Practica_PLN_Grupo08

2025-12-05

Práctica 8

CUIDADO AL EJECUTAR (Limpia el Global Enviroment)

```
rm(list = ls())
```

1º Parte

Lectura del archivo 10000 palabras

```
# Leer el archivo
lineas <- readLines("10000_formas_ortograficas.txt",
                    encoding = "UTF-8")
lineas[0:4]
```

```
## [1] "Forma\tFrecuencia\tFrec. norm." ",\t27823866\t56483.65"
## [3] "de\t26286396\t53362.52"         ".\t19226627\t39030.88"
```

```
# Buscar donde empieza la priemra linea
linea_inicio <- grep("^[^0-9\t\n]+\t[0-9]+\t[0-9]+", lineas)[1]

# Extraer 10000 líneas a partir del inicio
datos <- lineas[linea_inicio:(linea_inicio + 9999)]

# Separar cada línea por tabulador (\t)
partes <- strsplit(datos, "\t")
partes[0:4]
```

```
## [[1]]
## [1] ","        "27823866" "56483.65"
##
## [[2]]
## [1] "de"       "26286396" "53362.52"
##
## [[3]]
## [1] "."        "19226627" "39030.88"
##
## [[4]]
## [1] "la"       "15799962" "32074.6"
```

```r
# Extraer las columnas
formas <- sapply(partes, function(x) x[1])
frecuencias <- as.numeric(sapply(partes, function(x) x[2]))
frec_norm <- as.numeric(sapply(partes, function(x) x[3]))

# Crear el dataframe
tabla <- data.frame(Forma = formas,
                    Frecuencia = frecuencias,
                    Frec.norm = frec_norm,
                    stringsAsFactors = FALSE)
head(tabla)
```

```
##   Forma Frecuencia Frec.norm
## 1     ,   27823866  56483.65
## 2    de   26286396  53362.52
## 3     .   19226627  39030.88
## 4    la   15799962  32074.60
## 5   que   13350795  27102.69
## 6     y   11562228  23471.82
```

```r
tail(tabla)
```

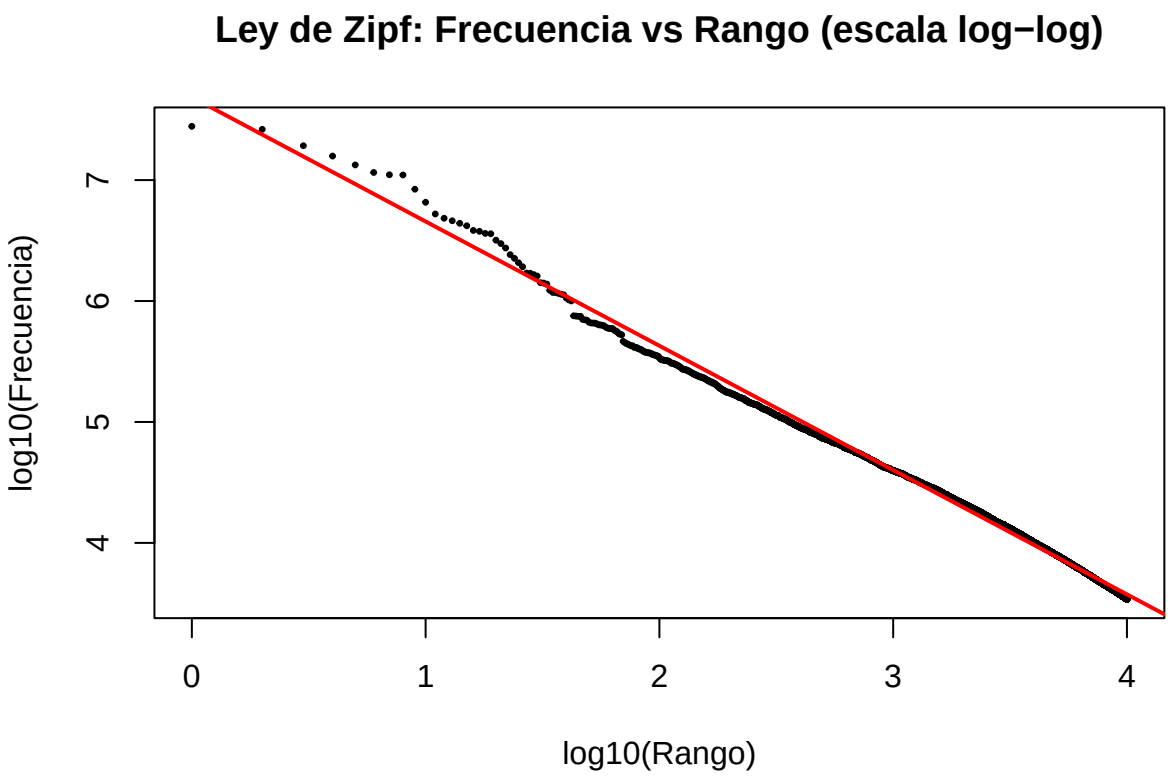
```
##                 Forma Frecuencia Frec.norm
## 9995       militancia       3422      6.94
## 9996         perciben       3421      6.94
## 9997         católico       3421      6.94
## 9998        convertía       3421      6.94
## 9999  especialización       3421      6.94
## 10000          golpeó       3421      6.94
```

Ley de Zipf

```r
## APARTADO 1
# Creamos la columna de rangos
tabla$rango <- 1:nrow(tabla)

# Gráfico log-log
plot(log10(tabla$rango),
     log10(tabla$Frecuencia),
     pch = 20, cex = 0.5,
     xlab = "log10(Rango)",
     ylab = "log10(Frecuencia)",
     main = "Ley de Zipf: Frecuencia vs Rango (escala log-log)")

# Ajuste lineal para visualizar la tendencia
modelo <- lm(log10(Frecuencia) ~ log10(rango), data = tabla)
abline(modelo, col = "red", lwd = 2)
```



Apartado 1.1

```
## Apartado 1.1
# 1.1.1) BUSCAR FORMAS REPETIDAS

duplicadas <- tabla$Forma[duplicated(tabla$Forma)]
duplicadas
```

```
## character(0)
```

```
# 1.1.2) SI HAY DUPLICADAS, MOSTRAR DETALLES

if (length(duplicadas) > 0) {

  lista_repetidas <- lapply(duplicadas, function(f) {
    indices <- which(tabla$Forma == f)
    data.frame(
      Forma = f,
      Linea = indices,
      Frecuencia = tabla$Frecuencia[indices]
    )
  })
```

```
  resultado <- do.call(rbind, lista_repetidas)
  resultado

} else {
  print("No hay formas repetidas en este listado.")}
```

```
## [1] "No hay formas repetidas en este listado."
```

Apartado 1.2

```
## Apartado 1.2

# Pasar todo a minúsculas
tabla$min_forma <- tolower(tabla$Forma)

# Contar cuántas formas básicas tienen variantes
conteo <- table(tolower(tabla$Forma))
duplicados <- conteo[conteo > 1]

cat("Formas básicas con duplicados no exactos:", length(duplicados), "\n")
```

```
## Formas básicas con duplicados no exactos: 760
```

```
# Mostrar primeros 5 ejemplos
cat("\nPrimeros 5 ejemplos:\n")
```

```
##
## Primeros 5 ejemplos:
```

```
for (i in 1:5) {
  forma <- names(duplicados)[i]
  variantes <- unique(tabla$Forma[tolower(tabla$Forma) == forma])
  cat(i, " '", forma, "' con las varientes: ", paste(variantes, collapse = ", "), "\n", sep = "")
}
```

```
## 1 'a' con las varientes: a, A
## 2 'abierto' con las varientes: abierto, Abierto
## 3 'acá' con las varientes: acá, Acá
## 4 'academia' con las varientes: Academia, academia
## 5 'acaso' con las varientes: acaso, Acaso
```

Apartado 1.3

```
library(stringi)
#Normalización
formas_norm <- stri_trans_nfc(tabla$Forma) # normaliza acentos
formas_norm <- gsub("\u00A0", " ", formas_norm)
```

```r
formas_norm <- gsub("\t", " ", formas_norm)

# Función para detectar si la palabra contiene caracter no español
tiene_no_espanoles <- function(texto) {
  grepl("[^A-Za-zñÑáéíóúÁÉÍÓÚüÜ .,;:¡!¿?---]", texto)
}

no_espanol <- tiene_no_espanoles(formas_norm)

# Número de palabras con caracteres no españoles (según nuestro criterio)
num_no_espanol <- sum(no_espanol)
cat("El nº de formas no españolas según nuestro criterio es: ", num_no_espanol)
```

```
## El nº de formas no españolas según nuestro criterio es:  223
```

```r
#Ejemplos
head(tabla$Forma[no_espanol],20)
```

```
##  [1] "\"" ")"  "("  "»"  "«"  "'"  "1"  "2"  """  """  "3"  "4"  "10" "5"  "6"
## [16] "20" "15" "7"  "12" "30"
```

2º Parte

Lectura del archivo 10000 palabras

```r
# Leer el archivo
lineas2 <- readLines("frecuencia_formas_ortograficas_1_3.txt",
                     encoding = "UTF-8")
lineas2[0:4]
```

```
## [1] "Forma\tFrecuencia\tFrec. norm." ",\t27823866\t56483.65"
## [3] "de\t26286396\t53362.52"         ".\t19226627\t39030.88"
```

```r
# Buscar donde empieza la priemra linea
linea_inicio2 <- grep("^[^0-9\t\n]+\t[0-9]+\t[0-9]+", lineas2)[1]


# Extraer todas las líneas a partir del inicio
datos2 <- lineas2[linea_inicio2:(linea_inicio2 + length(lineas2)-2)] #El -2 pq sinó aparecen 2 lineas v

# Separar cada línea por tabulador (\t)
partes2 <- strsplit(datos2, "\t")
partes2[0:4]
```

```
## [[1]]
## [1] ","        "27823866" "56483.65"
##
## [[2]]
## [1] "de"       "26286396" "53362.52"
```

```
##
## [[3]]
## [1] "."         "19226627" "39030.88"
##
## [[4]]
## [1] "la"        "15799962" "32074.6"
```

```
# Extraer las columnas
formas2 <- sapply(partes2, function(x) x[1])
frecuencias2 <- as.numeric(sapply(partes2, function(x) x[2]))
frec_norm2 <- as.numeric(sapply(partes2, function(x) x[3]))

# Crear el dataframe
tabla2 <- data.frame(Forma = formas2,
                     Frecuencia = frecuencias2,
                     Frec.norm = frec_norm2,
                     stringsAsFactors = FALSE)
head(tabla2)
```

```
##   Forma Frecuencia Frec.norm
## 1     ,   27823866  56483.65
## 2    de   26286396  53362.52
## 3     .   19226627  39030.88
## 4    la   15799962  32074.60
## 5   que   13350795  27102.69
## 6     y   11562228  23471.82
```

```
tail(tabla2)
```

```
##                   Forma Frecuencia Frec.norm
## 1484611         Paulitz          1         0
## 1484612        Paulitis          1         0
## 1484613       Paulistáo          1         0
## 1484614      Paulistana          1         0
## 1484615         Paulist          1         0
## 1484616 pauljohnsonesco          1         0
```

Apartado 2.3

```
library(stringi)

#Normalización
formas_norm2 <- stri_trans_nfc(tabla2$Forma) # normaliza acentos
formas_norm2 <- gsub("\u00A0", " ", formas_norm2)
formas_norm2 <- gsub("\t", " ", formas_norm2)

# Función para detectar si la palabra contiene caracter no español
tiene_no_espanoles <- function(texto) {
  grepl("[^A-Za-zñÑáéíóúÁÉÍÓÚüÜ .,;:¡!¿?---]", texto)
}
```

```r
no_espanol2 <- tiene_no_espanoles(formas_norm2)

# Número de palabras con caracteres no españoles (según nuestro criterio)
num_no_espanol2 <- sum(no_espanol2)
cat("El nº de formas no españolas según nuestro criterio es: ", num_no_espanol2)
```

```
## El nº de formas no españolas según nuestro criterio es:  77639
```

```r
#Ejemplos
head(tabla2$Forma[no_espanol2],20)
```

```
##  [1] "\"" ")"  "("  "»"  "«"  "'"  "1"  "2"  """  """  "3"  "4"  "10" "5"  "6"
## [16] "20" "15" "7"  "12" "30"
```

```r
tail(tabla2$Forma[no_espanol2],20)
```

```
##  [1] "Pathology®"         "pâtissiére"         "pâtissiere"
##  [4] "pâtisseries"        "PÂTISSERIE"         "patîsserie"
##  [7] "pâtiserie"          "patológicos3"       "Patrão"
## [10] "Pàtria"             "Pâtre"              "PÀTRICIA"
## [13] "patrilla$$$thrilla" "Patrimônio"         "Patrimonio18"
## [16] "patronos198"        "patrono58"          "PATT27"
## [19] "Paulão"             "paulavives12"
```

```r
#tabla2$Forma[no_espanol2]
```