Estadística aplicada con R: visualización y validación de datos poblacionales pragmáticos y fonéticos

Adrián Cabedo Nebot

Table of contents

Material de carácter gratuito diseñado para el uso de investigadores en lingüística, sobre todo a quienes se inician en la investigación con datos.

Email: adrian.cabedo@uv.es

1 Sobre estos materiales

Este libro de materiales, con finalidad didáctica, ofrece un primer acercamiento al análisis y explotación de bases de datos lingüísticas con R; con ello se pretende ofrecer un modo dee superar las limitaciones de herramientas convencionales como Excel y Google Sheets. Los lectores adquirirán una mejor comprensión de R, un lenguaje de programación dirigido al análisis de datos. Además, explorarán técnicas estadísticas avanzadas para visualizar y analizar datos lingüísticos y poblacionales.

Los contenidos incluyen:

> Sobre R
>
> - **Introducción a R**: fundamentos del lenguaje y su entorno de programación.
> - **Visualización de datos con GGplot2**: creación de visualizaciones descriptivas atracticas.
> - **Análisis categórico**: aplicación de Mosaicplot y pruebas de chi cuadrado para investigar relaciones categóricas entre variables nominales.
> - **Análisis multidimensional**: identificación de patrones en datos mediante análisis de correspondencias múltiple.

- **Árboles de decisión**: utilización para tomar decisiones basadas en datos y explorar relaciones no lineales.
- **Mapas de calor**: visualización de correlaciones y tendencias en datos numéricos.

! Fuente de referencia y novedades

Los materiales ofrecen de un modo didáctico y resumido aspectos abordados en esta referencia bibliográfica (sobre todo, los referentes a la sección de estadística inferencial).

- Cabedo Nebot, A. (2021). *Fundamentos de estadística con R para lingüistas*. Tirant Lo Blanch.

Todo lo relacionado con la presentación de la visualización descriptiva, las tareas de limpieza o filtrado y los ejercicios de estos materiales son documentación nueva.

1.1 Objetivos específicos

A partir de los contenidos anteriores, por tanto, pueden desarrollase los siguientes objetivos:

Justificación de R

- Adquirir habilidades avanzadas en el manejo de bases de datos lingüísticas más allá de las hojas de cálculo tradicionales como Excel o Google Sheets.
- Familiarizarse con el programa R, aprendiendo los conceptos básicos de programación y análisis de datos en este entorno.
- Desarrollar la capacidad de representar datos de manera efectiva utilizando técnicas de visualización avanzadas, incluyendo barras, lolipops, diagramas de caja y líneas temporales utilizando GGplot2 en R.
- Adquirir estrategias de análisis de datos, como el uso de diversas técnicas estadísticas y de visualización; por ejemplo, Mosaicplot y pruebas de chi cuadrado para explorar relaciones entre variables categóricas, análisis de correspondencias múltiples y análisis de componentes para identificar patrones en datos multidimensionales, la construcción de árboles de decisiones para tomar decisiones basadas en datos y la exploración de relaciones no lineales, así como la generación de mapas de calor para visualizar patrones de correlación y tendencias en datos numéricos.

1.2 Conocimientos previos

Se recomienda a las personas interesadas en seguir estos materiales que tengan un conocimiento básico de programas de hojas de datos como, por ejemplo, Excel o, al menos, que conozcan su estructura general. También es recomendable que hayan realizado investigaciones previas con datos y que conozcan los fundamentos de la estadística y del método científico.

Generalmente, el investigador que se inicia en el análisis de datos lingüísticos no tiene una formación previa en estadística. Por ello, este material se ha diseñado para ser accesible a personas sin experiencia previa en el uso de R, pero que deseen adquirir habilidades avanzadas en el análisis de datos mediante lenguaje de programación. Aun así, es recomendable conocer la estructura de filas, columnas y celdas de una hoja de cálculo.

1.3 Formación previa en estadística general y el método científico

> Sobre cuestiones básicas en estadística
>
> Este libro se dirige sobre todo al análisis estadísticio mediante R y supone que el lector dispone de conocimientos básicos de estadística y del método de análisis científico. Si no es así, se comentan algunas cuestiones generales en este seminario que impartí para la Asociación de Jóvenes Lingüistas:
> https://acabedo.quarto.pub/seminario_ajl_2024/

1.4 Requisitos técnicos

Para seguir estos materiales es necesario que el usuario instale R (https://cran.rediris.es/) y RStudio (https://posit.co/download/rstudio-desktop) en su ordenador. Ambos son programas gratuitos y pueden instalarse en Linux, Windows y Mac.

También existe la posibilidad de acceder a una versión gratuita online de RStudio en https://posit.cloud/.

1.5 Sobre los datos utilizados de ejemplo

En este documentoo, emplearemos datos lingüísticos, específicamente datos pragmáticos y fonéticos, como ejemplos prácticos para aprender a usar R y desarrollar habilidades avanzadas en estadística. No obstante, es crucial entender que el objetivo principal de este libro va más allá de los datos lingüísticos. Las técnicas y pruebas que se enseñarán aquí son universales y aplicables a una variedad de datos en distintos campos y disciplinas. Nuestro propósito es formar investigadores de datos competentes y versátiles, capaces de abordar y resolver

problemas utilizando R y métodos estadísticos, sin importar el tipo de datos con el que trabajen en el futuro.

1.6 Bibliografía recomendada

Referencias recomendas

- Este mismo documento.
- Cabedo Nebot, A. (2021). *Fundamentos de estadística con R para lingüistas*. Tirant Lo Blanch.
- Gries, S. Th. (2021). *Statistics for Linguistics with R: A Practical Introduction*. De Gruyter. https://doi.org/10.1515/9783110718256
- Levshina, N. (2015). *How to do Linguistics with R: Data exploration and statistical analysis*. John Benjamins Publishing Company.
- Moore, D. S., & McCabe, G. P. (1999). *Introduction to the practice of statistics*. W.H. Freeman.
- Navarro, D. (2015). *Learning statistics with R: A tutorial for psychology students and other beginners. (*. University of Adelaide. https://learningstatisticswithr.com/

1.7 ¿Dónde voy cuando me atasco y no consigo generar algún gráfico, calcular algún método estadístico...?

Cuando te atascas, lo mejor es buscar ayuda en la comunidad de R. Hay muchos foros y sitios web donde puedes encontrar respuestas a tus preguntas. Algunos de los lugares más populares para obtener ayuda con R son:

Fuentes de ampliación y ayuda

https://chatgpt.com/?oai-dm=1
https://stackoverflow.com/
https://forum.posit.co/

2 Y hubo un principio...

Todo en la vida tiene un principio. También la investigación en datos; hay un momento concreto en el que simplemente se necesita la cuantificación de los datos para poder analizarlos. No obstante, en investigación filológica o lingüística se tiene muchas veces una consideración negativa hacia la estadística, aunque es un mal positivamente necesario. La estadística es una herramienta que nos permite cuantificar y analizar datos de manera objetiva y rigurosa. Sin ella, no podríamos hacer inferencias válidas sobre nuestras muestras de datos ni sacar conclusiones significativas de ellos.

Si es necesario aplicar estadística en nuestra investigación, es importante hacerlo de manera correcta y rigurosa. Esto implica entender los conceptos básicos de la estadística, saber cómo aplicar los métodos estadísticos adecuados a nuestros datos y ser capaz de interpretar los resultados de manera adecuada.

2.1 ¿Qué hago en mi investigación?

Muchos jóvenes lingüistas (y no tan jóvenes) se plantean qué estudiar, qué investigar, qué hacer con sus datos. Sobre todo, esto suele suceder al terminar el grado y comenzar estudios de máster o de doctorado. Por tanto, lo importante aquí, más incluso que el estudio estadístico, es saber qué investigar y de qué manera. La elección de un tema o disciplina será, normalmente, la que nos determinará el futuro profesional. Por tanto, es importante elegir bien.

Images created by an AI from OpenAI

2.2 ¿Por qué analizar datos?

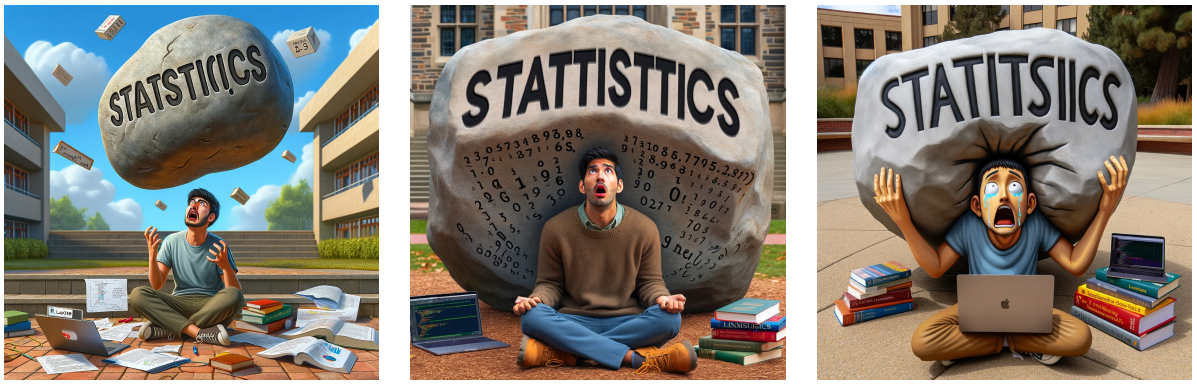
El análisis de datos es muy habitual en la actualidad. Grandes compañías como Google, Facebook, Amazon o Netflix utilizan el análisis de datos para tomar decisiones estratégicas y mejorar sus productos y servicios. En el ámbito académico, el análisis de datos es esencial para la investigación científica y la toma de decisiones basada en evidencias. En la vida cotidiana, el análisis de datos nos ayuda a entender mejor el mundo que nos rodea y a tomar decisiones informadas sobre nuestra salud, finanzas, educación y otros aspectos de nuestra vida.

Image created by an AI from OpenAI

2.3 ¿Cómo siente un lingüista la estadística?

Las tres imágenes de más abajo, generadas con inteligencia artificial, reflejan el terror que muchos estudiantes de lingüística sienten al enfrentarse a la estadística. Sin embargo, la estadística es una herramienta poderosa que puede ayudarnos a entender mejor los datos y a tomar decisiones informadas en nuestra investigación. Con la formación adecuada y la práctica, la estadística puede ser una aliada valiosa en nuestro trabajo como lingüistas.

Los investigadores jóvenes sienten una cierta traición hacia sí mismos cuando se enfrentan a la estadística. Se trata de una disciplina que, en principio, no tiene nada que ver con la lingüística, pero que puede ser necesaria en algún momento de la carrera investigadora.



Images created by an AI from OpenAI

3 Sobre R

R es un lenguaje de programación y un entorno de desarrollo para análisis estadístico y visualización de datos. Es un software libre y de código abierto que se utiliza ampliamente en la investigación científica, la academia y la industria para realizar análisis de datos, modelado estadístico, visualización de datos y generación de informes. Las posibilidades de uso son muy amplias:

Funcionalidades de R

- **Análisis estadístico**: desde análisis descriptivos básicos hasta modelos estadísticos avanzados y pruebas de hipótesis.
- **Visualización de datos**: creación de gráficos y mapas detallados para explorar y presentar datos de manera efectiva.

- **Manipulación de datos**: transformación, limpieza y preparación de datos para análisis mediante paquetes como dplyr y tidyr.
- **Modelado predictivo**: desarrollo de modelos de machine learning, incluyendo regresión, clasificación y clustering.
- **Generación de informes**: automatización de informes y creación de documentos reproducibles con R Markdown.
- **Interfaz de programación**: desarrollo de aplicaciones interactivas y dashboards usando Shiny para presentaciones dinámicas de datos.

4 Sobre R (II)

Vale, pero, ¿qué hace realmente R? Ponme un ejemplo. De acuerdo, veamos una imagen como la siguiente:

```
library(tidyverse)
library(gridExtra)

frase <- unlist(strsplit("la noche en la que suplico que no
                         salga el sol", " "))

# Crear el data frame
datos <- data.frame(
  palabra = frase,
  tiempo = seq(0, length(frase) - 1),
  pitch = runif(length(frase), min=80, max=90),
  intensidad = runif(length(frase), min=70, max=85)
)%>%filter(palabra!="")%>%mutate(tiempo=row_number())

# Create the plot with labels
plot <- ggplot(datos) +
  geom_point(aes(x = tiempo, y = pitch, color = "Pitch")) +
  geom_smooth(aes(x = tiempo, y = pitch, color = "Pitch")) +
  geom_text(aes(x = tiempo, y = pitch, label = palabra), vjust = -1,
            hjust = 0.5, size = 3.5, check_overlap = TRUE) +
  geom_point(aes(x = tiempo, y = intensidad, color = "Intensidad")) +
  geom_smooth(aes(x = tiempo, y = intensidad, color = "Intensidad")) +
  labs(color = "Variable") + # Add text labels above points
```

```
  theme_minimal()
plot
```

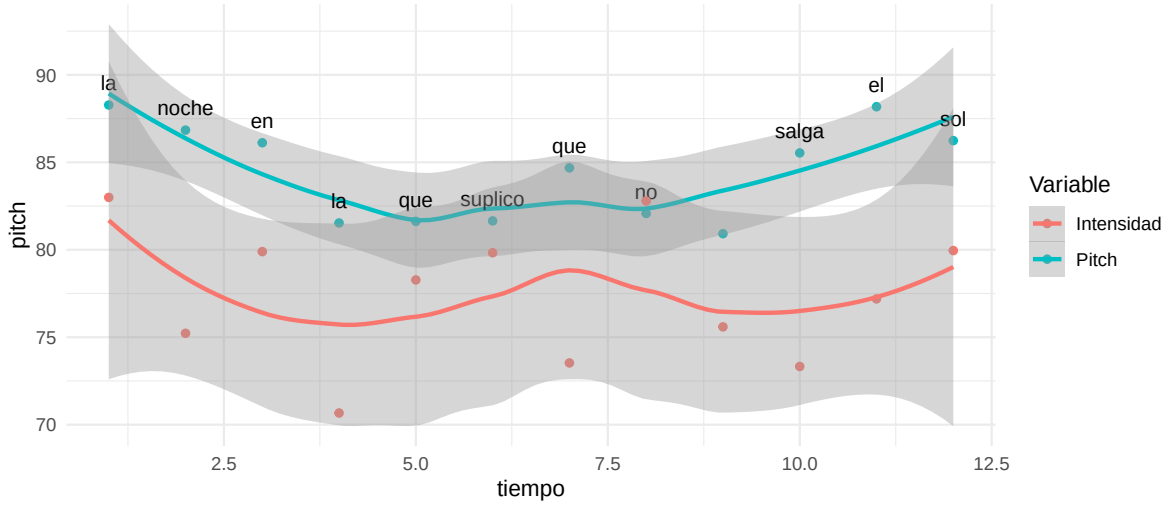


Figura extraída con GGplot2. Curva melódica y de intensidad del enunciado *la noche en la que no salga el sol*

En la Figura anterior, se visualiza la curva melódica y de intensidad del enunciado *la noche en la que no salga el sol*, un enunciado extraído de una canción de Enrique Iglesias. En el eje x (sí, el horizontal) se representa el tiempo, en el eje y (sí, el vertical) se representa la frecuencia fundamental (pitch) y la intensidad. Cada punto representa una palabra del enunciado y la curva suavizada muestra la tendencia general de la curva.

Esta es una de las muchas posibilidades que ofrece R para visualizar datos. En este caso, se ha utilizado el paquete ggplot2, uno de los más populares y versátiles para la visualización de datos en R. Como se verá más abajo, las posibilidades de visualización en R son casi infinitas y permiten crear gráficos detallados y personalizados para explorar y presentar datos de la manera más adecuada para nuestros intereses de investigación.

5 ¿Qué más puedo hacer con R?

R no solo sirve para crear gráficos o realizar pruebas estadísticas, sino que también puede ser utilizado para realizar análisis de texto, crear documentos científicos, programar scripts que realicen funciones de manera automática, crear aplicaciones web para consultar datos, entre otras muchas funcionalidades. Por ejemplo, hace casi 6 años que no utilizo Microsoft Word, ya que R me permite crear documentos científicos de manera más eficiente, personalizada y puedo

integrar directamente los gráficos y los análisis estadísticos sin tener que copiar y pegarlos cada vez. De hecho, este propio libro de materiales está escrito en R, concretamente con el paquete Quarto.

A continuación, se presentan algunas de las posibilidades que ofrece R:

Más funcionalidades

- Escribir documentos científicos mediante Rmarkdown o Quarto (este mismo documento ha sido escrito en R).
- Exportar tus documentos a varios formatos: PDF, Word o Powerpoint.
- Modificación de las plantillas. Ejemplo: los artículos de la revista *Normas* de la UV se generan a partir de una plantilla de Quarto.
- Programar scripts que realicen funciones de manera automática. Por ejemplo, abre todos los archivos de una carpeta e impórtalos.
- Crear aplicaciones web para consultar datos e incluso corpus lingüísticos. Ej.:

Oralstats Aroca.

6 ¿Por qué RStudio?

RStudio es un programa que sirve de "caparazón" para R. Permite que muchas tareas sean más sencillas y rápidas. También es multiplataforma.

Justificación de R

- RStudio es un entorno de desarrollo integrado (IDE) para R.
- RStudio simplifica la programación en R y mejora la productividad del usuario.
- RStudio es gratuito y de código abierto.
- RStudio es multiplataforma (Windows, Mac y Linux).

7 Bases de datos lingüísticas de ejemplo

En este libro se usarán dos bases de datos lingüísticas para ejemplificar los análisis estadísticos y la visualización de datos con R. Las bases de datos son las siguientes:

1. Base de datos Idiolectal. Se trata de la base de datos utilizada para realizar el siguiente artículo:

   Cabedo Nebot, A. Análisis melódico del habla como herramienta distintiva para el perfil idiolectal de hablantes. Revista da ABRALIN, [S. l.], v. 21, n. 2, p. 48–70, 2022. DOI: https://doi.org/10.25189/rabralin.v21i2.2103.

   Recoge datos de tonemas de los mismos cinco hablantes en dos géneros discursivos distintos (podcast y entrevista).

2. Base de datos Fonocortesía. Se trata de la base de datos recogida para el proyecto Fonocortesía, dirigida por el catedrático Antonio Hidalgo Navarro en el marco del proyecto de investigación Fonocortesía, subvencionado por Ministerio de Ciencia, Innovación y Universidades (FFI2009-07034). Con esta base de datos se han realizado diferentes investigaciones de las que destacamos la siguiente:

   > Cabedo Nebot A. y Hidalgo Navarro A. (2023). Caracterización fónica de la (des)cortesía en el español hablado de Valencia. Aproximación cualitativo-cuantitativa. Círculo de Lingüística Aplicada a la Comunicación, 93, 131-149. https://doi.org/10.5209/clac.82314

8 Uso de Excel o Google Sheets

Por experiencia, muchos investigadores en lingüística utilizan Excel o Google Sheets para almacenar y analizar datos. Sin embargo, estos programas tienen limitaciones en cuanto a la cantidad de datos que pueden manejar y las operaciones estadísticas que pueden realizar. Además, no son tan flexibles ni potentes como R para el análisis de datos.

De todos modos, es cierto que algunas investigaciones solo requieren un análisis básico de los datos, por lo que Excel o Google Sheets pueden ser suficientes. Si solo queremos extraer un gráficos sencillo de barras, por ejemplo, o calcular la media de una variable, Excel o Google Sheets pueden ser una buena opción. Sin embargo, si queremos realizar análisis estadísticos más avanzados, como regresiones, análisis de varianza o análisis de componentes principales, R es mejor opción.

Uso de Excel

- Formato tabular. Filas y columnas.
- Organización de datos.
- Pocos datos
- Tablas dinámicas para estadística básica.
- Limitaciones: no permite realizar análisis estadísticos avanzados.

8.1 Métodos de exploración avanzada en Excel o Google Sheets

A medio camino entre la hoja de cálculo clásica y R, estaría la opción de "tabla dinámica" que podemos encontrar en Excel o Google Sheets.

Sobre tablas dinámicas

Tablas dinámicas: "Una tabla dinámica es una herramienta avanzada para calcular, resumir y analizar datos que le permite ver comparaciones, patrones y tendencias en ellos. Las tablas dinámicas funcionan de forma un poco distinta dependiendo de la plataforma que use para ejecutar Excel." (extraído de: https://support.microsoft.com/es-es/office/crear-una-tabla-din%C3%A1mica-para-analizar-datos-de-una-hoja-de-c%C3%A1lculo-a9a84538-bfe9-40a9-a8e9-f99134456576)

8.2 Ejercicio

Explora la base de datos *Idiolectal* y la base de datos *Fonocortesía* en Excel o Google Sheets.

9 Definir la construcción de la base de datos (estructura)

La construcción de la base de datos es un paso fundamental en cualquier investigación. La base de datos debe estar bien estructurada y organizada para que los análisis posteriores sean precisos y fiables. En el caso de las bases de datos lingüísticas, es importante tener en cuenta las variables que se van a analizar y cómo se van a medir. Por ejemplo, si estamos analizando la cortesía en el habla, es importante definir qué variables vamos a utilizar para medir la cortesía (por ejemplo, la frecuencia fundamental, la intensidad, la duración, etc.) y cómo vamos a codificarlas (por ejemplo, si vamos a utilizar escalas numéricas, categóricas, etc.).

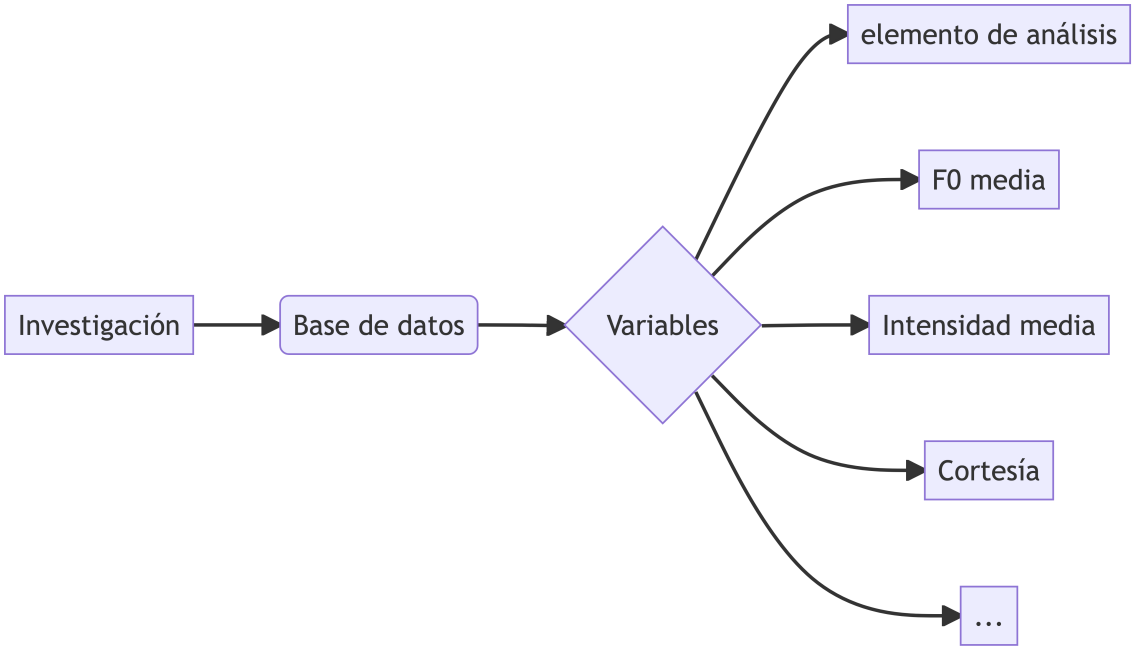


Figura. Proceso de construcción de la base de datos.

Cuando la base de datos tiene una estructura definida, es más fácil realizar análisis estadísticos y visualizaciones de datos. He sido testigo de muchas investigaciones en las que se ha tenido que volver a recoger los datos porque no estaban bien estructurados o porque no se habían definido las variables de manera adecuada. Por tanto, es importante dedicar tiempo y esfuerzo a la construcción de la base de datos para asegurarse de que los datos sean fiables y válidos.

10 Uso general de R

En esta sección se presentan algunas de las funcionalidades básicas de R que se utilizarán en los ejemplos prácticos de este libro. Estas funcionalidades incluyen la instalación de librerías, la carga de datos, la visualización de datos y la manipulación de datos. A continuación, se presentan los pasos básicos para comenzar a trabajar con R.

10.1 Instalar librerías

En R, una librería es un conjunto de funciones y datos que se utilizan para realizar tareas específicas. Existen muchas librerías en R que se pueden utilizar para realizar análisis estadísticos, visualizaciones de datos, manipulación de datos, etc. El proceso de instalar librerías en R es muy sencillo. Solo tienes que ejecutar el siguiente comando:

```r
install.packages("tidyverse")
install.packages("FactoMineR")
install.packages("factoextra")
install.packages("partykit")
install.packages("randomForest")
install.packages("DataExplorer")
install.packages("heatmap.2")
install.packages("corrplot")
install.packages("ggwordcloud")
```

10.2 Cargar librerías

Cada vez que reinicies R, tendrás que cargar las librerías que necesitas para trabajar. Para cargar una librería en R, simplemente tienes que ejecutar el siguiente comando:

```r
library(tidyverse)
library(corrplot)
library(FactoMineR)
library(factoextra)
library(partykit)
library(randomForest)
library(gplots)
library(ggwordcloud)
```

10.3 Importar datos

Para importar datos en R, puedes utilizar la función `read_csv()` del paquete `readr`. Esta función te permite importar datos en formato CSV, que es un formato de archivo de texto que se utiliza para almacenar datos tabulares. También puedes importar datos en otros formatos, como Excel, utilizando las funciones correspondientes del paquete `readxl`. Incluso en R Studio puedes importar datos directamente desde Excel o Google Sheets, con la opción File > Import Dataset > From Excel.

```r
library(readxl)
fonocortesia <- read_xlsx("databases/corpus.xlsx")
```

10.4 Conoce la estructura de tus datos: str o summary

Conocer la estructura de la base de datos es importante, sobre todo, para cerciorarnos de que los datos se han importado correctamente y para saber de qué tipo son. Algunas importaciones podrían no ser correctas y, por tanto, es importante conocer la estructura de los datos. Por ejemplo, aunque no es muy frecuente, podría darse el error de que una variable numérica se importara como variable de carácter o viceversa. Para ello, podemos utilizar las funciones `str()` y `summary()`.

Ejemplo de *str* para las primeras cinco columnas:

```
str(fonocortesia[,c(1:5)])
```

```
tibble [282 x 5] (S3: tbl_df/tbl/data.frame)
 $ Conversacion    : chr [1:282] "VALESCO 114A" "VALESCO 130A" "VALESCO 194A" "VALESCO 114A"
 $ Cortes_Descortes: chr [1:282] "descortés" "descortés" "descortés" "descortés" ...
 $ Llama_Atencion  : chr [1:282] "acento;entonación;velocidad de habla" "acento;duración;ento
 $ Mediodeexpresion: chr [1:282] "Fórmulas indirectas" "Atenuación" "Intensificación" "Intens
 $ F0_Inicial      : num [1:282] 227 213 242 148 249 ...
```

Ejemplo de *summary* para las primeras cinco columnas:

```
summary(fonocortesia[,c(1:5)])
```

```
 Conversacion       Cortes_Descortes   Llama_Atencion     Mediodeexpresion
 Length:282         Length:282         Length:282         Length:282
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character




   F0_Inicial
 Min.   :  0.0
 1st Qu.:161.8
 Median :219.7
 Mean   :212.7
 3rd Qu.:252.2
 Max.   :490.0
 NA's   :42
```

10.5 Citar en R

En R, puedes citar paquetes y funciones utilizando la función `citation()`. Por ejemplo, para citar el paquete `tidyverse`, puedes ejecutar el siguiente comando:

```
citation()
```

```
To cite R in publications use:

  R Core Team (2024). _R: A Language and Environment for Statistical
  Computing_. R Foundation for Statistical Computing, Vienna, Austria.
  <https://www.R-project.org/>.

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {R: A Language and Environment for Statistical Computing},
    author = {{R Core Team}},
    organization = {R Foundation for Statistical Computing},
    address = {Vienna, Austria},
    year = {2024},
    url = {https://www.R-project.org/},
  }

We have invested a lot of time and effort in creating R, please cite it
when using it for data analysis. See also 'citation("pkgname")' for
citing R packages.
```

```
citation("tidyverse")
```

```
To cite package 'tidyverse' in publications use:

  Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R,
  Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller
  E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V,
  Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to
  the tidyverse." _Journal of Open Source Software_, *4*(43), 1686.
  doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.

A BibTeX entry for LaTeX users is
```

```
@Article{,
  title = {Welcome to the {tidyverse}},
  author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy
  year = {2019},
  journal = {Journal of Open Source Software},
  volume = {4},
  number = {43},
  pages = {1686},
  doi = {10.21105/joss.01686},
}
```

10.6 Data frames

Un data frame es una estructura de datos en R que se utiliza para almacenar datos en forma tabular. Es similar a una matriz, pero cada columna puede contener un tipo de datos diferente.

```r
data.frame(
  cortesia = c("cortés", "descortés", "cortés"),
  f0_media = c(145, 187, 135),
  sexo = c("Hombre", "Mujer", "Hombre")
)
```

```
   cortesia f0_media   sexo
1    cortés      145 Hombre
2 descortés      187  Mujer
3    cortés      135 Hombre
```

> **Visualizar data frames**
>
> Los data frames se pueden visualizar en RStudio en la pestaña “Environment” o escribiendo el nombre del data frame en la consola. Para mejorar la visualización, puedes usar la función `View()`.

11 Tareas de limpieza y manipulación de datos

> **Advertencia**
>
> La limpieza y manipulación de datos es una parte esencial del análisis de datos. Los datos pueden contener errores, valores ausenetes o vacíos, valores atípicos y otros problemas que pueden afectar la validez y fiabilidad de los análisis. Por tanto, es importante realizar tareas de limpieza y manipulación de datos para asegurarse de que los datos sean precisos y fiables. Por ejemplo, un error consciente en la base de datos Fonocortesía es que hay un valor "desconocido" en la variable *Cortes_Descortes*.

Entre las tareas de limpieza y revisión de datos, se pueden incluir las siguientes:

1. **Revisión y corrección de valores ausentes**:
    - Identificar y manejar los valores no disponibles (`NA` en R).
    - Decidir si imputar los valores ausentes con la media, mediana, moda, o algún otro método, o eliminar las filas/columnas con esos valores vacíos.
2. **Detección y manejo de valores atípicos**:
    - Identificar valores atípicos o outliers que pueden distorsionar el análisis. Por ejemplo, un valor de de F0 de un hablante de 600 Hz puede proceder de una distorsión acústica, de influencia ambiental, de una canción, de un golpe que coincide con la persona hablando…
    - Decidir si eliminar, transformar o tratar de otra manera estos valores.
3. **Estandarización y normalización de datos**:
    - Estandarizar unidades de medida para asegurarse de que sean consistentes. Por cierto, en corpus lingüísticos es habitual trabajar con frecuencias normalizadas por millón de palabras; en fonética, por ejemplo, la melodía se normaliza mediante el uso de semitonos, ya que neutralizan la diferencia tonal biológica entre hombres y mujeres.
    - Normalizar o estandarizar variables si es necesario para ciertos tipos de análisis.
4. **Conversión de tipos de datos**:
    - Asegurarse de que los datos estén en los tipos adecuados (por ejemplo, convertir variables categóricas a factores en R).
5. **Revisión de la coherencia de los datos**:

- Verificar que no haya inconsistencias en los datos (por ejemplo, un valor de edad negativo).
- Asegurar que los valores categóricos estén correctamente codificados y no haya variaciones como “Hombre” y “hombre”.

6. **Eliminación de duplicados**:
   - Identificar y eliminar registros duplicados que puedan afectar el análisis.
7. **Corrección de errores tipográficos y de entrada de datos**:
   - Revisar y corregir errores tipográficos o de entrada manual en los datos.
8. **Creación de variables derivadas**:
   - Crear nuevas variables que puedan ser útiles para el análisis, como agregar una variable que represente la diferencia entre dos fechas (edad, duración, etc.).
9. **Filtrado de datos irrelevantes**:
   - Eliminar columnas o filas que no sean relevantes para el análisis específico.
   - Una conducta habitual en bases de datos lingüísticas es repetir la información en varias columnas. Por ejemplo, en la base de datos Fonocortesía, las variables *curva melódica* y *tonema* refieren prácticamente el mismo fenómeno.

11.1 ¿Qué es Tidyverse?

Para las tareas de limpieza o de transformación en R, se puede utilizar el paquete `tidyverse`. Este paquete es una colección de paquetes de R diseñados para la ciencia de datos:

https://www.tidyverse.org/

> **Note**
>
> Colección de paquetes de R diseñados para la ciencia de datos.
>
> - **dplyr**: manipulación de datos.

- **ggplot2**: visualización de datos.
- **tidyr**: limpieza de datos.
- **readr**: importación de datos.
- ...

11.2 Ver y filtrar datos

En ocasiones, puede ser útil ver los datos para identificar posibles problemas o errores. Para filtrar los datos, podemos utilizar la función `filter()` para seleccionar las filas que cumplan ciertas condiciones.

1. Ver los datos (table)
2. Filtrar datos (filter)

! Notas importantes sobre comandos

- El operador `%>%` se utiliza para encadenar funciones en R. Se lee de izquierda a derecha, lo que facilita la lectura del código.
- El operador `==` se utiliza para comparar si dos valores son iguales.
- EL operador `!=` se utiliza para comparar si dos valores son diferentes.
- El operador `<-` se utiliza para crear un nuevo objeto (variable, dataframe...) en R.

En el siguiente bloque de código se utilizan los comandos mencionados anteriormente:

```
table(fonocortesia$Cortes_Descortes)
```

```
     cortés desconocido   descortés
        135           1         146
```

```
fonocortesia%>%filter(Cortes_Descortes=="desconocido")
```

```
# A tibble: 1 x 36
  Conversacion Cortes_Descortes Llama_Atencion Mediodeexpresion F0_Inicial
  <chr>        <chr>            <chr>          <chr>                 <dbl>
1 VALESCO 194A desconocido      entonación     desconocido             296
# i 31 more variables: F0_Final <dbl>, F0_Media <dbl>, F0_Maxima <dbl>,
#   F0_Minima <dbl>, Intensidad_Maxima <dbl>, Intensidad_Minima <dbl>,
```

```
#   Intensidad_Primera <dbl>, Intensidad_Ultima <dbl>, Intensidad_Media <dbl>,
#   Silabas <dbl>, Duracion <dbl>, Duracion_Pausa_Anterior <dbl>,
#   Duracion_Pausa_Posterior <dbl>, Continuacion_Pausa <chr>,
#   Curva_Melodica <chr>, Otro_Curva_Melodica <chr>,
#   Inflexion_Local_Interna <chr>, Tonema <chr>, Unidad_Del_Discurso <chr>, ...
```

```
table(fonocortesia$Cortes_Descortes)
```

```
     cortés desconocido   descortés
        135           1         146
```

```
fonocortesia_filt <- fonocortesia%>%filter(Cortes_Descortes!="desconocido")
```

11.3 Seleccionar columnas: select

Para seleccionar columnas específicas de un data frame en R, se puede utilizar la función `select()` del paquete `dplyr`. Esta función permite seleccionar las columnas que se desean mantener en el data frame y descartar las que no son necesarias.

> **Sobre select**
>
> Es recomendable almacenar el resultado de la función `select()` en un nuevo objeto para no perder los datos originales. Por ejemplo, si hacemos esto:
> fonocortesia <- fonocortesia %>%select(F0_Media, Duracion)
> Estaremos sobreescribiendo la base de datos original. Si queremos mantener la base de datos original, podemos hacer esto:
> fonocortesia_sel <- fonocortesia %>%select(F0_Media, Duracion)

```
fonocortesia_sel <- fonocortesia%>%select(Cortes_Descortes, F0_Media,
                                          Intensidad_Media)
```

11.4 Ordenar datos: arrange

Para ordenar los datos en R, se puede utilizar la función `arrange()` del paquete `dplyr`. Esta función permite ordenar los datos en función de una o varias columnas. Por ejemplo, si queremos ordenar los datos por la columna `F0_Media`, podemos hacer lo siguiente:

```
fonocortesia_ord <- fonocortesia%>%arrange(F0_Media)
```

11.5 Reordernar columnas:: relocate

Para reordenar las columnas de un data frame en R, se puede utilizar la función `relocate()` del paquete `dplyr`. Esta función permite mover una columna a una posición específica en el data frame. Por ejemplo, si queremos mover la columna `Cortes_Descortes` después de la columna `Intensidad_Media`, podemos hacer lo siguiente:

```
fonocortesia_reord2 <- fonocortesia%>%relocate(Cortes_Descortes,
                                               .after = Intensidad_Media)
```

11.6 Crear nuevas columnas: mutate

Para crear nuevas columnas en un data frame en R, se puede utilizar la función `mutate()` del paquete `dplyr`. Esta función permite crear nuevas columnas a partir de las columnas existentes. Por ejemplo, si queremos crear una nueva columna que contenga el valor en semitonos de la columna `F0_Media`, podemos ejecutar el siguiente comando:

```
fonocortesia_nueva <- fonocortesia%>%mutate(F0_media_norm =
                                            12*log2(F0_Media/1))
```

> ⚠ Sobre mutate
>
> Igual que al seleccionar o filtrar hay que almacenar el resultado en un nuevo objeto para no perder los datos originales. Si en el código anterior hubiéramos hecho `fonocortesia_nueva <- fonocortesia %>%mutate(F0_media = 12*log2(F0_Media/1))`, estaríamos sobreescribiendo la variable F0_Media.

11.6.1 Crear columnas usando varias columnas mediante suma de variables

Podemos crear nuevas variables realizando cualquier operación matemática: suma, resta, multiplicación, división, etc. Por ejemplo, si queremos crear una nueva columna que contenga la suma de las columnas `F0_Media` e `Intensidad_Media`, podemos ejecutar el siguiente código:

```
fonocortesia_nueva2 <- fonocortesia%>%mutate(Nueva_columna =
(F0_Media + Intensidad_Media) )
```

11.6.2 Crear columnas usando varias columnas mediante media de variables

La media puede computarse de varias maneras. Por ejemplo, si queremos crear una nueva columna que contenga la media de las columnas `F0_Media` e `Intensidad_Media`, podemos ejecutar el siguiente código, aunque también existe la función `mean()`:

```
fonocortesia_nueva3 <- fonocortesia%>%mutate(Nueva_columna =
(F0_Media + Intensidad_Media)/2)
```

11.6.3 Crear columnas usando varias columnas usando rowwise

Normalmente, las funciones de `dplyr` operan por columnas. Sin embargo, a veces es necesario operar por filas. Para ello, se puede utilizar la función `rowwise()`. Por ejemplo, si queremos crear una nueva columna que contenga la media de dos celdas contiguas, puede ejecutarse este código:

```
fonocortesia_nueva4 <- fonocortesia%>%rowwise()%>%
  mutate(Nueva_columna = mean(c(F0_Media, Intensidad_Media),na.rm=T))
```

11.6.4 Crear una columna de índice usando row_number

Para crear una columna de índice en un data frame en R, se puede utilizar la función `row_number()`. Esta función asigna un número único a cada fila del data frame. Por ejemplo, si queremos crear una columna de índice en el data frame `fonocortesia`, se realizaría de la siguiente manera:

```
fonocortesia_nueva5 <- fonocortesia%>%mutate(id=row_number())
```

11.6.5 Crear columnas usando condiciones

R, como lenguaje de programación, permite aplicar secuencias de computación condicional. Por ejemplo, si queremos crear una nueva columna que contenga el valor "Intensificación" si la columna `Mediodeexpresion` es igual a "Intensificación" y "Atenuación" en caso contrario, podemos ejecutar siguiente código que aparece más abajo. Esto serviría para normalizar la variable Medio de expresión que disponía inicialmente de 16 categorías. Puedes comprobarlo ejecutando `table(fonocortesia$Mediodeexpresion)`:

```
fonocortesia_nueva3 <- fonocortesia%>%mutate(Medio_nuevo =
ifelse(Mediodeexpresion=="Intensificación",
"Intensificación",
"Atenuación"))
```

11.6.6 Crear columnas usando paste

Otra opción interesante es la de combinar columnas en una nueva columna. Por ejemplo, si queremos crear una nueva columna que contenga la concatenación de las columnas `Cortes_Descortes` y `Mediodeexpresion`, podemos ejecutar el siguiente código:

```
fonocortesia_nueva4 <- fonocortesia%>%mutate(Nueva_columna =
                                               paste(Cortes_Descortes,
                                                     Mediodeexpresion,
                                                     sep = "_"))

head(fonocortesia_nueva4$Nueva_columna)
```

```
[1] "descortés_Fórmulas indirectas" "descortés_Atenuación"
[3] "descortés_Intensificación"     "descortés_Intensificación"
[5] "descortés_Intensificación"     "cortés_Intensificación;Humor"
```

11.7 Agrupar datos: group_by

Las agrupaciones son realmente importantes en R. Permiten calcular medias, sumas, desviaciones típicas, etc. por grupos. Por ejemplo, si queremos calcular la media de la columna `F0_Media` y de la columna `Intensidad_Media` por grupo de la columna `Cortes_Descortes`, podemos ejecutar el siguiente código:

```
fonocortesia_agrup <- fonocortesia%>%group_by(Cortes_Descortes)%>%
  mutate(F0_media_mean = mean(F0_Media,na.rm = T),
         Intensidad_media_mean = mean(Intensidad_Media,na.rm = T))

head(fonocortesia_agrup)%>%select(Cortes_Descortes, F0_Media,
                                  Intensidad_Media, F0_media_mean,
                                  Intensidad_media_mean)
```

```
# A tibble: 6 x 5
# Groups:   Cortes_Descortes [2]
```

```
  Cortes_Descortes F0_Media Intensidad_Media F0_media_mean Intensidad_media_mean
  <chr>               <dbl>            <dbl>         <dbl>                 <dbl>
1 descortés             298               85          230.                  75.9
2 descortés             174.              78          230.                  75.9
3 descortés             315               81          230.                  75.9
4 descortés             224               85          230.                  75.9
5 descortés             226.              83          230.                  75.9
6 cortés                295               80          199.                  77.6
```

11.8 Resumir datos: summarise

A diferencia de la creación de nuevas columnas, la función `summarise()` se utiliza para resumir los datos. Por ejemplo, si queremos calcular la media de la columna `F0_Media` y de la columna `Intensidad_Media` por grupo de la columna `Cortes_Descortes`, podemos ejecutar el siguiente código:

```r
fonocortesia_resumen <- fonocortesia%>%group_by(Cortes_Descortes)%>%
  summarise(F0_media_mean = mean(F0_Media,na.rm = T),
            Intensidad_media_mean = mean(Intensidad_Media,na.rm = T))

head(fonocortesia_resumen)
```

```
# A tibble: 3 x 3
  Cortes_Descortes F0_media_mean Intensidad_media_mean
  <chr>                    <dbl>                 <dbl>
1 cortés                    199.                  77.6
2 desconocido               299                   82
3 descortés                 230.                  75.9
```

12 Visualización de datos

En esta sección, aprenderemos a visualizar datos utilizando el paquete *ggplot2* en R. GGplot2 es una librería de visualización de datos en R que permite crear gráficos de alta calidad de manera sencilla y flexible.

12.1 ¿Por qué visualizar datos?

Visualizar datos es una parte fundamental del análisis de datos. Las visualizaciones permiten explorar los datos, identificar patrones y tendencias, comunicar resultados y conclusiones, y

tomar decisiones adecuadas y coherentes. Las visualizaciones efectivas pueden ayudar a resumir y presentar datos de manera clara y concisa; esto facilita la interpretación y comprensión de los datos.

Images created by an AI from OpenAI

12.2 ¿Qué es ggplot2?

GGplot2 es una librería de visualización de datos en R que permite crear gráficos de alta calidad de manera sencilla y flexible. GGplot2 se basa en la gramática de gráficos, un enfoque que descompone los gráficos en componentes básicos (datos, estética, geometría, estadísticas y facetas) y permite construir gráficos complejos combinando estos componentes de manera intuitiva.

> ⚠ Sobre ggplot2
>
> La curva de aprendizaje para el uso de Ggplot2 puede ser un poco inclinada.

12.3 Tipos de gráficos

Los gráficos más comunes que se pueden crear con ggplot2 incluyen: gráfico de barras, gráfico de líneas, gráfico de dispersión, gráfico de violín, gráfico de áreas, gráfico de burbujas, gráfico de donut, gráfico de lolipop, gráfico de mapa de calor, gráfico de densidad, gráfico de correlaciones, gráfico de árbol…

12.3.1 Gráficos de barras

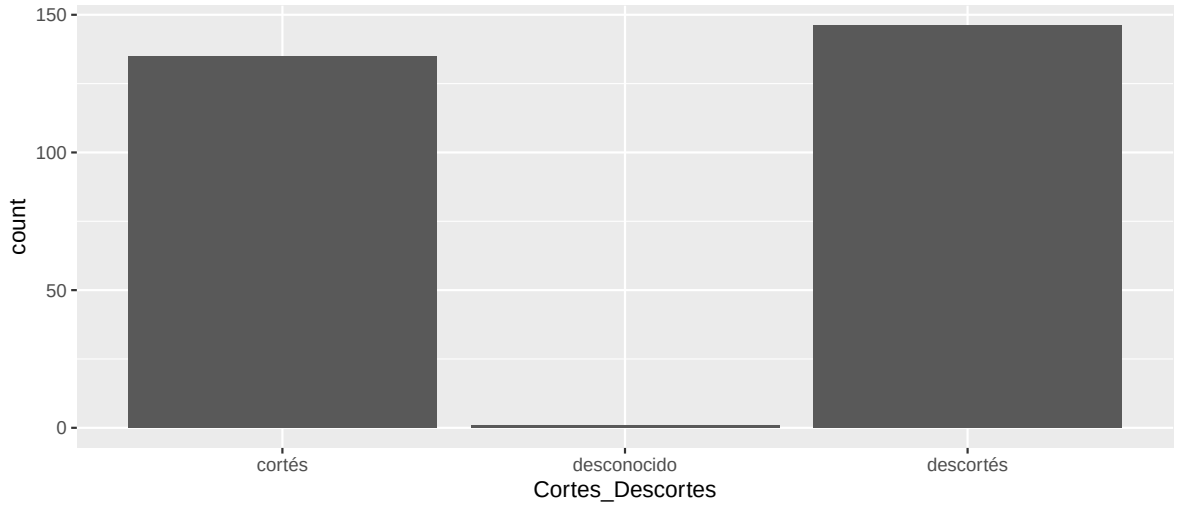
El gráfico de barras es uno de lo más utilizados en investigaciones primerizas, dado que permite visualizar la frecuencia de una variable categórica.

En las siguientes subsecciones se presentan varios ejemplos de gráficos de barras utilizando la base de datos Fonocortesía.

12.3.1.1 Barras 1

Este es el modo más simple de crear un gráfico de barras en ggplot2. En este caso, se muestra la frecuencia de la variable `Cortes_Descortes`.

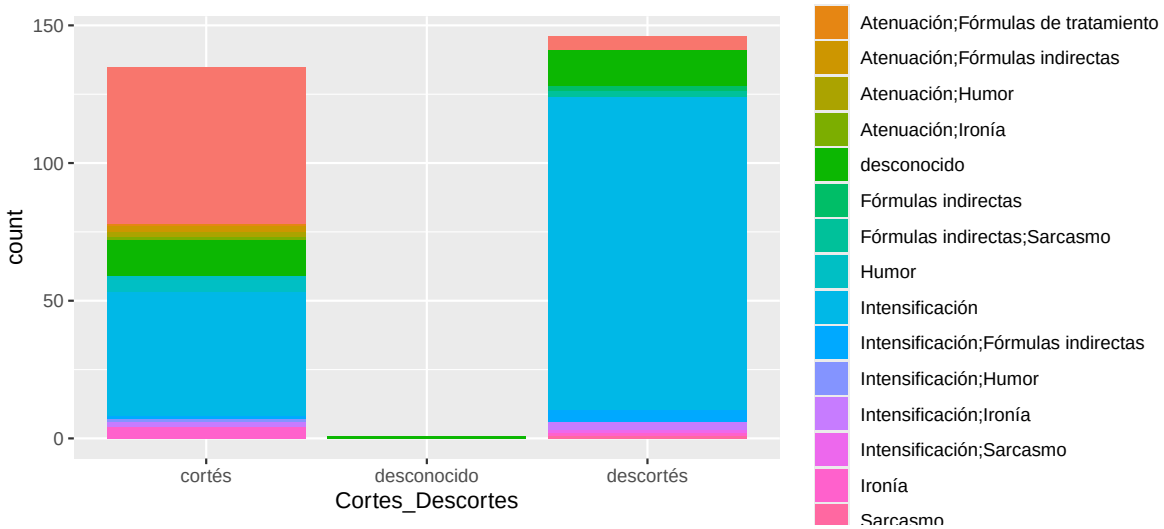
```
ggplot(fonocortesia, aes(x=Cortes_Descortes)) +
  geom_bar(stat="count")
```



12.3.1.2 Barras 2

En este caso, se muestra la frecuencia de la variable `Cortes_Descortes` agrupada por la variable `Mediodeexpresion`. Como se comentaba en secciones anteriores, esta visualización es de difícil acceso, ya que la variable `Mediodeexpresion` tiene 16 categorías. En este caso, la visualización nos indica que esta variable debe ser procesada de otra manera o que se debe simplificar.

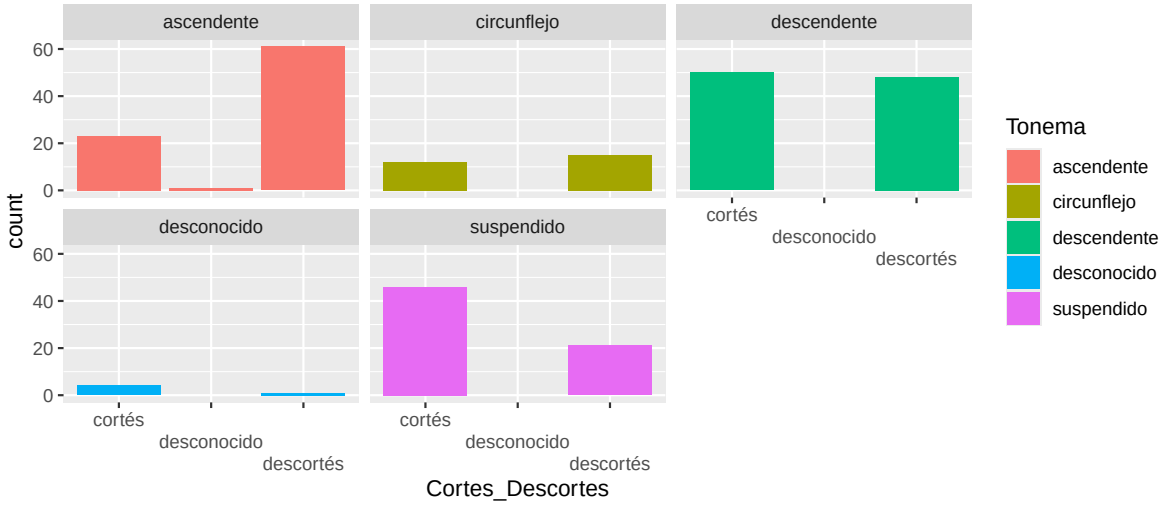
```
fonocortesia%>% ggplot(aes(x=Cortes_Descortes, fill=Mediodeexpresion)) +
  geom_bar(stat="count")
```



12.3.1.3 Barras 3

En el siguiente gráfico de barras, se muestra la frecuencia de la variable `Cortes_Descortes` agrupada por la variable `Tonema`. En este caso, la variable `Tonema` tiene 5 categorías, lo que facilita la visualización. El proceso facet_wrap permite dividir la visualización en cinco gráficos, uno por cada categoría de la variable `Tonema`.

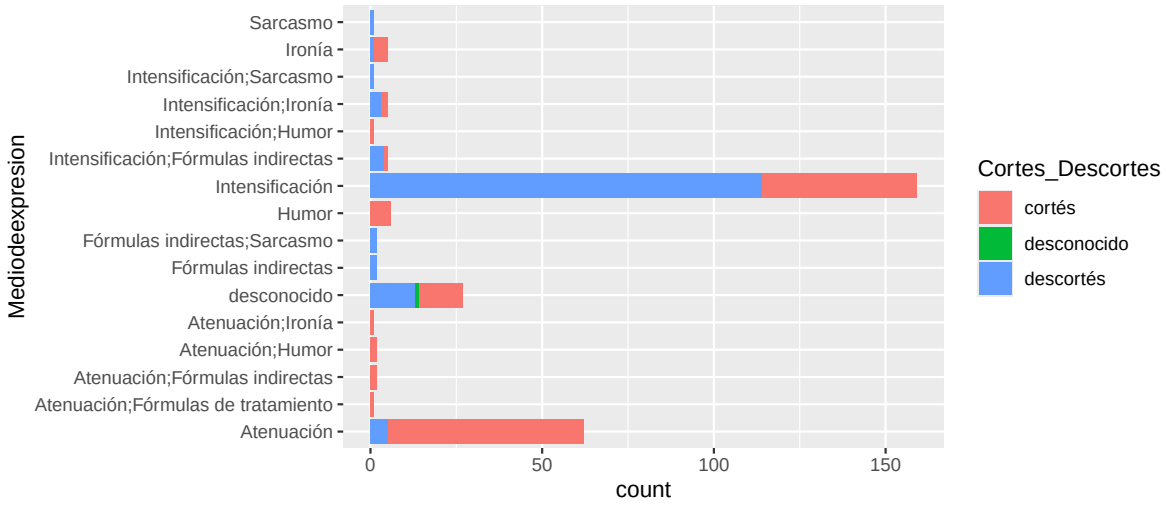
```
fonocortesia%>% ggplot(aes(x=Cortes_Descortes, fill=Tonema)) +
  scale_x_discrete(guide = guide_axis(n.dodge=3))+
  geom_bar(stat="count") + facet_wrap(~Tonema)
```



12.3.1.4 Barras 4

El siguiente gráfico de barras muestra la frecuencia de la variable `Cortes_Descortes` agrupada por la variable `Mediodeexpresion`. En este caso, se ha utilizado la función `coord_flip()` para girar el gráfico y facilitar la lectura de las etiquetas.

```
fonocortesia%>% ggplot(aes(x=Mediodeexpresion, fill=Cortes_Descortes)) +
  geom_bar(stat="count") + coord_flip()
```

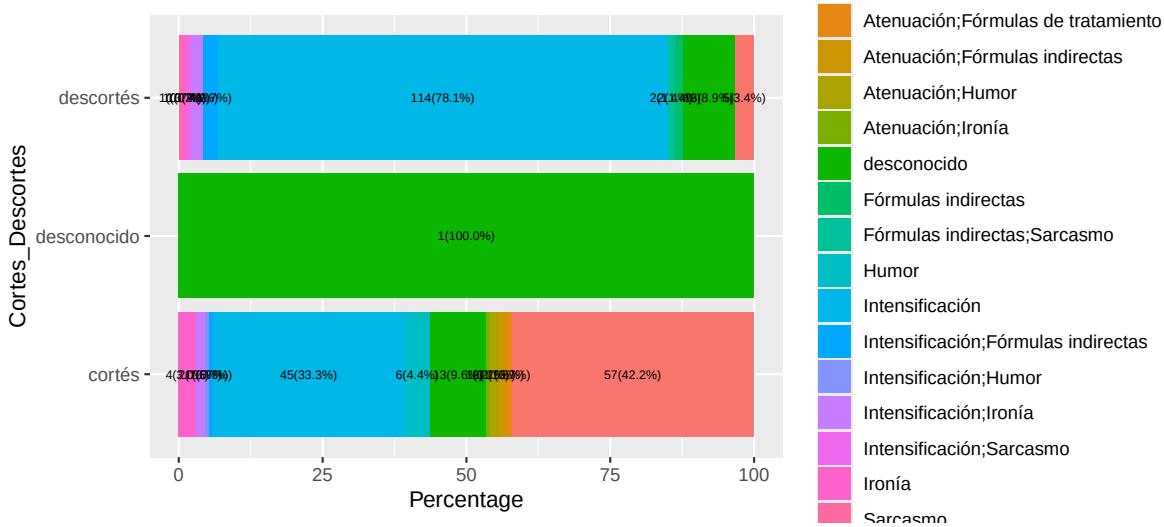


12.3.1.5 Barras 5

En el siguiente gráfico de barras, se muestra la frecuencia de la variable `Cortes_Descortes` agrupada por la variable `Mediodeexpresion`. En este caso, se ha utilizado la función `geom_text()` para añadir etiquetas con el número de observaciones y el porcentaje de cada categoría.

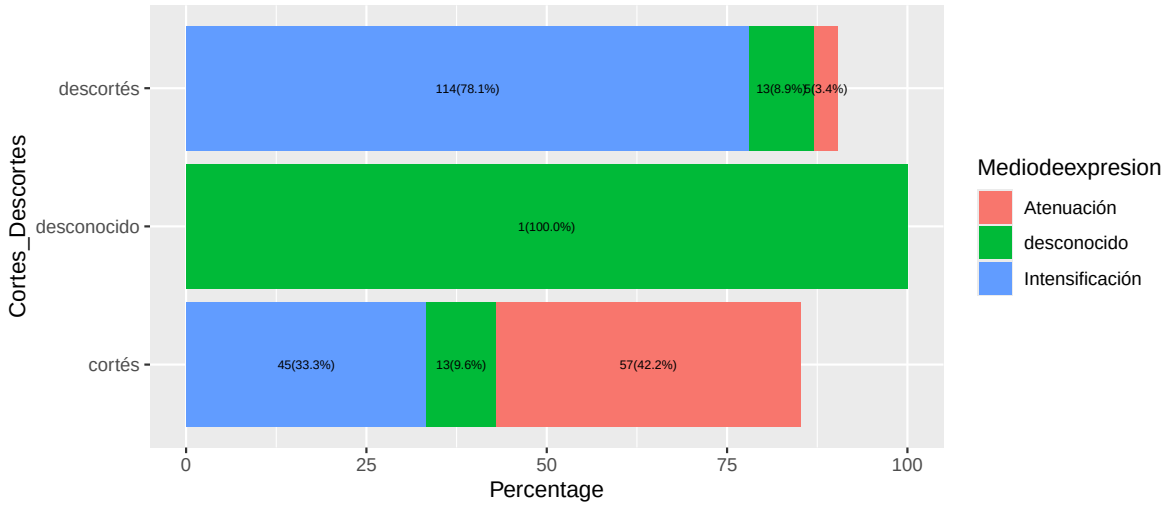
```
data <- fonocortesia%>%group_by(Cortes_Descortes,Mediodeexpresion)%>%
  summarise(Total = sum(n())) %>%
  mutate(Percentage = Total / sum(Total) * 100)

ggplot(data, aes(x=Cortes_Descortes,y=Percentage,
                 fill=Mediodeexpresion))+
  geom_bar(stat="identity") + geom_text(
    aes(label = paste(Total, "(", sprintf("%.1f%%", Percentage), ")",
                      sep = "")),
    position = position_stack(vjust = 0.5),
```

```
    size = 2  # Adjust text size
  ) + coord_flip()
```



Si reducimos el número de variantes para la variable `Mediodeexpresion`, la visualización es más clara:

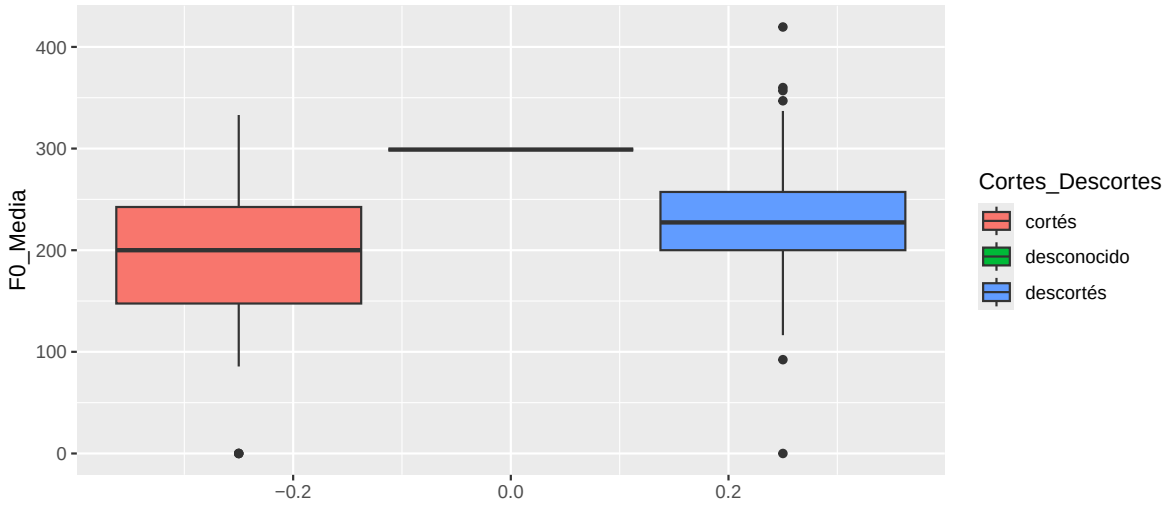
```
ggplot(data%>%filter(Mediodeexpresion%in%
                       c("Atenuación","Intensificación","desconocido")), aes(x=Cortes_Desc
  geom_bar(stat="identity") + geom_text(
    aes(label = paste(Total, "(", sprintf("%.1f%%",
                                          Percentage), ")", sep = "")),
    position = position_stack(vjust = 0.5),
    size = 2  # Adjust text size
  ) + coord_flip()
```



12.3.2 Diagramas de caja

El diagrama de caja es una forma de visualizar la distribución de un conjunto de datos. Muestra la mediana, los cuartiles, los valores atípicos y la dispersión de los datos. En el siguiente gráfico de caja, se muestra la distribución de la variable `F0_Media` agrupada por la variable `Cortes_Descortes`.

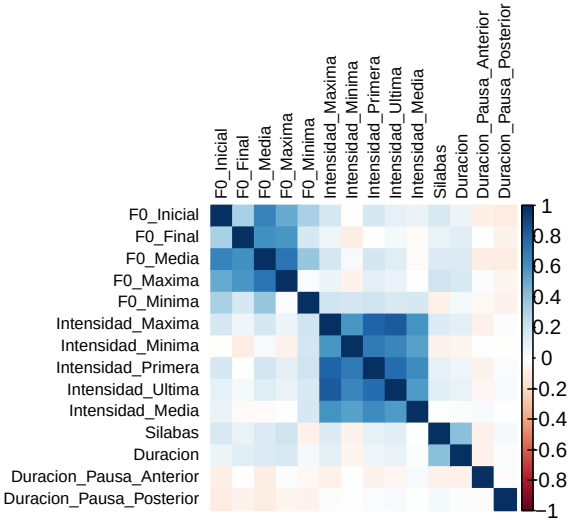
```
ggplot(fonocortesia, aes(y=F0_Media,fill=Cortes_Descortes)) +
  geom_boxplot()
```



12.3.3 Gráfico de correlaciones

El gráfico de correlaciones es una forma de visualizar la relación entre dos o más variables. Muestra la fuerza y la dirección de la relación entre las variables. En el siguiente gráfico de correlaciones, se muestra la matriz de correlaciones de las variables numéricas de la base de datos Fonocortesía.

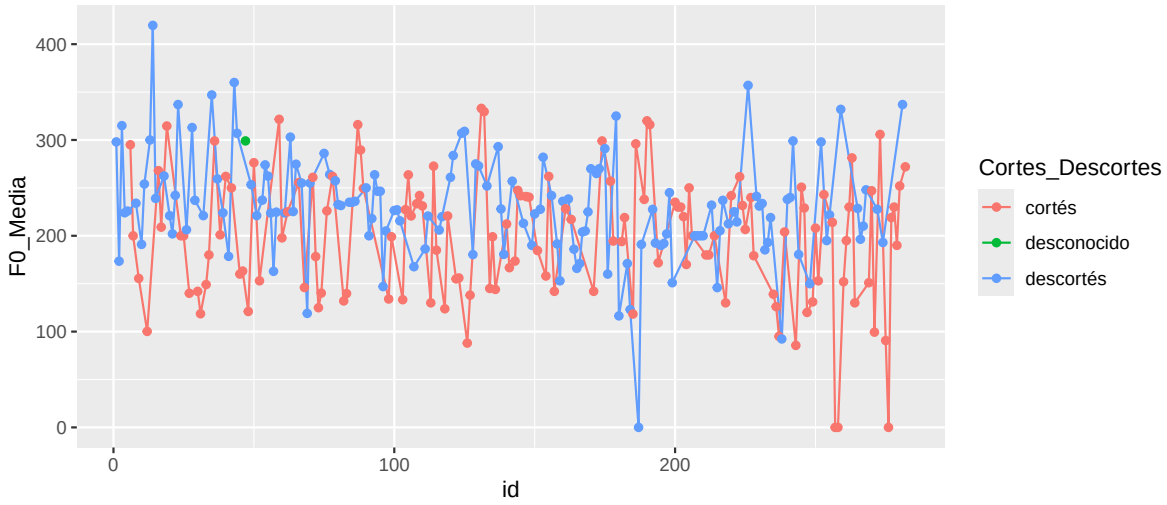
```
library(corrplot)
datosnum <- fonocortesia%>%select_if(is.numeric)
correlaciones <- cor(datosnum,use = "complete.obs")
corrplot(correlaciones, method = "color",tl.col = "black",tl.cex = 0.7)
```



12.3.4 Gráficos de líneas

Los gráficos de líneas son una forma de visualizar la evolución de una variable a lo largo del tiempo o de otra variable continua. En el siguiente gráfico de líneas, se muestra la evolución de la variable `F0_Media` a lo largo de las observaciones de la base de datos Fonocortesía.

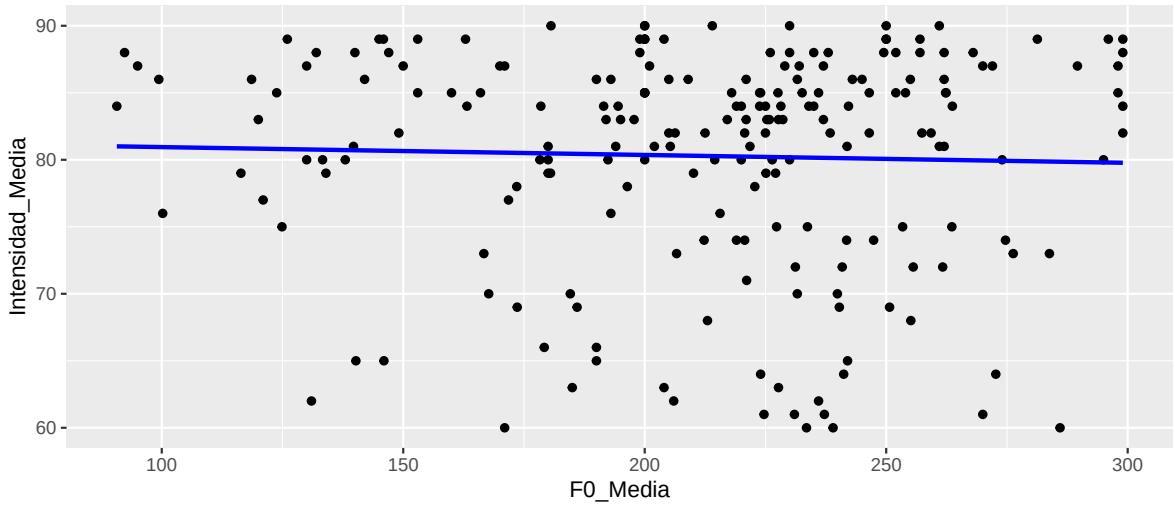
```
ggplot(fonocortesia%>%mutate(id=row_number()),
       aes(x=id,y=F0_Media,
      fill=Cortes_Descortes,
color = Cortes_Descortes)) + geom_line() + geom_point()
```



12.3.5 Gráficos de dispersión

Un gráfico de dispersión es una forma de visualizar la relación entre dos variables continuas. Muestra cómo una variable depende de otra. En el siguiente gráfico de dispersión, se muestra la relación entre las variables `F0_Media` e `Intensidad_Media`.

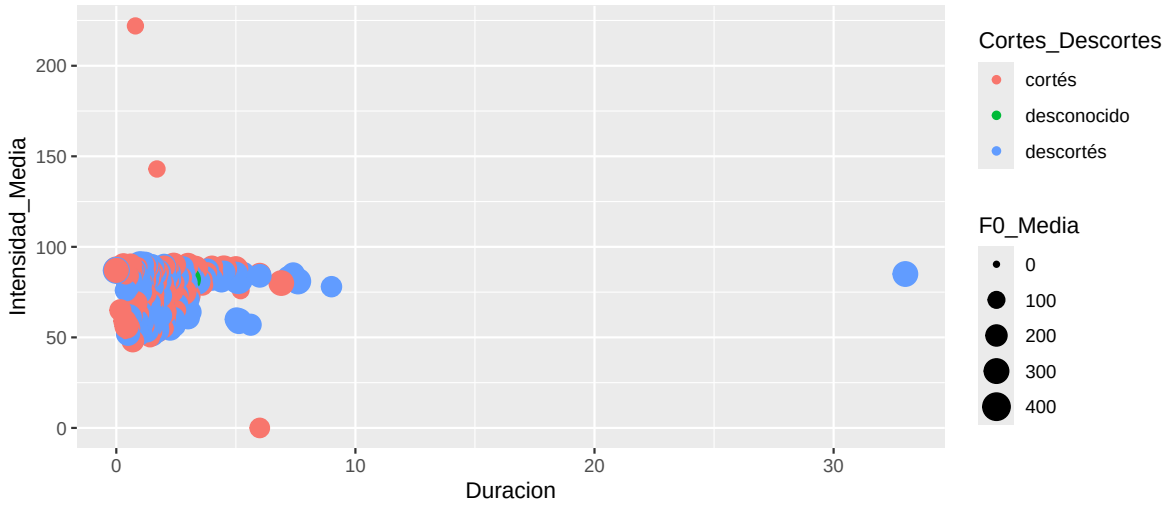
```
ggplot(fonocortesia%>%filter(between(Intensidad_Media,60,90),
F0_Media<300), aes(x=F0_Media, y=Intensidad_Media))  +
  geom_point() + geom_smooth(method = "lm", se = FALSE, color = "blue")
```



12.3.6 Gráficos de burbujas

Un gráfico de burbujas es una forma de visualizar la relación entre tres variables: dos variables continuas y una variable categórica. Muestra cómo una variable depende de dos variables continuas y de una variable categórica. En el siguiente gráfico de burbujas, se muestra la relación entre las variables `F0_Media`, `Intensidad_Media` y `Cortes_Descortes`.

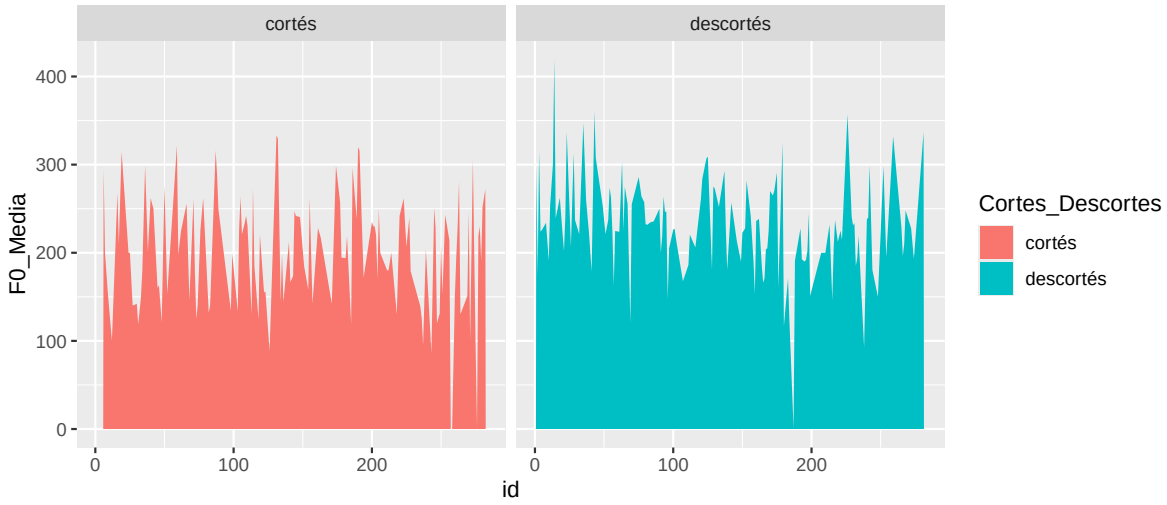
```
ggplot(fonocortesia, aes(x=Duracion, y=Intensidad_Media,
size=F0_Media, fill = Cortes_Descortes, color=Cortes_Descortes)) +
  geom_point()
```



12.3.7 Gráficos de áreas

Las áreas sirven para visualizar la distribución de una variable a lo largo de un eje. En el siguiente gráfico de áreas, se muestra la distribución de la variable `F0_Media` a lo largo de las observaciones de la base de datos Fonocortesía, agrupada por la variable `Cortes_Descortes`.

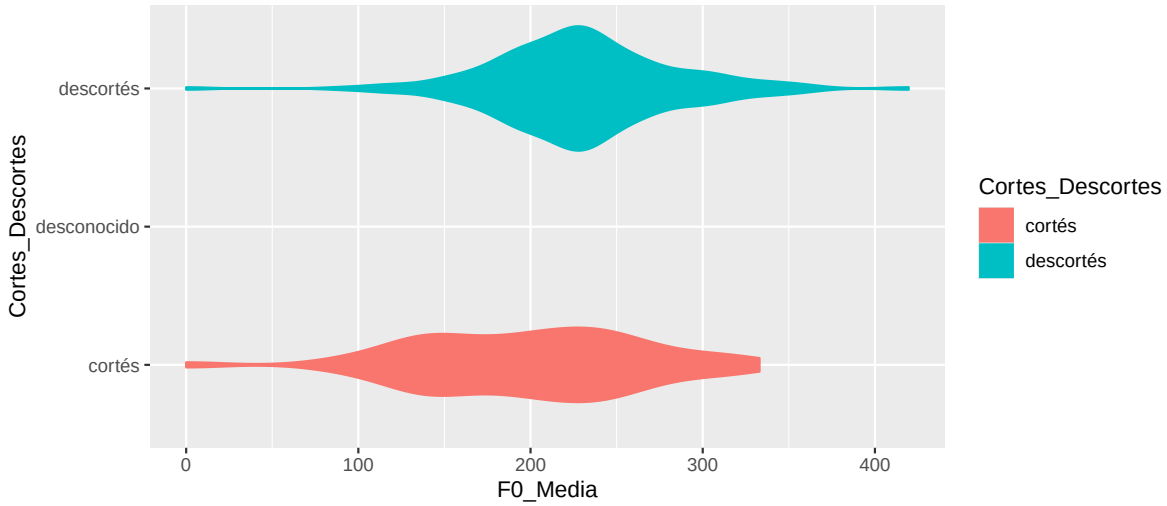
```
ggplot(fonocortesia%>%mutate(id=row_number())%>%
filter(Cortes_Descortes!="desconocido"),
aes(x=id, y=F0_Media, fill=Cortes_Descortes)) +
  geom_area() + facet_wrap(~Cortes_Descortes)
```



12.3.8 Gráficos de violín

El gráfico de violín es una combinación de un diagrama de caja y un gráfico de densidad. Muestra la distribución de los datos en función de una variable categórica.

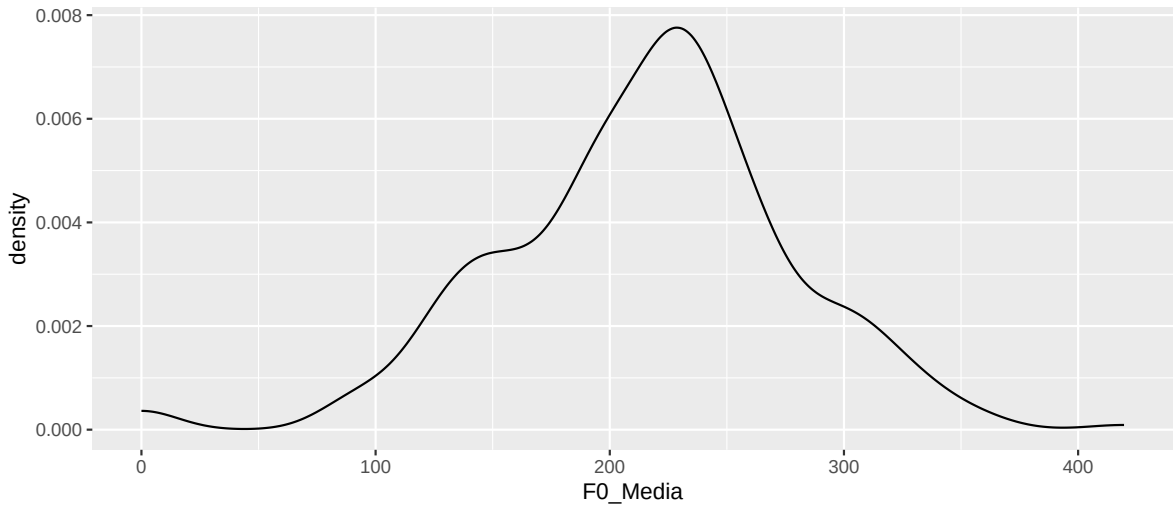
```
ggplot(fonocortesia, aes(x=Cortes_Descortes, y=F0_Media,
color=Cortes_Descortes, fill=Cortes_Descortes)) +
  geom_violin() + coord_flip()
```



12.3.9 Gráficos de densidad

La densidad de un conjunto de datos es una estimación de la distribución de probabilidad subyacente de los datos. Los gráficos de densidad muestran la distribución de los datos en forma de una curva suave.

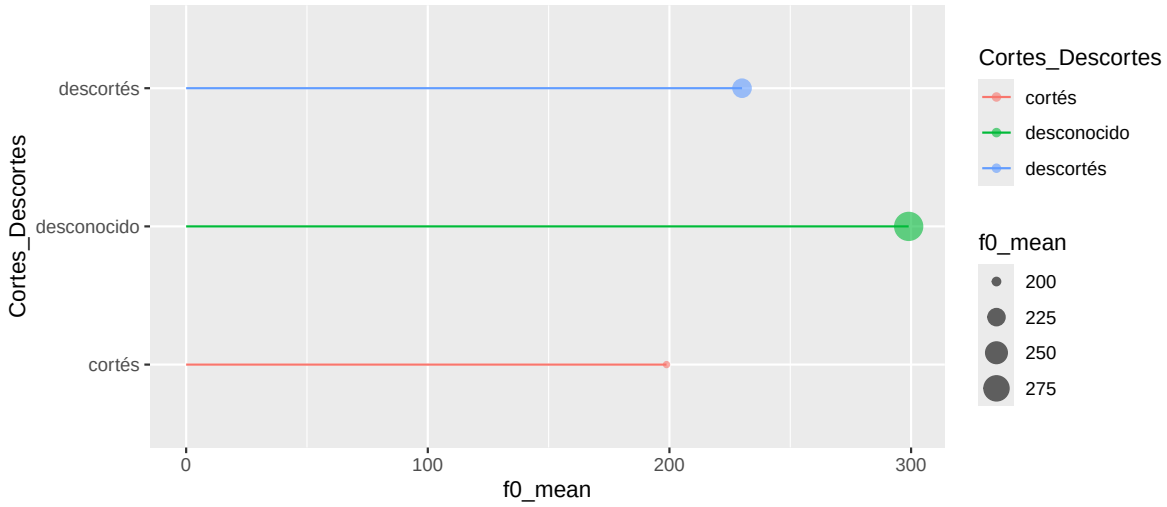
```
ggplot(fonocortesia, aes(x=F0_Media)) +
  geom_density()
```



12.3.10 Gráficos de lolipop

El gráfico de lolipop es una forma de visualizar la distribución de una variable categórica. Muestra la media de una variable continua para cada categoría de la variable categórica.

```
lolipop <- fonocortesia%>%group_by(Cortes_Descortes)%>%
  summarise(f0_mean=mean(F0_Media,na.rm = T))

ggplot(lolipop, aes(x=Cortes_Descortes, y=f0_mean,
                    fill = Cortes_Descortes, color =Cortes_Descortes)) +
  geom_point(aes(size = f0_mean), alpha = 0.6) +
  geom_segment(aes(x=Cortes_Descortes, xend=Cortes_Descortes,
                   y=0, yend=f0_mean)) +coord_flip()
```

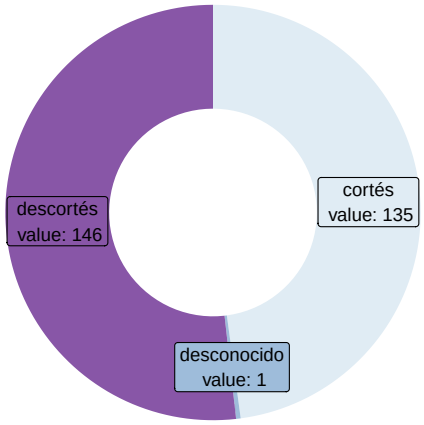


12.3.11 Gráficos de donut

A diferencia de lo que puede pensarse, el gráfico circular (o donut) es quizá uno de los que conlleva más secuencias de código. En este caso, se muestra la frecuencia de la variable `Cortes_Descortes` en un gráfico circular.

```
data <- fonocortesia%>%group_by(Cortes_Descortes)%>%
  summarise(count=n())%>%na.omit()%>%
  rename(category=Cortes_Descortes, count=count)

data$fraction <- data$count / sum(data$count)

data$ymax <- cumsum(data$fraction)

data$ymin <- c(0, head(data$ymax, n=-1))

data$labelPosition <- (data$ymax + data$ymin) / 2

data$label <- paste0(data$category, "\n value: ", data$count)

ggplot(data, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=category)) +
  geom_rect() +
  geom_label( x=3.5, aes(y=labelPosition, label=label), size=3) +
  scale_fill_brewer(palette=3) +
  coord_polar(theta="y") +
  xlim(c(2, 4)) +
```

```
  theme_void() +
  theme(legend.position = "none")
```

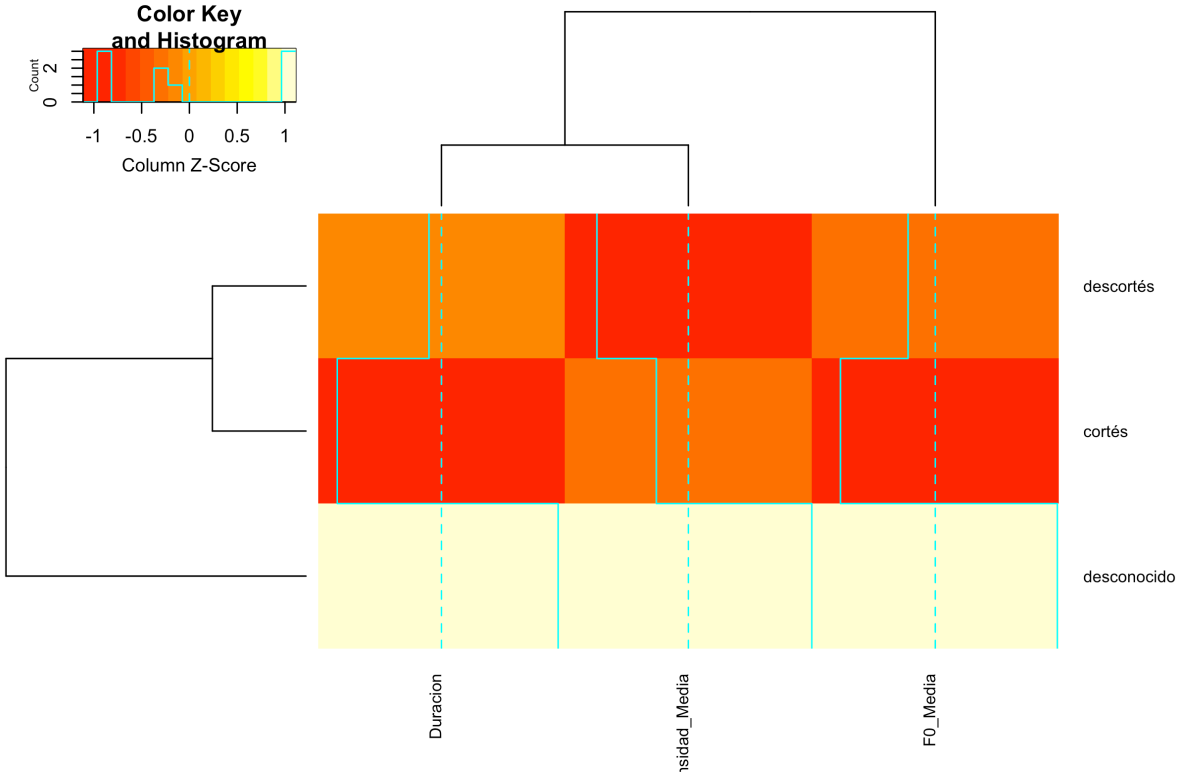


12.3.12 Gráficos de mapa de calor

En este caso, el mapa de calor que utilizamos para ejemplificar esta función no procede de la librería ggplot2, sino de gplots. En este caso, se muestra la media de las variables `F0_Media`, `Duracion` e `Intensidad_Media` agrupadas por la variable `Cortes_Descortes`. Para el mapa de calor es imprescindible disponer de valores numéricos que puedan proyectarse; en este caso, hemos utilizado la media de las variables, pero podrían haberse usado otras medidas de tendencia central. También es importante que no haya valores NA en las variables y que se aplique el procedimiento scale="column" para normalizar los datos. En caso contrario, los resultados no se visualizarán de la manera más adecuado. Por ejemplo, para este mapa de calor se escalan los valores de intensidad, que se miden en decibelios, y los valores de F0, medidos en hercios.

```
library(gplots)
png(filename='heatmap.png', width=2400, height=1550, res=300)


p <- fonocortesia %>%
  group_by(Cortes_Descortes) %>%
  summarise_all(mean, na.rm = TRUE) %>%
  column_to_rownames(var="Cortes_Descortes") %>%
  select_if(is.numeric) %>%
```

```
  select(F0_Media, Duracion, Intensidad_Media)

heatmap.2(as.matrix(p), na.rm = TRUE,
          scale="column", cexCol = 0.8, cexRow = 0.8)
```

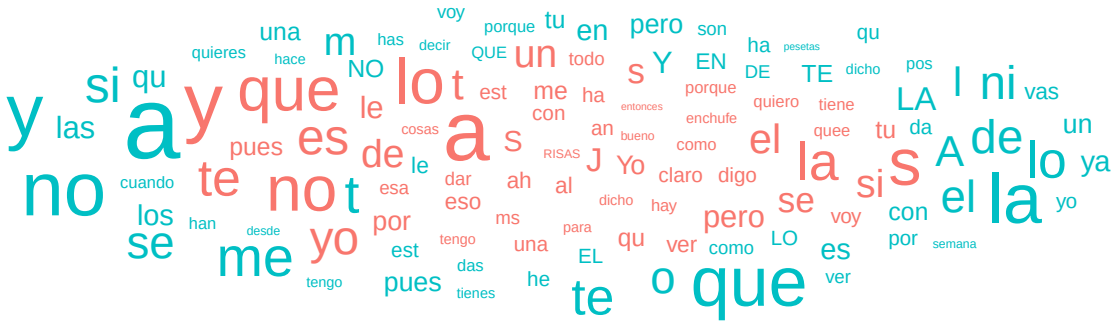


En el mapa de calor anterior, se observa que los enunciados descorteses duran más y tienen un F0 más alta, mientras que los enunciados corteses, a diferencia de lo anterior, presentan intensidades más altas. Debe recordarse que los valoro están relativizados en valores entre -1 y 1.

12.3.13 Nube de palabras

Las nubes de palabras son una forma visual de representar la frecuencia de las palabras en un texto. En este caso, se muestra una nube de palabras con las palabras más frecuentes en la variable `Elemento_Analizado` de la base de datos Fonocortesía. Para ello, se ha utilizado la librería `ggwordcloud`.

```r
library(ggwordcloud)

df_words <- fonocortesia %>%select(Cortes_Descortes,
                                   Elemento_Analizado)%>%
  mutate(word = str_split(Elemento_Analizado, " "),
         cortesia=Cortes_Descortes) %>%  # Dividir el texto en palabras
  unnest(word) %>%  group_by(cortesia,word) %>%
  summarise(frecuencia = n())%>%
  mutate(word2 = gsub("[^a-zA-Z]","",word))%>%filter(frecuencia>2)


ggplot(df_words%>%filter(!word%in%c("A:","B:","C:","L:","M:","(...)")),
       aes(label = word2, size = frecuencia,color=cortesia)) +
  geom_text_wordcloud(area_corr = TRUE) +
  scale_size_area(max_size = 20) +
  theme_minimal()
```



13 Ejercicios

13.1 Enunciados

1. Importa los datos del archivo **`idiolectal.xlsx`** y explora su estructura.
2. Haz dos dataframes según la variable genre. Cada uno de ellos debe contener los datos solo de un género.

3. Crea un dataframe llamado “piquito_relocado” y ubica la variable *tonemes* delante de *genre*
4. Visualiza en un gráfico de líneas en el dataframe “idiolectal” la evolución de los tonemas solo en el archivo *5pangelreal*.
5. Crea un dataframe llamado “piquito_filtrado” en el que filtre todos los datos que no sean NA en la variable *tonemeMAS*.
6. Crea un diagrama de caja de la variable *dur* en el dataframe “idiolectal”. ¿Sabrías crearlos por hablante en un mismo gráfico? Hay varias maneras de hacerlo.
7. ¿Cuántos tonemas hay en el dataframe “idiolectal”? Visualízalo en una tabla y en un gráfico de barras usando la base de datos “piquito_filtrado”.
8. Correlaciona en el dataframe “idiolectal” las variables numéricas de este estudio.
9. Haz una tabla de frecuencias de cada hablante en el dataframe “idiolectal” y saca la media de *tonemeMAS*, de *dur* y de *body*.
10. Visualiza la información anterior en un mapa de calor.

13.2 Soluciones

1. Importa los datos del archivo **`idiolectal.xlsx`** y explora su estructura.

```r
library(readxl)
idiolectal <- read_excel("databases/idiolectal.xlsx")
str(idiolectal)
```

```
tibble [1,218 x 27] (S3: tbl_df/tbl/data.frame)
 $ ...1                : chr [1:1218] "1" "2" "3" "4" ...
 $ filename            : chr [1:1218] "5pangelreal" "5pangelreal" "5pangelreal" "5pangelreal"
 $ spk                 : chr [1:1218] "angelreal" "angelreal" "angelreal" "angelreal" ...
 $ phon                : chr [1:1218] "i*" "e*" "e*" "e*" ...
 $ word                : chr [1:1218] "intrusismo" "momento" "ser" "ser" ...
 $ ip                  : chr [1:1218] "claro a lo mejor ahí sí que se podría considerar intru
 $ ip_dur              : num [1:1218] 2441 716 456 378 1567 ...
 $ tmin                : num [1:1218] 1982616 1983404 1984151 1985149 1986698 ...
 $ tmax                : num [1:1218] 1982703 1983443 1984254 1985221 1986789 ...
 $ words               : num [1:1218] 11 3 2 2 8 1 3 3 9 8 ...
 $ dur                 : num [1:1218] 87 39 103 72 91 35 130 118 57 71 ...
 $ toneme              : chr [1:1218] "yes" "yes" "yes" "yes" ...
 $ desplazamiento      : chr [1:1218] "no" "no" "no" "no" ...
 $ dur_first           : num [1:1218] 31 84 26 20 27 60 130 118 168 37 ...
 $ word_first          : chr [1:1218] "claro" "aquel" "puede" "puede" ...
 $ desplazamiento_first: chr [1:1218] NA NA NA NA ...
 $ phon_preanac        : chr [1:1218] NA "e" NA NA ...
```

```
$ word_preanac        : chr [1:1218] NA "en" NA NA ...
$ dur_preanac         : num [1:1218] NA 34 NA NA 69 NA NA 174 152 51 ...
$ tonemeMAS           : num [1:1218] NA -1.31 NA NA 4.11 ...
$ circunflejo         : chr [1:1218] "no" "no" "no" NA ...
$ MAStag              : chr [1:1218] NA "PV" NA NA ...
$ body                : num [1:1218] NA 26.4 NA NA NA ...
$ spk2                : chr [1:1218] "5pangelreal" "5pangelreal" "5pangelreal" "5pangelreal"
$ spk3                : chr [1:1218] "5pangelrealangelreal" "5pangelrealangelreal" "5pangelr
$ genre               : chr [1:1218] "5p" "5p" "5p" "5p" ...
$ tonemes             : chr [1:1218] NA "suspendido" NA NA ...
```

2. Haz dos dataframes según la variable genre. Cada uno de ellos debe contener los datos solo de un género.

```
table(idiolectal$genre)
```

```
 5p pod
609 609
```

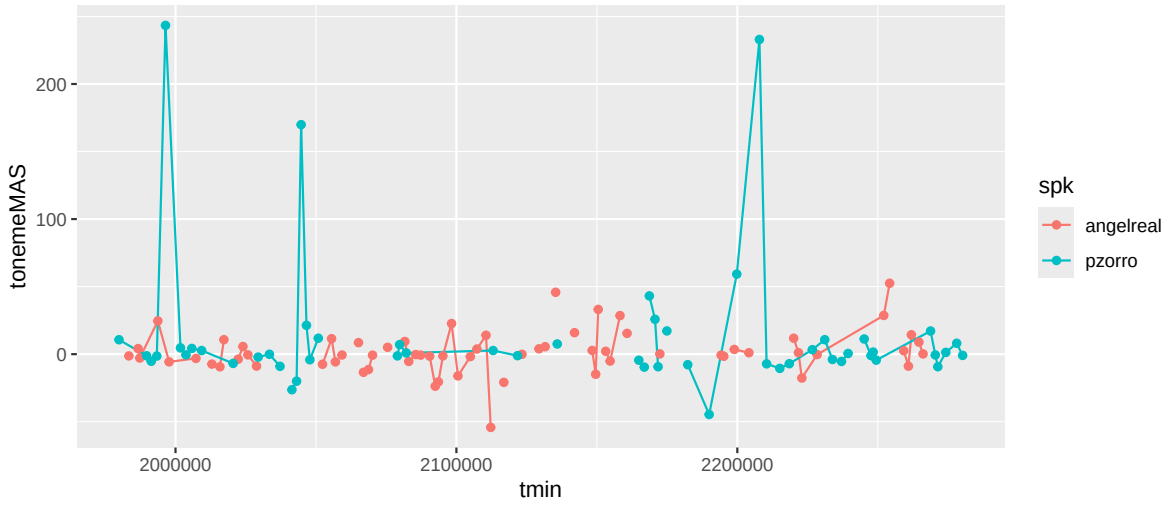
```
idiolectal_5p <- idiolectal%>%filter(genre=="5p")
idiolectal_pod <- idiolectal%>%filter(genre=="pod")
```

3. Crea un dataframe llamado "piquito_relocado" y ubica la variable *tonemes* delante de *genre*

```
piquito_relocado <- idiolectal%>%relocate(tonemes, .before = genre)
```

4. Visualiza en un gráfico de líneas en el dataframe "idiolectal" la evolución de los tonemas solo en el archivo *5pangelreal*.

```
library(tidyverse)
idiolectal%>%filter(filename=="5pangelreal")%>%
  ggplot(aes(x=tmin,y=tonemeMAS, color=spk, fill=spk)) +
  geom_line() + geom_point()
```

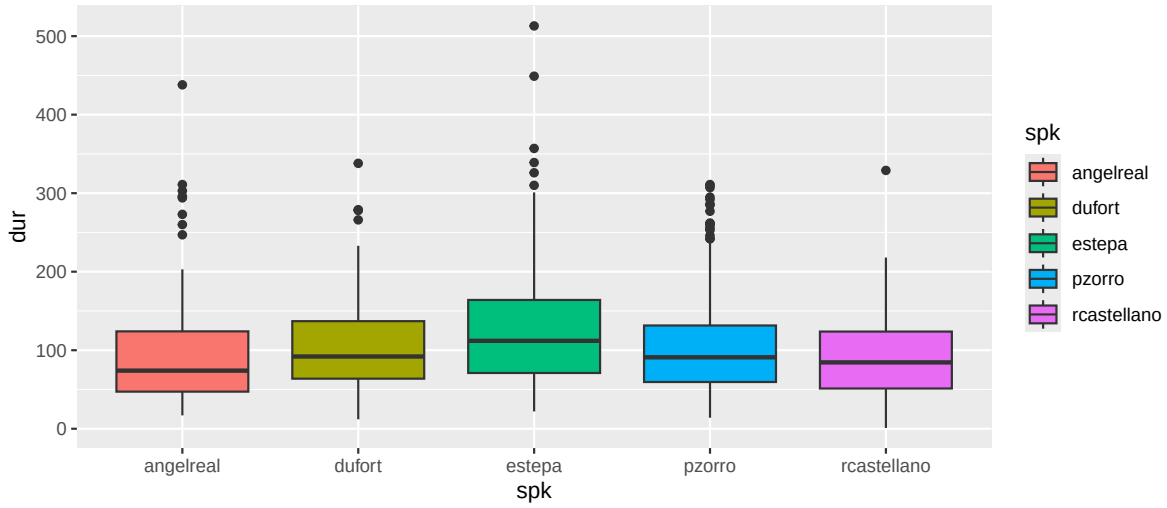


5. Crea un dataframe llamado "piquito_filtrado" en el que filtre todos los datos que no sean NA en la variable *tonemeMAS*.

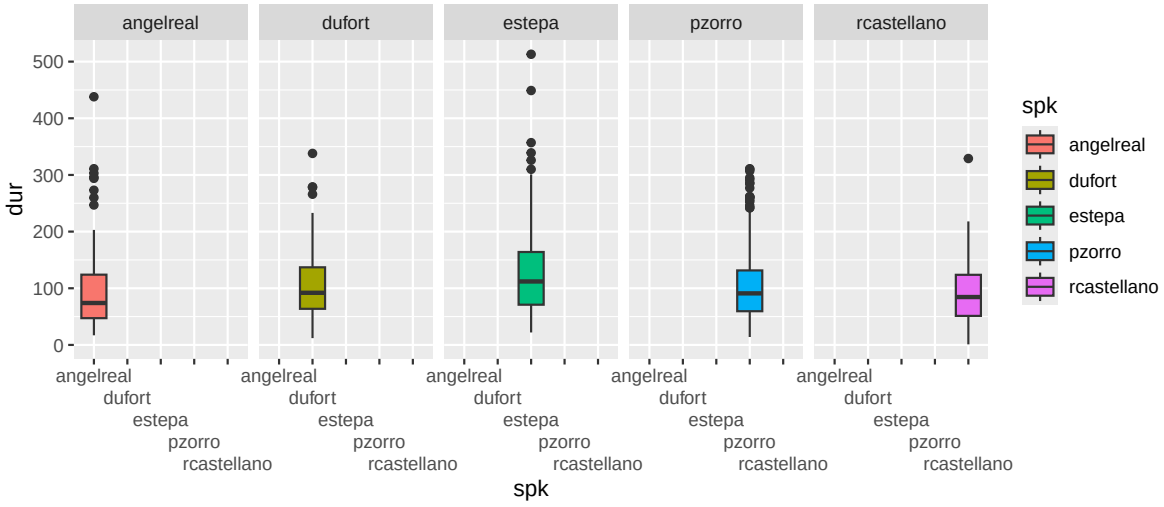
```
piquito_filtrado <- idiolectal%>%filter(!is.na(tonemeMAS))
```

6. Crea un diagrama de caja de la variable *dur* en el dataframe "idiolectal". ¿Sabrías crearlos por hablante en un mismo gráfico? Hay varias maneras de hacerlo.

```
ggplot(idiolectal, aes(x=spk, y=dur, fill=spk)) + geom_boxplot()
```



```
ggplot(idiolectal, aes(x=spk, y=dur, fill=spk)) +
  scale_x_discrete(guide = guide_axis(n.dodge=5))+
  geom_boxplot() + facet_wrap(~spk,ncol = 5)
```

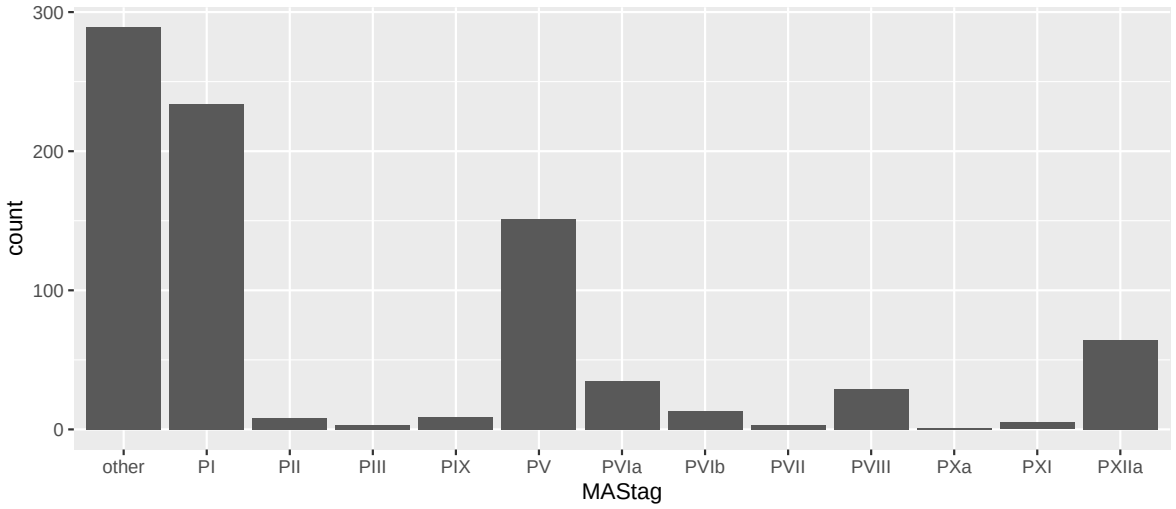


7. ¿Cuántos tonemas hay en el dataframe "idiolectal"? Visualízalo en una tabla y en un gráfico de barras usando la base de datos "piquito_filtrado"

```
table(piquito_filtrado$MAStag)
```

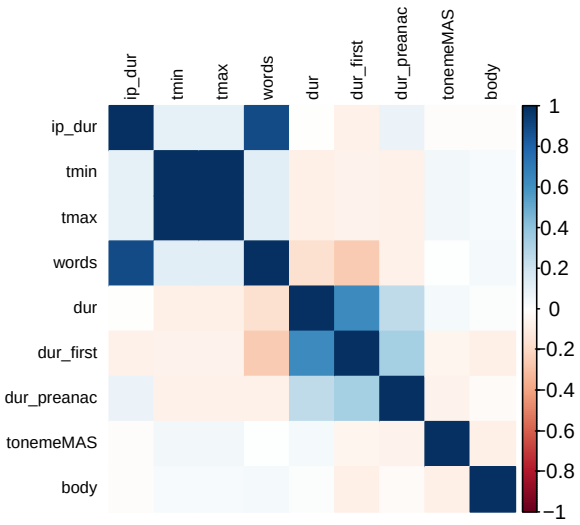
```
other    PI   PII  PIII   PIX    PV  PVIa  PVIb  PVII PVIII   PXa   PXI PXIIa
  289   234     8     3     9   151    35    13     3    29     1     5    64
```

```
ggplot(piquito_filtrado, aes(x=MAStag)) + geom_bar()
```



8. Correlaciona en el dataframe "idiolectal" las variables numéricas de este estudio.

```
library(corrplot)
correlaciones <- cor(idiolectal%>%
                     select_if(is.numeric),use = "complete.obs")
corrplot(correlaciones, method = "color",
         tl.col = "black",tl.cex = 0.7)
```



9. Haz una tabla de frecuencias de cada hablante en el dataframe "idiolectal" y saca la media de *tonemeMAS*, de *dur* y de *body*

```r
idiolectal%>%group_by(spk)%>%summarise(
  tonemeMAS_mean = mean(tonemeMAS,na.rm = T),
  dur_mean = mean(dur,na.rm = T),
  body_mean = mean(body,na.rm = T))
```

```
# A tibble: 5 x 4
  spk         tonemeMAS_mean dur_mean body_mean
  <chr>                <dbl>    <dbl>     <dbl>
1 angelreal             1.02     93.7      5.88
2 dufort                2.75    106.       4.38
3 estepa                0.240   130.      -4.50
4 pzorro               11.0      99.8      7.28
5 rcastellano           1.10     87.0     15.4
```

10. Visualiza la información anterior en un mapa de calor.

```r
library(gplots)
mapacalor <- idiolectal%>%group_by(spk)%>%summarise(
tonemeMAS_mean = mean(tonemeMAS,na.rm = T),
dur_mean = mean(dur,na.rm = T),
body_mean = mean(body,na.rm = T))%>%
  column_to_rownames(var="spk")
heatmap.2(as.matrix(mapacalor), na.rm = TRUE,
scale="column", cexRow = 0.5, cexCol = 0.5)
```

14 Estadística descriptiva

Cuando hablamos de estadística descriptiva normalmente nos estamos refiriendo a un conjunto de operaciones básicas como la media, la mediana, la moda, la desviación típica, etc. También entraría dentro de la estadística descriptiva el estudio de las distribuciones de las variables de nuestras bases de datos.

14.1 Precaución

> ⚠ Correlaciones espúreas (ápud Levshina, 2015)
>
> Los cómputos estadísticos solo relacionan números; el valor explicativo que otorgamos a esas relaciones es una cuestión de interpretación. Por ejemplo, la correlación entre el número de personas que se ahogan en una piscina y el número de películas en las que aparece Nicolas Cage es del 66%. Esto no significa que una cosa cause la otra, sino que ambas variables están relacionadas con el tiempo. Pueden verse más de estas relaciones aquí:
> https://www.tylervigen.com/spurious-correlations

14.2 Tratamiento previo

Antes de empezar este apartado sobre estadística descriptiva crearemos una modificación de la base de datos *fonocortesía* que hemos utilizado en el tema anterior. En esta nueva base de datos, que llamaremos *corpusren*, hemos renombrado las variables para que sean más accesibles en gráficos y tablas y hemos eliminado los casos en los que la variable *Cortes_Descortes* tiene el valor *desconocido*. También hemos eliminado los casos en los que la variable *Tonema* tiene el valor *desconocido*. La base de datos *corpusren* es la que utilizaremos en este tema.

```
library(kableExtra)
library(tidyverse)
library(corrplot)
library(psych)
library(flextable)
library(FactoMineR)
library(factoextra)
library(partykit)
library(readxl)
library(randomForest)
library(DataExplorer)
library(gplots)
library(ggwordcloud)
corpus <- read_excel("databases/corpus.xlsx")
corpusren <- corpus %>% rename(



  conv = Conversacion,  cort = Cortes_Descortes,
```

```
  Llam = Llama_Atencion,  med = Mediodeexpresion,
  F0I = F0_Inicial,  F0F = F0_Final,
  F0M = F0_Media,  F0X = F0_Maxima,  F0N = F0_Minima,
  IU = Intensidad_Ultima,  IM = Intensidad_Media,
  IN = Intensidad_Minima,  IX = Intensidad_Maxima,
  IP = Intensidad_Primera,  Sil = Silabas,
  dur = Duracion,  DPA = Duracion_Pausa_Anterior,
  DPP = Duracion_Pausa_Posterior,  CP = Continuacion_Pausa,
  Cur = Curva_Melodica,  OCur = Otro_Curva_Melodica,
  ILI = Inflexion_Local_Interna,  To = Tonema,
  Disc = Unidad_Del_Discurso,  Vmod = Valormodal,
  OVmod = Otro_Valor_Modal,  Fton = Fenomeno_Tonal,
  Fdur = Fenomeno_Duracion,  Fpau = Fenomeno_Pausas,
  Fve = Fenomeno_Velocidad,  Fam = Fenomeno_Amplitud,
  UF = Unidad_Fonica,  EPra = Estrategia_Pragmatica,
  Efec = Efecto_Pragmatico_Asociado,  EA = Elemento_Analizado,
  Fr = Fragmento)%>%filter(cort!="desconocido", To != "desconocido")

options(scipen = 1, digits = 2)
```

14.3 Tablas simples o tablas de contingencia

En R, el método table sirve para crear tablas de frecuencia. Si usamos, por ejemplo, table(corpusren$cort), obtendremos una tabla con las frecuencias de la variable *cort*. Por su parte, una tabla de contingencia es una tabla que muestra la distribución conjunta de dos o más variables categóricas. En R, podemos crear tablas de contingencia con la función table, que nos permite cruzar dos o más variables categóricas, pero también con otras librerías, como *flextable*.

i Tablas de contingencia

- Procedimiento table
- Procedimiento *flextable* de la librería flextable.

```
table(corpusren$cort)
```

```
   cortés descortés
      131       145
```

Ejemplo con table

```
set_flextable_defaults(
  font.family = "Times",
  font.size = 11,
  padding = 0,
  font.color = "black",
  table.layout = "autofit",
  digits = 1,
  theme_fun = "theme_box"
  )

p <- as.data.frame(table(corpusren$cort))
flextable(p)
```

Var1	Freq
cortés	131
descortés	145

Ejemplo con la librería *flextable*.

Con la variable tonema:

```
flextable(corpusren%>%group_by(To)%>%
            summarise(cantidad=n())%>%arrange(cantidad))
```

To	cantidad
circunflejo	27
suspendido	67
ascendente	84
descendente	98

Tabla resumen de las frecuencias de la variable Tonema

Procedimiento *prop.table* de la librería *base*. Sirve para calcular las proporciones de las tablas de contingencia.

Proporción por fila:

```
prop.table(table(corpusren$To,corpusren$cort),margin = 1)
```

```
             cortés descortés
ascendente     0.27       0.73
circunflejo    0.44       0.56
descendente    0.51       0.49
suspendido     0.69       0.31
```

Proporción por columna:

```
prop.table(table(corpusren$To,corpusren$cort),margin = 2)
```

```
             cortés descortés
ascendente    0.176      0.421
circunflejo   0.092      0.103
descendente   0.382      0.331
suspendido    0.351      0.145
```

Propoción del total:

```
prop.table(table(corpusren$To,corpusren$cort))
```

```
             cortés descortés
ascendente    0.083      0.221
circunflejo   0.043      0.054
descendente   0.181      0.174
suspendido    0.167      0.076
```

Debemos recordar, no obstante, que nos encontramos en este caso en una fase de descripción y que, por tanto, sin la aplicación de pruebas estadísticas inferenciales no podemos controlar de manera precisa la repercusión de estos datos, que pertenecen a nuestra muestra, en relación con la población general de potenciales enunciados corteses o descorteses del español hablado.

La librería *flextable* permite realizar las mismas dos operaciones que hemos realizado anteriormente, la de observar las proporciones por columnas y por filas, en una misma tabla.

```
proc_freq(corpusren, "cort","To",)%>%fontsize(size = 10,part="all")
```

		To				
cort		**ascendente**	**circunflejo**	**descendente**	**suspendido**	**Total**
cortés	Count	23 (8.3%)	12 (4.3%)	50 (18.1%)	46 (16.7%)	131 (47.5%)
	Mar. pct [(1)]	27.4% ; 17.6%	44.4% ; 9.2%	51.0% ; 38.2%	68.7% ; 35.1%	
descortés	Count	61 (22.1%)	15 (5.4%)	48 (17.4%)	21 (7.6%)	145 (52.5%)
	Mar. pct	72.6% ; 42.1%	55.6% ; 10.3%	49.0% ; 33.1%	31.3% ; 14.5%	
Total	Count	84 (30.4%)	27 (9.8%)	98 (35.5%)	67 (24.3%)	276 (100.0%)

[(1)] Columns and rows percentages

14.4 Resumen estadístico

Los resúmenes estadísticos permiten explorar de manera inicial una base de datos. Por defecto, el comando general para efectuar esa operación de resumen es *summary*. Con este comando, R ofrecerá una pantalla con una muestra de las frecuencias más pobladas de las variables categóricas y datos de estadística descriptiva como la media, la mediana, los valores mínimos y máximos, los valores vacíos, el primer y el tercer cuartil.

Con finalidad de ejemplificación, podemos observar valores de algunas variables de la base de datos *Fonocortesía*:

```
kable(summary(corpusren[,c(2,23,5:7)]))
```

cort	To	F0I	F0F	F0M
Length:276	Length:276	Min. : 0	Min. : 0	Min. : 0
Class :character	Class :character	1st Qu.:166	1st Qu.:146	1st Qu.:180
Mode :character	Mode :character	Median :220	Median :211	Median :221
NA	NA	Mean :214	Mean :212	Mean :216
NA	NA	3rd Qu.:252	3rd Qu.:267	3rd Qu.:251
NA	NA	Max. :490	Max. :519	Max. :420
NA	NA	NA's :41	NA's :41	NA

¿Para qué pueden servir entonces las tablas para analizar nuestras bases de datos? Por ejemplo, nos sirven para observar equilibrio en los datos. En Fonocortesía, ¿se han recogido un número similar de casos de cada conversación? Si quisiéramos conocer, en ese sentido, el total de recuento por conversación podríamos ejecutar el comando *table*. En el siguiente ejemplo, se combina *table*, con la propiedad *arrange* de la librería *dplyr* que permite ordenar los datos:

```
flextable(table(corpusren$conv) %>%
            as.data.frame() %>% arrange(desc(Freq)))
```

Var1	Freq
VALESCO 025A	45
VALESCO 130A	39
VALESCO 171A	24
VALESCO 114A	23
VALESCO 183A	22
VALESCO 37B	21
VALESCO 84A	20
VALESCO 140A	19
VALESCO 80A	11
VALESCO 129B	10
VALESCO 69A	10
VALESCO 162A	9
VALESCO 194A	8
VALESCO 126A	7
VALESCO 165A	3
VALESCO 179B	3
VALESCO 193A	1
VALESCO 279b	1

En general, hay una distribución más o menos constante por conversación, de 10 a 20 casos de media, aunque la 25A y la 130A, como decíamos anteriormente, recogen el mayor número de enunciados corteses o descorteses recogidos.

14.4.1 Valores estadísticos generales de una variable

También puede darse la situación de que estemos interesados en una variable en particular. Por ejemplo, si queremos recoger valores estadísticos para la variable *F0_Media (F0M)*, podemos

usar el comando *describe* de la librería *psych*. Los valores concretos que se recogerán serán los siguientes:

```
describe(corpusren$F0M)
```

```
   vars   n mean sd median trimmed mad min max range skew kurtosis  se
X1    1 276  216 61    221     217  52   0 420   420 -0.3        1 3.7
```

Podemos describir cada uno de estos valores estadísticos:

1. *Vars*. Número de variables analizadas. En este caso hemos tomado los valores únicamente de la variable F0_media.
2. *n*. Cantidad de elementos: 276
3. *Mean*. Media de la variable, que serían 216 Hz.
4. *Sd*. Desviación típica, es decir, los datos se desvían 61 Hz de media por encima o por debajo, precisamente, de la media de 216 Hz que se indicaba previamente.
5. *Median*. Valor central de la variable, por encima y por debajo del cual se sitúan la mitad superior e inferior de los datos. Es un valor robusto que no se ve afectado por la presencia de datos extremos positivos o negativos. En este caso, sería algo superior a la media: 221 Hz.
6. *Trimmed*. Se trata de la media con valores extremos eliminados. Serían 216.6 Hz
7. *mad*. Desviación típica de la mediana. 52 Hz.
8. *min* Valor mínimo recogido para esta variable. En el caso de la F0_media, se recoge 0 porque hay algún dato vacío.
9. *max*. Valor máximo de la variable. Esta opción sirve para detectar casos extremos. Para la F0_media sería 420 Hz, que seguramente sea un error de alguna muestra de audio o, también, puede tratarse de un enunciado afectado por fenómenos paralingüísticos como las risas o por algún ruido medioambiental (golpes, por ejemplo).
10. *range*. Se trata del rango de valores entre el valor mínimo y el valor máximo.
11. *skew*. Es el grado de asimetría. Si se acerca a 0, como es el caso, significa que la distribución de la variable está ciertamente normalizada y que los valores se distribuyen de manera equilibrada alrededor de la media y la mediana y se distribuyen en forma positiva o negativa a los lados siguiendo la regla expuesta anteriormente del 68 %, 95 % y 99.8 %.
12. *kurtosis*. La kurtosis es la forma de la curva de la distribución. Un valor de 1.01 indica que la curva refleja una concentración de valores algo más elevada de lo normal en el centro, es decir, hay más valores cercanos a la media, aunque no es un valor desproporcionado. De hecho, asimetría y kurtosis están relacionados.
13. *se*. Es el error estándar de la media. En el caso de F0_Media indica que la media de nuestra muestra, 216 Hz, se encuentra seguramente a una distancia máxima de 3.7 desviaciones típica de la presunta media de toda la población. En general, valores pequeños

de este valor indican que nuestra muestra es bastante adecuada. Al mismo tiempo, si obtuviéramos más datos, el valor se iría reduciendo de manera exponencial.

14.4.2 Valores estadísticos por grupos

Todos los valores que hemos tomado para la variable en conjunto pueden especificarse por una variable de grupo; por ejemplo, podemos observar los valores de estadística descriptiva para F0_Media según los grupos establecidos en la variable *Cortes_Descortes*; en la librería *psych* puede realizarse mediante el comando *describeBy*:

```
p <- describeBy(x=corpusren$F0M,group = corpusren$cort)
p
```

```
 Descriptive statistics by group
group: cortés
   vars   n mean sd median trimmed mad min max range skew kurtosis  se
X1    1 131  198 66    200     200  71   0 333   333 -0.4     0.27 5.8
------------------------------------------------------------
group: descortés
   vars   n mean sd median trimmed mad min max range skew kurtosis  se
X1    1 145  232 52    228     230  41  92 420   327 0.43     0.94 4.3
```

14.5 Valores del resumen estadístico

En caso de querer acceder a un valor concreto del resumen estadístico, podemos obtenerlos mediante los siguientes comandos:

```
mean(x = corpus$F0_Media)
median(x = corpus$F0_Media)
max(x = corpus$F0_Media)
min(x = corpus$F0_Media)
range(x = corpus$F0_Media)
quantile(x = corpus$F0_Media)
IQR(x = corpus$F0_Media)
var(x = corpus$F0_Media)
sd(x = corpus$F0_Media)
skew(x = corpus$F0_Media)
skew(x = corpus$Duracion)
kurtosi(x = corpus$F0_Media)
```

15 Relaciones de estadística inferencial

Son muchas las pruebas estadísticas o técnicas de visualización que pueden realizarse mediante R. Sin embargo, en esta sección vamos a centrarnos específicamente en las siguientes: la prueba de chi cuadrado, la prueba T Test (o ANOVA, para más de dos categorías), el análisis múltiple de correspondencias, el árbol de decisiones y la prueba de Random Forest.

15.1 Chi cuadrado

En palabras de Moore (2005: 620): "el estadístico Ji cuadrado es una medida de la diferencia entre los recuentos observados y los recuentos esperados en una tabla de contingencia".

La prueba de chi cuadrado puede emplearse de dos formas distintas en el análisis de datos. En la primera, se busca determinar si alguna frecuencia categórica difiere de lo esperado bajo ciertas condiciones de proporcionalidad, que pueden ser de equilibrio (igual proporción esperada para cada categoría) o de un desajuste conocido (por ejemplo, sabemos que en la conversación espontánea habrá proporcionalmente más casos de cortesía que de descortesía, con una proporción conocida de 70% de cortesía y 30% de descortesía). Esta aplicación de la chi cuadrado, conocida como prueba de bondad de ajuste, contrapone las frecuencias de las categorías de una única variable.

En la segunda aplicación de la chi cuadrado, que es la más común, se analizan dos variables categóricas cruzándolas en una tabla de contingencia. El objetivo es observar si existe una relación de dependencia entre las dos variables; es decir, si una categoría de una variable específica (por ejemplo, hombre de la variable *sexo*) está relacionada con una categoría o categorías de otra variable (por ejemplo, descortesía de la variable *Cortes_Descortes*). Si esta relación es significativa, encontraremos una cantidad considerable de casos en este cruce de variables.

Según Moore (2013: 621):

> Interpreta el estadístico Ji cuadrado, X2, como una medida de la distancia entre los recuentos observados y los recuentos esperados. Como cualquier distancia, su valor siempre es cero o positivo. Es cero sólo cuando los recuentos observados son exactamente iguales a los recuentos esperados. Los valores de X2 grandes constituyen una evidencia en contra de H0, ya que indican que los recuentos observados están lejos de lo que esperaríamos si H0 fuera cierta. Aunque la hipótesis alternativa Ha es de muchas colas, la prueba Ji cuadrado es de una cola, ya que cualquier violación de H0 tiende a producir un valor de X2 grande. Los valores pequeños de X2 no constituyen ninguna evidencia en contra de H0.

La prueba de chi cuadrado determina la posible relación entre variables, pero no identifica inicialmente qué categorías están significativamente relacionadas. Para ello, se utiliza el residuo

estandarizado, un valor calculado por la prueba. Residuales mayores o menores a 1.96 indican que las categorías están excesivamente representadas o subrepresentadas en los datos, sugiriendo una relación significativa entre las variables si no existiera relación entre ellas.

Dado que es improbable que no haya relación alguna, un residuo significativo sugiere una alta probabilidad de que la hipótesis nula (H0) sea falsa, indicando una relación entre las variables. Además, la fuerza de esta relación no se mide por el valor de la prueba ni por el valor p, sino por otros indicadores como la V de Cramer, donde valores superiores al 60-70 % indican una relación fuerte.

La prueba de chi cuadrado proporciona varios resultados clave:

- **Datos observados**: recuento por categoría o cruces de categorías.
- **Datos esperados**: frecuencia esperada bajo la hipótesis de no relación entre las variables.
- **Grados de libertad**: parámetros técnicos que ayudan a interpretar las relaciones inferenciales. Se calculan como (número de filas - 1) x (número de columnas - 1). Por ejemplo, una tabla 2x2 tiene un grado de libertad, mientras que una 2x3 tiene dos.
- **Valor del estadístico**: un valor alto sugiere rechazar H0, mientras que un valor bajo sugiere lo contrario.
- **Valor p**: si es inferior a 0.05, indica una relación significativa entre las variables al nivel de significación del 95 %, común en estudios científicos.
- **Residuos**: los residuos estandarizados de Pearson comparan la diferencia entre los datos observados y esperados, proyectándolos en una estandarización normalizada. Un valor superior o inferior a 1.96 sugiere que el cruce de categorías está significativamente por encima o por debajo de lo esperado bajo la hipótesis de no relación.

15.1.1 Bondad de ajuste

La bondad de ajuste es una prueba de chi cuadrado que se utiliza para comparar un modelo nulo con un modelo alternativo. En este caso, el modelo nulo es que no hay relación entre las variables y el modelo alternativo es que sí la hay. En el caso de la bondad de ajuste, se compara una variable categórica con una distribución de probabilidad conocida. En el caso de la fonocortesía, por ejemplo, podríamos comparar la variable *Cortes_Descortes* con una distribución de probabilidad conocida, como la proporción 1/2, 1/2. En este caso, la prueba de chi cuadrado nos dirá si la distribución de la variable *Cortes_Descortes* se ajusta a la distribución 1/2, 1/2.

```
corpusren %>%group_by(cort)%>%summarise(cantidad=n())%>%
  arrange(desc(cantidad))%>%flextable()
```

cort	cantidad
descortés	145
cortés	131

Tabla de frecuencias absolutas de la variable combinación fonética

Valores de la prueba chi cuadrado

```
table <- table(corpusren%>%select(cort))
chi <- chisq.test(table,p = c(1/2,1/2))
chi
```

```
    Chi-squared test for given probabilities

data:  table
X-squared = 0.7, df = 1, p-value = 0.4
```

Valores de los residuos estandarizados

```
chi$residuals
```

```
cort
   cortés descortés
     -0.6       0.6
```

La interpretación de los datos señala que hay un equilibro entre los valores corteses y descorteses y que, por tanto, no se han recogido datos desajustados o con pesos distintos.

15.1.2 Chi cuadrado entre dos variables

La prueba de chi-cuadrado es una de las más conocidas en disciplina humanística para observar la relación entre dos variables categóricas. En esta sección vamos a analizar relaciones de variables de las dos bases de datos de este estudio. En primer lugar, observaremos la relación entre el tipo de combinación fonética y género para transmitir atenuación; en segundo lugar, analizaremos la relación entre la variable Cortes_Descortes y los tonemas.

15.1.2.1 Relación entre cortesía y tonemas

En este ejemplo, se analiza la relación entre dos variables de la base de datos Fonocortesía: **Cortes_Descortes** y **tonema**. El objetivo es determinar si la parte final de los enunciados presenta un comportamiento melódico diferente según la categoría cortés o descortés.

Primero, se guarda el resultado de la prueba de chi cuadrado en una variable llamada `pruebachi`. Aunque no es esencial, esta práctica facilita el acceso a valores observados, esperados y residuos. Además, se utiliza el comando `assocstats` de la librería `vcd`, que proporciona datos relevantes como la V de Cramer, útil para medir la fuerza de la relación entre dos variables.

El siguiente código permite acceder al valor de la prueba de chi cuadrado y a los datos generados con `assocstats`:

```
library(broom)
pruebachi <- chisq.test(table(corpusren$cort,corpusren$To))
pruebachi
```

```
    Pearson's Chi-squared test

data:  table(corpusren$cort, corpusren$To)
X-squared = 26, df = 3, p-value = 8e-06
```

```
assocstats(table(corpusren$cort,corpusren$To))
```

```
                    X^2 df   P(> X^2)
Likelihood Ratio 27.053  3 5.7384e-06
Pearson          26.250  3 8.4528e-06

Phi-Coefficient   : NA
Contingency Coeff.: 0.29
Cramer's V        : 0.31
```

- Likelihood ratio test y Pearson's Chi-squared test son una pruebas de bondad de ajuste que se utiliza para comparar un modelo nulo con un modelo alternativo.
- Phi-coefficient, Contingency coefficient y Cramer's V son medidas de la fuerza de la relación entre dos variables categóricas. Sus valores se sitúan entre 0 y 1, siendo 0 la ausencia de relación y 1 la relación perfecta. La relación entre cortesía y tonemas es de 0.29 o 0.31, lo que indica que no todas las categorías están igualmente emparentadas.

En este caso Phi_Coefficient es NA porque solo se puede calcular para variables que tengan solo dos categorías.

Para acceder a los valores observados, esperados y residuos, el código es el que sigue:

15.1.2.2 Valores observados

```
pruebachi$observed
```

```
           ascendente circunflejo descendente suspendido
cortés             23          12          50         46
descortés          61          15          48         21
```

15.1.2.3 Valores esperados

```
pruebachi$expected
```

```
           ascendente circunflejo descendente suspendido
cortés             40          13          47         32
descortés          44          14          51         35
```

15.1.2.4 Valores de los residuos

```
pruebachi$residuals
```

```
           ascendente circunflejo descendente suspendido
cortés          -2.67       -0.23        0.51       2.52
descortés        2.54        0.22       -0.49      -2.39
```

Los residuos estandarizados muestran una relación significativa entre el tonema suspendido y la categoría cortés, así como entre la categoría descortés y el tonema ascendente. Desde un punto de vista epistemológico, estos datos son relevantes porque sugieren un comportamiento diferente al comúnmente aceptado en la comunidad científica. Según el conocimiento compartido, los valores de cortesía suelen estar asociados a subidas tonales; sin embargo, en nuestra muestra, se observa lo contrario.

Para una mayor claridad, los datos de los residuos pueden representarse en un gráfico llamado mosaicplot, como se muestra a continuación:

```
  mosaicplot(table(corpusren$cort,corpusren$To),

main =" Mosaicplot cortesía vs Tonema", shade = TRUE,cex.axis = 0.5,las = 1
```

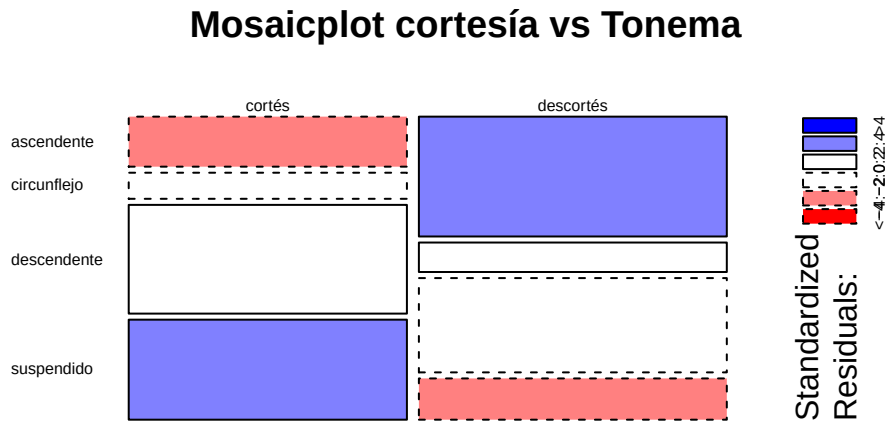


Figure 1: Mosaicplot de Cortes_Descortes y Tonema

15.1.3 Ejercicios

15.1.3.1 Enunciados

1. En la base de datos *Idiolectal*, realizar una prueba de bondad de ajuste para la variable *tonemes*. ¿Hay alguna categoría que se desvíe de la proporción esperada (igualdad de proporción)?
2. En la base de datos *Idiolectal*, realizar una prueba de chi cuadrado para la relación entre la variable *tonemes* y la variable *genre*. ¿Hay alguna relación significativa entre ambas variables?
3. Realiza un mosaicplot para la relación entre la variable *tonemes* y la variable *genre* en la base de datos *Idiolectal*.

15.1.3.2 Soluciones

❗ Práctica

1. En la base de datos *Idiolectal*, realizar una prueba de bondad de ajuste para la variable *tonemes*. ¿Hay alguna categoría que se desvíe de la proporción esperada (igualdad de proporción)?

```
table <- table(idiolectal%>%select(tonemes)%>%na.omit())
chi <- chisq.test(table,p = c(1/4,1/4,1/4,1/4))
chi
```

```
    Chi-squared test for given probabilities

data:  table
X-squared = 222, df = 3, p-value <2e-16
```

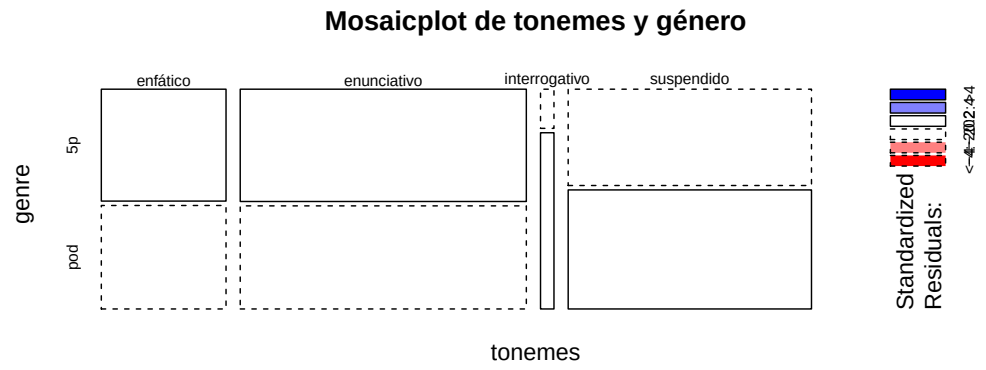
2. En la base de datos *Idiolectal*, realizar una prueba de chi cuadrado para la relación entre la variable *tonemes* y la variable *genre*. ¿Hay alguna relación significativa entre ambas variables?

```
table <- table(idiolectal%>%select(tonemes,genre)%>%na.omit())
chi <- chisq.test(table)
chi
```

```
    Pearson's Chi-squared test

data:  table
X-squared = 7, df = 3, p-value = 0.08
```

3. Realiza un mosaicplot para la relación entre la variable *tonemes* y la variable *genre* en la base de datos *Idiolectal*.

```
mosaicplot(table(idiolectal%>%select(tonemes,genre)%>%
na.omit()),shade=T,main="Mosaicplot de tonemes y género")
```



15.2 T test y ANOVA

Las pruebas T test (o t de Student) y ANOVA (*analysis of variance*) son pruebas usadas para observar la diferencia entre grupos ya creados a partir de una variable numérica. En el caso de la prueba T test se trata de un máximo de dos grupos, mientras que para la ANOVA se parte de más de dos. Un ejemplo concreto sería observar si en la base de datos *Fonocortesía* hay diferencia entre los registros de cortesía y los de descortesía en función de cualquiera de las variables numéricas de las base de datos: duración, F0_Media, Intensidad_Media…

15.2.1 Ejemplo de T. Test: F0 Media según (des)cortesía

En este caso analizamos como variable independiente la variables *Cortes_Descortes* y, por tanto, intentamos averiguar si existe una diferencia notable entre los valores medios de la F0 para los enunciados corteses y para los enunciados descorteses.

```
t.test(corpusren$F0M~corpusren$cort)
```

```
    Welch Two Sample t-test

data:  corpusren$F0M by corpusren$cort
t = -5, df = 245, p-value = 6e-06
alternative hypothesis: true difference in means between group cortés and group descortés is
95 percent confidence interval:
 -47 -19
sample estimates:
```

```
mean in group cortés mean in group descortés
                198                      232
```

Las medias de los enunciados corteses son significativamente inferiores, con 198 Hz, frente a los 232 Hz de los enunciados descorteses. La prueba T de Student con un valor de -5, 245 grados de libertad y un valor p de 6e-06 indica que la diferencia es por tanto significativa y que, en esta situación, nos permite contemplar la presencia de dos grupos distintos. En otras palabras, los enunciados descorteses destacan por una F0 superior a los enunciados corteses.

Las pruebas estadísticas relacionadas con medias suelen acompañarse de algún tipo de gráfico para la visualización de estas. En concreto, podemos acompañar esta prueba de un diagrama de caja que, aunque permite visualizar el valor de la mediana y no la media, es un dato de tendencia central de los datos y, a no ser que haya *outliers* muy marcados, permite fácilmente detectar la diferencia entre grupos:

```
ggplot(corpusren, aes(x=cort, y=F0M, fill = cort)) +
  geom_boxplot() + theme_minimal() +
  labs(title="Diagrama de caja de la relación
       entre F0 Media y Cortes_Descortes",
       x="Cortes_Descortes", y="F0 Media")
```

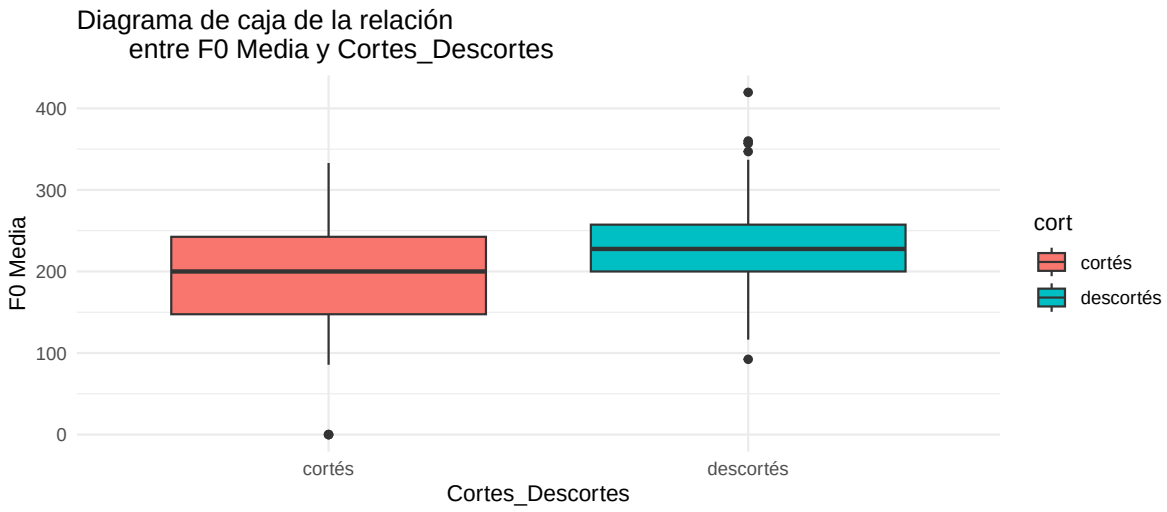


Figure 2: Diagrama de caha de la relación entre F0 Media y Cortes_Descortes

15.2.2 Ejemplos ANOVA: Variables fónicas según medio de expresión

Esta sección nos servirá para introducir la prueba ANOVA (*analysis of variance*), que es una extensión de la prueba T de Student para más de dos grupos. Se usa para comparar las medias

de dos o más grupos y determinar si al menos uno de los grupos es significativamente diferente de los demás. Para ello, se completa con una prueba post hoc, como la de Tukey (HSDTukey), que permite comparar los grupos dos a dos. Como la variable *Medio de expresión* tiene muchas categorías, vamos a simplificarla en tres: Atenuación, Intensificación y Otro valor. Para ello, usaremos el siguiente código:

```
corpusren <- corpusren%>%mutate(med2 = ifelse(

  med == "Atenuación", "Atenuación",
  ifelse(
    med == "Intensificación", "Intensificación",
    "Otro_valor"
  )

  )

)
```

Para aplicar la prueba ANOVA y la de diferenciación de grupos (Tukey), vamos a analizar si hay diferencias significativas entre los tres categorías de la variable *Medio de expresión* en relación con las variables *F0M*, *dur* e *IM*.

15.2.2.1 F0 Media

En primer lugar, observamos la variable *F0M*:

```
TukeyHSD(aov(F0M~med2, data= corpusren))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = F0M ~ med2, data = corpusren)

$med2
                            diff lwr upr p adj
Intensificación-Atenuación   32  11  53  0.00
Otro_valor-Atenuación        -4 -30  22  0.93
Otro_valor-Intensificación  -36 -57 -14  0.00
```

Los datos que se observan indican que hay diferencias significativas entre dos grupos, con valores inferiores a 0.05. En concreto, Intensificación y Atenuación se diferencian entre sí, al

igual que Otro valor de Intensificación, pero no Atenuación de Otro valor. El diagrama de caja que se presenta a continuación muestra la diferencia entre los tres grupos:

15.2.2.1.1 Diagrama de caja

```
ggplot(corpusren, aes(x=med2, y=F0M, fill = med2)) +
  geom_boxplot() + theme_minimal() +
  labs(title="Diagrama de caja de la variable F0M
       por género discursivo",
       x="Género discursivo", y="F0 Media")
```

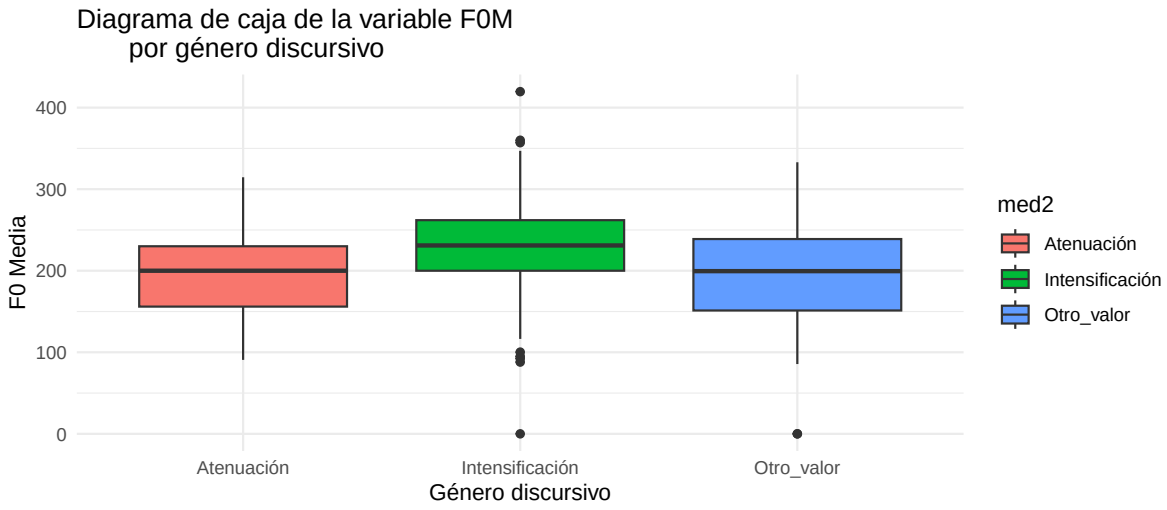


Figure 3: Diagrama de caja de la variable rango tonal por género discursivo

15.2.2.1.2 Descripción por grupos

```
library(psych)
describeBy(x= corpusren$F0M, group = corpusren$med2)
```

```
 Descriptive statistics by group
group: Atenuación
   vars  n mean sd median trimmed mad min max range skew kurtosis  se
X1    1 61  199 53    200     197  49  91 315   224 0.18    -0.66 6.8
------------------------------------------------------------
group: Intensificación
   vars   n mean sd median trimmed mad min max range  skew kurtosis  se
```

```
X1    1 157  230 58    231     231  46   0 420   420 -0.28      1.6 4.6
------------------------------------------------------------
group: Otro_valor
   vars  n mean sd median trimmed mad min max range  skew kurtosis  se
X1    1 58  195 67    200     197  65   0 333   333 -0.52     0.69 8.8
```

15.2.2.2 Duración

En cuanto a la duración, los valores de la prueba ANOVA y el contraste post HOC Tukey son los siguientes:

```
TukeyHSD(aov(dur~med2, data= corpusren))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = dur ~ med2, data = corpusren)

$med2
                             diff   lwr  upr p adj
Intensificación-Atenuación -0.11 -0.95 0.72  0.95
Otro_valor-Atenuación       0.29 -0.73 1.30  0.79
Otro_valor-Intensificación  0.40 -0.45 1.25  0.51
```

En el caso de la velocidad de habla no hay diferencias significativas entre los tres grupos. El diagrama de caja señala esa falta de diferencia:

```
ggplot(corpusren%>%filter(dur<20), aes(x=med2,
 y=dur, fill = med2)) + geom_boxplot() +
  theme_minimal() + labs(title="Diagrama de caja de
  la variable duración por género discursivo",
  x="Género discursivo", y="Duración")
```

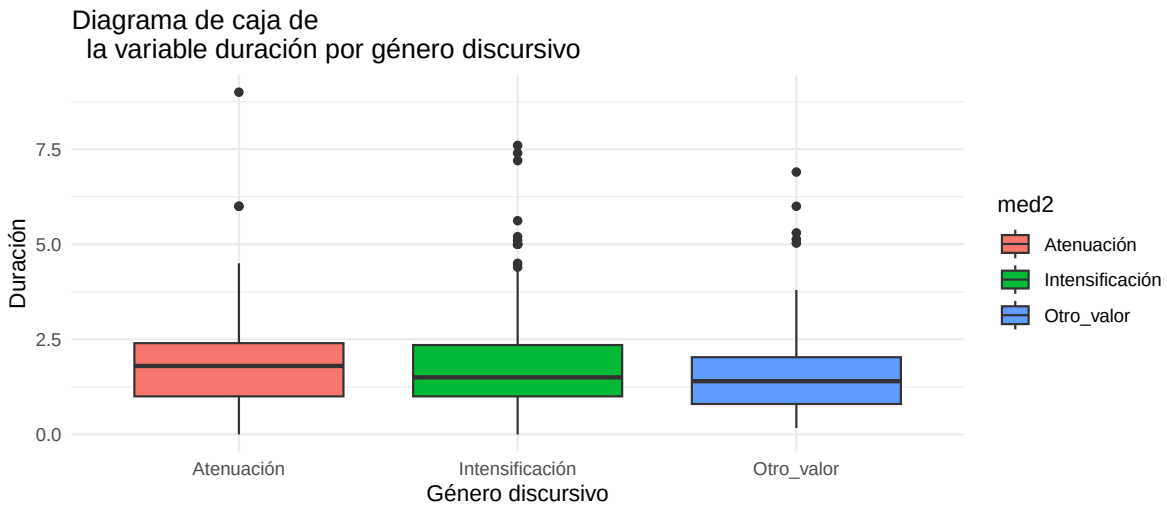


Figure 4: Diagrama de caja de la variable velocidad por género discursivo

15.2.2.3 Intensidad media

Finalmente, la variable de intensidad media tampoco presenta diferencias significativas entre grupos:

```
TukeyHSD(aov(IM~med2, data= corpusren))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = IM ~ med2, data = corpusren)

$med2
                            diff   lwr upr p adj
Intensificación-Atenuación -1.7  -7.4 4.0  0.76
Otro_valor-Atenuación      -3.8 -10.8 3.1  0.40
Otro_valor-Intensificación -2.1  -8.0 3.7  0.67
```

15.2.2.3.1 Descripción por grupos

```
describeBy(x= corpusren$IM, group = corpusren$med2)
```

```
 Descriptive statistics by group
```

```
group: Atenuación
   vars  n mean sd median trimmed mad min max range skew kurtosis  se
X1    1 61   79 15     83      81 5.9   0  90    90   -3       12 1.9
------------------------------------------------------------
group: Intensificación
   vars   n mean sd median trimmed mad min max range skew kurtosis se
X1    1 157   77 13     82      78 8.9  50 143    93 0.21      3.1  1
------------------------------------------------------------
group: Otro_valor
   vars  n mean sd median trimmed mad min max range skew kurtosis  se
X1    1 58   75 24     76      73  17  48 222   174  3.9       22 3.1
```

15.2.2.3.2 Diagrama de caja

```
ggplot(corpusren, aes(x=med2, y=IM, fill = med2)) +
  geom_boxplot() + theme_minimal() +
  labs(title="Diagrama de caja de la variable
       intensidad por género discursivo",
       x="Género discursivo", y="Intensidad Media")
```

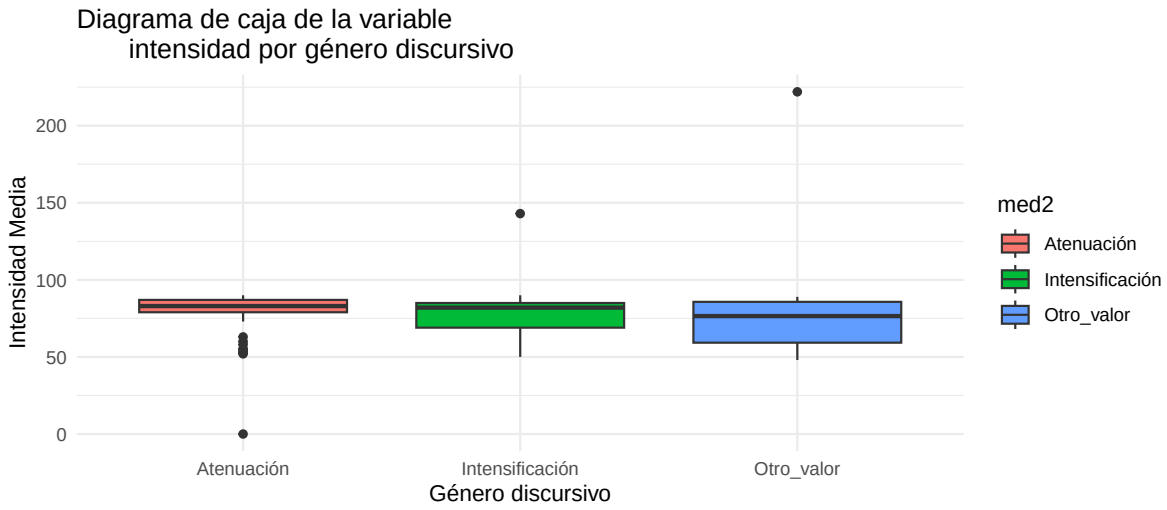


Figure 5: Diagrama de caja de la variable intensidad por género discursivo

15.2.3 Ejercicios

15.2.3.1 Enunciados

1. En la base de datos *Idiolectal*, realizar una prueba ANOVA para la variable *tonemeMAS* según la variable *genre*. ¿Hay alguna diferencia significativa entre los grupos?
2. En la base de datos *Idiolectal*, realizar una prueba T.Test para la variable *dur* según la variable *genre*. ¿Hay alguna diferencia significativa entre los grupos?
3. En la base de datos *Idiolectal*, realiza un boxplot con los datos anteriores.

15.2.3.2 Soluciones

! Práctica

1. En la base de datos *Idiolectal*, realizar una prueba ANOVA para la variable *tonemeMAS* según la variable *genre*. ¿Hay alguna diferencia significativa entre los grupos?

```
TukeyHSD(aov(tonemeMAS~genre, data= idiolectal))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = tonemeMAS ~ genre, data = idiolectal)

$genre
        diff  lwr upr p adj
pod-5p 0.022 -5.8 5.9  0.99
```

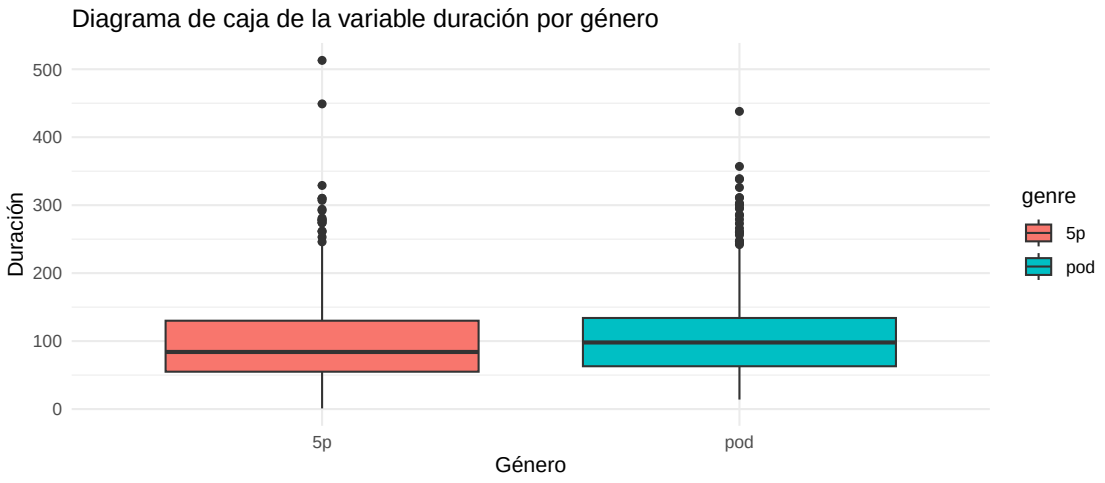
2. En la base de datos *Idiolectal*, realizar una prueba T.Test para la variable *dur* según la variable *genre*. ¿Hay alguna diferencia significativa entre los grupos?

```
t.test(idiolectal$dur~idiolectal$genre)
```

```
    Welch Two Sample t-test

data:  idiolectal$dur by idiolectal$genre
t = -2, df = 1215, p-value = 0.01
alternative hypothesis: true difference in means between group 5p and group pod is not equa
95 percent confidence interval:
 -15.6  -1.7
sample estimates:
 mean in group 5p mean in group pod
               98               107
```

3. En la base de datos *Idiolectal*, realiza un boxplot con los datos anteriores.

```r
ggplot(idiolectal, aes(x=genre, y=dur, fill = genre)) +
  geom_boxplot() + theme_minimal() +
  labs(title="Diagrama de caja de la variable duración por género",
       x="Género", y="Duración")
```



15.3 Análisis múltiple de correspondencias

El análisis múltiple de correspondencia (AMC) es una técnica exploratoria utilizada para visualizar la relación entre más de dos variables categóricas, aunque con solo dos variables se denomina análisis de correspondencias. En R, la librería recomendada para realizar AMC es FactoMineR, complementada por FactoInvestigate, que genera informes automáticos con gráficos y tendencias, y FactoShiny, que permite un análisis interactivo. El AMC es útil para identificar coocurrencias entre categorías sin necesidad de una variable independiente y puede complementarse con el análisis de clúster para agrupar elementos derivados del análisis.

15.3.1 Condiciones de la prueba

El análisis múltiple de correspondencias (AMC) requiere que las variables categóricas sean factores en R, y aunque no tiene restricciones severas, el uso de muchas variables y categorías puede dificultar la visualización gráfica con FactoMineR. Para interpretar el papel de cada categoría en las dimensiones generadas, se recomienda usar el proceso `dimdesc`, que proporciona datos sobre las tres primeras dimensiones. Es común que las dos primeras dimensiones expliquen solo un 15-20% de la variabilidad, lo cual es aceptable ya que AMC es una

herramienta de visualización. Validaciones posteriores, como la chi-cuadrado, determinarán relaciones significativas entre variables.

En palabras de Greenacre (2007:88):

> CA is performed with the objective of accounting for a maximum amount of inertia along the first axis. The second axis accounts for a maximum of the remaining inertia, and so on. Thus the total inertia is also decomposed into components along principal axes, i.e., the principal inertias.

Un 20 % o menos de inercia explicada en las dos primeras dimensiones del AMC debe interpretarse como un núcleo de agrupación, indicando que los elementos en estas dimensiones comparten características definidas de variación. Además, FactoMineR permite crear un dendograma del análisis de clúster basado en el AMC, lo que mejora la visualización de los clústeres formados en las primeras dimensiones del análisis.

15.3.2 Ejemplo de AMC

Para el análisis múltiple de correspondencias (AMC) de la base de datos Fonocortesía, podemos utilizar cualquier variable categórica, excluyendo las columnas referenciales o de ejemplo, como Elemento_analizado o Fragmento. En este ejemplo, utilizaremos cuatro variables para evitar sobresaturar el gráfico resultante: Cortes_Descortes, Llama_Atencion, Tonema y Mediodeexpresion.

Primero, se computa el mapa de dimensiones que proyectará las cercanías entre las categorías estudiadas. Hay varios gráficos que se pueden generar, pero inicialmente nos centraremos en dos: el gráfico de relación entre los individuos y el gráfico de relación entre las variantes. El primero se observa en el siguiente gráfico:

```
library(FactoMineR)
mcacortesia <- MCA(corpusren[c(2,3,4,23)],
                   graph =FALSE,level.ventil = 0.05)
fviz_mca_biplot(mcacortesia, invisible=c("var"))
```

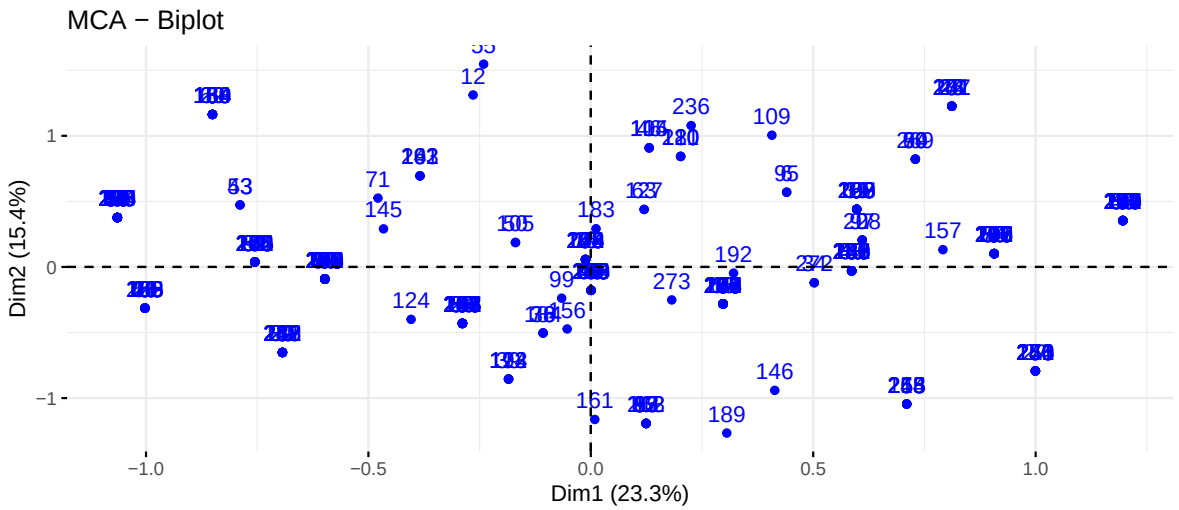


Figure 6: Gráficos extraídos de la prueba ACM

En el gráfico anterior, las dos primeras dimensiones, que suelen ser las más importantes para explicar la variación (Greenacre 2007), explican un 35% de la variación en la base de datos. Este porcentaje es significativo dado que las bases de datos lingüísticas suelen contener muchas variables y categorías.

Es crucial entender el concepto de dimensión en este contexto. Una dimensión busca reducir el espacio de variación entre los datos, similar al análisis de componentes principales (PCA), que también reduce la variación en un conjunto de datos. Las dimensiones del AMC crean un espacio virtual donde las proximidades y distancias se traducen en puntuaciones en un gráfico de ejes.

Las categorías y registros reciben puntuaciones en estas coordenadas según su cercanía o lejanía. Para interpretar estas dimensiones, se utiliza el comando `dimdesc`, que muestra qué variables y categorías puntúan más alto en cada una de las dos principales dimensiones generadas, permitiendo etiquetar cada dimensión (por ejemplo, dimensión de la cortesía, dimensión de la prosodia).

En los gráficos resultantes, la proximidad entre categorías indica que comparten características comunes. Cada categoría representa un grupo de individuos que comparten características específicas de agrupación.

```
fviz_mca_biplot(mcacortesia, invisible=c("ind"))
```

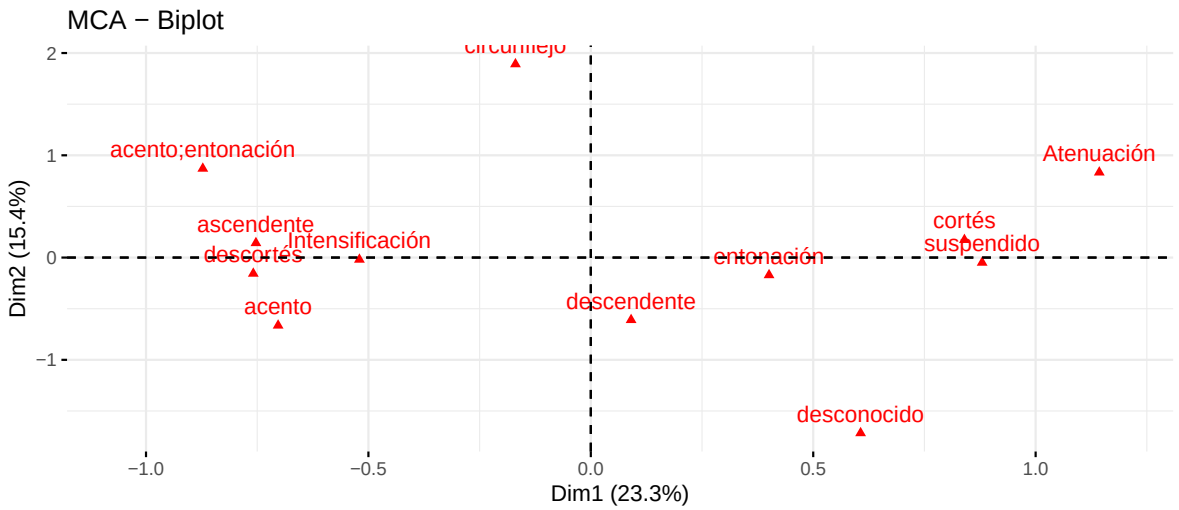


Figure 7: Mapa sobre la relación entre las categorías de algunas variables de la base de datos Fonocortesía

En el gráfico anterior, se observan fenómenos interesantes que una prueba chi cuadrado podrá confirmar o refutar. Inicialmente, vemos que en la primera dimensión, la categoría cortés tiene una puntuación alta, acompañada por la categoría Atenuación, que también tiene una puntuación alta en la segunda dimensión y aún más en la primera. Además, la categoría entonativa suspendido muestra una puntuación alta en la primera dimensión, indicando un grupo de enunciados caracterizados por ser corteses, mayoritariamente con atenuación pragmática y entonación suspendida.

Por otro lado, la categoría descortés se encuentra en el extremo opuesto de la primera dimensión, que podría denominarse "dimensión de cortesía". Esta categoría se asocia con intensificación y tonema ascendente, formando un grupo caracterizado por enunciados descorteses, intensificados y con entonación ascendente.

Las demás categorías proyectadas en el gráfico se relacionan principalmente con cuestiones entonativas en la segunda dimensión, especialmente con la variable Llama la atención, que indica los fenómenos fónicos que en un momento particular despertaron el interés de los investigadores y condicionaron la catalogación de un enunciado como cortés o descortés.

Para profundizar y observar la importancia de cada categoría en cada una de las tres primeras dimensiones, se puede utilizar el siguiente código:

```
dimdesc(mcacortesia,axes = c(1:2))
```

Representación de los *eigenvalues*. Los eigenvalues son una medida de la importancia de cada dimensión en el análisis. En general, se considera que una dimensión es importante si su

eigenvalue es superior a 1. En este caso, las dos primeras dimensiones son las más importantes, con valores de 0.25 y 0.15, respectivamente.

```
fviz_eig(mcacortesia, )
```

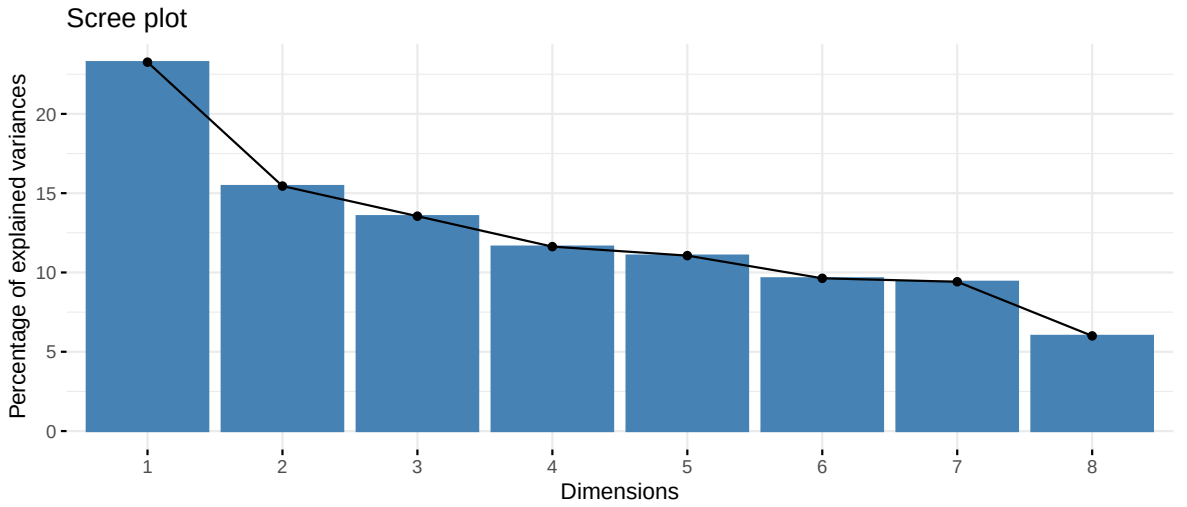
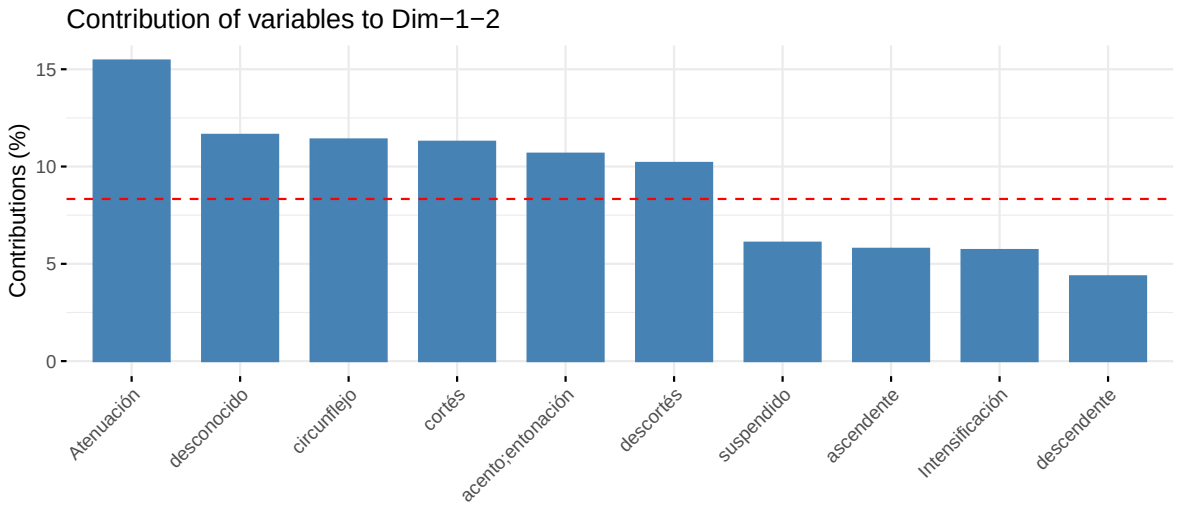


Gráfico sobre la importancia de las categorías en las primeras dos dimensiones:

```
fviz_contrib(mcacortesia, choice = "var", axes = 1:2, top = 10)
```



El resultado del comando `dimdesc` confirma muchas de las observaciones previas. Primero, se proporciona la puntuación de las variables en las dimensiones, acompañada de un valor p que indica significatividad. Luego, para cada dimensión, se puntúan también las categorías, igualmente acompañadas de un valor p.

En la primera dimensión, puntúan muy alto los valores previamente mencionados: enunciados corteses y atenuados, con una notable entonación suspendida. En el extremo opuesto, encontramos enunciados descorteses e intensificados, con un acento y entonación prominentes y vinculados a un tonema ascendente. La segunda dimensión, en cambio, destaca factores más fonéticos y un medio de expresión específico, como la atenuación, que se relaciona más con el tonema circunflejo y otros aspectos fónicos (acento, duración, etc.).

15.3.2.1 Análisis de clúster derivado

El *análisis múltiple de correspondencias* puede completarse con la librería *FactoMiner* mediante el uso del análisis de clúster que, al mismo tiempo, generará un conjunto de grupos que integran además características identitarias. La exploración de los resultados del análisis múltiple de correspondencias hace suponer que realmente hay dos grandes grupos; para no forzar completamente esta agrupación, hemos optado por indicar la creación de 3 grupos, aunque podríamos haber sugerido la creación de más grupos. Esta última opción no tendría demasiado sentido si tomamos en consideración todo lo visto anteriormente.

```
cluster <- HCPC(mcacortesia, nb.clust = 3, graph = FALSE)
fviz_cluster(cluster, geom = "point", palette =
 c("#00AFBB", "#E7B800", "#FC4E07"), ggtheme = theme_minimal())
```

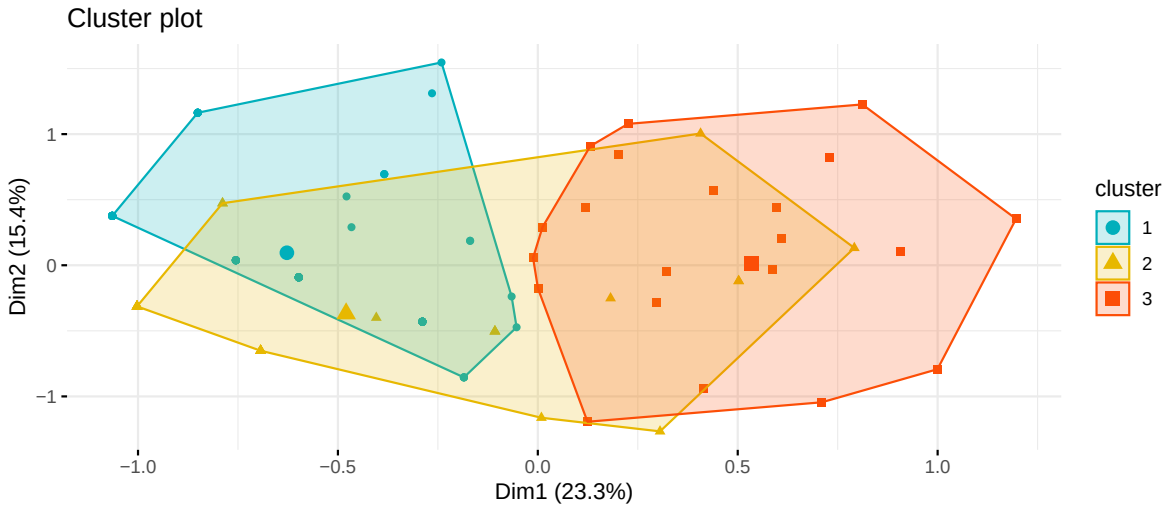


Figure 8: Gráfico del análisis de clúster generado a partir de los resultados del AMC

Los grupos generados se han coloreado con tres colores distintos. De todos modos, con el siguiente código podemos ver cuántos registros constituyen cada grupo y, además, cuáles son las características de cada uno de ellos.

```
table(cluster$data.clust$clust)
cluster$desc.var
```

15.4 Descripción de categorías con FactoMineR

La librería *Factominer* permite realizar una exploración de los datos relacionada parcialmente con el análisis múltiple de correspondencias . El procedimiento se llama *catdes* y permite observar en qué medida las variantes de una variable se correlacionan de modo significativo con categorías o medias numéricas de otras variables. De esta manera, las asociaciones generadas se utilizan para explicar cada una de las categorías de la variable de entrada.

Los elementos que hay que conocer para poder entender la prueba son los siguientes:

1. *Cla/mod*. Se trata del porcentaje de casos de la categoría analizada dentro de la variable que aparece en el resultado.
2. *Mod/Cla*. Se trata del porcentaje de casos de la variable del resultado dentro de la variable de entrada.
3. *Global*. Se trata del porcentaje de casos que representa el cruce de categorías sobre el total.
4. *p.value*. Valor de significación estadística.
5. *V.test*. Si es superior a 0 indica que el valor es superior a la media o frecuencia general esperada; si es inferior a 0 indica que el valor es inferior.

Para entender algo mejor conceptos de esta prueba, como *v.test*, puede acudirse a Le, Josse y Husson (2008).

15.4.1 Desviaciones fónicas y atenuación

```
corpusselect <- corpusren%>%select(cort, To, med2)

catedes <- catdes(corpusselect,num.var = 1,proba = 0.01)
catedes
```

```
Link between the cluster variable and the categorical variables (chi-square test)
=================================================================================
     p.value df
med2 1.6e-16  2
```

```
To   8.5e-06  3

Description of each cluster by the categories
=============================================
$cortés
                     Cla/Mod Mod/Cla Global p.value v.test
med2=Atenuación           92      43     22 2.2e-16    8.2
To=suspendido             69      35     24 6.9e-05    4.0
To=ascendente             27      18     30 8.8e-06   -4.4
med2=Intensificación      28      34     57 6.4e-14   -7.5

$descortés
                     Cla/Mod Mod/Cla Global p.value v.test
med2=Intensificación    72.0    77.9     57 6.4e-14    7.5
To=ascendente           72.6    42.1     30 8.8e-06    4.4
To=suspendido           31.3    14.5     24 6.9e-05   -4.0
med2=Atenuación          8.2     3.4     22 2.2e-16   -8.2
```

```r
plot.catdes(catedes, show= "quali", cex.names = 1.2,
            col.upper = "blue",col.lower = "red")
```

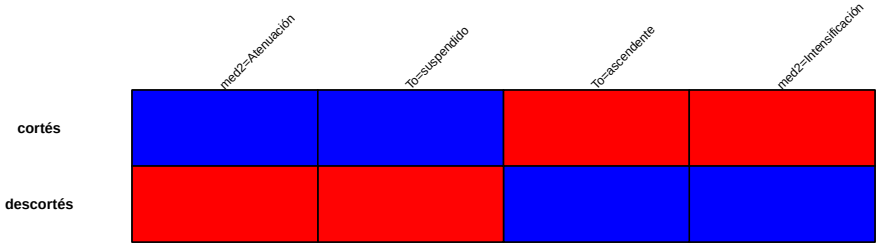


Figure 9: Gráfico para representar la descripción de categorías en FactoMineR

En el anterior gráfico, el color azul señala los casos de sobrepoblación, mientras que el color rojo indica casos de categorías poco pobladas. No es, por tanto, un mapa de calor conven-

cional, ya que en este gráfico solo se representan las categorías o valores que han resultado ser significativamente distintos.

15.4.2 Ejercicios

15.4.2.1 Enunciados

1. En la base de datos *Idiolectal*, realiza una descripción de categorías para la variable *genre* en función de las variables *tonemes*, *spk*, *spk2*, *MAStag* y *circunflejo*.
2. En la base de datos *Idiolectal*, realiza un gráfico con los resultados obtenidos.
3. En la base de datos *Idiolectal*, realiza un análisis múltiple de correspondencias con las variables *genre*, *tonemes*, *spk*, *spk2*, *MAStag* y *circunflejo*.
4. En la base de datos *Idiolectal*, realiza un análisis de clúster con los resultados obtenidos.
5. En la base de datos *Idiolectal*, realiza un gráfico con los resultados obtenidos en el análisis múltiple de correspondencias.

15.4.2.2 Soluciones

! Práctica

1. En la base de datos *Idiolectal*, realiza una descripción de categorías para la variable *genre* en función de las variables *tonemes*, *spk*, *spk2*, *MAStag* y *circunflejo*.

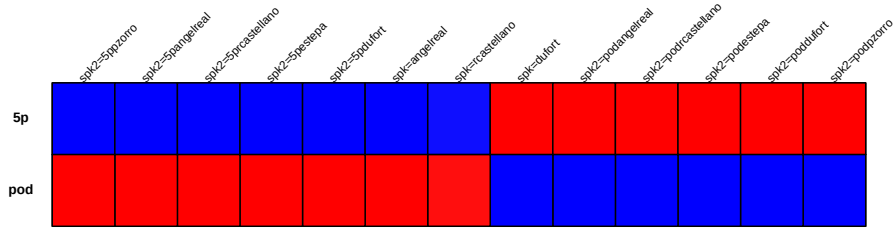
```
idiolectalselect <- idiolectal%>%select(genre,tonemes,spk,spk2,MAStag,circunflejo)

catedes2 <- catdes(idiolectalselect,num.var = 1,proba = 0.01)

catedes2
```

```
Link between the cluster variable and the categorical variables (chi-square test)
================================================================================
      p.value df
spk2 1.6e-256  9
spk   4.6e-15  4

Description of each cluster by the categories
=============================================
$`5p`
                   Cla/Mod Mod/Cla Global  p.value v.test
spk2=5ppzorro          100    46.6   23.3 3.4e-105   21.8
spk2=5pangelreal       100    16.9    8.5  8.8e-34   12.1
```

```
spk2=5prcastellano      100    15.9    8.0  9.9e-32   11.7
spk2=5pestepa           100    13.6    6.8  5.2e-27   10.8
spk2=5pdufort           100     6.9    3.4  1.1e-13    7.4
spk=angelreal            71    16.9   12.0  9.3e-08    5.3
spk=rcastellano          63    15.9   12.6  5.6e-04    3.5
spk=dufort               26     6.9   13.3  2.5e-11   -6.7
spk2=podangelreal         0     0.0    3.5  5.3e-14   -7.5
spk2=podrcastellano       0     0.0    4.7  1.8e-18   -8.8
spk2=podestepa            0     0.0    7.1  5.1e-28  -11.0
spk2=podd ufort           0     0.0    9.9  1.1e-39  -13.2
spk2=podpzorro            0     0.0   24.9 4.7e-114  -22.7

$pod
                    Cla/Mod Mod/Cla Global  p.value v.test
spk2=podpzorro          100    49.8   24.9 4.7e-114   22.7
spk2=podd ufort         100    19.7    9.9  1.1e-39   13.2
spk2=podestepa          100    14.1    7.1  5.1e-28   11.0
spk2=podrcastellano     100     9.4    4.7  1.8e-18    8.8
spk2=podangelreal       100     7.1    3.5  5.3e-14    7.5
spk=dufort               74    19.7   13.3  2.5e-11    6.7
spk=rcastellano          37     9.4   12.6  5.6e-04   -3.5
spk=angelreal            29     7.1   12.0  9.3e-08   -5.3
spk2=5pdufort             0     0.0    3.4  1.1e-13   -7.4
spk2=5pestepa             0     0.0    6.8  5.2e-27  -10.8
spk2=5prcastellano        0     0.0    8.0  9.9e-32  -11.7
spk2=5pangelreal          0     0.0    8.5  8.8e-34  -12.1
spk2=5ppzorro             0     0.0   23.3 3.4e-105  -21.8
```

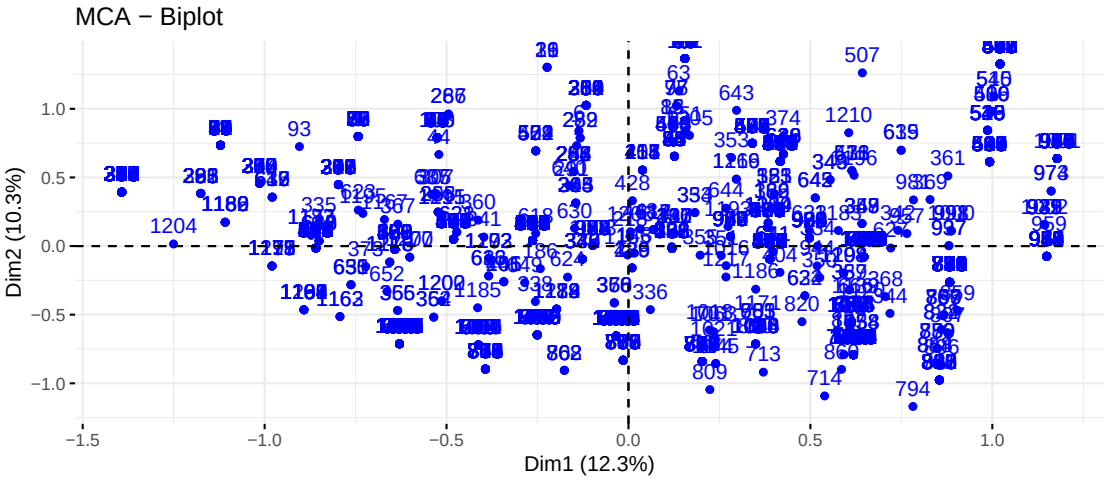
2. En la base de datos *Idiolectal*, realiza un gráfico con los resultados obtenidos.

```
plot.catdes(catedes2, show= "quali", cex.names = 1.2,
            col.upper = "blue",col.lower = "red")
```



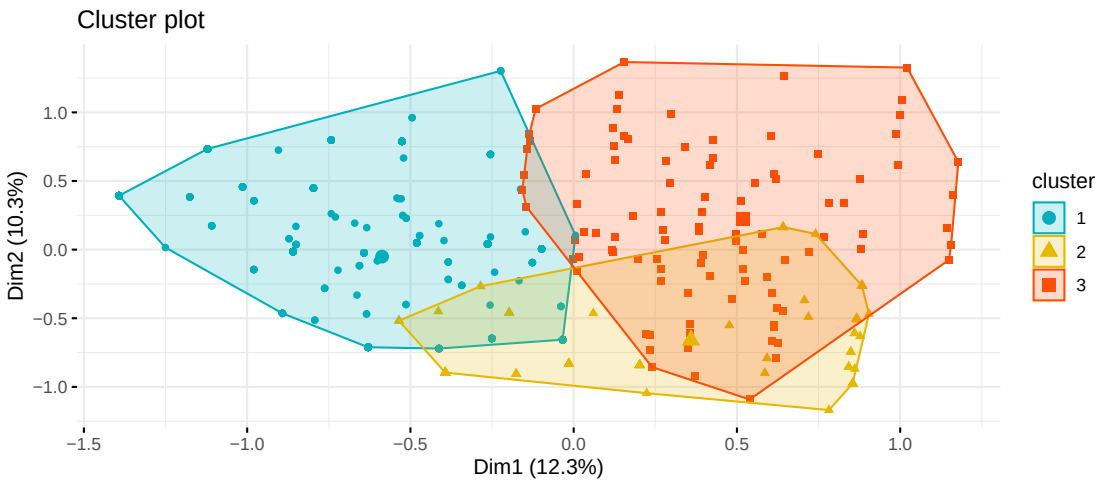
3. En la base de datos *Idiolectal*, realiza una análisis múltiple de correspondencias con las variables *genre*, *tonemes*, *spk*, *spk2*, *MAStag* y *circunflejo*.

```
library(FactoMineR)
library(factoextra)
idiolectalselect2 <- idiolectal%>%select(genre,tonemes,spk,spk2,MAStag,circunflejo)

mcaidiolectal <- MCA(idiolectalselect2,
                     graph =FALSE,level.ventil = 0.05)

fviz_mca_biplot(mcaidiolectal, invisible=c("var"))
```



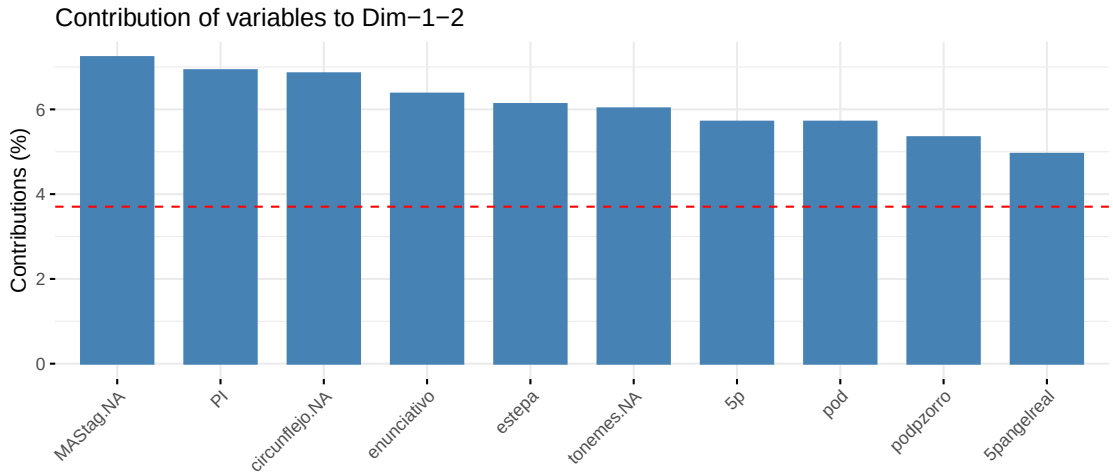
4. En la base de datos *Idiolectal*, realiza un análisis de clúster con los resultados obtenidos.

```
clusteridiolectal <- HCPC(mcaidiolectal, nb.clust = 3, graph = FALSE)

fviz_cluster(clusteridiolectal, geom = "point", palette =
 c("#00AFBB", "#E7B800", "#FC4E07"), ggtheme = theme_minimal())
```

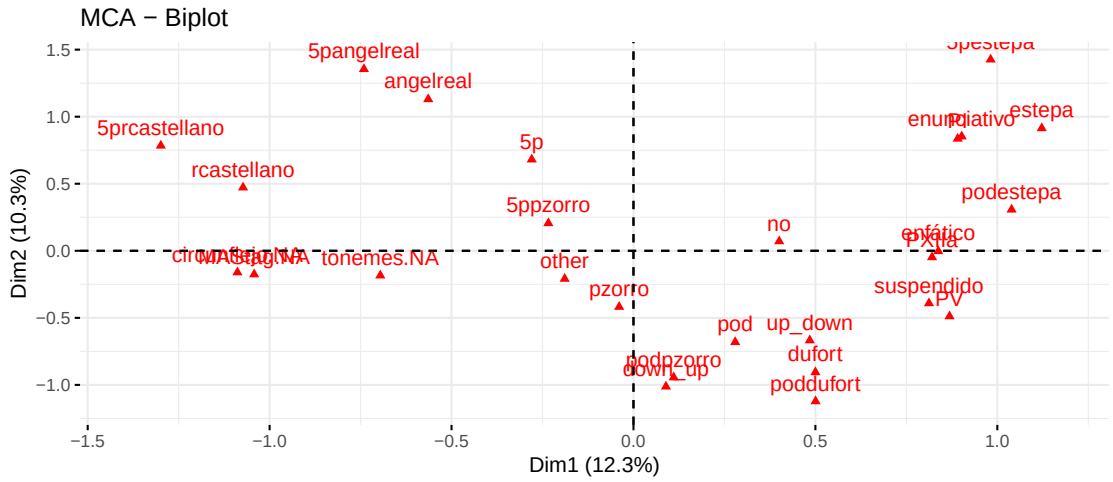


5. En la base de datos *Idiolectal*, realiza un gráfico con los resultados obtenidos en el análisis múltiple de correspondencias.

```
fviz_contrib(mcaidiolectal, choice = "var", axes = 1:2, top = 10)
```




```
fviz_mca_biplot(mcaidiolectal, invisible=c("ind"))
```



15.5 Árbol de decisiones y Random Forest

El árbol de decisiones es una técnica de visualización y reducción de variabilidad ampliamente utilizada para clasificar las categorías de una variable independiente utilizando un conjunto de variables explicativas, que pueden ser cuantitativas o nominales. La técnica Random Forest, derivada del árbol de decisiones, simula datos generando múltiples árboles para mejorar la clasificación. Un procedimiento común en estudios estadísticos es dividir la base de datos en dos partes para calcular la clasificación en una parte y validarla en la otra. Sin embargo, con

bases de datos pequeñas, se puede usar toda la base para identificar reglas de clasificación y las variables más relevantes. Estas técnicas son robustas y dependen de cómo se configura la base de datos, evitando variables redundantes que puedan solapar información. Por ejemplo, variables como duración y número de sílabas pueden ser redundantes en el árbol de decisiones. Las clasificaciones no serán perfectas, siempre habrá variabilidad. Los árboles de decisiones, como el análisis de clúster, proponen secuencias de clasificación según la influencia de varias variables, pero existirán intersecciones entre categorías. Ejemplos detallados se presentarán en las secciones siguientes.

15.5.1 Condiciones de la prueba

Tanto el *árbol de decisiones* como *RandomForest* solo necesitan una variable independiente, numérica o nominal, y un conjunto de variables independientes, numéricas o nominales. Las variables nominales deben ser **variables de factor**. Hay que intentar que las variables numéricas no incluyan valores vacíos, comúnmente llamados NA en R.

15.5.2 Árbol con variable independiente numérica

En Levinson y Torreira (2015) se aplican tanto la técnica del árbol de decisiones como la de Random Forest para caracterizar los valores temporales de FTO (Floor Transfer Offset), una variable independiente numérica que mide el intervalo temporal entre la finalización de un enunciado por un hablante y el comienzo de otro enunciado por otro interlocutor. En su caracterización, se consideran tanto valores fónicos (descenso tonal, duración del grupo entonativo) como categóricos (acto de habla).

Tomando como referencia este ejemplo, analizaremos en esta sección la variable FM0, que mide la media de tono en un enunciado. Aquí, la variable independiente es numérica, por lo que se trata de un árbol de decisiones clásico.

Primero, utilizaremos solo los casos de la variable independiente que no tengan valores perdidos o iguales a 0. Los árboles de decisiones no permiten valores ausentes y no es conveniente incluir valores iguales a 0, ya que estos indican la imposibilidad de registrar valores tonales para ese enunciado, lo que impide calcular el rango tonal. Utilizaremos la librería `partykit`, que ofrece configuraciones útiles para la representación gráfica, evitando solapamientos en las categorías.

Aquí está el código en R para realizar el análisis:

```
corpusarbol <- corpusren%>%filter(F0M>0)
corpusarbol[sapply(corpusarbol, is.character)] <-
  lapply(corpusarbol[sapply(corpusarbol, is.character)],as.factor)
```

```
library(partykit)
```

A continuación, realizamos el primer árbol de decisiones y lo convertimos en gráfico. Si se desea, puede accederse a los datos numéricos y a las clasificaciones que la librería produce. Para ello, únicamente hay que almacenar el cómputo del árbol de decisiones en una variables y, posteriormente, invocarlo desde el terminal. Por ejemplo, en nuestro caso práctico vamos a almacenar el análisis en una variable/objeto que se llama *corpustree* y que podemos introducir directamente en el terminal.

Posteriormente, podemos usar la función *plot* para convertir el árbol de decisiones en un gráfico interpretable. En esta comando, se adjunta una especificación *ep_args = list(justmin = 15)* que se utiliza para que si hay nodos con muchas categorías no se solapen unas con otras.

```
library(dplyr)
corpustree <- partykit::ctree(FOM~cort+IM+To+med2+Vmod,
data= corpusarbol, control = ctree_control(maxdepth = 3))
plot(corpustree, ep_args = list(justmin = 15),
     gp = gpar(fontsize = 8))
```

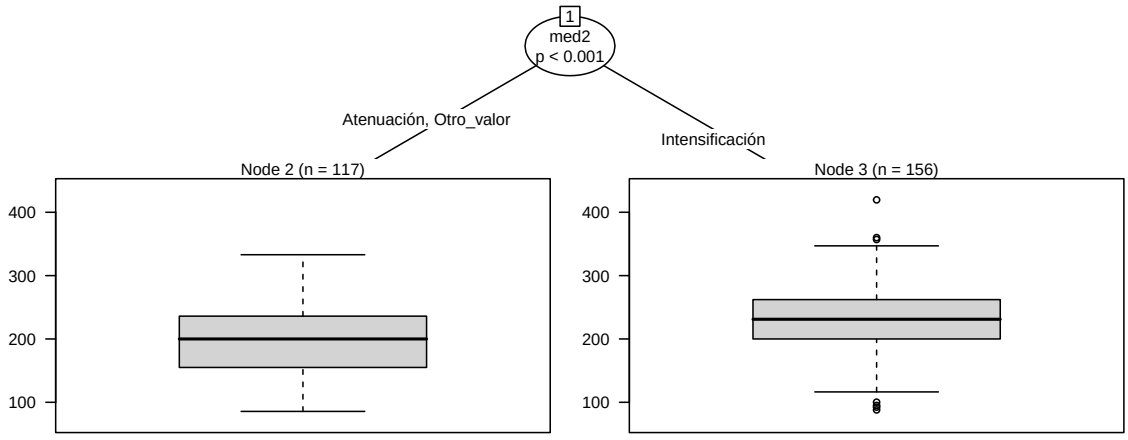


Figure 10: Árbol de decisiones de la variable rango tonal en general

15.5.3 Árbol con variable independiente categórica

En análisis que combinan fonética y pragmática, es común utilizar una variable independiente compuesta por grupos predefinidos. En investigaciones sobre género discursivo y atenuación, se pueden utilizar variables categóricas como el género discursivo, la combinatoria fónica, la

atenuación pragmática y el hablante como variables independientes. En esta sección, ejemplificaremos la clasificación basada en el género discursivo y la atenuación pragmática.

Primero, filtraremos la base de datos para eliminar casos desviados fónicamente que no indiquen atenuación pragmática. Estos casos pueden estar marcados como "sí" (transmiten atenuación pragmática) o "no" (no transmiten atenuación pragmática). Luego, ejecutaremos el gráfico del árbol de decisiones correspondiente.

Aquí está el código en R para realizar el análisis:

```
library(dplyr)
corpustree <- partykit::ctree(cort~F0M+IM+To+med2+Vmod,
        data= corpusarbol, control = ctree_control(maxdepth = 3))
plot(corpustree, ep_args = list(justmin = 15),
     gp = gpar(fontsize = 8))
```

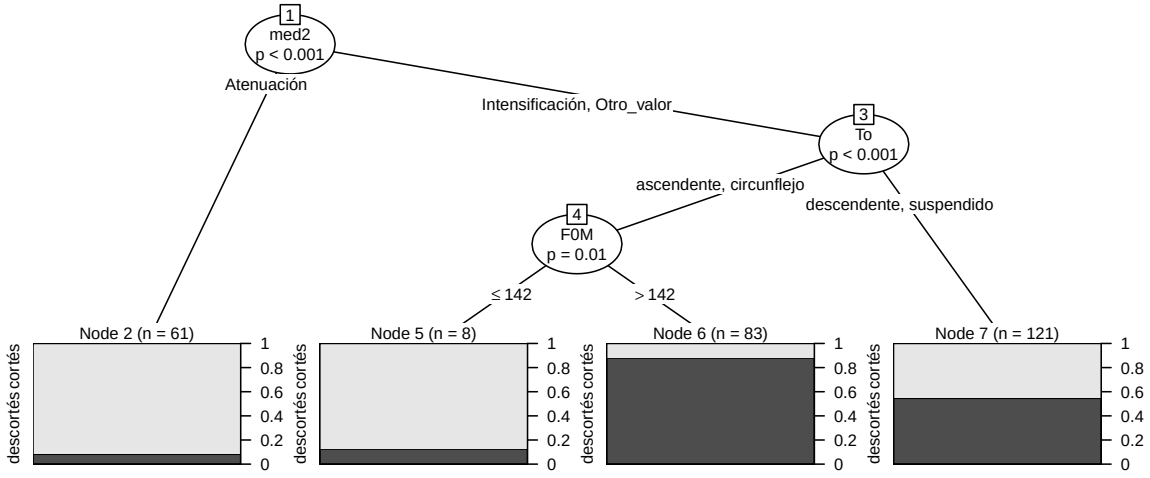


Figure 11: Árbol de decisiones de la variable Cortes_Descortes

Resultados del árbol de decisiones

- El nodo inicial se basa en el medio de expresión (med2); cuando el enunciado presenta atenuación, se clasifica como cortés.
- Si se utiliza intensificación u otros valores, se analiza el tonema (To).
  - Un tonema ascendente y circunflejo se clasifica aún más como descortés si la media de F0 es alta.

- Un tonema descendente y suspendido se clasifica tanto como cortés como descortés.

15.5.4 Random Forest

La técnica de Random Forest genera múltiples árboles de decisiones para determinar con qué variables se clasifican mejor los datos de una variable independiente, ya sea numérica o categórica. Esta técnica ayuda a reducir las variables explicativas para optimizar la explicación de la variable de entrada.

Para ejemplificar el uso de Random Forest, utilizaremos la base de datos Fonocortesía, tomando como variable independiente la variable **Cortes_Descortes**. Utilizaremos todas las variables explicativas que no tengan más de 20 niveles y las variables numéricas más relevantes. Podemos usar el comando `str` para conocer la tipología de las variables y la cantidad de factores.

Aquí está el código en R para realizar un análisis con Random Forest:

```
corpusren$cort <- gsub("cortés","cor",corpusren$cort)
corpusren$cort <- gsub("descor","des",corpusren$cort)
corpusren$cort <- as.factor(corpusren$cort)
cortesia_cforest<- randomForest(cort~Llam+med+
FOM+IM+Sil+dur+DPA+DPP+CP+Cur+ILI+To+Vmod, data=corpusren%>%
  mutate_if(is.character,as.factor), na.action = na.roughfix)
print(cortesia_cforest)
```

```
Call:
 randomForest(formula = cort ~ Llam + med + FOM + IM + Sil + dur +      DPA + DPP + CP + Cur
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 18%
Confusion matrix:
    cor des class.error
cor 101  30        0.23
des  20 125        0.14
```

Inicialmente y sin más información, *RandomForest* genera un grupo de 500 árboles simulados, en los que cambia cada vez un dato de algunas de las variables empleadas. Porsteriormente, en el resultado principal de la prueba, observamos un porcentaje de clasificación adecuada de los grupos de la variable independiente. En general, hay un porcentaje de error del 19 %;

los casos descorteses se clasifican bien en un 85 % (20 casos se han incluido como corteses), mientras que los de cortesía se clasifican algo peor, con un 76 % (32 casos se han considerado descorteses). Por tanto, la clasificación es en cierto modo interesante.

Podemos acceder a la contribución de cada variable en el proceso de clasificación. Se utiliza en este caso el código *importance*:

```
importance(cortesia_cforest)
```

```
     MeanDecreaseGini
Llam             12.9
med              29.3
FOM              16.5
IM               13.7
Sil               8.1
dur               9.5
DPA               5.3
DPP               5.2
CP                5.7
Cur               8.4
ILI               5.3
To                6.4
Vmod             10.2
```

Además, *RandomForest* permite visualizar las contribuciones de las variables en forma de gráfico, como el que se incluye a continuación:

```
varImpPlot(cortesia_cforest)
```

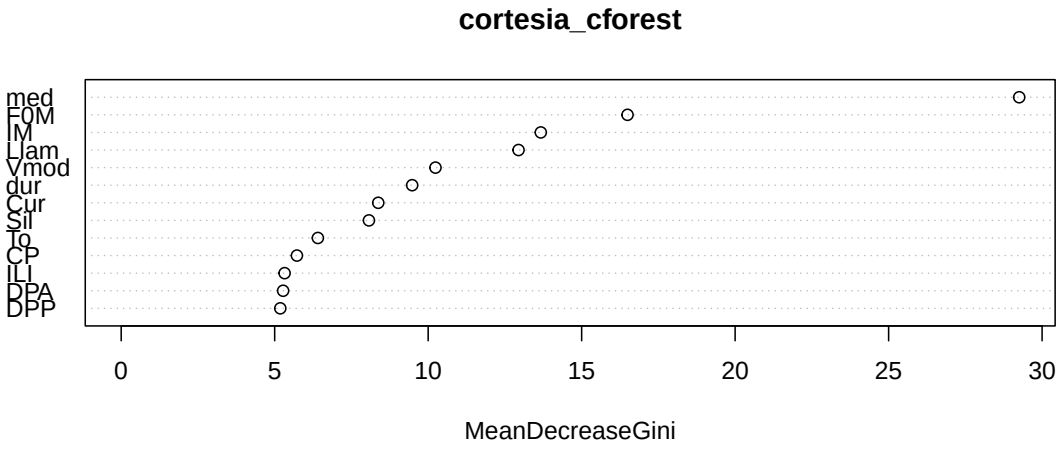


Figure 12: Importancia de las variables en Random Forest

En el gráfico anterior, se observa que hay una variable mucho más importante que el resto; se trata de *med*, que es el medio de expresión de la cortesía o descortesía. En general, como hemos visto en secciones anteriores, se trata generalmente de valores como *atenuación* o *intensificación*. Después, las otras cuatros variables que puntúan de manera alta serían la F0 y la intensidad medias, la variable *llama la atención* y la variable *Valor modal*.

RandomForest es complejo de interpretar en muchas ocasiones porque no produce un gráfico de clasificación como sucede con la prueba del árbol de decisiones tradicional. No obstante, podemos acceder a los valores de la base de datos con la predicción sugerido por el análisis de la prueba. En los siguientes ejemplos, vemos una muestra de los primeros ocho casos de la base de datos y la predicción que se ha generado basada en el resultado de la prueba:

```
head(cortesia_cforest$predicted, n= 8)
```

```
  1   2   3   4   5   6   7   8
des cor des des des des cor des
Levels: cor des
```

Podemos acceder a la puntuación que ha recibido cada uno de esos registros. De esta manera, podemos comparar las predicciones realizadas con el valor real.

```
head(margin(cortesia_cforest), n=8)
```

```
  des   des   des   des   des   cor   cor   des
 0.52 -0.64  0.71  0.84  0.83 -0.70  0.65  0.92
```

Podemos observar que los registros que no se han clasificado bien porque han puntuado de manera negativa han sido el 2 y el 6. El resto se ha clasificaco adecuadamente, aunque con porcentajes de seguridad medio altos.

15.5.5 Ejercicios

15.5.5.1 Enunciados

1. En la base de datos *Idiolectal*, realiza un análisis de Random Forest con la variable *genre* como variable independiente y las variables *tonemes*, *tonemeMAS*, *dur*, *spk*, *spk2*, *MAStag* y *circunflejo* como variables explicativas.
2. En la base de datos *Idiolectal*, realiza un gráfico con los resultados obtenidos.
3. En la base de datos *Idiolectal*, realiza un árbol de decisiones con la variable *genre* como variable independiente y las variables *tonemes*, *tonemeMAS*, *dur*, *spk*, *MAStag* y *circunflejo* como variables explicativas. Excluye a "pzorro" de la variable *spk*.
4. Si eliminamos *genre* y lo cambiamos por *spk2*, ¿cómo cambian los resultados?
5. En la base de datos *Idiolectal*, realiza un árbol de decisiones con la variable *spk2* como variable independiente y las variables *tonemes*, *tonemeMAS*, *dur*, *MAStag* y *circunflejo* como variables explicativas.

15.5.5.2 Soluciones

! Práctica

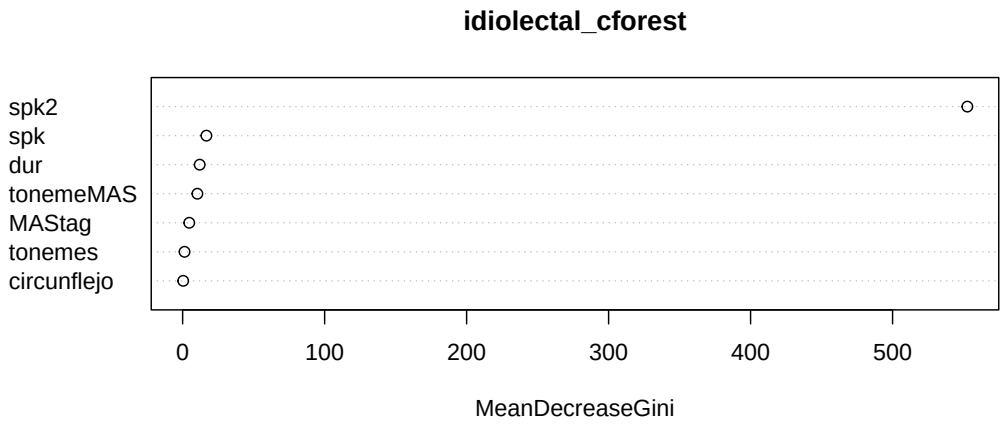
1. En la base de datos *Idiolectal*, realiza un análisis de Random Forest con la variable *genre* como variable independiente y las variables *tonemes*, *tonemeMAS*, *dur*, *spk*, *spk2*, *MAStag* y *circunflejo* como variables explicativas.

```r
library(randomForest)
idiolectal <- idiolectal%>%select(genre,tonemes,
tonemeMAS,dur,spk,spk2,MAStag,circunflejo)%>%
  mutate_if(is.character,as.factor)

idiolectal_cforest<-
randomForest(genre~tonemes+tonemeMAS+dur+spk+spk2+MAStag+circunflejo,

             data=idiolectal, na.action = na.roughfix)

print(idiolectal_cforest)
```

```
Call:
 randomForest(formula = genre ~ tonemes + tonemeMAS + dur + spk +      spk2 + MAStag + circ
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 0%
Confusion matrix:
     5p pod class.error
5p  609   0           0
pod   0 609           0
```

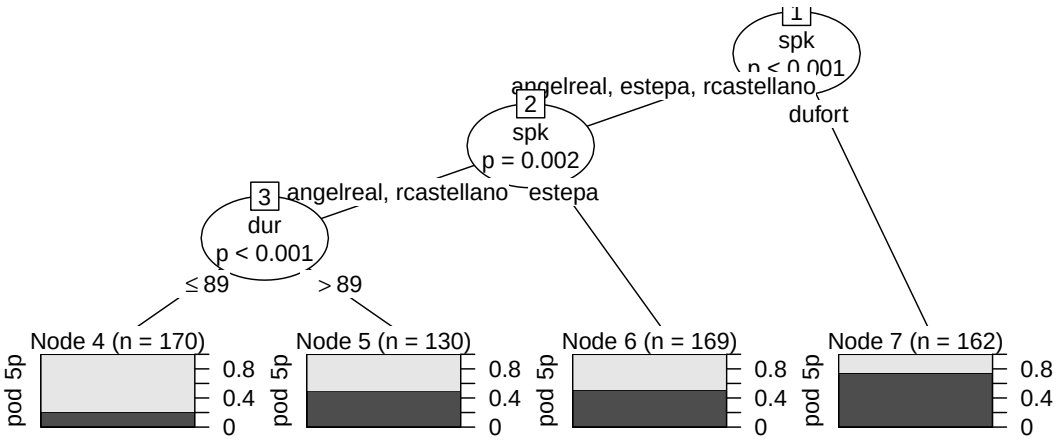
2. En la base de datos *Idiolectal*, realiza un gráfico con los resultados obtenidos.

```r
varImpPlot(idiolectal_cforest)
```



3. Realiza un árbol de decisiones con la variable *genre* como variable independiente y las variables *tonemes*, *tonemeMAS*, *dur*, *spk*, *MAStag* y *circunflejo* como variables explicativas. Excluye a "pzorro" de la variable *spk*.

```
library(partykit)
idiolectal_tree<- partykit::ctree(genre~tonemes+
tonemeMAS+dur+spk+MAStag+circunflejo,
        data= idiolectal%>%filter(spk!="pzorro")%>%select(genre,tonemes,tonemeMAS,dur,sp

plot(idiolectal_tree, ep_args = list(justmin = 15))
```



4. Si eliminamos *genre* y lo cambiamos por *spk2*, ¿cómo cambian los resultados?

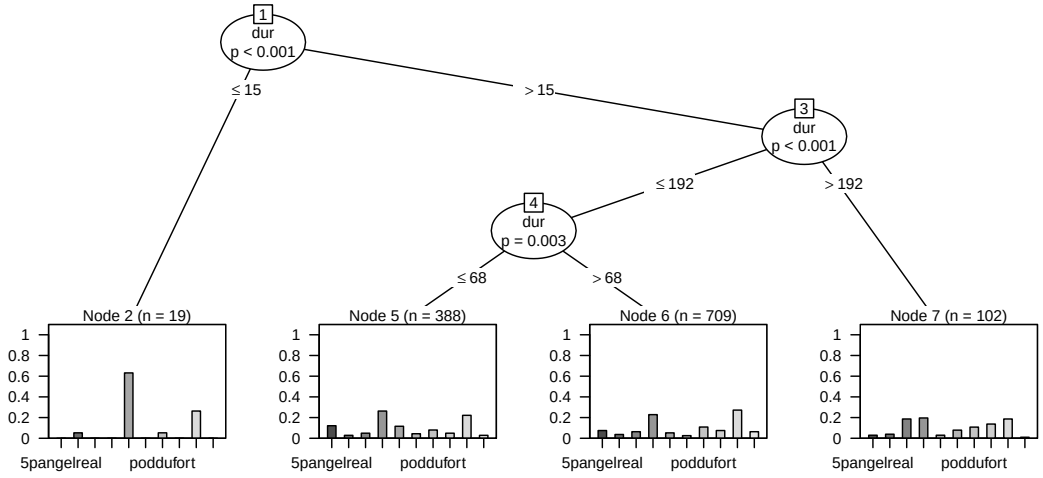
```
idiolectal_cforest2<-
  randomForest(spk2~tonemes+tonemeMAS+dur+MAStag+circunflejo,
               data=idiolectal, na.action = na.roughfix)

print(idiolectal_cforest2)
```

```
Call:
 randomForest(formula = spk2 ~ tonemes + tonemeMAS + dur + MAStag +      circunflejo, data
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 75%
Confusion matrix:
               5pangelreal 5pdufort 5pestepa 5ppzorro 5prcastellano
5pangelreal              0        0        1       51             1
5pdufort                 0        0        2       25             0
5pestepa                 1        0        2       28             0
5ppzorro                 1        2        5      100             3
5prcastellano            0        0        0       41            12
podangelreal             0        0        1       16             0
podd ufort               0        0        2       53             2
podestepa                0        0        4       30             2
podpzorro                0        0        4      113             4
podrcastellano           0        0        0       19             0
               podangelreal podd ufort podestepa podpzorro podrcastellano
5pangelreal               1          3         1        45              0
5pdufort                  0          1         2        12              0
5pestepa                  0          5         3        44              0
5ppzorro                  2          9         2       160              0
5prcastellano             1          4         2        37              0
podangelreal              2          1         0        23              0
podd ufort                1          8         2        52              0
podestepa                 1          4         4        41              0
podpzorro                 1          0         3       178              0
podrcastellano            0          1         0        37              0
               class.error
5pangelreal           1.00
5pdufort              1.00
```

```
5pestepa              0.98
5ppzorro              0.65
5prcastellano         0.88
podangelreal          0.95
podducfort            0.93
podestepa             0.95
podpzorro             0.41
podrcastellano        1.00
```

5. Realiza un árbol de decisiones con la variable *spk2* como variable independiente y las variables *tonemes*, *tonemeMAS*, *dur*, *MAStag* y *circunflejo* como variables explicativas.

```
idiolectal_tree2<- partykit::ctree(spk2~tonemes+tonemeMAS+dur+
                                     MAStag+circunflejo,
        data= idiolectal, control = ctree_control(maxdepth = 3))

plot(idiolectal_tree2, ep_args = list(justmin = 15),
     gp = gpar(fontsize = 8))
```



16 Ejercicios finales sobre el curso (usando la base de datos Idiolectal)

16.1 Enunciados

1. Renombra la variable *ip* y llámala *intonational_phrase*.
2. Crea una variable llamada id2 en la que se recoja el número de fila para cada grupo de la variable *filename*.
3. Crea un dataframe llamado *idiolectal_pod* que recoja solo las variables *spk*, *phon*, *tonemes*, *words* de la categoría "pod" en *genre*. Este dataframe no debe tener valores vacíos en ninguna variable.
4. Crea un diagrama de caja con la variable *dur* según las categorías de la variable *genre*.
5. Construye un gráfico de líneas para el filename "poddufort" en el que *tmin* sea la línea de tiempo, *tonemeMAS* sea el eje de las ordenadas y *spk* sea el color de las líneas.
6. Haz un mapa de calor de los hablantes en el *genre* "pod" con todas las variables numéricas que no sean tmin o tmax.
7. Realiza un análisis de correspondencias múltiples con las variables *tonemes*, *spk2*,*phon*, *MAStag* y *circunflejo*. Visualiza un mapa con las categorías más relacionadas.
8. Computa una variable que recoja la velocidad de habla. Recomendación: fíjate en la variable *dur* y en la variable *words*.
9. Realiza un Wordcloud con la variable word, pero usa solo las 50 palabras más frecuentes.
10. Transforma la variable *genre* en una variable de factor y realiza un análisis randomForest con las variables *tonemes*, *tonemeMAS*, *dur*, *spk*, *MAStag* y *circunflejo* (las variables de carácter también debes transformarlas a factor). La variable independiente será *genre*.

16.2 Soluciones

- Renombra la variable *ip* y llámala *intonational_phrase*.

```
library(tidyverse)
library(readxl)
idiolectal <- read_xlsx("databases/idiolectal.xlsx",sheet = 1)
idiolectal <- idiolectal%>%dplyr::rename(intonational_phrase = ip)
```

- Crea una variable llamada id2 en la que se recoja el número de fila para cada grupo de la variable *filename*.

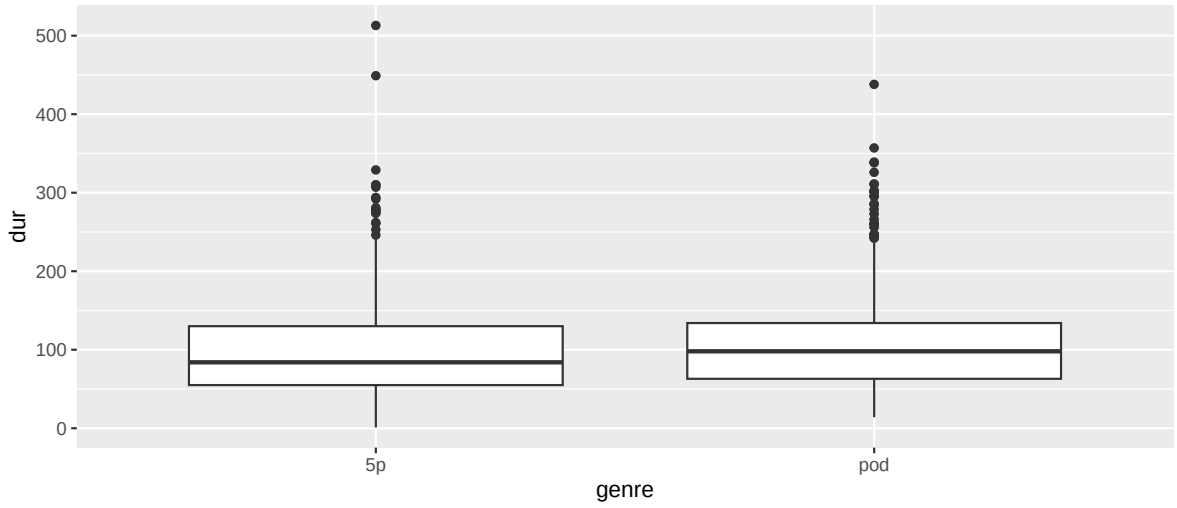
```
idiolectal <- idiolectal%>%group_by(filename)%>%
  mutate(id2 = row_number())
```

- Crea un dataframe llamado *idiolectal_pod* que recoja solo las variables *spk*, *phon*, *tonemes*, *words* de la categoría "pod" en *genre*. Este dataframe no debe tener valores vacíos en ninguna variable.

```
idiolectal_pod <- idiolectal%>%filter(genre == "pod")%>%
  select(spk,phon,tonemes,words)%>%
  na.omit()
```

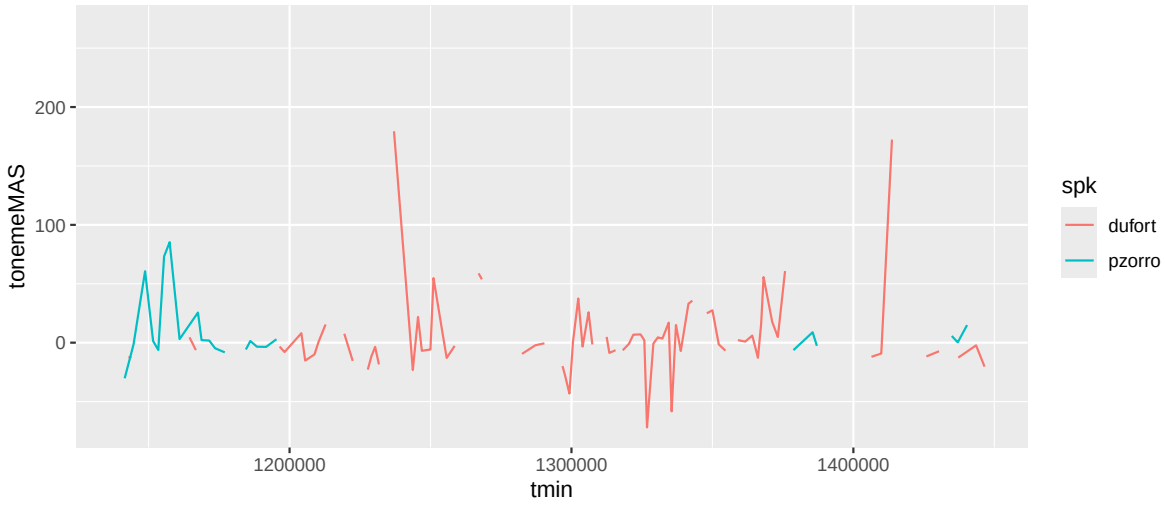
- Crea un diagrama de caja con la variable *dur* según las categorías de la variable *genre*.

```
library(ggplot2)
ggplot(idiolectal, aes(x = genre, y = dur)) +
  geom_boxplot()
```



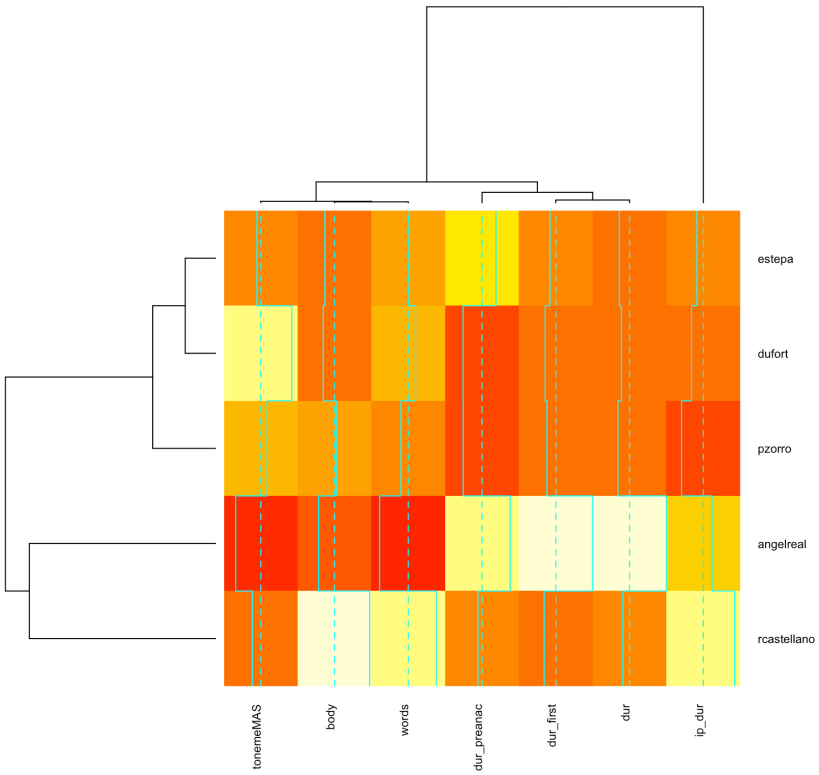
- Construye un gráfico de líneas para el filename "poddufort" en el que *tmin* sea la línea de tiempo, *tonemeMAS* sea el eje de las ordenadas y *spk* sea el color de las líneas.

```
idiolectal%>%filter(filename == "poddufort")%>%
  ggplot(aes(x = tmin, y = tonemeMAS, color = spk)) +
  geom_line()
```



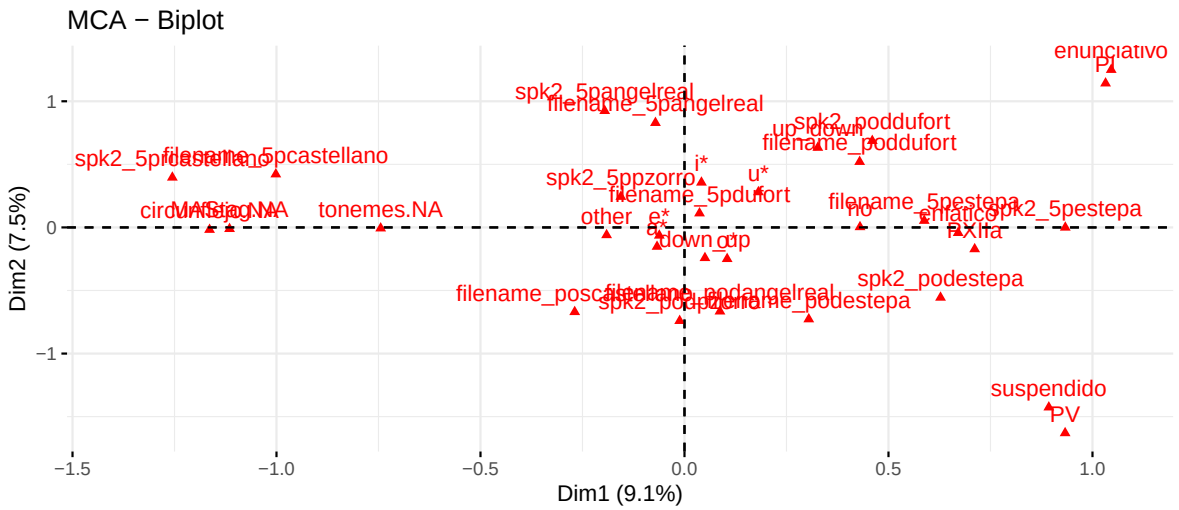
- Haz un mapa de calor de los hablantes en el *genre* "pod" con todas las variables numéricas que no sean tmin o tmax.

```
library(gplots)
idiolectal_pod2 <- idiolectal%>%filter(genre == "pod")%>%
  select(-tmin,-tmax)%>%na.omit()%>%
  group_by(spk)%>%
  summarise_all(mean,na.rm=T)%>%column_to_rownames(var="spk")%>%
  select(ip_dur,dur,words,dur_first,dur_preanac,tonemeMAS,body)

heatmap.2(as.matrix(idiolectal_pod2), scale = "column",
          cexRow = 0.8, cexCol = 0.8)
```



- Realiza un análisis de correspondencias múltiples con las variables *tonemes*, *spk2*,*phon*, *MAStag* y *circunflejo*. Visualiza un mapa con las categorías más relacionadas.

```
library(FactoMineR)
library(factoextra)

idiolectal_mca <- MCA(idiolectal%>%select(tonemes,spk2,
                      phon,MAStag,circunflejo),
                      graph =FALSE,level.ventil = 0.05)

fviz_mca_biplot(idiolectal_mca, invisible=c("ind"))
```

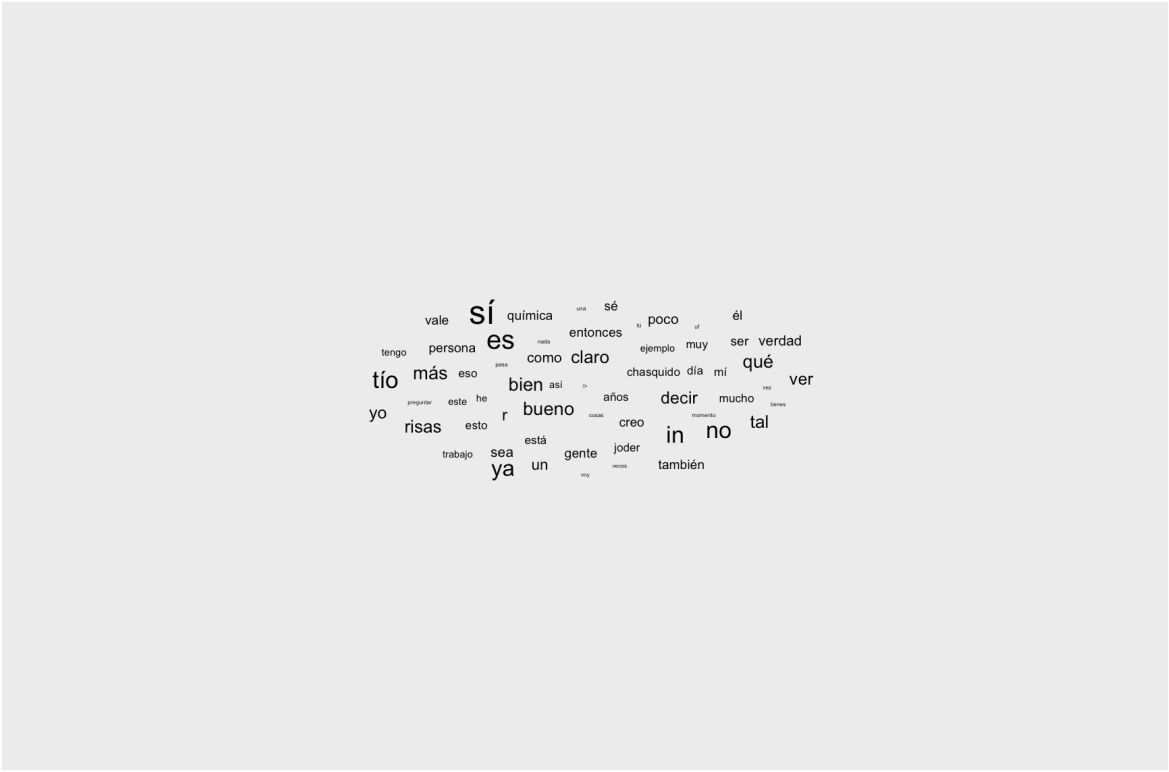


- Computa una variable que recoja la velocidad de habla. Recomendación: fíjate en la variable *dur* y en la variable *words*.

```
idiolectal <- idiolectal%>%mutate(velocidad = words/dur)
```

- Realiza un Wordcloud con la variable word, pero usa solo las 50 palabras más frecuentes.

```
library(ggwordcloud)
library(dplyr)

idiolectal%>%count(word)%>%top_n(50)%>%
  ggplot(aes(label = word, size = n)) +
  geom_text_wordcloud()
```



- Transforma la variable *genre* en una variable de factor y realiza un análisis randomForest con las variables *tonemes*, *tonemeMAS*, *dur*, *spk*, *MAStag* y *circunflejo* (las variables de carácter también debes transformarlas a factor). La variable independiente será *genre*.

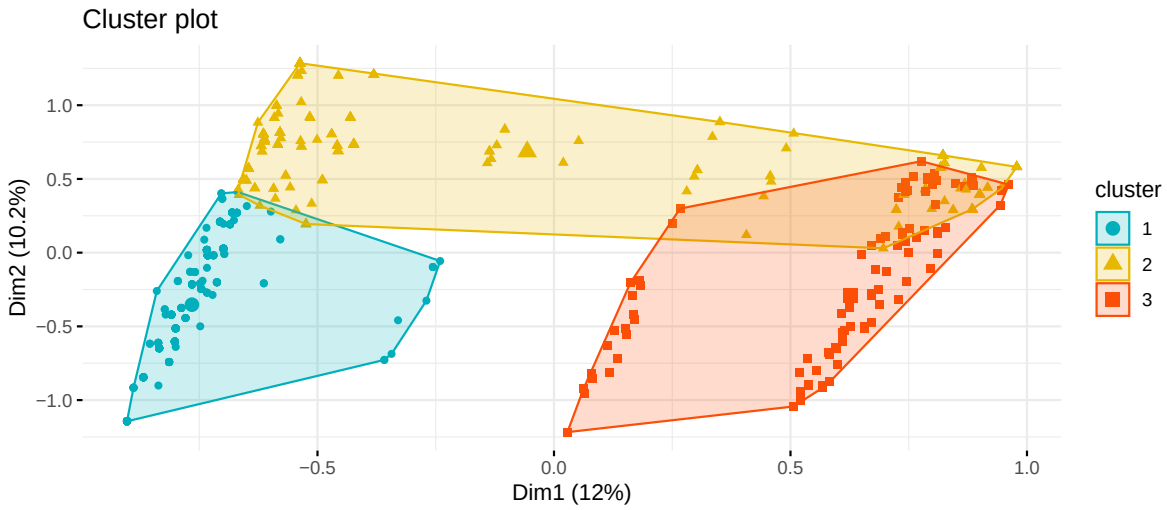
```
idiolectal2 <- idiolectal%>%
  mutate(genre = as.factor(genre),tonemes = as.factor(tonemes),
         spk = as.factor(spk),MAStag = as.factor(MAStag),
         circunflejo = as.factor(circunflejo))
idiolectal_cforest3<-
  randomForest(genre~tonemes+tonemeMAS+dur+spk+MAStag+circunflejo,
               data=idiolectal2, na.action = na.roughfix)

print(idiolectal_cforest3)
```

```
Call:
 randomForest(formula = genre ~ tonemes + tonemeMAS + dur + spk +      MAStag + circunflejo,
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2
```

```
        OOB estimate of  error rate: 42%
Confusion matrix:
     5p pod class.error
5p  339 270        0.44
pod 239 370        0.39
```

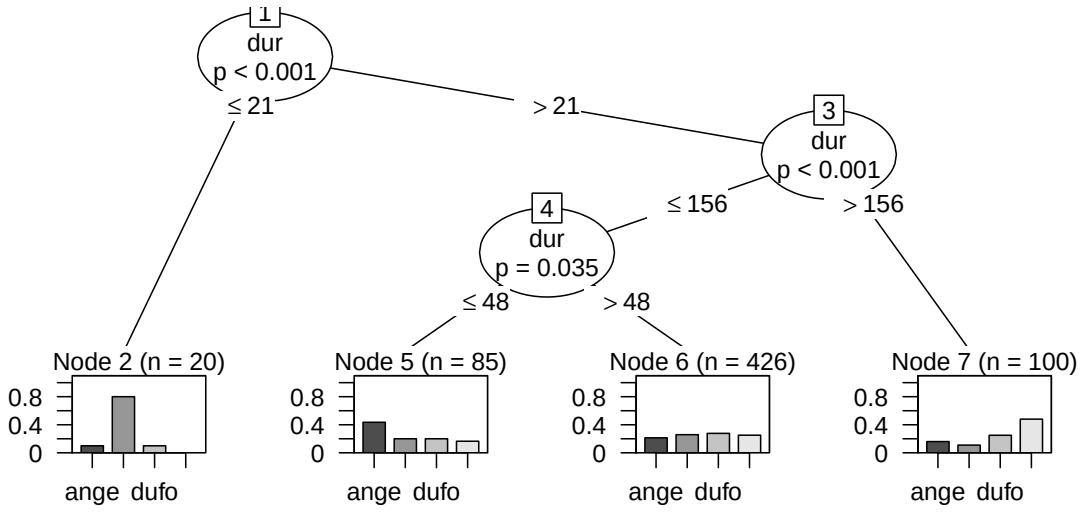
- Realiza un análisis de clúster con las variables *genre*, *tonemes*, *spk2*, *MAStag* y *circunflejo*.

```
library(FactoMineR)
library(factoextra)

idiolectal_mca <- MCA(idiolectal%>%select(genre,tonemes,spk2,
                                          MAStag,circunflejo),
                      graph =FALSE,level.ventil = 0.05)

idiolectal_cluster <- HCPC(idiolectal_mca, nb.clust = 3, graph = FALSE)

fviz_cluster(idiolectal_cluster, geom = "point", palette =
               c("#00AFBB", "#E7B800", "#FC4E07"),
             ggtheme = theme_minimal())
```



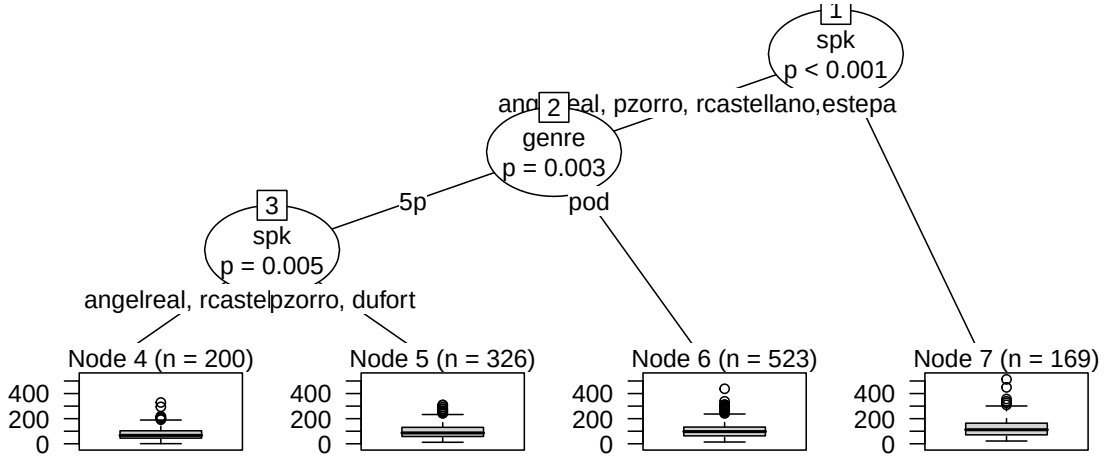
- Haz un árbol de decisiones con la variable *spk* como variable independiente y las variables *tonemes*, *tonemeMAS*, *dur*, *MAStag* y *circunflejo* como variables explicativas. Debes filtrar previamente la categoría "pzorro" de la variable *spk*.

```
library(stringr)
idiolectal3 <- idiolectal%>%filter(spk != "pzorro")

idiolectal_tree3<- partykit::ctree(spk~tonemes+tonemeMAS+dur
+MAStag+circunflejo,
data= idiolectal3%>%mutate(spk = str_trunc(spk, 4,ellipsis = ""))%>%
  mutate_if(is.character,as.factor),
control = ctree_control(maxdepth = 3))

plot(idiolectal_tree3)
```



- Haz un árbol de decisiones con la variable *dur* como variable independiente y las variables *tonemes*, *tonemeMAS*, *genre,spk*, *MAStag* y *circunflejo* como variables explicativas.

```
idiolectal_tree4 <- partykit::ctree(dur~tonemeMAS+tonemes
+genre+spk+circunflejo,
data= idiolectal%>%mutate_if(is.character,as.factor),
control = ctree_control(maxdepth = 3))

plot(idiolectal_tree4)
```



- Haz una prueba ANOVA para la variable *spk* como independiente y la variable *dur* como dependiente.

```
Tuk <- TukeyHSD(aov(dur~spk, data = idiolectal%>%
mutate(spk = as.factor(spk))))

Tuk
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = dur ~ spk, data = idiolectal %>% mutate(spk = as.factor(spk)))

$spk
                        diff    lwr  upr p adj
pzorro-angelreal         6.1  -9.32 21.5  0.82
rcastellano-angelreal   -6.7 -25.95 12.6  0.88
dufort-angelreal        12.5  -6.58 31.5  0.38
estepa-angelreal        36.1  17.28 55.0  0.00
rcastellano-pzorro     -12.8 -27.88  2.3  0.14
dufort-pzorro            6.4  -8.46 21.2  0.77
estepa-pzorro           30.0  15.47 44.6  0.00
dufort-rcastellano      19.1   0.35 37.9  0.04
estepa-rcastellano      42.8  24.22 61.4  0.00
estepa-dufort           23.7   5.33 42.0  0.00
```

- Realiza una prueba chi cuadrado (bondad de ajuste) con la hipótesis de igualdad de proporciones en la variable *tonemes*.

```
table <- table(idiolectal%>%select(tonemes))
bondad <- chisq.test(table, p = c(1/4,1/4,1/4,1/4))
bondad$residuals
```

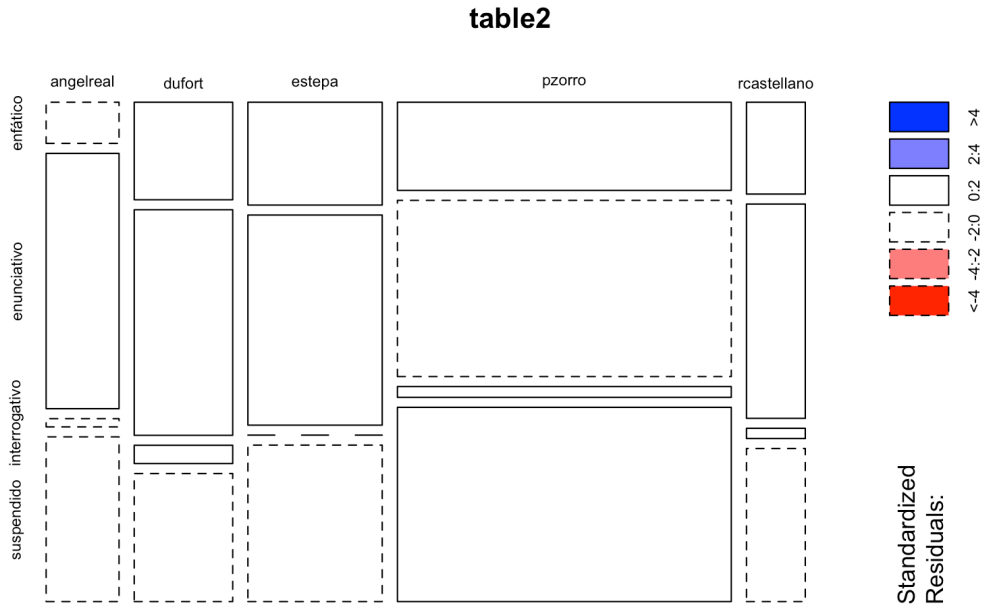
```
              tonemes
filename       enfático enunciativo interrogativo suspendido
  5pangelreal    -0.537       1.209        -0.416     -0.829
  5pcastellano    0.850       0.500        -0.103     -1.127
  5pdufort        1.004      -0.305        -1.118     -0.126
  5pestepa       -0.222       0.051        -1.221      0.391
  podangelreal   -0.896      -0.078        -1.196      1.007
  podd ufort     -0.270       0.151         2.838     -0.638
  podestepa       1.885      -0.835        -1.262     -0.148
  poscastellano  -1.958      -0.736         2.747      1.553
```

- Realiza una prueba chi cuadrado que observe a relación entre *spk* y *tonemes*. Analiza los residuos estandarizados en un mosaicplot.

```
table2 <- table(idiolectal$spk,idiolectal$tonemes)

bondad2 <- chisq.test(table2)


bondad2$residuals
```

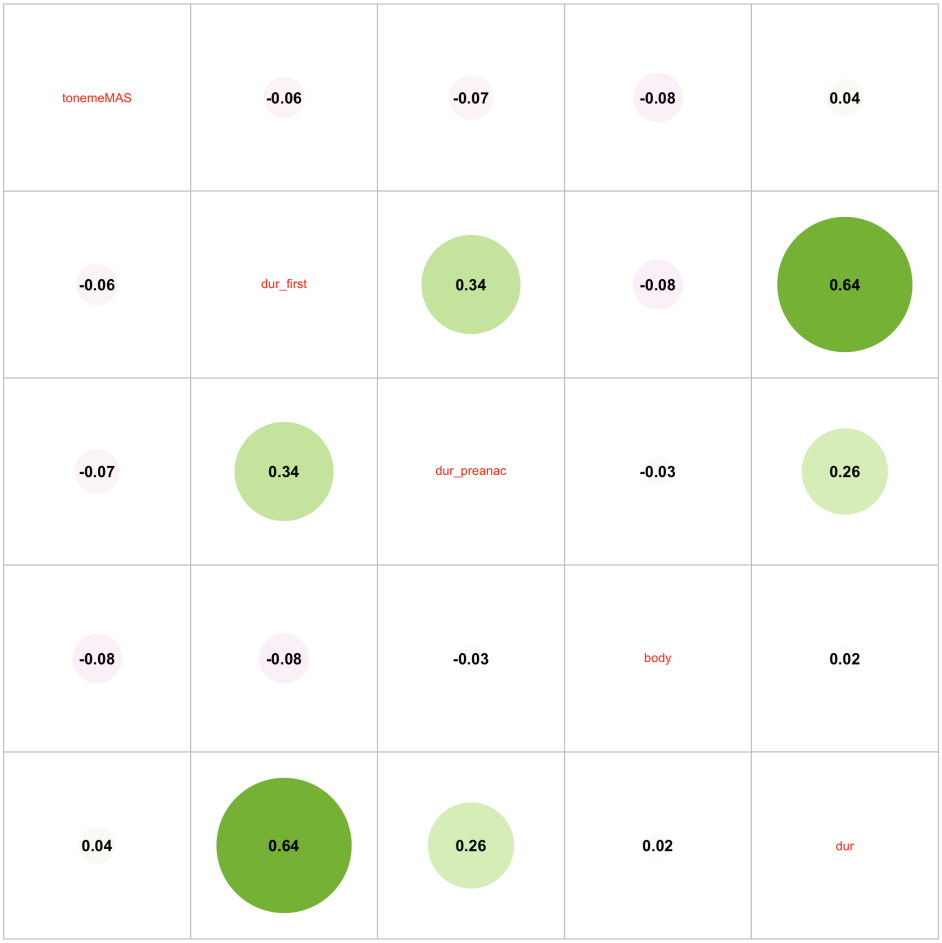
```
            enfático enunciativo interrogativo suspendido
  angelreal   -1.731       1.330        -0.138     -0.170
  dufort       0.426       0.696         1.163     -1.333
  estepa       0.764       0.298        -1.454     -0.528
  pzorro       0.035      -1.310         0.323      1.320
  rcastellano  0.139       0.290         0.076     -0.431
```

```
mosaicplot(table2, shade = TRUE)
```



- Realiza una prueba de correlación de Pearson entre las variables *tonemesMAS*, *dur_first*,*dur_preanac*,*body* y *dur* y visualízalo en un gráfico.

```
correlacion <- cor(idiolectal%>%select(tonemeMAS,dur_first,
    dur_preanac,body,dur)%>%na.omit(), method = "pearson")

library(corrplot)

corrplot(correlacion,addCoef.col = 'black',cl.pos = "n",
         tl.pos = 'd', col = COL2('PiYG'),tl.cex = 0.8)
```



17 Referencias

17.1 Librerías utilizadas

1. library(tidyverse)
2. library(readxl)
3. library(gridExtra)
4. library(tidyverse)
5. library(corrplot)

6. library(FactoMineR)
7. library(factoextra)
8. library(partykit)
9. library(randomForest)
10. library(gplots)
11. library(ggwordcloud)

17.2 Bibliografía

- https://bookdown.org/content/2031/
- http://factominer.free.fr/
- François Husson, canal de *Youtube*: https://www.youtube.com/channel/UCyz4M1pwJBNfjMFaUCHCNU
- Breiman, L., Cutler, A., Liaw, A., & Wiener, M. (2022). *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/Rand
- Cui, B. (2024). *DataExplorer: Automate Data Exploration and Treatment*. http://boxuancui.github.io/DataExplorer/
- Gohel, D., & Skintzos, P. (2024). *flextable: Functions for Tabular Reporting*. https://ardata-fr.github.io/flextable-book/
- Greenacre, M.J.. (2007). Correspondence analysis in practice. Chapman & Hall/CRC.
- Greenacre, M.J. and Blasius, J. (2006). Multiple correspondence analysis and related methods. Chapman & Hall/CRC.
- Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651-674. https://doi.org/10.1198/106186006X133933
- Hothorn, T., & Zeileis, A. (2015). partykit: A Modular Toolkit for Recursive Partytioning in R. *Journal of Machine Learning Research*, *16*, 3905-3909.
- Hothorn, T., & Zeileis, A. (2023). *partykit: A Toolkit for Recursive Partytioning*. http://partykit.r-forge.r-project.org/partykit/
- Husson, F., Josse, J., Le, S., & Mazet, J. (2024). *FactoMineR: Multivariate Exploratory Data Analysis and Data Mining*. http://factominer.free.fr
- Husson F., Lê S., Pagès J. (2017). Exploratory Multivariate Analysis by Example Using R. 2nd edition. Chapman & Hall/CRC.
- Kassambara, A., & Mundt, F. (2020). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. http://www.sthda.com/english/rpkgs/factoextra

- Lê, S., Josse, J., & Husson, F. (2008). FactoMineR: A Package for Multivariate Analysis. *Journal of Statistical Software*, *25*(1), 1-18. https://doi.org/10.18637/jss.v025.i01
- Levinson, S.C., Torreira, F., 2015. Timing in turn-taking and its implications for processing models of language. Front. Psychol. 6. https://doi.org/10.3389/fpsyg.2015.00731
- Levshina, N., 2015. How to do Linguistics with R: Data exploration and statistical analysis. John Benjamins Publishing Company, Amsterdam / Philadelphia.
- Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News*, *2*(3), 18-22.
- Pennec, E. L., & Slowikowski, K. (2023). *ggwordcloud: A Word Cloud Geom for ggplot2*. https://github.com/lepennec/ggwordcloud
- R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
- Revelle, W. (2024). *psych: Procedures for Psychological, Psychometric, and Personality Research*. https://personality-project.org/r/psych/ https://personality-project.org/r/psych-manual.pdf
- Warnes, G. R., Bolker, B., Bonebakker, L., Gentleman, R., Huber, W., Liaw, A., Lumley, T., Maechler, M., Magnusson, A., Moeller, S., Schwartz, M., & Venables, B. (2024). *gplots: Various R Programming Tools for Plotting Data*. https://github.com/talgalili/gplots
- Wei, T., & Simko, V. (2021a). *corrplot: Visualization of a Correlation Matrix*. https://github.com/taiyun/corrplot
- Wei, T., & Simko, V. (2021b). *R package «corrplot»: Visualization of a Correlation Matrix*. https://github.com/taiyun/corrplot
- Wickham, H. (2023). *tidyverse: Easily Install and Load the Tidyverse*. https://tidyverse.tidyverse.org
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686
- Wickham, H., & Bryan, J. (2023). *readxl: Read Excel Files*. https://readxl.tidyverse.org
- Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-Based Recursive Partitioning. *Journal of Computational and Graphical Statistics*, *17*(2), 492-514. https://doi.org/10.1198/106186008X319331
- Zhu, H. (2024). *kableExtra: Construct Complex Table with kable and Pipe Syntax*. http://haozhu233.github.io/kableExtra/