Penitencia de Newton

Grupo Cookies

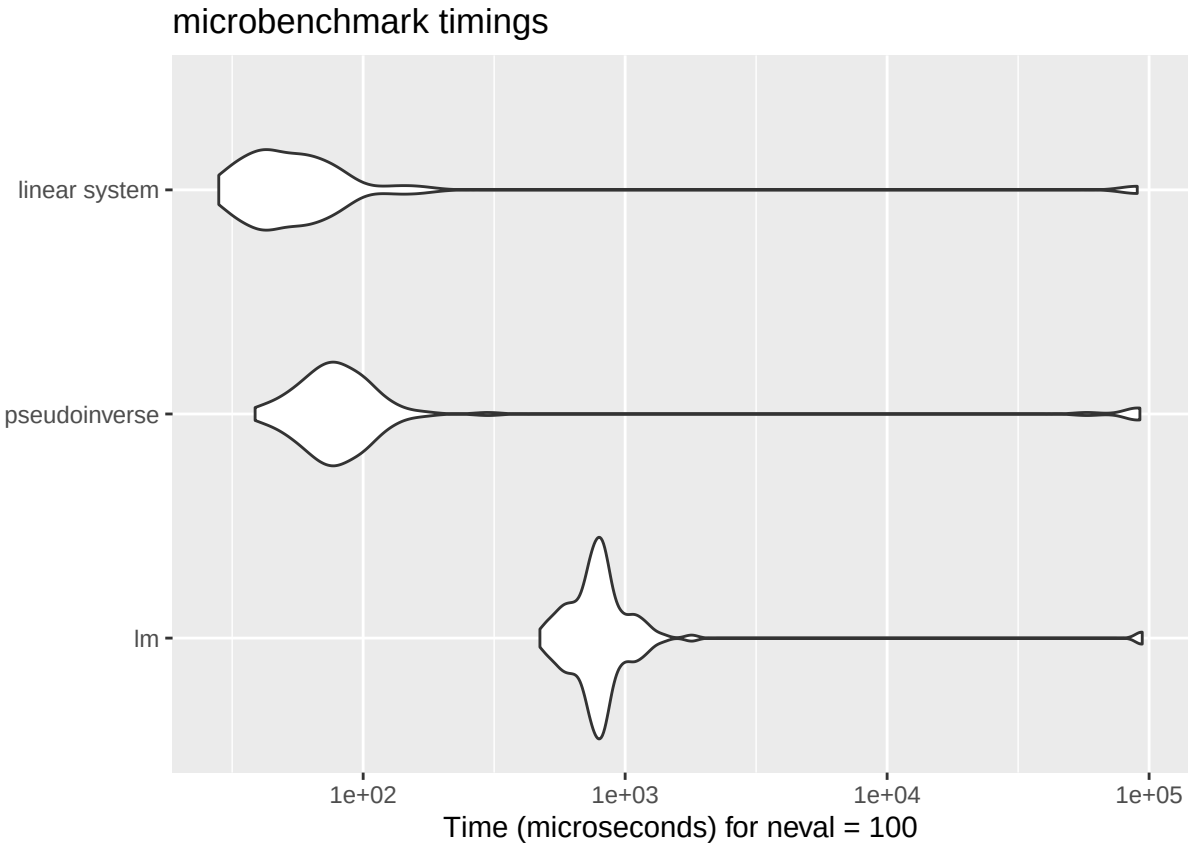
2025-05-07

Biblioteca Microbenchmark

```
library(microbenchmark)
set.seed(2017)
n <- 100  # número de observaciones
p <- 10   # número de variables (columnas de X)

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: microseconds
##           expr     min       lq     mean  median       uq      max neval
##             lm 472.783 688.8690 2667.378 785.125 859.9050 94152.65   100
##  pseudoinverse  38.680  64.6450 4277.118  79.676  99.1160 92234.71   100
##  linear system  28.100  39.6955 2757.560  51.970  69.0805 90119.35   100
```

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



##Generación 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
```

##Método 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] 78  2 37 42 96 30 92 64 58 21
```

```
#Muestra Ordenada
res
```

```
##  [1]  2 21 30 37 42 58 64 78 92 96
```

```
#Ordenación con el comando SORT de R-Cran
sort(x)
```

```
##  [1]  2 21 30 37 42 58 64 78 92 96
```

Algoritmo que sume del 1 al 100

```
suma <- sum(1:100)
suma
```

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

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 0.003899574 secs
```

```
# Algoritmo para sumar según 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 0.00184679 secs
```

```
ti <-Sys.time()
a=1
b=100  # a y b son el inicio y el fin de la suma
c=0
for(i in a:b) {
c=c+i
print(c)
tf <-Sys.time()



}
```

```
## [1] 1
## [1] 3
## [1] 6
## [1] 10
## [1] 15
## [1] 21
## [1] 28
## [1] 36
## [1] 45
## [1] 55
## [1] 66
## [1] 78
## [1] 91
## [1] 105
## [1] 120
## [1] 136
## [1] 153
## [1] 171
## [1] 190
## [1] 210
## [1] 231
## [1] 253
## [1] 276
## [1] 300
## [1] 325
## [1] 351
## [1] 378
## [1] 406
## [1] 435
## [1] 465
## [1] 496
## [1] 528
## [1] 561
## [1] 595
## [1] 630
## [1] 666
## [1] 703
## [1] 741
## [1] 780
## [1] 820
```

```
## [1] 861
## [1] 903
## [1] 946
## [1] 990
## [1] 1035
## [1] 1081
## [1] 1128
## [1] 1176
## [1] 1225
## [1] 1275
## [1] 1326
## [1] 1378
## [1] 1431
## [1] 1485
## [1] 1540
## [1] 1596
## [1] 1653
## [1] 1711
## [1] 1770
## [1] 1830
## [1] 1891
## [1] 1953
## [1] 2016
## [1] 2080
## [1] 2145
## [1] 2211
## [1] 2278
## [1] 2346
## [1] 2415
## [1] 2485
## [1] 2556
## [1] 2628
## [1] 2701
## [1] 2775
## [1] 2850
## [1] 2926
## [1] 3003
## [1] 3081
## [1] 3160
## [1] 3240
## [1] 3321
## [1] 3403
## [1] 3486
## [1] 3570
## [1] 3655
## [1] 3741
## [1] 3828
## [1] 3916
## [1] 4005
## [1] 4095
## [1] 4186
## [1] 4278
## [1] 4371
## [1] 4465
```

```
## [1] 4560
## [1] 4656
## [1] 4753
## [1] 4851
## [1] 4950
## [1] 5050
```

```
tf-ti
```

```
## Time difference of 0.006603956 secs
```

‘‘‘