Recuperación y análisis de texto con R

Clase 7 - 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

1. R Markdown
2. Repaso general
3. Consigna de trabajo final e intercambio

1. R Markdown

- Qué es R Markdown?

  *"Los documentos de **R Markdown** son totalmente reproducibles. Utilizando una interfaz como RStudio para unir el texto narrativo y el código para producir resultados elegantemente formateados. Permitiendo compaginar varios lenguajes, incluidos LaTeX, R, Python y SQL."*
  (https://rmarkdown.rstudio.com/)

- ¿Por qué es tan importante?

Instalación

```
install.packages(c("rmarkdown","tinytex"))

##Instalo lo que necesito de latex para correr documentos en pdf:
tinytex::install_tinytex()
```

Formatos prestablecidos

- Artículos
- PDF
- Documentos Word
- Presentaciones Beamer
- Libros
- Tesis
- y varios más

Encabezados

Se define título, autor/a, tipo de documento (html_document, pdf_document, word_document, beamer_presentation, ioslides_presentation), y se cargan librerías LaTeX.

Listas de contenido, figuras y tablas (PDF)

```
---
title: "Titulo"
author: "Nombre"
output:
    pdf_document:
     fig_caption: true
     number_sections: true
     toc: true
---
```

Listas de contenido, figuras y tablas (HTML)

```
---
title: "Titulo"
author: "Nombre"
output:
    html_document:
     toc: true
     toc_depth: 3
     number_sections: true
---
```

Partes del documento

```
# Título de orden 1
## Título de orden 2
### Título de orden 3
#### Título de orden 4
##### Título de orden 5
```

Efectos de fuentes

- `_italicas_` o `*italicas*` : *italicas* o *italicas*
- `__negritas__` o `**negritas**` : **negritas** o **negritas**
- `**_negrita e itálica_**` o `__*alternativamente*__` : ***negrita e itálica*** o ***alternativamente***
- `~~tachado~~` : ~~tachado~~
- `Subindices y superíndices: SO~4~^2^` : Subindices y superíndices: $SO_4{}^2$
- `Fórmulas: $\frac{\sqrt{\lambda} }{n_i}$, $\mbox{SO}_4^=$` : $\frac{\sqrt{\lambda}}{n_i}$, $SO_4^=$
- y varios más

Viñetas

```
* Francia

- Croacia

+ Bélgica
```

- Francia
- Croacia
- Bélgica

Enumeración

```
1. Francia
1. Croacia
1. Bélgica
```

1. Francia
    1. Croacia
        1. Bélgica

Enumeración

```
(@uno) Francia

(@dos) Croacia

(@tres) Bélgica

* ¿Quién salió campeon @uno, @dos? o @tres?
```

(1) Punto uno
(2) Punto dos
(3) Punto tres

- ¿Qué es el punto 1? ¿y el 2? ¿o el 3?

Caracteres especiales

Para que se impriman caracteres especiales tales como:

- Contra barra y espacio: \
- Tilde grave: ‘
- Asterisco: *
- Barra baja: _
- Paréntesis: {} [] ()
- Numeral: #
- Otros: + - . ! : |

Se debe utilizar una \ antes del símbolo

Nota al pie

- Para poner una nota al pie debo escribir [1], y posteriormente en otra linea poner el contenido de la nota como se ve a continuación.

```
[^1]: Dejo un espacio y acá escribo la nota al pie.
```

[1] Dejo un espacio y acá escribo la nota al pie.

Incluir imágenes

- Poner una imagen desde un archivo:

```
![](imagenes/logoFCS_UdelaR_n2_RGB.jpg)
```

Parametros imágenes

Las opciones son lo que le pasamos al comando entre los corchetes y nos permiten controlar cosas de la imagen. Aquí os recopilo las que yo uso más:

- *height*: la altura que debe tener la figura, escalará el gráfico hasta que tenga esta altura
- *width*: la anchura que debe tener la figura, escalará el gráfico hasta que tenga esta anchura
- *scale*: cuánto hay que escalar la figura, sobre 1
- *angle*: cuánto hay que girar la figura, en grados

Hipervínculos

Se puede poner un enlace con un texto:

Página principal de RMarkdown

o directamente:

https://rmarkdown.rstudio.com/

- `[Página principal de RMarkdown](https://rmarkdown.rstudio.com/)`
- `<https://rmarkdown.rstudio.com/>`

Poner código R

- Código incrustado en el texto

Somos ‘ r 2 + 2 ‘

Somos 4

o

La cantidad de titulares de TUS es de ‘ r nrow(tus) ‘

Chunks

Hay tres formas de insertar rápidamente un *chunk* en el documento:

- con el atajo de teclado **Ctrl + Alt + i**
- con el botón de la barra superior (incluso ya pudiendo definir el lenguaje a utilizar)
- o directamente tipeando los delimitadores ```` ```{r} ```` y ```` ``` ````.

Cuando se renderice el archivo .Rmd, R Markdown ejecutará cada fragmento de código (chunk) e insertará los resultados debajo del fragmento de código en su informe final.

Opciones del Chunk

El resultado de cada Chunk puede personalizarse con opciones de la librería *knitr* , sus argumentos se definen entre {} del encabezado del chunk. Aquí, el top five de argumentos:

- **include = FALSE** impide que el código y los resultados aparezcan en el archivo renderizado. R Markdown de todos modos ejecuta el código en el chunk, y los resultados pueden ser utilizados por otros chunks
- **echo = FALSE** impide mostrar el código, pero no los resultados que aparecen en el archivo terminado. Esta es una forma útil de insertar figuras.

Opciones del Chunk

- **message = FALSE** impide que los mensajes generados por código aparezcan en el archivo final.
- **warning = FALSE** evita que las advertencias generadas por el código aparezcan en el final.
- **fig.cap = "..."** agrega un título a los resultados gráficos.

Otras opciones:

Para la lista completa de opciones se puede ver la Guia de R Markdown o la propia página de *knitr*.

Opciones globales

para definir opciones globalmente, que apliquen a todos los chunks de tu archivo, debemos usar: `knitr::opts_chunk$set` en cualquier chunk. Knitr tratará cada option definida por `knitr::opts_chunk$set` como la opción predeterminada para todo el documento, pero puede ser redefinido individualmente en cada encabezado de chunk.

Recursos útiles

- **Hoja de Referencia RMarkdown:**

https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-spanish.pdf

- **Tutorial:**

http://fobos.inf.um.es/R/taller5j/30-markdown/guiabreve.pdf

- **Libro en repositorio:**

https://github.com/rstudio/rmarkdown-book

Tabla simple

Table 1: Título

Centrado	Derecha	Izquierda
valor	10	200
10	200	valor
200	valor	10

```
| Centrado | Derecha | Izquierda |
|:--------:|--------:|:----------|
```

```
Table: Título
```

Función knitr::kable()

Por defecto hace unas tablas muy bonitas. Tiene pocas opciones, así que, por un lado es muy fácil de aprender a usar pero, por otro, si queremos algo más concreto puede quedarse corta. Una característica a destacar es que en un pdf, si quedara muy larga la tabla para una página, por defecto kable() la divide en dos y la continúa en la siguiente.

Función knitr::kable()

```
kable( df, caption = "BBDD airquality con kable()",
                  align = c('l', 'c', 'r', 'r', 'c', 'l'),
                  col.names = c("Ozono","Solar.R","Viento","T
                  row.names = TRUE,
                  digits = 1,
                  format.args = list( decimal.mark = ",")
       )
```

Función knitr::kable()

Table 2: BBDD airquality con kable()

	Ozono	Solar.R	Viento	Temp	Mes	Día
1	41	190	7,4	67	5	1
2	36	118	8,0	72	5	2
3	12	149	12,6	74	5	3
4	18	313	11,5	62	5	4
5	NA	NA	14,3	56	5	5
6	28	NA	14,9	66	5	6
7	23	299	8,6	65	5	7
8	19	99	13,8	59	5	8
9	8	19	20,1	61	5	9
10	NA	194	8,6	69	5	10
11	7	NA	6,9	74	5	11
12	16	256	9,7	69	5	12
13	11	290	9,2	66	5	13
14	14	274	10,9	68	5	14
15	18	65	13,2	58	5	15

Parámetros

Los documentos R Markdown pueden incluir uno o más parámetros cuyos valores se pueden establecer cuando se procesa el informe. Por ejemplo, el archivo siguiente utiliza un parámetro de variable que determina qué variable será utlizada en el informe. Los parametros son declarados usando el campo `params` dentro del preámbulo (YAML) al inicio del documento.

```
title: "Documento_prueba"
output: pdf_document
params:
  variable: "A1_1"
```

Usando Parametros (I)

Los parámetros están disponibles dentro del entorno de `knitr` como una lista de solo lectura llamada `params`. Para acceder a un parámetro en el código, lo debemos llamar mediante `params$<nombre del parametro>`

> *Aquí se analiza la variable ' r params$variable ' , que presenta una media de ' r mean(enaj_chica[params$variable]) ' y ....*

Usando Parametros (II)

Los parámetros están disponibles dentro del entorno de `knitr` como una lista de solo lectura llamada `params`. Para acceder a un parámetro en el código, lo debemos llamar mediante `params$<nombre del parametro>`

```
# Primero uso la función attach()
# para juntar la base y la variable de interés
# attach(enaj_chica$A1_1)
```

Aquí se analiza la variable ' r params$A1_1 ' , que presenta una media de ' r mean(params$A1_1) ' y . . . .

Renderizando con otro parametro

Si modificamos el argumento de params al renderizar el documento, se crea un informe que usa el nuevo conjunto de valores de parámetros. Aquí modificamos nuestro informe para usar la variable "A1_2":

render("Informe_ENAJ.Rmd", params = list(variable = "A1_2"))

2. Repaso general

Objetivos del curso

Bases teóricas:

- Contextualizar las **Ciencias sociales computacionales**
- Emergencia de nuevos recursos y técnicas para la investigación social en la era digital.

Objetivos del curso

Generalidades del lenguaje R:

- R como software libre y gratuito
- Comunidades y foros
- Tidyverse
- Manipulación básica de strings

Objetivos del curso

Exploración de fuentes de datos textuales:

- Exploración y obtención de datos de diversa índole, contemplando las diferentes fuentes posibles: OCR, web sacraping, prensa digital, redes sociales, audio, Youtube, APIs.

Objetivos del curso

Análisis textual:

- Codificación manual de textos y creación de redes multinivel (categorías, códigos y citas) mediante la plataforma RQDA().
- Abordaje de los requerimientos previos (limpieza y homogeneización) para el análisis de textos.
- Trabajo con minería de textos, el cual se centrará en la noción de *corpus* y sus posibilidades analíticas, desde lo más descriptivo a la aplicación de técnicas más complejas.

Objetivos del curso

Análisis textual:

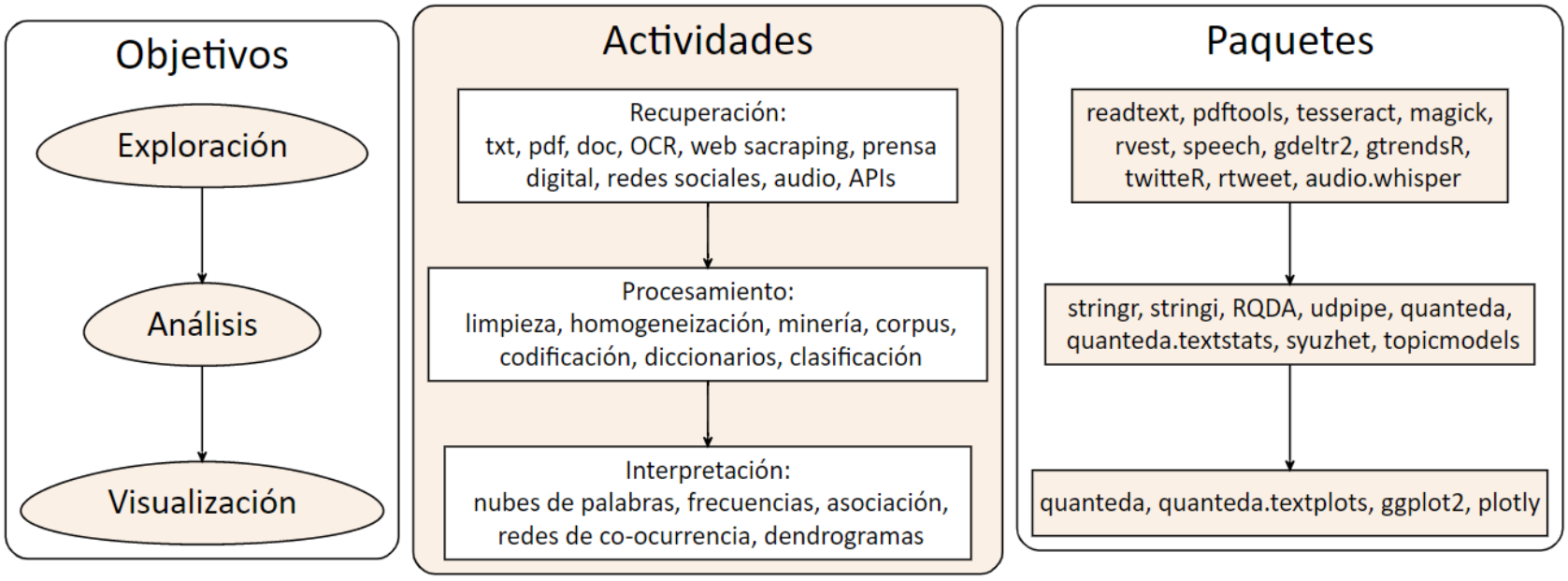
- Trabajo con diccionarios: Introducción al uso de diccionarios (manuales y automáticos), para la clasificación de documentos masivos según intereses particulares.
- Clasificación de textos: clasificación de textos según temas o emociones asociadas a partir de la aplicación de diferentes técnicas existentes.

Objetivos del curso

Visualización:

- Exploración de las diferentes posibilidades gráficas de visualización de los resultados del análisis textual (nubes de palabras, frecuencias, dendrogramas, etc.) y algunos ejemplos de visualización interactiva.

Esquema del curso



Metodología

- El enfoque del curso es práctico (hands-on)
- Trabajaremos con estretagia de live-coding y ejercicios prácticos para cada tema.
- Posibilidad de clonar repositorio GitHub y trabajar con proyecto y control de versiones.
- https://github.com/elinagomez/analisistextoEPUdelar2023

Tutorial R+ GitHub

Consejo

- Elegir un tema de interés
- Hacerse una pregunta inicial
- Identificar una fuente textual para responderla

3. Consigna de trabajo final e intercambio

- Consigna de trabajo final

3. Consigna de trabajo final e intercambio

Intercambio

1. Tema / Pregunta
2. Recuperación
3. Pre-procesamiento
4. Análisis
5. Visualización