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]

[a] *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 secuéncia numérica. Praa etso se usa la biblioteca de T tictoc.

##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 0.04098511 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 (Juan Manuel Pegorin), aniquiroga122000@gmail.com (Analia Quiroga), gabrielmondacanicolas@gmail.com (Nicolas Mondaca), carricartfranco9@gmail.com (Franco Carricart)

[1] Esta es la primera nota del autor.

[2]

```
tail(A)
```

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

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

```
## Time difference of 0.004107237 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.

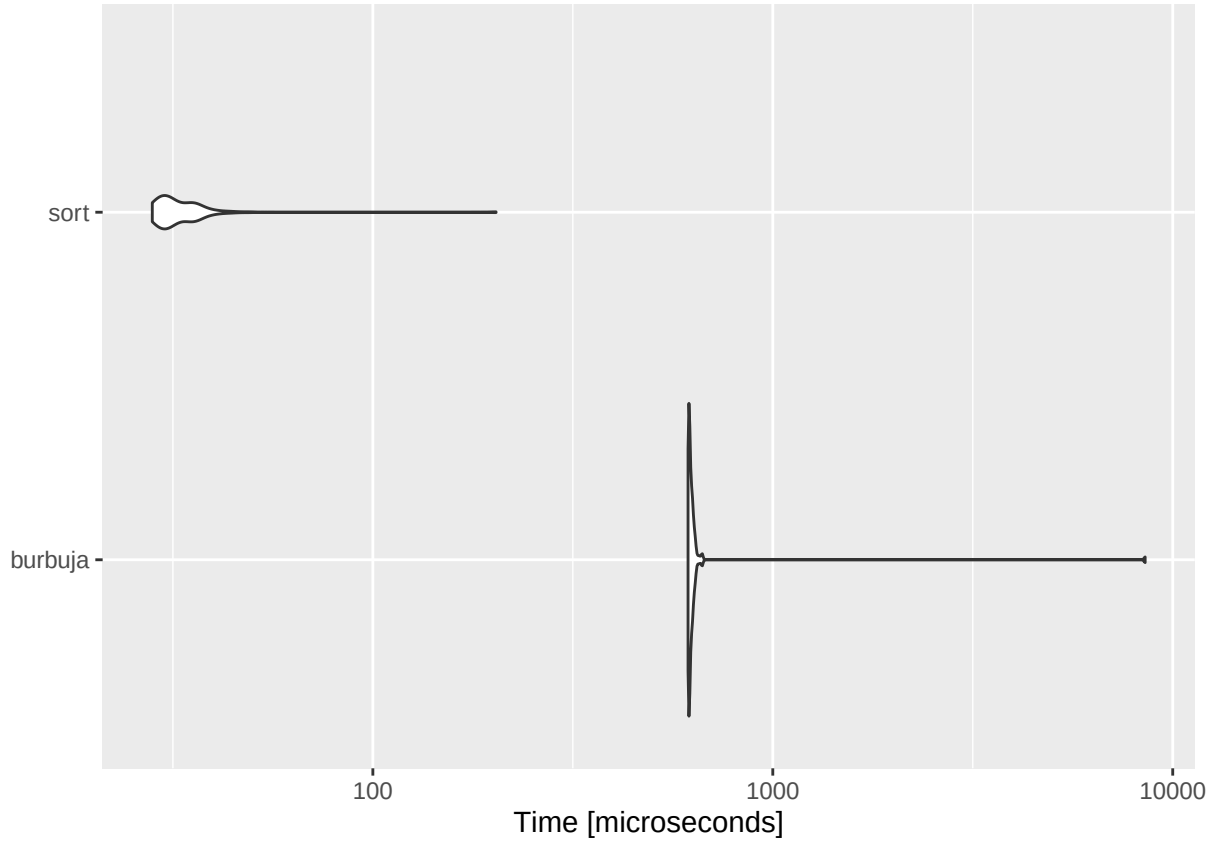
```
library(microbenchmark)
x<-sample(1:100,100)
mbm<-microbenchmark(
##método de ordenamiento directo o burbuja
"burbuja"={
  burbuja<-function(x){
    n<-length(x)
    for(j in 1:(n-1)){
      for(i in 1:(n-j)){
        if(x[i]>x[i+1]){
```

```
          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 613.919 617.0145 702.91431 619.7355 628.0100 8532.097   100
##     sort  28.088  29.8150  34.38443  31.2910  35.2595  203.276   100
```

```
library(ggplot2)
autoplot(mbm)
```

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



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 asi cuántos dias 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", delim=";", escape_double = FALSE, col_types = cols(...2 = col_numbe
```

```
## New names:
## * `` -> `...2`
## * `` -> `...3`

## Warning: One or more parsing issues, see `problems()` for details
```

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

```
## [1] 0
```

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

```
## [1] NA
```

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

```
## [1] NA
```

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

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

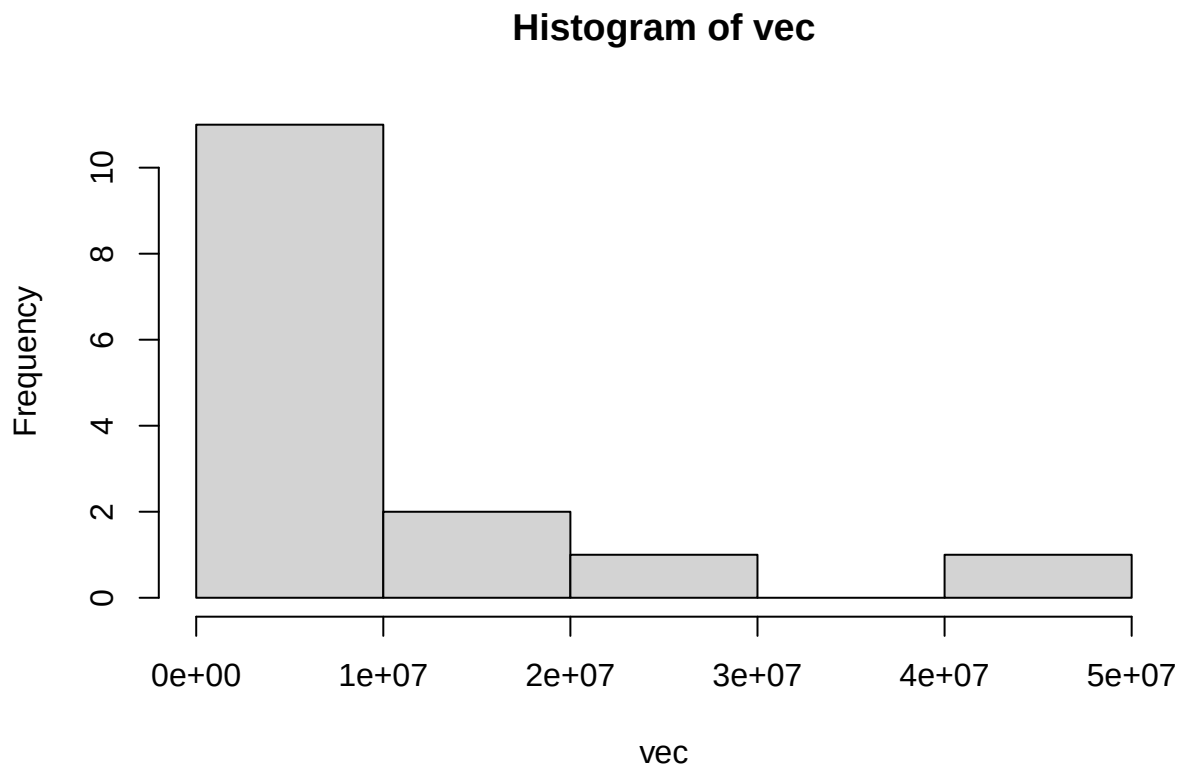
```r
vec
```

```
##  [1]    51778.00    83880.36   135886.18   220135.62   356619.70   577723.91
##  [7]   935912.74  1516178.64  2456209.39  3979059.21  6446075.93 10442643.00
## [13] 16917081.66 27405672.29 44397189.11
```

```r
dia
```

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

```r
hist(vec)
```



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.

En el proximo ejemplo se trabaja con el archivo "casos.csv" para ver la progresion de los casos del Covid, como tambien el histograma y al función de densidad. Para esto se importa el archivo "casos.csv" en la carpeta donde se esta trabajando para poder obteenr los datos necesarios.

```
library(readr)
casos_A<-read_delim("casos.csv", delim =";", escape_double=FALSE, col_types = cols('Covid en Argentina
```

```
## New names:
## * `` -> `...2`
## * `` -> `...3`
```

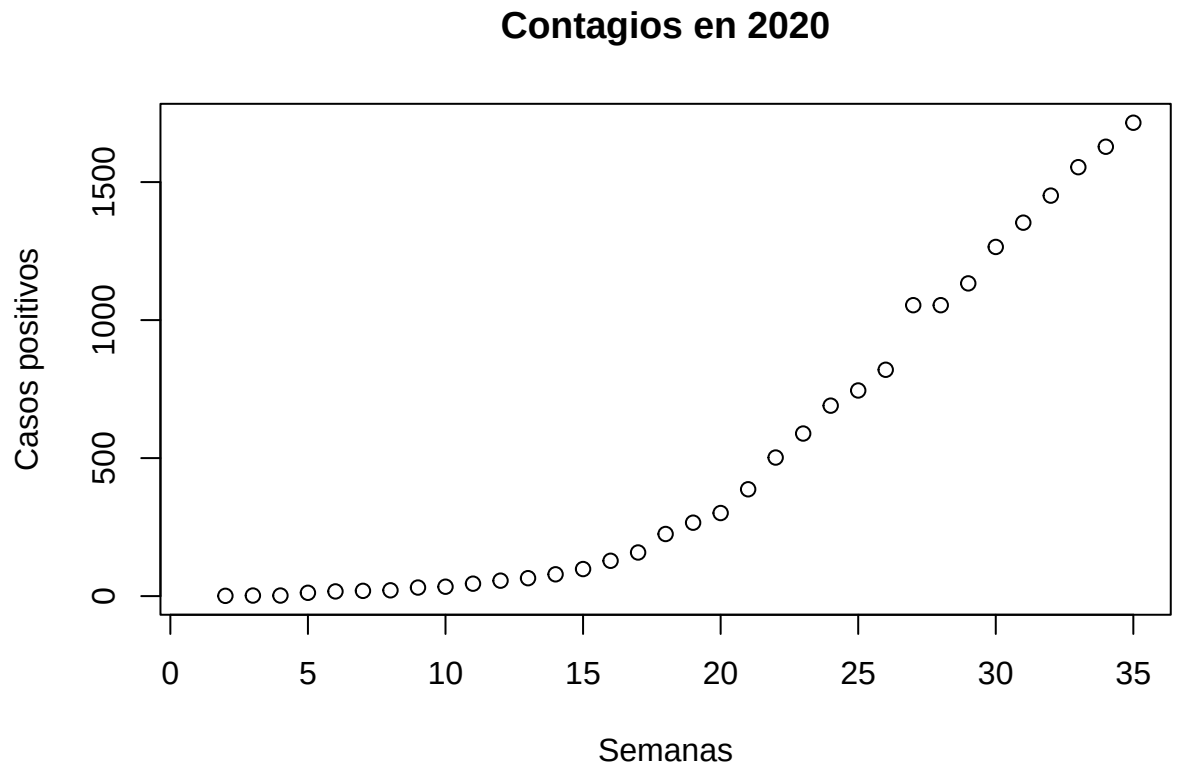
```
## Warning: The following named parsers don't match the column names: Covid en
## Argentina
```

```
casos_A$...2
```

```
##  [1] "Casos" "1"     "2"     "2"     "12"    "17"    "19"    "21"    "31"
## [10] "34"    "45"    "56"    "65"    "79"    "98"    "128"   "158"   "225"
## [19] "266"   "301"   "387"   "502"   "589"   "690"   "745"   "820"   "1054"
## [28] "1054"  "1133"  "1265"  "1353"  "1451"  "1554"  "1628"  "1715"
```

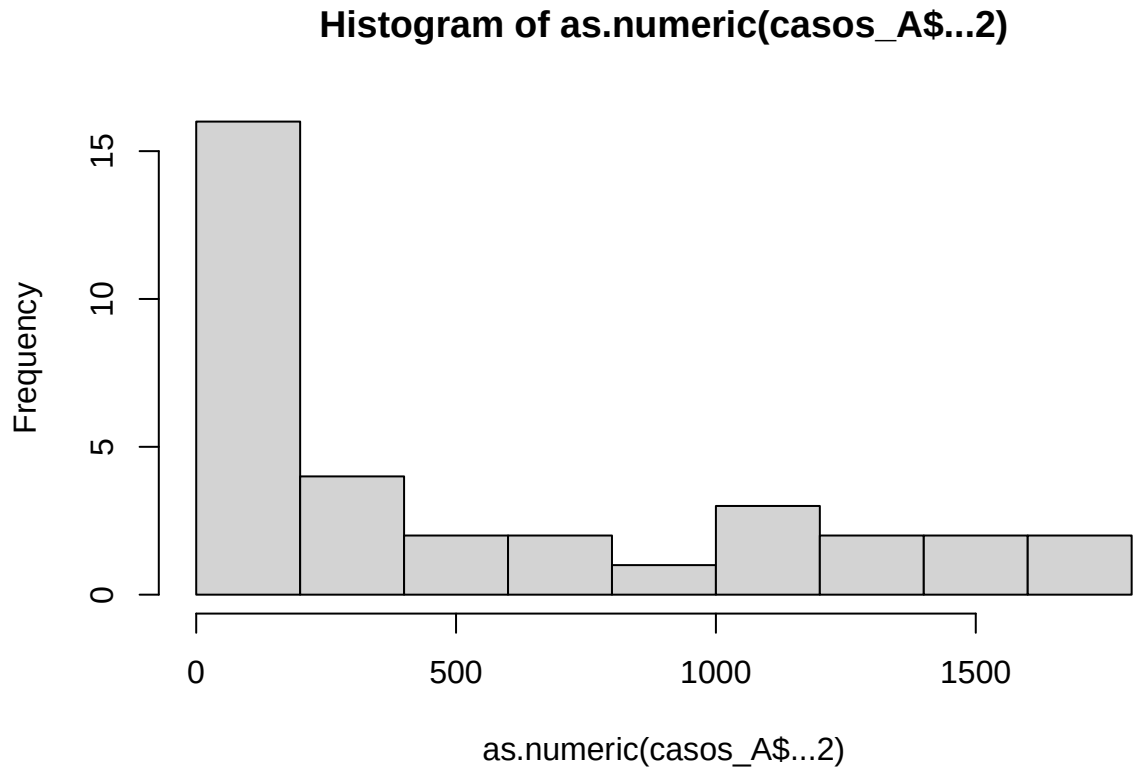
```
plot(casos_A$...2, main="Contagios en 2020", ylab="Casos positivos", xlab="Semanas")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```



```
hist(as.numeric(casos_A$...2))
```

```
## Warning in hist(as.numeric(casos_A$...2)): NAs introduced by coercion
```

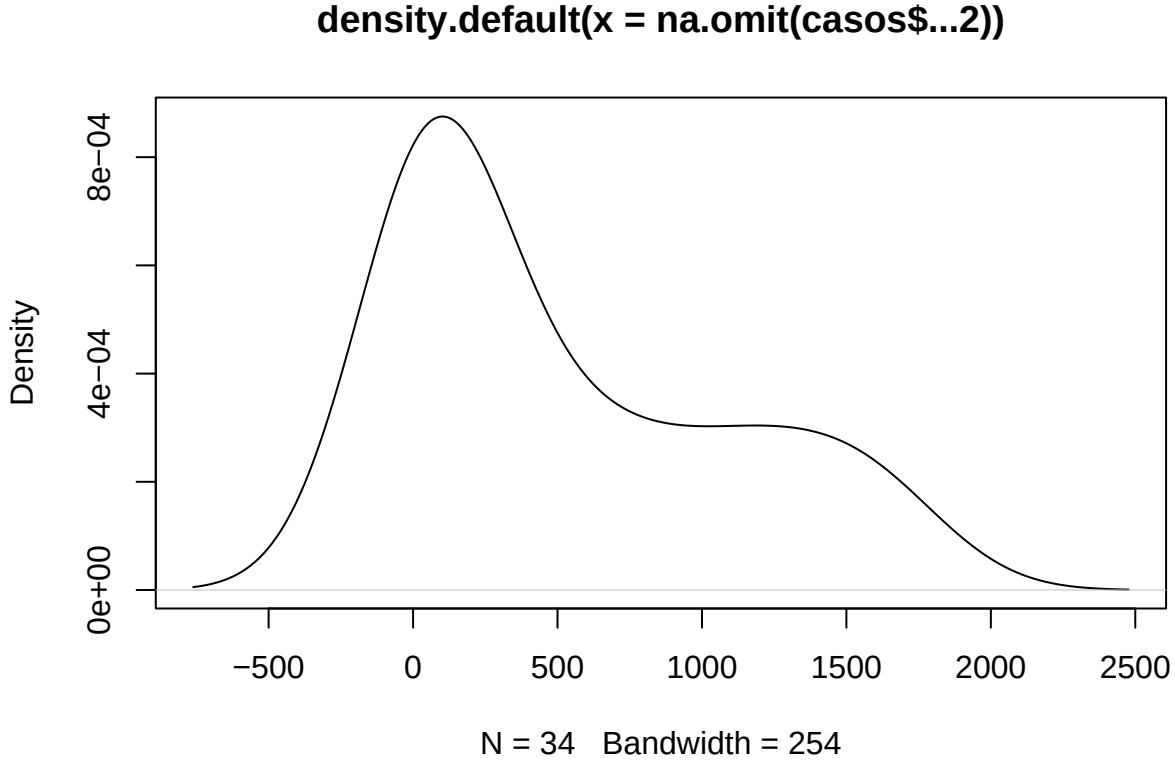



```
casos<- read_delim("casos.csv", delim =";", escape_double = FALSE, col_types = cols(...2 = col_number(
```

```
## New names:
## * `` -> `...2`
## * `` -> `...3`
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
plot(density(na.omit(casos$...2)))
```



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 trabajo. Se aprendió a formular bucles for y while, como así tambien la creación de vectores, matrices y funciones. Dentro de estas expresiones se puede nombrara que también se analizaron los comandos para conocer la longitud d eun 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 librerias como tictoc, microbenchmark, entre otras.

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