

emissions_analysis_spain

Carlos Vega González

Emissions-Analysis-Spain: Carlos Vega González

Índice de contenidos

1. Introducción
 2. Carga de datos y paquetes necesarios
 3. Transformación de las columnas anuales
 4. Limpieza de la columna Year
 5. Manejo de valores faltantes
 6. Filtrado de datos por países válidos
 7. Análisis de emisiones de gases de efecto invernadero
 8. Visualización de emisiones de gases de efecto invernadero
 9. Análisis de gases fluorados
 10. Visualización de emisiones por tipo de gas en España
 11. Industria más contaminante en España
 12. Creación de la categoría LEVEL_EMISSIONS
-

1. Introducción

El objetivo principal es obtener información relevante sobre las emisiones de gases contaminantes y estudiar cómo varían a lo largo del tiempo. Con este fin, a lo largo del documento se emplean diferentes técnicas de transformación, limpieza y visualización de datos que permiten extraer conclusiones valiosas a partir de un conjunto complejo de observaciones.

2. Carga de datos y paquetes necesarios

¿Cómo lo resolví?

Para llevar a cabo el EDA, se requiere contar con diversas librerías de manipulación y visualización en R, como **tidyr**, **dplyr** y **ggplot2**. Una vez cargados los paquetes, se importa el conjunto de datos que contiene la información de emisiones y temperaturas.

Características del código:

- Se incluyen los paquetes esenciales para la limpieza, transformación y visualización de datos.
- Se realiza la lectura del archivo CSV con las emisiones anuales.

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(here)
```

```
## here() starts at C:/Users/carlo/Emissions-Analysis-Spain
```

```
df <- read.csv(here("data", "annual_surface_temperature_change.csv"))
```

```
head(df)
```

```
##           Country ISO3           Indicator
## 1 Advanced Economies AETMP Annual Net Emissions/Removals
## 2 Advanced Economies AETMP Annual Net Emissions/Removals
## 3 Advanced Economies AETMP Annual Net Emissions/Removals
## 4 Advanced Economies AETMP Annual Net Emissions/Removals
## 5 Advanced Economies AETMP Annual Net Emissions/Removals
## 6 Advanced Economies AETMP Annual Net Emissions/Removals
##           Unit CTS_Code           CTS_Name
## 1 Million metric tons of CO2 equivalent ECNGDR Annual Net Emissions/Removals
## 2 Million metric tons of CO2 equivalent ECNGDR Annual Net Emissions/Removals
## 3 Million metric tons of CO2 equivalent ECNGDR Annual Net Emissions/Removals
## 4 Million metric tons of CO2 equivalent ECNGDR Annual Net Emissions/Removals
## 5 Million metric tons of CO2 equivalent ECNGDR Annual Net Emissions/Removals
## 6 Million metric tons of CO2 equivalent ECNGDR Annual Net Emissions/Removals
```

##	Industry	Gas_Type	F1970	F1971	F1972		
## 1	Agriculture	Carbon dioxide	20.88061	20.90178	21.0756		
## 2	Agriculture	Greenhouse gas	1033.08823	1040.84282	1047.9256		
## 3	Agriculture	Methane	682.42929	683.55683	687.6448		
## 4	Agriculture	Nitrous oxide	329.77833	336.38421	339.2052		
## 5	Buildings and other Sectors	Carbon dioxide	1954.79994	1966.15720	2027.9067		
## 6	Buildings and other Sectors	Greenhouse gas	2021.34487	2027.30977	2085.5205		
##	F1973	F1974	F1975	F1976	F1977	F1978	F1979
## 1	21.4966	21.5829	22.27892	22.69633	22.87232	23.34186	23.67585
## 2	1060.9815	1079.6948	1106.31771	1101.24346	1091.06990	1092.01945	1098.01927
## 3	693.6221	715.1081	721.97998	714.41997	704.14859	695.94847	690.70742
## 4	345.8628	343.0038	362.05880	364.12716	364.04899	372.72912	383.63600
## 5	2043.1960	1936.5554	1915.62013	2019.09638	1975.88609	2007.32809	1992.52330
## 6	2100.0210	1992.5491	1969.40888	2073.02843	2029.48069	2060.73756	2049.90122
##	F1980	F1981	F1982	F1983	F1984	F1985	F1986
## 1	24.41998	24.4313	24.41967	25.27343	25.17984	26.05422	26.13985
## 2	1102.96018	1105.1247	1099.06584	1091.17287	1110.97642	1104.16805	1098.86367
## 3	696.89165	698.2196	696.01244	686.31127	692.88804	688.64104	683.56636
## 4	381.64855	382.4738	378.63373	379.58817	392.90854	389.47279	389.15746
## 5	1819.97033	1736.4197	1685.14981	1662.94712	1714.23716	1737.38721	1742.68487
## 6	1873.82084	1789.7035	1739.38672	1718.25960	1768.64742	1793.66007	1798.86937
##	F1987	F1988	F1989	F1990	F1991	F1992	F1993
## 1	25.86996	25.96634	26.03349	22.59076	21.09534	19.38771	18.66021
## 2	1079.53991	1070.84518	1080.43720	1187.86998	1166.72242	1153.61517	1174.48659
## 3	668.36022	665.99125	664.90706	615.72891	610.74495	607.49185	608.59934
## 4	385.30974	378.88759	389.49665	549.55031	534.88214	526.73561	547.22704
## 5	1736.87574	1749.86169	1701.10784	1607.33278	1654.15481	1613.92776	1647.49590
## 6	1792.86703	1804.94057	1755.36820	1649.91952	1696.23608	1653.64302	1685.92254
##	F1994	F1995	F1996	F1997	F1998	F1999	F2000
## 1	19.38278	20.19042	20.11904	19.74929	21.11679	21.4564	21.36624
## 2	1160.97363	1173.37117	1185.36148	1173.45796	1176.44629	1166.4071	1149.54907
## 3	613.23357	617.95942	618.23699	613.91722	613.21087	611.3288	610.49815
## 4	528.35728	535.22132	547.00546	539.79145	542.11862	533.6219	517.68468
## 5	1603.04937	1624.42720	1739.09851	1671.56359	1596.85643	1623.5783	1646.79427
## 6	1638.82776	1659.66797	1775.30639	1705.56017	1628.46122	1654.8282	1677.77723
##	F2001	F2002	F2003	F2004	F2005	F2006	F2007
## 1	21.45535	22.37747	22.79268	21.96567	22.05452	21.66871	22.6995
## 2	1164.69876	1159.42479	1148.76797	1160.13109	1151.02719	1139.64448	1157.3065
## 3	605.52727	604.09107	595.93127	592.07119	600.56592	596.63984	601.7565
## 4	537.71613	532.95625	530.04402	546.09423	528.40675	521.33593	532.8505
## 5	1664.67648	1635.82463	1676.81788	1656.20873	1626.17116	1540.46249	1498.9572
## 6	1695.06959	1664.22728	1705.77410	1684.35348	1655.12791	1568.72861	1528.8616
##	F2008	F2009	F2010	F2011	F2012	F2013	F2014
## 1	23.05679	21.71696	23.14387	22.9582	25.79134	24.54153	24.83977
## 2	1143.01153	1141.92075	1145.27353	1129.1092	1116.60783	1161.17023	1173.41862
## 3	592.14657	590.13786	588.43869	583.6073	585.05505	581.65391	581.54015
## 4	527.80817	530.06594	533.69097	522.5437	505.76144	554.97479	567.03871
## 5	1541.86827	1497.33779	1528.64337	1435.7811	1370.82075	1456.71773	1384.94582
## 6	1573.27185	1528.48657	1560.57387	1464.9221	1399.93303	1487.56936	1413.99670
##	F2015	F2016	F2017	F2018	F2019	F2020	F2021
## 1	25.49511	25.3845	25.21957	24.67924	24.98413	25.10287	25.03577
## 2	1179.37760	1167.4599	1172.24202	1180.33767	1178.92966	1148.43100	1143.35208
## 3	587.09166	592.5807	597.60823	599.38026	592.82555	590.21630	584.74042
## 4	566.79083	549.4947	549.41422	556.27817	561.11998	533.11183	533.57589

```
## 5 1392.70155 1363.3443 1368.89682 1417.21527 1414.01903 1348.43847 1384.78739
## 6 1421.93906 1391.4140 1397.20120 1446.08507 1442.81014 1375.84559 1412.66068
```

3. Transformación de las columnas anuales

¿Cómo lo resolví?

Para analizar las emisiones de manera más efectiva, es conveniente transformar las columnas anuales (**F1970** a **F2021**) en un formato largo (*long format*). Esto facilita agrupar, filtrar y resumir la información por año y por país.

Características del código:

- Se utiliza la función `gather()` para pasar de un formato amplio a uno largo.
- Se registra el número de filas y columnas antes y después de la transformación para confirmar el proceso.

```
df_transformado <- gather(df, key = "Year", value = "Value", F1970:F2021)
dimensiones_original <- dim(df)
dimensiones_transformado <- dim(df_transformado)
dimensiones_original
```

```
## [1] 20448    60
```

```
dimensiones_transformado
```

```
## [1] 1063296   10
```

4. Limpieza de la columna Year

¿Cómo lo resolví?

La columna `Year` contiene un prefijo “F” que es innecesario para el análisis. Por ello, se elimina mediante la función `sub()` y, posteriormente, se convierte la columna a factor para un mejor manejo de los datos categóricos.

Características del código:

- El prefijo “F” se elimina con la expresión regular adecuada.
- Se trabaja con la columna resultante como factor para facilitar los agrupamientos posteriores.

```
df_transformado$Year <- as.factor(sub("F", "", df_transformado$Year))
```

5. Manejo de valores faltantes

¿Cómo lo resolví?

Para evitar sesgos en el análisis debido a la presencia de valores faltantes en la columna `Value`, se optó por sustituir estos valores por la media anual de cada año. De esta forma, se preserva la variabilidad por períodos en lugar de usar una media global.

Características del código:

- Se agrupan los datos por `Year`.
- Se utiliza la media anual como criterio de reemplazo de valores NA.
- Se mantiene la estructura del conjunto de datos al finalizar el proceso.

```
df_transformado <- df_transformado %>%  
  group_by(Year) %>%  
  mutate(Value = ifelse(is.na(Value), round(mean(Value, na.rm = TRUE), 2), round(Value, 2))) %>%  
  ungroup()  
  
anyNA(df_transformado$Value)
```

```
## [1] FALSE
```

```
df_transformado$Value[1:50]
```

```
## [1] 20.88 1033.09 682.43 329.78 1954.80 2021.34 51.82 14.72 96.10  
## [10] 73.81 339.34 0.90 242.34 152.47 153.77 0.03 1.27 53.00  
## [19] 53.54 0.12 0.42 30.33 30.33 9139.45 9841.49 647.35 54.68  
## [28] 2865.32 2875.25 1.08 8.86 186.59 763.49 576.80 0.10 627.30  
## [37] 103.30 984.39 2.29 251.51 73.81 73.81 73.81 73.81 2484.02  
## [46] 2495.68 4.01 7.65 112.45 72.96
```

6. Filtrado de datos por países válidos

¿Cómo lo resolví?

Dado que el conjunto de datos puede incluir agregados regionales o clasificaciones (p. ej., “Advanced Economies”) que no corresponden a países específicos, se optó por filtrar los datos conservando únicamente aquellos países válidos encontrados en una lista externa.

Características del código:

- Se lee un archivo CSV que contiene la lista de países válidos.
- Se realiza el filtrado comparando la columna `Country` con los niveles disponibles en la lista.
- Se elimina la información de países no contemplados en la lista para obtener un subconjunto más exacto.

```

países <- read.csv(here("data", "lista_paises_final.csv"), fileEncoding = "ISO-8859-1")

países$Country <- as.factor(países$Country)

df_filtrado <- df_transformado %>%
  filter(Country %in% levels(países$Country))

df_filtrado$Country <- droplevels(as.factor(df_filtrado$Country))
nrow(df_filtrado)

## [1] 820872

```

7. Análisis de emisiones de gases de efecto invernadero

¿Cómo lo resolví?

Para focalizar el estudio en las emisiones de gases de efecto invernadero (GEI), se filtraron únicamente las observaciones correspondientes a la variable `Gas_Type == "Greenhouse gas"`. Luego, se calculó el total de emisiones por año y se ordenaron de mayor a menor.

Características del código:

- Se seleccionan solo las filas cuyo tipo de gas es **Greenhouse gas**.
- Se agrupan las emisiones por año y se suman para obtener el total anual.
- El resultado se ordena para identificar rápidamente los años con mayores emisiones.

```

greenhouse <- subset(df_filtrado, Gas_Type == "Greenhouse gas")
emisiones_greenhouse <- aggregate(Value ~ Year, data = greenhouse, sum)
emisiones_greenhouse <- emisiones_greenhouse[order(-emisiones_greenhouse$Value), ]
emisiones_greenhouse[1:3, ]

```

```

##      Year      Value
## 52 2021 207942.0
## 50 2019 207173.1
## 49 2018 201381.7

```

8. Visualización de emisiones de gases de efecto invernadero

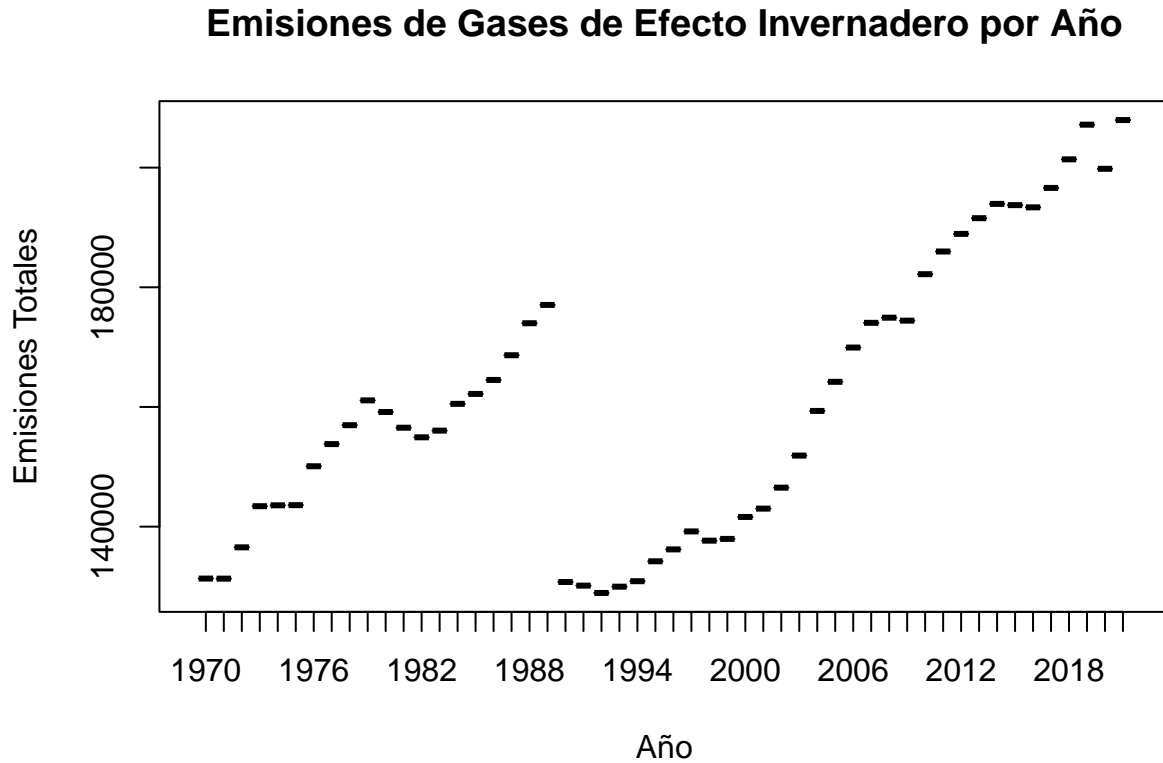
¿Cómo lo resolví?

Para representar gráficamente la tendencia de las emisiones totales de GEI a lo largo de los años, se utiliza un diagrama de líneas. Este tipo de visualización facilita la identificación de patrones o posibles cambios bruscos en las emisiones.

Características del código:

- Se realiza un gráfico de líneas con el año en el eje **X** y las emisiones totales en el eje **Y**.
- El título y las etiquetas de los ejes se establecen para mayor claridad.

```
plot(emisiones_greenhouse$Year, emisiones_greenhouse$Value, type = "l",  
      main = "Emisiones de Gases de Efecto Invernadero por Año",  
      xlab = "Año", ylab = "Emisiones Totales")
```



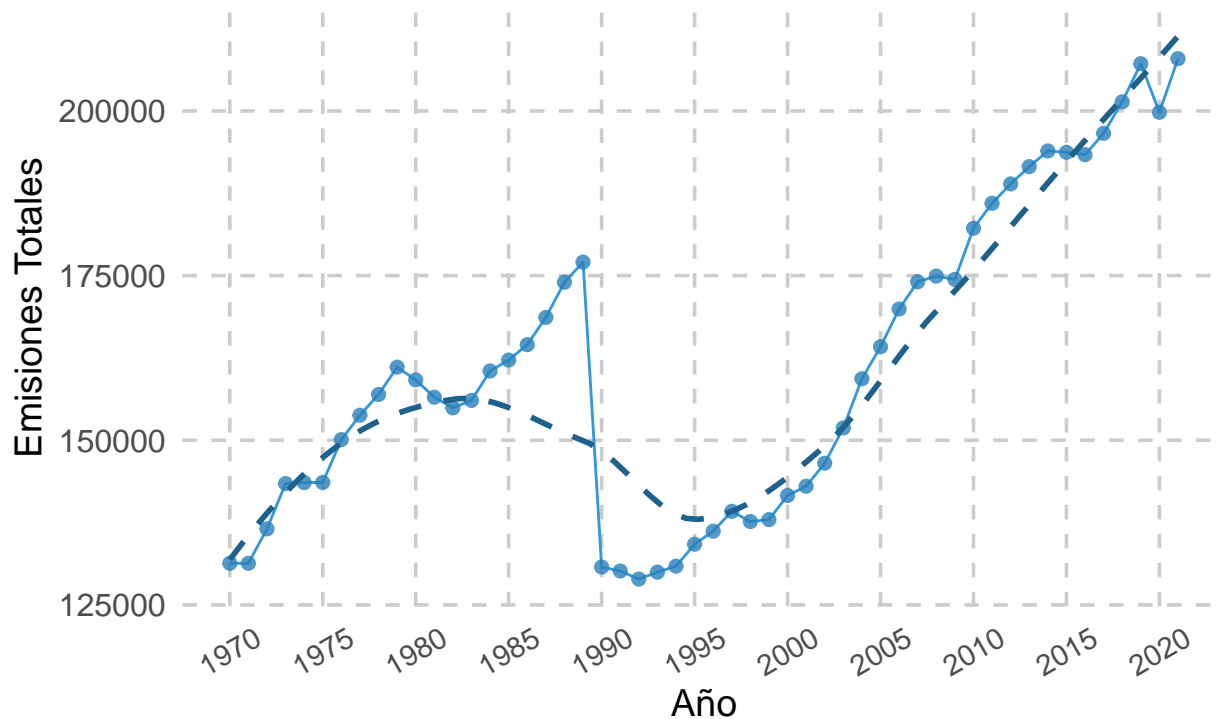
```
emisiones_greenhouse$Year <- as.numeric(as.character(emisiones_greenhouse$Year))  
  
ggplot(emisiones_greenhouse, aes(x = Year, y = Value)) +  
  geom_line(color = "#3498DB") +  
  geom_point(color = "#2980B9", size = 2, alpha = 0.8) +  
  geom_smooth(se = FALSE, linetype = "dashed", color = "#1F618D") +  
  scale_x_continuous(breaks = seq(min(emisiones_greenhouse$Year, na.rm = TRUE),  
                                  max(emisiones_greenhouse$Year, na.rm = TRUE), by = 5)) +  
  labs(title = "Emisiones de Gases de Efecto Invernadero por Año",  
        subtitle = "Tendencia histórica de emisiones",  
        x = "Año", y = "Emisiones Totales") +  
  theme_minimal(base_size = 14) +  
  theme(  
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5),  
    plot.subtitle = element_text(size = 12, hjust = 0.5),  
    axis.text.x = element_text(angle = 30, vjust = 0.5),
```

```
axis.text.y = element_text(size = 12),
panel.grid.major = element_line(linetype = "dashed", color = "gray80"),
panel.grid.minor = element_blank()
)
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

Emisiones de Gases de Efecto Invernadero por Año

Tendencia histórica de emisiones



9. Análisis de gases fluorados

¿Cómo lo resolví?

Se pone el foco en las emisiones de gases fluorados, buscando identificar si hubo una caída destacable en algún año en particular. Después, se determina qué países tuvieron mayores emisiones promedio en ese período específico.

Características del código:

- Se filtran los datos donde `Gas_Type == "fluorinado gases"`.
- Se agrupan las emisiones por año y se calculan las diferencias para detectar el año con la mayor caída.
- Se listan los cinco países con mayor promedio de emisiones en el año identificado.


```

fluorinado <- df_filtrado %>%
  filter(Gas_Type == "Fluorinated gases")

fluorinado_summary <- fluorinado %>%
  group_by(Year) %>%
  summarise(total_emissions = sum(Value, na.rm = TRUE)) %>%
  arrange(desc(total_emissions))

diferencias <- fluorinado_summary %>%
  mutate(diff = c(NA, diff(total_emissions))) %>%
  filter(!is.na(diff)) %>%
  slice(which.min(diff) + 1) %>%
  pull(Year)

fluorinado_caida <- fluorinado %>%
  filter(Year == diferencias)

top5_paises <- fluorinado_caida %>%
  group_by(Country) %>%
  summarise(media_emisiones = mean(Value, na.rm = TRUE)) %>%
  arrange(desc(media_emisiones)) %>%
  slice_head(n = 5)

top5_paises

```

```

## # A tibble: 5 x 2
##   Country          media_emisiones
##   <fct>              <dbl>
## 1 China, P.R.: Mainland      256.
## 2 United States             116.
## 3 India                     35.4
## 4 Japan                     33.6
## 5 Saudi Arabia              32.9

```

10. Visualización de emisiones por tipo de gas en España

¿Cómo lo resolví?

Para analizar de forma particular la evolución de las emisiones en España, se filtran los datos por país y se excluyen aquellos registros donde el tipo de gas se marca como "Not Applicable". Se genera un gráfico de líneas para observar la evolución de cada tipo de gas a lo largo del tiempo.

Características del código:

- Se trabaja únicamente con la información de España (`Country == "Spain"`).
 - Se representan múltiples series en un mismo lienzo, diferenciadas por `Gas_Type`.
 - El uso de **ggplot2** facilita la personalización de ejes, leyendas y colores.
-

```

spain_gas <- df_filtrado %>% filter(Country == "Spain")
spain_gas$Year <- as.numeric(as.character(spain_gas$Year))

ggplot(spain_gas, aes(x = Year, y = Value, color = Gas_Type, group = Gas_Type)) +
  geom_line(linewidth = 1.2) +
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed", linewidth = 0.8) +
  scale_x_continuous(breaks = seq(min(spain_gas$Year, na.rm = TRUE),
                                   max(spain_gas$Year, na.rm = TRUE), by = 5)) +
  scale_color_manual(values = c("Carbon dioxide" = "#E74C3C",
                                "Fluorinated gases" = "#F1C40F",
                                "Greenhouse gas" = "#2ECC71",
                                "Methane" = "#3498DB",
                                "Nitrous oxide" = "#9B59B6")) +
  labs(title = "Evolución de Emisiones por Tipo de Gas en España",
       subtitle = "Datos de emisiones de gases de efecto invernadero por tipo",
       x = "Año", y = "Emisiones",
       color = "Tipo de Gas") +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    axis.text.x = element_text(angle = 45, vjust = 0.5),
    legend.position = "top"
  )

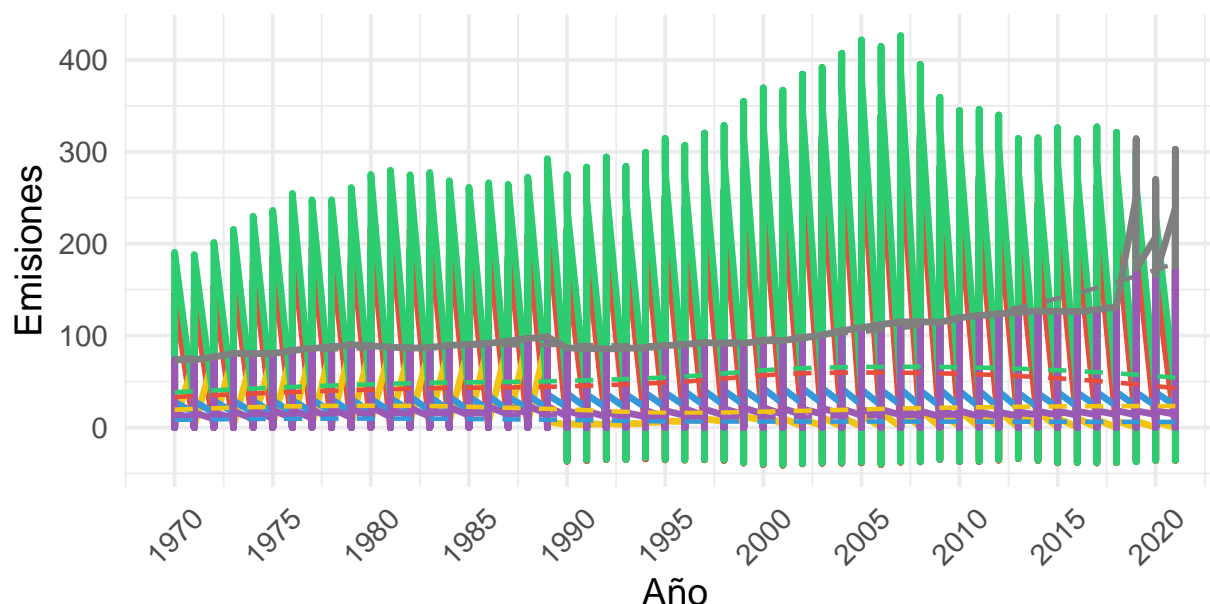
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

Evolución de Emisiones por Tipo de Gas en España

Datos de emisiones de gases de efecto invernadero por tipo

le Gas — Carbon dioxide — Fluorinated gases — Greenhouse gas — Methane —

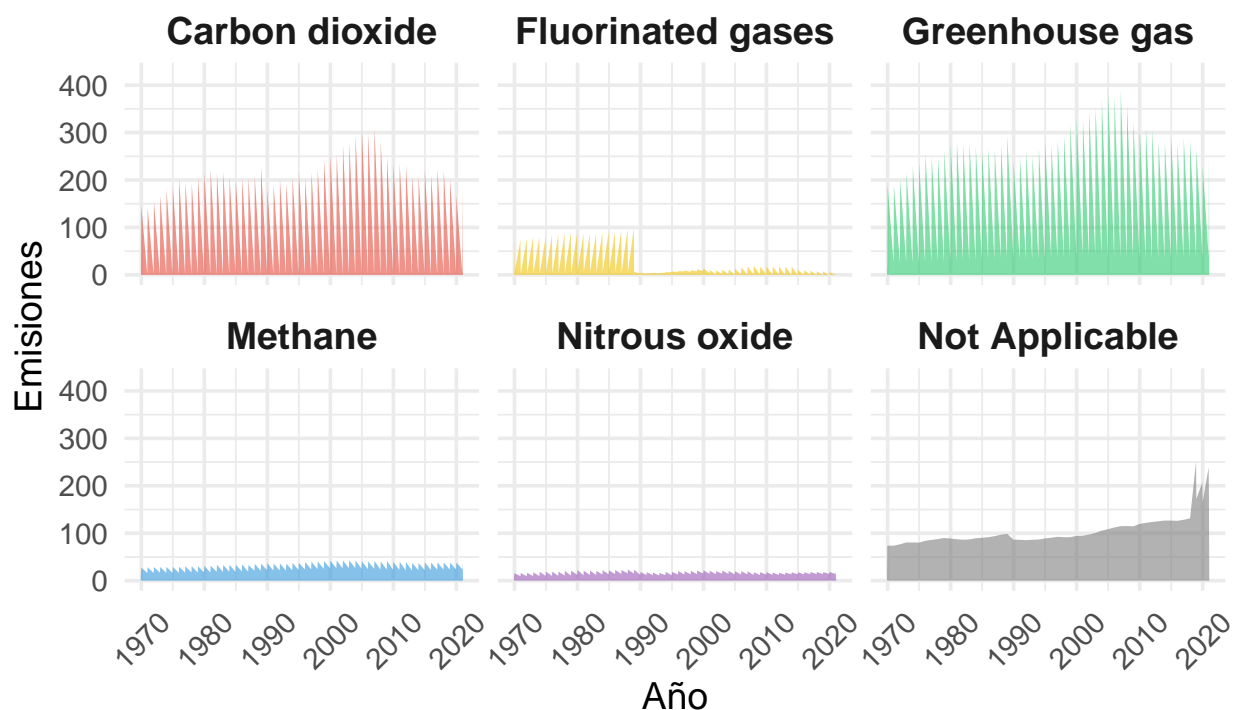


```
y_max <- max(spain_gas$Value, na.rm = TRUE)
ggplot(spain_gas, aes(x = Year, y = Value, fill = Gas_Type)) +
  geom_area(alpha = 0.6) +
  scale_fill_manual(values = c("Carbon dioxide" = "#E74C3C",
                                "Fluorinated gases" = "#F1C40F",
                                "Greenhouse gas" = "#2ECC71",
                                "Methane" = "#3498DB",
                                "Nitrous oxide" = "#9B59B6")) +
  scale_y_continuous(limits = c(0, y_max)) +
  labs(title = "Evolución de Emisiones por Tipo de Gas en España",
        subtitle = "Datos de emisiones de gases de efecto invernadero por tipo",
        x = "Año", y = "Emisiones",
        fill = "Tipo de Gas") +
  theme_minimal(base_size = 14) +
  facet_wrap(~Gas_Type, ncol = 3) +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    axis.text.x = element_text(angle = 45, vjust = 0.5),
    legend.position = "none",
    strip.text = element_text(size = 14, face = "bold"),
  )
```

```
## Warning: Removed 64 rows containing non-finite outside the scale range
## ('stat_align()').
```

Evolución de Emisiones por Tipo de Gas en España

Datos de emisiones de gases de efecto invernadero por tipo



11. Industria más contaminante en España

¿Cómo lo resolví?

Entre los diferentes sectores o industrias, se identifica cuál emite más gases contaminantes en España, basándose en la suma total de emisiones para cada industria.

Características del código:

- Se filtran industrias válidas (excluyendo valores nulos o etiquetas genéricas).
- Se agrupan los datos por industria y se calculan las emisiones totales.
- Se ordenan los resultados para determinar las industrias que más contribuyen a la contaminación.

```
industria_spain <- spain_gas %>%
  filter(!is.na(Industry) & Industry != "") %>%
  group_by(Industry) %>%
  summarise(total_emissions = sum(Value, na.rm = TRUE)) %>%
  arrange(desc(total_emissions))
```

```
industria_spain
```

```
## # A tibble: 25 x 2
```

```
##   Industry
```

```
total_emissions
```

```
##      <chr>                                <dbl>
## 1 Not Applicable                        125389.
## 2 Energy                                24027.
## 3 Energy Industries                      8000.
## 4 Mineral Industry                      7724.
## 5 Electronics Industry                  7218.
## 6 Other                                7214.
## 7 Road Transportation                   6202.
## 8 Manufacturing Industries and Construction 5263.
## 9 Land-use, land-use change and forestry 4551.
## 10 Product Uses as Substitutes for ODS    3955.
## # i 15 more rows
```

```
industria_spain <- spain_gas %>%
  filter(!is.na(Industry) & Industry != "" & Industry != "Not Applicable") %>%
  group_by(Industry) %>%
  summarise(total_emissions = sum(Value, na.rm = TRUE)) %>%
  arrange(desc(total_emissions))

industria_spain
```

```
## # A tibble: 24 x 2
##   Industry                                total_emissions
##   <chr>                                <dbl>
## 1 Energy                                24027.
## 2 Energy Industries                      8000.
## 3 Mineral Industry                      7724.
## 4 Electronics Industry                  7218.
## 5 Other                                7214.
## 6 Road Transportation                   6202.
## 7 Manufacturing Industries and Construction 5263.
## 8 Land-use, land-use change and forestry 4551.
## 9 Product Uses as Substitutes for ODS    3955.
## 10 Non-energy Products from Fuels and Solvent Use 3791.
## # i 14 more rows
```

12. Creación de la categoría LEVEL_EMISSIONS

¿Cómo lo resolví?

Por último, se propone una categorización de las emisiones netas anuales de España en niveles “bajas”, “medias” y “altas” para observar de forma clara los patrones y cambios temporales.

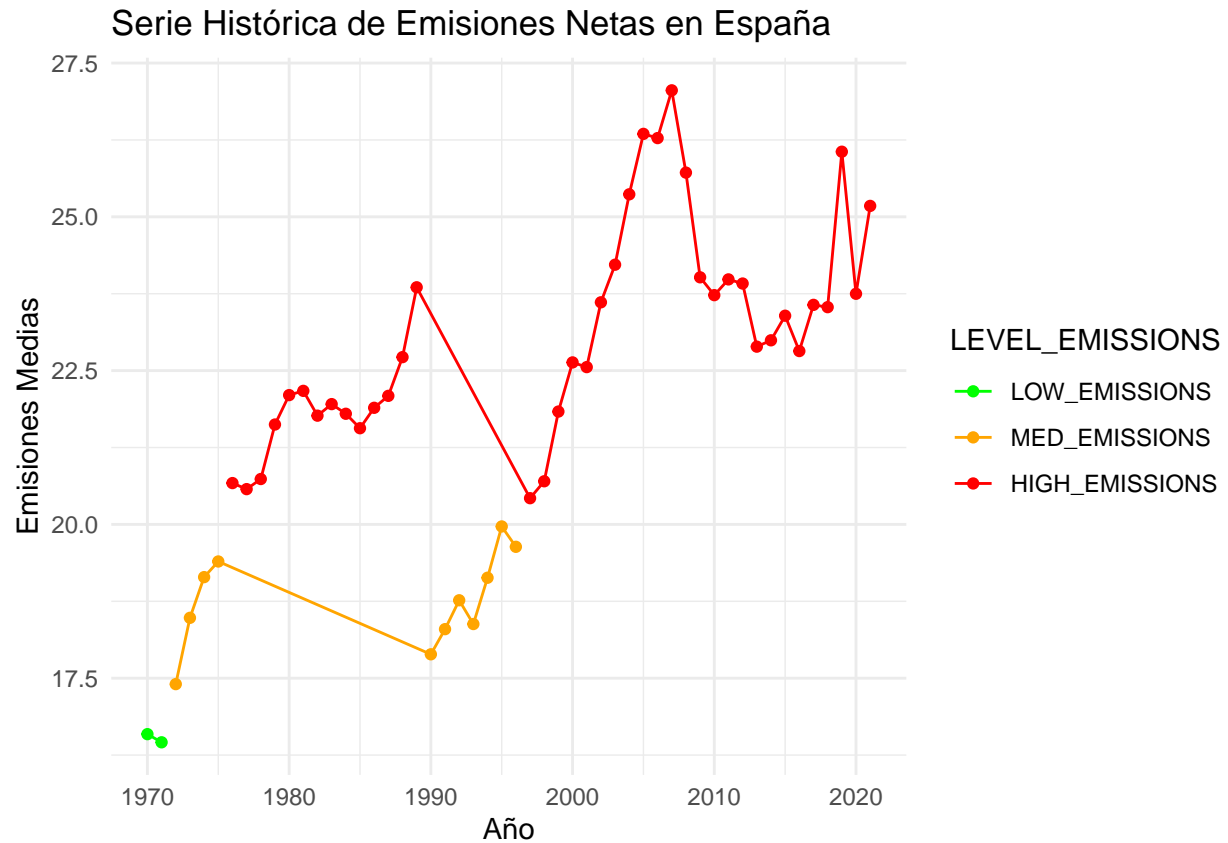
Características del código:

- Se crea la columna LEVEL_EMISSIONS con la función cut() empleando puntos de corte adecuados.
- Se visualiza la serie histórica coloreando los puntos según su nivel de emisiones.
- La paleta de colores se ajusta para dejar claro cada nivel.

```
head(df_filtrado)
```

```
## # A tibble: 6 x 10
##   Country ISO3 Indicator Unit CTS_Code CTS_Name Industry Gas_Type Year Value
##   <fct>    <chr> <chr>    <chr> <chr>    <chr>    <chr>    <chr>    <fct> <dbl>
## 1 Afghani~ AFG   Annual N~ Mill~ ECNGDR   Annual ~ Agricul~ Carbon ~ 1970  0.02
## 2 Afghani~ AFG   Annual N~ Mill~ ECNGDR   Annual ~ Agricul~ Greenho~ 1970  12.2
## 3 Afghani~ AFG   Annual N~ Mill~ ECNGDR   Annual ~ Agricul~ Methane 1970   9.4
## 4 Afghani~ AFG   Annual N~ Mill~ ECNGDR   Annual ~ Agricul~ Nitrous~ 1970   2.75
## 5 Afghani~ AFG   Annual N~ Mill~ ECNGDR   Annual ~ Buildin~ Carbon ~ 1970   0.8
## 6 Afghani~ AFG   Annual N~ Mill~ ECNGDR   Annual ~ Buildin~ Greenho~ 1970   0.82
```

```
spain_anual <- df_filtrado %>%
  filter(Country == "Spain" & Indicator == "Annual Net Emissions/Removals") %>%
  group_by(Year) %>%
  summarise(media_emisiones = mean(Value, na.rm = TRUE)) %>%
  mutate(LEVEL_EMISSIONS = cut(media_emisiones,
                              breaks = c(-Inf, 17, 20, Inf),
                              labels = c("LOW_EMISSIONS", "MED_EMISSIONS", "HIGH_EMISSIONS"))))
spain_anual$Year <- as.numeric(as.character(spain_anual$Year))
ggplot(spain_anual, aes(x = Year, y = media_emisiones, color = LEVEL_EMISSIONS)) +
  geom_point() +
  geom_line() +
  labs(title = "Serie Histórica de Emisiones Netas en España",
       x = "Año", y = "Emisiones Medias") +
  theme_minimal() +
  scale_color_manual(values = c("LOW_EMISSIONS" = "green",
                                "MED_EMISSIONS" = "orange",
                                "HIGH_EMISSIONS" = "red"))
```



```
spain_anual$Year <- as.numeric(as.character(spain_anual$Year))

ggplot(spain_anual, aes(x = Year, y = media_emisiones, color = LEVEL_EMISSIONS)) +
  geom_point(size = 2, alpha = 0.8) +
  geom_line(size = 1) +
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed") +
  scale_color_manual(values = c("LOW_EMISSIONS" = "#2ECC71",
                                "MED_EMISSIONS" = "#F39C12",
                                "HIGH_EMISSIONS" = "#E74C3C")) +
  labs(title = "Serie Histórica de Emisiones Netas en España",
        subtitle = "Clasificación por niveles de emisiones",
        x = "Año", y = "Emisiones Medias",
        color = "Nivel de Emisiones") +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    axis.text.x = element_text(angle = 30, vjust = 0.5),
    axis.text.y = element_text(size = 12),
    legend.position = "top",
    legend.title = element_text(face = "bold"),
    panel.grid.major = element_line(linetype = "dashed", color = "gray80"),
    panel.grid.minor = element_blank()
  )
)
```

```

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

## 'geom_smooth()' using formula = 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : at 1970

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : radius 2.5e-05

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : all data on boundary of neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 1970

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 0.005

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : at 1971

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : radius 2.5e-05

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : all data on boundary of neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 2.5e-05

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : zero-width neighborhood. make span bigger
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : zero-width neighborhood. make span bigger

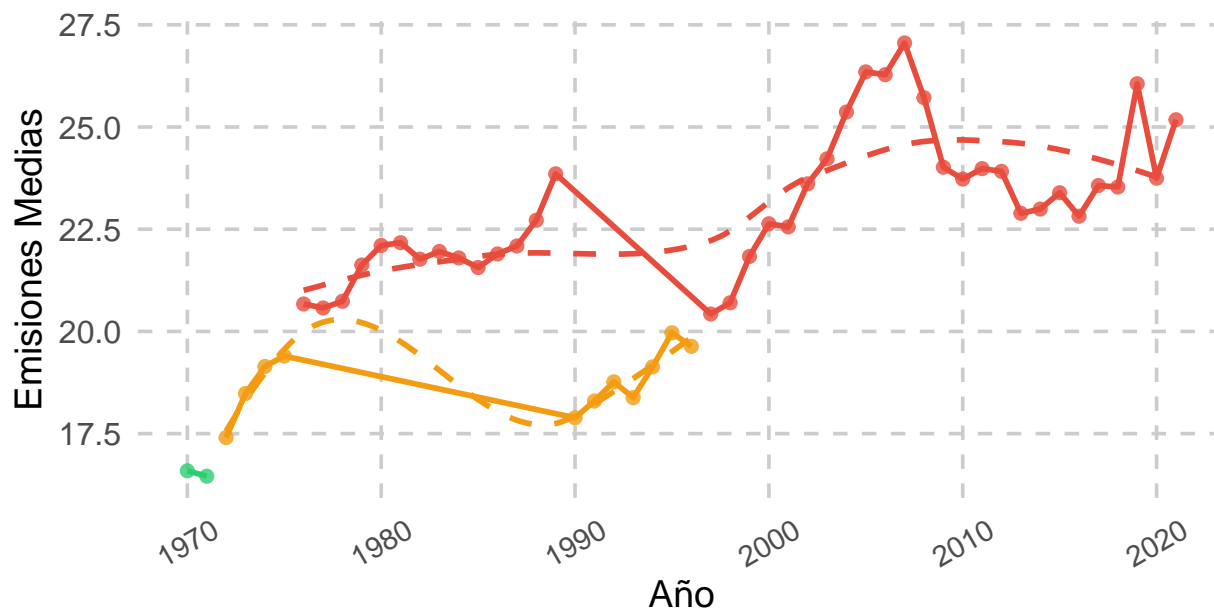
## Warning: Failed to fit group 1.
## Caused by error in 'predLoess()':
## ! NA/NaN/Inf en llamada a una función externa (arg 5)

```


Serie Histórica de Emisiones Netas en España

Clasificación por niveles de emisiones

Nivel de Emisiones ● LOW_EMISSIONS ● MED_EMISSIONS ● HIGH_EMISSIONS



spain_anual

```
## # A tibble: 52 x 3
##   Year media_emisiones LEVEL_EMISSIONS
##   <dbl>         <dbl> <fct>
## 1 1970          16.6 LOW_EMISSIONS
## 2 1971          16.5 LOW_EMISSIONS
## 3 1972          17.4 MED_EMISSIONS
## 4 1973          18.5 MED_EMISSIONS
## 5 1974          19.1 MED_EMISSIONS
## 6 1975          19.4 MED_EMISSIONS
## 7 1976          20.7 HIGH_EMISSIONS
## 8 1977          20.6 HIGH_EMISSIONS
## 9 1978          20.7 HIGH_EMISSIONS
## 10 1979         21.6 HIGH_EMISSIONS
## # i 42 more rows
```