

Diagramas Aluviales *Sandkey* con R-CRAN

Conocimiento de Campos Disciplinares

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

December 12, 2018

1 ¿Que es un diagrama aluvial?

Alluvial diagram is a variant of a Parallel Coordinates Plot (PCP) but for categorical variables. Variables are assigned to vertical axes that are parallel. Values are represented with blocks on each axis. Observations are represented with alluvia (sing. "alluvium") spanning across all the axes.

You create alluvial diagrams with function `alluvial()`. This is one example using Titanic dataset. Let's convert it to a data frame.

```
> setwd("/media/rpalma/OS/AAA_Datos/2018/Posgrado/Doctorado Hipotesis a la Tesis/R-CRAN/Al  
> APPncia <- read.csv2("Residuos_Agricolas.csv")  
> tit <- as.data.frame(APPncia, stringsAsFactors = FALSE)  
> head(tit)
```

	Residuo	Finan	Medio	Exito	Freq	Monto	TIR
1	Leguminosas	CyT_SN	PyME_ANR	No	0	5.045360	0.2535910
2	Frutas	CyT_SN	PyME_ANR	No	0	47.610275	0.2078233
3	Vinificacion	CyT_SN	PyME_ANR	No	35	5.644619	0.2672615
4	Destilacion	CyT_SN	PyME_ANR	No	0	16.222262	0.2214048
5	Leguminosas	BID	PyME_ANR	No	0	33.161698	0.2447643
6	Frutas	BID	PyME_ANR	No	0	29.141628	0.2986382

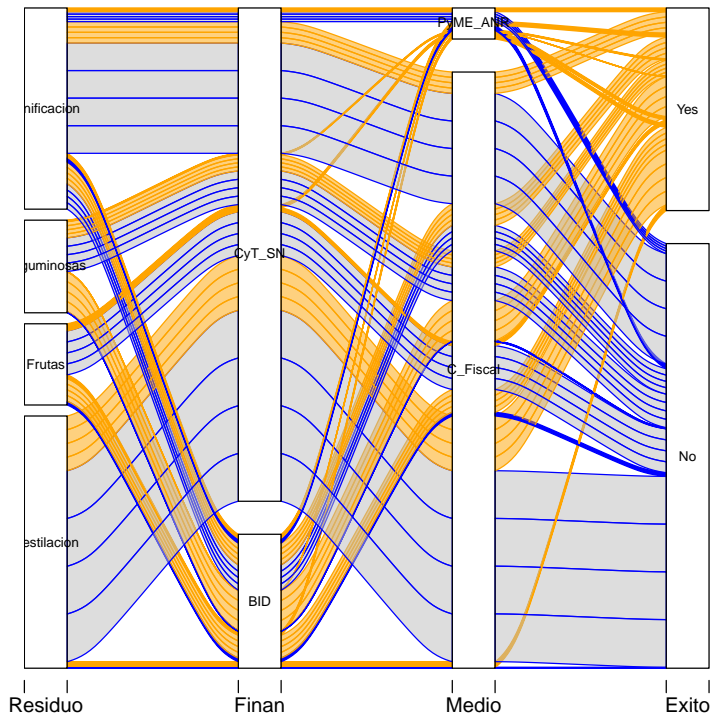
1.1 Creando el Diagrama

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.

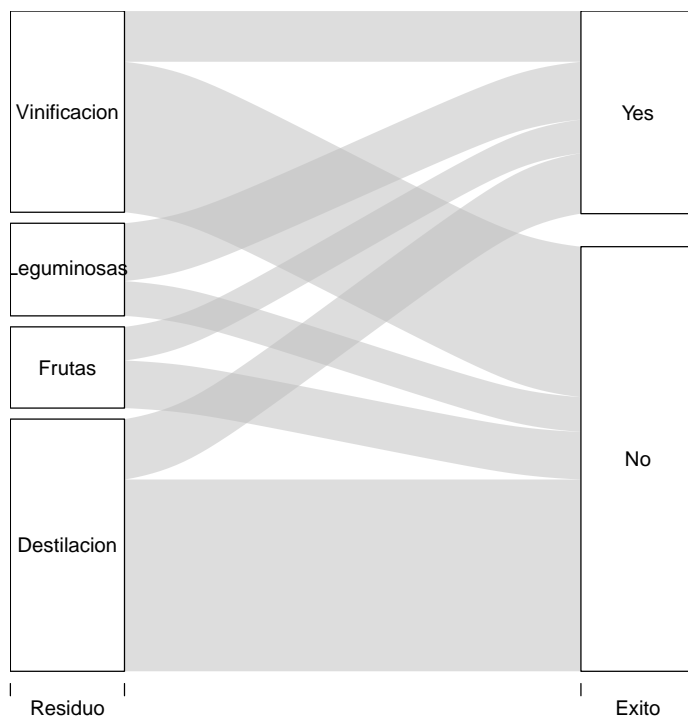
```
> library(alluvial)  
> alluvial(tit[,1:4], freq=tit$Freq,  
+         col = ifelse(tit$Exito == "Yes", "orange", "grey"),  
+         border = ifelse(tit$Exito == "Yes", "orange", "blue"),  
+         hide = tit$Freq == 0,
```

```
+      cex = 0.7
+ )
```



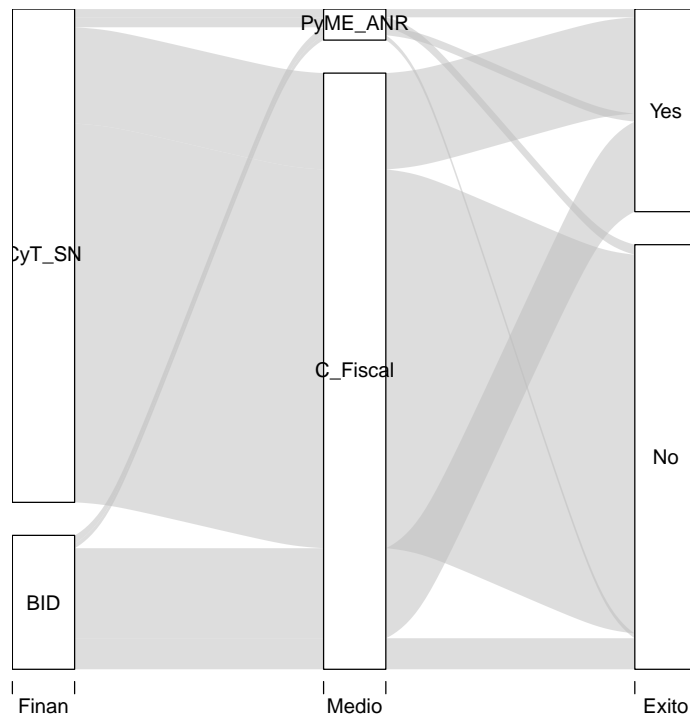
Ya hemos armado el dataframe y acomodado los datos para que puedan ser impresos veremos el más simple de los diagramas que podemos hacer. Por defecto alluvial nos imprime los graficos en color gris y usa transparencias.

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



En este caso utilizamos a la salida las categorías "Si Sobrevive" / "No Sobrevive" con sólo dos niveles de filtro.

```
> # Survival status, Sex, and Class
> tit %>% group_by(Finan, Medio, Exito) %>%
+ summarise(n = sum(Freq)) -> tit3d
> alluvial(tit3d[,1:3], freq=tit3d$n)
```

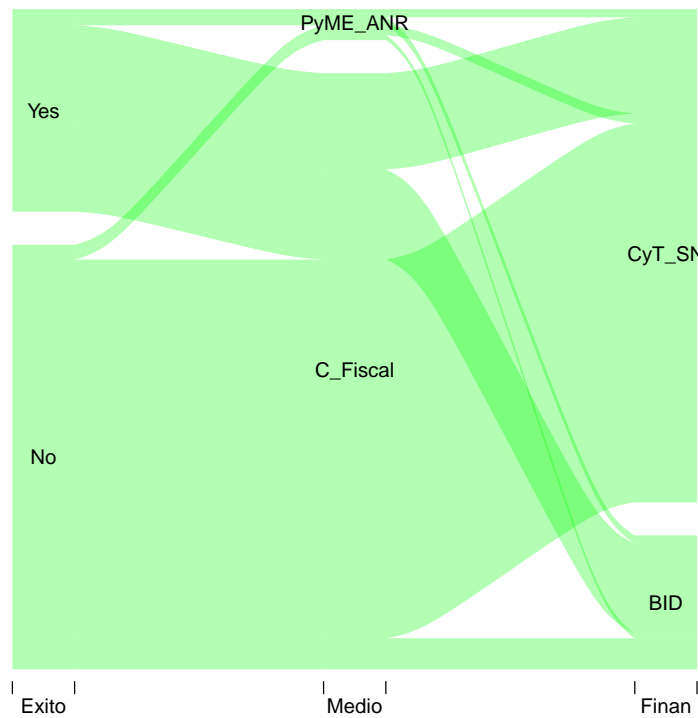


En este ejemplo hemos realizado el mismo ejemplo anterior pero en tres niveles. Hemos clasificado por sexo antes de las categorías anteriores.

```
> tit %>% group_by(Exito, Medio, Finan) %>%
+ summarise( n= sum(Freq)) -> x
> tit %>% group_by(Exito, Medio, Finan) %>%
+ summarise( n= sum(Freq)) -> y
```

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

```
> alluvial(y[,1:3], freq=y$n,
+ # col = RColorBrewer::brewer.pal(8, "Set1"),
+ col = ifelse(y$Finan == "PyME_ANR", "blue", "green"),
+ alpha = 0.3,
+ blocks = FALSE,
+ ordering = list(
+ order(y$Exito, y$Finan == "PyME_ANR"),
+ order(y$Medio, y$Finan == "PyME_ANR"),
+ NULL
+ )
+ )
>
```



2 Casos de Éxito y Fracaso

Determinación de VAN casos Exitosos

VAN de los casos exitosos en Millones

```
> casosexito <- APPncia[APPncia$Exito=="Yes",]
> head(casosexito)
```

	Residuo	Finan	Medio	Exito	Freq	Monto	TIR
17	Leguminosas	CyT_SN	PyME_ANR	Yes	5	35.52642	0.2872053
18	Frutas	CyT_SN	PyME_ANR	Yes	11	47.15262	0.2145507
19	Vinificacion	CyT_SN	PyME_ANR	Yes	13	16.64807	0.2634952
20	Destilacion	CyT_SN	PyME_ANR	Yes	0	26.83331	0.2195922
21	Leguminosas	BID	PyME_ANR	Yes	1	19.34117	0.2543593
22	Frutas	BID	PyME_ANR	Yes	13	22.60184	0.2450026

```
> mean(casosexito$Monto)
```

```
[1] 26.65392
```

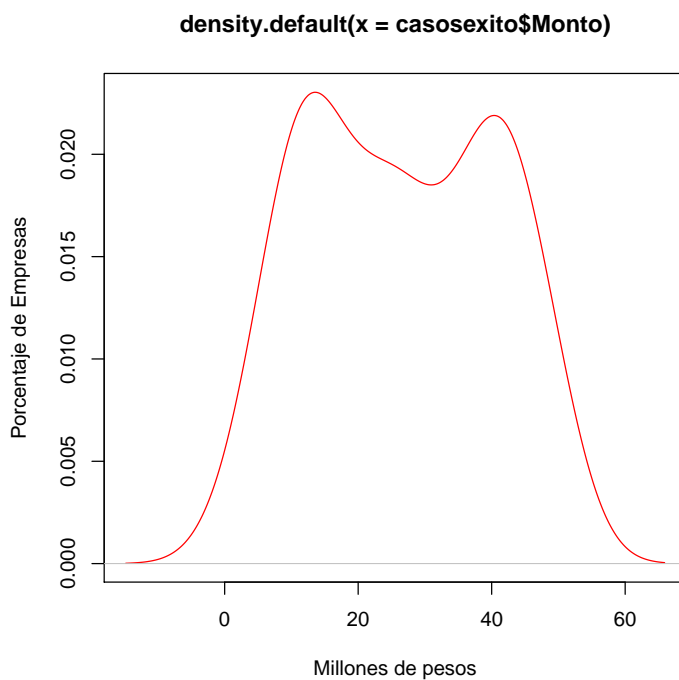
```
> sd(casosexito$Monto)
```

```
[1] 13.94772
```

```
> var(casosexito$Monto)
```

```
[1] 194.5388
```

```
> graf_1 <- density(casosexito$Monto)
> plot(graf_1, xlab="Millones de pesos", ylab="Porcentaje de Empresas", col="red")
>
```



VAN de los casos de fracaso

```
> casosfracaso <- APPncia[APPncia$Exito=="No",]
> head(casosfracaso)
```

	Residuo	Finan	Medio	Exito	Freq	Monto	TIR
1	Leguminosas	CyT_SN	PyME_ANR	No	0	5.045360	0.2535910
2	Frutas	CyT_SN	PyME_ANR	No	0	47.610275	0.2078233
3	Vinificacion	CyT_SN	PyME_ANR	No	35	5.644619	0.2672615
4	Destilacion	CyT_SN	PyME_ANR	No	0	16.222262	0.2214048
5	Leguminosas	BID	PyME_ANR	No	0	33.161698	0.2447643
6	Frutas	BID	PyME_ANR	No	0	29.141628	0.2986382

```
> mean(casosfracaso$Monto)
```

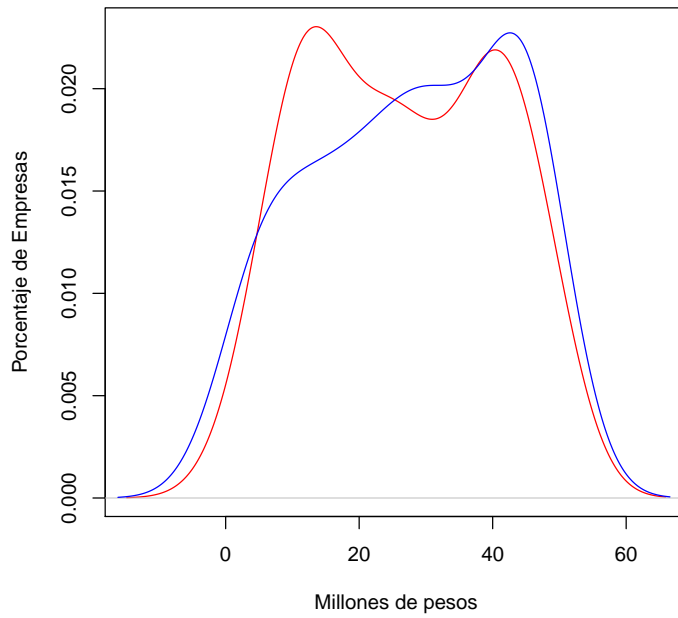
```
[1] 27.93163
```

```
> sd(casosfracaso$Monto)
```

```
[1] 14.7574
```

```
> plot(graf_1, xlab="Millones de pesos", main="Rojo Exito - Azul Fracaso", ylab="Porcentaj
> lines(density(casosfracaso$Monto), col="blue")
>
```

Rojo Exito – Azul Fracaso



Ejercicios Propuestos

* Realizar las mismas gráficas con el TIR * Analizar la comparación con ROI (TIR * VAN)