

# Técnicas y Herramientas Modernas I-Módulo de programación en Rstudio

Juan Manuel Pegorin<sup>a,1,\*</sup>, Analia Quiroga<sup>a</sup>, Nicolas Mondaca<sup>a,2</sup>, Franco Carricart<sup>a,2</sup>

<sup>a</sup>Centro universitario M5500 Mendoza

---

## Abstract

Este trabajo se realizó con el objetivo de ahondar en los conocimientos de el lenguaje de programación estadístico R. Se resolvieron ejercicios con una complejidad ascendente y se utilizaron diversas herramientas del lenguaje para resolverlos. Además se explica dentro de cada inciso, lo que se plantea a resolver y el enfoque que se ha dado a la resolución correspondiente.

*Keywords:* eficiencia, iteración, modelo matemático

---

### 1.Ejercicio N 1: Generar un vector secuencia.

Se comparó dos códigos que realizan lo mismo con la finalidad de ver cuánto tiempo toman cada uno, con el fin de observar su rendimiento y eficiencia. Ambos códigos sirven para generar un vector con una secuencia numérica.

##Codigo generado con for:

```
A<- c()
start_time<-Sys.time()
for (i in 1:50000) {A[i]<-(i*2)}
head (A)
```

```
## [1]  2  4  6  8 10 12
```

```
tail(A)
```

```
## [1] 99990 99992 99994 99996 99998 100000
```

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

```
## Time difference of 0.0537889 secs
```

##Codigo generado con R

```
ti<- Sys.time()
A<-seq(1,100000,2)
head(A)
```

```
## [1]  1  3  5  7  9 11
```

---

\*Corresponding author

Email addresses: [juanmapegorin@gmail.com](mailto:juanmapegorin@gmail.com) (Juan Manuel Pegorin), [aniquiroga122000@gmail.com](mailto:aniquiroga122000@gmail.com) (Analia Quiroga), [gabrielmondacanicolan@gmail.com](mailto:gabrielmondacanicolan@gmail.com) (Nicolas Mondaca), [carricartfranco9@gmail.com](mailto:carricartfranco9@gmail.com) (Franco Carricart)

<sup>1</sup>Esta es la primera nota del autor.

<sup>2</sup>

```
tail(A)
```

```
## [1] 99989 99991 99993 99995 99997 99999
```

```
tf<-Sys.time()
tf-ti
```

```
## Time difference of 0.004096031 secs
```

## 2.Ejercicio N 2: Implementacion de serie de Fibonacci.

La famosa serie de Fibonacci es la mostrada a continuación.

$$f_0 = 0; f_1 = 1; f_{n+1} = f_n + f_{n-1}$$

En el presente ejercicio se observa cuántas iteraciones se deben realizar para que el valor de la serie en cuestión alcance un número mayor a 1.000.000

```
f0<-0
f1<-1
it<-0
f2<-0
vec<- c(f1,f2)
while(f2<=1000000){
  it<-(it+1)
  f2<-(f0+f1)
  vec<- c(vec,f2)
  f0<-f1
  f1<-f2
}
it
```

```
## [1] 30
```

```
tail(vec)
```

```
## [1] 121393 196418 317811 514229 832040 1346269
```

Podemos observar que el algoritmo requiere al menos 30 iteraciones para poder superar el millón.

## 3.Ejercicio N 3:Ordenamiento de un vector por el método burbuja

En el presente ejercicio se realiza el ordenamiento de los valores numéricos de un vector mediante el método burbuja y mediante el metodo sort nativo de R.

```
library(microbenchmark)
x<-sample(1:100,100)
mbm<-microbenchmark(
  ##método de ordenamiento directo o burbuja
  "burbuja"={
    burbuja<-function(x){
      n<-length(x)
      for(j in 1:(n-1)){
        for(i in 1:(n-j)){
          if(x[i]>x[i+1]){
```

```

        temporal<-x[i]
        x[i]<-x[i+1]
        x[i+1]<-temporal
    }
}
}
return(x)
}
res<-burbuja(x)
},
##método de R sort
"sort"={
    sort(x)
}
)
mbm

```

```

## Unit: microseconds
##      expr      min       lq      mean  median      uq      max neval
## burbuja 585.323 589.0805 859.83222 598.871 611.4400 20211.086   100
##      sort  28.269  29.8965  42.98925  32.171  37.8115   553.887   100

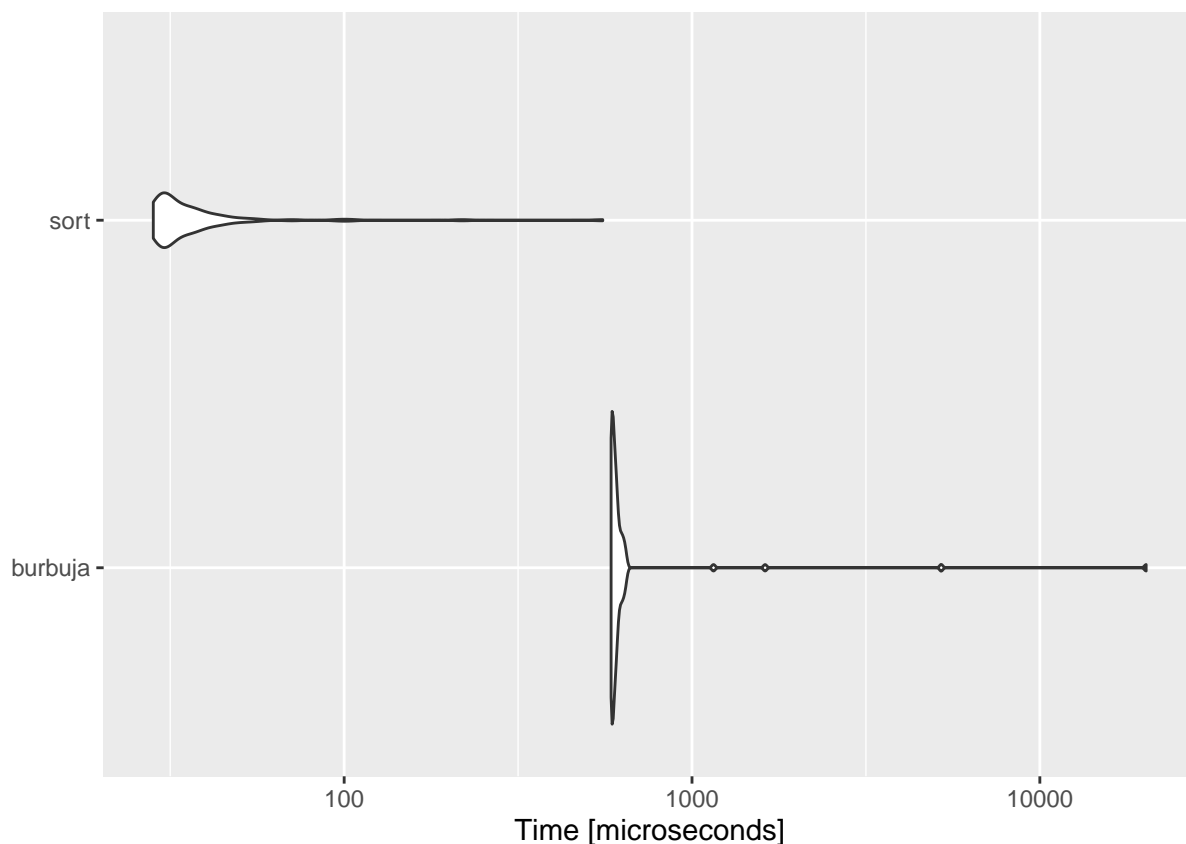
```

```

library(ggplot2)
autoplot(mbm)

```

## Coordinate system already present. Adding new coordinate system, which will replace the existing one



#### 4.Ejercicio N 4:Progresión geométrica de los casos de Covid-19.

En el presente ejercicio resolveremos mediante un modelo matemático la incógnita del virus en la pandemia, determinando así cuántos días son necesarios para que se contagien 40 millones de habitantes. Además haremos uso de los datos tomados del archivo “casos” de los contagios en Argentina.

```
library(readr)
#casos_A <- read_delim("C:casos.csv", ";", escape_double = FALSE, trim_ws = TRUE, skip = 1)
location <- getwd()
setwd(location)
casos_A <- read_delim("casos.csv", ";", escape_double = FALSE, trim_ws = TRUE, skip = 1)
```

```
## Rows: 34 Columns: 3
## -- Column specification -----
## Delimiter: ";"
## chr (1): Fecha
## dbl (1): Casos
## lgl (1): E_P+1
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
summary(casos_A$Casos)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   36.75   245.50   514.71   995.50  1715.00
```

```
m <- length(casos_A$Casos)
F <- (casos_A$Casos[2:m])/(casos_A$Casos[1:m-1])

mean(F,na.rm = TRUE)
```

```
## [1] 1.350739
```

```
sd(F,na.rm = TRUE)
```

```
## [1] 0.8554107
```

```
var(F,na.rm = TRUE)
```

```
## [1] 0.7317275
```

```
f1<- 51778
f2<-0
dia<-0
vector<- c(f1)
F<-1.62

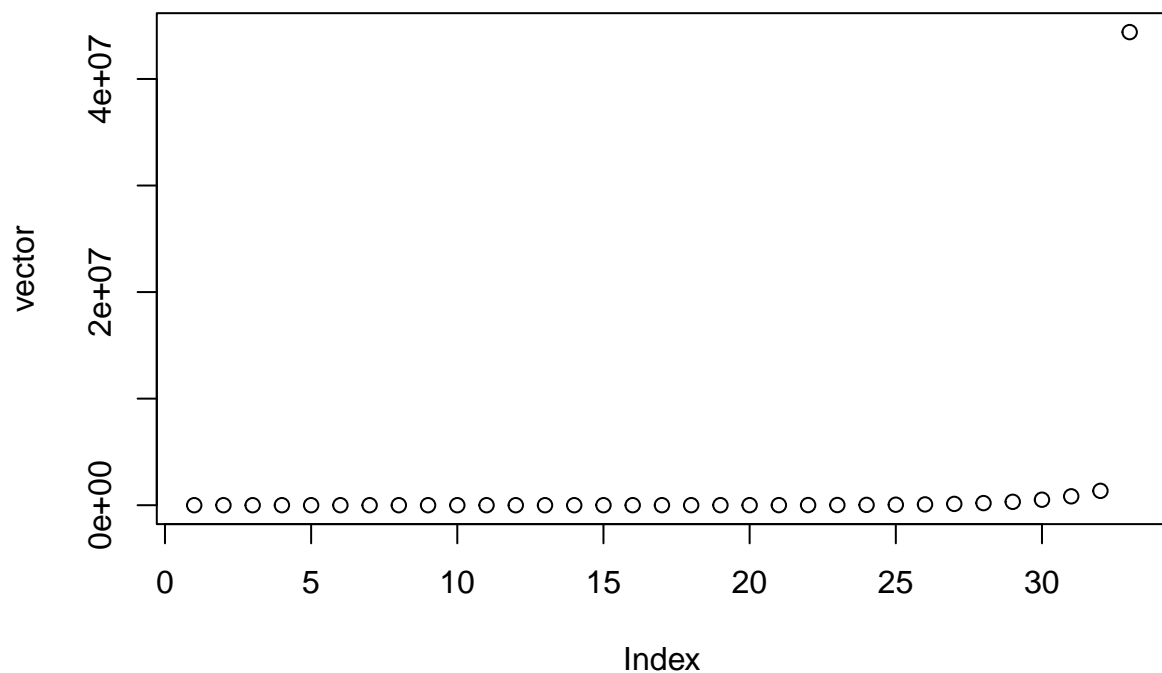
while(f2<=40000000){
  dia<- dia+1
  f2<- F*f1
  vector<- c(vec,f2)
  f1<- f2
}
dia
```

```
## [1] 14
```

```
vector
```

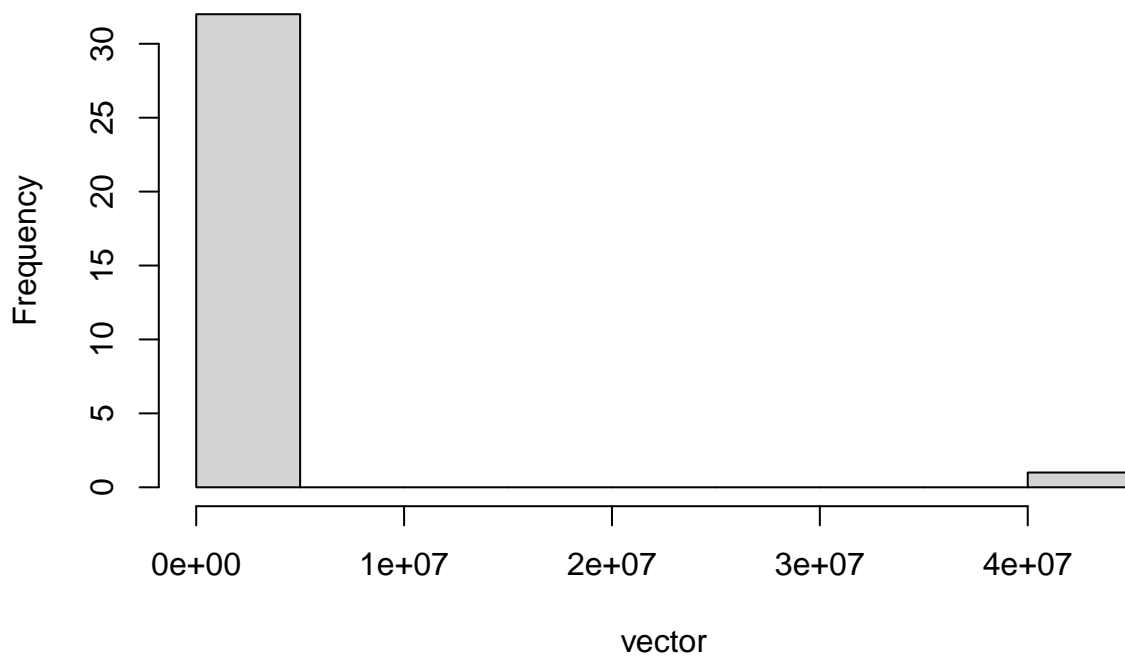
```
## [1]      1      0      1      2      3      5      8     13
## [9]     21     34     55     89    144    233    377    610
## [17]    987   1597   2584   4181   6765  10946  17711  28657
## [25]  46368  75025 121393 196418 317811 514229 832040 1346269
## [33] 44397189
```

```
plot(vector)
```



```
hist(vector)
```

**Histogram of vector**



Como se puede observar en el vector que almacenó la cantidad de casos en Argentina por día, según el modelo matemático, al día catorce recién se superará la cantidad de infectados buscada.

### Conclusiones

A través de este trabajo podemos concluir que RStudio es una gran herramienta que mezcla la programación con la necesidad de hacer informes que capture datos en forma de vectores, matrices como así también hojas

de cálculo. Se pueden realizar archivos embebidos en R, con diferentes gráficos que representen los datos con los cuales se ha trabajado. Se aprendió a formular bucles for y while, como así también la creación de vectores, matrices y funciones. Dentro de estas expresiones se puede nombrar que también se analizaron los comandos para conocer la longitud de un vector, las posiciones de los datos en este y también la manera de ordenarlos.

Otra ventaja que presenta RStudio es la cantidad de librerías que se pueden instalar. En este caso se aprendieron a usar librerías como tictoc, microbenchmark, entre otras.

Finalmente, encontramos la ventaja de que RStudio no solo se puede instalar en la máquina de cada usuario, sino que se puede utilizar a través de la nube creando una cuenta en RStudio Cloud.