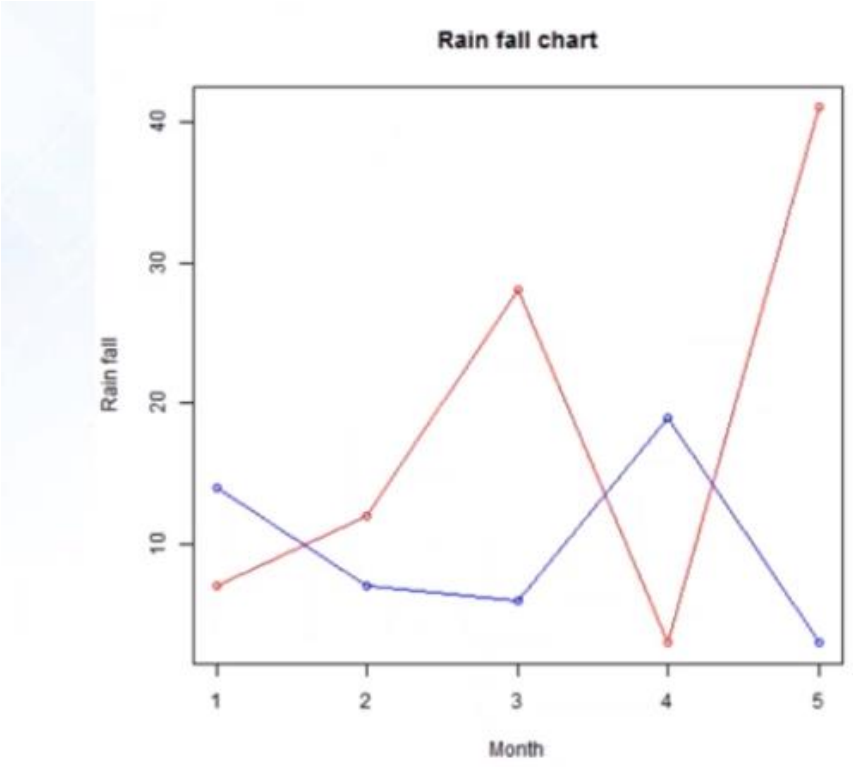


Grafico de Líneas

- Un grafico de líneas se utiliza para realizar el seguimiento de los cambios en cortos y largos periodos de tiempo.



Ejemplo: Grafico de Líneas

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Untitled1* Untitled2* Untitled3* Untitled4*

```
1  data = c(9,11,25,30,11,3,4,5,9,9,25)
2  plot(data, type="o", col = "green")
```

2:36 (Top Level) R Script

Console Terminal Jobs

```
> source('~/.active-rstudio-document')
>
```

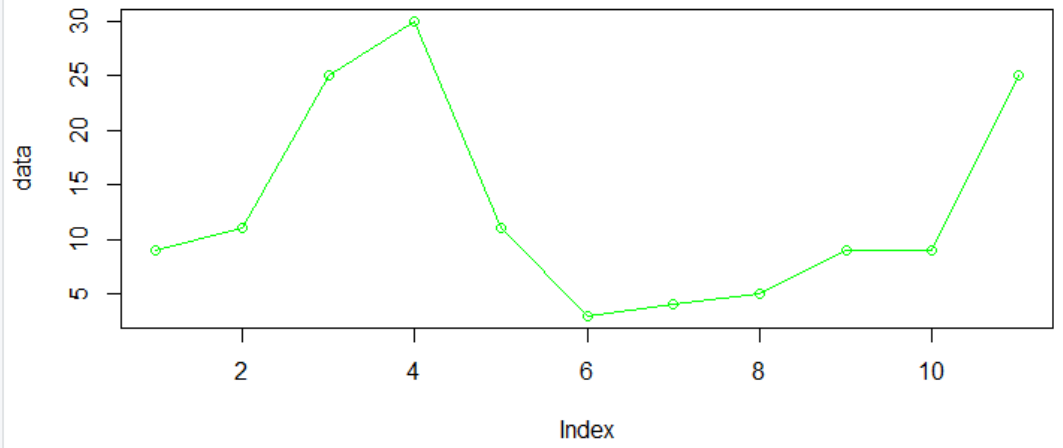
Environment History Connections Tutorial

Global Environment

values	
data	num [1:11] 9 11 25 30 11 3 4 5 9 9 ...
i	7
names	chr [1:5] "Hr" "Marketing" "Finance" "Sales" "IT"
option	"mode"
v	num [1:5] 14 34 78 56 98
x	256

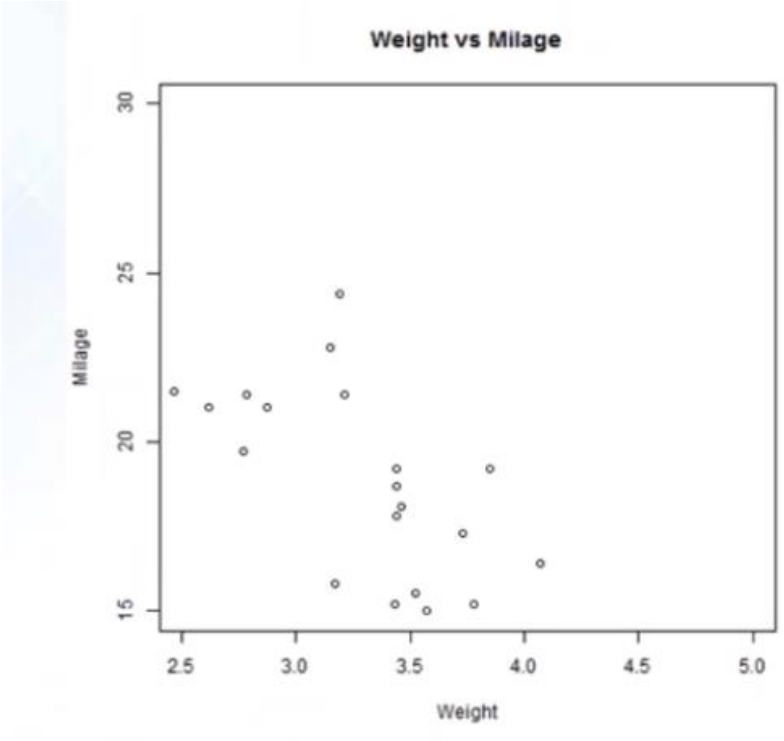
Files Plots Packages Help Viewer

Zoom Export Publish

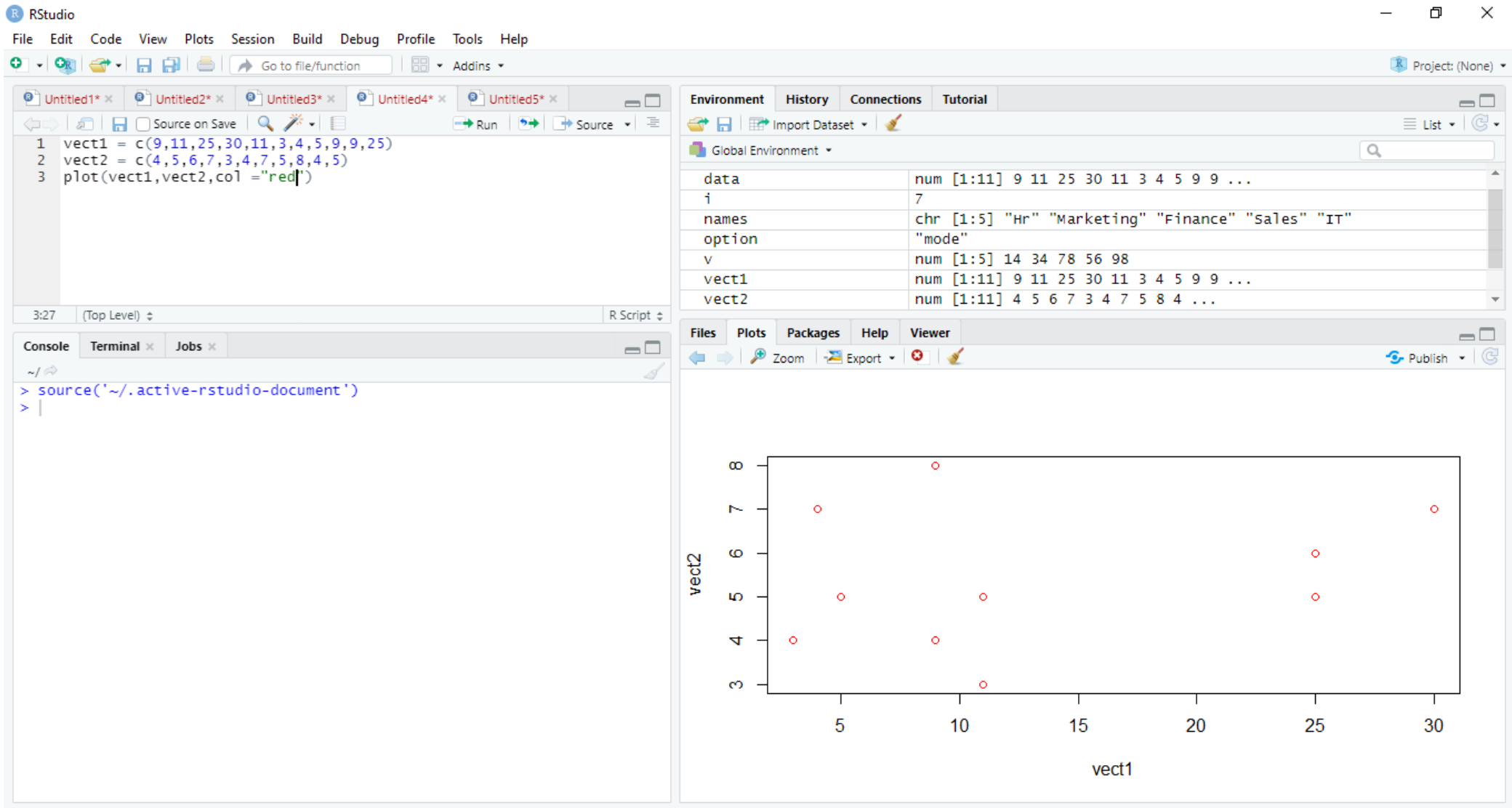


Scatter Plot

- Scatter plot muestra como una variable es afectada por otra (correlación)



Ejemplo: Scatter Plot



Demo

- Crear un grafico de dispersión

Preguntas

- Alguna pregunta?