Concept Mapping con R

Etnografía y Análisis Cualitativo de Datos. Grado en Antropología Social y Cultural

Arturo Alvarez-Roldan
aalvarez@ugr.es
Departamento de Antropología Social, Universidad de Granada

06 / June / 2025

DOI: 10.5281/zenodo.15659666

Índice

Resumen: Este documento detalla un procedimiento completo para aplicar el método Concept Mapping en R con fines de análisis cultural. Utilizando datos de clasificación de tarjetas y puntuaciones de importancia y viabilidad de 76 ideas para mejorar una empresa, se construyen matrices de proximidad y se aplican técnicas como escalado multidimensional (MDS), análisis de conglomerados y gráficos Go-zone. Se analiza cómo diferentes perfiles de los participantes (según su experiencia y tamaño de empresa) valoran las ideas de mejora. El enfoque integra análisis cualitativos y cuantitativos, facilitando la comprensión de cómo se organizan colectivamente las percepciones y prioridades dentro de una comunidad.

Abstract: This document outlines a comprehensive procedure for applying the Concept Mapping method in R for cultural analysis purposes. Using card sorting data and ratings of importance and feasibility for 76 business improvement ideas, proximity matrices are constructed and techniques such as multidimensional scaling (MDS), cluster analysis, and Go-zone plots are applied. The analysis explores how different participant profiles (based on experience and company size) evaluate the improvement ideas. The approach integrates both qualitative and quantitative analysis, facilitating an understanding of how perceptions and priorities are collectively organized within a community.

1 Introducción

El **método *Concept Mapping* (CM)** o **mapeo conceptual** es una técnica de investigación participativa que se utiliza para la **planificación y evaluación de intervenciones y diagnósticos en organizaciones y comunidades** (Kane y Trochim 2007; Trochim y Linton 1986; Trochim y McLinden 2017). Su objetivo principal es reunir a los diferentes grupos de interés (*stakeholders*) de una organización o comunidad y **facilitar la expresión de sus ideas** con el fin de construir un **marco conceptual compartido** que permita una evaluación conjunta del problema.

1.1 Características principales del *Concept Mapping*

- **Enfoque participativo:** Involucra activamente a los participantes en todas las etapas del proceso.
- **Método mixto:** Combina técnicas **cualitativas y cuantitativas** para la recolección y análisis de datos.
- **Flexible y accesible:** Ha demostrado ser eficaz para producir resultados útiles de forma relativamente económica.
- **Aplicable a distintos contextos:** Se adapta bien a grupos heterogéneos y a distintas fases de diseño, implementación o evaluación de proyectos.

1.2 Etapas del mapeo conceptual

El proceso se divide en seis etapas:

1. **Definición del enfoque y la pregunta de investigación:** Se informa a los participantes del proceso y se formula conjuntamente el problema o tema de interés.
2. **Lluvia de ideas (*brainstorming*):** Los participantes generan respuestas a una pregunta abierta.
3. **Estructuración y clasificación:** Se organizan y agrupan las ideas recogidas.
4. **Análisis cuantitativo:** Se procesan los datos para generar mapas conceptuales (por ejemplo, mediante técnicas de análisis multivariado).
5. **Interpretación de resultados:** Los participantes analizan e interpretan colectivamente los resultados.
6. **Aplicación de los resultados:** Se acuerda de forma consensuada cómo utilizar los hallazgos obtenidos.

Este enfoque permite que los grupos de interés definan el contenido, interpreten los datos y decidan cómo emplear los resultados, mientras que el equipo investigador facilita el proceso. Aunque no siempre incluye una

fase de acción directa, el mapeo conceptual puede generar insumos valiosos que promuevan transformaciones posteriores en la comunidad u organización.

2 Crear el proyecto y cargar los datos

Para desarrollar nuestro *concept mapping* con `R`, utilizaremos los datos del *sampling project* de la aplicación `RCmap`, desarrollada por Haim Bar (https://haimbar.github.io/RCMap/), traducidos y adaptados al castellano (Bar y Mentch 2017).

Durante la revisión de este conjunto de datos, se identificaron 4 pares de enunciados cuya distancia era cero según las clasificaciones de los 10 participantes. Esto indica que su significado era prácticamente idéntico, por lo que decidimos eliminar uno de cada par, quedándonos con un total de **76 enunciados únicos**, que serán los que analizaremos.

2.1 Organización del proyecto

Comenzamos creando un nuevo proyecto en `R`, al que llamaremos `"Concept Mapping"`. Dentro de este proyecto, organizamos los archivos en las siguientes carpetas:

- `codigo`
- `datos`
- `tablas`
- `resultados`
- `graficos`

2.2 Carga de bibliotecas

Cargamos las bibliotecas necesarias:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggrepel)
library(RColorBrewer)
```

2.3 Importación de los datos

2.3.1 Lista de enunciados

Leemos el archivo con los 76 enunciados que proponen ideas para mejorar el funcionamiento de una empresa:

```
enunciados <- read_csv2("datos/enunciados_76.csv")
```

```
## i Using "',' " as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 76 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (1): enunciado
## dbl (1): ID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
enunciados <- as.data.frame(enunciados)
```

Visualizamos los primeros 20 enunciados en la Tabla 1:

```
knitr::kable(enunciados [0:20,],
             format = "latex",
             booktabs = TRUE,
             caption = "20 primeros enunciados") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

Tabla 1: 20 primeros enunciados

ID	enunciado
1	promocionar la imagen de la organización en lugar de solo programas específicos
2	establecer un enfoque de equipo de "círculo de calidad" para los empleados del programa
3	mejorar los beneficios médicos para los empleados
4	mejorar la comunicación entre los empleados
5	gerentes de programa más amigables
6	reducir informes, memorandos y reuniones innecesarias
7	mejorar la limpieza de las oficinas y lugares del programa
8	informatizar las listas de correo para la comunicación
9	permitir a los empleados opciones de horario flexible
10	realizar análisis de efectividad de todos los programas actuales principales
11	desarrollar una mejor estrategia para determinar los aumentos salariales anuales
12	iniciar un grupo de relaciones laborales y de gestión
13	explorar opciones para instalaciones del programa y expansión de oficinas
14	iniciar un programa para reducir el ausentismo y la rotación de empleados
15	desarrollar una red informática interna para el personal directivo
16	investigar posibles problemas de seguridad en el lugar de trabajo
18	disminuir el tiempo de espera de los clientes
19	desarrollar un programa de incentivos para empleados
20	automatizar todas las fases del sistema de gestión del programa
22	mejorar la información y materiales de los programas

2.3.2 Clasificaciones de los enunciados

A continuación, cargamos las clasificaciones realizadas por los 10 participantes (S1 a S10):

```
clasificaciones <- read_csv("datos/clasificaciones_76.csv")
```

```
## Rows: 76 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): enunciado
## dbl (10): S1, S2, S3, S4, S5, S6, S7, S8, S9, S10
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
clasificaciones <- as.data.frame(clasificaciones)
```

Mostramos una parte del conjunto en la Tabla 2:

```
knitr::kable(clasificaciones [1:20,],
             format = "latex",
             booktabs = TRUE,
             caption = "Clasificaciones de las 76 ideas (truncada)") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

Tabla 2: Clasificaciones de las 76 ideas (truncada)

enunciado	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
E1	1	1	1	1	1	1	1	1	1	1
E2	2	2	2	2	2	2	2	2	2	2
E3	3	3	3	3	3	3	3	3	3	3
E4	4	4	2	4	4	4	4	3	4	2
E5	4	5	4	2	2	5	5	4	5	4
E6	5	4	5	5	5	6	2	5	4	5
E7	6	5	4	2	6	5	5	4	5	4
E8	7	6	6	5	5	6	6	6	4	6
E9	3	3	7	3	3	3	4	3	3	3
E10	1	1	8	6	1	7	1	7	1	7
E11	5	3	5	7	7	8	4	8	3	8
E12	8	3	5	4	8	4	4	8	2	9
E13	9	7	9	8	9	8	7	9	2	10
E14	4	8	7	7	7	8	4	5	3	11
E15	7	6	6	7	10	6	2	6	4	6
E16	2	2	9	8	11	2	7	9	6	12
E18	2	7	5	8	11	9	7	9	6	12
E19	3	9	2	3	7	8	4	3	2	3
E20	7	6	6	5	10	9	6	6	6	6
E22	2	1	1	6	1	9	7	2	6	13

2.3.3 Datos sociodemográficos

Importamos también variables sociodemográficas de los participantes, como el tamaño de la empresa y los años de experiencia:

```
demograficos <- read_csv("datos/demograficos.csv")
```

```
## Rows: 10 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): sujeto, tamano_empresa
## dbl (1): anos
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
demograficos <- as.data.frame(demograficos)
```

```
knitr::kable(demograficos,
             format = "latex",
             booktabs = TRUE,
             caption = "Características sociodemográficas de los sujetos") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

Tabla 3: Características sociodemográficas de los sujetos

sujeto	tamano_empresa	anos
S1	grande	4.2
S2	pequeña	9.9
S3	pequeña	8.2
S4	mediana	6.8
S5	pequeña	7.2
S6	grande	8.6
S7	pequeña	1.2
S8	pequeña	2.3
S9	grande	0.5
S10	mediana	4.6

2.3.4 Puntuaciones de importancia

Los participantes valoraron la importancia de cada idea en una escala Likert de 5 puntos:

```
puntuaciones_importancia <- read_csv("datos/puntuaciones_importancia_76.csv")
```

```
## Rows: 10 Columns: 77
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): sujeto
## dbl (76): E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
puntuaciones_importancia <- as.data.frame(puntuaciones_importancia)
```

```
knitr::kable(puntuaciones_importancia [,1:18],
             format = "latex",
             booktabs = TRUE,
             caption = "Puntuaciones sobre la importancia de las ideas (truncada)") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

Tabla 4: Puntuaciones sobre la importancia de las ideas (truncada)

sujeto	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E18
S1	3	5	3	3	4	5	4	5	2	4	2	4	2	4	2	1	3
S2	4	3	2	1	5	2	3	2	3	4	3	2	1	4	2	3	2
S3	4	4	5	2	2	1	3	4	2	1	2	5	5	4	3	1	4
S4	4	4	4	4	5	5	5	4	4	3	4	3	4	3	3	5	2
S5	1	4	2	3	5	3	4	4	2	4	1	4	4	2	5	3	4
S6	1	4	3	3	4	5	3	5	4	1	4	1	3	5	4	5	1
S7	4	4	5	2	4	1	3	2	3	4	5	5	4	4	3	3	3
S8	4	3	3	3	3	5	2	5	2	2	3	1	3	4	1	3	3
S9	4	1	5	3	2	4	4	4	5	1	5	2	1	1	4	3	2
S10	1	3	4	2	1	4	4	2	2	3	2	2	3	4	2	2	2

2.3.5 Puntuaciones de viabilidad

También valoraron la viabilidad de implementar cada idea:

```
puntuaciones_viabilidad <- read_csv("datos/puntuaciones_viabilidad_76.csv")
```

```
## Rows: 10 Columns: 77
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): sujeto
## dbl (76): E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
puntuaciones_viabilidad <- as.data.frame(puntuaciones_viabilidad)
```

```
knitr::kable(puntuaciones_viabilidad [,1:18],
             format = "latex",
             booktabs = TRUE,
             caption = "Puntuaciones sobre la viabilidad de las ideas (truncada)") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

Tabla 5: Puntuaciones sobre la viabilidad de las ideas (truncada)

sujeto	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E18
S1	1	3	3	4	2	5	3	4	4	4	4	5	5	4	5	1	4
S2	3	5	5	4	3	5	3	4	5	3	4	4	4	3	4	5	4
S3	2	4	3	4	3	4	3	4	3	2	3	4	1	4	4	5	3
S4	2	4	5	5	3	5	3	1	5	2	5	5	3	4	3	4	3
S5	1	4	5	4	4	4	4	3	5	3	5	4	2	3	2	1	4
S6	4	3	3	3	2	5	3	5	4	5	3	1	5	5	4	2	1
S7	2	3	5	4	3	4	3	5	5	2	4	1	1	4	3	3	3
S8	2	3	2	2	2	4	1	2	2	2	2	1	4	3	3	1	3
S9	3	4	3	3	5	4	4	2	3	4	4	2	5	4	4	3	4
S10	3	5	5	5	4	5	4	2	5	2	4	5	2	3	2	1	3

2.4 Preparación del *data frame* final

Vamos a construir un único data frame, llamado `puntuaciones_76`, que contendrá:

- sujeto
- tamano_empresa
- anos (experiencia)
- enunciado
- importancia
- viabilidad

2.4.1 Integración de los datos

Primero, unimos las puntuaciones con los datos sociodemográficos:

```
puntuaciones_importancia <- puntuaciones_importancia %>%
  left_join(demograficos, by = "sujeto")%>%
  relocate(tamano_empresa, anos, .after = sujeto)
```

```
puntuaciones_viabilidad <- puntuaciones_viabilidad %>%
  left_join(demograficos, by = "sujeto")%>%
  relocate(tamano_empresa, anos, .after = sujeto)
```

2.4.2 Transformación a formato largo

Reestructuramos los datos para que cada fila represente una combinación única de sujeto y enunciado:

```
puntuaciones_importancia_long <- puntuaciones_importancia %>%
  pivot_longer(cols = starts_with("E"), names_to = "enunciado", values_to = "importancia")

puntuaciones_viabilidad_long <- puntuaciones_viabilidad %>%
  pivot_longer(cols = starts_with("E"), names_to = "enunciado", values_to = "viabilidad")
```

2.4.3 Fusión de los *data frames*

Combinamos los datos de importancia y viabilidad:

```
puntuaciones_76 <- puntuaciones_importancia_long %>%
  inner_join(puntuaciones_viabilidad_long, by = c("sujeto", "enunciado"))
```

Eliminamos las columnas duplicadas:

```
puntuaciones_76 <- puntuaciones_76 %>%
  dplyr::select(-tamano_empresa.y, -anos.y) %>%
  dplyr::rename(
    tamano_empresa = tamano_empresa.x,
    anos = anos.x
  )
```

Visualizamos una muestra del resultado en la Tabla 6.

```
knitr::kable(puntuaciones_76 [1:10,],
             format = "latex",
             booktabs = TRUE,
             caption = "Importancia y viabilidad de los enunciados (truncada)") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

2.4.4 Guardado del archivo final

Finalmente, exportamos el *data frame* `puntuaciones_76` a un archivo `.csv`:

Tabla 6: Importancia y viabilidad de los enunciados (truncada)

sujeto	tamano_empresa	anos	enunciado	importancia	viabilidad
S1	grande	4.2	E1	3	1
S1	grande	4.2	E2	5	3
S1	grande	4.2	E3	3	3
S1	grande	4.2	E4	3	4
S1	grande	4.2	E5	4	2
S1	grande	4.2	E6	5	5
S1	grande	4.2	E7	4	3
S1	grande	4.2	E8	5	4
S1	grande	4.2	E9	2	4
S1	grande	4.2	E10	4	4

```
write_csv(puntuaciones_76, file= "datos/puntuaciones_76.csv")
```

3 Crear la matriz de proximidades

Para poder representar gráficamente los mapas conceptuales, primero necesitamos construir una **matriz de proximidades** entre los enunciados, a partir de las clasificaciones individuales realizadas por los participantes.

Definimos dos funciones para ello:

1. `crear_matriz_proximidad()` genera una matriz binaria de proximidad entre los enunciados para un solo sujeto.
2. `crear_matriz_proximidad_total()` agrupa las matrices individuales para obtener una matriz promedio.

```
crear_matriz_proximidad <- function(clasificaciones, sujeto) {
  enunciados <- as.character(clasificaciones$enunciado)
  matriz <- matrix(0, nrow = length(enunciados), ncol = length(enunciados),
                   dimnames = list(enunciados, enunciados))
  clasificacion <- clasificaciones[[sujeto]]
  for (grupo in unique(clasificacion)) {
    indices <- which(clasificacion == grupo)
    if (length(indices) > 1) {
      for (i in 1:(length(indices) - 1)) {
        for (j in (i + 1):length(indices)) {
          a1 <- enunciados[indices[i]]
          a2 <- enunciados[indices[j]]
          matriz[a1, a2] <- 1
          matriz[a2, a1] <- 1
        }
      }
    }
  }
  return(matriz)
}

crear_matriz_proximidad_total <- function(clasificaciones) {
  sujetos <- colnames(clasificaciones)[-1]
  n_sujetos <- length(sujetos)
```

```
  enunciados <- as.character(clasificaciones$enunciado)
  matriz_suma <- matrix(0, nrow = length(enunciados), ncol = length(enunciados),
                        dimnames = list(enunciados, enunciados))
  for (sujeto in sujetos) {
    matriz_sujeto <- crear_matriz_proximidad(clasificaciones, sujeto)
    matriz_suma <- matriz_suma + matriz_sujeto
  }
  diag(matriz_suma) <- n_sujetos
  return(matriz_suma / n_sujetos)
}
matriz_proximidades <- crear_matriz_proximidad_total(clasificaciones)
```

Convertimos la matriz resultante en *data frame* para facilitar su visualización:

```
matriz_proximidades <- as.data.frame(matriz_proximidades)
```

A continuación, mostramos las primeras filas de la matriz:

```
  knitr::kable(matriz_proximidades [1:20, 1:14],
               format = "latex",
               booktabs = TRUE,
               caption = "Matriz de proximidades entre las ideas (truncada)") %>%
    kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                              font_size = 8)
```

Tabla 7: Matriz de proximidades entre las ideas (truncada)

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
E1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0
E2	0.0	1.0	0.0	0.2	0.2	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0
E3	0.0	0.0	1.0	0.1	0.0	0.0	0.0	0.0	0.8	0.0	0.2	0.1	0.0	0.1
E4	0.0	0.2	0.1	1.0	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.3	0.0	0.2
E5	0.0	0.2	0.0	0.1	1.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.1
E6	0.0	0.1	0.0	0.2	0.0	1.0	0.0	0.4	0.0	0.0	0.2	0.1	0.0	0.1
E7	0.0	0.1	0.0	0.0	0.8	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
E8	0.0	0.0	0.0	0.1	0.0	0.4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
E9	0.0	0.0	0.8	0.2	0.0	0.0	0.0	0.0	1.0	0.0	0.3	0.2	0.0	0.3
E10	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
E11	0.0	0.0	0.2	0.1	0.0	0.2	0.0	0.0	0.3	0.0	1.0	0.4	0.1	0.5
E12	0.0	0.1	0.1	0.3	0.0	0.1	0.0	0.0	0.2	0.0	0.4	1.0	0.1	0.1
E13	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	1.0	0.1
E14	0.0	0.0	0.1	0.2	0.1	0.1	0.0	0.0	0.3	0.0	0.5	0.1	0.1	1.0
E15	0.0	0.1	0.0	0.1	0.0	0.3	0.0	0.7	0.0	0.0	0.1	0.0	0.0	0.1
E16	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0
E18	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.4	0.0
E19	0.0	0.2	0.4	0.3	0.0	0.0	0.0	0.0	0.5	0.0	0.3	0.2	0.2	0.3
E20	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0
E22	0.3	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.1	0.0

Convertimos la matriz de proximidades en una **matriz de distancias**, que se utilizará para los análisis posteriores.

```
matriz_distancias <- (1 - matriz_proximidades)
```

Guardamos ambas matrices en archivos `.RDta` para poder reutilizarlas fácilmente:

```r
save(matriz_proximidades, file = "tablas/matriz_proximidades_76.RDta")
save(matriz_distancias, file = "tablas/matriz_distancias_76.RDta")
```

4 Escalado multidimensional (MDS)

En esta sección, aplicamos un **escalado multidimensional no métrico (nMDS)** para representar visualmente las distancias entre los enunciados en un espacio bidimensional.

Primero, cargamos la matriz de distancias:

```r
load("tablas/matriz_distancias_76.RDta")
matriz_distancias <- as.matrix(matriz_distancias)
```

Ejecutamos el escalado con `isoMDS`:

```r
fit <- isoMDS(matriz_distancias, k = 2)
```

```
## initial  value 36.549329
## iter   5 value 32.208532
## iter  10 value 30.149720
## final  value 29.760814
## converged
```

```r
x <- fit$points[, 1]
y <- fit$points[, 2]
fit
```

```
## $points
##            [,1]         [,2]
## E1   0.33625447 -0.098679602
## E2   0.07829846  0.148927677
## E3  -0.27783842 -0.113415757
## E4  -0.09653035  0.085557281
## E5   0.06224058 -0.278239587
## E6   0.12410387  0.367937456
## E7   0.07093852 -0.613045326
## E8   0.52404570  0.551656320
## E9  -0.98643297 -0.293195858
## E10  0.75036023 -0.625592858
## E11 -0.68348991  0.170837513
## E12 -0.83229317  0.511301670
## E13 -0.27550149  0.914247494
## E14 -0.65650907  0.001688157
## E15  0.46742142  0.564842900
## E16  0.27372762  1.113261522
## E18  0.10912085  1.028726031
## E19 -0.82339729  0.014078707
## E20  0.60640847  0.755418312
## E22  0.91765539  0.035022143
## E23  0.36821014  1.130597523
## E24 -0.93271104 -0.568965610
## E25  1.15411736 -0.420537585
## E26  0.33746475  0.912815862
## E27 -0.93940787 -0.034271592
## E29 -0.38910861  0.323173500
## E30  0.56328455 -0.141500487
```

```
## E31 -1.00060727 -0.190973838
## E32  0.26646034 -1.065184130
## E33  0.74101749 -0.213361060
## E34  1.17792000 -0.359056375
## E35 -0.75742045 -0.545780679
## E36  0.53279139 -0.987346895
## E37  0.90673248 -0.507164075
## E38  0.50281963  0.953620988
## E39 -0.74497950  0.175235491
## E40  0.20490798 -1.130104088
## E41 -0.41211421  0.272259752
## E42 -0.38390080  0.074171374
## E43 -0.10760222 -1.056111763
## E44  0.17072248 -0.946230529
## E45  0.89153217 -0.431249150
## E47 -0.65259835 -0.017394063
## E48 -0.69260989  0.662229350
## E49 -0.08180478 -0.747131479
## E50  0.78085828  0.675643162
## E51  0.72295707  0.620638619
## E52  0.82822889  0.697091064
## E53  0.72744712  0.643994275
## E54 -0.86958184  0.350875874
## E55 -0.38696984  0.484927725
## E56 -0.78782383  0.351776757
## E57 -0.70346926 -0.292990550
## E58 -0.37289593  0.552980575
## E59 -0.55026276  0.591249053
## E60 -0.97217827 -0.483110255
## E61  0.43051624 -0.996782071
## E62  1.15646853 -0.237313411
## E63 -0.77822014 -0.321423622
## E64 -0.94787119  0.244242640
## E65 -0.03821010  0.865403025
## E66  0.02407328 -1.070030701
## E67  0.77171873  0.238596705
## E68  0.32441982 -0.796748979
## E69 -0.10466405  0.546746703
## E70 -0.10633109 -0.435958088
## E71 -0.08556585 -0.345413769
## E72  0.04663615  0.812395999
## E73  0.42498382 -0.377469076
## E74 -0.30615072  0.663823457
## E75  0.96704728 -0.391797492
## E76  0.71423171 -0.705051528
## E77 -1.02694707 -0.341037901
## E78 -0.28711378 -0.638247962
## E79 -0.21987949 -0.667121471
## E80  0.21284962  0.377036605
##
## $stress
## [1] 29.76081
```

El valor de stress obtenido es del 29,8%, lo que supera el umbral del 20% generalmente aceptado. Esto indica

que el ajuste no es óptimo, por lo que debemos interpretar con cautela los resultados.

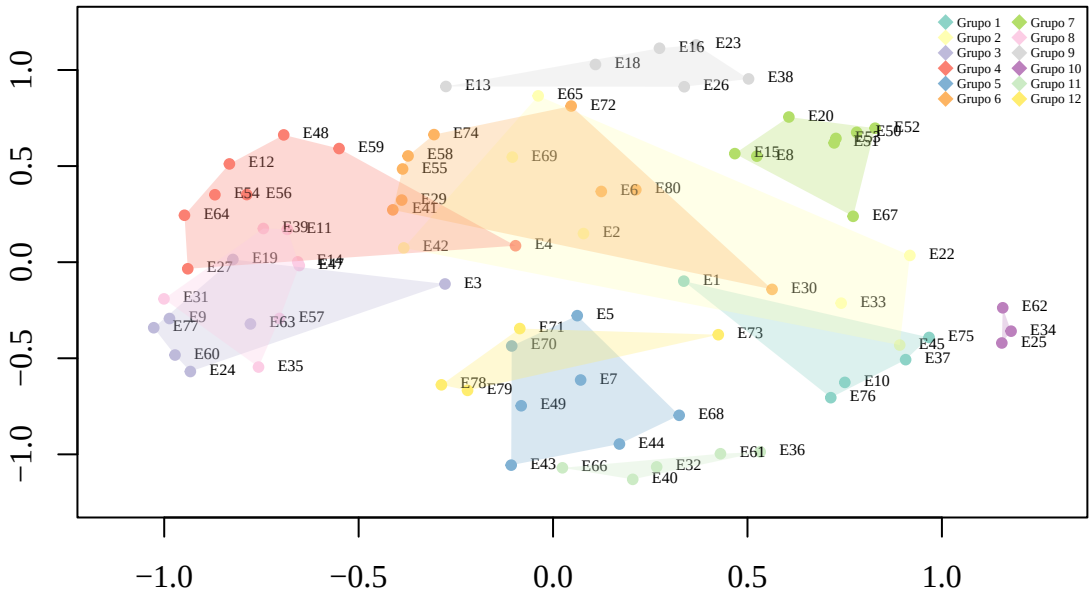
Para identificar grupos de enunciados similares, realizamos un **análisis de conglomerados jerárquico** con el método `ward.D`:

```
matriz_distancias_dist <- as.dist(matriz_distancias)
hclust_ward <- hclust(matriz_distancias_dist, method = "ward.D")
clusters <- cutree(hclust_ward, k = 12) # cortamos el dendrogama en 12 grupos
palette <- brewer.pal(12, "Set3") # elegimos una paleta de colores
```

Graficamos los resultados del MDS, coloreando los puntos según el grupo al que pertenecen.

```
par(family = "Times", font.main = 1)  # Establecer parámetros gráficos globales
plot(x, y, pch = 20,
     col = palette[clusters],
     xlab = "", ylab = "",
     main = "Figura 1: Similitud entre enunciados (MDS)",
     cex.main = 0.8,
     xlim = range(x) + c(-0.1, 0.1),
     ylim = range(y) + c(-0.1, 0.1)
)
text(x, y, labels = rownames(matriz_distancias), pos = 4, cex = 0.5)

# Añadimos polígonos que encierran los puntos de cada grupo

for (i in 1:12) {
  idx <- which(clusters == i)
  if (length(idx) > 2) {
    ch <- chull(x[idx], y[idx])
    ch <- c(ch, ch[1])  # cerrar el polígono
    polygon(x[idx][ch], y[idx][ch], col = adjustcolor(palette[i], alpha.f = 0.3),
            border = NA)
  }
}

# Agregamos una leyenda para indicar el grupo al que pertenece cada color

legend("topright",
       legend = paste("Grupo", 1:12),
       col = palette,
       pch = 18,
       pt.cex = 1.1,
       cex = 0.4,
       bty = "n",
       ncol = 2
)
```

Figura 1: Similitud entre enunciados (MDS)



5 Análisis de conglomerados

5.1 Utilizando la biblioteca `cluster`

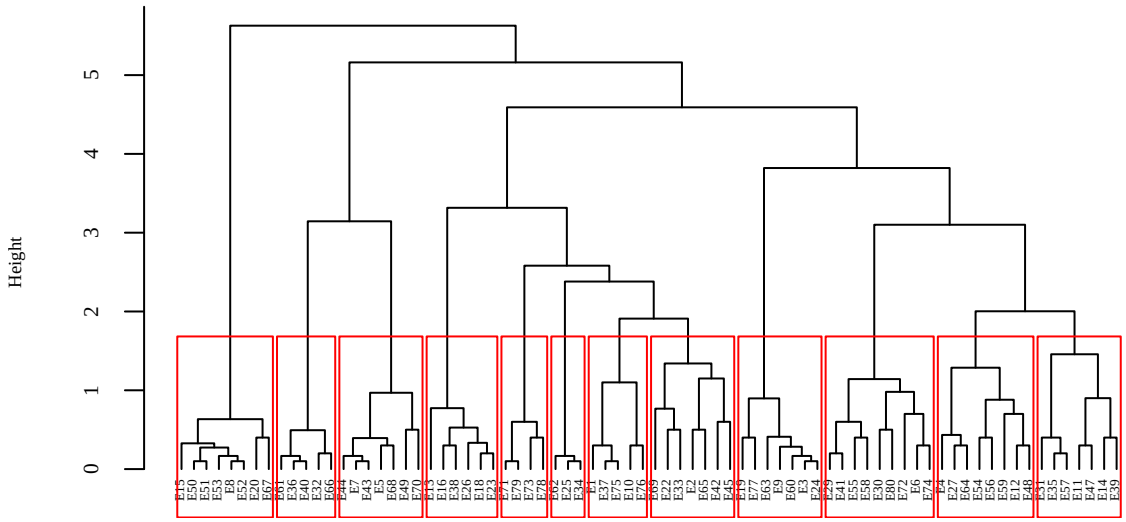
Comenzamos cargando la matriz de distancias previamente guardada:

```
load("tablas/matriz_distancias_76.RDta")
matriz_distancias <- as.dist(matriz_distancias)
```

Realizamos el análisis de conglomerados jerárquico con el método `ward.D` y graficamos el dendrograma correspondiente:

```
hclust_ward <- hclust(matriz_distancias, method = 'ward.D')

plot(hclust_ward,
     hang = -1,
     cex = 0.4,
     main = expression("Figura 2: Similitud entre enunciados (Dedrograma)"),
     cex.main = 0.8,
     family = "Times",
     xlab = "",
     sub = "",
     cex.lab = 0.6,
     cex.axis = 0.7
)
grupos <- cutree(hclust_ward, k = 12)
rect.hclust(hclust_ward, k = 12, border = "red")
```

Figura 2: Similitud entre enunciados (Dedrograma)



5.2 Utilizando la biblioteca `factoextra`.

Aseguramos el formato adecuado de la matriz de distancias:

```
load("tablas/matriz_distancias_76.RDta")
matriz_distancias <- as.dist(matriz_distancias)
```

Visualizamos el dendrograma con mejor presentación gracias a la función `fviz_dend`:

```
hclust_ward <- hclust(matriz_distancias, method = "ward.D")

fviz_dend(hclust_ward,
          k = 12,
          rect = TRUE,
          show_labels = TRUE,
          cex = 0.4,
          palette = "Set2",
          main = "",
          sub = "",
          xlab = "",
          ylab = "Altura"
)
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

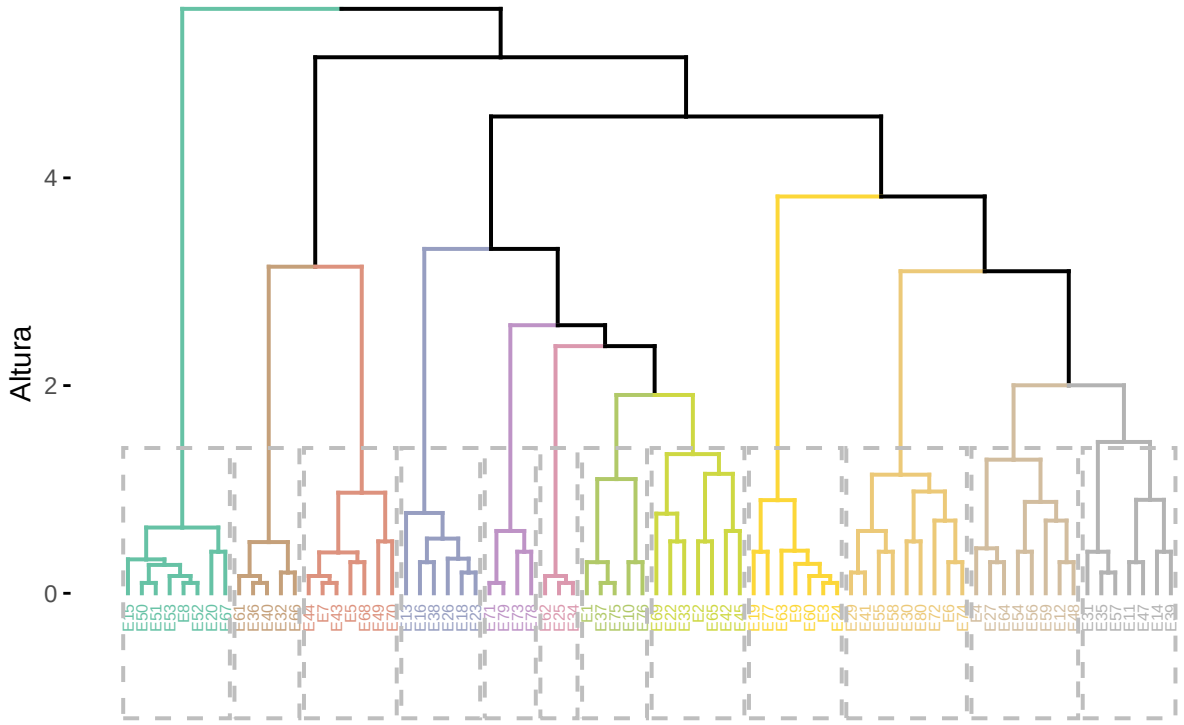


Figura 3.1: Similitud entre enunciados (Dendrograma)

```
fviz_dend(hclust_ward,
          k = 12,
          rect = TRUE,
          show_labels = TRUE,
          cex = 0.5,
          palette = "Set2",
          main = "",
          xlab = "",
          ylab = "",
          horiz = FALSE,
          type = "circular",
          repel = TRUE
) +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank())
```

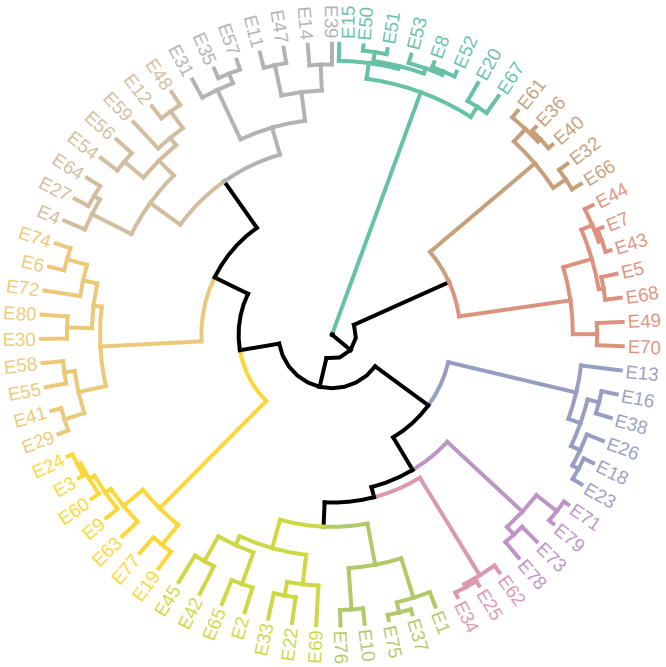


Figura 3.2: Similitud entre enunciados (Dendrograma circular)

6 Gráfico de coordenadas paralelas

Esta técnica permite visualizar y comparar las puntuaciones promedio asignadas a cada *cluster* por diferentes grupos de participantes y según distintas variables de evaluación (por ejemplo, importancia y viabilidad). En el contexto del *concept mapping*, este tipo de visualización se conoce como *pattern matching*.

Con esta herramienta es posible representar simultáneamente más de dos combinaciones de grupo y variable, facilitando el análisis de patrones de respuesta. Por ejemplo, se pueden comparar las valoraciones de viabilidad realizadas por empresas de distinto tamaño, o bien observar cómo las personas con más experiencia califican la viabilidad de los clusters frente a cómo las personas con menos experiencia evalúan su importancia.

6.1 Cargar los datos

```
puntuaciones_76 <- read_csv("datos/puntuaciones_76.csv")
```

```
## Rows: 760 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): sujeto, tamano_empresa, enunciado
## dbl (3): anos, importancia, viabilidad
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

6.2 Agrupación por conglomerados

Cargamos la matriz de distancias y realizamos el análisis de conglomerados

```
load("tablas/matriz_distancias_76.RDta")
hclust_ward <- hclust(as.dist(matriz_distancias), method = "ward.D")
```

```r
clusters <- cutree(hclust_ward, k = 12)
names(clusters) <- rownames(matriz_distancias)

puntuaciones_76 <- puntuaciones_76 %>%
  mutate(cluster = as.factor(clusters[enunciado]))
```

6.3 Cálculo de promedios por grupo

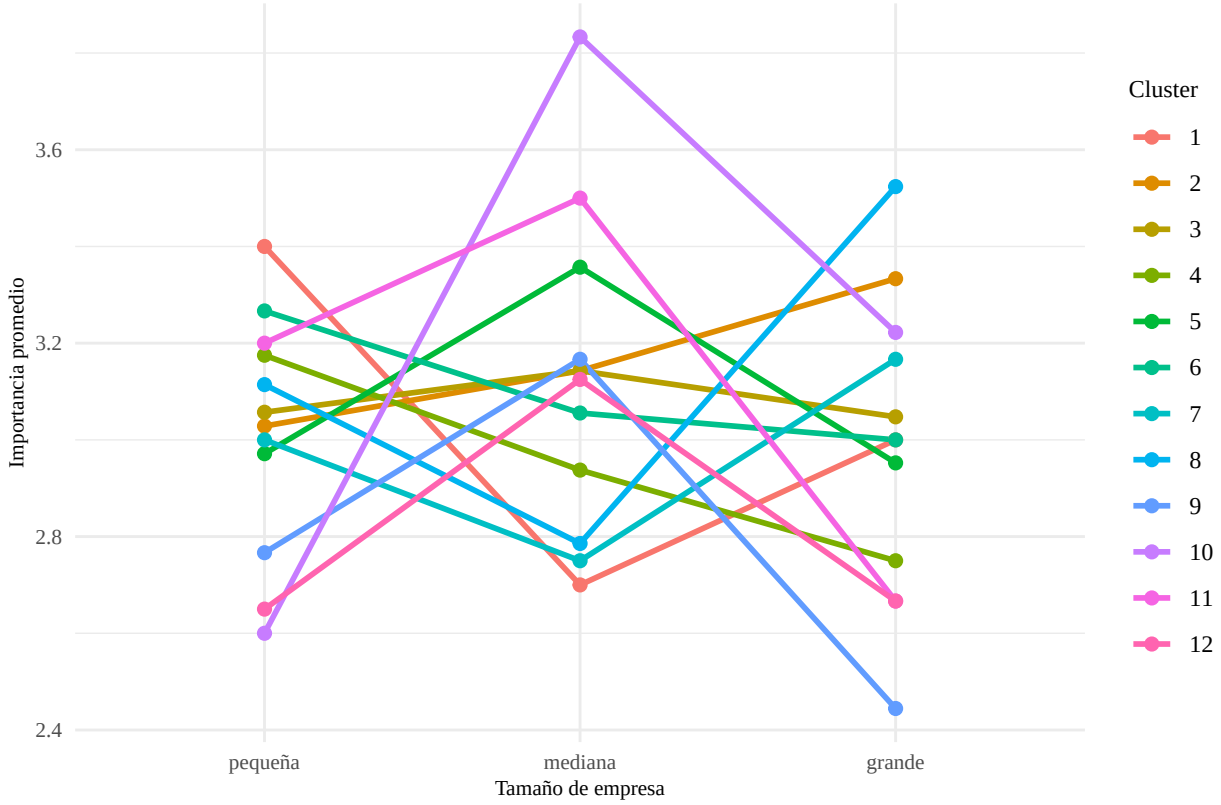
```r
resumen_clusters <- puntuaciones_76 %>%
  group_by(cluster, tamano_empresa) %>%
  summarise(
    importancia = mean(importancia, na.rm = TRUE),
    viabilidad = mean(viabilidad, na.rm = TRUE),
    .groups = "drop"
  )

resumen_clusters$tamano_empresa <- factor(
  resumen_clusters$tamano_empresa,
  levels = c("pequeña", "mediana", "grande")
)
```

6.4 Gráfico de coordenadas paralelas: Importancia

```r
ggplot(resumen_clusters, aes(x = tamano_empresa, y = importancia,
                             group = cluster, color = cluster)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Figura 4: Importancia promedio por cluster y tamaño de empresa",
    x = "Tamaño de empresa",
    y = "Importancia promedio",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times"),
    plot.title = element_text(size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 9),
    legend.text = element_text(size = 9)
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

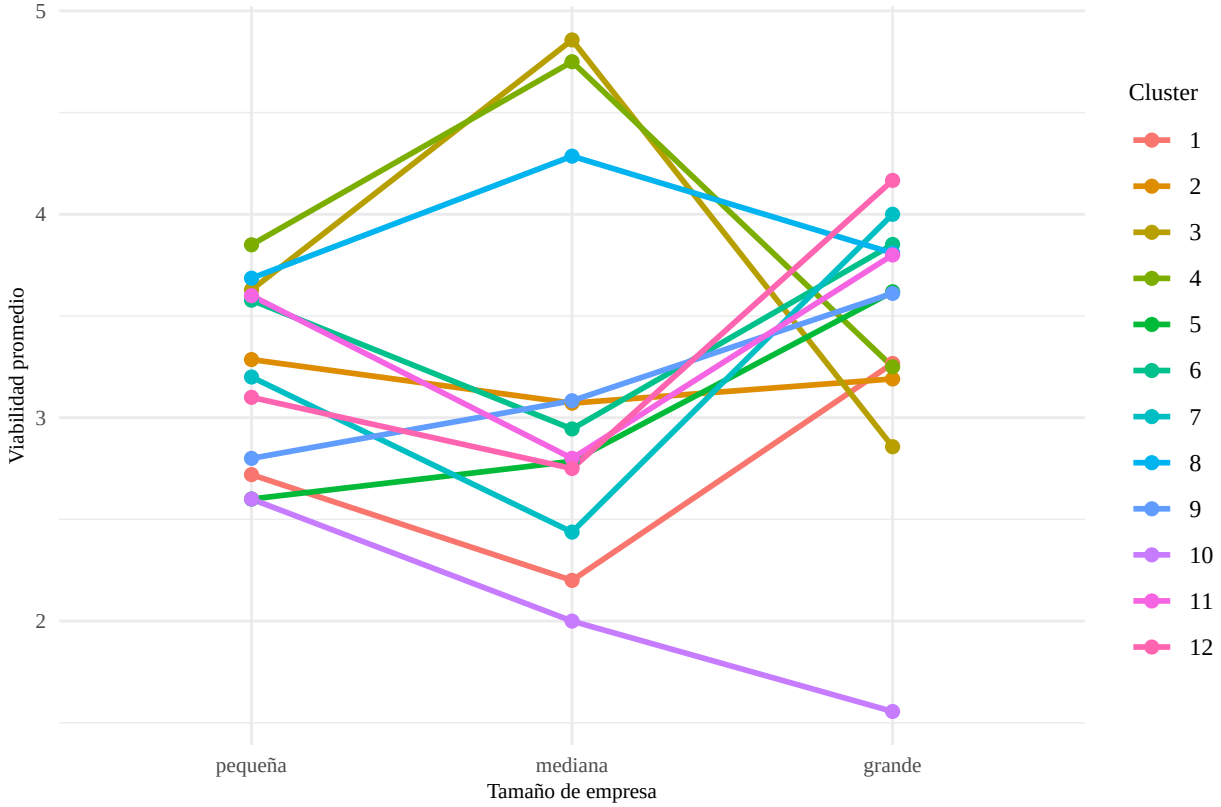
Figura 4: Importancia promedio por cluster y tamaño de empresa



6.5 Gráfico de coordenadas paralelas: Viabilidad.

```
ggplot(resumen_clusters, aes(x = tamano_empresa, y = viabilidad,
                             group = cluster, color = cluster)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Figura 5: Viabilidad promedio por cluster y tamaño de empresa",
    x = "Tamaño de empresa",
    y = "Viabilidad promedio",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times"),
    plot.title = element_text(size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 9),
    legend.text = element_text(size = 9)
  )
```

Figura 5: Viabilidad promedio por cluster y tamaño de empresa



6.6 Comparación conjunta: Importancia y Viabilidad normalizadas

Primero, transformamos los datos a formato largo y escalamos las variables:

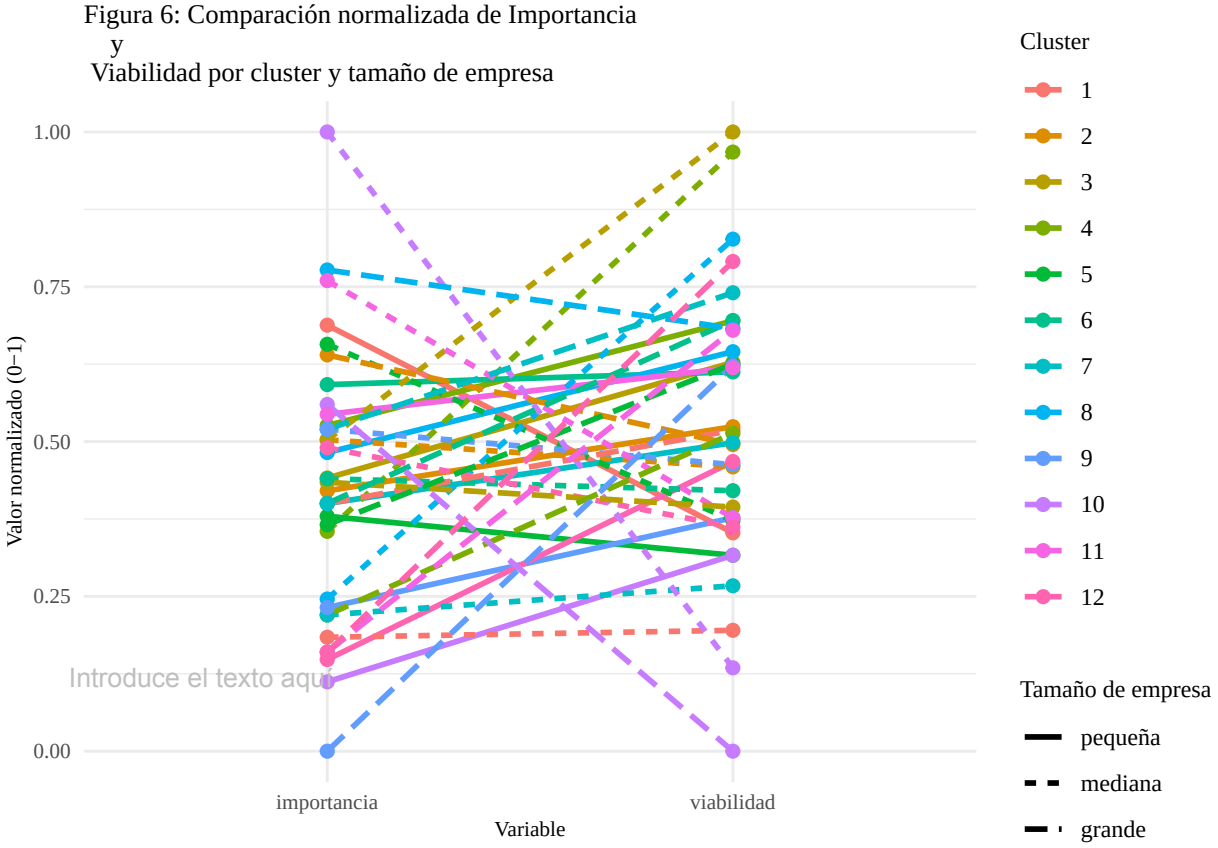
```
resumen_largo <- resumen_clusters %>%
  pivot_longer(cols = c(importancia, viabilidad),
               names_to = "variable",
               values_to = "valor") %>%
  group_by(variable) %>%
  mutate(valor_escalado = (valor - min(valor)) / (max(valor) - min(valor))) %>%
  ungroup()
```

Luego graficamos:

```
ggplot(resumen_largo, aes(x = variable, y = valor_escalado,
                          group = interaction(cluster, tamano_empresa), color = cluster)) +
  geom_line(aes(linetype = tamano_empresa), size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Figura 6: Comparación normalizada de Importancia
    y\n Viabilidad por cluster y tamaño de empresa",
    x = "Variable",
    y = "Valor normalizado (0-1)",
    color = "Cluster",
    linetype = "Tamaño de empresa"
  ) +
  theme_minimal() +
  theme(
```

```
    text = element_text(family = "Times"),
    plot.title = element_text(size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 9),
    legend.text = element_text(size = 9)
  )
```

Figura 6: Comparación normalizada de Importancia y Viabilidad por cluster y tamaño de empresa



7 Gráficos *Go-Zone* o de dispersión bivariada

Un gráfico *Go-Zone*, en el contexto de un *concept mapping*, es una representación gráfica bidimensional que permite visualizar la valoración de los enunciados según dos variables —usualmente, **importancia** y **viabilidad**. Cada enunciado se representa como un punto en el plano cartesiano, lo que facilita observar su posición relativa en ambas dimensiones.

El gráfico se divide en cuatro cuadrantes mediante las medias generales de cada eje. Esta división permite identificar fácilmente los enunciados que destacan tanto por su alta importancia como por su alta viabilidad (cuadrante superior derecho). Además, los puntos pueden colorearse según el *cluster* al que pertenece cada enunciado, lo que facilita la interpretación temática y la priorización dentro del mapa conceptual.

Es importante señalar que el gráfico *Go-Zone* refleja únicamente diferencias descriptivas entre grupos, pero **no permite inferir si esas diferencias son estadísticamente significativas**. Para ello, es necesario realizar pruebas estadísticas —como la prueba t que veremos más adelante— que consideran la variabilidad de los datos y el tamaño muestral.

7.1 Gráfico *Go-Zone* de los enunciados: importancia vs. viabilidad

Cargamos los datos.

```
puntuaciones_76 <- read_csv("datos/puntuaciones_76.csv")
```

```
## Rows: 760 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): sujeto, tamano_empresa, enunciado
## dbl (3): anos, importancia, viabilidad
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Calculamos los promedios de importancia y viabilidad por enunciado:

```
promedios <- puntuaciones_76 %>%
  group_by(enunciado) %>%
  summarise(
    importancia = mean(importancia, na.rm = TRUE),
    viabilidad = mean(viabilidad, na.rm = TRUE)
  )
```

Cargamos la matriz de distancias y realizamos el análisis de conglomerados jerárquico.

```
load("tablas/matriz_distancias_76.RDta")
hclust_ward <- hclust(as.dist(matriz_distancias), method = "ward.D")
clusters <- cutree(hclust_ward, k = 12)
names(clusters) <- rownames(matriz_distancias)
```

Asignamos los clusters a los promedios:

```
promedios$cluster <- as.factor(clusters[promedios$enunciado])
```

Calculamos las medias globales para definir los cuadrantes.

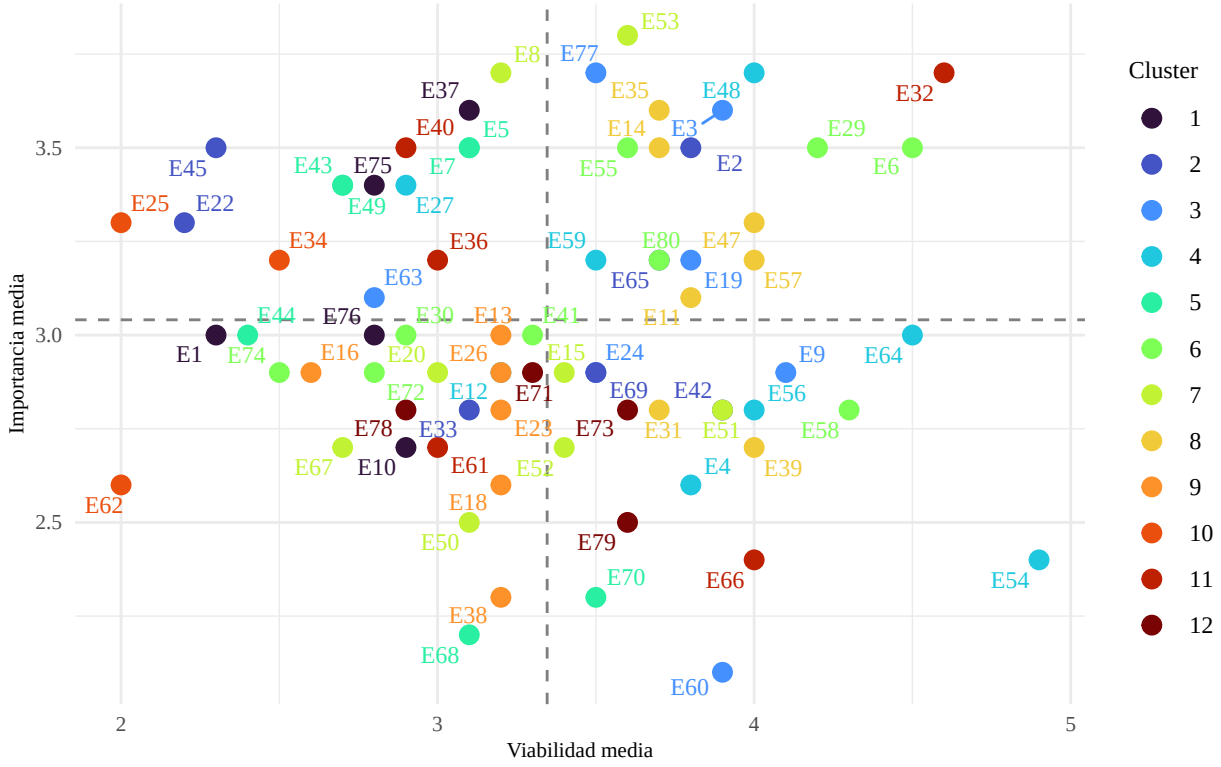
```
media_importancia <- mean(promedios$importancia, na.rm = TRUE)
media_viabilidad <- mean(promedios$viabilidad, na.rm = TRUE)
```

Generamos el gráfico:

```
ggplot(promedios, aes(x = viabilidad, y = importancia, color = cluster,
                      label = enunciado)) +
  geom_vline(xintercept = media_viabilidad, linetype = "dashed", color = "gray50") +
  geom_hline(yintercept = media_importancia, linetype = "dashed", color = "gray50") +
  geom_point(size = 3) +
  geom_text_repel(size = 3, max.overlaps = 100, family = "Times") +
  scale_color_viridis_d(option = "turbo") +
  labs(
    title = "Figura 7: Gráfico Go-Zone: Importancia vs. Viabilidad",
    subtitle = "División en cuadrantes según medias globales",
    x = "Viabilidad media",
    y = "Importancia media",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times"),
```

```
    plot.title = element_text(size = 10),
    plot.subtitle = element_text(size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 9),
    legend.text = element_text(size = 9)
  )
```

Figura 7: Gráfico Go−Zone: Importancia vs. Viabilidad
División en cuadrantes según medias globales



7.2 Gráfico *Go-Zone* de los enunciados: importancia vs. experiencia (años en la empresa)

Cargamos los datos:

```
puntuaciones_76 <- read_csv("datos/puntuaciones_76.csv")
```

```
## Rows: 760 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): sujeto, tamano_empresa, enunciado
## dbl (3): anos, importancia, viabilidad
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Calculamos la mediana de los años de experiencia:

```
mediana_anos <- median(puntuaciones_76$anos, na.rm = TRUE)
```

Creamos una nueva variable categórica (`anosfac`) según la experiencia:

- **Más experiencia**: `E` = por encima de la mediana.
- **Menos experiencia**: `D` = por debajo de la mediana.

```
puntuaciones_76 <- puntuaciones_76 %>%
  mutate(anosfac = factor(ifelse(anos > mediana_anos, "E", "D")))
```

Cargamos la matriz de distancias y los *clusters*:

```
load("tablas/matriz_distancias_76.RDta")
hclust_ward <- hclust(as.dist(matriz_distancias), method = "ward.D")
clusters <- cutree(hclust_ward, k = 12)
names(clusters) <- rownames(matriz_distancias)
```

Calculamos la importancia media por enunciado y grupo de experiencia:

```
resumen_enunciado <- puntuaciones_76 %>%
  group_by(enunciado, anosfac) %>%
  summarise(importancia = mean(importancia, na.rm = TRUE), .groups = "drop") %>%
  pivot_wider(names_from = anosfac, values_from = importancia) %>%
  rename(importancia_E = E, importancia_D = D)
```

Asignamos los *clusters*:

```
resumen_enunciado <- resumen_enunciado %>%
  mutate(cluster = as.factor(clusters[enunciado]))
```

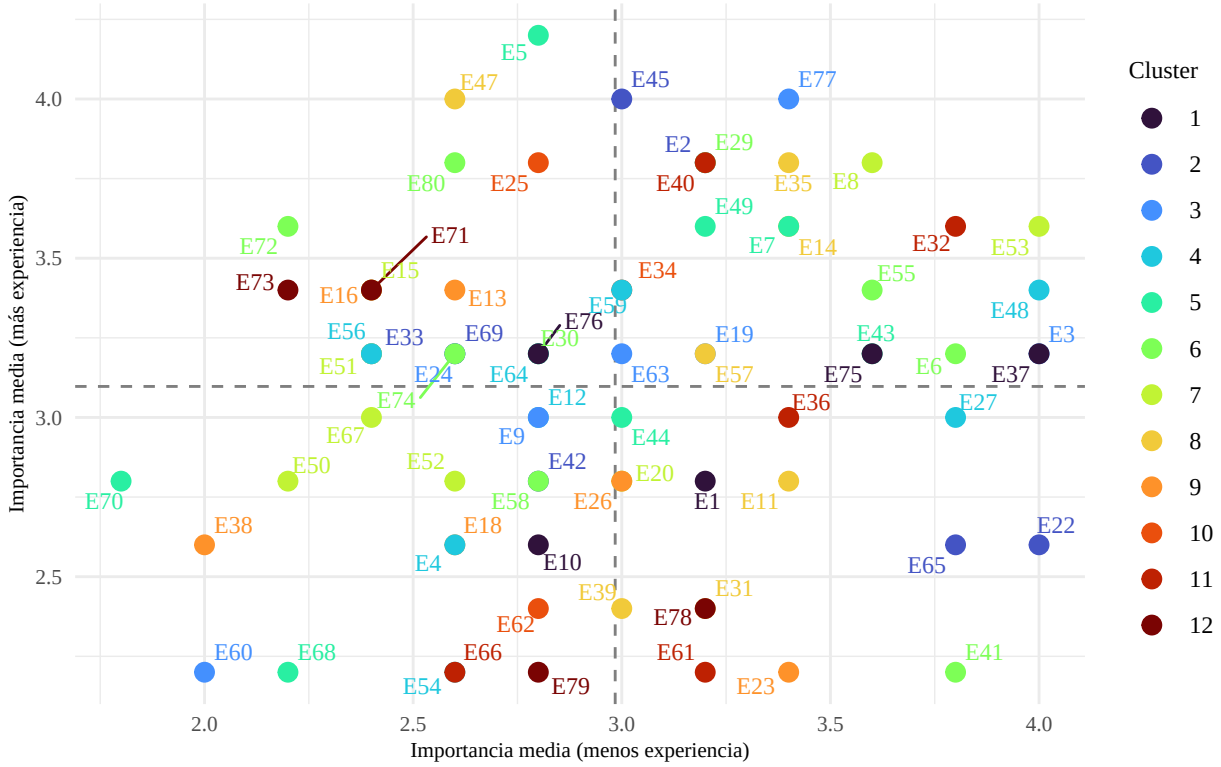
Calculamos las medias para los cuadrantes.

```
media_x <- mean(resumen_enunciado$importancia_D, na.rm = TRUE)
media_y <- mean(resumen_enunciado$importancia_E, na.rm = TRUE)
```

Graficamos.

```
ggplot(resumen_enunciado, aes(x = importancia_D, y = importancia_E,
                              color = cluster, label = enunciado)) +
  geom_vline(xintercept = media_x, linetype = "dashed", color = "gray50") +
  geom_hline(yintercept = media_y, linetype = "dashed", color = "gray50") +
  geom_point(size = 3) +
  geom_text_repel(size = 3, max.overlaps = 100, family = "Times") +
  scale_color_viridis_d(option = "turbo") +
  labs(
    title = "Figura 8: Gráfico Go-Zone por Enunciado",
    subtitle = "Importancia media según experiencia (años), coloreado por cluster",
    x = "Importancia media (menos experiencia)",
    y = "Importancia media (más experiencia)",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times"),
    plot.title = element_text(size = 10),
    plot.subtitle = element_text(size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 9),
    legend.text = element_text(size = 9)
  )
```

Figura 8: Gráfico Go−Zone por Enunciado
Importancia media según experiencia (años), coloreado por cluster



7.3 Gráfico *Go-Zone* de los *clusters*: importancia vs. experiencia

Cargamos los datos y preparamos la variable experiencia.

```
puntuaciones_76 <- read_csv("datos/puntuaciones_76.csv")
```

```
## Rows: 760 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): sujeto, tamano_empresa, enunciado
## dbl (3): anos, importancia, viabilidad
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mediana_anos <- median(puntuaciones_76$anos, na.rm = TRUE)

puntuaciones_76 <- puntuaciones_76 %>%
  mutate(anosfac = factor(ifelse(anos > mediana_anos, "E", "D")))
```

Cargamos la matriz de distancias y calculamos los *clusters*:

```
load("tablas/matriz_distancias_76.RDta")
hclust_ward <- hclust(as.dist(matriz_distancias), method = "ward.D")
clusters <- cutree(hclust_ward, k = 12)
names(clusters) <- rownames(matriz_distancias)
```

Asignamos los *clusters* al *dataset*:

```
puntuaciones_76 <- puntuaciones_76 %>%
  mutate(cluster = as.factor(clusters[enunciado]))
```

Calculamos la importancia media por cluster y grupo de experiencia:

```
resumen <- puntuaciones_76 %>%
  group_by(cluster, anosfac) %>%
  summarise(importancia = mean(importancia, na.rm = TRUE), .groups = "drop") %>%
  pivot_wider(names_from = anosfac, values_from = importancia) %>%
  rename(importancia_E = E, importancia_D = D)
```

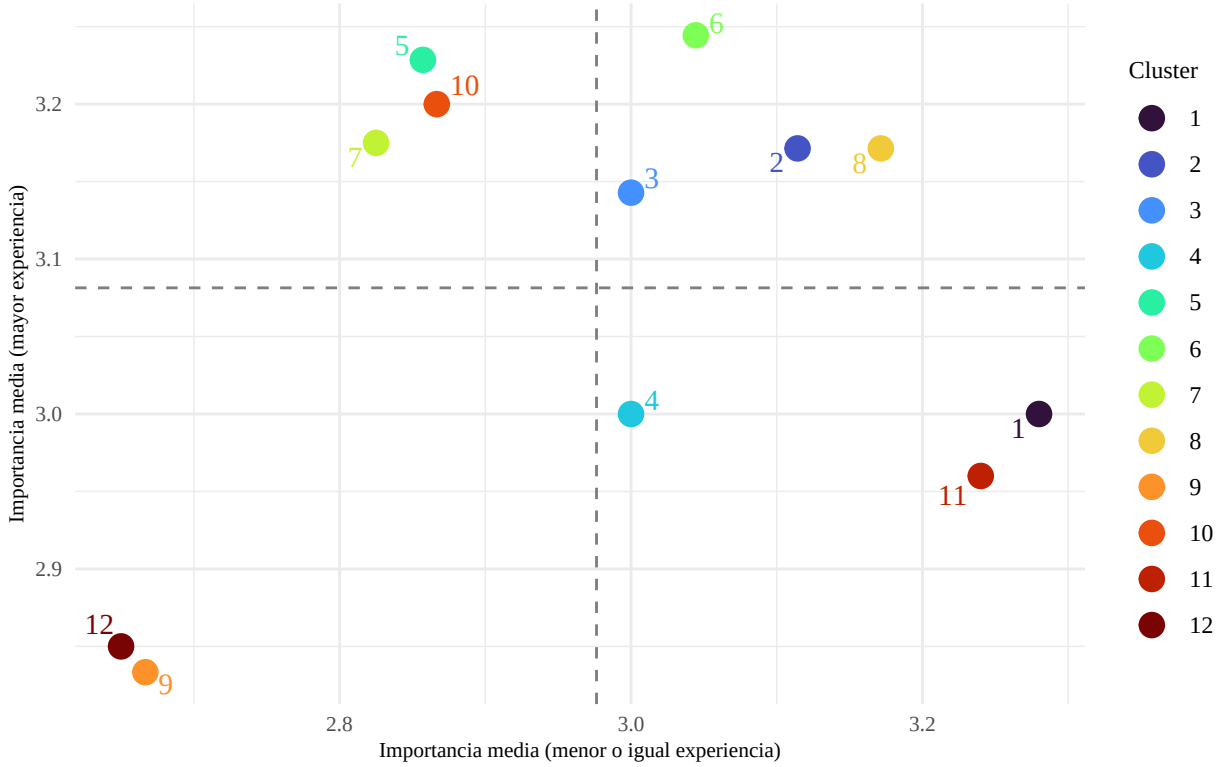
Calculamos las medias generales para los cuadrantes.

```
media_x <- mean(resumen$importancia_D, na.rm = TRUE)
media_y <- mean(resumen$importancia_E, na.rm = TRUE)
```

Graficamos.

```
ggplot(resumen, aes(x = importancia_D, y = importancia_E,
                    label = cluster, color = cluster)) +
  geom_vline(xintercept = media_x, linetype = "dashed", color = "gray50") +
  geom_hline(yintercept = media_y, linetype = "dashed", color = "gray50") +
  geom_point(size = 4) +
  geom_text_repel(size = 4, max.overlaps = 100, family = "Times") +
  scale_color_viridis_d(option = "turbo") +
  labs(
    title = "Figura 9: Gráfico Go-Zone por Clusters",
    subtitle = "Importancia media según experiencia (años)",
    x = "Importancia media (menor o igual experiencia)",
    y = "Importancia media (mayor experiencia)",
    color = "Cluster"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times"),
    plot.title = element_text(size = 10),
    plot.subtitle = element_text(size = 10),
    axis.title = element_text(size = 8),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 9),
    legend.text = element_text(size = 9)
  )
```

Figura 9: Gráfico Go−Zone por Clusters
Importancia media según experiencia (años)



8 Análisis estadístico de las puntuaciones

8.1 Calculo de media y desviación estándar de la importancia para todos los sujetos

Primero, calculamos la media y la desviación estándar de la puntuación de **importancia** para cada enunciado considerando todos los sujetos.

```
estadisticos_importancia <- puntuaciones_importancia %>%
  dplyr::select(starts_with("E")) %>%
  summarise(across(everything(),
                   list(media = ~ mean(.x, na.rm = TRUE),
                        sd = ~ sd(.x, na.rm = TRUE)))) %>%
  pivot_longer(cols = everything(),
               names_to = c("Enunciado", ".value"),
               names_sep = "_")

estadisticos_importancia
```

```
## # A tibble: 76 x 3
##    Enunciado media    sd
##    <chr>     <dbl> <dbl>
##  1 E1          3   1.41
##  2 E2          3.5 1.08
##  3 E3          3.6 1.17
##  4 E4          2.6 0.843
##  5 E5          3.5 1.43
```

```
##  6 E6          3.5 1.65
##  7 E7          3.5 0.850
##  8 E8          3.7 1.25
##  9 E9          2.9 1.10
## 10 E10         2.7 1.34
## # i 66 more rows
```

Guardamos los resultados en un archivo `csv`.

```
write_csv(estadisticos_importancia, "resultados/estadisticos_importancia.csv")
```

8.2 Calculo de media y desviación estándar de importancia por grupos

A continuación, analizamos las puntuaciones de importancia según la experiencia de los sujetos, medida en años en la empresa. Para ello, dicotomizamos la variable:

- **Más experiencia (E)**: por encima de la mediana.
- **Menos experiencia (D)**: por debajo de la mediana.

Añadimos la variable dicotómica `anosfac` al conjunto de datos:

```
puntuaciones_importancia <- puntuaciones_importancia %>%
  mutate(anosfac = factor(ifelse(anos > mediana_anos, "E", "D")))
```

Calculamos los estadísticos por grupo:

```
estadisticos_importancia_experiencia <- puntuaciones_importancia %>%
  group_by(anosfac) %>% # Agrupar por la variable experiencia
  summarise(across(starts_with("E"),
                   list(media = ~ mean(.x, na.rm = TRUE),
                        sd = ~ sd(.x, na.rm = TRUE)),
                   .names = "{.col}_{.fn}")) %>%
  pivot_longer(cols = -anosfac, # Transformar las columnas en formato largo
               names_to = c("enunciado", ".value"),
               names_sep = "_")
estadisticos_importancia_experiencia
```

```
## # A tibble: 152 x 4
##    anosfac enunciado media    sd
##    <fct>   <chr>     <dbl> <dbl>
##  1 D       E1          3.2 1.30
##  2 D       E2          3.2 1.48
##  3 D       E3          4   1
##  4 D       E4          2.6 0.548
##  5 D       E5          2.8 1.30
##  6 D       E6          3.8 1.64
##  7 D       E7          3.4 0.894
##  8 D       E8          3.6 1.52
##  9 D       E9          2.8 1.30
## 10 D       E10         2.8 1.30
## # i 142 more rows
```

Guardamos los resultados:

```
write_csv(estadisticos_importancia_experiencia,
          "resultados/estadisticos_importancia_experiencia.csv")
```

8.3 Comparación de medias por enunciados (según experiencia)

Creamos un *data frame* para almacenar los resultados de las comparaciones de medias:

```
resultados <- tibble(
  Enunciado = character(),
  diff_media = numeric(),
  t_value = numeric(),
  df = numeric(),
  p_value = numeric(),
  CI_lower = numeric(),
  CI_upper = numeric(),
  cohen_d = numeric()
)
```

Reorganizamos los datos para el análisis:

```
enunciados <- puntuaciones_importancia %>%
  dplyr::select(anosfac, starts_with("E")) %>%
  pivot_longer(cols = -anosfac, names_to = "Enunciado", values_to = "Puntuacion")
```

Ejecutamos una prueba t para cada enunciado:

```
for (enun in unique(enunciados$Enunciado)) {
  # Filtrar datos para el enunciado actual
  datos <- enunciados %>% filter(Enunciado == enun)

  # Realizar la prueba de igualdad de varianzas
  var_test <- var.test(Puntuacion ~ anosfac, data = datos)

  # Realizar el t-test dependiendo del resultado de igualdad de varianzas
  t_test <- t.test(
    Puntuacion ~ anosfac,
    data = datos,
    var.equal = var_test$p.value > 0.05 # Si p > 0.05, asumir varianzas iguales
  )

  # Calcular Cohen's d
  mean_diff <- diff(t_test$estimate) # Diferencia de medias
  pooled_sd <- sqrt(
    ((var(datos$Puntuacion[datos$anosfac == "E"]) * (sum(datos$anosfac == "E") - 1)) +
       (var(datos$Puntuacion[datos$anosfac == "D"]) * (sum(datos$anosfac == "D") - 1))) /
      (sum(datos$anosfac == "E") + sum(datos$anosfac == "D") - 2)
  )
  cohen_d <- mean_diff / pooled_sd

  # Agregar resultados al data frame
  resultados <- resultados %>%
    add_row(
      Enunciado = enun,
      diff_media = mean_diff,
      t_value = t_test$statistic,
      df = t_test$parameter,
      p_value = t_test$p.value,
      CI_lower = t_test$conf.int[1],
      CI_upper = t_test$conf.int[2],
```

```
      cohen_d = cohen_d
    )
}

# Aplicamos corrección de valores p por comparaciones múltiples
resultados <- resultados %>%
  mutate(
    p_holm = p.adjust(p_value, method = "holm")
  )
```

Mostramos una parte de los resultados:

```
knitr::kable(resultados [1:20,],
             format = "latex",
             booktabs = TRUE,
             caption = "Importancia por experiencia en la empresa,
             comparación de medias (truncada)") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

Tabla 8: Importancia por experiencia en la empresa, comparación de medias (truncada)

Enunciado	diff_media	t_value	df	p_value	CI_lower	CI_upper	cohen_d	p_holm
E1	-0.4	0.4264014	8.000000	0.6810572	-1.7632236	2.5632236	-0.2696799	1
E2	0.6	-0.8660254	4.721311	0.4282809	-2.4130401	1.2130401	0.5477226	1
E3	-0.8	1.0886621	8.000000	0.3080085	-0.8945600	2.4945600	-0.6885304	1
E4	0.0	0.0000000	8.000000	1.0000000	-1.3044729	1.3044729	0.0000000	1
E5	1.4	-1.6977494	8.000000	0.1279881	-3.3015797	0.5015797	1.0737510	1
E6	-0.6	0.5523448	8.000000	0.5958023	-1.9049617	3.1049617	-0.3493335	1
E7	0.2	-0.3535534	8.000000	0.7328099	-1.5044729	1.1044729	0.2236068	1
E8	0.2	-0.2390457	8.000000	0.8170802	-2.1293415	1.7293415	0.1511858	1
E9	0.2	-0.2721655	8.000000	0.7923872	-1.8945600	1.4945600	0.1721326	1
E10	-0.2	0.2236068	8.000000	0.8286677	-1.8625528	2.2625528	-0.1414214	1
E11	-0.6	0.6708204	8.000000	0.5212275	-1.4625528	2.6625528	-0.4242641	1
E12	0.2	-0.1961161	8.000000	0.8494094	-2.5516720	2.1516720	0.1240347	1
E13	0.8	-0.9428090	8.000000	0.3733749	-2.7567094	1.1567094	0.5962848	1
E14	0.2	-0.2540003	8.000000	0.8059018	-2.0157495	1.6157495	0.1606439	1
E15	1.0	-1.3867505	8.000000	0.2029340	-2.6628832	0.6628832	0.8770580	1
E16	1.0	-1.1785113	8.000000	0.2724557	-2.9567094	0.9567094	0.7453560	1
E18	0.0	0.0000000	8.000000	1.0000000	-1.4944615	1.4944615	0.0000000	1
E19	0.0	0.0000000	8.000000	1.0000000	-1.9015797	1.9015797	0.0000000	1
E20	-0.2	0.2324953	8.000000	0.8219909	-1.7836998	2.1836998	-0.1470429	1
E22	-1.4	1.7232809	8.000000	0.1231312	-0.4734066	3.2734066	-1.0898985	1

Y los guardamos:

```
write_csv(resultados, "resultados/diferencias_medias_importancia_experiencia.csv")
```

Conclusión: No se encontraron diferencias estadísticamente significativas en las puntuaciones de importancia de ningún enunciado según la experiencia en la empresa.

8.4 Comparación de medias por clusters (según experiencia)

Podemos comparar las medias de los clusters siguiendo el mismo procedimiento.

Creamos el *data frame* para almacenar los resultados:

```r
resultados_cluster <- tibble(
  Cluster = character(),
  diff_media = numeric(),
  t_value = numeric(),
  df = numeric(),
  p_value = numeric(),
  CI_lower = numeric(),
  CI_upper = numeric(),
  cohen_d = numeric()
)
```

Agrupamos las puntuaciones por *cluster*, sujeto y experiencia.

```r
datos_clusters <- puntuaciones_76 %>%
  group_by(sujeto, cluster, anosfac) %>%
  summarise(puntuacion = mean(importancia, na.rm = TRUE), .groups = "drop")
```

Ejecutamos la prueba t para cada *cluster*:

```r
for (cl in unique(datos_clusters$cluster)) {
  datos <- datos_clusters %>% filter(cluster == cl)

  # Prueba de igualdad de varianzas
  var_test <- var.test(puntuacion ~ anosfac, data = datos)

  # T-test
  t_test <- t.test(
    puntuacion ~ anosfac,
    data = datos,
    var.equal = var_test$p.value > 0.05
  )

  # Calcular diferencia de medias y Cohen's d
  mean_diff <- diff(t_test$estimate)
  pooled_sd <- sqrt(
    ((var(datos$puntuacion[datos$anosfac == "E"]) * (sum(datos$anosfac == "E") - 1)) +
       (var(datos$puntuacion[datos$anosfac == "D"]) * (sum(datos$anosfac == "D") - 1))) /
      (sum(datos$anosfac == "E") + sum(datos$anosfac == "D") - 2)
  )
  cohen_d <- mean_diff / pooled_sd

  # Guardar resultados
  resultados_cluster <- resultados_cluster %>%
    add_row(
      Cluster = as.character(cl),
      diff_media = mean_diff,
      t_value = t_test$statistic,
      df = t_test$parameter,
      p_value = t_test$p.value,
      CI_lower = t_test$conf.int[1],
      CI_upper = t_test$conf.int[2],
      cohen_d = cohen_d
    )
}
```

```
# Aplicar corrección de valores p (Holm)
resultados_cluster <- resultados_cluster %>%
  mutate(
    p_holm = p.adjust(p_value, method = "holm")
  )
```

Mostramos una parte de los resultados:

```
knitr::kable(resultados_cluster, format = "latex", booktabs = TRUE,
             caption = "Comparación de medias por cluster según experiencia") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"), font_size = 8)
```

Tabla 9: Comparación de medias por cluster según experiencia

Cluster	diff_media	t_value	df	p_value	CI_lower	CI_upper	cohen_d	p_holm
1	-0.2800000	0.5937323	8.000000	0.5690945	-0.8074955	1.3674955	-0.3755092	1
2	0.0571429	-0.2279212	8.000000	0.8254250	-0.6352887	0.5210030	0.1441500	1
3	0.1428571	-0.4662524	8.000000	0.6534672	-0.8494040	0.5636897	0.2948839	1
4	0.0000000	0.0000000	8.000000	1.0000000	-0.4557641	0.4557641	0.0000000	1
5	0.3714286	-1.8571429	4.428868	0.1298959	-0.9061408	0.1632836	1.1745603	1
6	0.2000000	-0.9233805	8.000000	0.3828143	-0.6994700	0.2994700	0.5839971	1
7	0.3500000	-0.9027562	8.000000	0.3930248	-1.2440414	0.5440414	0.5709532	1
8	0.0000000	0.0000000	8.000000	1.0000000	-0.6056481	0.6056481	0.0000000	1
9	0.1666667	-0.5423261	8.000000	0.6023692	-0.8753436	0.5420103	0.3429972	1
10	0.3333333	-0.7624929	8.000000	0.4676498	-1.3414320	0.6747654	0.4822428	1
11	-0.2800000	0.7107423	8.000000	0.4974384	-0.6284603	1.1884603	-0.4495129	1
12	0.2000000	-0.5929995	8.000000	0.5695614	-0.9777424	0.5777424	0.3750458	1

Guardamos los resultados:

```
write_csv(resultados_cluster, "resultados/diferencias_medias_clusters_experiencia.csv")
```

Conclusión: No se encontraron diferencias estadísticamente significativas en las puntuaciones de importancia de ningún cluster según la experiencia en la empresa

8.5 ANOVA: Comparación por tamaño de empresa

Preparamos los datos en formato largo:

```
datos_largos <- puntuaciones_76 %>%
  dplyr::select(sujeto, tamano_empresa, enunciado, importancia)
```

Creamos un *data frame* para almacenar los resultados del ANOVA:

```
resultados_anova <- tibble(
  Enunciado = character(),
  Media_Grande = numeric(),
  Media_Mediana = numeric(),
  Media_Pequena = numeric(),
  F_value = numeric(),
  p_value = numeric(),
  eta2 = numeric()
)
```

Realizamos el ANOVA para cada enunciado:

```r
# Lista de enunciados únicos
enunciados_unicos <- unique(datos_largos$enunciado)

for (enun in enunciados_unicos) {
  datos <- datos_largos %>% filter(enunciado == enun)

  # ANOVA
  modelo <- aov(importancia ~ tamano_empresa, data = datos)
  resumen <- summary(modelo)

  # Extraemos sumas de cuadrados
  ss_efecto <- resumen[[1]]$`Sum Sq`[1]
  ss_total <- sum(resumen[[1]]$`Sum Sq`)
  eta2 <- ss_efecto / ss_total

  # Extraemos valores
  F_value <- resumen[[1]]$`F value`[1]
  p_value <- resumen[[1]]$`Pr(>F)`[1]

  # Medias por grupo
  medias <- datos %>%
    group_by(tamano_empresa) %>%
    summarise(Media = mean(importancia, na.rm = TRUE)) %>%
    pivot_wider(names_from = tamano_empresa, values_from = Media,
                names_prefix = "Media_") %>%
    rename_with(~ gsub("ñ", "n", .x))  # Normalizamos nombres si es necesario

  # Agregamos al data frame
  resultados_anova <- resultados_anova %>%
    add_row(
      Enunciado = enun,
      Media_Grande = medias$Media_grande %||% NA,
      Media_Mediana = medias$Media_mediana %||% NA,
      Media_Pequena = medias$Media_pequena %||% NA,
      F_value = F_value,
      p_value = p_value,
      eta2 = eta2
    )
}
```

Visualizamos una parte de los resultados:

```r
knitr::kable(resultados_anova [1:20,],
             format = "latex",
             booktabs = TRUE,
             caption = "Importancia por tamaño de la empresa, ANOVA (truncada)") %>%
  kableExtra::kable_styling(latex_options = c("striped", "hold_position"),
                            font_size = 8)
```

Guardamos los resultados:

```r
write_csv(resultados_anova, "resultados/anova_importancia_tamano_empresa.csv")
```

Conclusión: No se encontraron diferencias estadísticamente significativas en las puntuaciones de importancia según el tamaño de la empresa del sujeto para ninguno de los enunciados analizados.

Tabla 10: Importancia por tamaño de la empresa, ANOVA (truncada)

Enunciado	Media_Grande	Media_Mediana	Media_Pequena	F_value	p_value	eta2
E1	2.666667	2.5	3.4	0.3492872	0.7168150	0.0907407
E2	3.333333	3.5	3.6	0.0450161	0.9562566	0.0126984
E3	3.666667	4.0	3.4	0.1573034	0.8573827	0.0430108
E4	3.000000	3.0	2.2	1.1666667	0.3653545	0.2500000
E5	3.333333	3.0	3.8	0.2070611	0.8177754	0.0558559
E6	4.666667	4.5	2.4	3.4339623	0.0913699	0.4952381
E7	3.666667	4.5	3.0	3.6842105	0.0807071	0.5128205
E8	4.666667	3.0	3.4	1.5016892	0.2866353	0.3002364
E9	3.666667	3.0	2.4	1.3495763	0.3193559	0.2782875
E10	2.000000	3.0	3.0	0.5250000	0.6131370	0.1304348
E11	3.666667	3.0	2.8	0.3243534	0.7333057	0.0848126
E12	2.333333	2.5	3.4	0.4827586	0.6362008	0.1212121
E13	2.000000	3.5	3.4	1.2863248	0.3343725	0.2687500
E14	3.333333	3.5	3.6	0.0377358	0.9631618	0.0106667
E15	3.333333	2.5	2.8	0.2729805	0.7688510	0.0723514
E16	3.000000	3.5	2.6	0.2675159	0.7727612	0.0710059
E18	2.000000	2.0	3.2	2.6250000	0.1410480	0.4285714
E19	2.333333	3.0	3.8	1.5281690	0.2813867	0.3039216
E20	3.666667	3.0	2.4	0.8946629	0.4508153	0.2035794
E22	3.666667	4.5	2.6	1.6226415	0.2636388	0.3167587

Referencias bibliográficas

Bar, Haim, y Lucas Mentch. 2017. «R-CMap—An open-source software for concept mapping». *Evaluation and Program Planning* 60: 284-92.

Kane, Mary, y William MK Trochim. 2007. *Concept mapping for planning and evaluation.* Sage.

Trochim, William MK, y Rhoda Linton. 1986. «Conceptualization for planning and evaluation». *Evaluation and program planning* 9 (4): 289-308.

Trochim, William MK, y Daniel McLinden. 2017. «Introduction to a special issue on concept mapping». *Evaluation and program planning* 60: 166-75.