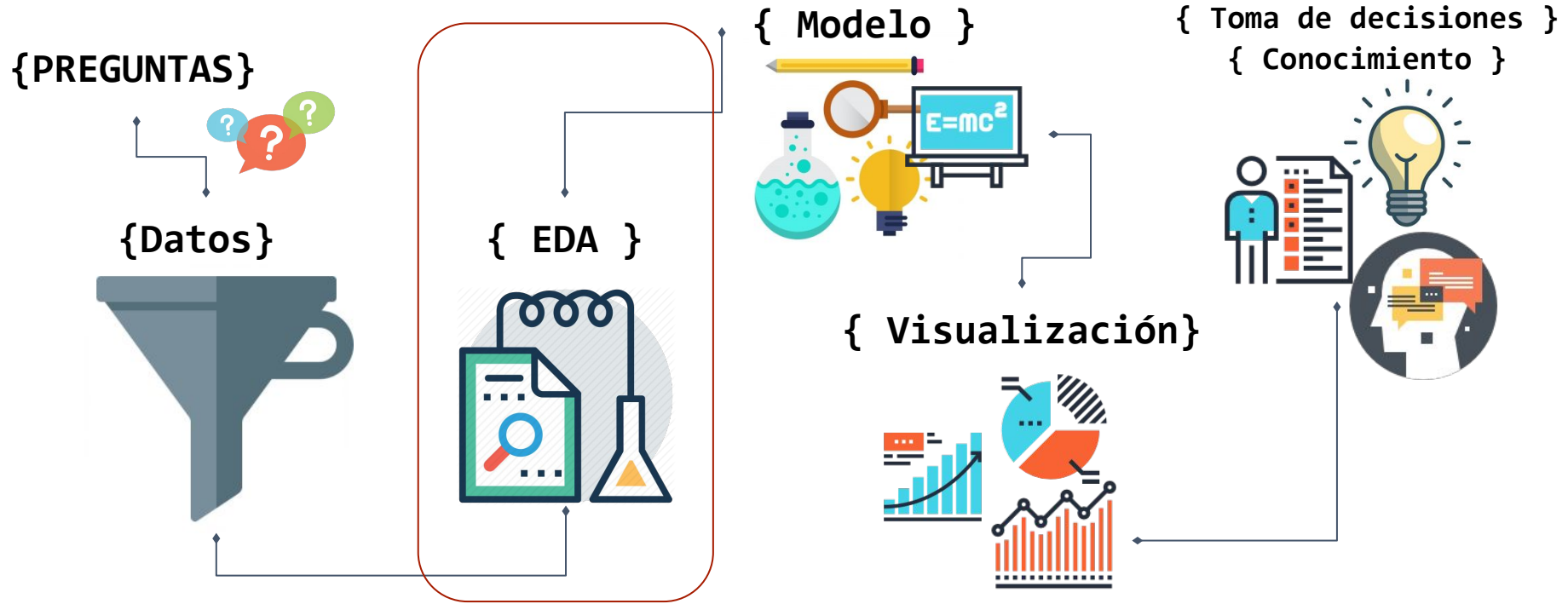


{ TidyVerse II}

PROCESO DE ANÁLISIS



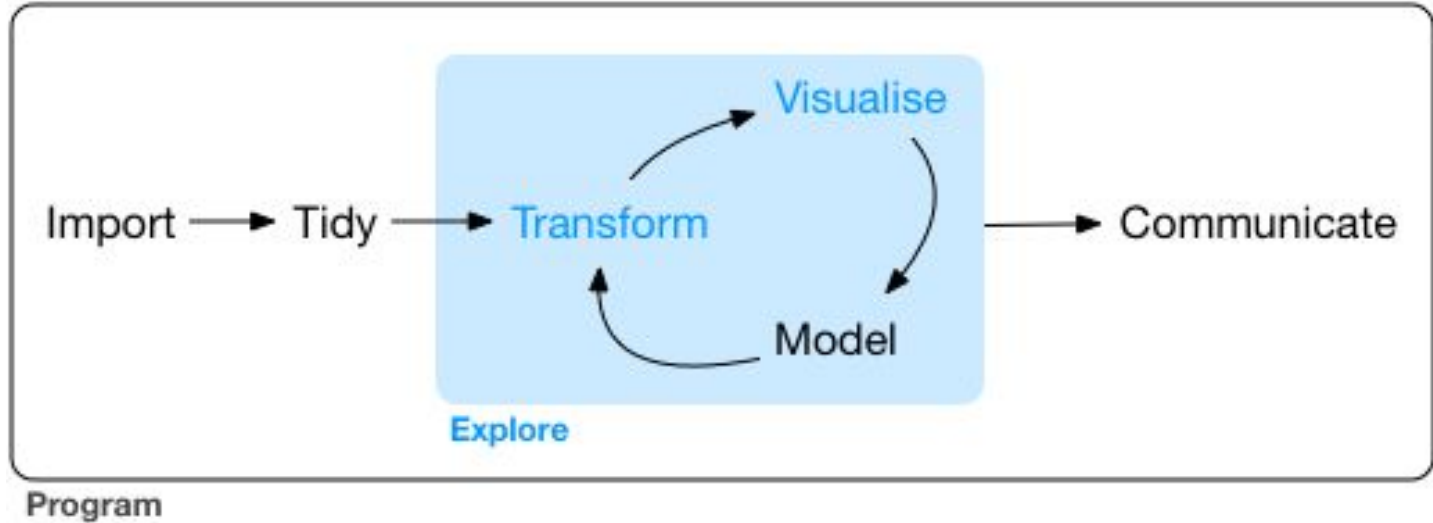
Instalación de paquetes

Para esta clase necesitamos instalar dos paquetes:

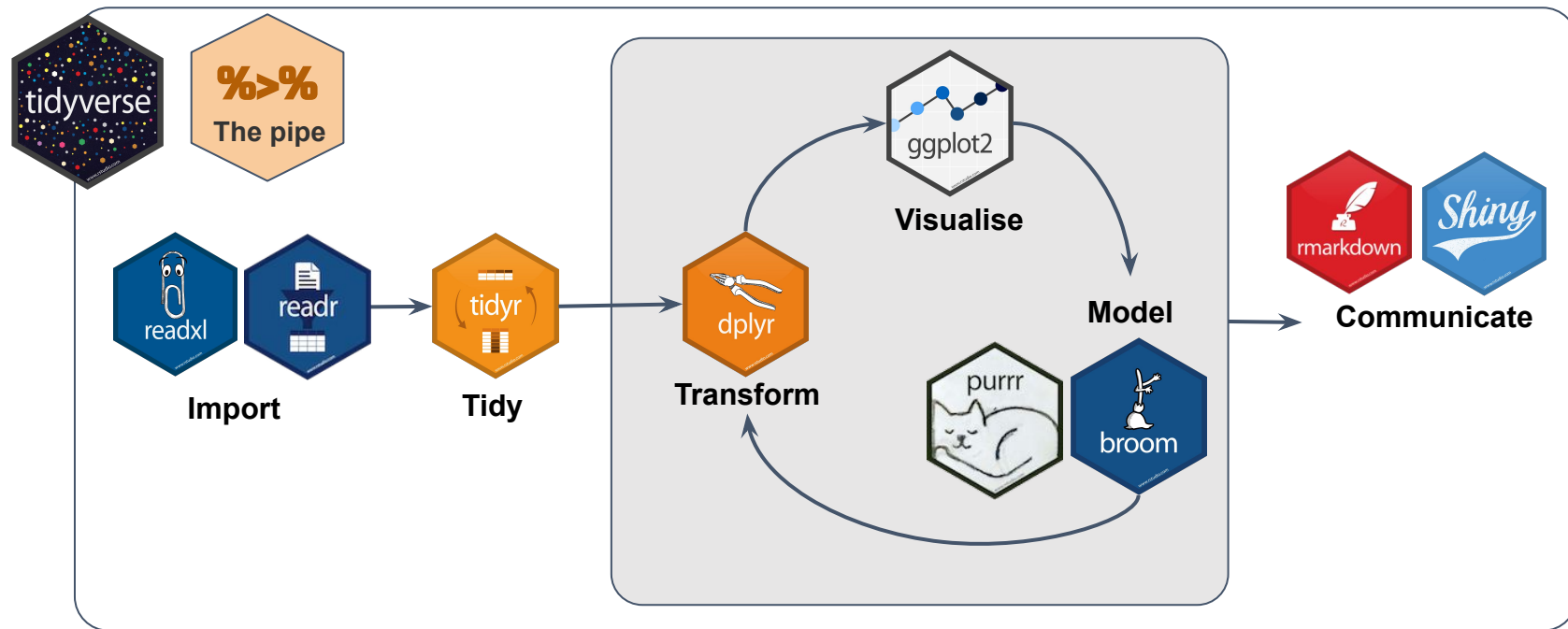
```
install.packages('tidyverse')  
install.packages('reshape2')  
install.packages('nycflights13')
```



QUE PUEDO HACER CON R?



Tidyverse: introducción y paquetes



Función Summarise

La función summarise permite agrupar los datos utilizando una función de agregación.

	sex	esViernes	Prom	count
1	Female	FALSE	18.50821	78
2	Female	TRUE	14.14556	9
3	Male	FALSE	20.80442	147
4	Male	TRUE	19.85700	10

tips %>%

```
group_by(sex,esViernes=day=='Fri') %>%  
summarise(Prom=mean(total_bill),  
           Cant=n())
```



Función Summarise

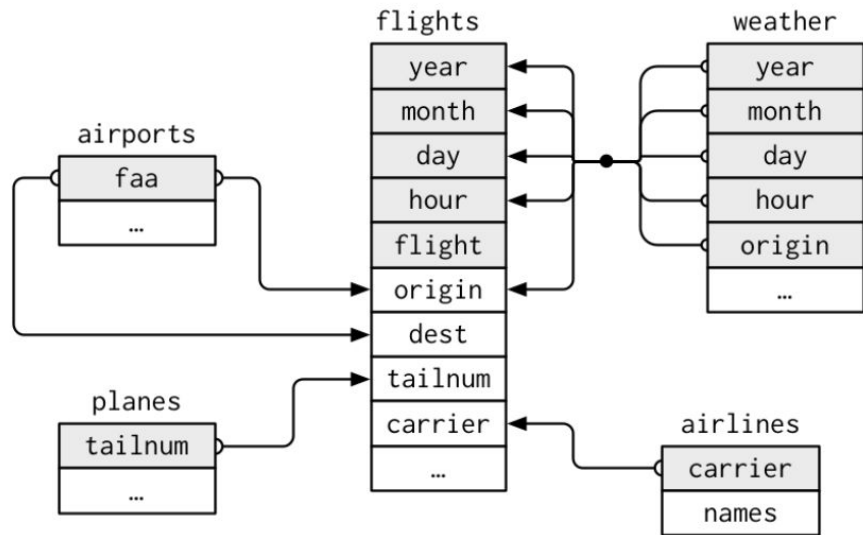
Funciones que se pueden aplicar	
<code>min(), max()</code>	Valores max y min
<code>mean()</code>	media
<code>median()</code>	mediana
<code>sum()</code>	suma de los valores
<code>var, sd()</code>	varianza y desviación típica
<code>n()</code>	Es el número de valores de un vector



Funciones Join

En este gráfico se visualizan las relaciones entre los dataset del paquete nycflights13.

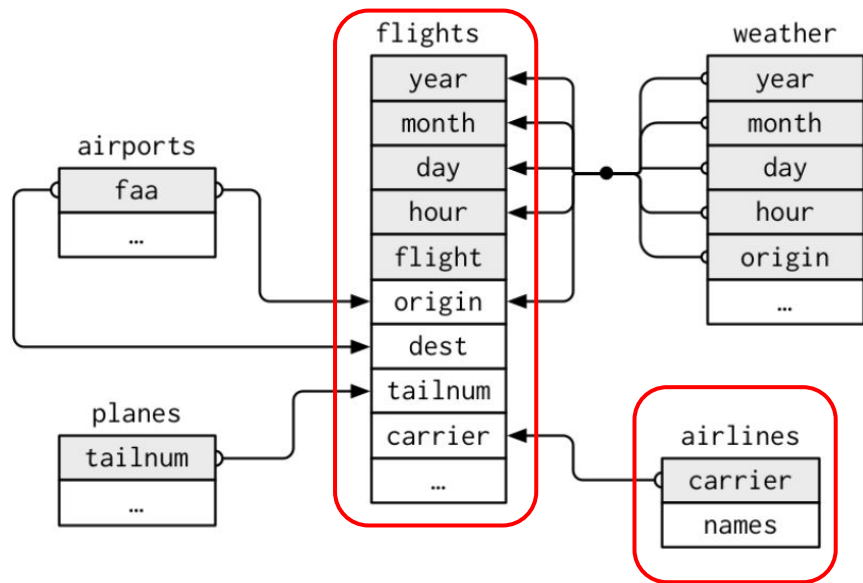
Estos dataset corresponden a datos de puntualidad de la aerolíneas para todos los vuelos que salen de Nueva York en 2013.



Función Inner_Join

Para realizar un inner join entre flights y airlines, teniendo en cuenta que la unión tiene las columnas con igual denominación.

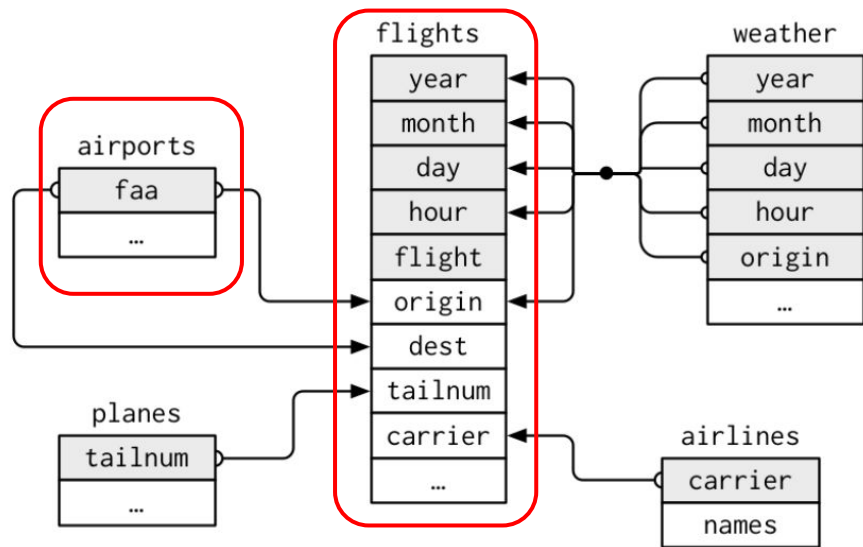
```
flights %>%  
inner_join(aerolineas,  
           by='carrier')
```



Función Inner_Join

Para realizar un inner join entre flights y airports, y esta union tiene las columnas con diferente denominación.

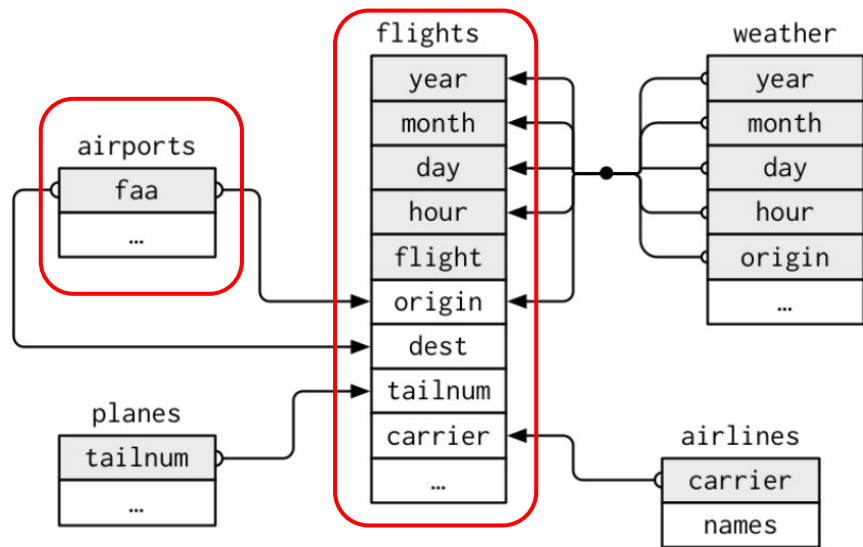
```
flights %>%  
inner_join(airports,  
           by=c('origin'='faa'))
```



Función Inner_Join

Para realizar un inner join entre flights y airports, y esta union tiene las columnas con diferente denominación.

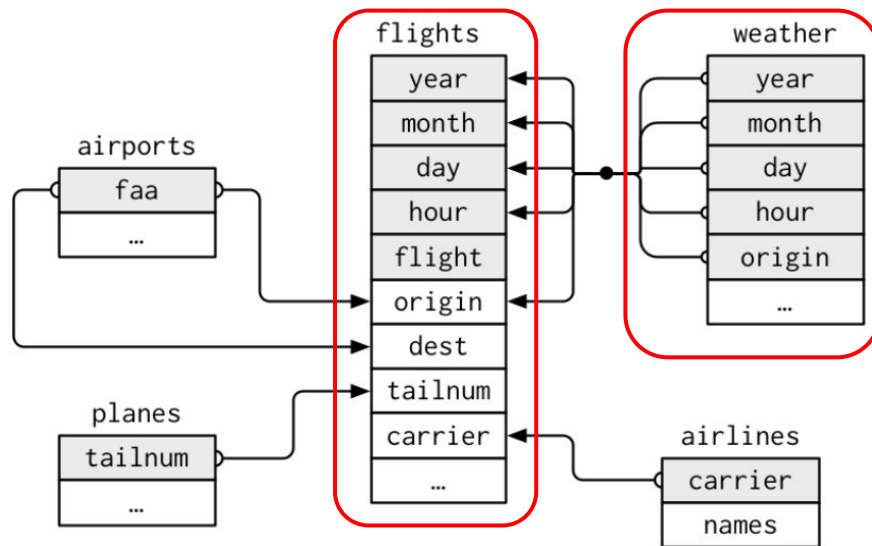
```
flights %>%  
inner_join(airports,  
           by=c('origin'='faa'))
```



Función left_Join

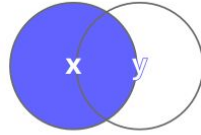
Para realizar un left join entre flights y airlines:

```
flights %>%  
left_join(weather)
```

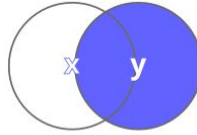


dplyr *joins*

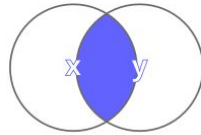
left_join(x, y)



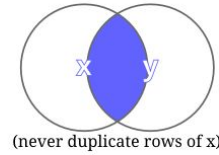
right_join(x, y)



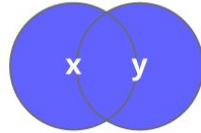
inner_join(x, y)



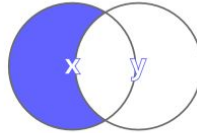
semi_join(x, y)



full_join(x, y)



anti_join(x, y)



Funcion Gather y Spread

Estas funciones corresponden a la librería “tidyr” y permiten formatear una estructura de datos.

	mes	año	destino_bioetanol	destino_otros
1	Enero	2018	127.018	21.016



	mes	año	destino	cantidad
1	Enero	2018	destino_bioetanol	127.018
2	Enero	2018	destino_otros	21.016



Funcion Gather y Spread

Gather es una función toma n columnas y las une en base a pares clave-valor. De esta manera cada fila corresponderá a una unica observacion.

```
base%>%
```

```
gather(
```

```
key='destino',
```

```
value='cantidad',
```

```
c('destino_bioetanol', 'destino_otros'))
```

	mes	año	destino_bioetanol	destino_otros
1	Enero	2018	127.018	21.016



	mes	año	destino	cantidad
1	Enero	2018	destino_bioetanol	127.018
2	Enero	2018	destino_otros	21.016



Funcion Gather y Spread

Spread es una función realiza lo opuesto a gather generando múltiples columnas en base a un par valor-clave.

```
base%>%  
  spread(  
    key='destino',  
    value='cantidad')
```

	mes	año	destino	cantidad
1	Enero	2018	destino_bioetanol	127.018
2	Enero	2018	destino_otros	21.016



	mes	año	destino_bioetanol	destino_otros
1	Enero	2018	127.018	21.016

