

Técnicas y herramientas modernas I

Módulo 2- Teoría de colas

Grupo: Trabajo Aprobado

2025-04-24

```
# Cargar la biblioteca microbenchmark
library(microbenchmark)
library(ggplot2)

# Fijar semilla para reproducibilidad
set.seed(2017)

# Crear matriz de diseño y vector respuesta
n <- 2000
p <- 50
X <- matrix(rnorm(n * p), n, p)
y <- X %*% rnorm(p) + rnorm(n)

# Función para verificar si los coeficientes son equivalentes
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
}

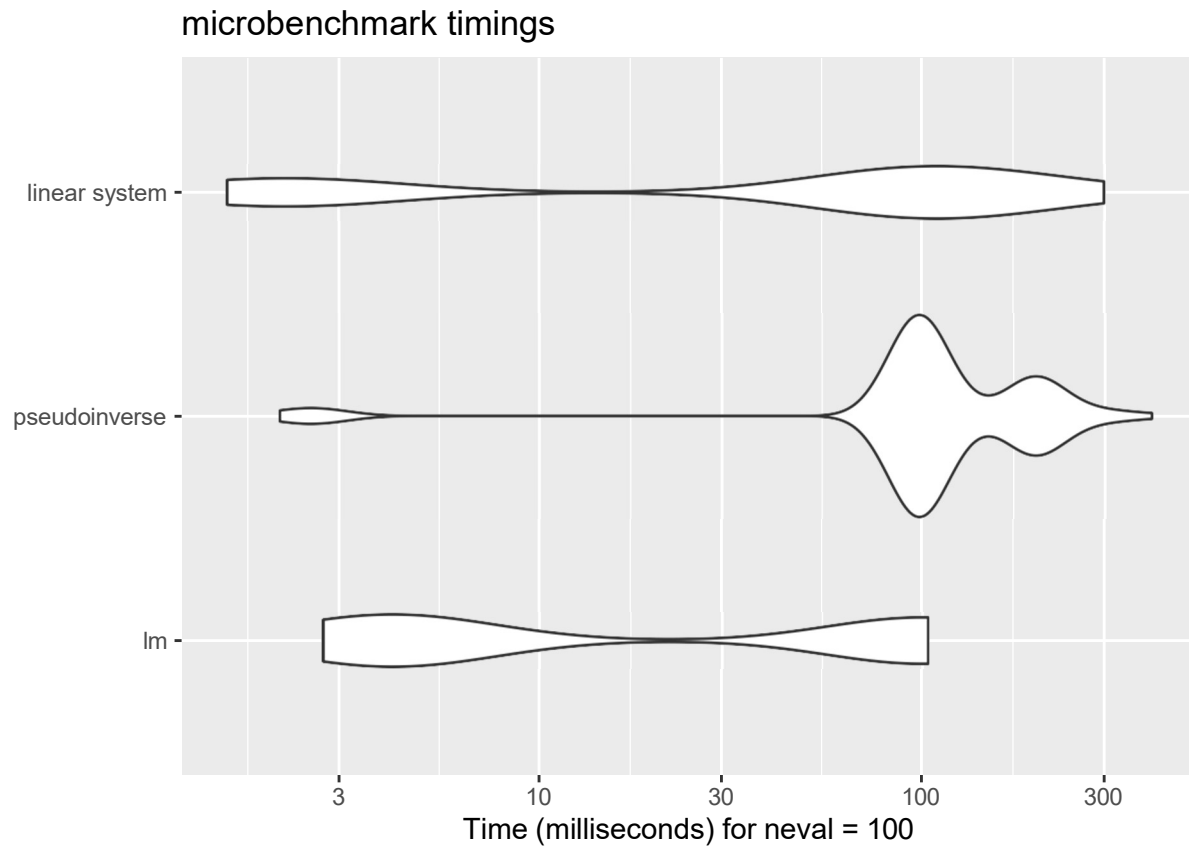
# Comparar tres formas de calcular coeficientes del modelo lineal
mbm <- microbenchmark(
  "lm" = {
    b <- lm(y ~ X + 0)$coef
  },
  "pseudoinverse" = {
    b <- solve(t(X) %*% X) %*% t(X) %*% y
  },
  "linear system" = {
    b <- solve(t(X) %*% X, t(X) %*% y)
  },
  check = check_for_equal_coefs,
  times = 100
)

# Mostrar resultados
print(mbm)
```

Unit: milliseconds

```
##          expr      min        lq      mean   median        uq      max  neval
##          lm  2.724722  4.369903  47.13468  5.73848   98.98584 103.3669   100
## pseudoinverse 2.104331 98.206189 126.57312 99.91501 196.21727 398.8304   100
## linear system 1.536522  2.461997  80.53721 97.35699  99.85531 299.4204   100
```

```
# Gráfico comparativo de los métodos
autoplot(mbm)
```



Ejercicio 2

Generar un vector secuencia

```
cat("1.4 Secuencia con for:\n")
```

```
## 1.4 Secuencia con for:
```

```
A <- numeric(50000)
start_time <- Sys.time()
for (i in 1:50000) {
  A[i] <- i * 2
}
end_time <- Sys.time()
print(head(A))
```

```
## [1]  2  4  6  8 10 12
```

```
print(tail(A))
```

```
## [1] 99990 99992 99994 99996 99998 100000
```

```
cat("Tiempo con for: ", end_time - start_time, "\n")
```

```
## Tiempo con for: 0.004497051
```

```
cat("\nSecuencia con seq:\n")
```

```
##
```

```
## Secuencia con seq:
```

```
start_time <- Sys.time()
B <- seq(2, 100000, by = 2)
end_time <- Sys.time()
print(head(B))
```

```
## [1] 2 4 6 8 10 12
```

```
print(tail(B))
```

```
## [1] 99990 99992 99994 99996 99998 100000
```

```
cat("Tiempo con seq: ", end_time - start_time, "\n")
```

```
## Tiempo con seq: 0.0009708405
```

Ejercicio 5

Implementación de una serie Fibonacci o Fibonacci

```
# Serie de Fibonacci hasta superar 1 millón
```

```
serie_fibonacci <- function() {
  fib <- c(0, 1)
  i <- 2
  while (TRUE) {
    fib[i + 1] <- fib[i] + fib[i - 1]
    if (fib[i + 1] > 1e6) break
    i <- i + 1
  }
  return(fib)
}
```

```
fibonacci <- serie_fibonacci()
fibonacci
```

```
## [1] 0 1 1 2 3 5 8 13 21
## [10] 34 55 89 144 233 377 610 987 1597
## [19] 2584 4181 6765 10946 17711 28657 46368 75025 121393
## [28] 196418 317811 514229 832040 1346269
```

```
cat("Cantidad de elementos hasta superar 1.000.000:", length(fibonacci), "\n")
```

```
## Cantidad de elementos hasta superar 1.000.000: 32
```

```
cat("Último número:", tail(fibonacci, 1), "\n")
```

```
## Último número: 1346269
```

Ejercicio 7

Ordenación de un vector por método burbuja

```
library(microbenchmark)
set.seed(123)
x <- sample(1:100, 10)
cat("\n1.7 Burbuja (10 elementos):\n")

##
## 1.7 Burbuja (10 elementos):
print(x)

## [1] 31 79 51 14 67 42 50 43 97 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)
print(res)

## [1] 14 25 31 42 43 50 51 67 79 97
print(sort(x))

## [1] 14 25 31 42 43 50 51 67 79 97
cat("\nComparación con 20.000 elementos usando microbenchmark:\n")

##
## Comparación con 20.000 elementos usando microbenchmark:
x_big <- sample(1:20000, 20000)

burbuja_lenta <- 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)
}
```

```

# Solo para comparación, comentar la burbuja si tarda mucho
# microbenchmark(burbuja_lenta(x_big), sort(x_big), times = 1)

cat("AVISO: No se ejecuta burbuja con 20.000 elementos porque tarda mucho.\n")

## AVISO: No se ejecuta burbuja con 20.000 elementos porque tarda mucho.
cat("Comparar manualmente si se desea. \n")

## Comparar manualmente si se desea.

```

Ejercicio 8

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 0.00307703 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 0.001567841 secs

```