Tecnicas y herramientas modernas

Grupo Los Ritmocerontes

2025-04-30

Generacion de un vector secuencia

```
A <- 0
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
```

```
R <- seq(from = 1, to = 100000, by = 2)
head(R)
```

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

```
tail(R)
```

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

Metodo burbuja

```
# Tomo una muestra de 10 números ente 1 y 100
x<-sample(1:100,10)
# Creo una funcion para ordenar
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)
#Muestra obtenida
x
```

```
##  [1] 53 97 75 25 80 39 21 56 30  7
```

```r
#Muestra Ordenada
res
```

```
##  [1]  7 21 25 30 39 53 56 75 80 97
```

```r
#Ordanacion con el coamando SORT de R-Cran
sort(x)
```

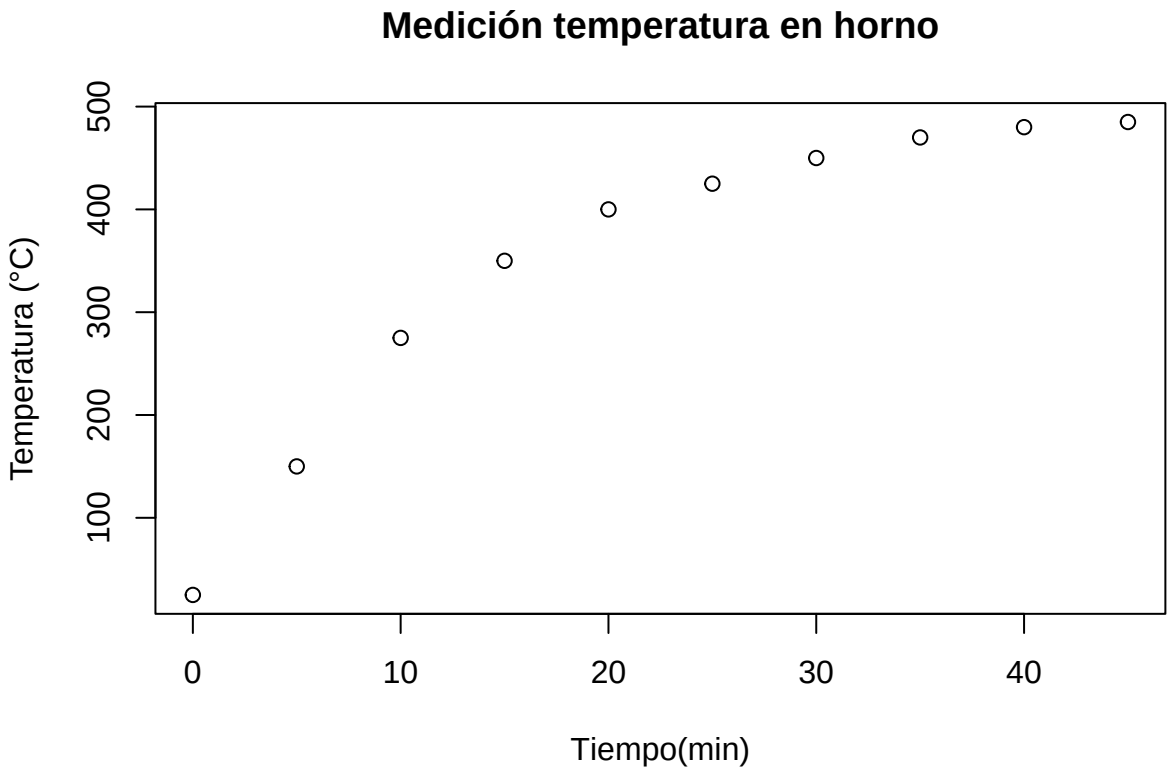
```
##  [1]  7 21 25 30 39 53 56 75 80 97
```

Medicion temperatura horno

```r
library(readxl)
medicion_temperatura_horno <- read_excel("medicion_temperatura_horno.xlsx",
    col_types = c("numeric", "numeric", "numeric",
        "numeric"))
medicion_temperatura_horno
```

```
## # A tibble: 10 x 4
##    `Tiempo (min)` `Temp Sensor 1 (°C)` `Temp Sensor 2 (°C)` `Temp Sensor 3 (°C)`
##             <dbl>                <dbl>                <dbl>                <dbl>
##  1              0                   25                   24                   25
##  2              5                  150                  148                  149
##  3             10                  275                  273                  276
##  4             15                  350                  348                  349
##  5             20                  400                  397                  398
##  6             25                  425                  423                  424
##  7             30                  450                  448                  449
##  8             35                  470                  468                  469
##  9             40                  480                  478                  479
## 10             45                  485                  484                  485
```

```r
plot(medicion_temperatura_horno$`Tiempo (min)`,medicion_temperatura_horno$`Temp Sensor 1 (°C)`,main = "
```

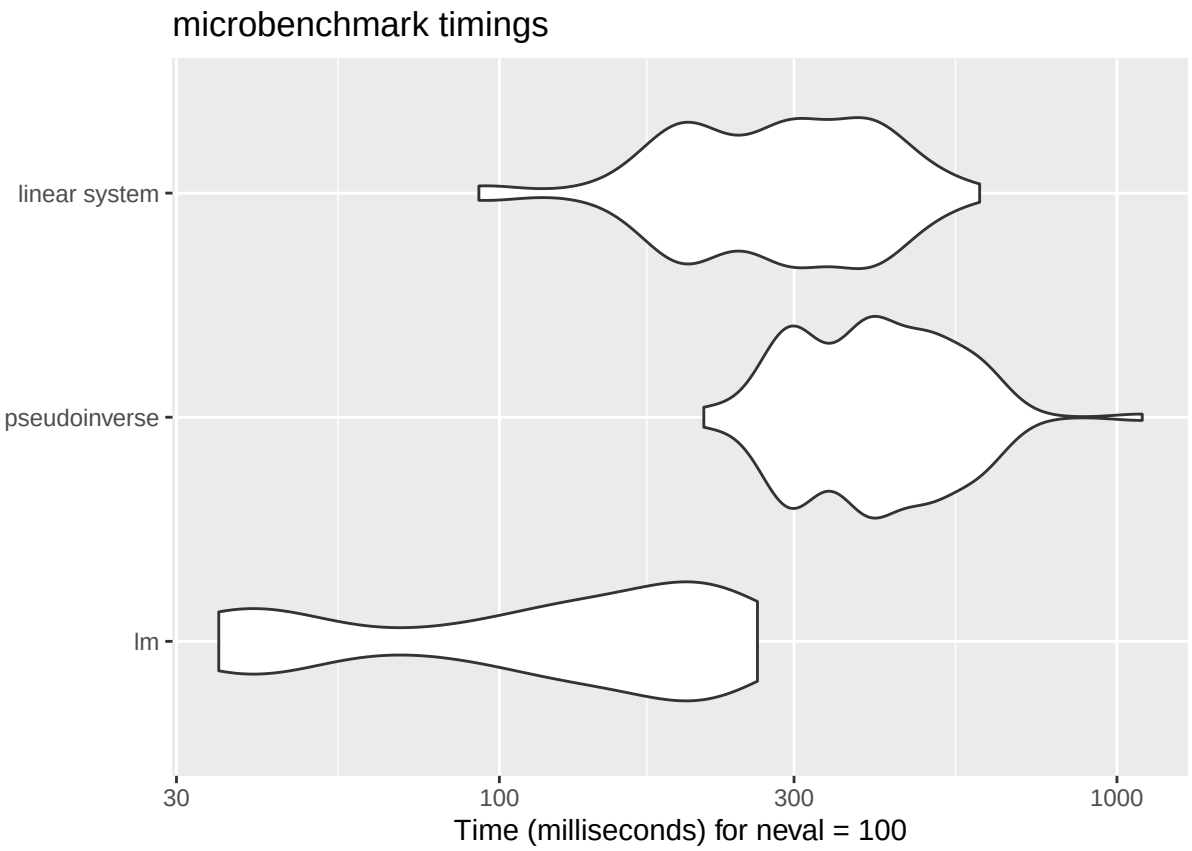


Biblioteca Microbenchmark

```
library(microbenchmark)
set.seed(2017)
n <- 10000
p <- 100
X <- matrix(rnorm(n*p), n, p)
y <- X %*% rnorm(p) + rnorm(n)  # Corregido: rnorm(n) en lugar de rnorm(100)

check_for_equal_coefs <- function(values) {
  tol <- 1e-12
  max_error <- max(c(abs(values[[1]] - values[[2]]),
                     abs(values[[2]] - values[[3]]),
                     abs(values[[1]] - values[[3]])))
  max_error < tol
}

mbm <- microbenchmark("lm" = { b <- lm(y ~ X + 0)$coef },
                      "pseudoinverse" = { b <- solve(t(X) %*% X) %*% t(X) %*% y },
                      "linear system" = { b <- solve(t(X) %*% X, t(X) %*% y) },
                      check = check_for_equal_coefs)

mbm
```

```
## Unit: milliseconds
##           expr       min        lq     mean   median       uq       max neval
##             lm  35.12278  51.70943 137.2735 135.5663 209.3838  261.7841   100
##  pseudoinverse 214.32082 301.09552 421.5151 399.0311 499.1255 1098.9487   100
##  linear system  92.64077 202.13270 296.7770 298.8450 399.5325  599.3821   100
```

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



Ejercicio 9

Penitencia de Newton

```
ti <- Sys.time()
suma <- 0
for (i in 1:100) {
  suma <- suma + i
}
print(suma)
```

```
## [1] 5050
```

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

```
## Time difference of 0.0044415 secs
```

```
# Algoritmo para sumar segun Newton
t2 <- Sys.time()
N <- 100
  suma_formula <- N * (N + 1) / 2

print(suma_formula)
```

```
## [1] 5050
```

```
t3 <- Sys.time()
t3-t2
```

```
## Time difference of 0.00193119 secs
```