

# ENTREGA MÓDULO 2

Ana Clara Videla, Aldana Sanchez, Amparo Comadrán, Sol Torrent

2022-06-15

## RESOLUCIÓN DE EJERCICIOS MÓDULO 2

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

Se procede a comparar dos maneras distintas (códigos distintos) para poder evaluar la diferencia en sus tiempos de implementación. Así, podemos ver el rendimiento y eficiencia de cada forma . Ambos códigos permiten realizar la misma tarea.

###código 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 0.08317113 secs
```

###código generado con paquete de R

```
start_time2<-Sys.time()
B<- seq(1,50000,2)
head (B)
```

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

```
tail (B)
```

```
## [1] 49989 49991 49993 49995 49997 49999
```

```
end_time2 <- Sys.time()
end_time2 - start_time2
```

```
## Time difference of 0.03699303 secs
```

## Ejercicio N°2: Implementación de la serie de Fibonacci

La serie de Fibonacci puede describirse como se muestra a continuación

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

Para el ejercicio debemos encontrar cuantas iteraciones se necesitan para generar un número en la serie que sea mayor a 1.000.000

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

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

```
tail(S)
```

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

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

Primero se crea un vector o muestra de 10 números aleatorios entre 1 y 100. Para ello podemos crear un vector aleatorio con `runif` o utilizar el comando `sample` de R.

```
library(microbenchmark)
x<-sample(1:100,10)
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 12.3 13.1 132.448  15.50 23.00 11394.9   100
##      sort 45.1 46.8  73.409  49.45 86.95   561.5   100

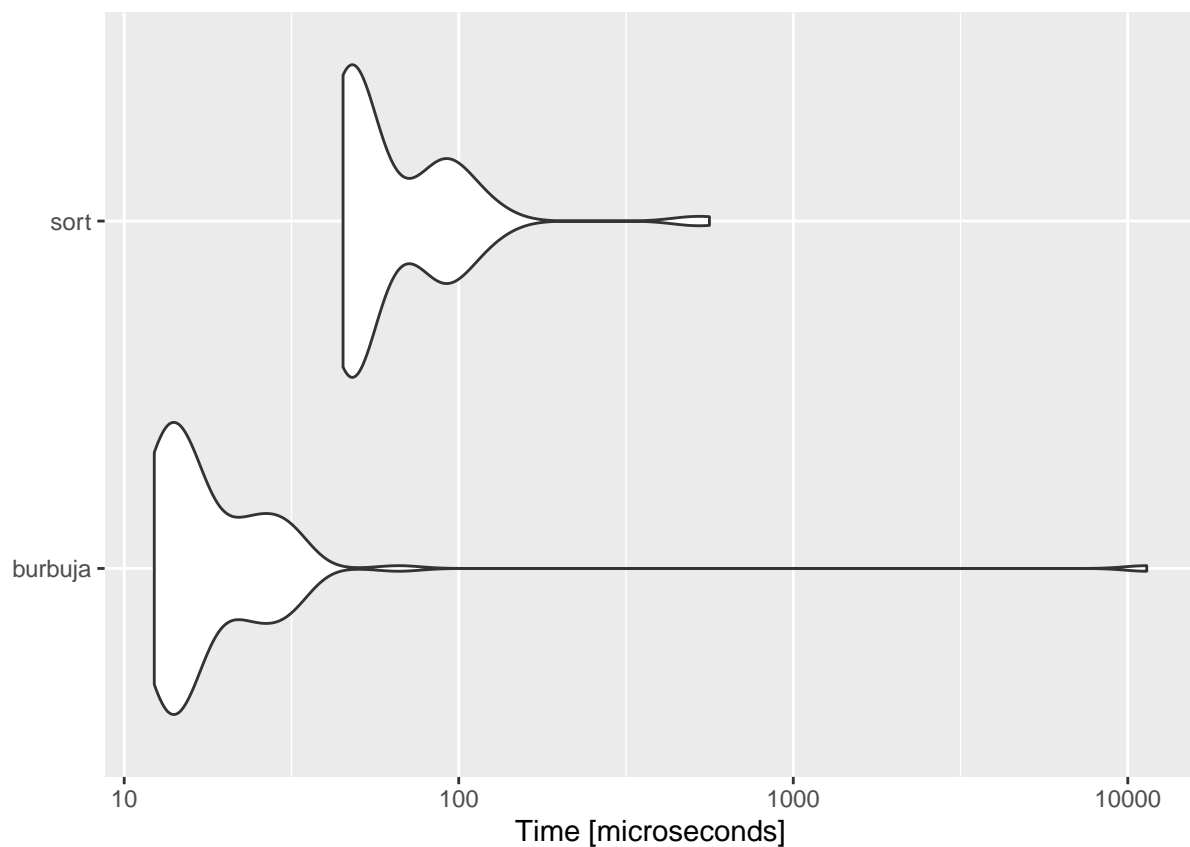
```

```

library(ggplot2)
autoplot(mbm)

```

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



## Ejercicio N°4: Progresión geométrica de COVID-19

En el siguiente ejercicio se resuelve, a partir de un modelo matemático, la incógnita del virus en la pandemia. Determinando cuántos días son necesarios para que se contagien 40 millones de habitantes. Se procede a hacer uso de los datos tomados del archivo “casos” de los contagios en Argentina brindado por la cátedra. A partir del modelo matemático, para este caso se utilizara un valor de  $F=1.62$ ; como dato de partida se toma una cantidad de 1175 casos en la fecha 04/07/2020

```
f1<- 1175
f2<-0
dia<-0
vector<- c(f1)
F<-1.62

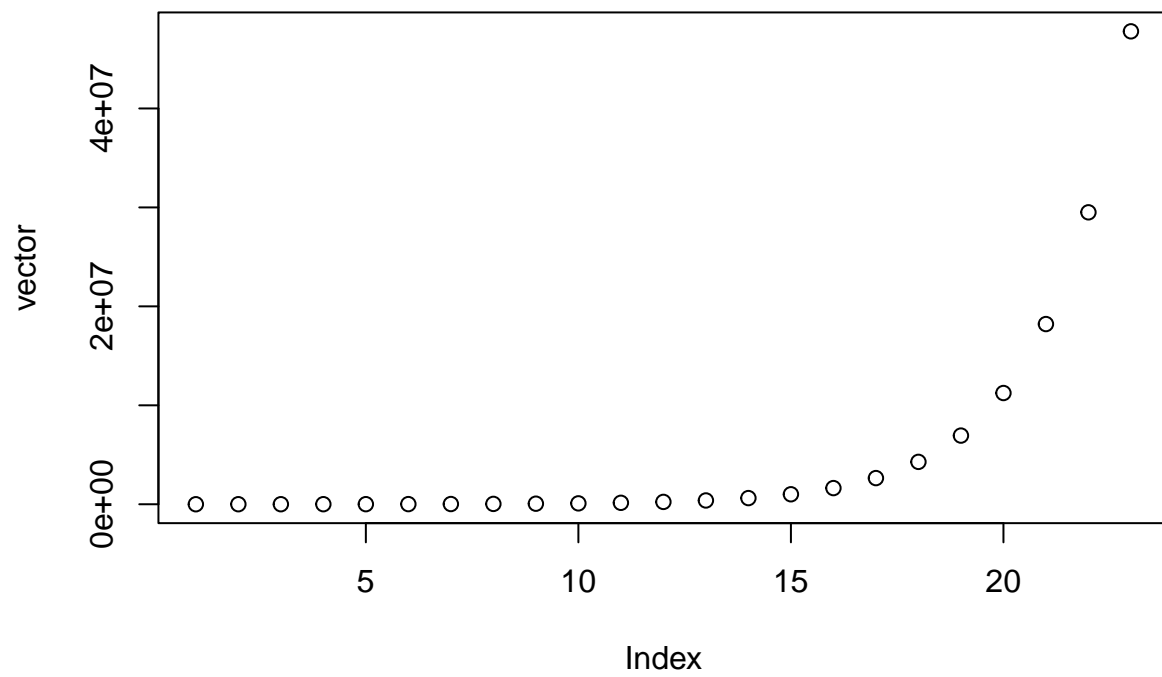
while(f2<=40000000){
  dia<-dia+1
  f2<-F*f1
  vector<-c(vector,f2)
  f1<-f2
}
dia
```

```
## [1] 22
```

```
vector
```

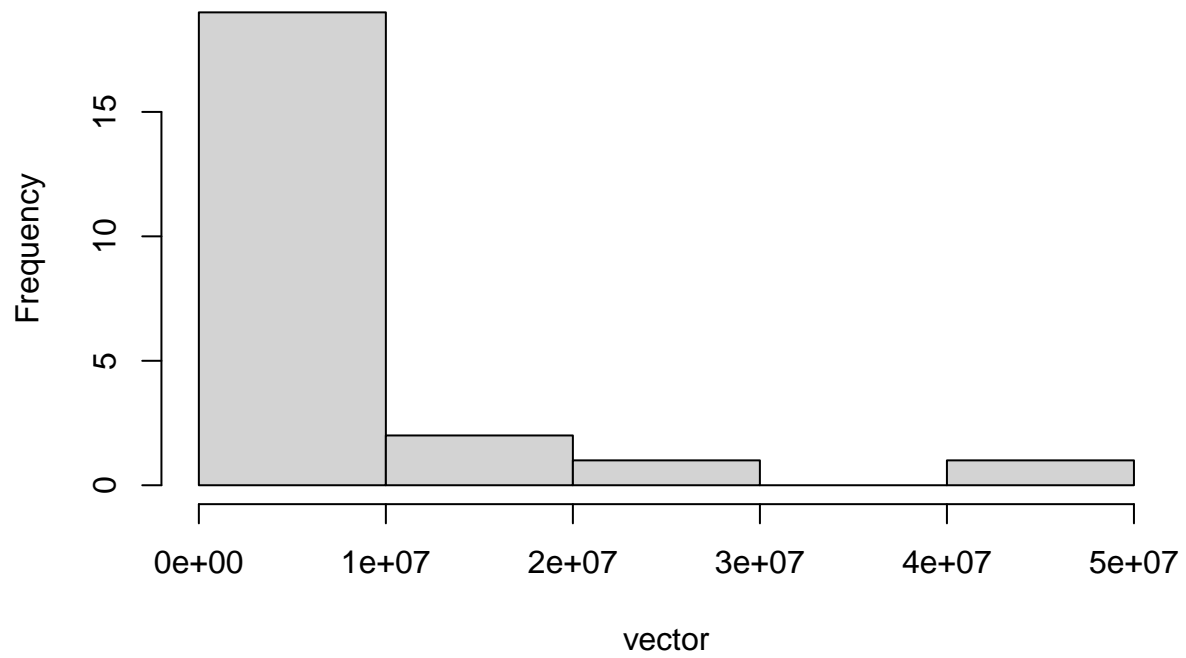
```
## [1]      1175.000      1903.500      3083.670      4995.545      8092.784
## [6]     13110.309     21238.701     34406.696     55738.847     90296.933
## [11]    146281.031    236975.270    383899.937    621917.898   1007506.995
## [16]   1632161.332   2644101.358   4283444.201  6939179.605 11241470.960
## [21] 18211182.956 29502116.388 47793428.549
```

```
plot(vector)
```



```
hist(vector)
```

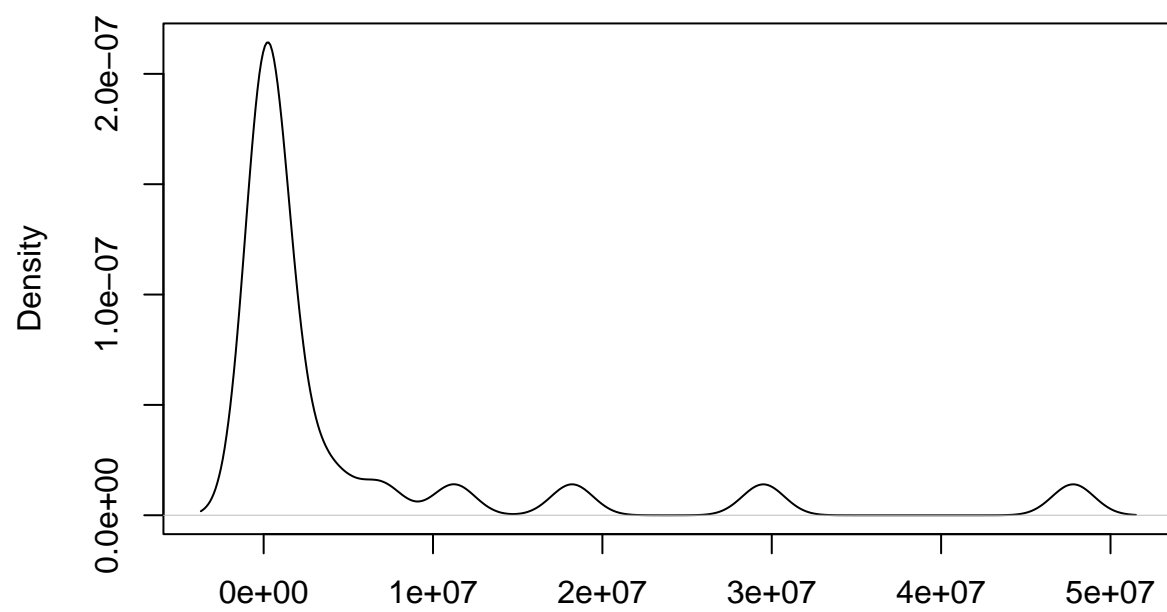
## Histogram of vector



Podemos observar que el número de contagiados superará los 40.000.000 a los 22 días, según el modelo matemático. Esto se aplica para casos en Argentina.

```
plot(density(vector))
```

**density.default(x = vector)**



N = 23 Bandwidth = 1.236e+06