

# Molina\_Del\_Moral\_PRA2

Autor: Rubén Molina Del Moral

2024-12-26

Para empezar se procede a la carga de los datos:

## 1º Carga de los datos de la plataforma ‘dades obertes de Catalunya’

```
# Se cargan los datos seleccionando el archivo .csv correspondiente y con la codificación correcta
ruta <- file.path("C:/Users/ruben/Desktop/UOC/ASIGNATURAS OBLIGATORIAS/M2.859 - Visualización de datos/Instal_lacions_d_autoconsum_el_ctric_20241214.csv")
datos <- read.csv(ruta, sep = ";", header = TRUE, fileEncoding = "UTF-8")
```

Acto y seguido, se procede al tratamiento de los datos para trabajar con datos que se puedan interpretar y transformar los registros con potencias anómalas:

## 2º Tratamiento de los datos

Se transforman los registros con el formato xxx.xxx.xxx.xxx.xxx. a xxx sin decimales

```
# Carga la libreria de tidyverse
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.2      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.2      v tibble    3.2.1
## v lubridate  1.9.2      v tidyr     1.3.0
## v purrr      1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Reemplaza el formato con múltiples puntos (que marcan decimales) por el formato deseado
datos$Potència <- gsub("\\\\.*$", "", datos$Potència)
```

Se añade una nueva variable como ‘Provincia’ para saber en que provincias se ubican las instalaciones.

```

# Se carga la librería de dplyr
library(dplyr)
# Se añade una nueva variable en el conjunto de datos con el nombre de 'Provincias'
datos_1 <- datos %>%
  mutate(Provincia = case_when(
    Comarca == "Anoia" ~ "Barcelona",
    Comarca == "Alt Penedès" ~ "Barcelona",
    Comarca == "Alt penedès" ~ "Barcelona",
    Comarca == "Baix Llobregat" ~ "Barcelona",
    Comarca == "Baix llobregat" ~ "Barcelona",
    Comarca == "bages" ~ "Barcelona",
    Comarca == "Bages" ~ "Barcelona",
    Comarca == "BAGES" ~ "Barcelona",
    Comarca == "Berguedà" ~ "Barcelona",
    Comarca == "Barcelonès" ~ "Barcelona",
    Comarca == "Baix Llobregat" ~ "Barcelona",
    Comarca == "Baix llobregat" ~ "Barcelona",
    Comarca == "Baix penedès" ~ "Barcelona",
    Comarca == "Garraf" ~ "Barcelona",
    Comarca == "Lluçanes" ~ "Barcelona",
    Comarca == "Maresme" ~ "Barcelona",
    Comarca == "Moianès" ~ "Barcelona",
    Comarca == "Osona" ~ "Barcelona",
    Comarca == "Vallès Occidental" ~ "Barcelona",
    Comarca == "Vallès Oriental" ~ "Barcelona",
    Comarca == "Alt Empordà" ~ "Girona",
    Comarca == "Baix Empordà" ~ "Girona",
    Comarca == "Garrotxa" ~ "Girona",
    Comarca == "Gironès" ~ "Girona",
    Comarca == "Pla de l'Estany" ~ "Girona",
    Comarca == "Ripollès" ~ "Girona",
    Comarca == "Selva" ~ "Girona",
    Comarca == "Alta Ribagorça" ~ "Lleida",
    Comarca == "Aran" ~ "Lleida",
    Comarca == "Cerdanya" ~ "Lleida",
    Comarca == "Garrigues" ~ "Lleida",
    Comarca == "Noguera" ~ "Lleida",
    Comarca == "Segarra" ~ "Lleida",
    Comarca == "Segrià" ~ "Lleida",
    Comarca == "SEGRÍÀ" ~ "Lleida",
    Comarca == "Alt Urgell" ~ "Lleida",
    Comarca == "Urgell" ~ "Lleida",
    Comarca == "Pla d'Urgell" ~ "Lleida",
    Comarca == "Pallars Jussà" ~ "Lleida",
    Comarca == "Pallars Sobirà" ~ "Lleida",
    Comarca == "Solsonès" ~ "Lleida",
    Comarca == "Alt Camp" ~ "Tarragona",
    Comarca == "Baix Camp" ~ "Tarragona",
    Comarca == "Baix Ebre" ~ "Tarragona",
    Comarca == "Baix Penedès" ~ "Tarragona",
    Comarca == "Conca de Barberà" ~ "Tarragona",
    Comarca == "Montsià" ~ "Tarragona",
    Comarca == "Ribera d'Ebre" ~ "Tarragona",
  ))

```

```

Comarca == "Priorat" ~ "Tarragona",
Comarca == "Tarragonès" ~ "Tarragona",
Comarca == "Terra Alta" ~ "Tarragona",
TRUE ~ NA_character_ # Mantiene NA en otras filas
))

```

Se calculan las frecuencias y proporciones de las instalaciones de autoconsumo eléctrico por provincias

```

# Calcula las frecuencias por provincias
frecuencias_provincias <- table(datos_1$Provincia)
# Muestra las frecuencias de las provincias deseadas
provincias_seleccionadas <- c("Barcelona", "Girona", "Lleida", "Tarragona")
frecuencias_filtradas <- frecuencias_provincias[provincias_seleccionadas]
# Muestra el resultado
print(frecuencias_filtradas)

```

```

##
## Barcelona      Girona      Lleida Tarragona
##      66666      18836      9615      17409

```

```

# Calcula las proporciones y se multiplica por 100
proporciones_provincias <- prop.table(frecuencias_provincias) * 100
# Filtra solo las provincias deseadas
provincias_seleccionadas <- c("Barcelona", "Girona", "Lleida", "Tarragona")
proporciones_provincias_filtradas <- proporciones_provincias[provincias_seleccionadas]
# Muestra las proporciones por provincias
print(proporciones_provincias_filtradas)

```

```

##
## Barcelona      Girona      Lleida Tarragona
## 59.244974 16.739242  8.544692 15.471091

```

Se descarga el archivo de “frecuencia\_por\_provincias.csv” en formato.csv

```

# Guarda el dataframe 'frecuencias_filtradas' en un archivo CSV
write.csv(frecuencias_filtradas, "frecuencia_por_provincias.csv", row.names = FALSE)

```

Se descarga el archivo de “proporcion\_frecuencia\_por\_provincias.csv” en formato.csv

```

# Guarda el dataframe 'proporciones_provincias_filtradas' en un archivo CSV
write.csv(proporciones_provincias_filtradas, "proporcion_frecuencia_por_provincias.csv", row.names = FALSE)

```

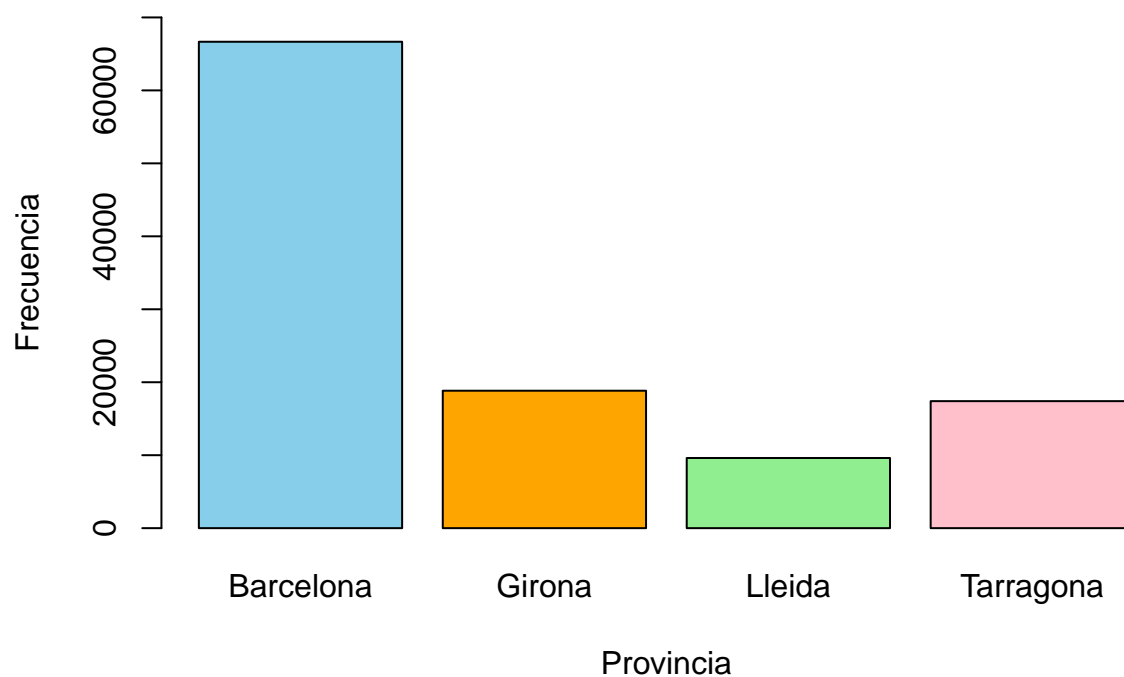
Se realiza el gráfico de barras para las frecuencias de las instalaciones de autoconsumo por provincias

```

# Gráfico de barras de las frecuencias por provincias
barplot(frecuencias_filtradas,
        main = "Frecuencias instalaciones de autoconsumo por Provincias",
        col = c("skyblue", "orange", "lightgreen", "pink"),
        ylab = "Frecuencia",
        xlab = "Provincia",
        ylim = c(0,70000),
        border = "black")

```

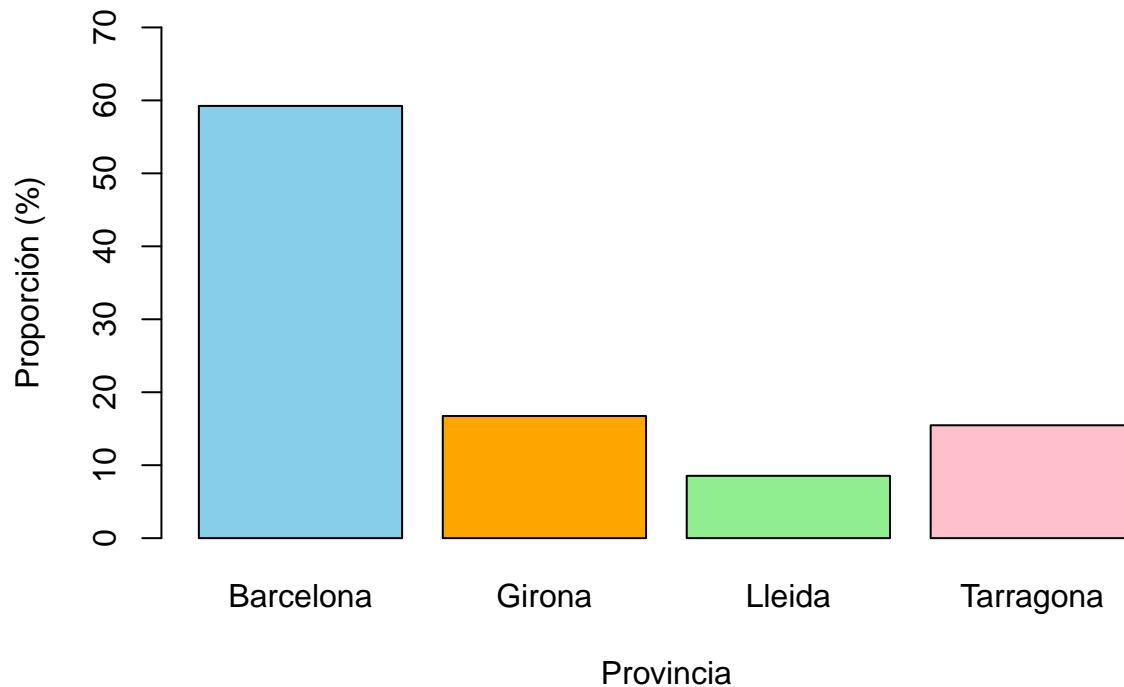
## Frecuencias instalaciones de autoconsumo por Provincias



Se realiza el gráfico de barras para las proporciones de las instalaciones de autoconsumo por provincias

```
# Gráfico de barras de las proporciones por provincias
barplot(proporciones_provincias_filtradas,
  main = "Proporción instalaciones de autoconsumo por Provincias",
  col = c("skyblue", "orange", "lightgreen", "pink"),
  ylab = "Proporción (%)",
  xlab = "Provincia",
  ylim = c(0,70),
  border = "black")
```

## Proporción instalaciones de autoconsumo por Provincias



Se calcula la potencia de las instalaciones de autoconsumo por provincias

```
# Carga la libreria de tidyverse
library(tidyverse)

# Transforma la variable 'Potència' a numérica
datos_1$Potència <- as.numeric(gsub(",", ".", datos_1$Potència))

# Se Agrupa por provincia y se calcula la potencia total para cada una de ellas
potencia_por_provincias <- datos_1 %>%
  group_by(Provincia) %>%
  summarise(
    total_potencia = sum(Potència, na.rm = TRUE)
  )

# Muestra el resultado
print(potencia_por_provincias)
```

```
## # A tibble: 4 x 2
##   Provincia total_potencia
##   <chr>         <dbl>
## 1 Barcelona    7193120
## 2 Girona       1879316
## 3 Lleida       829036
## 4 Tarragona    1664879
```

```
# Calcula las proporciones de la potencia por provincia
proporciones_potencia_por_provincias <- potencia_por_provincias %>%
  mutate(proporcion = (total_potencia / sum(total_potencia)) * 100)

# Filtra solo las provincias deseadas
provincias_seleccionadas <- c("Barcelona", "Girona", "Lleida", "Tarragona")
proporciones_provincias_filtradas <- proporciones_potencia_por_provincias %>%
  filter(Provincia %in% provincias_seleccionadas)

# Muestra las proporciones por provincias
print(proporciones_provincias_filtradas)
```

```
## # A tibble: 4 x 3
##   Provincia total_potencia proporcion
##   <chr>          <dbl>         <dbl>
## 1 Barcelona      7193120         62.2
## 2 Girona         1879316         16.2
## 3 Lleida         829036          7.17
## 4 Tarragona     1664879         14.4
```

Se extraen los datos con el total de las potencias por provincias

```
# Guarda el dataframe 'potencia_por_provincias' en un archivo CSV
write.csv(potencia_por_provincias, "potencia_por_provincias.csv", row.names = FALSE)
```

Se extraen los datos con la proporción del total de las potencias por provincias

```
# Guarda el dataframe 'proporciones_provincias_filtradas' en un archivo CSV
write.csv(proporciones_provincias_filtradas, "proporcion_potencia_por_provincias.csv", row.names = FALSE)
```

Se calcula la frecuencia y las proporciones de las instalaciones por tecnología

```
# Calcula las frecuencias de las instalaciones por tecnología
frecuencias_tecnologia <- table(datos_1$Tecnologia)
# Muestra las frecuencias de las tecnologías deseadas
tecnologias_seleccionadas <- c("HIDROELECTRICA", "COGENERACIO", "RESIDUS", "BIOGÀS", "FOTOVOLTAICA",
                              "ENERGIES RESIDUALS", "EÒLICA", "EOLICA")
tecnologias_filtradas <- frecuencias_tecnologia[tecnologias_seleccionadas]
# Muestra el resultado de las frecuencias por tecnología
print(tecnologias_filtradas)
```

```
##
##   HIDROELECTRICA      COGENERACIO      RESIDUS      BIOGÀS
##           7           19           3           14
##   FOTOVOLTAICA ENERGIES RESIDUALS      EÒLICA      EOLICA
##   112478           1           2           1
```

```
# Calcula proporciones y multiplica por 100 para porcentaje
proporciones_tecnologias <- prop.table(frecuencias_tecnologia) * 100
# Filtra solo las tecnologías deseadas
tecnologias_seleccionadas <- c("HIDROELECTRICA", "COGENERACIO", "RESIDUS", "BIOGÀS", "FOTOVOLTAICA",
```

```

"ENERGIES RESIDUALS", "EÒLICA", "EOLICA")
proporciones_tecnologias_filtradas <- proporciones_tecnologias[tecnologias_seleccionadas]
# Muestra las proporciones por tecnologías
print(proporciones_tecnologias_filtradas)

```

```

##
##      HIDROELECTRICA      COGENERACIO      RESIDUS      BIOGÀS
##      6.220785e-03      1.688499e-02      2.666051e-03      1.244157e-02
##      FOTOVOLTAICA ENERGIES RESIDUALS      EÒLICA      EOLICA
##      9.995734e+01      8.886835e-04      1.777367e-03      8.886835e-04

```

Se calcula la potencia de las instalaciones de autoconsumo por Tecnologías

```

# Carga la libreria de tidyverse
library(tidyverse)

# Convierte la columna de 'Potència' a variable numérica
datos_1$Potència <- as.numeric(gsub(",", ".", datos_1$Potència))

# Agrupa por tecnologías y calcula la potencia total para cada una de ellas
potencia_por_tecnologias <- datos_1 %>%
  group_by(Tecnologia) %>%
  summarise(
    total_potencia = sum(Potència, na.rm = TRUE)
  )

# Muestra el resultado de potencia por tecnología
print(potencia_por_tecnologias)

```

```

## # A tibble: 9 x 2
##   Tecnologia      total_potencia
##   <chr>          <dbl>
## 1 #N/A          0
## 2 BIOGÀS        4979
## 3 COGENERACIO   191984
## 4 ENERGIES RESIDUALS 4330
## 5 EOLICA        5
## 6 EÒLICA        20
## 7 FOTOVOLTAICA   11348369
## 8 HIDROELECTRICA 3704
## 9 RESIDUS       12960

```

```

# Calcula las proporciones de la potencia por tecnología
proporciones_potencia_por_tecnologias <- potencia_por_tecnologias %>%
  mutate(proporcion = (total_potencia / sum(total_potencia)) * 100)

# Filtra solo las tecnologías deseadas
tecnologias_seleccionadas <- c("HIDROELECTRICA", "COGENERACIO", "RESIDUS", "BIOGÀS", "FOTOVOLTAICA",
  "ENERGIES RESIDUALS", "EÒLICA", "EOLICA")
proporciones_tecnologias_filtradas <- proporciones_potencia_por_tecnologias %>%
  filter(Tecnologia %in% tecnologias_seleccionadas)

```

```
# Muestra las proporciones de las tecnologías
print(proporciones_tecnologias_filtradas)
```

```
## # A tibble: 8 x 3
##   Tecnologia      total_potencia proporción
##   <chr>          <dbl>      <dbl>
## 1 BIOGÀS          4979    0.0430
## 2 COGENERACIO     191984   1.66
## 3 ENERGIES RESIDUALS  4330    0.0374
## 4 EOLICA           5    0.0000432
## 5 EÒLICA          20    0.000173
## 6 FOTOVOLTAICA    11348369 98.1
## 7 HIDROELECTRICA   3704    0.0320
## 8 RESIDUS        12960    0.112
```

Se extraen en formato .csv los datos con el total de las potencias por tecnologías

```
# Guarda el dataframe 'potencia_por_tecnologias' en un archivo CSV
write.csv(potencia_por_tecnologias, "potencia_por_tecnologias.csv", row.names = FALSE)
```

Se extraen en formato .csv los datos con la proporción del total de las potencias por tecnologías

```
# Guarda el dataframe 'proporciones_tecnologias_filtradas' en un archivo CSV
write.csv(proporciones_tecnologias_filtradas, "proporción_potencia_por_tecnologias.csv", row.names = FALSE)
```

Se extraen en formato .csv los datos con el tratamiento realizado en archivo de excel

```
# Guarda el dataframe 'datos_1' en un archivo CSV
write.csv(datos_1, "Instalaciones_autoconsumo_electrico_2024.csv", row.names = FALSE)
```

Como punto final, se calculan las instalaciones en función de la potencia que se han puesto en servicio en 4 periodos diferentes.

Se asigna tipo de variable en “Data.de.posada.en.servei” y “Potència”

```
# Transforma la variable de "Data.de.posada.en.servei" a fecha
datos_1$Data.de.posada.en.servei=as.Date(datos_1$Data.de.posada.en.servei,format="%d/%m/%Y")
head(datos_1$Data.de.posada.en.servei)
```

```
## [1] "1987-11-05" "2020-07-14" "1991-06-25" "1998-11-06" "1990-06-20"
## [6] "1996-02-26"
```

```
# Se verifica que la variable de 'Data.de.posada.en.servei' se ha transformado como variable de fecha
str(datos_1$Data.de.posada.en.servei)
```

```
## Date[1:112526], format: "1987-11-05" "2020-07-14" "1991-06-25" "1998-11-06" "1990-06-20" ...
```

```
# Transforma la variable de "Potència" a numérica
datos_1$Potència <- as.numeric(datos_1$Potència)
```

Se calcula la potencia de las instalaciones inscritas entre los años 1985-1995, 1995-2005, 2005-2015 2015-2024



```

# Se carga la librería dplyr
library(dplyr)

# Suma las potencias desde el año 1985 al 1995
suma_1985_1995 <- datos_1 %>%
  filter(Data.de.posada.en.servei >= as.Date("1985-01-01") & Data.de.posada.en.servei < as.Date("1994-12-31"))
  summarise(Suma_Potencia = sum(Potència, na.rm = TRUE))
# Suma las potencias desde el año 1995 al 2005
suma_1995_2005 <- datos_1 %>%
  filter(Data.de.posada.en.servei >= as.Date("1995-01-01") & Data.de.posada.en.servei < as.Date("2004-12-31"))
  summarise(Suma_Potencia = sum(Potència, na.rm = TRUE))
# Suma las potencias desde el año 2005 al 2015
suma_2005_2015 <- datos_1 %>%
  filter(Data.de.posada.en.servei >= as.Date("2005-01-01") & Data.de.posada.en.servei <= as.Date("2014-12-31"))
  summarise(Suma_Potencia = sum(Potència, na.rm = TRUE))
# Suma las potencias desde el año 2015 al 2024
suma_2015_2024 <- datos_1 %>%
  filter(Data.de.posada.en.servei >= as.Date("2015-01-01") & Data.de.posada.en.servei <= as.Date("2024-12-31"))
  summarise(Suma_Potencia = sum(Potència, na.rm = TRUE))

# Muestra los resultados de los 4 períodos
print(suma_1985_1995)

```

```

## Suma_Potencia
## 1          53273

```

```

print(suma_1995_2005)

```

```

## Suma_Potencia
## 1          81786

```

```

print(suma_2005_2015)

```

```

## Suma_Potencia
## 1          68467

```

```

print(suma_2015_2024)

```

```

## Suma_Potencia
## 1        11362825

```

Se calculan las potencias para cada una de las comarcas

```

# Se carga la librería dplyr
library(dplyr)
# Agrupa por comarcas y calcula la potencia total para cada una de ellas
potencia_por_comarcas <- datos_1 %>%
  group_by(Comarca) %>%
  summarise(
    total_potencia = sum(Potència, na.rm = TRUE)
  )

```

```
)

# Muestra el resultado de la portencia por comarcas
print(potencia_por_comarcas)
```

```
## # A tibble: 49 x 2
##   Comarca      total_potencia
##   <chr>          <dbl>
## 1 Alt Camp      117200
## 2 Alt Empordà   369133
## 3 Alt Penedès   315503
## 4 Alt Urgell    36042
## 5 Alt penedès    359
## 6 Alta Ribagorça 6976
## 7 Anoia         308890
## 8 Aran          805
## 9 BAGES         250
## 10 Bages        374570
## # i 39 more rows
```

Se extraen en formato .csv los datos con el total de las potencias por comarcas

```
# Guarda el dataframe 'potencia_por_comarcas' en un archivo CSV
write.csv(potencia_por_comarcas, "potencia_por_comarcas.csv", row.names = FALSE)
```