Fundamentos de ciencia de datos con **R**

Gema Fernández-Avilés y José-María Montero

2023-07-20

Índice general

Prefacio

¡Hola, mundo!

El siglo XXI está siendo testigo de grandes cambios vertiginosos en el contexto social y tecnológico, entre otros. Los tiempos han cambiado, la sociedad se ha globalizado y "exige" respuestas inmediatas a problemas muy complejos. Vivimos en el mundo de la **información**, de los **datos**, o mejor, de las **bases de datos masivas**, y los ciudadanos y, sobre todo, las empresas y los gobiernos, dirigen su mirada hacia el mundo científico para que les ayude a "**oír las historias**" que cuentan esos datos acerca de la realidad de la que han sido extraídos. Y dado su enorme volumen y sofisticación (en el nuevo mundo las imágenes y los textos, por ejemplo, también son datos), exigen algoritmos de nueva generación en el campo del *machine learning*, o incluso del *deep learning*, para "oír las historias" que cuentan. No parecen mirar al "antiguo" investigador científico, sino al "nuevo" *científico de datos*.

Ello, inevitablemente, se traduce en la necesidad de profesionales con una gran capacidad de adaptación a este nuevo paradigma: los científicos de datos, también llamados por algunos los "nuevos hombres del Renacimiento", para lo cual las universidades y demás instituciones educativas especializadas se apresuran a incluir el grado de Ciencia de Datos en su oferta educativa y a ofrecer seminarios de software estadístico de acceso abierto para sus estudiantes de primeros cursos.

Con la emergencia de la nueva sociedad, en la que el manejo de la ingente cantidad de información que genera se hace absolutamente necesario para circular por ella, la **ciencia de datos** ha venido para quedarse. Sin embargo, el mundo de la ciencia de datos es cualquier cosa menos sencillo. En él, cualquier ayuda, cualquier guía es bienvenida. Por ello, es muy recomendable que la persona que se quiera introducir en él, sea con fines de investigación o con fines profesionales, se agarre de la mano de un guía especializado que le lleve, de una manera amena, comprensible y eficiente, desde el planteamiento de su problema y la captura de la información necesaria para poderle dar una solución, hasta la redacción de las conclusiones finales que ha obtenido con los modernos informes reproducibles colaborativos. Y como en la parte central de ese camino tendrá que luchar con grandes gigantes (en la actualidad denominados técnicas estadísticas y algoritmos), el guía tendrá que explicarle, de manera sencilla y amena, en qué consiste la lucha (las técnicas y los algoritmos) y cómo llegar a la victoria lo más rápido posible, enseñándole a moverse por el mundo del software estadístico, en nuestro caso **R**, que le permitirá realizar los cálculos necesarios para vencer al problema planteado a una velocidad vertiginosa.

En resumen, la información masiva y el moderno tratamiento estadístico de la misma son la "mano invisible" que gobierna la sociedad del siglo XXI, y este manual pretende ser ese guía que le llevará de la mano cuando quiera caminar por ella.

¿Por qué este libro?

Lo dicho anteriormente ya justifica por sí solo la aparición de este manual. Afortunadamente, no es el primero en la materia, pues son ya bastantes los materiales de calidad publicados sobre ciencia de datos. Sin embargo, quizás, este pueda ser considerado el más completo. Y ello por varias razones.

La primera es su **completitud**: este manual lleva de la mano al lector desde el planteamiento del problema hasta el informe que contiene la solución al mismo; o desde no saber qué hacer con la información de la que dispone, hasta ser capaz de transformar tales bases de datos masivas, y casi imposibles de manejar, en respuestas a problemas fundamentales de una empresa, institución o cualquier agente social.

La segunda es su **amplitud temática**:

(I) Parte de las dos primeras preguntas que un neófito se puede hacer sobre esta temática: ¿qué es eso de la ciencia de datos que está en boca de todos? Y, ¿qué diablos es **R** y cómo funciona?

(II) Enseña cómo moverse en la jungla del *big data* y de los "nuevos" tipos de datos, siempre bajo el paraguas de la ética de los datos y del buen gobierno de dichos datos.

(III) Muestra al lector cómo obtener conocimiento de la oscuridad del enorme banco de información a su disposición, que no sabe cómo abordar ni manejar.

(IV) No deja a nadie atrás, y de forma previa al contenido central del manual (las técnicas de ciencia de datos), incluye unas breves, pero magníficas, secciones sobre los rudimentos de la probabilidad, la inferencia estadística y el muestreo, para aquellos no familiarizados con estas cuestiones.

(V) Aborda una treintena de técnicas de ciencia de datos en el ámbito de la modelización, análisis de datos cualitativos, discriminación, *machine learning* supervisado y no supervisado, con especial incidencia en las tareas de clasificación y clusterización –así como, en el caso no supervisado, de reducción de la dimensionalidad, escalamiento multidimensional y análisis de correspondencias–, *deep learning*, análisis de datos textuales y de redes, y, finalmente, ciencia de datos espaciales (desde las perspectivas de la geoestadística, la econometría espacial y los procesos de punto).

(VI) Hace especial hincapié en la reproducibilidad en tiempo real (o no) entre los distintos miembros de un equipo (sea universitario, empresarial o de otro tipo) y en la difusión de los resultados obtenidos, enseñando al lector cómo generar informes reproducibles mediante RMarkdown y documentos Quarto o en otros modernos formatos.

(VII) Dedica un capítulo a la creación de aplicaciones web interactivas (con Shiny).

(VIII) Para aquellos con pasión por la codificación, y que quieran compartir código y colaborar con otros desarrolladores, este manual aborda la gestión rápida y eficaz de proyectos (del tamaño que sean) mediante Git, un sistema de control de versiones distribuido, gratuito y de código abierto, y GitHub, un servicio de alojamiento de repositorios Git del cual, aquellos no familiarizados con la cuestión de la codificación, o con aversión a ella, podrán tomar el código que necesitan.

(IX) Muestra al lector los primeros pasos para iniciarse en el geoprocesamiento en la nube.

(X) Y, finalmente, aborda más de una docena de casos de uso (en medicina, periodismo, economía, criminología, marketing, moda, demanda de electricidad, cambio climático, reconocimiento de patrones en la forma de tuitear...) que ilustran la puesta en práctica de todos los conocimientos anteriormente adquiridos.

La tercera razón es que todo lo que el lector aprende en este manual lo puede reproducir y poner en práctica inmediatamente con **R**, puesto que el manual está lleno de *chunks* (o trozos de código **R**) que no tiene más que cortar y pegar para reproducir los ejemplos que se muestran en el libro, cuyos datos están en el paquete `CDR` ; o utilizar dichas *chunks* para abordar el problema que le ocupa con los datos que tenga a su disposición. Una buena razón, sin duda. Por consiguiente, el manual es una buena combinación "teoría-práctica-software" que permite abordar cualquier problema que el científico de datos se plantee en cualquier disciplina o situación empresarial, médica, periodística...

La cuarta es su **variedad de perspectivas**. Son **más de 40 los participantes** en este manual. Algunos de ellos, prestigiosos profesores universitarios; otros, destacados miembros de instituciones públicas; otros, CEO de empresas en la órbita de la ciencia de datos; otros, *big names* del mundo de **R** software... El manual es, sin duda, un magnífico ejemplo de colaboración Universidad-Empresa para buscar soluciones a los problemas de las sociedades modernas.

¿A quién va dirigido?

*Fundamentos de ciencia de datos con **R*** está dirigido a todos aquellos que desean desarrollar las habilidades necesarias para abordar proyectos complejos de ciencia de datos y "pensar con datos" (como lo acuñó Diane Lambert, de Google). El deseo de resolver problemas utilizando datos es su piedra angular. Por tanto, como se avanzó anteriormente, este manual no deja a nadie atrás, y lo único que requiere es "el deseo de resolver problemas utilizando datos". No excluye ninguna disciplina, no excluye a las personas que no tengan un elevado nivel de análisis estadístico de datos, no excluye a nadie. Se ha procurado una combinación de rigor y sencillez, y de teoría y práctica, todo ello con sus correspondientes códigos en **R**, que satisfaga tanto a los más exigentes como a los principiantes.

También está destinado a todos aquellos que quieran sustituir la navegación por la web (la búsqueda del vídeo, publicación de blog o tutorial *online* que solucione su problema –frustración tras frustración por la falta de consistencia, rigor e integridad de dichos materiales, así como por su sesgo hacia paquetes singulares para la implementación de las cuestiones que tratan–), por

una "**biblia de la ciencia de datos**" rigurosa pero sencilla, práctica y de aplicación inmediata sin ser ni un experto estadístico ni un experto informático.

Pero si a alguien está destinado especialmente, es a la comunidad hispanohablante. Este manual es un guiño a dicha comunidad, para que tenga a su disposición, en su lengua nativa, uno de los mejores manuales de diencia de datos de la actualidad.

El paquete `CDR`

El paquete `CDR` contiene la mayoría de conjuntos de datos utilizados en este libro que no están disponibles en otros paquetes. Para instalarlo use la función `install_github()` del paquete `remotes` .

```
# este comando solo necesita ser ejecutado una vez
# si el paquete remotes no está instalado, descomentar para instalarlo

# install.packages("remotes")
remotes::install_github("cdr-book/CDR")
```

La lista de todos los conjuntos de datos puede obtenerse haciendo `data()` .

```
library('CDR')
data(package = "CDR")
```

Este paquete ayudará al lector a reproducir todos los ejemplos del libro. De acuerdo con las mejores prácticas en **R**, el paquete `CDR` solo contiene los datos utilizados en el libro.

¿Por qué R?

R es un lenguaje de código abierto para computación estadística que se ha consolidado entre la comunidad científica internacional, en las últimas dos décadas, como una herramienta de

primer nivel, estableciéndose como líder permanente en el ámbito de la implementación de metodologías estadísticas para el análisis de datos. La utilidad de **R** para la ciencia de datos deriva de un fantástico ecosistema de paquetes (activo y en crecimiento), así como de un buen elenco de otros excelentes recursos: libros, manuales, blogs, foros y *chats* interactivos en las redes sociales, y una gran comunidad dispuesta a colaborar, a orientar y a resolver diferentes cuestiones relacionadas con **R**.

Por otra parte, **R** es el lenguaje estadístico y de análisis de datos más utilizado en la mayoría de los entornos académicos y, cómo no, por una larga lista de importantes empresas, entre las que se cuentan Facebook (análisis de patrones de comportamientos relacionado con actualizaciones de estado e imágenes de perfil), Google (para la efectividad de la publicidad y la previsión económica), Twitter (visualización de datos y agrupación semántica), Microsoft (adquirió la empresa Revolution R), Uber (análisis estadístico), Airbnb (ciencia de datos), IBM (se unió al grupo del consorcio R), New York Times (visualización)...

La comunidad **R** también es particularmente generosa e inclusiva, y hay grupos increíbles, como *R-Ladies* y *Minority R Users*, diseñados para ayudar a garantizar que todos aprendan y usen las capacidades de **R**.

Agradecimientos

No queremos dar por finalizado este prefacio sin agradecer a los 44 autores participantes en esta obra su esfuerzo por condensar, en no más de 20 páginas, la teoría, práctica y tratamiento informático de la parte de la ciencia de datos que les fue encargada. Y no solo eso; el "más difícil todavía" fue que debían dirigirse a un abanico de potenciales lectores tan grande como personas haya con "el deseo de resolver problemas utilizando datos". Era misión imposible. Sin embargo, a la vista del resultado, ha sido misión cumplida. El esfuerzo mereció la pena.

Además, nos gustaría agradecer el apoyo incondicional recibido por (en orden alfabético): Itzcoátl Bueno, Ismael Caballero, Emilio L. Cano, Diego Hernangómez, Michal M. Kinel, Ricardo Pérez, Manuel Vargas y Jorge Velasco.

También queremos poner de manifiesto que la edición de este texto ha sido financiada por diversos entes de la Universidad de Castilla-La Mancha. En su mayor parte, por el **Máster en *Data Science* y *Business Analytics* (con R software)** (a través de la orgánica: 02040M0280), pero también por la Facultad de Ciencias Jurídicas y Sociales de Toledo (a través de su contrato programa: orgánica 00440710), el Departamento de Economía Aplicada I (mediante sus fondos departamentales, DEAI 00421I126) y el Grupo de Investigación Economía Aplicada y Métodos Cuantitativos (que ha dedicado parte de sus fondos a la edición de esta obra, orgánica 01110G3044-2023-GRIN-34336).

A todos, eternamente agradecidos por ayudarnos en este reto de transformar la oscuridad en conocimiento, de convertir en una ciencia y en un arte la difícil tarea de sacar valor de los datos, el petróleo del futuro. Quizás en este momento no seamos conscientes de que hemos puesto nuestro granito de arena a la ciencia que, a buen seguro, juegue uno de los papeles más importantes de este siglo, caracterizado por el predominio de la información. Una ciencia, la ciencia de datos, que combina el análisis estadístico de datos, la algoritmia y el conocimiento del

negocio para sacar valor del bien más abundante de la sociedad en la que vivimos: la información. Una disciplina cuyo dominio caracteriza a los científicos de datos (también denominados los nuevos personajes del Renacimiento), profesión que ya fue calificada hace más de veinte años en la *Harvard Business Review* y en *The New York Times*, entre otros, como la "más *sexy* del siglo XXI".

Este manual está publicado por McGraw Hill. Las copias físicas están disponibles en McGraw Hill. La versión *online* se puede leer de forma gratuita en https://cdr-book.github.io/ y tiene la licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional. Si tiene algún comentario o sugerencia, no dude en contactar con los editores y los autores. ¡Gracias!

Parte I

Ciencia de datos de texto y redes

Capítulo 1

Minería de textos

Víctor Casero-Alonso[a], *Ángela Celis*[a] y *María Lozano Zahonero*[b]

[a]Universidad de Castilla-La Mancha
[b] Università degli Studi di Roma Tor Vergata

1.1. Introducción

En la actualidad, entre el 80 % y el 90 % de los datos que se generan diariamente son datos no estructurados (véase Cap. **??**). Un ejemplo típico de datos no estructurados son los textos, desde los comentarios o mensajes de las redes sociales, reseñas, blogs y microblogs, chats o whatsapp hasta las noticias periodísticas, los discursos políticos o las obras literarias. En consecuencia, aprender a procesar y analizar datos exige aprender a procesar y analizar textos.

Los textos precisan, sin embargo, un tratamiento especial. A diferencia de la mayoría de los datos que se tratan en este libro, que son datos estructurados, los datos textuales requieren que se les otorgue un orden y estructura para su manejo y análisis con el software **R**. Además, al utilizar un lenguaje natural –es decir, un idioma como, por ejemplo, el español, el chino o el inglés–, los textos no pueden ser procesados directamente por un ordenador. Es preciso "traducirlos" antes a un lenguaje formal que los ordenadores puedan entender.

La **minería de textos** (en inglés, *text mining*), también conocida como **análisis de textos** (en inglés, *text analysis*), puede definirse como el proceso para detectar, extraer, clasificar, analizar y visualizar la información no explícita que contienen los textos, transformando los datos textuales en datos estructurados y el lenguaje natural en lenguaje formal a fin de determinar, después, de manera automática, patrones recurrentes y desviaciones de los mismos. La minería de textos utiliza muchas técnicas y métodos diferentes, la mayoría procedentes del **procesamiento del lenguaje natural** (PLN), un ámbito de la inteligencia artificial que se ocupa de la comunicación entre los seres humanos y las máquinas mediante el tratamiento computacional del lenguaje humano.

Este capítulo constituye una primera aproximación a la minería de textos con **R**. Su objetivo es proporcionar un marco teórico y aplicado básico de este ámbito. Para ello, en la Sec. 1.2, se presentan los conceptos y fases fundamentales de la minería de textos. La Sec. 1.3 está dedicada al análisis de sentimientos, que constituye uno de los campos de la minería de textos de mayor desarrollo en la actualidad. La Sec. 1.4 se centra en algunos paquetes de **R** que permiten realizar análisis textuales de distintos tipos. Cierra el capítulo un ejemplo, en el que se aplica y se amplía lo estudiado anteriormente. Dos referencias útiles sobre el tema son Fradejas Rueda (2022) y Jockers (2014).

1.2. Conceptos y tareas fundamentales

Lo primero que se necesita para hacer un análisis de textos son los textos. Esta afirmación podría parecer banal, pero no lo es. El volumen de textos en circulación es ingente, pero, en la mayor parte de los casos, es necesario realizar una serie de operaciones complejas para poder extraer y recopilar los datos textuales que se quiere analizar. En muchas ocasiones también es difícil acceder después a estos datos, ya que los textos pueden presentar formatos muy heterogéneos, no siempre interpretables o fáciles de convertir en un formato interpretable. Baste pensar, por ejemplo, en una nota escrita a mano. Dado que este capítulo es una primera aproximación a la minería de textos, se parte del supuesto de que el texto o los textos están disponibles ya en un fichero, denominado **corpus**, legible por **R**. En este contexto, *corpus* es la colección de textos con el mismo origen: por ejemplo, el *corpus* de las obras de un autor, que para poder manejarse requiere metadatos con detalles adicionales.

1.2.1. Preparación de los datos

Una vez constituido el *corpus*, la primera fase es la **preparación de los datos**. Los textos suelen contener un cierto grado de "suciedad", es decir, elementos que alteran o impiden el análisis. La validez de los resultados que se obtengan dependerá, en gran parte, de una buena "limpieza" inicial. Entre las operaciones de "limpieza" generales figuran una serie de transformaciones cuya finalidad es evitar el recuento incorrecto de palabras, como el cambio de mayúsculas por minúsculas y la eliminación de los signos de puntuación, los números y los espacios en blanco en exceso.

La siguiente operación de preparación, que tiene un importante peso en el análisis, es la eliminación de las **palabras vacías** (en inglés, *stopwords*.) En la lengua no todas las palabras tienen el mismo tipo de significado. Las palabras con significado léxico, como *mesa* o *corpus*, son palabras a las que corresponde un concepto que se puede definir o explicar. Otras palabras, sin embargo, son palabras funcionales, cuyo contenido es puramente gramatical. Son palabras como el artículo *el*, la preposición *de* o la conjunción *o*: se puede explicar cómo se usan, pero no definirlas asociándolas a un concepto porque carecen de contenido léxico-semántico.

Las palabras vacías son, con gran diferencia respecto de las palabras léxicas, las más frecuentes de la lengua, pero, dado su escaso o nulo significado léxico, en los análisis de tipo semántico, como el análisis de sentimientos o el modelado de temas, carecen de valor informativo, por lo que es conveniente eliminarlas. No es aconsejable eliminarlas, sin embargo, en otros tipos de

análisis, como los análisis estilométricos, donde tienen un importante valor informativo como se verá en la Sec. 1.2.4. Las palabras vacías pertenecen a clases cerradas, es decir, a clases de palabras con un número de elementos limitado, finito. Es posible confeccionar, por tanto, listas de palabras vacías para facilitar su eliminación. En el ejemplo de aplicación que se verá en la Sec. 1.5, se aprenderá a usar estas listas y se podrá apreciar con detalle la diferente información que proporciona una tabla de frecuencias con y sin palabras vacías.

1.2.2. Segmentación del texto: tokenización

La segunda fase de la minería de textos consiste en la **segmentación del texto**, denominada también **tokenización**. El texto se divide en *tokens*, secuencias de texto con valor informativo. De esta manera, se pasa del lenguaje natural a un lenguaje formal comprensible por el software, dándole formato de vector o tabla. Así se pueden aplicar algunas de las herramientas que se utilizan con datos numéricos para manejar el texto y obtener resúmenes y visualizaciones que muestren la información no explícita contenida en él en forma de patrones recurrentes.

Generalmente, los *tokens* son **palabras**, es decir, secuencias de caracteres entre dos espacios en blanco y/o signos de puntuación, pero pueden ser también **oraciones**, **líneas**, **párrafos** o **n-gramas**. Como se verá en el ejemplo de aplicación, un primer análisis del significado consiste en eliminar las palabras vacías y obtener las frecuencias[1] de las palabras con valor informativo para responder a la pregunta "¿Qué se dice?" (Silge and Robinson, 2017).

Nota
También puede ser útil obtener la **tasa de riqueza léxica** (TTR, del inglés *type-token ratio*), que mide la relación entre el número de palabras diferentes que contiene un texto (*types*) y el número de palabras totales de dicho texto (*tokens*)[a].

$$TTR = \frac{Types}{Tokens}$$

[a] Véase https://www.fundeu.es/consideraciones-teoricas/

1.2.2.1. *N*-gramas

El análisis puede proseguir estudiando la frecuencia de los *n-gramas*, secuencias de n palabras consecutivas en el mismo orden. Se tienen así bigramas o 2-gramas (secuencias de dos palabras), trigramas o 3-gramas (secuencias de tres palabras), etc. El estudio de los *n-gramas* responde al principio de Firth: *"You shall know a word by the company it keeps"* (Firth, 1957, 11). Este principio es el fundamento del llamado **análisis de colocaciones**: para conocer el significado de una palabra es preciso conocer las palabras con las que aparece, el contexto relevante. En un sentido amplio, el análisis de colocaciones consiste en examinar los contextos izquierdo y/o derecho de una palabra. La segmentación en *n-gramas* permite tener en cuenta este contexto relevante que indicará, por ejemplo, que *banco* es, con toda probabilidad, un asiento en las

[1] Frecuencias relativas si se comparan distintos textos.

secuencias *banco de madera* o *banco en la terraza*, pero no lo es en secuencias como *banco de peces*, *banco de arena*, *banco de inversiones*, *banco de datos* o *banco de pruebas*. La división en *n-gramas* permitirá también considerar en el análisis, al menos hasta cierto punto, el peso de la ambigüedad, la negación o el distinto significado que pueden tener las palabras según el ámbito temático. Por ejemplo, la forma *larga* no tiene el mismo significado en los bigramas *falda larga*, *mano larga* y *cara larga*, ni tiene tampoco el mismo valor informativo en *es larga* / *no es larga* o en *de larga experiencia* (valor positivo) y en *se me hizo larga* (valor negativo). En el ejemplo de aplicación (Sec. 1.5), se verá la segmentación en *n-gramas* en la práctica, y cómo la visualización de redes contribuye a complementar el análisis.

1.2.3. *Stemming* y lematización

La tokenización se puede refinar mediante el ***stemming***, o reducción de las palabras "flexionadas" a su raíz, y la **lematización**, o extracción del lema de cada palabra. Un ejemplo de *stemming* sería reducir las palabras *texto*, *textos*, *textual* y *textuales*, que **R** cuenta como cuatro palabras diferentes, a la raíz "text". El *stemming* puede proporcionar un recuento más preciso en algunos casos, pero en otros, al eliminar los sufijos de las palabras, puede crear confusión. Además, como en el ejemplo anterior, las raíces pueden no coincidir con palabras existentes, lo que hace que sean difíciles de interpretar y resulten extrañas si se visualizan en nubes de palabras. Con la lematización se reducen las formas flexionadas de una misma palabra al lema, que es la forma que encabeza la entrada de la palabra en el diccionario. Por ejemplo, si se quiere buscar el significado de la palabra *niñas* no se encontrará como tal sino bajo el lema *niño* y si se quiere buscar *iremos* se tendrá que buscar el lema *ir*. En el caso anterior, la lematización reduciría las formas *texto*, *textos*, *textual* y *textuales* a dos lemas: *texto* y *textual*. La lematización evita la dispersión de significado en varias formas, pero a veces es compleja y puede conducir a la pérdida de información pertinente.

1.2.4. Campos de aplicación de la minería de textos

La minería de textos tiene varios campos de aplicación. Entre ellos destacan tres:

1. El **análisis de sentimientos**, que se tratará con detalle en la Sec. 1.3 y en el ejemplo de aplicación (Sec. 1.5.4).

2. El **modelado de temas** o **tópicos** (en inglés, *topic modelling*), que, como su propio nombre indica, tiene por objeto identificar los temas principales sobre los que versa el texto haciendo uso de técnicas de clasificación no supervisada del campo del aprendizaje automático, como por ejemplo LDA (*latent Dirichlet allocation*). Se ilustrará en el Cap. **??**.

3. La **estilometría** o **análisis estilométrico**, que es una aplicación de la minería de textos cuya finalidad consiste en determinar las relaciones existentes entre el estilo de los textos y los metadatos incluidos en ellos. Se utiliza principalmente en la atribución de autoría. El concepto base es el de **huella lingüística**, constituida por el conjunto de rasgos lingüísticos que caracterizan el estilo de un autor como un estilo individual y único y permiten

identificarlo. Un punto clave es que, contrariamente a lo que podría pensarse, los rasgos que conforman en mayor medida la huella lingüística son los que tienen un mayor índice de frecuencia. La mayor parte de los enfoques utilizan el vector de las "palabras más frecuentes" (MFW, por sus siglas en inglés), que son, como se ha visto antes, las palabras vacías y no las palabras con significado léxico, para determinar el estilo de un autor. Esto es debido fundamentalmente a que las palabras vacías se usan de manera involuntaria e inconsciente, configurando de esta manera, sin ningún tipo de filtros racionales, una clave estilística idiosincrásica (Lozano Zahonero, 2020). De lo anterior se deduce fácilmente que en este tipo de análisis no deben eliminarse las palabras vacías.

En la actualidad, el análisis estilométrico se usa en ámbitos muy dispares: desde la criminología o los servicios de inteligencia para identificar a los autores de mensajes o notas en casos de asesinatos, terrorismo, secuestro o acoso, por ejemplo, hasta el derecho civil o la literatura en cuestiones de derechos de autor o detección de plagio, entre muchas otras cuestiones.

1.3. Análisis de sentimientos

El **análisis de sentimientos** (en inglés, *sentiment analysis*) es una aplicación de la minería de textos que tiene como finalidad la detección, extracción, clasificación, análisis y visualización de la dimensión subjetiva asociada a los temas o tópicos presentes en los textos. La dimensión subjetiva comprende no sólo los sentimientos, sino también las **emociones**, sensaciones y estados afectivos y anímicos, así como las opiniones, creencias, percepciones, puntos de vista, actitudes, juicios y valoraciones. De ahí que reciba también el nombre de **minería de opinión** (en inglés, *opinion mining*) (Lozano Zahonero, 2020).

El análisis de sentimientos asigna a esta dimensión subjetiva una polaridad, que puede ser positiva o negativa (Pang and Lee, 2008). Algunas técnicas añaden además una polaridad neutra. En algunos casos, el análisis de sentimientos se refina hasta llegar a las emociones básicas: este subcampo del análisis de sentimientos se conoce como **detección de emociones**.

La primera aplicación del análisis de sentimientos fue la investigación de mercados. A partir del año 2000, se registra un crecimiento exponencial de textos como reseñas, chats, foros, blogs, microblogs o comentarios y mensajes de las redes sociales, en los que predomina la expresión de emociones y opiniones personales. Mediante el análisis de sentimientos se extrae de ellos información que permite conocer los gustos del consumidor y diseñar productos a su medida. Esta idea se extendió después a otros ámbitos, en especial a aquellos en los que predomina la comunicación persuasiva, como las campañas publicitarias o políticas. Recientemente, ha empezado a utilizarse también con fines predictivos y preventivos en muchas esferas: desde cuáles son los políticos, las empresas, las películas, canciones u obras literarias que obtendrán un mayor rendimiento, mejores resultados o más votos o ventas hasta cómo detectar y prevenir, por ejemplo, conductas suicidas mediante el análisis de mensajes en las redes sociales.

En el análisis de sentimientos y la detección de emociones existen dos enfoques principales: el enfoque basado en el aprendizaje automático (*machine learning*), en el que se usan algoritmos de aprendizaje supervisado, y el enfoque semántico, basado en diccionarios o **lexicones**. Este último enfoque es el que se verá en detalle en el ejemplo de aplicación.

En **R** están implementados varios lexicones para el análisis de sentimientos. Dos de los más utilizados son `bing` , de Bing Liu y colaboradores (Liu, 2015), y `NRC` , de Saif Mohammad y Peter Turney, ambos incluidos tanto en el paquete `tidytext` como en `syuzhet` (Jockers, 2017). Estos lexicones tienen en común que están basados en unigramas, es decir, en palabras sueltas, y que tienen como idioma original el inglés, si bien hay versiones traducidas automáticamente a distintas lenguas. La diferencia principal entre los dos lexicones es que `bing` clasifica las palabras de forma binaria en polaridad positiva/negativa, mientras que `NRC` , además de la polaridad positiva/negativa, permite detectar también ocho emociones básicas (*ira, miedo, anticipación, confianza, sorpresa, tristeza, alegría y asco*). En el ejemplo de aplicación se compararán ambos diccionarios. Como se verá, los resultados del análisis dependerán en buena medida del lexicón elegido, así como del idioma del texto y de si el lexicón se elaboró originalmente en ese idioma o es una versión traducida automáticamente de otra lengua.

1.4. Minería de textos en R

En **R** existen diversos paquetes y funciones que facilitan la minería de textos, entre los que destacan:

- `tidytext` : con la filosofía del `tidyverse` , puede combinarse con los conocidos paquetes `dplyr` , `broom` , `ggplot2` , etc. Se puede destacar la función `unnest_tokens()` , que automatiza el proceso de *tokenización* y el almacenamiento en formato *tidy* en un único paso.
- `tm` : destaca por tener soporte *back-end* de base de datos integrada, gestión avanzada de metadatos y soporte nativo para leer en varios formatos de archivo.
- `tokenizers` : incluye *tokenizadores* de palabras, oraciones, párrafos, *n*-gramas, *tweets*, expresiones regulares, así como funciones para contar caracteres, palabras y oraciones, y para dividir textos más largos en documentos separados, cada uno con el mismo número de palabras.
- `wordcloud` : permite visualizar **nubes de palabras**. Las palabras más frecuentes aparecen en mayor tamaño permitiendo, de un vistazo, conocer las palabras clave del texto.
- `quanteda` : maneja **matrices de documentos-términos** y destaca en tareas cuantitativas, como el recuento de palabras o sílabas.
- `syuzhet` : incluye distintas funciones que facilitan el análisis de textos, en particular el *análisis de sentimientos* de textos literarios.
- `gutenbergr` : almacena las obras del proyecto Gutenberg[2]; muy útil si se quieren analizar textos literarios.

[2]Proyecto desarrollado por Michael Hart en 1971 para crear una biblioteca de libros electrónicos gratuitos, y accesibles en internet, a partir de libros en soporte físico, generalmente de dominio público. Cuenta con más de 50000 libros.]

1.5. Ejemplo de aplicación

1.5.1. Declaración institucional del Estado de Alarma 2020

La "Declaración institucional del presidente del Gobierno anunciando el Estado de Alarma en la crisis del coronavirus" (en adelante, "la Declaración"), pronunciada en La Moncloa el 13 de marzo de 2020 es el objeto de análisis. Esta se puede encontrar en el paquete `CDR` que acompaña este libro. Se le van a aplicar las operaciones y técnicas mencionadas en la Sec. 1.2.

```
library("CDR")
data("declaracion")
```

1.5.2. Segmentación en palabras y oraciones

Las primeras tareas del análisis son la preparación, limpieza y segmentación o tokenización de los textos, como se vió en las Sec. 1.2.1 y 1.2.2. A continuación, se verá una segmentación en palabras individuales. La función `tokenize_words()` del paquete `tokenizers` prepara el texto convirtiéndolo a minúsculas, elimina todos los signos de puntuación y finalmente segmenta el texto en palabras.

```
library("tokenizers")
palabras <- tokenize_words(declaracion)
tokenizers::count_words(declaracion)
#> [1] 922
```

Con la última sentencia se obtiene la longitud de la Declaración, es decir, el número de palabras utilizadas: 922.

La frecuencia de cada palabra se puede obtener y presentar con el código de abajo. La primera sentencia crea la tabla de frecuencias; la tercera la transforma en el tipo `tibble`, creando la columna recuento, y ordena la tabla de forma descendente, de mayor a menor frecuencia.

```
library("tidyverse")
tabla <- table(palabras[[1]])
( tabla <- tibble(palabra = names(tabla),
                  recuento = as.numeric(tabla)) |>
    arrange(desc(recuento)) )
#> # A tibble: 390 x 2
#>    palabra recuento
#>    <chr>      <dbl>
#>  1 de            43
#>  2 y             41
#>  3 la            35
#>  4 a             31
#>  5 los           26
```

```
#>  6 en            22
#>  7 que           20
#>  8 el            17
#>  9 al            14
#> 10 para          14
#> # i 380 more rows
```

En la primera fila de la salida se indican las dimensiones de la `tibble`, por lo que se puede ver que en la Declaración hay 390 "palabras" distintas (los números se consideran como palabras).

El resultado son las palabras más utilizadas en el texto, que, como puede apreciarse, son palabras vacías. Esto no debería sorprender porque, como ya se ha visto, estas palabras son las más frecuentes. En la siguiente Sección se verá cómo eliminarlas para obtener datos con valor informativo.

Para otras formas de segmentar el texto (oraciones, párrafos, *tweets*, etc.) véase `?tokenize_words`. Por ejemplo, para segmentar en oraciones:

```
oraciones <- tokenize_sentences(declaracion)
count_sentences(declaracion)
#> [1] 44
```

Las tres primeras oraciones y la última se obtienen con el siguiente código.

```
oraciones[[1]][1:3] # primeras 3 oraciones
#> [1] "Buenas tardes."
↪
#> [2] "Estimados compatriotas."
↪
#> [3] "En el día de hoy, acabo de comunicar al Jefe del Estado la celebración, mañana,
↪  de un Consejo de Ministros extraordinario, para decretar el Estado de Alarma en
↪  todo nuestro país, en toda España, durante los próximos 15 días."
oraciones[[1]][count_sentences(declaracion)] # última oración
#> [1] "Buenas tardes."
```

También podría medirse la longitud de cada oración, en número de palabras, normalmente para comparaciones con otros textos. Para ello hay que separar cada oración en palabras y obtener la longitud de cada oración, con la función `sapply()`, que puede verse en la Fig. 1.1.

```
palabras_oracion <- tokenize_words(oraciones[[1]])
longitud_o <- sapply(palabras_oracion, length)
head(longitud_o)
#> [1]  2  2 39 33 33 32
```

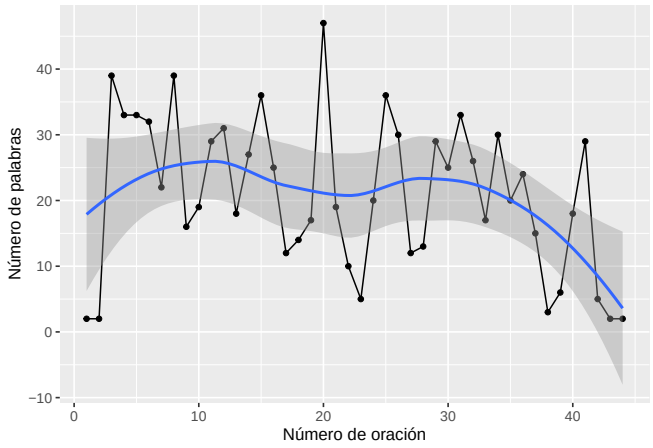

Figura 1.1: Número de palabras en cada oración de la Declaración.

1.5.3. Análisis exploratorio

1.5.3.1. Eliminación de palabras vacías

Se lleva a cabo con el paquete `stopwords`, que contiene listas de *palabras vacías* en diferentes idiomas. Para el ejemplo, se define una tabla con la misma estructura que la tabla de la Declaración con las 308 palabras vacías españolas que tiene el paquete:

```
library("stopwords")
tabla_stopwords <- tibble(palabra = stopwords("es"))
```

La siguiente sentencia 'limpia' la tabla de la Declaración quitando las palabras vacías españolas. Además, se hace uso de la función `kable()` para una visualización más sofisticada de la tabla (con la longitud que se desee):

```
tabla <- tabla  |> anti_join(tabla_stopwords)
knitr::kable(tabla[1:10,],
             caption = "Palabras más frecuentes (sin palabras vacías)")
```

El resultado, Tabla 1.1, se puede considerar el primer análisis léxico con valor informativo: la palabra más frecuente es *virus*, seguida de *recursos* y *social*. Se podría ver que en total hay 319 palabras no vacías distintas.

El método de eliminar palabras con el paquete `stopwords` no es perfecto. Por ejemplo, *va* y *cada* (posiciones 9 y 10 de la tabla) no son muy informativas. En estos casos, como se ha visto antes, se pueden utilizar listas de palabras vacías de otros paquetes como, por ejemplo, `tidytext` o `tokenizers` o el listado en español propuesto por Fradejas Rueda (2022), o pueden confeccionarse listas *ad hoc*.

Tabla 1.1: Palabras más frecuentes (sin palabras vacías)

palabra	recuento
virus	9
recursos	7
social	5
alarma	4
conjunto	4
emergencia	4
españa	4
semanas	4
va	4
cada	3

1.5.3.2. Nubes de palabras

Una manera habitual de mostrar la información de forma visual es con las denominadas **nubes de palabras**, acudiendo a la función `wordcloud()` del paquete con el mismo nombre. Como esta función contine un componente aleatorio, se fija con `set.seed()` (para la reproducibilidad del gráfico por parte del lector).

```
set.seed(12)
library("wordcloud")
wordcloud(tabla$palabra, tabla$recuento,
          max.words = 50, colors = rainbow(3))
```

Figura 1.2: Nube de palabras más frecuentes de la Declaración.

El resultado se muestra en la Fig. 1.2. Como se puede observar, el tamaño de letra de la palabra,

y en este caso también el color, están relacionados con su frecuencia.

1.5.4. Análisis de sentimientos y detección de emociones

1.5.4.1. Lexicón `bing`

Como se ha visto en la Sec. 1.3, el lexicón `bing`, es uno de los repertorios léxicos que se pueden encontrar en **R** para el análisis de sentimientos. \index{análisis!de sentimientos. Es un diccionario de polaridad (positiva/negativa) cuyo idioma original es el inglés. Se puede obtener con la función `get_sentiments()` del paquete `tidytext`. Contiene 2005 palabras positivas y 4781 palabras negativas, por lo que tiene un marcado sesgo hacia la polaridad negativa.

Para ilustrar el uso de `bing`, se ha traducido al inglés (automáticamente) la Declaración. A continuación se carga el texto y se genera el objeto `tabla`, replicando el procedimiento descrito anteriormente de preparación, limpieza, segmentación en palabras y eliminación de palabras vacías (obviamente, en idioma inglés).

```
data("EN_declaracion")
tabla <- table(tokenize_words(EN_declaracion)[[1]])
tabla <- tibble(word = names(tabla),
                recuento = as.numeric(tabla))
tabla <- tabla |> anti_join(tibble(word=stopwords("en"))) |>
  arrange(desc(recuento))
```

Los sentimientos positivos de la Declaración se obtienen con:

```
library("tidytext")
pos <- get_sentiments("bing") |>
  dplyr::filter(sentiment=="positive")
pos_EN <- tabla |> semi_join(pos)
knitr::kable(pos_EN)
```

Análogamente se pueden obtener los sentimientos negativos. Las siete palabras más frecuentes de cada tipo que aparecen en la Declaración se presentan conjuntamente en la Tabla 1.2.

1.5.4.2. Lexicón NRC

Para poder observar las similitudes y diferencias en el análisis según el lexicón elegido, se aplica también `NRC` a la Declaración (véase la Tabla 1.2).

Tabla 1.2: Palabras más frecuentes de la Declaración utilizando `bing` y `NRC`

positivas bing	fr	negativas bing	fr	positivas NRC	fr	negativas NRC	fr
extraordinary	6	virus	9	resources	7	virus	9
protect	4	alarm	4	extraordinary	6	alarm	4
work	4	emergency	4	protect	4	emergency	4
like	3	vulnerable	3	maximum	3	government	3
decisive	2	difficult	2	public	3	discipline	2
good	2	hard	2	council	2	avoid	1
adequate	1	unfortunately	2	good	2	combat	1

Con el léxico `NRC` pueden detectarse emociones. La misma palabra puede tener asociada distintas emociones/sentimientos. En la Fig. 1.3 se puede observar la dispar frecuencia de palabras de cada tipo:

```
emo <- get_sentiments("nrc")
emo |> ggplot(aes(sentiment)) +
  geom_bar(aes(fill=sentiment), show.legend = FALSE)
```

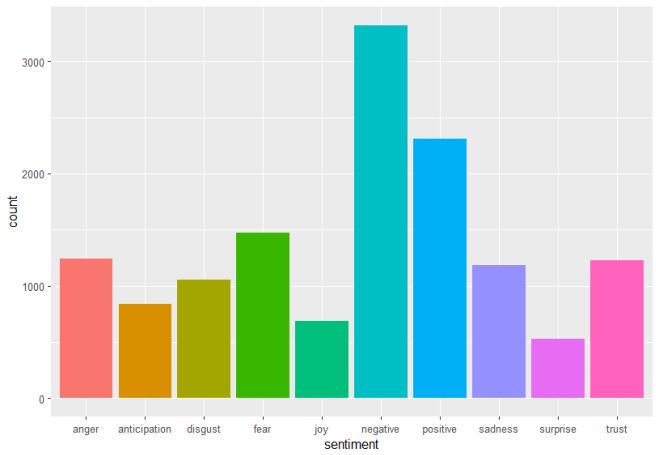

Figura 1.3: Gráfico de barras con la frecuencia de las emociones del lexicón NRC.

El análisis de sentimientos y la detección de emociones de la Declaración mediante NRC se puede realizar con el siguiente código, mediante el cual se obtiene la tabla de frecuencias por emociones y sentimientos:

```
emo_tab <- tabla |> inner_join(emo)
head(emo_tab, n=7)
#> # A tibble: 7 x 3
#> word recuento sentiment
#> <chr> <dbl> <chr>
#> 1 virus 9 negative
#> 2 resources 7 joy
#> 3 resources 7 positive
#> 4 resources 7 trust
#> 5 extraordinary 6 positive
#> 6 alarm 4 fear
#> 7 alarm 4 negative
```

Como se ha mencionado anteriormente, algunas palabras tienen asociados distintos sentimientos; por ejemplo, *resources*. La información de la tabla se puede visualizar con un gráfico de barras (Fig. 1.4) o con una nube de palabras (Fig. 1.5).

```
emo_tab |>
  dplyr::count(sentiment) |>
  ggplot(aes(x=sentiment, y=n)) +
  geom_bar(stat = "identity", aes(fill=sentiment), show.legend = FALSE) +
  geom_text(aes(label = n), vjust=-0.25)
```

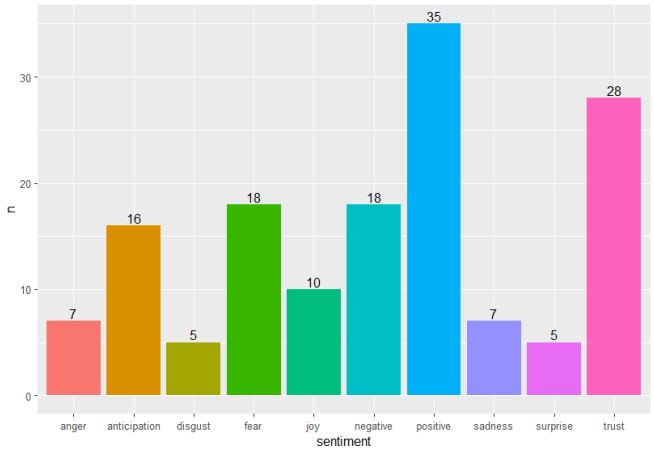

Figura 1.4: Frecuencia de emociones de la Declaración utilizando NRC.

Entre las distintas opciones para dibujar nubes de palabras para el análisis de sentimientos es interesante la que se obtiene con el paquete `syuzhet`, dado que permite visualizar las palabras agrupadas por emociones. Su obtención requiere distintos pasos en los que primero las palabras se agrupan por emoción y después se organizan en una **matriz de documentos** con la función `TermDocumentMatrix()` del paquete `tm`. Finalmente, la función `comparison.cloud()`

permite visualizar el gráfico (tiene distintos argumentos opcionales que admiten distintas posibilidades). En el ejemplo que figura a continuación sólo se han escogido tres emociones:[3].

```
library("syuzhet")
palabras_EN2 <- get_tokens(EN_declaracion)
emo_tab2 <- get_nrc_sentiment(palabras_EN2, lang = "english" )
emo_vec <- c(
  paste(palabras_EN2[emo_tab2$anger> 0], collapse = " "),
  paste(palabras_EN2[emo_tab2$anticipation > 0], collapse = " "),
  paste(palabras_EN2[emo_tab2$disgust > 0], collapse = " "))
library("tm")
corpus <- Corpus(VectorSource(emo_vec))
TDM <- as.matrix(TermDocumentMatrix(corpus))
colnames(TDM) <- c('anger', 'anticipation', 'disgust')
set.seed(1)
comparison.cloud(TDM, random.order = FALSE,
                 colors = c("firebrick", "forestgreen", "orange3"),
                 title.size = 1.5, scale = c(3.5, 1), rot.per = 0)
```

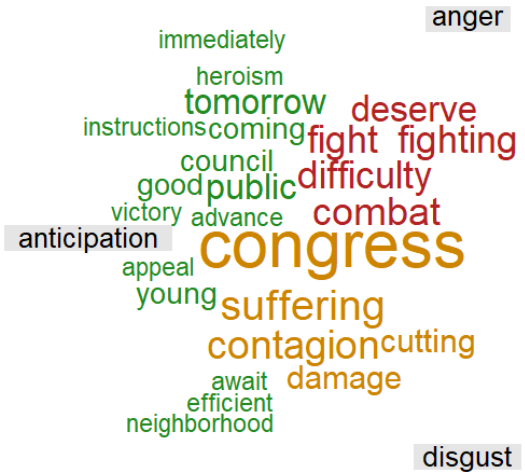

Figura 1.5: Nube de palabras de tres emociones NRC seleccionadas.

1.5.5. *N-gramas*

El siguiente código muestra la obtención de *n-gramas* con `tokenizers` :

[3]Se deja al lector el análisis de la Declaración con más emociones, en castellano, etc.

```
bigramas <- tokenize_ngrams(declaracion, n = 2,
                            stopwords = tabla_stopwords$palabra)
head(bigramas[[1]], n = 3)
#> [1] "buenas tardes"          "tardes estimados"       "estimados compatriotas"
trigramas <- tokenize_ngrams(declaracion, n = 3,
                             stopwords = tabla_stopwords$palabra)
head(trigramas[[1]], n = 3)
#> [1] "buenas tardes estimados"       "tardes estimados compatriotas"
#> [3] "estimados compatriotas día"
```

Se han eliminado de los bigramas y trigramas aquellas combinaciones con al menos una palabra vacía (*stopword*).

Se procede ahora a obtener los bigramas con `tidytext` . Para el resto de *n-gramas* el procedimiento es análogo, haciendo las modificaciones oportunas. En el último paso se ordenan por frecuencia (de mayor a menor):

```
declara2 <- tibble(texto = declaracion)
bigramas <- declara2 |>
  unnest_tokens(bigram, texto, token = "ngrams", n = 2) |>
  dplyr::count(bigram, sort = TRUE)
bigramas[1:5, ]
#> # A tibble: 5 x 2
#>   bigram         n
#>   <chr>      <int>
#> 1 todos los      6
#> 2 de la          5
#> 3 de los         5
#> 4 del estado     5
#> 5 estado de      5
```

Una forma de eliminar las palabras vacías es:

```
bigramas_limpios <- bigramas |>
  tidyr::separate(bigram, c("word1", "word2"), sep = " ") |>
  dplyr::filter(!word1 %in% tabla_stopwords$palabra) |>
  dplyr::filter(!word2 %in% tabla_stopwords$palabra) |>
  tidyr::unite(bigram, word1, word2, sep = " ")
bigramas_limpios[1:5, ]
#> # A tibble: 5 x 2
#>   bigram                     n
#>   <chr>                  <int>
#> 1 autoridades sanitarias     2
#> 2 buenas tardes              2
#> 3 disciplina social          2
#> 4 haga falta                 2
#> 5 ministros extraordinario   2
```

1.5.5.1. Significado y contexto

Como se ha visto en la Sec. 1.2.2, con los *n-gramas* se puede hacer un análisis de colocaciones para extraer los distintos significados y valores informativos a partir del contexto. En este caso, se puede ver cómo la palabra *atender* cambia de sentido cuando va precedida de *no* o *sin*. A continuación, se filtran los bigramas cuya primera palabra es *no*:

```
bigramas_no <- bigramas |>
  tidyr::separate(bigram, c("word1", "word2"), sep = " ") |>
  dplyr::filter(word1 == "no") |>
  dplyr::count(word1, word2, sort = TRUE)
bigramas_no
#> # A tibble: 3 x 3
#>   word1 word2       n
#>   <chr> <chr>   <int>
#> 1 no    atiende     1
#> 2 no    cabe        1
#> 3 no    es          1
```

Estos resultados se pueden utilizar para el análisis de sentimientos y la detección de emociones.

1.5.6. Análisis de redes

En esta Sección se proporcionan las instrucciones necesarias para realizar un **análisis** básico **de redes** (véase Cap. **??**) utilizando los paquetes `igraph` y `ggraph`. Dada la corta extensión de la Declaración no es posible obtener conclusiones. En la Fig. 1.6 se pueden ver los gráficos de redes de bigramas, tanto sin palabras vacías como con ellas.

```
library("igraph")
library("ggraph")
set.seed(1)
graf_bigramas_l <- bigramas_limpios |>
  tidyr::separate(bigram, c("first", "second"), sep = " ") |>
  dplyr::filter(n > 1) |>
  graph_from_data_frame()
g1 <- ggraph(graf_bigramas_l, layout = "fr") +
  geom_edge_link(arrow = arrow(length = unit(4, 'mm'))) +
  geom_node_point(size=0) +
  geom_node_text(aes(label = name))
graf_bigramas <- bigramas |>
  tidyr::separate(bigram, c("first", "second"), sep = " ") |>
  dplyr::filter(n > 2) |>
  graph_from_data_frame()
g2 <- ggraph(graf_bigramas, layout = 'fr') +
  geom_edge_link0() +
  geom_node_point(size=0) +
  geom_node_label(aes(label = name))
```

```
library("patchwork")
g1+g2
```

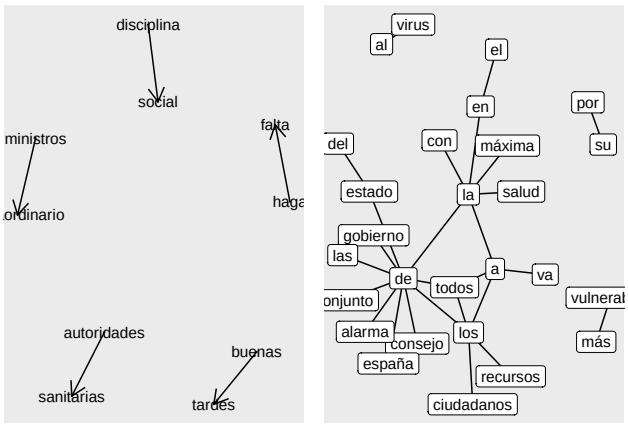

Figura 1.6: Redes de bigramas sin palabras vacías y con ellas.

Resumen

En este capítulo se introduce al lector en la minería de textos. En particular:

- Se presentan los conceptos y tareas fundamentales de este ámbito, así como sus principales campos de aplicación.
- Se pone de relieve la importancia de la preparación de los datos y su segmentación (a distintos niveles) para obtener buenos resultados, acordes con el objetivo de la investigación.
- Se muestra el uso de **R** para el análisis de textos y de sentimientos.
- Se presenta un ejemplo de aplicación para ilustrar las técnicas de minería de textos.
- Se mencionan otros análisis plausibles de minería de textos, como la estilometría o el modelado de temas (véase el Cap. **??**).

Bibliografía

Firth, J. (1957). A synopsis of linguistic theory, 1930–1955. *En Selected Papers of J.R. Firth 1952–1959, ed. Frank Palmer, 168-205. Londres: Longman.*

Fradejas Rueda, J. M. (2022). *Cuentapalabras. Estilometría y análisis de datos con R para filólogos.* http://www.aic.uva.es/cuentapalabras/.

Jockers, M. (2014). *Text analysis with R for students of literature.* Nueva York: Springer.

Jockers, M. (2017). Introduction to the syuzhet package. https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions.* Cambridge University Press.

Lozano Zahonero, M. (2020). Una nueva visión de la supuesta influencia de *Madame Bovary* en *La Regenta* a través de la estilometría y el análisis de sentimientos basados en lenguaje R. *Orillas: rivista d'ispanistica*, 9:573–607.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.

Silge, J. and Robinson, D. (2017). *Text Mining with R: A Tidy Approach.* O'Reilly Media, Inc. https://www.tidytextmining.com.

Índice alfabético