

PAC2

Visualización

José Ignacio Funes Castillo

2023-04-20

Index

3. Tipus de Gràfics

3.1. Gràfic de línies

3.1.1 Origen

El gràfic de línies és una de les formes més antigues de representar dades. El seu creador va ser William Playfair XVIII.

Segons font trobada a l'INE (institut nacional de estadística).

https://www.ine.es/expo_graficos2010/expogra_autor2.htm

3.1.2 Descripció/Funcionament

Aquest tipus de gràfic utilitza línies per a connectar punts de dades en un eix de coordenades, generalment representant una variable en l'eix i i una altra en l'eix x (encara que pot haver-hi més eixos). S'utilitza principalment per a mostrar tendències i canvis al llarg del temps.

3.1.3 Aplicacions

Pot ser utilitzat en una varietat de camps, com a finances (per exemple, per a mostrar el rendiment d'accions), i anàlisis de dades en general.

Font Imatge: https://datavizcatalogue.com/ES/metodos/grafica_de_linea.html

3.2. Gauge diagram

3.2.1 Origen

Els diagrames de mesuradors han estat utilitzats durant dècades en diverses àrees, des de l'enginyeria fins al disseny dels indicadors del quadre de comandament d'en automòbils.

3.2.2 Descripció/Funcionament

Aquest tipus de diagrama representa dades mitjançant una escala circular, amb una agulla que assenyalava un valor específic en aquesta escala. S'utilitza per a mostrar el rendiment o la posició d'una variable en relació amb uns certs límits o estàndards.

Segons font: <https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=types-gauge-charts>

3.2.3 Aplicacions

Els taulers d'automòbils que mostren la velocitat del vehicle són un exemple clàssic de diagrames de mesuradors. També es poden trobar en panells de control de fàbriques per a indicar el rendiment de la maquinària, o en panells de control de programari per a mostrar mètriques clau de rendiment.

Font Imatge: <https://www.phdata.io/blog/how-to-use-the-gauge-chart-template/>

3.3. UpSet: Visualizing Intersecting Sets

3.3.1 Origen

El mètode UpSet va ser proposat en un article de recerca titulat "UpSet: Visualization of Intersecting Sets" per Alexander Lex, Nils Gehlenborg i Hendrik Strobelt, publicat en 2014.

DOI: [10.1109/TVCG.2014.2346248](https://doi.org/10.1109/TVCG.2014.2346248)

3.3.2 Descripció/Funcionament

UpSet és una tècnica de visualització dissenyada per a mostrar conjunts de dades i les seves interseccions. Consisteix en una matriu de barres horitzontals que representen els conjunts de dades, i les interseccions entre ells es mostren com a barres verticals connectades. Aquesta tècnica és útil per a comprendre la interacció entre diferents conjunts de dades i per a identificar patrons d'intersecció.

3.3.3 Aplicacions

S'utilitza en bioinformàtica per a visualitzar la intersecció de conjunts de gens o proteïnes en diferents condicions experimentals. També pot ser útil en anàlisi de dades socials per a visualitzar la intersecció de conjunts d'usuaris en xarxes socials, entre altres camps.

Font Imatge:

https://upload.wikimedia.org/wikipedia/commons/4/44/Upset_Plot.png

4. Tipus de dades i estructures

4.1. Gràfic de línies

Els gràfics de línies són ideals per a representar dades quantitatives que poden ser mesurats en una escala numèrica, com a quantitats, mesures o valors al llarg del temps o un altre tipus de sèrie. No obstant això, també poden ser utilitzats per a dades qualitatives si s'assigna un valor numèric a cada categoria.

Es recomana tenir una estructura de dades en forma de parells de valors (x, y), on "x" representa la variable independent (per exemple, el temps) i "y" representa la variable dependent (per exemple, la mesura o quantitat).

No hi ha una limitació estricta pel que fa al tamany del conjunt de dades per a un gràfic de línies. No obstant això, a mesura que augmenta el nombre de punts de dades, la llegibilitat del gràfic pot disminuir.

4.2. Gauge diagram

Els diagrama de medidors són més adequats per representar dades quantitatives, especialment quan es vol mostrar la posició d'una variable en relació amb certs límits o estàndards.

Es necessita un sol valor numèric que es vol representar en el medidor, juntament amb els límits o estàndards associats.

No hi ha una limitació estricta pel que fa al tamany del conjunt de dades per a un diagrama de medidors. No obstant això, a causa de la seva naturalesa compacta i simplificada, són més efectius quan s'utilitzen per representar un sol valor o mètrica en lloc de conjunts de dades grans i complexos.

4.3. UpSet: Visualizing Intersecting Sets

El gràfic UpSet és més adequat per representar dades qualitatives, especialment quan es vol visualitzar la intersecció entre conjunts de dades.

Les dades s'estructuren en forma de taula o matriu on les files representen els elements individuals i les columnes representen els conjunts, amb valors binaris (1 o 0) que indiquen la presència o absència de l'element en cada conjunt.

UpSet pot manejar conjunts de dades de mida variable. No obstant això, a mesura que augmenta la complexitat de les dades i el nombre de conjunts, la interpretació de la visualització pot tornar-se més difícil per a l'usuari.

Segons Upsetapp, "UpSet funciona mejor para datos de conjuntos con **más de tres y menos de unos 30** conjuntos. Para menos de 4 conjuntos, un diagrama de Venn proporcional al área podría ser la mejor opción, ya que son inmediatamente familiares para todos." <https://upset.app/>

```
if (!require('readr')) install.packages("readr")
```

```
## Loading required package: readr
if (!require('data.table')) install.packages("data.table")
## Loading required package: data.table
if (!require('robustbase')) install.packages("robustbase")
## Loading required package: robustbase
if (!require('ggplot2')) install.packages("ggplot2")
## Loading required package: ggplot2
if (!require('knitr')) install.packages("knitr")
## Loading required package: knitr
if (!require('pROC')) install.packages("pROC")
## Loading required package: pROC
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

if (!require('openxlsx'))install.packages("openxlsx")
## Loading required package: openxlsx
if (!require('ggforce')) install.packages("ggforce")
## Loading required package: ggforce
```

Llegeix les dades del fitxer “africa.xlsx” i les guarda en una variable anomenada `datos` (dataframe), que es la variable amb la que treballarem en endavant.

El fitxer ha sigut extret dels recursos de l'assignatura plantejats a l'enunciat <https://data.world/datasets/energy> -> <https://data.world/afdb/7e7a0724-2959-4239-aa0d-6d5bf61f07f9>

5. Visualitzacions

```
library(openxlsx)
datos <- read.xlsx("africa.xlsx")
```

5.1. Gràfic de línies

```
# Cargar Las Librerías necesarias
library(ggplot2)
```

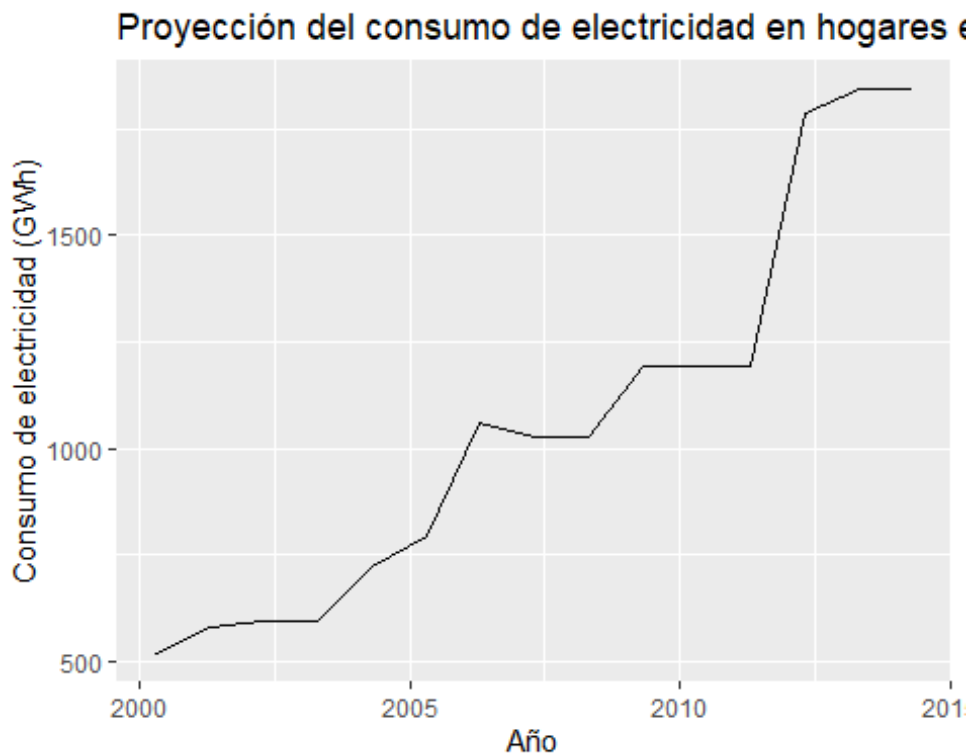
```

# Filtrar los datos para obtener solo las observaciones de Ethiopia y el
# indicador específico
datos_ethiopia <- subset(datos, RegionName == "Ethiopia" & IndicatorName
== "Consumption of electricity in Households ( GWh )")

# Convertir el campo Date a un formato de fecha
datos_ethiopia$Date <- as.Date(as.character(datos_ethiopia$Date), "%Y")

# Crear el gráfico de línea
ggplot(data = datos_ethiopia, aes(x = Date, y = Value)) +
  geom_line() +
  labs(x = "Año", y = "Consumo de electricidad (GWh)", title =
"Proyección del consumo de electricidad en hogares en Etiopía (2000-
2014)")

```



5.2. Gauge diagram

```

# Cargar las librerías necesarias
library(ggplot2)
library(ggforce)

# Filtrar los datos para obtener solo las observaciones de Etiopía y el
# indicador específico
datos_ethiopia <- subset(datos, RegionName == "Ethiopia" & IndicatorName
== "Consumption of electricity in Households ( GWh )")

# Convertir el campo Date a un formato de fecha

```

```

datos_ethiopia$Date <- as.Date(as.character(datos_ethiopia$Date), "%Y")

# Establecer el rango mínimo y máximo para el consumo de electricidad
rango_min <- min(datos_ethiopia$Value)
rango_max <- max(datos_ethiopia$Value)

# Fijar el valor de consumo
valor_consumo <- 800

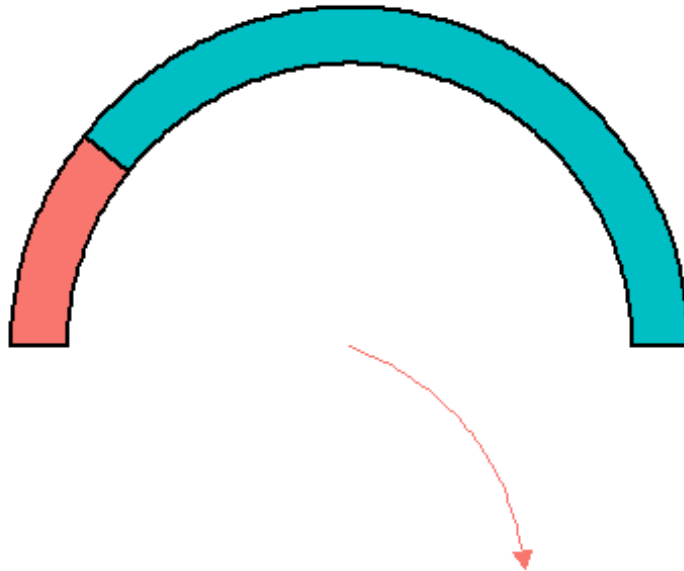
# Crear el gráfico de medidores con el valor de consumo fijo
ggplot() +
  geom_arc_bar(data = datos_ethiopia, aes(x0 = 0.5, y0 = 0.5, r0 = 0.5, r
= 0.6,
      start = -pi / 2, end = pi / 2,
      fill = factor("Energía total")),
      color = "black", size = 1) +
  geom_arc_bar(data = datos_ethiopia, aes(x0 = 0.5, y0 = 0.5, r0 = 0.5, r
= 0.6,
      start = -pi / 2, end = -pi / 2 + (pi * (valor_consumo
- rango_min) / (rango_max - rango_min)),
      fill = factor("Consumo de electricidad (GWh)")),
      color = "black", size = 1) +
  geom_text(data = datos_ethiopia, aes(x = 0.5 + 0.55 * cos(-pi / 2 + (pi
* (valor_consumo - rango_min) / (rango_max - rango_min))),
      y = 0.5 + 0.55 * sin(-pi / 2 + (pi * (valor_consumo -
rango_min) / (rango_max - rango_min))),
      label = paste("Valor de consumo:", valor_consumo,
"GWh")),
      color = "black", size = 3, fontface = "bold", hjust = 0.5,
vjust = 0.5) +
  geom_curve(data = datos_ethiopia, aes(x = 0.5, y = 0.5, xend = 0.5 +
0.5 * cos(-pi / 2 + (pi * (valor_consumo - rango_min) / (rango_max -
rango_min))),
      yend = 0.5 + 0.5 * sin(-pi / 2 + (pi * (valor_consumo -
rango_min) / (rango_max - rango_min))),
      color = factor("Consumo de electricidad (GWh)")),
      arrow = arrow(type = "closed", length = unit(0.1,
"inches")), curvature = -0.3) +
  coord_fixed() +
  theme_void() +
  theme(legend.position = "none") +
  labs(title = "Consumo de electricidad en Etiopía",
      subtitle = "",
      caption = paste("Rango mínimo:", rango_min, "GWh | Rango máximo:",
rango_max, "GWh"))

## 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.
```

Consumo de electricidad en Etiopía



Valor de consumo: 800 GWh

Rango mínimo: 518.9 GWh | Rango máximo: 1843.27113494726 GWh

5.3. UpSet: Visualizing Intersecting Sets

```
# Seleccionar aleatoriamente cinco RegionName diferentes
```

```
region <- c("Madagascar", "Egypt", "Central Africa", "Niger", "Africa",  
"Morocco")
```

```
# Filtrar los datos originales para seleccionar solo las observaciones  
del año 2014 y para las regiones aleatorias
```

```
datos_2014_region <- subset(datos, Date == 2014 & RegionName %in% region  
& IndicatorName == "Consumption of electricity in industry, GWh")
```

```
valor <- datos_2014_region$Value
```

```
valor
```

```
## [1] 234.0060 246.5244 9477.2143 11779.0161 38445.1129  
243786.1527
```

```
datos_2014_regionv2 <- subset(datos, Date == 2014 & RegionName %in%  
region & IndicatorName == "Consumption of electricity in Service Public (  
GWh )")
```

```
valor1 <- datos_2014_regionv2$Value
```

```
valor1
```

```

## [1] 109916.96152 34098.92069 4857.88267 1147.42051 158.41838
## [6] 73.74721

datos_2014_regionv3 <- subset(datos, Date == 2014 & RegionName %in%
region & IndicatorName == "Consumption of electricity in Households ( GWh
)")

valor2 <- datos_2014_regionv3$Value
valor2

## [1] 498.4065 526.9125 6462.6913 8910.0664 59617.6412
174022.7996

# Obtener el campo Value y guardarlo en una variable llamada valor

# Categorizar binariamente el consumo
consumo_categoria <- cut(valor,
breaks = c(-Inf, 3000, Inf),
labels = c("0", "1"),
right = FALSE)
consumo_categoria1 <- cut(valor1,
breaks = c(-Inf, 3000, Inf),
labels = c("0", "1"),
right = FALSE)
consumo_categoria2 <- cut(valor2,
breaks = c(-Inf, 9000, Inf),
labels = c("0", "1"),
right = FALSE)

consumo_por_electricidad <- as.character(consumo_categoria)
consumo_por_Service <- as.character(consumo_categoria1)
consumo_por_electricidad_hogares <- as.character(consumo_categoria2)

consumo_por_electricidad <- as.numeric(consumo_por_electricidad)
consumo_por_Service <- as.numeric(consumo_por_Service)
consumo_por_electricidad_hogares <-
as.numeric(consumo_por_electricidad_hogares)
consumo_por_electricidad

## [1] 0 0 1 1 1 1

consumo_por_Service

## [1] 1 1 1 0 0 0

consumo_por_electricidad_hogares

## [1] 0 0 0 0 1 1

# Instalar y cargar la librería UpSetR si no está instalada
if (!requireNamespace("UpSetR", quietly = TRUE)) {

```



```

install.packages("UpSetR")
}
library(UpSetR)

# Crear el dataframe de datos
data <- data.frame(
  region <- c("Madagascar", "Egypt", "Central Africa", "Niger", "Africa",
"Morocco"),
  Consumo_Electricidad = consumo_por_electricidad, # Categorías de
consumo de electricidad
  Consumo_Service = consumo_por_Service, # Categorías de consumo de
carbón
  Consumo_Electricidad_Hogares = consumo_por_electricidad_hogares #
Categorías de consumo de electricidad en hogares
)

# Crear el gráfico UpSet
upset(data[, c("Consumo_Electricidad", "Consumo_Service",
"Consumo_Electricidad_Hogares")], sets = names(data)[-1], main.bar.color
= "#56B4E9", matrix.color="black")

```

