

R aplicado al análisis cualitativo / FCS-UdelaR

Clase 4 - Educación Permanente FCS



Mag. Elina Gómez (UMAD/FCS)

elina.gomez@cienciassociales.edu.uy

www.elinagomez.com



Este trabajo se distribuye con una licencia Creative Commons Attribution-ShareAlike 4.0 International License

Objetivos de hoy

- Presentación del paquete **RQDA** para procesamiento y codificación manual de textos.
- Creación de redes multinivel (categorías, códigos y citas) mediante la plataforma RQDA().

Encuadre teórico: Teoría Fundamentada

- Uno de los enfoques más difundidos para el análisis cualitativo es la denominada *Teoría Fundamentada* (TF), planteada por Glaser y Strauss (1967) y con posteriores actualizaciones y modificaciones (Strauss y Corbin, 2012).
- El objetivo central es la generación de teoría partiendo del análisis de los datos y plantea un proceso de estructuración y análisis de los datos que involucra un conjunto de pasos y reglas para su procesamiento.

Encuadre teórico: Teoría Fundamentada

Algunos aspectos distintivos comunes de la TF y posteriores:

- Comparación constante
- Muestreo teórico
- Elaboración de memorandos (metodológicos, teóricos, analíticos y descriptivos)
- Sensibilidad teórica

(Estrada, Giraldo y Arzuaga, 2020)

Encuadre teórico: Teoría Fundamentada

- El método comparativo constante implica un proceso de recolección y codificación de forma **sistemática**, descubrimiento de patrones y generación de teoría que se encuentra fundamentada en los datos (método inductivo).
- El proceso de codificación en la TF clásica, se divide en tres pasos: abierta, axial y selectiva.

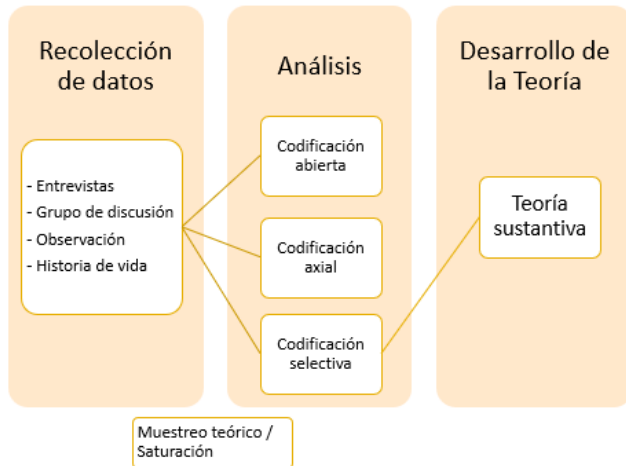
Encuadre teórico: Teoría Fundamentalada

Abierta: establecer códigos según categorías y conceptos

Axial: incorporación y comparación relacional de nuevos datos con categorías establecidas previamente (esquema)

Selectiva: al existir una insuficiencia de información y datos, se procede a la búsqueda de nuevos datos que aporten información relevante para expandir el análisis, con capacidad explicativa para generar teoría.

Encuadre teórico: Teoría Fundamentada



Encuadre teórico: Teoría Fundamentada

- La codificación permite estructurar los datos a partir de dimensiones, categorías, conceptos. Se asocian fragmentos de textos a cada uno.

Tipos:

- Pre-establecidos
- Emergentes
- En vivo (citas)

Encuadre teórico: Teoría Fundamentada

Existen software y paquetes que ayudan al proceso de análisis y codificación de la información.

- Nvivo
- Atlas ti
- MAXQDA
- RQDA
- entre otros.

RQDA

- **RQDA** es un paquete para el análisis cualitativo de textos, el cual permite la codificación de textos.
- Se conecta con todas las funcionalidades de R por lo que se puede combinar el análisis cualitativo y cuantitativo.

¿Qué nos permite?

- Importar archivos de texto plano
- Producir documentos
- Codificar los documentos
- Agrupar códigos en categorías
- Hacer notas de documentos, de códigos, de codificaciones y de proyectos.
- Recuperar la codificación y regresar al documento original.
- Renombrar códigos y categorías de códigos.

Instalación

#Documentación y adaptación para versión R nueva en:
<https://github.com/RQDA/RQDA>

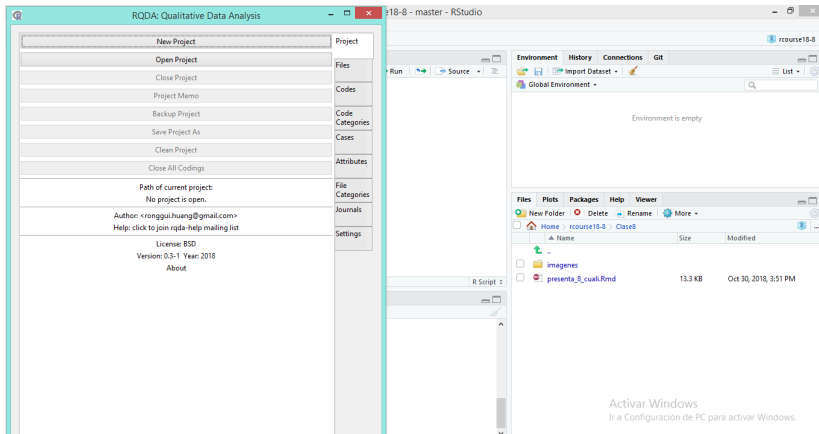
```
pkgs <- c("RSQLite", "gWidgets2RGtk2", "DBI",  
          "stringi", "RGtk2", "igraph", "gWidgets2",  
          "devtools")  
install.packages(pkgs)  
  
library(RGtk2) ##Poner OK para instalar Gtk2  
devtools::install_github("RQDA/RQDA",  
  INSTALL_opts = "--no-multiarch")
```

Instalación

```
library(RQDA) ##Cargo y chequeo que esté bien  
RQDA() #Abro interfaz
```

Información completa: <http://rqda.r-forge.r-project.org/>

Interface



Interface

- El **RQDA** funciona en base a archivos **.rqda**
- El menú incluye: **Proyectos, Archivos, Códigos, Categoría de códigos, Casos, Atributos, Memos**

Otras funcionalidades

Ventajas y desventajas

Ventajas:

- Libre y gratuito
- Combinación con múltiples funcionalidades de R
- Permite exportación de conjuntos de códigos según filtros específicos

Desventajas:

- Por el momento sólo admite textos planos (.txt)
- No permite establecer relaciones lógicas complejas entre códigos.

Menú

- **Proyectos:** archivo .rqda que almacena todos los archivos, códigos, categorías, etc. Se puede configurar el nombre del codificador para identificar a el/la responsable del proyecto.
- **Archivos:** se importan los archivos de texto plano .txt (codificación ASCII por defecto pero se puede modificar).
- **Códigos:** son las *etiquetas* que se asigna a cada cita en los diferentes archivos. Un resumen se puede obtener con `getCodingTable()` (“Número de codificaciones para cada código”, “Número promedio de palabras asociadas con cada código” y “Número de archivos asociados con cada código”).
- **Categoría de códigos:** son las categorías de nivel superior, a las que se asocia conceptualmente a los códigos.
- **Categoría de archivos:** son las categorías que agrupan a cada uno de los archivos según algún criterio de nivel superior.

Menú

- **Casos:** es la unidad de análisis que se está trabajando, se puede asignar a un archivo o una parte del mismo.
- **Atributos:** los atributos sirven para la clasificación cuantitativa. Se puede asignar a un archivo o a un caso.
- **Memos:** son notas que se pueden crear tanto para los archivos, códigos, categorías, etc.
- **Journal:** notas de campo.

Ejemplo práctico

■ Importo los archivos

The screenshot displays the RQDA (Qualitative Data Analysis) interface within an RStudio environment. The RQDA window is titled "RQDA: Qualitative Data Analysis" and features a menu bar with options: Import, Nuevo, Eliminar, Abrir, Memo, rename, and Attribute. Below the menu, a "Selected File id is 1" section lists several files: Mujeres_Adultos_1, Mujeres_Jovenes_1, Mujeres_Jovenes_2, Mujeres_Jovenes_3, Varones_Adultos_1, and Varones_Jovenes_1. A sidebar on the right of the RQDA window contains tabs for Project, Files, Codes, Code Categories, Cases, Attributes, File, Categories, Journals, and Settings.

The RStudio interface shows the "Environment" pane on the right, listing the loaded data objects:

Object	Description
a	2 obs. of 6 variables
base	2034 obs. of 657 variables
base2	165 obs. of 657 variables
base3	165 obs. of 657 variables
base4	345 obs. of 657 variables
tablacodigos	18 obs. of 8 variables

The "Console" pane at the bottom left shows the execution of R commands, including a loop that reads and processes data files. The "Environment" pane also displays the "Usage" and "Arguments" for the `getCodingaByOne` function, which takes `cid` (code id), `fid` (file id), and `codingTable` as arguments.

Ejemplo práctico

- Creo los códigos a lo largo del texto con la función *Mark*

The screenshot displays the RStudio environment with the RQDA (Qualitative Data Analysis) package loaded. The interface is divided into several panes:

- Top Left:** A menu bar with options: `File`, `Edit`, `Import Dataset`, `Import Text File`, `Import Excel`, `Import CSV`, `Import JSON`, `Import XML`, `Import HTML`, `Import PDF`, `Import Word`, `Import PowerPoint`, `Import Video`, `Import Audio`, `Import Image`, `Import Other`.
- Top Right:** A toolbar with icons for `File`, `Edit`, `Import Dataset`, `Import Text File`, `Import Excel`, `Import CSV`, `Import JSON`, `Import XML`, `Import HTML`, `Import PDF`, `Import Word`, `Import PowerPoint`, `Import Video`, `Import Audio`, `Import Image`, `Import Other`.
- Left Pane:** A list of files and folders, including `class2`, `112`, `113`, `114`, `115`, `116`, `117`, `118`, `119`, `120`, `121`, `122`, `123`, `124`, `125`, `126`, `127`, `128`, `129`, `119:1`, `Console`, `~/R_Cuali`, `don't`, `5: In r`, `don't`, `Error: l`, `In addi`, `1: In f`, `the c`, `2: In d`, `row n`, `3: In r`, `don't`, `4: In r`, `don't`.
- Right Pane:** A list of data frames and variables, including `Global Environment`, `Data`, `a` (2 obs. of 6 variables), `base` (2034 obs. of 657 variables), `base2` (165 obs. of 657 variables), `base3` (165 obs. of 657 variables), `base4` (345 obs. of 657 variables), `tablacodigos` (18 obs. of 8 variables).
- Bottom Left:** A console window showing the execution of the `Mark` function. The output indicates that the function is creating codes along the text. The console text is:


```

      1: In f
      the c
      2: In d
      row n
      3: In r
      don't
      4: In r
      don't
      >
      
```
- Bottom Right:** A pane showing the documentation for the `Mark` function. It includes the title `Mark`, a description `Se puede asignar a un`, and a list of arguments: `cid` (code id, an integer), `fid` (file id, an integer vector), and `codingTable` (name of coding table). The details section states: `It gets codings of a code with cid from files which are specified by fid`.

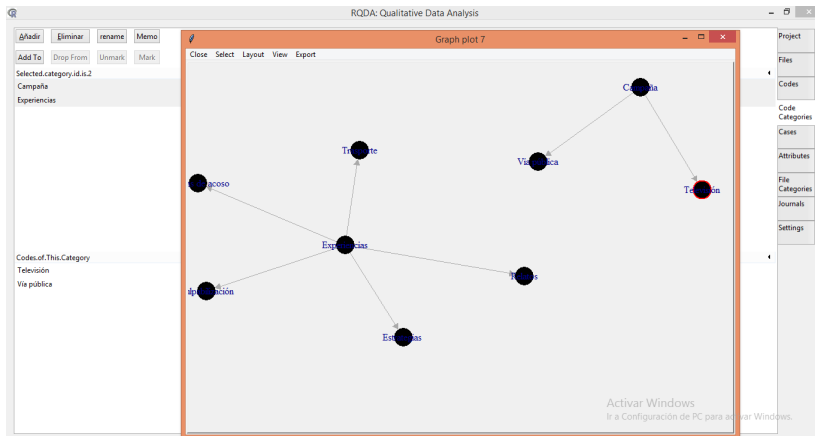
Ejemplo práctico

- Puedo plotear las categorías y códigos asociados

The screenshot shows the RQDA (Qualitative Data Analysis) package interface within RStudio. The main window is titled "RQDA: Qualitative Data Analysis". On the left, there is a list of codes, with "Campaña" selected. A context menu is open over "Campaña", showing options: "Add New Code to Selected Category", "Codings of selected category", "Memo", "Plot Selected Code Categories", "Plot Selected Code Categories with d3", and "Sort by created time". The "Plot Selected Code Categories" option is highlighted. On the right, there is a list of code categories, including "Codes", "Code Categories", "Cases", "Attributes", "File Categories", "Journals", and "Settings". The R console on the bottom left shows an error message: "Error in get_codings_of_one_code(cid, fid=NULL, codingTable=c(\"coding\", \"codi...". The Environment pane on the right shows a list of data objects: "a" (2 obs. of 6 variables), "base" (2034 obs. of 657 variables), "base2" (165 obs. of 657 variables), "base3" (165 obs. of 657 variables), "base4" (345 obs. of 657 variables), and "tablacodigos" (18 obs. of 8 variables).

Ejemplo práctico

- Puedo plotear las redes de categorías y códigos asociados a las mismas



Funciones

- Existen funciones que me permiten enriquecer el análisis e interactuar con otros paquetes

Resumen de los códigos

La función *summaryCodings()* nos devuelve una lista con:

- Número de codificaciones por código
- Número de caracteres asociados a cada código
- Número de archivos asociados a cada código

Agregando el argumento *byFile= TRUE* se puede obtener la información para cada archivo

Tabla con codificaciones

Con la función *getCodingTable()* obtenemos un data frame con todos los códigos y características de las codificaciones

```
library(RQDA)
getCodingTable()
```

Codificación por palabra

Con la función *codingBySearch()* es posible codificar todas las menciones que involucren a una palabra o expresión determinada, con un código. Se debe indicar el separador que define hasta dónde se hará la codificación.

```
codingBySearch("taxi", fid=1, cid=1, seperator = "\n")
```

```
#fid - número de archivo
```

```
#cid - número de código
```

```
#seperator - ("\\n" ; "[.!?]")
```

Relación entre archivos y códigos

La función *filesByCodes()* devuelve un data.frame con los códigos asociados a cada archivo

```
filesByCodes()  
  
##Para códigos específicos  
  
filesByCodes(codingTable = c("coding", "coding2"))
```

Búsqueda de códigos

La función *getCodingsByOne()* sirve para buscar las codificaciones asociadas a los códigos. Es posible buscar codificaciones cruzadas.

```
#obtengo  
getCodingsByOne(1)  
  
#Para hacer búsquedas cruzadas  
  
getCodingsByOne(1) %and% getCodingsByOne(9)  
  
#podría ser: %and%; %or%; %not%
```

Co-ocurrencia de códigos

La función *crossTwoCodes()* devuelve una matriz de co-ocurrencia de códigos.

```
#Busca la co-ocurrencia entre los códigos que se seleccionen por vector o seleccionando  
crossTwoCodes(relation = "exact", data=tabla_cods, cid1 = 1, cid2 = 9)  
  
#relation=c("overlap", "inclusion", "exact", "proximity")
```

Exporto a HTML

```
#Exporto los archivos codificados  
exportCodedFile("archivos.html", fid = 1)  
  
#Exporto los códigos  
exportCodings("codigos.html")
```


Otras funciones

Otras funciones se pueden encontrar en la documentación del paquete:

[RQDA](#)

RQDAPlus

Es una shiny app que complementa las funcionalidades y permite un procesamiento visual de los datos.

Instalación de RQDAPlus para el análisis rápido de datos 1. Escriba
remotes::install_github ("stats4sd/RQDAPlus", upgrade =
"always") 2. Escriba RQDAPlus::RQDAPlus
("C:/FilePath/nombreDeArchivo. rqda")

RQDAPlus

¿Qué puede hacer RQDAPlus?

- Tablas de frecuencia de co-ocurrencia de cualquier código (s), categorías de código y/o caso (s)
- Tablas de texto de co-ocurrencias de cualquier código (s), categorías de código y/o caso (s)
- Textos de salida de cualquier co-ocurrencia en formato csv o html
- Crear nubes de palabras para cualquier código(s), categorías de código, y/o caso (s) seleccionados.
- Tabla de matriz de adyacencia que muestre la frecuencia de los códigos.
- Crea análisis gráfico de redes mostrando la relación de los códigos entre archivos (y agrupación de códigos basado en la ubicación entre archivos).

Ejercicio

- 1 Abrir el proyecto *clase_4_Ejemplo_Acoso.rqda*
- 2 Crear dos categorías con dos códigos en cada una y citas asociadas
- 3 Averiguar el número de caracteres asociado a cada código
- 4 Hacer un plot de las categorías y códigos
- 5 Hacer un data.frame *cods* con los códigos asociados a cada documento.
- 6 Hacer un gráfico con ggplot con la frecuencia de los códigos asociados a cada documento

RQDAQuery

Para hacer consultas más específicas que no sean las de las funciones vistas, se debe utilizar la función `RQDAQuery` cuyo único argumento será una consulta SQL en formato `character`