Trabajo En Clase TyHM I

Franco Santilli[a,1,*], Guillermo Zaina[a], Marcos Santiago[a,1], Renzo Tagarot[a,1],
Fernando Rodriguez[a,1]

[a]*Paseo Dr.~Emilio Descotte 750 M5500 Mendoza*

Abstract

Introducción

En este archivo se muestra lo trabajado en clase como modo de introducción
a los conocimientos de R que deberemos aprender para poder realizar la entrega
final, fragmentos de códigos y que hacen y como funcionan

Fragmentos de Códigos

```r
A<-0
B<-1
F[1]<-A
F[2]<-B
for (i in 3:100) { F[i] <- (F[i-1]+F[i-2]) }
head (F)
```

```
## [1] 0 1 1 2 3 5
```

```r
x<-rnorm(100,50,25)
burbuja <- function(x){
  n<-length(x)
  for (j in 1:(n-1)) {
    for (i in 1:(n-j)) {
      if (x[i]>x[i+1]) {
        temp<-x[i]
        x[i]<-x[i+1]
        x[i+1]<-temp
      }
```

*Corresponding author
Email addresses:* `francosantilli47@gmail.com` (Franco Santilli),
`zainaguillermo@gmail.com` (Guillermo Zaina), `marcossantiagoh2@gmail.com` (Marcos
Santiago), `renzot0800@gmail.com` (Renzo Tagarot), `ferodriguez15@gmail.com` (Fernando
Rodriguez)

```
    }
  }
  return(x)
}
res<-burbuja(x)
res
```

```
##   [1]  -0.9955975   3.7762888   6.7675013  11.1163895  12.3716844  13.1919961
##   [7]  13.3235577  13.9882835  14.9464894  17.0024155  17.2823846  19.8319228
##  [13]  20.7407452  26.5450244  26.6651673  26.8584679  26.9035641  29.2543837
##  [19]  29.5005094  30.3653108  30.5200233  31.6349098  32.6137874  33.5095589
##  [25]  33.5540416  33.7823651  34.0849818  34.7108244  34.8191494  36.5633122
##  [31]  37.3269203  37.7295340  37.8636457  38.8057012  40.6706716  43.3481358
##  [37]  43.4034925  43.8193740  43.8973269  45.5642987  45.8492078  47.9877809
##  [43]  48.0796002  48.9344218  49.3184416  49.8699272  51.1979948  51.4383106
##  [49]  51.6791279  51.9140593  52.2237647  53.0105818  53.0483727  53.2638780
##  [55]  54.0587214  54.2859192  55.1423438  55.8238406  56.4796571  58.2861261
##  [61]  58.6620270  58.9426201  59.0187627  59.7637534  60.7078165  61.0700320
##  [67]  61.7135935  62.8429723  63.6349436  64.9877460  68.8069497  69.2966189
##  [73]  71.0958972  72.0087430  72.3313963  72.7570827  73.2754944  73.3878099
##  [79]  74.2040067  74.9742332  75.2200967  75.4997884  75.8548063  75.9915039
##  [85]  76.4902495  77.3575143  78.7751203  79.0690415  79.1522729  82.3537593
##  [91]  83.2717399  86.6115419  88.3162479  91.4907457  94.1650412  97.1811589
##  [97]  98.0851080 102.6618328 107.4188643 114.9425602
```

```
t0<-Sys.time()
x<-rnorm(100,50,25)
burbuja <- function(x){
  n<-length(x)
  for (j in 1:(n-1)) {
    for (i in 1:(n-j)) {
      if (x[i]>x[i+1]) {
        temp<-x[i]
        x[i]<-x[i+1]
        x[i+1]<-temp
      }
    }
  }
  return(x)
}
res<-burbuja(x)
res
```

```
##   [1]  -2.526493   2.220028   7.546517  14.127133  17.284483  20.599031
##   [7]  22.521082  23.081176  23.261928  24.320832  25.041102  26.293095
```

```
## [13]   31.044062   31.574734   31.944678   32.394472   32.603816   34.013013
## [19]   34.617305   34.677244   35.338116   35.353994   36.692042   37.088201
## [25]   37.346985   38.717020   39.848896   39.935346   40.290678   42.143486
## [31]   42.654941   42.703559   43.582756   43.587337   44.069049   45.075642
## [37]   45.113091   45.377827   45.694195   45.804847   46.039453   47.849003
## [43]   48.468012   50.117626   50.177270   50.732179   51.184511   51.250953
## [49]   51.486522   52.434571   53.398487   54.064371   55.646572   55.765504
## [55]   55.841154   56.250822   57.713666   58.205436   58.206399   58.583686
## [61]   59.406904   59.491970   61.205835   61.556379   61.856931   61.996656
## [67]   63.033385   63.687266   65.155896   66.645569   69.312501   70.069183
## [73]   70.617032   70.683327   70.845981   71.849792   72.043971   72.704888
## [79]   74.342622   75.344856   75.651586   76.052872   76.395079   77.223190
## [85]   78.177295   79.139505   80.280839   82.949626   83.253269   85.567116
## [91]   85.721124   87.087369   87.498939   87.700550   88.584241   90.569905
## [97]   92.306352   95.807388   99.993211  102.098806
```

```r
tf<-Sys.time()
#ahora medimos la velocidad del algoritmo
tf-t0
```

```
## Time difference of 0.1551981 secs
```

```r
library(tictoc)
tic()
x<-rnorm(100,50,25)
burbuja <- function(x){
  n<-length(x)
  for (j in 1:(n-1)) {
    for (i in 1:(n-j)) {
      if (x[i]>x[i+1]) {
        temp<-x[i]
        x[i]<-x[i+1]
        x[i+1]<-temp
      }
    }
  }
  return(x)
}
res<-burbuja(x)
res
```

```
## [1]  -7.435248  -2.481888  -1.218721   1.632252   6.426259   7.104327
## [7]   8.609149   9.144819  13.055219  14.771200  15.494357  16.014596
## [13]  16.266847  17.379334  17.862300  18.773446  19.075061  19.380523
## [19]  19.620055  24.498302  25.565413  26.832655  27.026582  29.225409
## [25]  30.703107  30.751534  32.587487  33.663776  33.759393  36.127879
```

3

```
##  [31]  37.708992  37.920027  38.656652  39.484726  40.036221  40.581770
##  [37]  41.000604  41.466927  42.050987  43.281008  43.708797  43.879256
##  [43]  43.937458  43.967858  44.168787  44.816738  44.888452  44.979318
##  [49]  46.723360  47.161659  47.273288  47.578258  48.893119  49.290713
##  [55]  49.305098  50.387461  51.104896  51.863854  51.990642  51.994089
##  [61]  53.180367  54.151726  54.971656  56.214497  57.780036  58.808878
##  [67]  59.427194  59.982873  61.974604  63.256512  65.687290  67.701015
##  [73]  67.815912  67.919376  69.014277  69.837554  70.495729  70.649720
##  [79]  73.487408  73.621006  73.753217  76.880168  77.038830  81.483575
##  [85]  82.917764  85.277221  86.488691  86.765721  87.485280  89.193139
##  [91]  89.687764  90.479060  91.090927  94.670840  95.881095  96.763501
##  [97] 106.443753 111.534408 112.251570 115.203556
```

```r
toc()
```

```
## 0.05 sec elapsed
```

```r
library(microbenchmark)
x<-rnorm(100,50,25)
mbm<-microbenchmark(
  "burbuja"={
    x<-rnorm(100,50,25)
burbuja <- function(x){
  n<-length(x)
  for (j in 1:(n-1)) {
    for (i in 1:(n-j)) {
      if (x[i]>x[i+1]) {
        temp<-x[i]
        x[i]<-x[i+1]
        x[i+1]<-temp
      }
    }
  }
  return(x)
}
res<-burbuja(x)
res
  },
"sort"={
  sort(x)
}
)

mbm
```
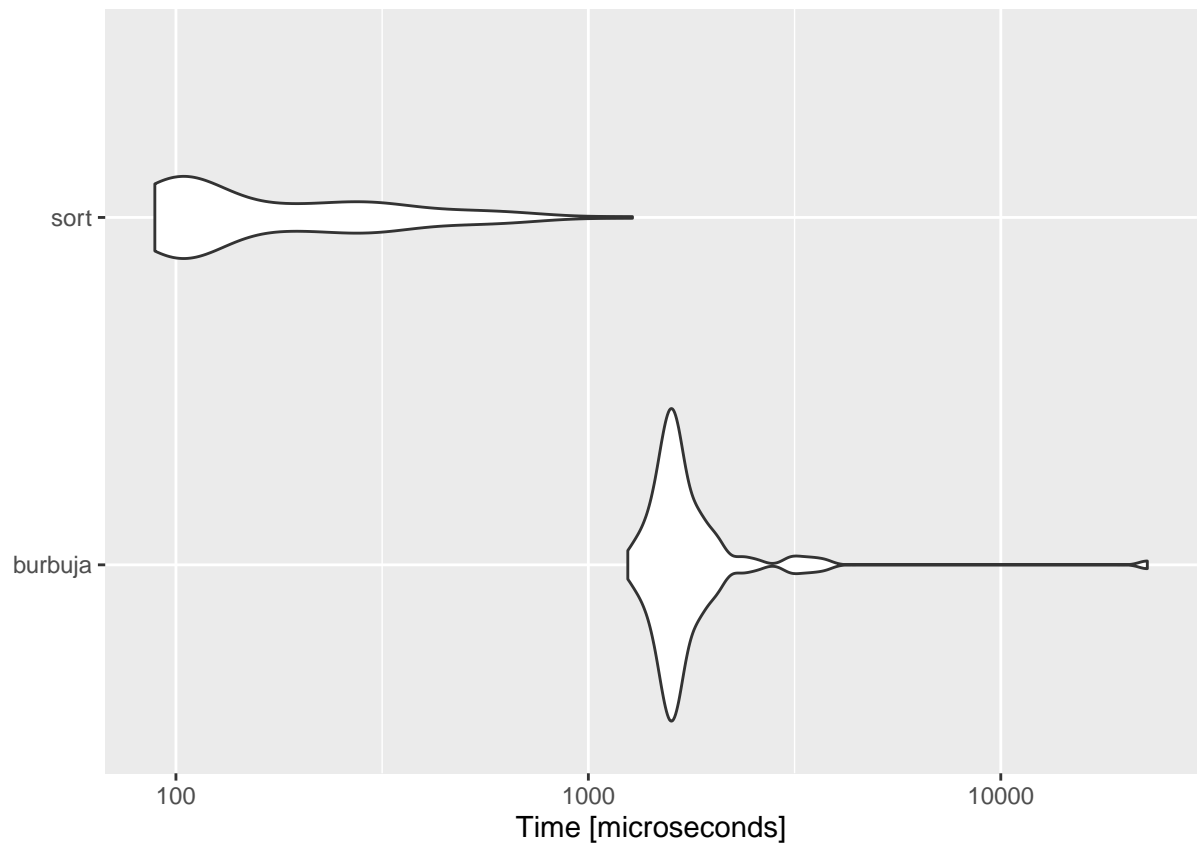
```
## Unit: microseconds
```

4

```
##     expr   min      lq     mean median      uq     max neval
## burbuja 1246.4 1524.25 1950.711 1623.45 1812.45 22661.8   100
##     sort   88.9  101.35  211.617  118.75  280.60  1279.3   100
```

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

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

creación de vectores

```
v1 <- c(1,2,3,4,5)
```

creación de un vector de 9 componentes

```r
v2 <- c(1,2,3,4,5,6,7,8,9)
```

creación de matrices

```r
m1<- matrix(v2,ncol=3,byrow=FALSE)
```

el byrow me deja ordenar los valores por fila en el caso de TRUE, o en columna en el caso de FALSE

averiguar que clase de ojeto hemos creado

para saber de qué clase se utiliza el comando class(nombre del objeto) vemos que nos dice que el vector es de tipo numérico y la matriz de tipo array o matriz, además siempre es numerica la matriz.

```r
class(v1)
```

```
## [1] "numeric"
```

```r
class(m1)
```

```
## [1] "matrix" "array"
```

creación de un vector de palabras

```r
v3<- c("a","b","c")
class(v3)
```

```
## [1] "character"
```

```r
v3
```

```
## [1] "a" "b" "c"
```

hay un comando que se llama dimnames que sirve para ponerle nombre a las filas y las columnas.

importar datos de la red o de excel

lo que nos dice al pegar un dato de excel es que enumera todos los cambios
que tuvimos que hacer en el archivo original para dejarlo ordenado y acomodado.

```
library(readr)
casos <- read_delim("C:/Users/franc/Downloads/casos.csv",
    delim = ";", escape_double = FALSE, col_types = cols(`Covid Argentina` = col_date(format
    trim_ws = TRUE)
```

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

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

dataset es un conjunto de datos de una tabla tomados de la vida real que
estan documentados y estan guardados en repositorios de datos.

ploteo de datos

```
casos$...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$...2,main="Contagios 2020",ylab="Semana",xlab="Casos Positivos")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introducidos por coerción
```

Contagios 2020