

Creación de gráficos: ggplot2

Sitio: [Plataforma de Formación On line del Instituto Andaluz de Administración Pública](#)
Curso: (I22F-PT05) Entorno de Programación R
Libro: Creación de gráficos: ggplot2

Imprimido por: ALFONSO LUIS MONTEJO RAEZ
Día: lunes, 18 de abril de 2022, 10:14

Tabla de contenidos

1. Introducción

2. Instalación

3. Partes de un ggplot

4. Modificaciones

4.1. Estética

4.2. Geometría

4.3. Theme

4.4. Otros

1. Introducción



ggplot2 es una librería, desarrollada por RStudio, que nos ofrece un conjunto de funciones para realizar gráficos de todo tipo usando una filosofía de capas.

La idea de este paquete es ir creando un gráfico partiendo de una base a la que se le van añadiendo modificaciones de aspecto, color, tipo de gráfico, leyenda, títulos, etc. en distintas capas.

La orden básica para la creación de un gráfico es **ggplot()** y se van añadiendo capas. Por ejemplo:

- **ggplot()+geom_bar()** para un gráfico de barras.
- **ggplot()+geom_point()** para un gráfico de dispersión.
- **ggplot()+geom_line()** para un gráfico de líneas.

2. Instalación

Ggplot2 es un paquete que hay que instalar y cargar para que funcione ya que no viene por defecto en el paquete base de R.

La instalación se puede hacer de dos formas:

- **mediante órdenes.**
- **mediante asistente.**

Mediante órdenes

Lo haríamos usando **install.packages('ggplot2', dependencies=T)** y a continuación usamos la orden **library(ggplot2)** para cargar el paquete y poder trabajar con él.

Mediante asistente

Como para cualquier paquete de R, el programa RStudio nos ofrece la posibilidad de instalar y cargar un paquete desde su asistente.

Este proceso ya lo hemos explicado anteriormente en la Unidad 1. Para más detalles ver:

Unidad 1 → Empezar a trabajar con RStudio → 3. Gestión de paquetes → 3.3. Instalación y 3.4. Carga

3. Partes de un ggplot

Con **ggplot2** podemos hacer gráficos muy complejos, donde siempre vamos a tener el control total para modificar cualquier cuestión sobre la creación y visualización del gráfico.

A la hora de realizar un gráfico con ggplot2 tenemos que tener en cuenta que hay aspectos que siempre deben estar presentes en el gráfico.

- Los **datos** con los que se van a trabajar.
- La **estética** que le vamos a dar al gráfico (color, tamaño, tipo de línea, etc.). Se determinan con la orden **aes()** dentro de ggplot().
- La **geometría** que vamos a representar (Barras, líneas, dispersión, polar, etc.). Se usan con la orden **geom_*()** y va cambiando según el tipo de gráfico elegido.

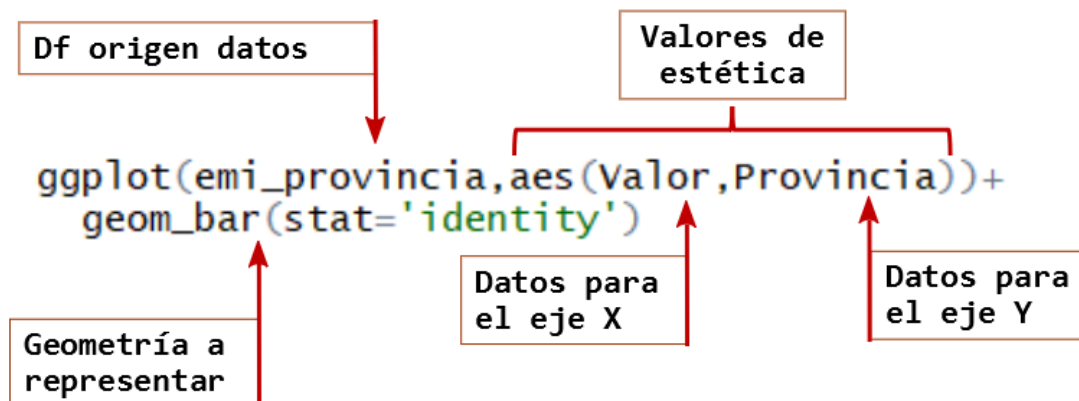
En general, si falta alguno de estos aspectos, el gráfico no se puede realizar.

Ejemplo

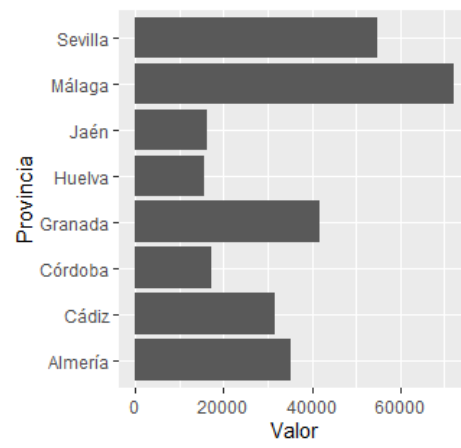
Vamos a usar los datos del fichero **emigraciones_prov.RData** para hacer gráficos. Este data frame contiene tres variables (Provincia, Sexo y Valor) con las emigraciones por provincia y sexo en Andalucía.

	Provincia	Sexo	Valor
4	Almería	Hombres	19170
5	Almería	Mujeres	15988
316	Cádiz	Hombres	16416
317	Cádiz	Mujeres	15373
451	Córdoba	Hombres	8780
452	Córdoba	Mujeres	8627
679	Granada	Hombres	21138
680	Granada	Mujeres	20489
1198	Huelva	Hombres	8021
1199	Huelva	Mujeres	7620
1438	Jaén	Hombres	8416
1439	Jaén	Mujeres	7907
1732	Málaga	Hombres	36564

Con estos datos vamos a hacer un gráfico que represente la emigración por provincias.



El resultado sería:



Hay que recordar que, en RStudio, los gráficos se muestran en la pestaña de **Plots**.

4. Modificaciones

Podemos modificar cualquier aspecto del gráfico actuando sobre cualquiera de los tres aspectos que hemos mencionado en el apartado anterior:

- Datos
- Estética
- Geometría

También podemos ir añadiendo modificaciones e ir combinando tipos de gráficos usando capas. Las capas se van definiendo "sumando" cosas al gráfico usando el operador suma (+).

Para que las capas funcionen correctamente es necesario que el + que siempre como último elemento de la línea, poniendo la siguiente capa a continuación.

Vamos a ir viendo algunos de los cambios que podemos hacer en los gráficos.

4.1. Estética

Como hemos dicho antes, podemos hacer cambios en la estética para personalizar la apariencia del gráfico.

Los cambios en la estética pueden afectar a varias partes del gráfico y necesitan parámetros que se incluyan dentro de la orden **aes()**.







Algunos de los más importantes son:

- **fill**. Para colorear el relleno.
- **color**. Para colorear las líneas.
- **linetype**. Para elegir el tipo de línea.
- **size**. Para elegir el tamaño de las líneas o de los puntos.
- **shape**. Para elegir la forma de los puntos.

Hay que tener en cuenta que no todos los parámetros se pueden usar con todos los tipos de gráficos. Por ejemplo, no tiene sentido definir el tipo de punto si lo que vamos a usar es un gráfico de líneas.

Los parámetros los podemos cambiar, o bien usando una variable del data frame (parámetros **color** y **fill**) o bien dándoles valores (parámetros **linetype**, **size** y **shape**).

Valores para el parámetro linetype

0. 'blank'	
1. 'solid'	
2. 'dashed'	
3. 'dotted'	
4. 'dotdash'	
5. 'longdash'	
6. 'twodash'	

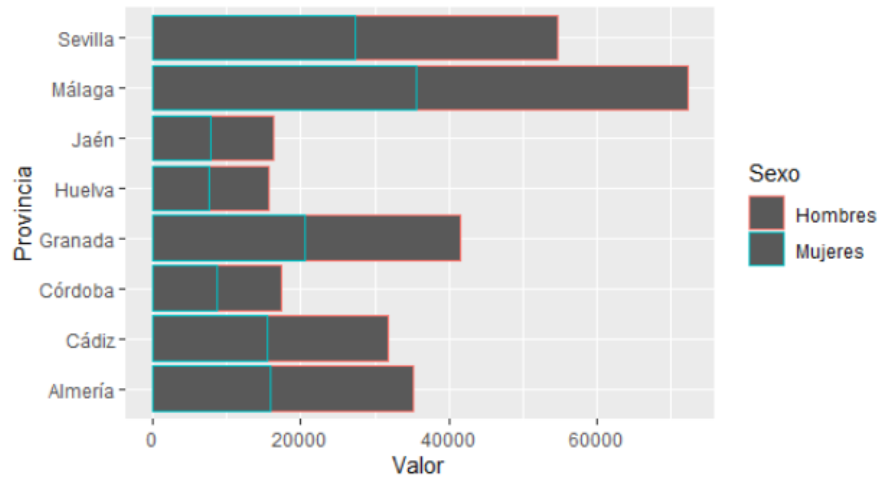
Valores para el parámetro shape

Circles	1 	10 	13 	16 	19 	20 	21 
Triangles	2 	6 	17 	24 	25 		
Diamonds	5 	9 	18 	23 			
Squares	0 	7 	12 	14 	15 	22 	
Other	3 	4 	8 	11 			

Ejemplo

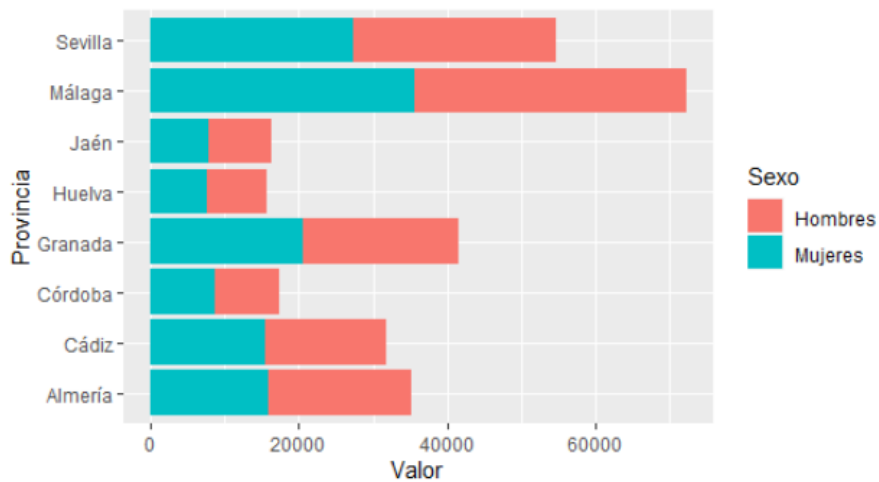
Vamos a ir cambiando algunos parámetros estéticos del gráfico anterior.


```
ggplot(emi_provincia,aes(Provincia,Valor,color=Sexo))+
  geom_bar(stat='identity')
```



Cambiamos el color exterior de la barra usando el parámetro **color** con la variable Sexo.

```
ggplot(emi_provincia,aes(Valor,Provincia,fill=Sexo))+
  geom_bar(stat='identity')
```



Aquí cambiamos el color del relleno de la barra con **fill** y la variable Sexo.

4.2. Geometría

La parte de la geometría va en consonancia con el tipo de gráfico que queremos hacer y las variables que dispongamos (dicretas, continuas).

Para la elección de la geometría a representar se usa la orden **geom_()** y su nombre va cambiando en función del tipo de gráfico.

- **geom_bar()**. Diagrama de barras.
- **geom_point()**. Diagrama de dispersión.
- **geom_line()**. Gráfico de líneas.
- **geom_histogram()**. Histograma.

Hay muchos más. Os animo a echarle un vistazo a la chuleta de ggplot2 para que os hagais una idea.

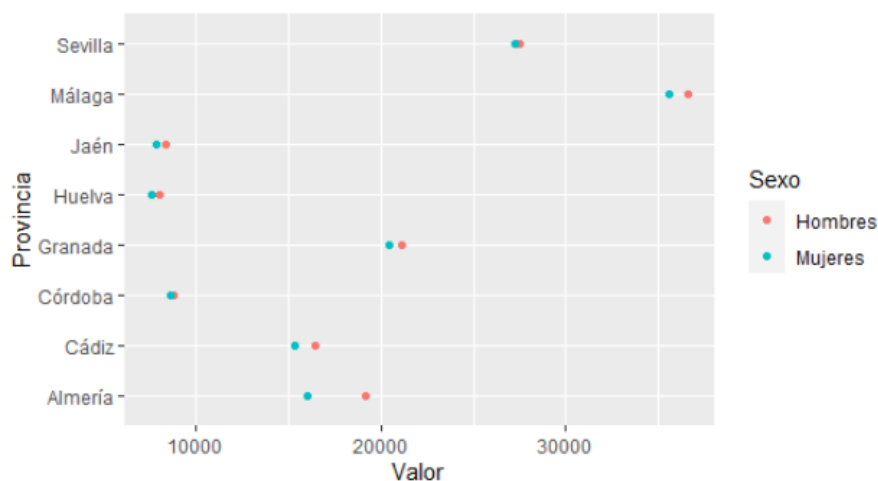
También podemos cambiar algunos aspectos dentro de la geometría como son las posiciones de algunos elementos; por ejemplo, la posición de las barras en un **geom_bar()**.

Estos cambios dependerán de cada tipo de gráfico.

Ejemplo

Vamos a cambiar el diagrama de barras por puntos.

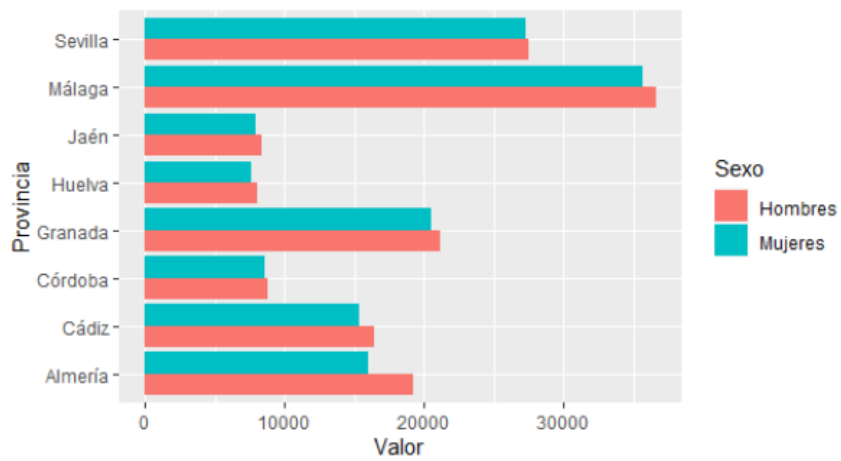
```
ggplot(emi_provincia, aes(Valor, Provincia, color=Sexo)) +  
  geom_point()
```



En este caso hemos cambiado el **geom_bar()** por **geom_point()**.

También podemos cambiar la posición de las barras dentro de un **geom_bar()**, con el parámetro **position**.

```
ggplot(emi_provincia,aes(Valor,Provincia,fill=Sexo))+  
  geom_bar(stat='identity',position='dodge')
```



4.3. Theme

Podemos seguir modificando aspectos del gráfico con la orden **theme()**.

Esta orden no es obligatoria en la construcción de un gráfico, pero nos permite modificar muchos otros parámetros estéticos, por ejemplo, la posición de la leyenda, el color del fondo del gráfico, la aparición de las etiquetas de los ejes, etc.

Los algunos de los elementos del gráfico sobre los que podemos actuar con la orden **theme()** son:

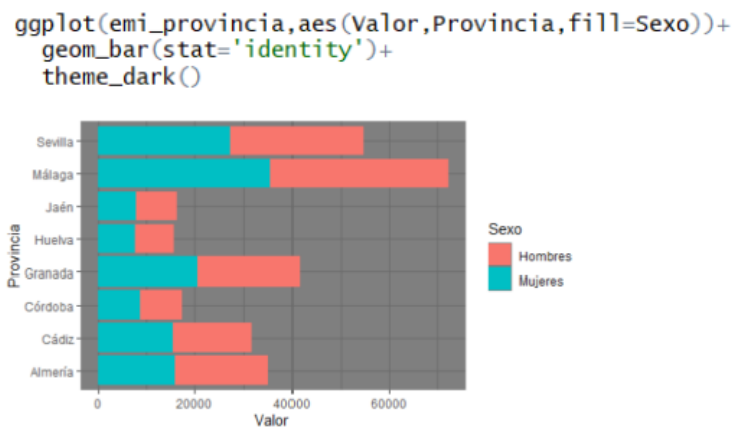
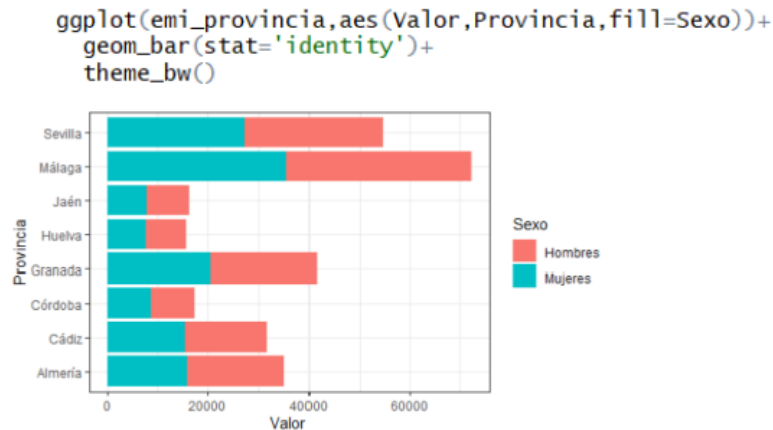
- Leyenda (legend)
- Ejes (axis)
- Gráfico (plot)

En ggplot2 también podemos encontrar varios temas predefinidos que modifican muchos de estos aspectos. Algunos de los más usados son `theme_bw()`, `theme_dark()`, `theme_classic()`.

Os dejo un enlace a todas las cuestiones que podéis cambiar dentro de la orden `theme()`. [Enlace](#).

Ejemplo

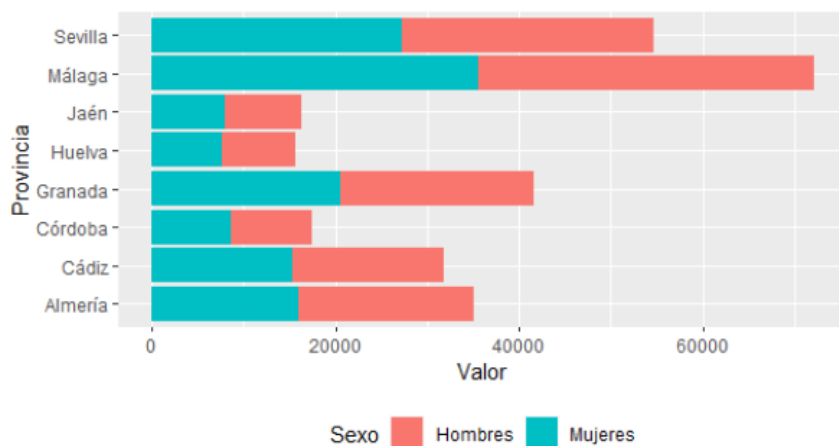
Vamos a ir cambiando algunos aspectos del gráfico con la orden **theme()**.



Podemos usar los temas predefinidos. En el primero hemos usado el tema **bw** y en el segundo el tema **dark**.

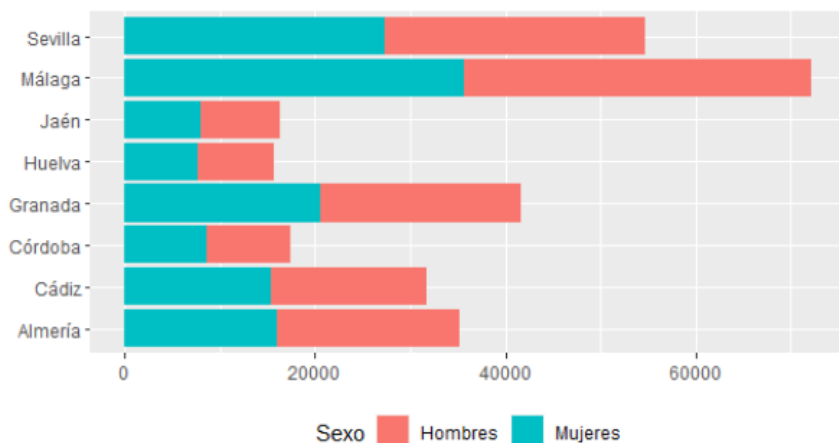
También podemos usar la orden **theme()** básica para hacer cambios en el gráfico. Si usamos esta orden tenemos que ir indicando que es lo que queremos modificar.

```
ggplot(emi_provincia,aes(Valor,Provincia,fill=Sexo))+
  geom_bar(stat='identity')+
  theme(legend.position='bottom')
```



Aquí, lo que hemos cambiado es la posición de la leyenda para que aparezca en la parte inferior. Para ello usamos el parámetro **legend.position**.

```
ggplot(emi_provincia,aes(Valor,Provincia,fill=Sexo))+
  geom_bar(stat='identity')+
  theme(legend.position='bottom',
        axis.title = element_blank())
```



En este último ejemplo, podemos ver como modificar varias cosas al mismo tiempo indicando qué es lo que queremos cambiar.

En este caso, la posición de la leyenda (inferior) y que no aparezcan los títulos de los ejes (usamos **element_blank()** para indicar que deje en blanco alguno de los parámetros).

4.4. Otros

Otras modificaciones que podemos hacer sobre el gráfico son las coordenadas y el faceteado.

Para las **coordenadas de representación** de los gráficos, las más frecuentes son las siguientes:

- **coord_cartesian()**. Las usadas por defecto en ggplot2.
- **coord_flip()**. Igual que el sistema cartesiano pero con los ejes cambiados.
- **coord_polar()**. Para realizar gráficos circulares como el diagrama de sectores.
- **coord_map()**. Para la realización de mapas.

Respecto al **faceteado**, orden lo que hace es dividir el gráfico principal en otros gráficos usando los valores de una variable categórica.

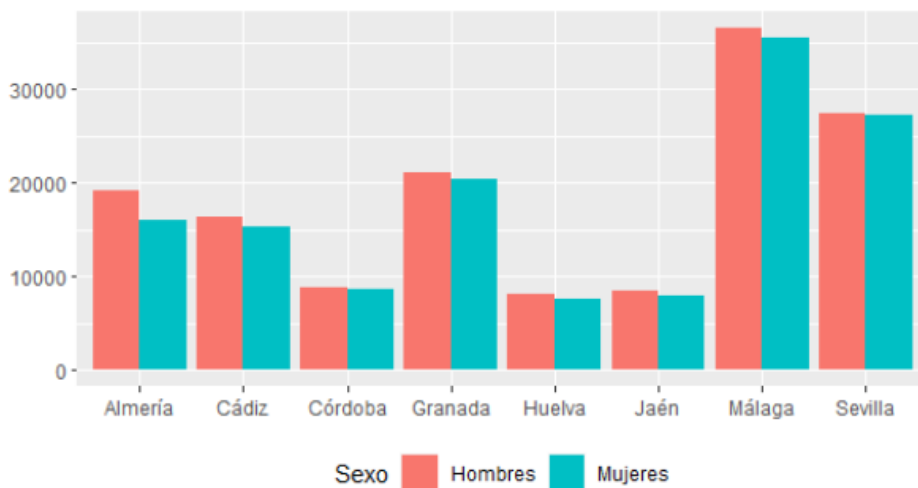
Las opciones que tenemos son:

- **facet_grid()**. Construye los gráficos en una rejilla en función de los valores de la variable categórica.
- **facet_wrap()**. Construye los gráficos en una plantilla rectangular.

Ejemplo

Siguiendo con el gráfico anterior, vamos a añadir un cambio de coordenadas y un faceteado a partir de la variable sexo.

```
ggplot(emi_provincia, aes(Valor, Provincia, fill=Sexo)) +  
  geom_bar(stat='identity', position='dodge') +  
  theme(legend.position='bottom',  
        axis.title = element_blank()) +  
  coord_flip()
```



En este gráfico, añadimos una capa más (usando el +) donde usamos **coord_flip()** para girar la orientación los ejes, poniendo las provincias en el eje abcisas y en el eje de ordenadas el valor de la emigración.

```
ggplot(emi_provincia,aes(Valor,Provincia,fill=Sexo))+
  geom_bar(stat='identity',position='dodge')+
  theme(legend.position='bottom',
        axis.title = element_blank())+
  facet_grid(~Sexo)
```



En este otro ejemplo, hemos añadido el faceteado con **facet_grid()** para generar un gráfico diferente para cada valor de la variable sexo.

En función de como se coloque la variable sexo dentro de la función **facet_grid()**, la colocación de los gráficos puede variar, por fila o por columna. En el primer gráfico aparecen por columna con **(~Sexo)**.

```
ggplot(emi_provincia,aes(Valor,Provincia,fill=Sexo))+
  geom_bar(stat='identity',position='dodge')+
  theme(legend.position='bottom',
        axis.title = element_blank())+
  facet_grid(Sexo~.)
```



En este caso, los gráficos aparecen por filas usando primero la variable **(Sexo~.)**. Recordad que la virgulilla se obtiene con la combinación de teclas Alt+126.