

Técnicas y Herramientas Modernas I-Módulo de programación en Rstudio

Francisco Artero^{a,1,*}, Marcos Ander Egg^a, Franco Consorte^{a,1}

^aCentro universitario M(5500) Mendoza

Abstract

El objetivo del módulo 2 fue aprender a utilizar y estudiar el lenguaje de programación R, realizando ejercitación de distintas complejidades y distintas herramientas.

Keywords: modelo matemático, iteración,

Ejercicio N1: Crear un vector secuencia.

Probamos dos códigos distintos para analizar como funcionan ambos y cuanto tiempo tardan. Su finalidad es la misma para los dos, crean un vector.

#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 0.06391287 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: f.r.artero1234@gmail.com (Francisco Artero), mf.mg.anderegg@gmail.com (Marcos Ander Egg), franco@gmail.com (Franco Consorte)

¹Primera nota.

```
tail(A)
```

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

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

```
## Time difference of 0.005836964 secs
```

Ejercicio N2: Serie de Fibonacci.

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

Observamos cuantas iteraciones se utilizan para que la serie logre un valor mayor a 1000000

```
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
```

Concluimos que el algoritmo utiliza alrededor de 30 iteraciones para lograr su objetivo. *Ejercicio N3: Ordenamiento de un vector por el método burbuja*

Realizamos el ordenamiento de los valores con ayuda del método burbuja y el método sort nativo de R.

```
library(microbenchmark)
x<-sample(1:100,100)
mbm<-microbenchmark(
  ##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)
},
##R sort
"sort"={
  sort(x)
}
)
mbm

```

```

## Unit: microseconds
##      expr   min    lq   mean median    uq   max neval
## burbuja 518.0 530.6 683.007 546.95 630.45 9022.7   100
##      sort  28.3  30.2  48.887  33.75  47.65  230.8   100

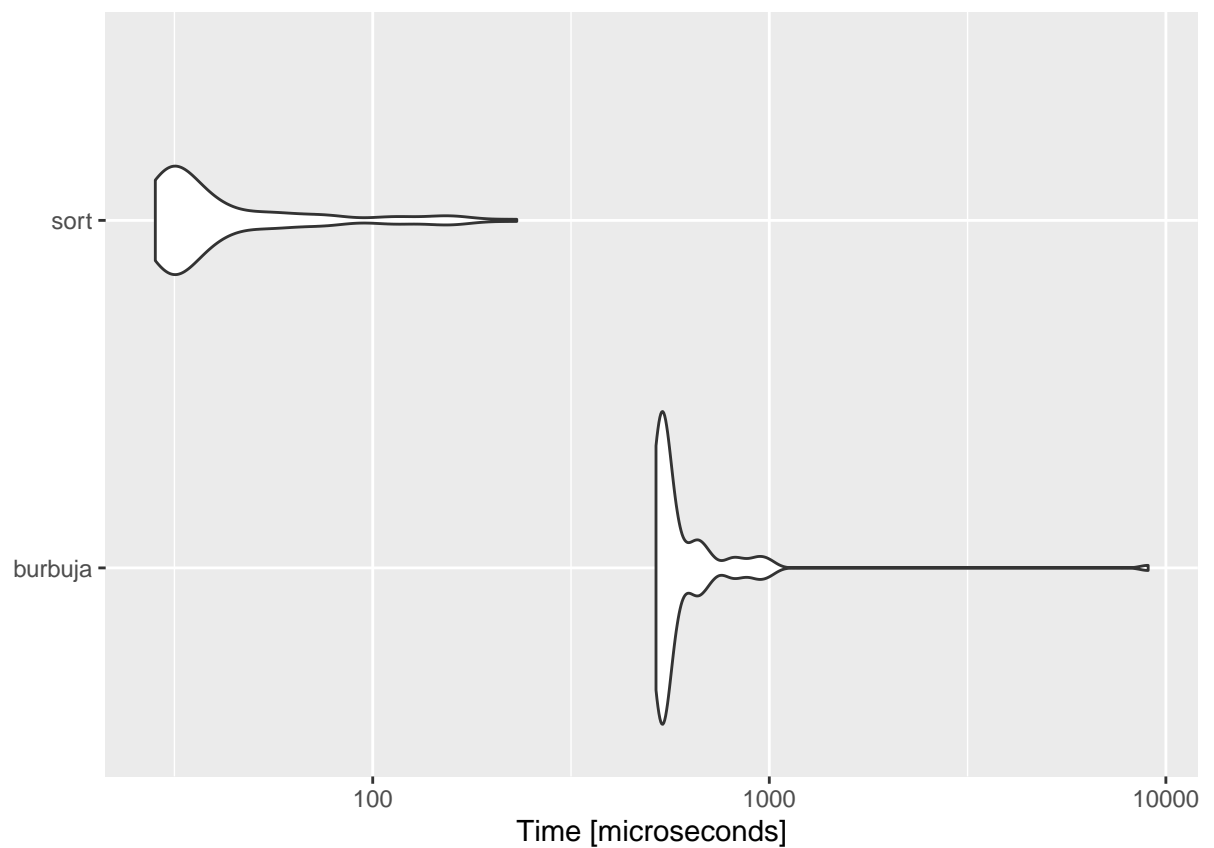
```

```

library(ggplot2)
autoplot(mbm)

```

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



Ejercicio N4: Progresión geométrica de los casos de Covid-19.

Utilizando los datos del archivo casos de los contagios de Argentina, vamos a calcular mediante un modelo, cuantos días tardaría en contagiarse 40 millones de habitantes.

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

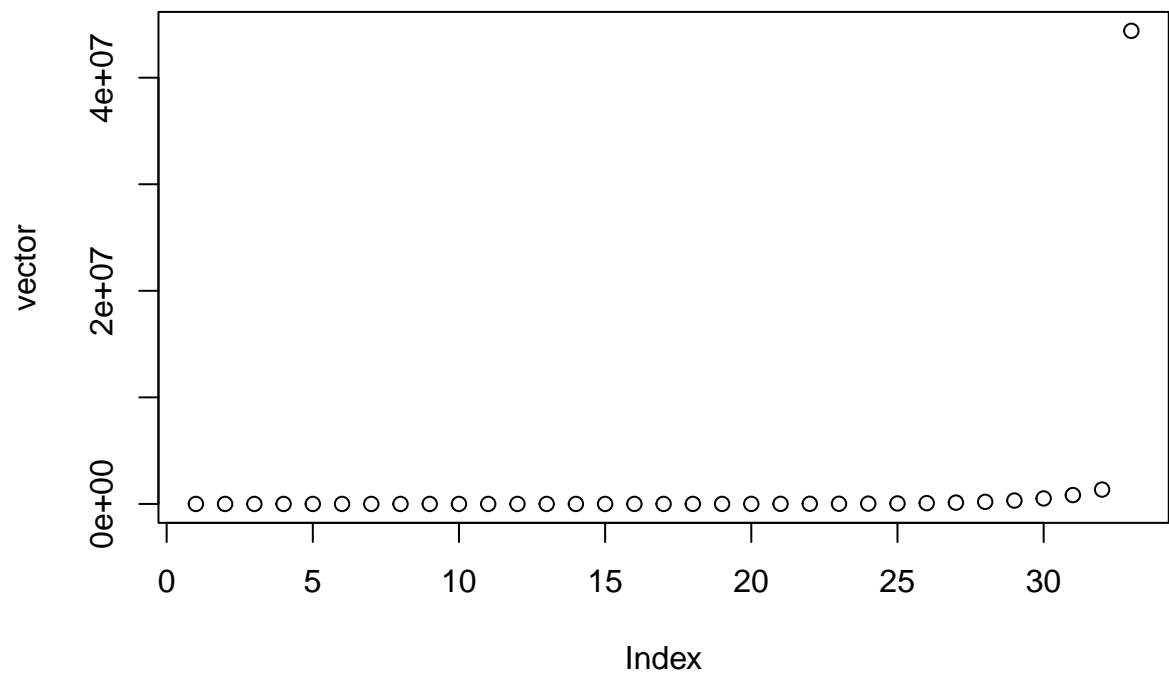
while(f2<=40000000){
  dia<-dia+1
  f2<-F*f1
  vector<-c(vec,f2)
  f1<-f2
}
dia
```

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

```
vector
```

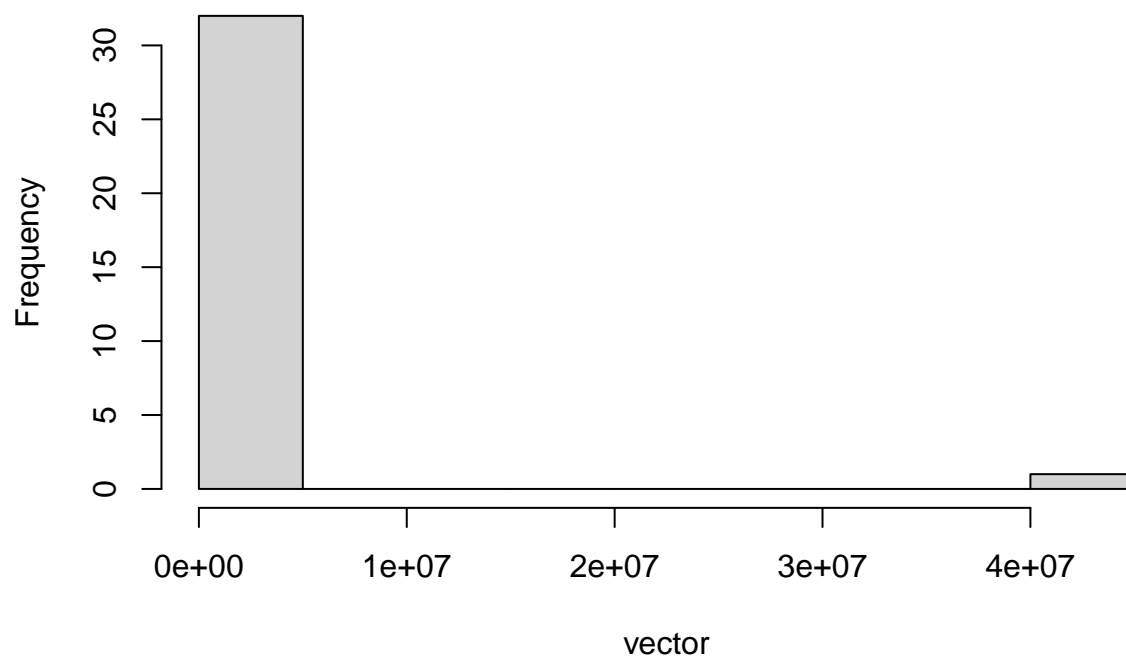
```
## [1]      1      0      1      2      3      5      8     13
## [9]     21     34     55     89    144    233    377    610
## [17]    987   1597   2584   4181   6765  10946  17711  28657
## [25]  46368  75025 121393 196418 317811 514229 832040 1346269
## [33] 44397189
```

```
plot(vector)
```



```
hist(vector)
```

Histogram of vector



El modelo nos indica que al día 14 lograremos la cantidad esperada.