

# Introducción al análisis de textos con *Quanteda*

## Text analysis with *Quanteda*: an introduction

**Sergio Castro-Cortacero; Nicolás Robinson-García**

Cómo citar este artículo:

**Castro-Cortacero, Sergio; Robinson-García, Nicolás** (2025). "Introducción al análisis de textos con *Quanteda* [Text analysis with *Quanteda*: an introduction]". *Infonomy*, 3(6) e25040.

<https://doi.org/10.3145/infonomy.25.040>

Artículo recibido: 16-12-2025

Artículo aprobado: 14-01-2026



**Sergio Castro-Cortacero**

<https://orcid.org/0009-0006-6362-5069>

<https://directorioexit.info/ficha7315>

Universidad de Granada

Departamento de Información y Comunicación

Unit for Computational Humanities and Social Sciences

EC3 Research Group

Granada, España

[cortacero@ugr.es](mailto:cortacero@ugr.es)



**Nicolás Robinson-García**

<https://orcid.org/0000-0002-0585-7359>

<https://directorioexit.info/ficha1592>

Universidad de Granada

Departamento de Información y Comunicación

Unit for Computational Humanities and Social Sciences

EC3 Research Group

Granada, España

[elrobin@ugr.es](mailto:elrobin@ugr.es)



## Resumen

El análisis cuantitativo de textos se ha consolidado como una metodología clave en el marco del giro computacional de las ciencias sociales y las humanidades. Este artículo presenta una introducción aplicada al uso del paquete *Quanteda* para el análisis sistemático de corpus textuales en el entorno *R*. El objetivo es mostrar, mediante un enfoque tutorial, cómo diseñar y ejecutar un flujo básico de análisis de textos cuantitativo. Para ello, se emplea como caso de estudio un conjunto de entradas del diario de Alexéi Navalny, sobre el que se aplican procesos de carga, limpieza, tokenización y eliminación de palabras vacías. Los resultados obtenidos a través del análisis de frecuencias, nubes de palabras y modelado de temas evidencian la capacidad de *Quanteda* para identificar patrones léxicos y temáticos relevantes. El trabajo demuestra que *Quanteda* es accesible, reproducible y especialmente adecuado para contextos docentes e investigaciones exploratorias en humanidades digitales.

## Palabras clave

Análisis de textos; *Quanteda*; Humanidades digitales; *R*; Análisis cuantitativo; Minería de textos; Procesamiento del lenguaje natural; *Topic modeling*; Visualización de datos; Metodología computacional.

## Abstract

Quantitative text analysis has become a core methodology within the computational turn in the social sciences and humanities. This paper presents an applied introduction to the *Quanteda* package for systematic text analysis in the *R* environment. The aim is to demonstrate, through a tutorial-oriented approach, how to design and implement a basic quantitative text analysis workflow. A set of diary entries by Alexei Navalny is used as a case study to illustrate corpus loading, preprocessing, tokenization, and stopword removal. Results obtained from frequency analysis, word clouds, and topic modeling show *Quanteda*'s ability to identify relevant lexical and thematic patterns. The paper demonstrates that *Quanteda* is an accessible and reproducible tool, particularly well suited for teaching contexts and exploratory research in digital humanities.

## Keywords

Text analysis; *Quanteda*; Digital humanities; *R*; Quantitative analysis; Text mining; Natural language processing; Topic modeling; Data visualization; Computational methodology.

## 1. Introducción

Desde que en los años 50 Harold Lasswell utilizara tarjetas perforadas generadas por tabuladoras para ayudarle en el análisis de datos textuales (Stone, 2020), los estudios de esta materia han experimentado un marcado “giro computacional” (Berry, 2011; Gallego-Cuiñas; Torres-Salinas, 2024). El cambio ha sido impulsado por los avances tecnológicos, especialmente en las áreas de lingüística computacional, *machine learning* y más recientemente, los grandes modelos de lenguaje y la inteligencia artificial generativa tipo *ChatGPT* y similares.

En esta nota compartimos nuestra experiencia con el análisis de datos textuales en general y más concretamente con el uso del paquete *Quanteda* (**Benoit et al.**, 2018). Para ello y tras una breve introducción al análisis de textos, explicaremos este paquete, desarrollado en el lenguaje de programación en abierto *R* (*R Core Team*, 2021). Seguidamente, detallaremos cómo emplear *Quanteda* para diseñar un análisis de datos textuales, incidiendo especialmente en todas las fases de pre-procesamiento de datos. Para ello, seguiremos una estructura modo tutorial, en la que utilizaremos unos textos extraídos de la obra *Patriot: A memoir* (**Navalny**, 2024), de las memorias de los últimos años de vida del disidente ruso Alexéi Navalny. Finalmente, concluiremos mostrando a modo de ejemplo algunos de los análisis que permite hacer *Quanteda* y los tipos de respuestas de investigación para los que puede servir este potente paquete de *R*.

## 2. Breve introducción al análisis cuantitativo de textos

El análisis de textos cuantitativo es un conjunto de metodologías para explorar grandes volúmenes de datos textuales a través de técnicas computacionales. A diferencia del análisis cualitativo, que se centra en la interpretación subjetiva del contenido, este enfoque extrae patrones medibles, conexiones semánticas, o tendencias y temáticas recurrentes, lo cual es especialmente útil en disciplinas tales como las ciencias políticas, la comunicación o las humanidades digitales entre otras. Hablamos de “datos” y no únicamente de “textos” porque la era digital ha transformado la concepción de los medios textuales; redes sociales, blogs, ediciones digitales de periódicos y otras plataformas virtuales son fábricas incesantes de producción textual (**Arcila-Calderón et al.**, 2016). No es fácil navegar en un océano desestructurado de información diseminada sin aparente relación. Sin embargo, las herramientas disponibles para estructurar estos datos y convertirlos en información interpretable son cada vez más accesibles.

Ya no es necesario recurrir a archivos físicos o hemerotecas para realizar un seguimiento mediático de un caso político: la sociedad de la información ha facilitado el acceso a grandes volúmenes de contenido digitalizado desde dispositivos portátiles (**Arnold et al.**, 2019). Este acceso no solo abarca enormes cantidades de información, sino que también genera “datos” en proporciones masivas. Este volumen inmenso requiere herramientas computacionales avanzadas para procesar, analizar y dar sentido a unos datos cada vez más grandes, diversos y variados, respondiendo así a las preguntas planteadas por los profesionales de la información (**Gallego-Cuiñas; Torres-Salinas**, 2024).

## 3. Breve descripción de *Quanteda*

No es posible entender *Quanteda* sin antes realizar una breve introducción al lenguaje en el que se basa: *R*. Este lenguaje, al igual que otros utilizados en la ciencia de datos como *Python* o *Julia* (**Wickham; Grolemund**, 2017), destaca por su capacidad para visualizar datos de forma eficiente. Con *R* es posible crear gráficos y modelos representativos de los materiales trabajados. Es versátil tanto para estudios estadísticos como para investigaciones que requieren mapas o grafos complejos (**Grolemund**, 2014). A lo largo del tiempo, y gracias a la contribución activa de su comunidad elaborando paquetes, *R* ha evolucionado más allá de su enfoque estadístico inicial. Hoy en día, permite realizar procesos avanzados que abarcan

desde el manejo de *big data* hasta la minería de texto. Es especialmente utilizado en el campo de las humanidades digitales por su accesibilidad, gratuidad y enorme comunidad que hay detrás dándole soporte. Existen paquetes específicos adaptados a las diferentes necesidades de los investigadores en estos campos, convirtiendo a *R* en una opción preferida para el análisis cuantitativo y la visualización de datos en proyectos interdisciplinarios (**Arnold et al.**, 2019).

*Quanteda* se inscribe en esta evolución de *R*, configurándose como un paquete diseñado específicamente para el análisis cuantitativo de textos, con el que generar visualizaciones, tablas y otros modelos a partir de corpus textuales. Desarrollado por Kenneth Benoit y Kohei Watanabe con el apoyo de una subvención del *European Research Council* (ERC-2011-StG 283794-Quantess), *Quanteda* se define en su página oficial como una herramienta que

“reduce las barreras para aprender y usar el análisis cuantitativo de textos y el procesamiento de lenguaje natural incluso para programadores experimentados en *R*”.

Una de las características más destacadas de *Quanteda* es su accesibilidad. Aunque el análisis textual puede parecer inicialmente una tarea compleja y árida, este paquete simplifica procesos que tradicionalmente requieren un alto nivel de conocimientos técnicos. Diseñado principalmente para fines académicos, *Quanteda* ofrece un conjunto de funciones intuitivas que permiten a profesionales no especializados en programación abordar tareas complejas de análisis de texto. Si bien requiere superar una curva de aprendizaje, esta no es pronunciada, lo que convierte a *Quanteda* en ideal tanto para expertos como para principiantes interesados en el análisis de textos.

¿Por qué centrarse en el análisis de textos? Como se ha mencionado previamente, la lingüística computacional y el análisis de grandes corpus han sido un foco central del giro computacional. Sin embargo, este tipo de análisis trasciende lo literario y tiene aplicaciones significativas en otras áreas, como las redes sociales. Plataformas como *Twitter*, *Facebook* y otros portales web son fuentes inagotables de datos textuales, útiles para estudios sociales, periodísticos o políticos. Por ejemplo, el trabajo de **Arcila-Calderón et al.** (2016) ilustra cómo los programas informáticos son esenciales para analizar casos como los *Papeles de Panamá*, donde el volumen de datos requería métodos avanzados para extraer información de uno de los mayores escándalos de fraude financiero que impactó a España.

A modo ilustrativo y siguiendo la línea de pensamiento planteada en el párrafo anterior, vamos a mostrar las funciones básicas de *Quanteda*, realizando un sencillo análisis de sentimientos y temático de un corpus compuesto por varias entradas del diario de Alexéi Navalny, una figura clave en el activismo político contemporáneo en Rusia. Estas entradas, publicadas en octubre de 2024 tras su encarcelamiento, reflejan los pensamientos y reflexiones del opositor al gobierno de Putin, condenado a 19 años de prisión.

Este material es un punto de partida ideal para demostrar las capacidades de *Quanteda*. El diario no solo ofrece un valioso contexto político y personal, sino que también evidencia cómo herramientas computacionales, como el que se va a aplicar

en este caso, pueden enriquecer el análisis crítico y proponer perspectivas únicas en el estudio de discursos y temas complejos.

#### 4. Análisis del diario de Alexei Navalny como caso de estudio

Vamos a exponer, paso a paso, el proceso de análisis a través de *Quanteda* aplicado al citado texto. Esta demostración constará de dos bloques: la carga y procesamiento del texto y la aplicación del paquete para realizar el análisis de contenido y la visualización de los datos. Todos los procesos y materiales que se detallarán están disponibles en:

*elrobin/ysp-Quanteda: Tutorial del paquete de análisis de textos Quanteda para #YoSigo UGR*

<https://github.com/elrobin/ysp-quanteda>

##### Paso 1. Procesamiento del texto

En primer lugar, deberemos instalar en *R* los paquetes incluidos en la siguiente función:

Código 1. Instalación de paquetes en *R*.

```
install.packages(c("Quanteda", "readtext", "dplyr", "stringi",  
"Quanteda.textplots", "ggplot2", "stringr", "tm", "topicmodels"))
```

Esta larga lista agrupa todos los elementos de los que haremos uso en este ejercicio. Iremos llamando poco a poco a cada uno según las necesidades que tengamos a lo largo del análisis. Aquí podemos observar dos tipos de paquetes: los que se encargan de trabajar sobre el texto y manipularlo (*Quanteda*, *readtext*, *dplyr*, *tm*, *stringi*, *stringr*) y los que se encargan de visualizar y transformar los datos anteriores en modelos interpretables (*Quanteda.textplots*, *ggplot2* y *topicmodels*).

Para este bloque no los utilizaremos todos, sino los necesarios para la carga y procesamiento de datos. Con ello, solo llamaremos a los siguientes:

Código 2. Llamada de las librerías en *R*.

```
library(Quanteda)  
library(readtext)  
library(dplyr)  
library(stringi)
```

Una vez cargados, el programa estará listo para leer el archivo de texto. Crearemos un objeto que almacene en texto plano el contenido del archivo. La fórmula que vamos a escribir para decirle a *Quanteda* qué archivo analizar será el siguiente:

Código 3. Exportación del material a *RStudio*

```
navalny_raw <- as.character(readtext("NAVALNY.txt"))
```

Debemos aclarar que *Quanteda* funciona bien con materiales configurados en textos simples. Es posible trabajar y almacenar otros formatos como *Word* o *PDF*, pero no

recomendamos su uso porque son materiales que pueden albergar más información de la necesaria para este tipo de análisis y podría entorpecer el estudio al trabajarlos en *R*. Por lo que, cuando más simple sea el material que se trabaje, más limpio será el resultado.

Dicho esto, para comprobar que *R* ha cargado el texto, usaremos una sencilla fórmula con el paquete *stringi* para recuperar los 75 primeros caracteres del texto.

Código 4. Comprobación de la exportación en *R*.

```
stri_sub(navalny_raw, 1, 75)
```

Si los pasos se han ejecutado correctamente, la línea debería de devolver la siguiente respuesta:

```
[1] "2022\nJanuary 17th\nExactly one year ago today I came home, to Russia.\nI didn"
```

En algunas ocasiones es necesario acotar el campo de análisis. Dependiendo del objeto de estudio, puede que nuestra área de interés esté concentrada en un punto concreto del archivo. Saber separar la información innecesaria de la útil es vital para desarrollar un análisis de contenido. Si el texto contiene secciones que no necesitamos podemos filtrarlas o limpiarlas en esta etapa a través del siguiente comando

Código 5. Creación de un objeto que limite la información que se trabajará en *R*.

```
(start_v <- stri_locate_first_fixed(navalny_raw, "2023\nJanuary 12th")[1])  
(end_v <- stri_locate_last_fixed(navalny_raw, "Alexei Navalny died")[1])  
navalny_fix <- stri_sub(navalny_raw, start_v, end_v)  
length(navalny_fix)
```

A través del objeto *navalny\_fix* se ha filtrado toda la información innecesaria por arriba y por bajo de nuestra área de interés por medio de los indicadores que hemos puesto en los objetos *start\_v* y *end\_v*. Como si de un folio recortado se tratara, se ha conseguido los datos que serán principalmente objeto del análisis. Sin embargo, el texto que manejamos ya está seleccionado, por lo que no es necesario aplicar este paso. Por tanto, seguiremos trabajando en los pasos siguientes con el objeto *navalny\_raw*.

La limpieza conlleva un orden, saber organizar la información que manejamos puede ser de ayuda para estructurar el análisis. Como estamos trabajando con un diario, vamos crear un índice que separe los años y un subíndice que diferencie, por cada año, las entradas por día.

Para la identificación de las entradas anuales se emplearán expresiones regulares, mediante una búsqueda basada en patrones definidos previamente, utilizando la función *grep*.

Código 6. Creación de un índice por años en *R*.

```
year_indices <- grep("^\\d{4}$", lines)

print(year_indices) # Muestra las líneas que contienen el año
```

Una vez desglosados los años, haremos que *R* separe de forma individual cada entrada del diario al año que pertenezca, de tal forma que consigamos 3 listas, cada una con una serie de sublistas.

Código 7. Creación de una función para dividir las entradas del diario por año y por día.

```
# Dividimos el texto en entradas anidadas (por año y día)
yearly_entries <- lapply(
  seq_along(year_indices),
  function(i) {
    start_year <- year_indices[i]
    end_year <- if (i < length(year_indices))
      year_indices[i + 1] - 1
    else
      length(lines)

    # Extraemos las líneas correspondientes al año actual
    year_lines <- lines[start_year:end_year]

    # Encontrar las entradas diarias dentro del año actual
    day_indices <- grep(
      "(January|February|March|April|May|June|July|August|September|October|November|December)\\s+\\d{1,2}(st|nd|rd|th)?$",
      year_lines
    )

    # Crear una sublista para cada día dentro del año
    entries <- lapply(seq_along(day_indices), function(j) {
      start_day <- day_indices[j]
      end_day <- if (j < length(day_indices))
        day_indices[j + 1] - 1
      else
        length(year_lines)
      year_lines[start_day:end_day]
    })

    # Devolver una lista con el año y sus entradas diarias
    list(year = year_lines[1], entries = entries)
  })
```

Una vez se tiene el set de datos organizado, podremos llevar a cabo un proceso de moldaje para adecuarlo a los criterios de análisis de *Quanteda*. Uno de esos pasos será manipularlo para convertir todo el texto a minúscula y eliminar los signos de puntuación.

Código 8. Eliminación de signos de puntuación y homogenización del texto en *R*.

```
# Convertir cada entrada diaria a minúsculas y eliminar signos de puntuación
yearly_entries <- lapply(yearly_entries, function(year) {
  year$entries <- lapply(year$entries, function(entry) {
    # Convertir el texto a minúsculas y eliminar puntuación
    entry <- char_tolower(entry)
    entry <- gsub("[[:punct:]]", "", entry)
    entry
  })
  year # Devolver la lista de año modificada
})
```

La necesidad de modificar la capitalización para homogeneizar el texto radica en que *Quanteda* es muy sensible a las mayúsculas. Esto significa que puede considerar como elementos distintos dos instancias de la misma palabra que solo se diferencian por la capitalización de la primera letra. Por ejemplo, las palabras "navalny" y "Navalny" se tratarían como términos diferentes en ese caso.

Una vez arreglado, dividiremos el texto para convertir cada palabra como unidades independientes por medio de la función *tokens*:

Código 9. Tokenización del texto en *R*.

```
# Tokenizar cada entrada diaria dentro de cada año
yearly_entries <- lapply(yearly_entries, function(year) {
  year$entries <- lapply(year$entries, function(entry) {
    tokens(entry, what = "word")
  })
  year # Devolver la lista de año modificada
})
```

Individualizados como elementos únicos, nos desprenderemos de todas las palabras que no son útiles para nuestro caso:

Código 10. Eliminación de palabras vacías en *R*.

```
# Eliminar palabras vacías en inglés en cada entrada diaria
yearly_entries <- lapply(yearly_entries, function(year) {
  year$entries <- lapply(year$entries, function(entry) {
    tokens_remove(entry, pattern = stopwords("en"))
  })
  year # Devolver la lista de año modificada
})
```



En este bloque, exploraremos dos enfoques principales para analizar nuestro conjunto de datos limpio y depurado, representado por el objeto *yearly\_entries*. Con él, ahora podemos hacer uso de las diferentes herramientas de representación que nos ofrece los paquetes que se han citado al principio del artículo: *Quanteda.textplots*, *ggplot2* y *topicmodels*.

Con *Quanteda.textplots* podemos construir nubes de palabras, muy útiles para poder identificar aquellas palabras más repetidas y como se distribuyen a lo largo del texto.

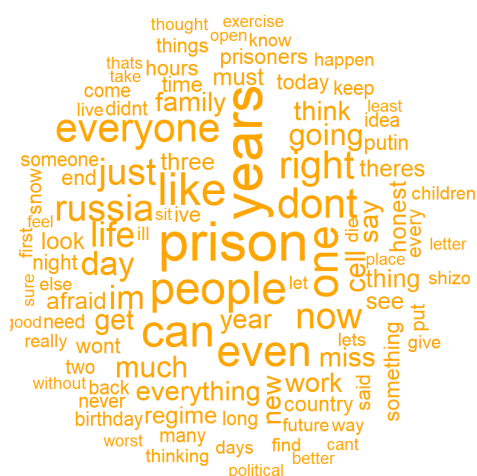


Figura 1. Representación de nube de palabras

```
library(Quanteda.textplots)

# Generar la nube de palabras a partir de la matriz de frecuencias
textplot_wordcloud(dfm_all, max_words = 100, color = "orange")
```

Con *ggplot2* podríamos construir una tabla de frecuencia de términos. Esta nos permite identificar las palabras más comunes, facilitando un primer visionado de aquellos posibles temas recurrentes que pudieran ser interesantes de analizar.

### Dispersión léxica

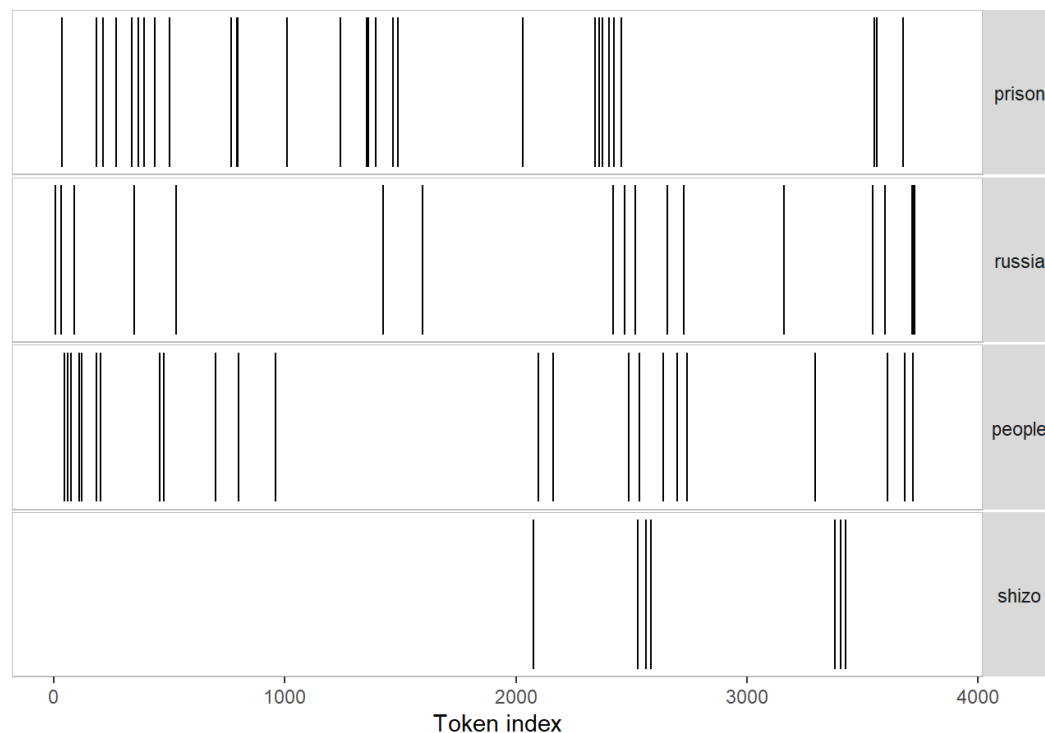


Figura 2. Representación de mapa de dispersión léxica

Código 12. Creación de una tabla de dispersión léxica con *ggplot2*.

```
library(ggplot2)

# Convertir el top 10 en un dataframe
top_words_df <- data.frame(
  word = names(top_words),
  frequency = as.numeric(top_words)
)

# Crear el gráfico de barras
ggplot(top_words_df, aes(x = reorder(word, frequency), y = frequency)) +
  geom_bar(stat = "identity", fill = "black") +
  coord_flip() +
  labs(title = "Top 15 Most Frequent Words",
       x = "Word",
       y = "Frequency") +
  theme_minimal()
```

Como se puede observar, tanto la nube de palabras como la tabla de frecuencias cumple el mismo cometido: analizar la frecuencia de aparición de los términos de un texto. La utilidad de una u otra dependerá del enfoque que se quiera dar, siendo esta última más exacta en términos cuantitativos.

Análisis de temáticas (*topic modelling*)

Por último, para la herramienta *topicmodels* podemos construir tablas temáticas con el objetivo de descubrir temas latentes que aparecen recurrentemente, agrupando palabras que suelen aparecer juntas. Cada tema representa un conjunto de palabras que tienden a repetirse en determinados contextos, permitiéndonos inferir posibles temas o conceptos sin necesidad de leer todo el texto.

	Topic 1	Topic 2	Topic 3	Topic 4
[1,]	"people"	"years"	"im"	"cell"
[2,]	"russia"	"prison"	"snow"	"hours"
[3,]	"everyone"	"life"	"birthday"	"putin"
[4,]	"years"	"going"	"like"	"can"
[5,]	"dont"	"right"	"shovels"	"night"
[6,]	"miss"	"even"	"days"	"like"
[7,]	"honest"	"one"	"boy"	"one"
[8,]	"afraid"	"regime"	"criminal"	"work"
[9,]	"now"	"die"	"prison"	"just"
[10,]	"must"	"happen"	"now"	"even"

Figura 3. Representación de tabla temática.

Código 13. Creación de una tabla de temática en R.

```
library(tm)
library(topicmodels)
# Convertir la matriz dfm a formato compatible con LDA
dfm_topics_lda <- convert(dfm_topics, to = "topicmodels")

# Definir el número de temas
num_topics <- 4 # Ajusta el número de temas según el análisis deseado

# Ajustar el modelo LDA
lda_model <- LDA(dfm_topics_lda, k = num_topics, control = list(seed = 1234))

# Obtener las palabras clave de cada tema
terms(lda_model, 10) # Muestra las 10 palabras más representativas de cada tema
```

Por ejemplo, este caso parece que el texto aborda temáticas diversas relacionadas con experiencias personales, política, y narrativas situacionales. La columna 1 parece centrarse en reflexiones sobre personas, emociones y situaciones generales, como "people", "everyone" y "miss". La columna 2 apunta a cuestiones políticas y de justicia, con términos como "russia", "prison", "regime" y "die", posiblemente relacionadas con un contexto opresivo o represivo. La columna 3 incluye palabras como "birthday", "shovels" y "boy", lo que podría aludir a escenas más cotidianas o personales, con una narrativa anecdótica. Finalmente, la columna 4 combina palabras como "cell", "hours",

"work", y "night", sugiriendo descripciones de entornos de trabajo o confinamiento, quizás bajo condiciones exigentes o restrictivas.

## 5. Conclusión

Las herramientas de análisis cuantitativo de textos se han convertido en un elemento central en la transición computacional que atraviesa actualmente las ciencias sociales y las humanidades. En este escenario, *Quanteda* destaca como un paquete especialmente adecuado para abordar corpus textuales desde una perspectiva sistemática y reproducible, integrándose de forma natural en el ecosistema del lenguaje R. Su relativa facilidad de uso, unida a la potencia de sus funciones, lo convierte en un recurso idóneo para introducir a estudiantes e investigadores en el análisis computacional de textos. En este sentido, *Quanteda* no solo facilita el tratamiento de grandes volúmenes de datos textuales, sino que contribuye a desarrollar competencias básicas en análisis de datos y pensamiento computacional, cada vez más necesarias en el ámbito de la investigación en documentación, comunicación y humanidades digitales.

## 6. Referencias

**Arcila-Calderón, Carlos; Barbosa-Caro, Eduar; Cabezuelo-Lorenzo, Francisco** (2016). Técnicas big data: Análisis de textos a gran escala para la investigación científica y periodística. *El Profesional de la Información*, 25(4), 623-631.

<https://doi.org/10.3145/epi.2016.jul.12>

**Arnold, Taylor; Ballier, Nicolas; Lissón, Paula; Tilton, Lauren** (2019). Beyond lexical frequencies: Using R for text analysis in the digital humanities. *Language Resources and Evaluation*, 53(4), 707-733.

<https://doi.org/10.1007/s10579-019-09456-6>

**Benoit, Kenneth; Watanabe, Kohei; Wang, Haiyan; Nulty, Paul; Obeng, Adam; Müller, Stefan; Matsuo, Akitaka** (2018). Quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30), 774.

<https://doi.org/10.21105/joss.00774>

**Berry, David M.** (2011). The computational turn: Thinking about the digital humanities. *Culture Machine*, 12.

<https://culturemachine.net/wp-content/uploads/2019/01/10-Computational-Turn-440-893-1-PB.pdf>

**Gallego-Cuiñas, Ana; Torres-Salinas, Daniel** (Eds.). (2024). *Humanities and Big Data in Ibero-America: Theory, methodology and practical applications*. De Gruyter.

<https://doi.org/10.1515/9783110753523>

**Grolemund, G.** (2014). *Hands-on programming with R: Write your own functions and simulations* (Primer). O'Reilly Media.

**Navalny, Alexéi** (2024). *Patriot: A memoir*. Alfred A. Knopf.

*R Core Team* (2021). *R: A Language and environment for statistical computing* (Version 4.1.2) [Software]. R Foundation for Statistical Computing.  
<https://www.r-project.org>

**Stone, Philip J.** (2020). Thematic text analysis: New agendas for analyzing text content. In C. W. Roberts (Ed.). *Text analysis for the social sciences* (1<sup>st</sup> ed., pp. 35-54). Routledge.  
<https://doi.org/10.4324/9781003064060-3>

**Wickham, H.; Grolemund, G.** (2017). *R for data science: Import, tidy, transform, visualize, and model data* (Primer). O'Reilly Media.