

Técnicas y Herramientas Modernas I-Módulo de programación en Rstudio

Juan Manuel Pegorin^{a,1,*}, Analia Quiroga^a, Nicolas Mondaca^{a,2}, Franco Carricart^{a,2}

^aCentro universitario M5500 Mendoza

^bCentro 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 0.2951241 secs
```

##Codigo generado con R

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

*Corresponding author

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

¹Esta es la primera nota del autor.

²

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

```
tail(A)
```

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

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

```
## Time difference of 0.06663394 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 2806.8 2869.30 4237.321 3171.6 3776.95 72639.7   100
##      sort   195.3  207.05  369.245  233.7  390.70  2618.4   100

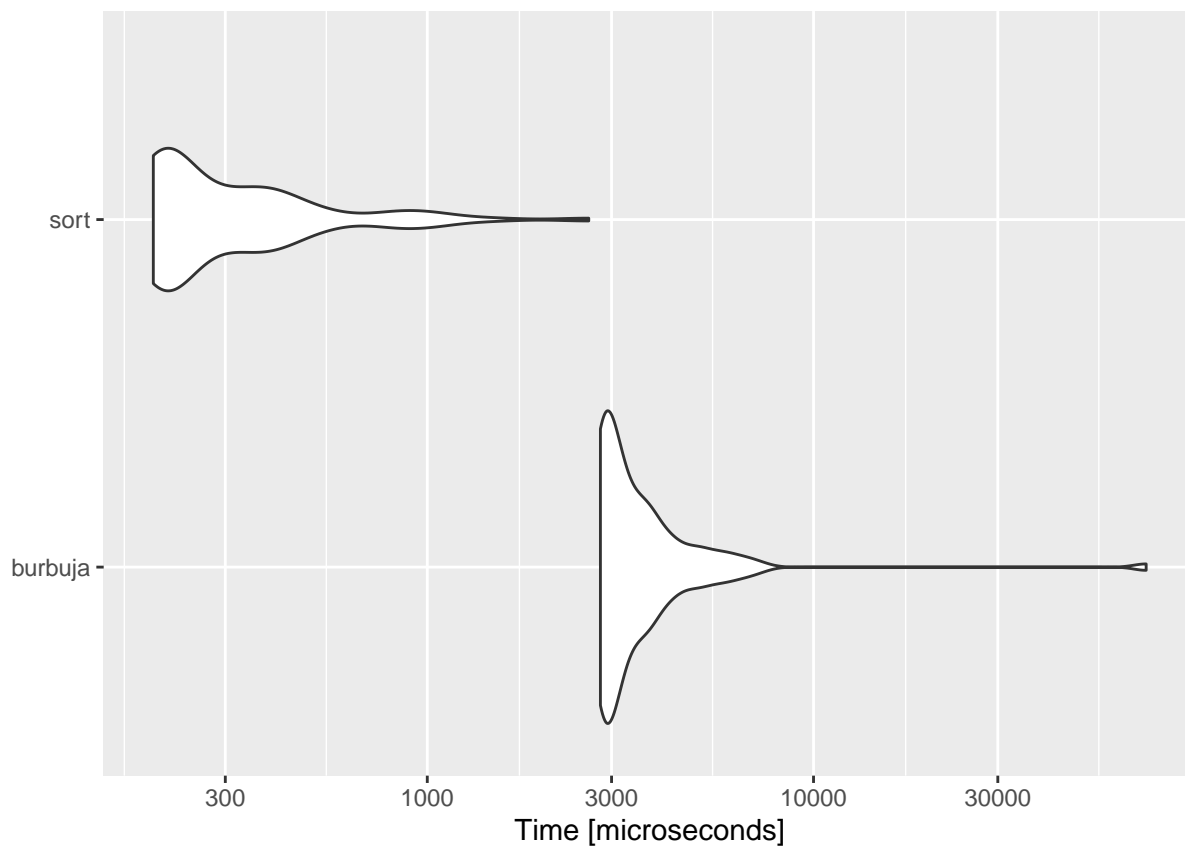
```

```

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.

```
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
```

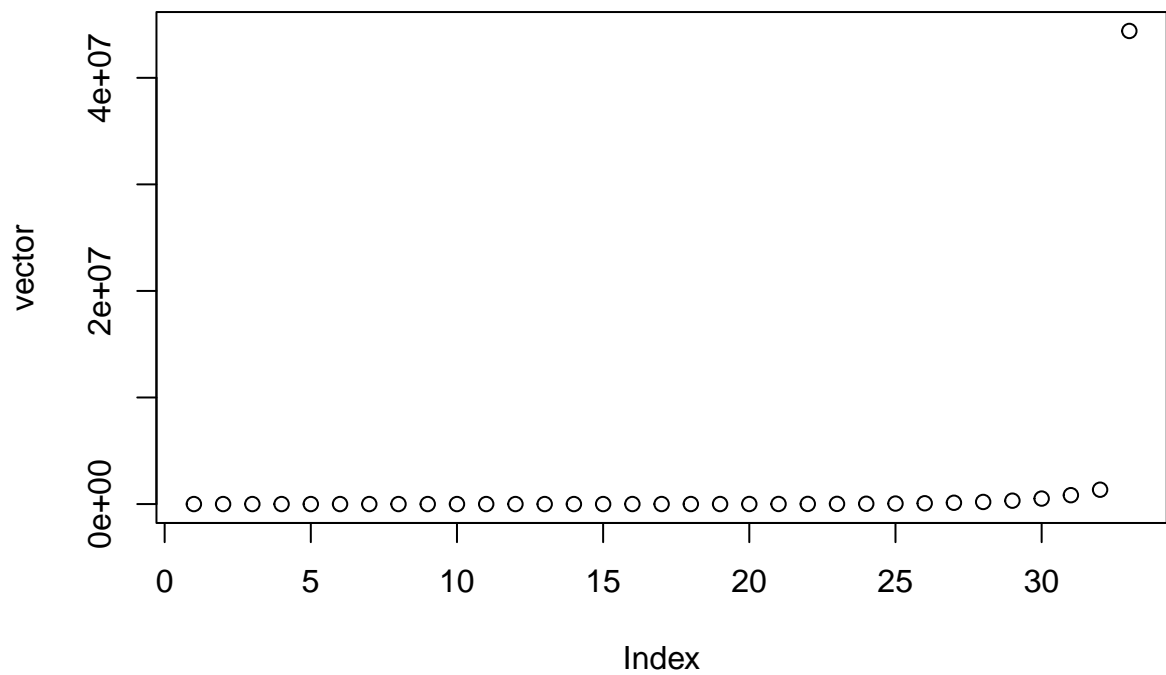
```
## [1] 14
```

```
vector
```

```
## [1]      1      0      1      2      3      5      8     13
## [9]     21     34     55     89    144    233    377    610
```

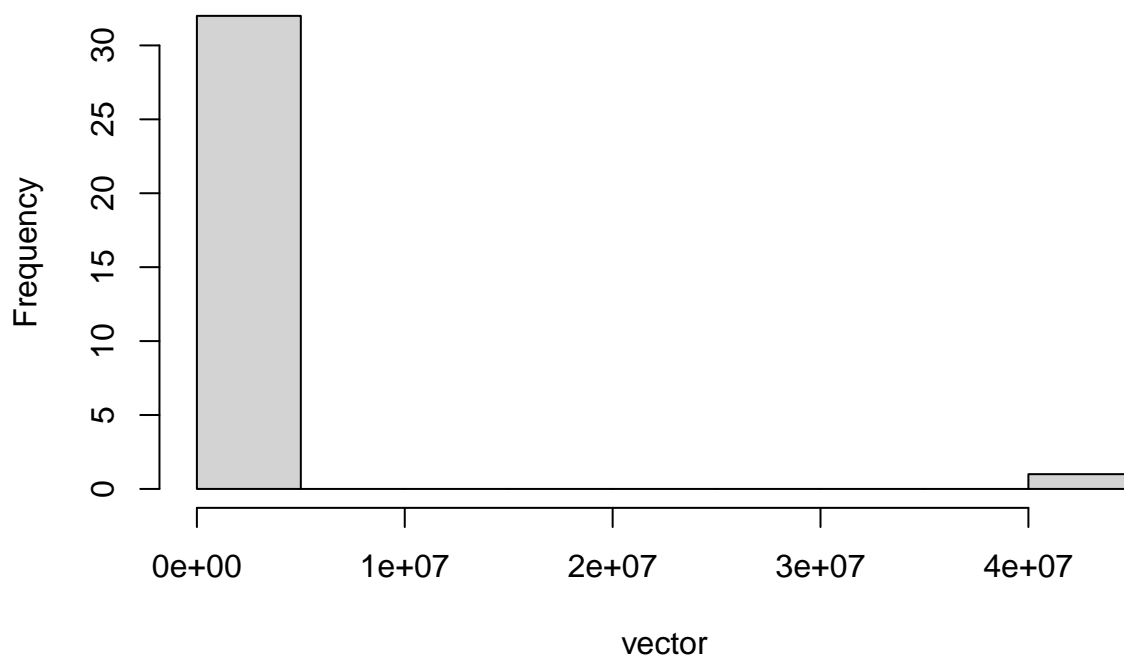
```
## [17]      987      1597      2584      4181      6765      10946      17711      28657
## [25]    46368    75025   121393   196418   317811   514229   832040  1346269
## [33] 44397189
```

```
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.