Recuperación y análisis de texto con R

Clase 3 - Educación Permanente FCS

Mag. Elina Gómez (UMAD)

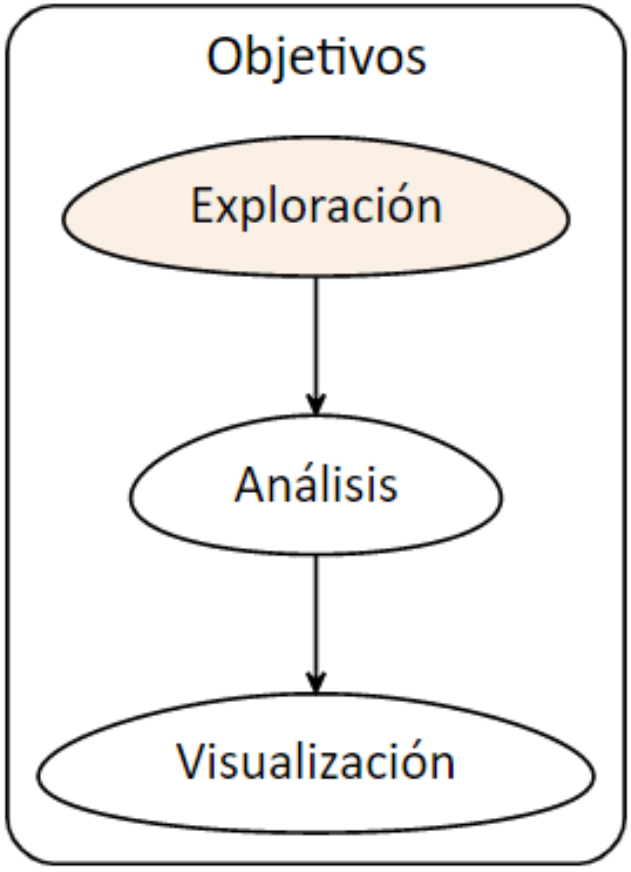
elina.gomez@cienciassociales.edu.uy

www.elinagomez.com

Mag. Gustavo Méndez Barbato

gustavo.mendez@cienciassociales.edu.uy

Objetivos de hoy



Fuentes de datos

Las fuentes de datos que vamos a ver son:

1. Recuperación de documentos en imagen o pdf (OCR)
2. Scraping web y parlamentario
3. Prensa digital
4. Google Trends
5. Audio
6. YouTube

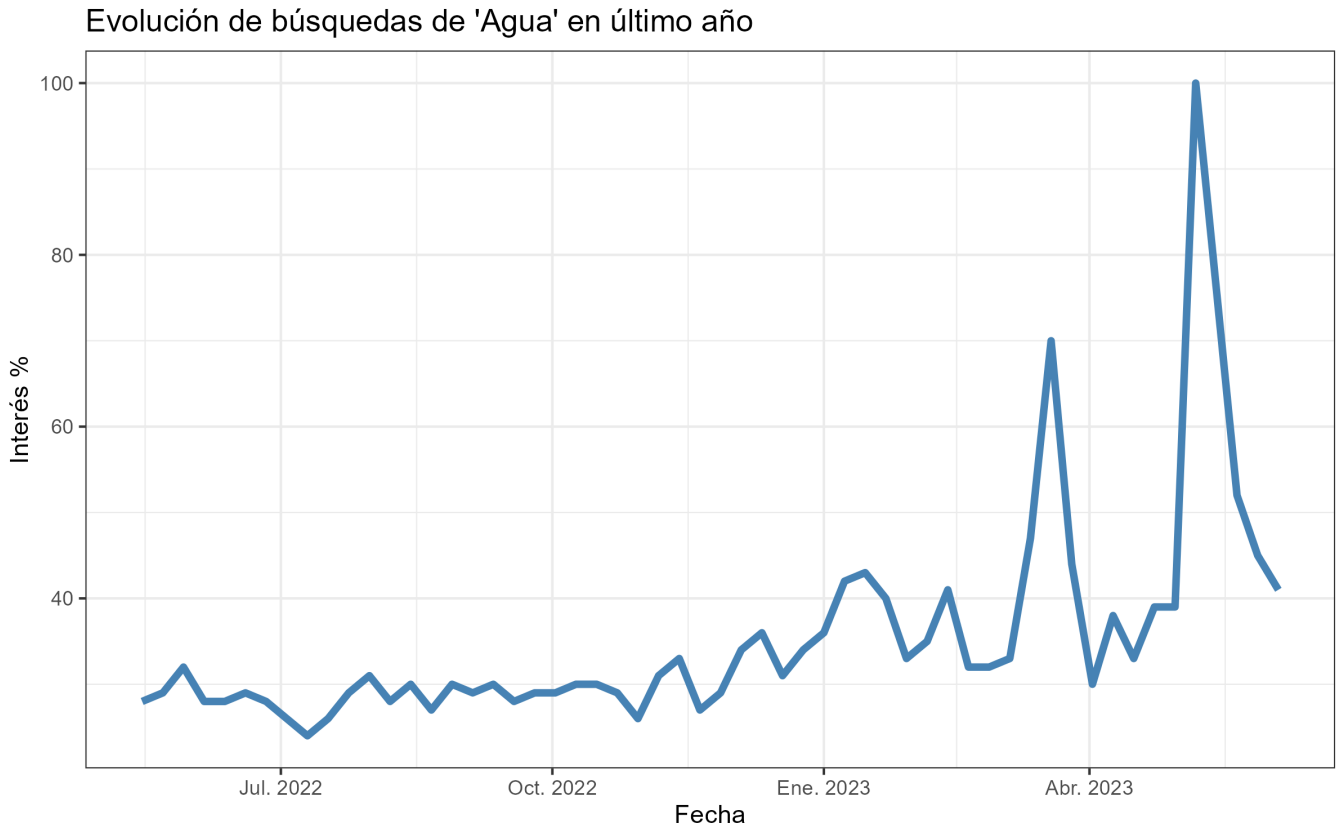
Objetivos de hoy

- Fuentes de datos: Google Trends, Audio y YouTube

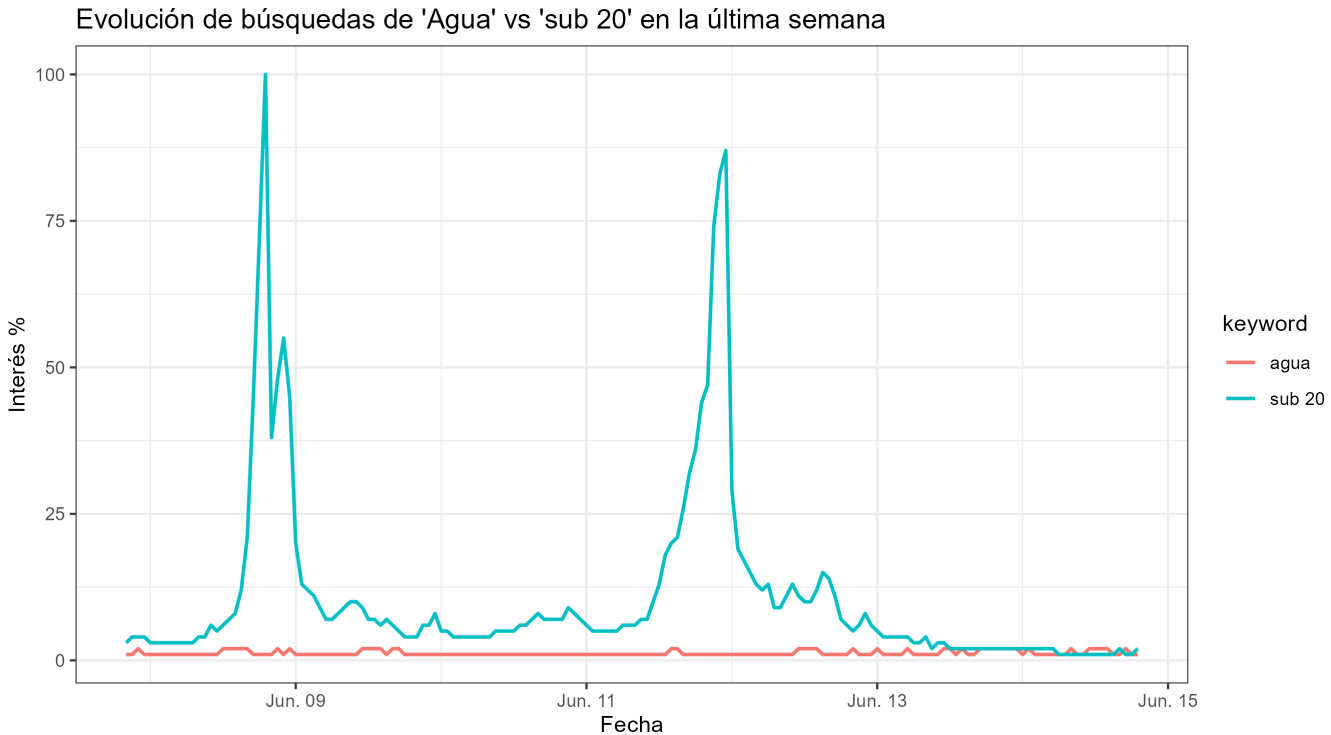
4. *gtrendsR*

- El paquete gtrendsR
- Permite realizar búsquedas de los términos más buscados en Google, proporciona una métrica propia para saber el volumen de búsqueda asociado.
- Permite análisis longitudinales, por países,departamentos, etc.
- Es útil para analizar intereses/preocupaciones de las personas lo cual nos puede dar información del ámbito *privado*, trascendiendo o complementando con los mensajes emitidos de carácter público (rrss)

4. *gtrendsR*: ejemplo con análisis del tema agua

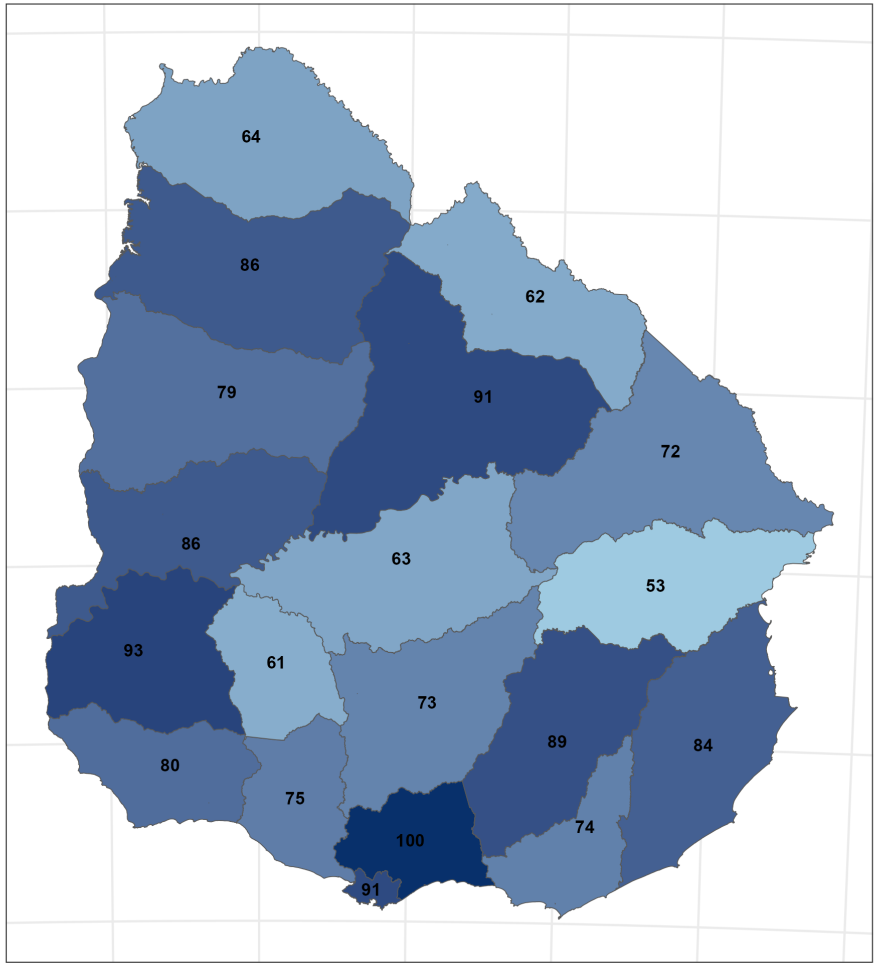


4. *gtrendsR*: ejemplo con análisis del tema agua

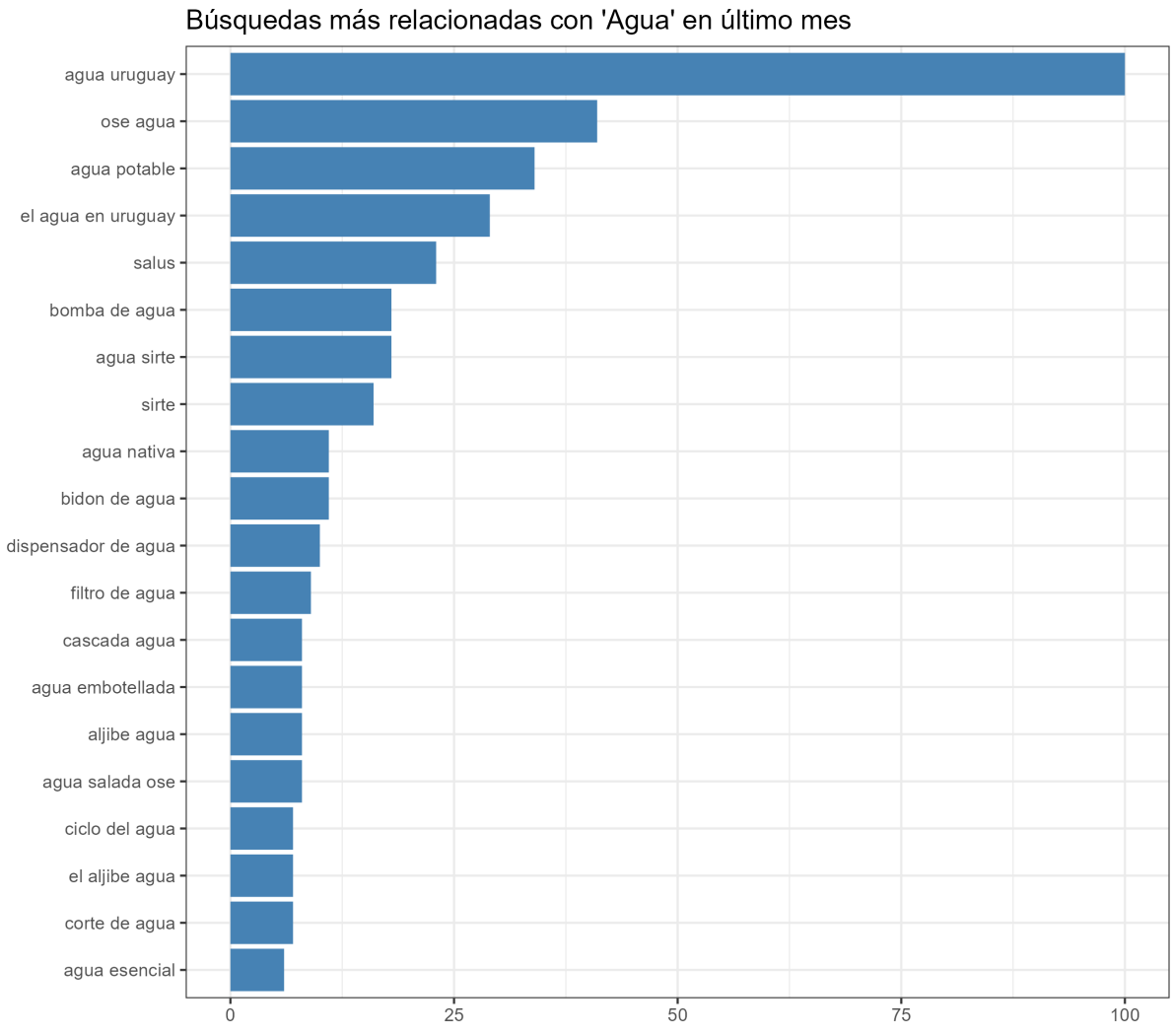


4. *gtrendsR*: ejemplo con análisis del tema agua

Búsquedas de 'Agua' en último mes por departamento



4. *gtrendsR*: ejemplo con análisis del tema agua



5. Audio

El paquete audio.whisper permite utilizar en R la herramienta de reconocimiento de voz *"Whisper" Automatic Speech Recognition model* desarrollada por openaAI.

Recuperar texto de audios es una fuente casi inagotable (entrevistas, discursos, conversaciones, podcast, etc.).

audio.whisper

- Tiene diferentes modelos que van desde el menos potente *(tiny)* al más potente *(large)*
- Cuanto mayor es la potencia y precisión del modelo más demora la transcripción
- No todos están disponibles para español
- Los pasos son sencillos y están bien explicados en el repositorio del paquete
- Se combina con la librería `av` para transformar los audios a formato de archivo *.wav de 16 bit*, que es el requerido por `audio.whisper`

audio.whisper + av

Obtengo un audio de interés y lo convierto a **.wav** con el paquete av

```
library(av) # conversor a .wav
library(audio.whisper) # transcpción

# 1. OBTENGO UN ARCHIVO DE AUDIO
# descargo para el ejemplo un audio de la web (podría ser un archivo que ya tengo en mi pc)
download.file("https://medios.presidencia.gub.uy/tav_portal/2018/noticias/AD_103/vazquez-cuidados.mp3", #
              "cuidados.mp3", # nombre del archivo que quedará en mi pc
              mode="wb") # modo web

# 2. CONVERSIÓN (av)
# convierto a .wav
av_audio_convert("cuidados.mp3", # nombre del archivo en mi pc
                 output = "cuidados.wav", # nombre del archivo convertido
                 format = "wav", sample_rate = 16000) # formato
```

audio.whisper

Realizo la transcripción con el modelo *tiny* (el menos potente)

```
# Descargo el modelo
# (podría saltear este paso poniendo la ruta en la función predict())
model <- whisper("tiny") # descargo modelo liviano
# lo corro indicando el idioma (es multilingual)
transcript <- predict(model, newdata = "cuidados.wav", language = "es")
# extraigo el df donde está el texto transcripto
texto_df <- transcript$data # df tiene 4 cols segmento, inicio, fin, texto
# guardo el df
save(texto_df,file="texto_df.RData") #o en el formato que quieras
```

audio.whisper

Construyo un cuadro con `knitr` y `kableExtra` con el texto

```
#olapso la columna text también podría usar un identificador y agrupar
texto_vec <- paste(texto_df$text,collapse="")
tabla1 <- knitr::kable(texto_vec,
          col.names = "Tabaré Vázquez - Sistema de Cuidados", # agrego nombre
          format = "html", table.attr = "style='width:100%;'") %>% #formato
  kableExtra::kable_styling(font_size = 24) %>% # defino tamaño de letra
  kableExtra::kable_classic() # defino el estilo de la tabla
```

audio.whisper

Tabaré Vázquez - Sistema de Cuidados

Con respecto al sistema nacional de cuidados, dijimos en aquel momento, se implementará este sistema priorizando y aquí definimos tres poblaciones que queríamos atender. La primera infancia, las personas con discapacidad y adultos mayores en situación de dependencia. Lo dijimos en junio, de 2014. Hoy, ¿qué tenemos? Se implementó un proyecto de ley para crear un sistema nacional de cuidados. Y en esa ley aprobada se creó una Junta Nacional de cuidados, una secretaria nacional de cuidados y un comité consultivo de cuidados, la ley 19.353, que constituye del alma institucional de este sistema que pretendemos y estamos seguro, cualquiera de hacer al próximo Gobierno va a continuar adelante porque una acción de este tipo. Por la importancia humana que tiene sin duda creemos y lo creemos sincera, pero modestamente debe constituirse en una política de Estado. Y atendimos a la primera infancia, al día hoy, hay tres, tres, mil, doscientos y en cuenta niños, onineas, de cero a tres años, que están siendo cuidados por personal capacitado especialmente para hacer esta tarea. En personas de situación de dependencia, 4,688 personas cuentan ya con un asistente personal. En algunos casos pagados por todos ustedes, por el Estado de los casos cuando la capacidad económica del hogar permite pagar una parte de otra parte de la pagada del Estado, o simplemente pagar las familias, pero los cuidadores son especializados, especialmente para realizar esta tarea. En este tipo, en tela existencia en casa es decir personas que están conectadas con un sistema central de respuesta hay 832 personas activas en este momento. Para formar a las personas hubo 40 cursos que aún se encuentra en marcha porque la capacitación es permanente. 1773 personas completaron el curso de atención a la dependencia. Más de 3.000 se formaron para la atención a la primera infancia. Y el portal de cuidados que ustedes pueden visitar, todos los datos que estamos pueden ser corroborados y aquí hasta de terminar hablar. El portal de cuidados tiene en atención lo han consultado más de 48.000 740 personas desde el año 2016.

audio.whisper + scraping

- La utilidad de la transcripción es mayor cuanto más audios tengamos
- Transcribir una entrevista puede ser divertido, 10 es agotador, más de 10 hay contratar a alguien y en general no tenemos recursos
- La potencia se acrecienta combinando herramientas
- Un buen ejemplo es realizar scrapeo de audios de la web con `rvest`

audio.whisper + scraping + rgtp3

También podemos usar el paquete `rgtp3` que permite conectar R con la herramienta de openAI *chatGPT3*

La API es de pago, pero para un ejercicio básico alcanza con lo que te permite utilizar gratis

- Acá pueden descargar un ejemplo con:

1. Descarga de audios `rvest`
2. Transcripción con `audio.whisper`
3. Resumen e identificación de tema principal con `rgpt3`

- *Acá* hay otro ejemplo de uso de `rgtp3` (no de audio) donde pueden ver los pasos para conectar con la API

6. YouTube

El paquete `youtubecaption` permite descargar los subtítulos de los videos de YouTube

Trabaja sobre la librería de Python `youtube-transcript-api`

Es necesario conectar R y Pyhton, lo que puede realizarse con librería `reticulate` que permite la instalación de *miniconda* o la interfaz *Anaconda* para gestionar los paquetes (ver archivo *instalaciones* del curso)

6. youtubecaption

- Es posible recuperar texto de todos los videos que cuentan con subtítulos (incoprorados o generados automáticamente)
- Si los subtítulos son automáticos la fidelidad generalmente depende de la claridad del audio
- `youtubecaption` recupera la transcripción de forma tabulada y ordenada para cada secuencia del video, por lo que luego es necesario agrupar por el identificador y recuperar la metadata original (fecha, resumen, canal, visualizciones, etc.)

6. youtubecaption

Hay tantas alternativas como videos de YouTube existan: discursos, conferencias, entrevistas, canciones, películas, programas de tv...

```
# hadley wickham
url <- "https://www.youtube.com/watch?v=cpbtcsGE0OA"
caption <- get_caption(url)

# suarez
url2 <- "https://www.youtube.com/watch?v=KsE8a9NOtnU"
caption2 <- get_caption(url2, language = "es")

# agarrate catalina
url3 <- "https://www.youtube.com/watch?v=LApsPiejZLI"
caption3 <- get_caption(url3, language = "es")
```

6. youtubecaption

También `youtubecaption` se potencia con la combinación de herramientas

- Acá hay un ejemplo de uso con videos del presidente Lacalle Pou:

1. Scraping con Apify
2. Descarga con `youtubecaption`
3. Análisis con `quanteda` y `udpipe`
4. Visualización con `ggplot2`

Otros recursos disponibles

- Recursos en línea para el estudio de la conflictividad http://observatoriodeconflictividad.org/
- Paquete ACEP: Analisis Computacional de Eventos de Protesta
- *ACEP es un paquete de funciones en lenguaje R utiles para la deteccion y el analisis de eventos de protesta en corpus de textos periodísticos. Sus funciones son aplicables a cualquier corpus de textos. Ademas de las funciones, ACEP contiene también bases de datos con colecciones de notas sobre protestas y una colección de diccionarios de palabras conflictivas y otros tópicos referidos a diferentes aspectos del análisis de eventos de protesta.*
- Autor: Agustín Nieto (Universidad Nacional de Mar del Plata)

Otros recursos disponibles

- Paquete internetarchive permite scrapear del sitio **Internet Archive**
- Hemeroteca o biblioteca digital archive.org *gestionada por una organización sin ánimo de lucro dedicada a la preservación de archivos, capturas de sitios públicos de la Web, recursos multimedia, etc.*