Diagramas Aluviales o Sandkey con R-CRAN

Maestría en Energía

PALMA Ricardo R. - Facultad de Ingeniería
<rpalma@uncu.edu.ar>
MASERA Gustavo A. - Facutad de Filosofía y Letras
<gustavo.masera@fing.uncu.edu.ar>
Universidad Nacional de Cuyo

August 24, 2021

1 ¿Que es un diagrama aluvial?

Alluvial diagramo Diagramas Aluviales en español es una variante del ploteo de coordenadas paralelas de variables categóricas. Fue muy utilizado por Napoleón Bonaparte para matematizar modelos de guerra planteados sobre mesas de arena. Es por ello que también se los conoce como Sandkey (o caja de arena). Las Varaibles se asignan a unos ejes verticales paralelos. Los valores se represetntan como bloques de ancho varaible proporcional al balance en el eje *y*.

Es muy sencillo crearlos con el comando alluvial() desde R-Cran. Este es un ejemplo en el que se muestra el caso de Freedonia y su estrategia de gestión de la energía. El modelo matemático sobre el que aluvial opera es el siguiente, pero no te preocupes, R-Cran se encarga de hacer todo el trabajo matemático por ti.

.

$$\frac{\partial(F_1,F_2)}{\partial(c,\omega)}_{(c_0,\omega_0)} = \left| \begin{array}{cc} \frac{\partial F_1}{\partial c} & \frac{\partial F_1}{\partial \omega} \\ \frac{\partial F_2}{\partial c} & \frac{\partial F_2}{\partial \omega} \end{array} \right|_{(c_0,\omega_0)}$$

$$= -4c_0 q\omega_0 - 4c_0\omega_0 p^2 = -4c_0\omega_0(q+p^2) > 0.$$

.

El programa y su formulación está validado por Se4All UNO Sustainable Energy for All program de Naciones Unidas

Ver: https://sustainabledevelopment.un.org/partnership/?p=10133

Invocando una biblioteca

Tengamos en cuenta esto al escribir en español

```
> library(readr)
> # APPncia <- read_delim("c:/agencia_leap.csv",
> #";", escape_double = FALSE, trim_ws = TRUE)
> # En caso de que tengas la base de datos en tu disco.
>
> #APPncia <- read.csv("~/agencia_leap.csv", sep=";")
> # En caso de tener la base de datos en tu espacio en la nube
```

```
>
> APPncia <- read_delim("https://themys.sid.uncu.edu.ar/rpalma/R-cran/Energia/agencia_leap
> # Obtener de la Web
> #APPncia <- read_delim(
> #"https://themys.sid.uncu.edu.ar
> #/rpalma/R-cran/Energia/agencia_leap.csv",
> # ";", escape_double = FALSE, trim_ws = TRUE)
>
>
>
>
> #APPncia <- read_delim("agencia_leap.csv",
> #                      ";", escape_double = FALSE, trim_ws = TRUE)
>
> tit <- as.data.frame(APPncia, stringsAsFactors = TRUE)
> head(tit)

    Fuente Sostenib.   Zona              Dest. Habitantes Ktoep
1   Carbon  Primaria Urbano           Perdidas         40    10
2 Petróleo  Primaria  Rural    A_Acondicionado         60     5
3      Gas  Generado Urbano  Cocina y Heladera         24    10
4   Eolica Renovable  Rural        Iluminación         18     5
5  Biomasa  Generado Urbano  Cocina y Heladera          6    10
6    Solar Renovable  Rural Arado y Transporte         10     5
```

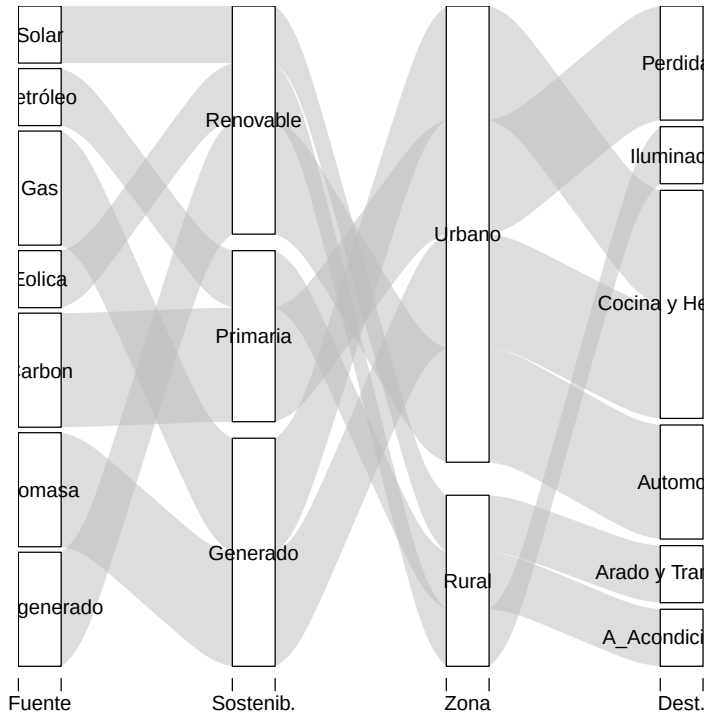
1.1 Creando el Diagrama

Procederemos a invocar la biblioteca alluvial

Este es un ejemplo muy cargado de atributos, no mires del código los descriptores gráficos y de color de los objetos. El objetivo es que se pueda ver que esperamos de esta biblioteca.

Uso más simple: El ancho de las barras representa Kilo Toneladas Equivalentes de Petróleo

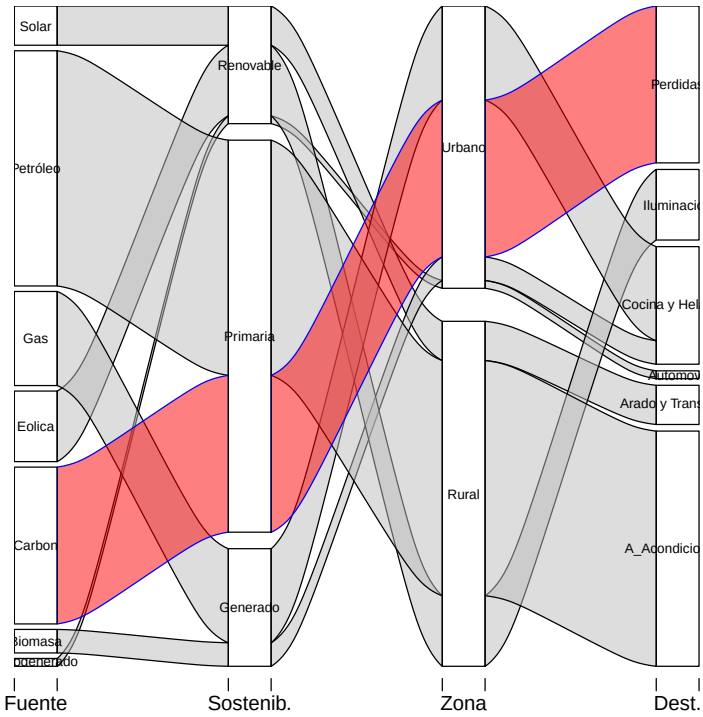
```
> library(alluvial)
> alluvial(tit[ ,1:4], freq=tit$Ktoep)
>
```



Uso Avanzado, marcar un camino con color

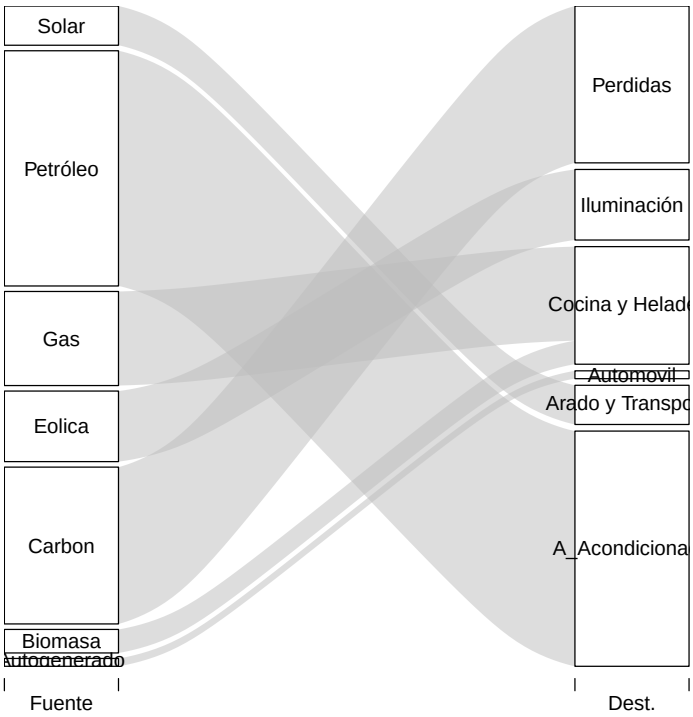
Este caso es para poner de manifiesto un aspecto que nos interese destacar. En este caso tanto Petróleo como Carbón están en el sector que ha sido financiado para obtener sostenibilidad en las energías primarias, pero a pesar de ello el uso posterior en las zonas urbanas hacen que se transformen en pérdidas.

```
> alluvial(tit[ ,1:4], freq = tit$Habitantes,
+          col = ifelse(tit$Dest. == "Perdidas", "red", "grey"),
+          border = ifelse(tit$Dest. == "Perdidas", "blue", "black"),
+          hide = tit$Habitantes == 0,
+          cex = 0.7
+ )
```



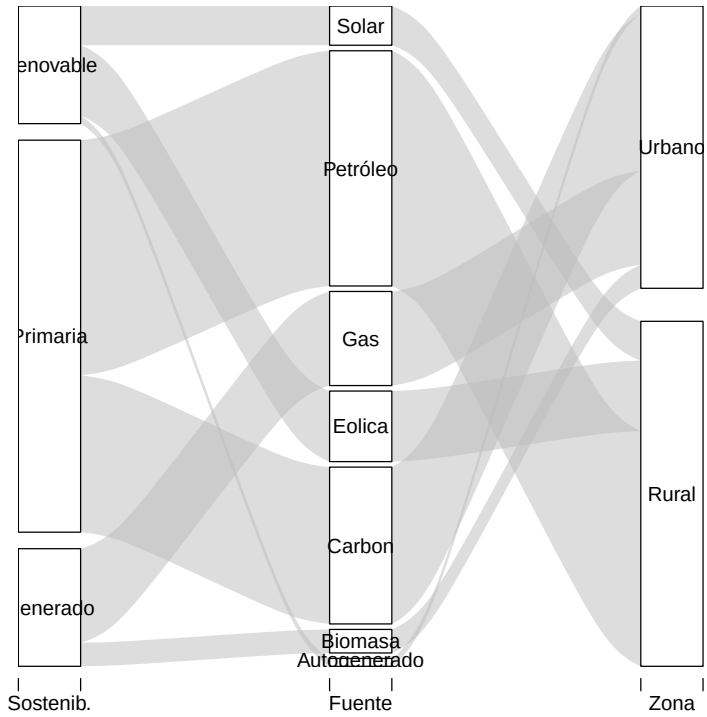
Ya hemos armado el dataframe y acomodado los datos para que puedan ser impresos veremos el más simple de los diagramas que podemos hacer. Por defecto alluvial nos imprme los graficos en color gris y usa transparencias.

```
> # Survival status and Class
> library(dplyr)
> #esta biblioteca habilita la funcion %>%
> # Survival status and Class
> tit %>% group_by(Fuente, Dest.) %>%
+   summarise(n = sum(Habitantes)) -> tit2d
> alluvial(tit2d[,1:2], freq=tit2d$n)
>
```



En este caso utilizamos a la salida las categorías "Urbano" / "Rural" con sólo dos niveles de filtro. Este gráfico aparece en el manula de LEAP y está tomado del caso de Costa Rica y el préstamo del BID para su proyecto nacional de Ciudades Inteligentes e Industrial 4.0 de 2019.

```
> # Survival status, Sex, and Class
> tit %>% group_by(Sostenib., Fuente, Zona) %>%
+   summarise(n = sum(Habitantes)) -> tit3d
> alluvial(tit3d[,1:3], freq=tit3d$n)
```

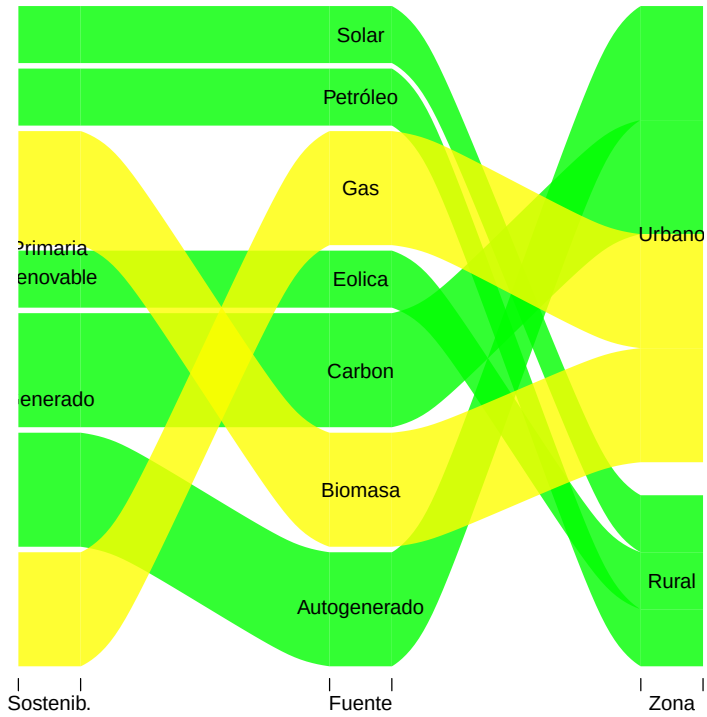


En este ejemplo hemos realizado el mismo ejemplo anterior pero en tres niveles. Hemos clasificado por Zona antes de las categorías anteriores, pero el ancho representa las Ktoep.

```
> tit %>% group_by(Sostenib., Fuente, Zona) %>%
+   summarise( n= sum(Habitantes)) -> x
> tit %>% group_by(Sostenib., Fuente, Zona) %>%
+   summarise( n= sum(Ktoep)) -> y
```

Para graficar en orden inverso es necesario que consideremos armar nuevas categorías x e y que nos permitan parte de lo que antes era nuestro punto de llegada.

```
> alluvial(y[,1:3], freq=y$n,
+    # col = RColorBrewer::brewer.pal(8, "Set1"),
+          col = ifelse(y$Sostenib. == "Generado", "Yellow", "Green"),
+          alpha = 0.8,
+          blocks = FALSE,
+          ordering = list(
+            order(y$Fuente, y$Zona == "Red"),
+            order(y$Fuente, y$Zona == "Blue"),
+            NULL
+          )
+ )
>
```


Solar
Petróleo
Gas
rimaria
enovable
Eolica
Carbon
enerado
Biomasa
Autogenerado
Urbano
Rural
Sostenib.
Fuente
Zona