

```
#####
##                               ##
##  LOS TEXTOS POÉTICOS DE FERNANDO DE HERRERA.           ##
##  APROXIMACIONES DESDE LA ESTILÍSTICA DE CORPUS Y LA ESTILOMETRÍA  ##
##  Tesis doctoral escrita por Laura Hernández Lorenzo      ##
##  Capítulo 6. Análisis con métodos de corpus y computacionales  ##
##  Apartado 6.2.1. Densidad léxica                          ##
##  Script 1 de la Tesis                                     ##
##                               ##
#####
```

```
# Creamos la ruta al directorio de trabajo donde está el corpus:
setwd("Dropbox/Tesis1/Corpus_Herrera/definitivo/Anexo_3")
```

```
# Introducimos en R el archivo del que queremos calcular la STTR:
text = scan("Pacheco_poemasmod2.txt", what="character", sep="\n")
```

```
# Convertimos a minúsculas:
Textminus <- tolower(text)
```

```
# Creamos una lista con las palabras del texto:
Textpalabras <- strsplit(Textminus, "\\W")
```

```
# Deshacemos la lista:
Textpalabrasdef <- unlist(Textpalabras)
```

```
# Eliminamos los huecos entre palabras:
no.huecos.v <- which(Textpalabrasdef!="")
Textpalabrasdef <- Textpalabrasdef[no.huecos.v]
```

```
Textpalabrasdef
totalwords= length(Textpalabrasdef)
```

```
# Establecemos la longitud que tendrán los segmentos a testar:
segmentLength = 2000
```

```
# Establecemos el número de segmentos en los que queda dividido el texto:
segmentNumber=as.integer(totalwords/segmentLength)
```

```
totalUnique= length(unique(Textpalabrasdef))
```

```
segments = matrix(list(), nrow=segmentNumber,ncol= segmentLength)
chunks=vector(mode="character",length=segmentNumber)
```

```

# Creamos el primer bucle:
for (i in 1:segmentNumber){
  j=1
  while(j<segmentLength+1){
segments[[i,j]] = c(segments[[i,j]],Textpalabrasdef[(i-1)*segmentLength+j])
  j=j+1
  }
}

```

```

# Creamos una función para calcular la TTR:
TTR <- function(textArg) 100*length(unique(textArg))/length(textArg)

```

```

# Creamos el segundo bucle:
TTRs=vector(mode="numeric",length=segmentNumber)
for(i in 1:segmentNumber){
  TTRs[i]=TTR(unlist(segments[i,]))
  meanTTR=mean(TTRs)
}

```

```

# Obtenemos la medida de las TTR por fragmentos del texto, esto es, la STTR:
meanTTR

```

```

#####
##                               ##
##  LOS TEXTOS POÉTICOS DE FERNANDO DE HERRERA.           ##
## APROXIMACIONES DESDE LA ESTILÍSTICA DE CORPUS Y LA ESTILOMETRÍA ##
##  Tesis doctoral escrita por Laura Hernández Lorenzo     ##
##  Capítulo 7. Análisis con métodos estilométricos       ##
## Apartado 7.1. Métodos supervisados de atribución de autoría: ##
##  cross-validation atendiendo al tamaño de las muestras  ##
##  Script 2 de la Tesis                                     ##
##                                                         ##
#####

```

```

# Una vez que tengo una tabla csv con todos los resultados ponderados de los diferentes
autores,
# introduzco la tabla en R:

```

```

tabla = read.csv("Size/Resultados.csv", header = FALSE)
tabla

```

```

# Creo dos variables para guardar, por un lado, los resultados (accuracy),
# y, por otro, el nº de palabras (words):
accuracy = tabla[,3]
accuracy
words = tabla[,4]

```

```

words
barplot(accuracy, words, plot = TRUE)
length(words)

# Escribo un bucle para obtener las medias de accuracy por intervalos:
final = c()
intervals = c(0, 1500, 3000, 4500, 6000, 9999999)
for(j in 2:length(intervals)) {
  R<- 0
  for(i in 1:length(words)) {
    if (words[i]>intervals[j-1] & words[i] < intervals[j])
      { R <- c(R, accuracy[i])
        }
  }
  final <- c(final, mean(R))
}
# Aquí termina el bucle

# Genero la figura 29:
barplot(final, xlab = "Texts size (words)", ylab= "Accuracy", col = c("red", "blue"), ylim =
c(0,1.0), names.arg = c("0-1500", "1500-3000", "3000-4500", "4500-6000", "6000-or more"))
#####
##                                     ##
##  LOS TEXTOS POÉTICOS DE FERNANDO DE HERRERA.          ##
##  APROXIMACIONES DESDE LA ESTILÍSTICA DE CORPUS Y LA ESTILOMETRÍA  ##
##  Tesis doctoral escrita por Laura Hernández Lorenzo      ##
##  Capítulo 7. Análisis con métodos estilométricos        ##
##  Apartado 7.7.Verificación de autoría: impostores        ##
##  Script 3 de la Tesis                                     ##
##                                     ##
#####

library(stylo) # version >= 0.6.7

## Para el corpus de sonetos de poesía áurea
# cargo los ficheros en R desde una carpeta
tokenized.texts = load.corpus.and.parse(files = "all", corpus.dir="Desktop/corpus")

# creo una lista de las palabras más frecuentes (limitada a 5000, por ejemplo, pero lo puedo
cambiar según el análisis):
features = make.frequency.list(tokenized.texts, head = 5000)

# creo una tabla de frecuencias relativas:

```

```

data = make.table.of.frequencies(tokenized.texts, features, relative = TRUE)

# Aplico Impostores: ¿quién es AN? (usando Delta clásica como medida de distancia)
  results = imposters(reference.set = data[-c(3),], test = data[3,])
write.csv(cbind(results, names(results)), file = "my_resultsANDeltadef3000.csv")

# Aplico Impostores: ¿quién es AN? (usando la Delta de Eder como medida de distancia)
  results = imposters(reference.set = data[-c(3),], test = data[3,], distance = "eder")
  write.csv(cbind(results, names(results)), file = "my_resultsANEder4000.csv")

# Aplico Impostores: ¿quién es AN? (usando Cosine Delta como medida de distancia)
  results = imposters(reference.set = data[-c(3),], test = data[3,], distance = "wurzburg")
  write.csv(cbind(results, names(results)), file = "my_resultsANWurzburgdef5000.csv")

# Para obtener los valores que marcan los límites de la zona gris
  imposters.optimize(data)
#####
#####
##
## LOS TEXTOS POÉTICOS DE FERNANDO DE HERRERA. ##
## APROXIMACIONES DESDE LA ESTILÍSTICA DE CORPUS Y LA ESTILOMETRÍA ##
## Tesis doctoral escrita por Laura Hernández Lorenzo ##
## Capítulo 7. Análisis con métodos estilométricos ##
## Apartado 7.8. Un mapa de la evolución de la poesía áurea: análisis de redes ##
## Script 4 de la Tesis ##
##
#####
#####

# Establecemos el directorio de trabajo donde está la tabla exportada de Gephi
  setwd("Desktop")

# Importamos la tabla en R y guardamos los datos de las fechas de nacimiento
  my.data = read.csv("ADSO table3.csv")
  Birth = my.data[,15]

# Creamos un histograma para las fechas de nacimiento
  hist(Birth)

# Creación del boxplot para evaluar la relación entre cronología y modularidad

```

```

Modularity = my.data[,20]
boxplot(Birth~Modularity, main = "Modularity class", ylab = "Birth date", col = c("green",
"blue", "brown"))

# Para ver si existe una correlación entre fecha de nacimiento y centralidad en el grafo

# Calculamos la diferencia de cada fecha de nacimiento a la media y convertimos a valores
absolutos
difference_to_mean_year = Birth - mean(Birth)
abs_difference_to_mean_year = abs(difference_to_mean_year)

# Guardamos los datos de closeness y harmonic closeness centrality
Closeness = my.data[,12]
Harmonic_closeness_centrality = my.data[,13]

# Creamos dos variables que incluyan los dos conjuntos de datos a testar (fecha de nacimiento
+ medida de centralidad)
x = abs_difference_to_mean_year
y = Closeness

# Generamos la regresión lineal
regression <- lm(x~y)

# Pedimos el resumen de resultados para ver el valor p obtenido
summary(regression)

# Excluimos a Joseph de Litala como outlier
Birth2 = my.data[-(4),15]
my.data2 = my.data[-(4),]

# Repetimos los procesos anteriores del análisis sin Joseph de Litala
difference_to_mean_year2 = Birth2 - mean(Birth2)
abs_difference_to_mean_year2 = abs(difference_to_mean_year2)
Closeness2 = my.data2[,12]
Harmonic_closeness_centrality2 = my.data2[,13]
x = abs_difference_to_mean_year2
y = Harmonic_closeness_centrality2
regression <- lm(x~y)
summary(regression)

```