# Ggplot2

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología A, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 4.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 4.2 Documentación de construcción

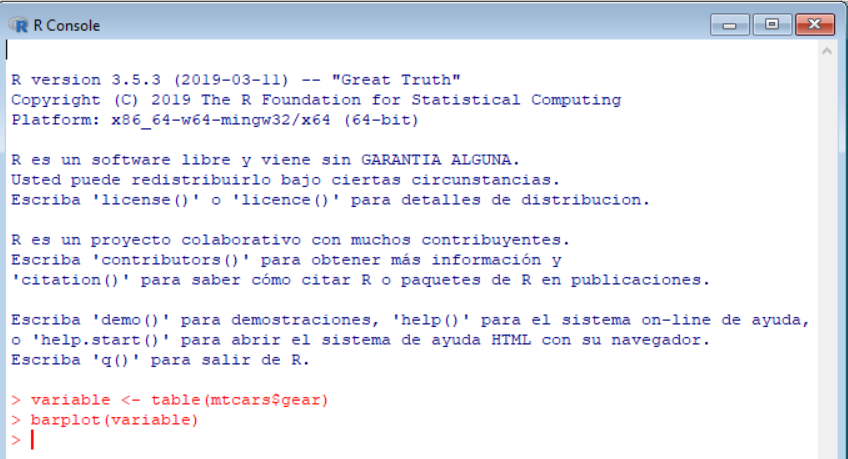
En este apartado trataremos de implementar la visualización de datos (Data Visualization). Concretamente trataremos de implementar todos los gráficos básicos que existen, los cuales son los siguientes:

* Gráfico de barras (Bar chart)
* Gráfico de pastel (Pie chart)
* Historgrama (Histogram)
* Gráfico de densidad del núcleo (kernel density plot)
* Gráfico de línea (Line chart)
* Diagrama de caja (Box plot)
* Nube de palabras (Word cloud)

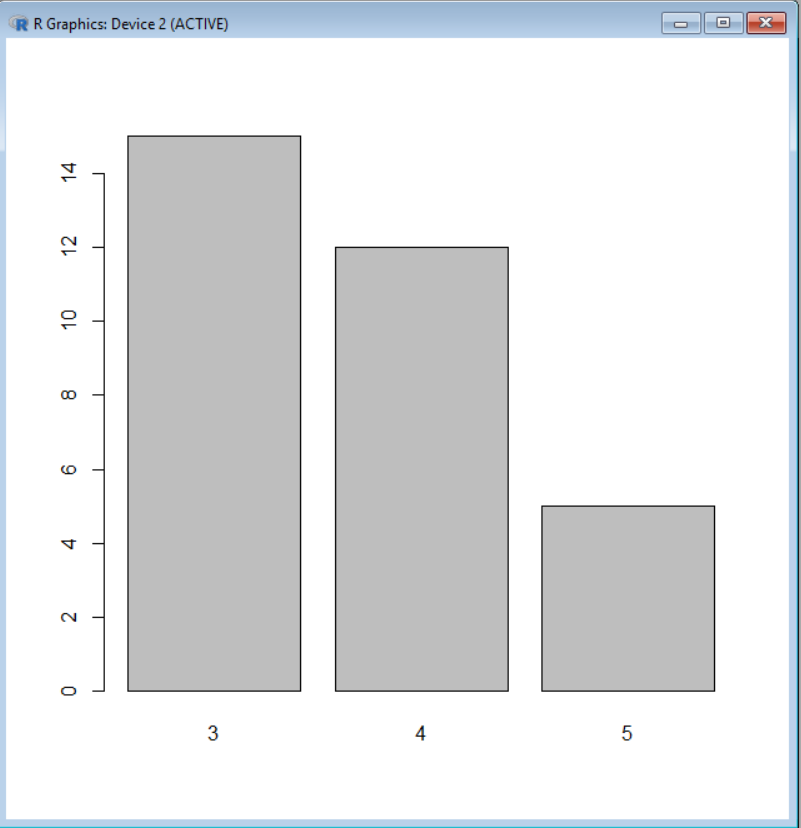
En cada uno de estos gráficos incluiremos tanto el resultado que hemos obtenido como el código que no lo ha generado, tratando de explicar cada una de las variables que hemos definido.

### **Gráfico de barras**

Éste grafico de visualización de datos es de los más simples y de los que más solemos utilizar, para crearlo únicamente tenemos que asignarle a una variable la tabla “mtcars$gear”, la cual lleva unos números por defecto que nos crea el gráfico que queremos. La implementación sería la siguiente:



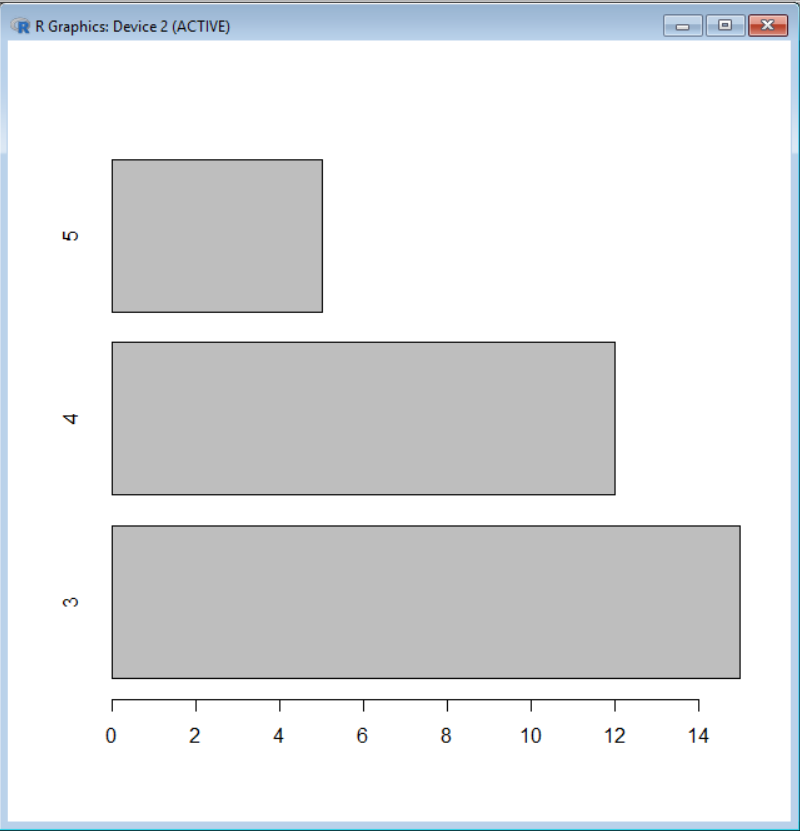
Una vez introducimos el barplot de la variable se nos crea el gráfico de barras:



Este tipo de gráfico lo podemos poner en horizontal añadiéndole al barplot “horiz= TRUE”, quedaría, por tanto, de la siguiente manera:

* Barplot(variable,horiz = TRUE)

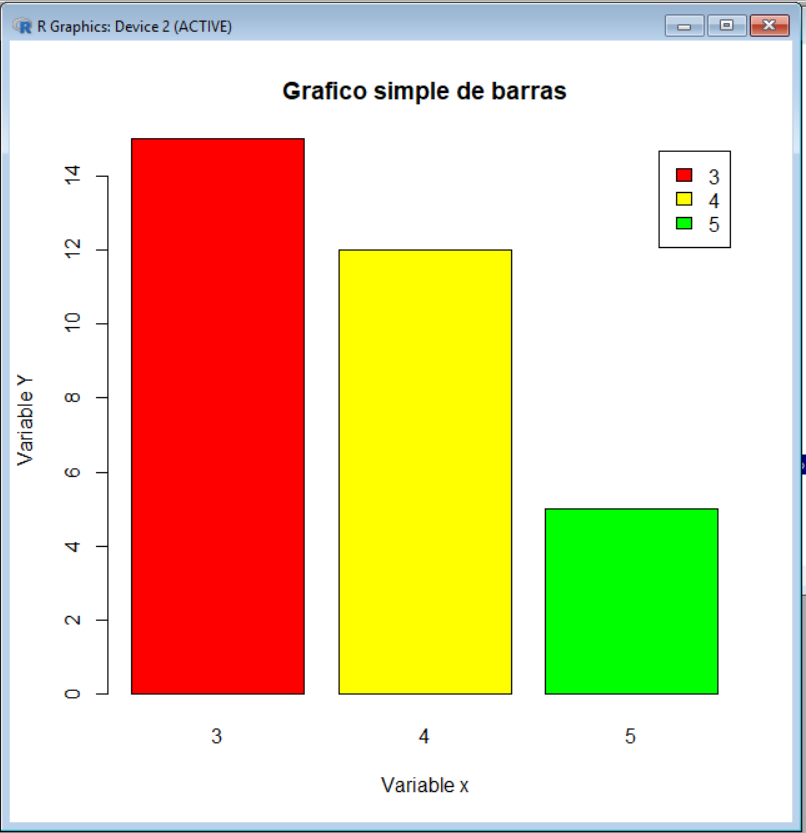
Éste sería el resultado.



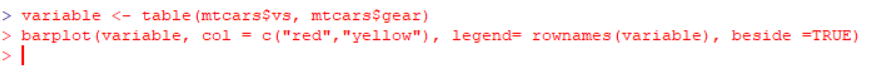
A este tipo de gráficos podemos asignarle colores, títulos, leyendas, etc. Un ejemplo más completo sería el siguiente:

* barplot(variable, main= "Grafico simple de barras", xlab = "Variable x", ylab ="Variable Y", legend = rownames(variable), col = c("red","yellow","green"))

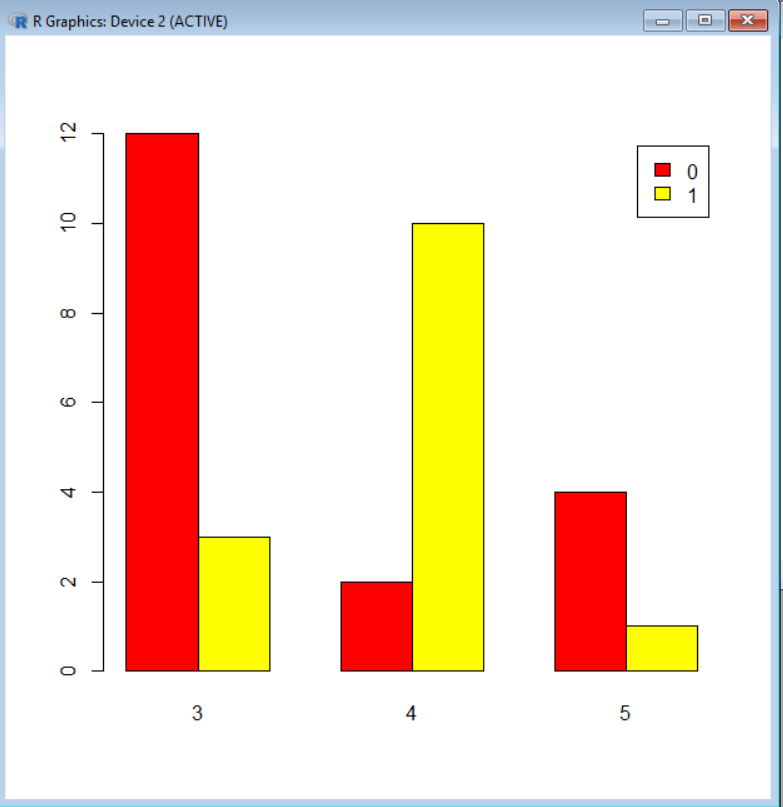
El resultado de eso sería lo siguiente:



En los gráficos de barras, además, pueden hacerse comparaciones o enfrentamientos, esto se implementaría de la siguiente manera:



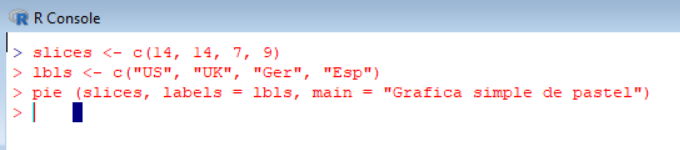
Y el resultado sería el siguiente:



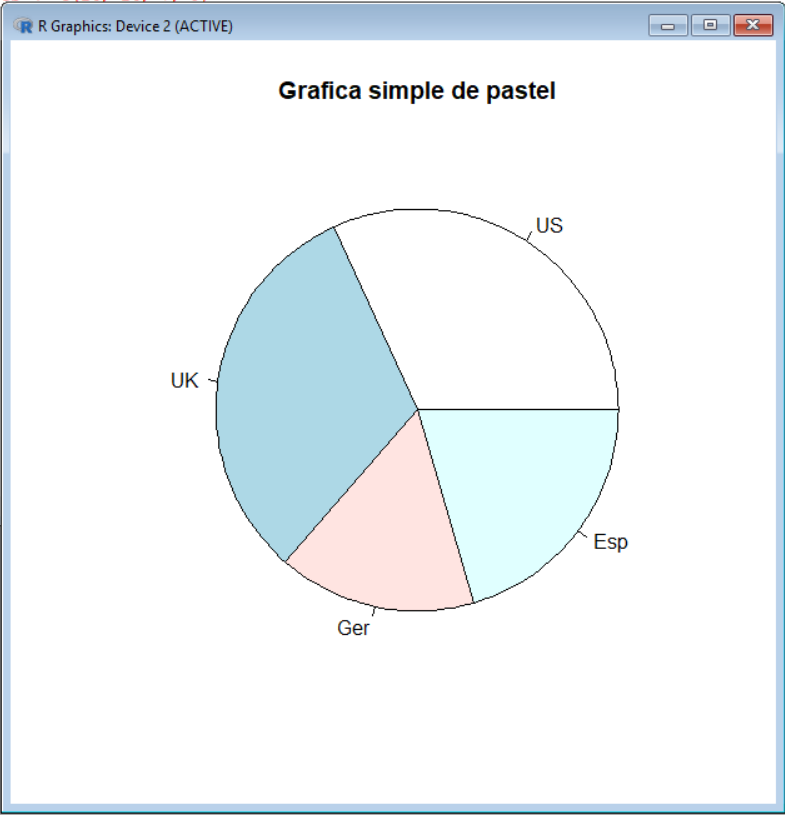
### **Gráfico de pastel**

Este tipo de gráfico también es bastante utilizado, en él un circulo se divide en sectores que representan una porción del todo. De ahí proviene “pastel”, ya que lo dividimos en cachos.

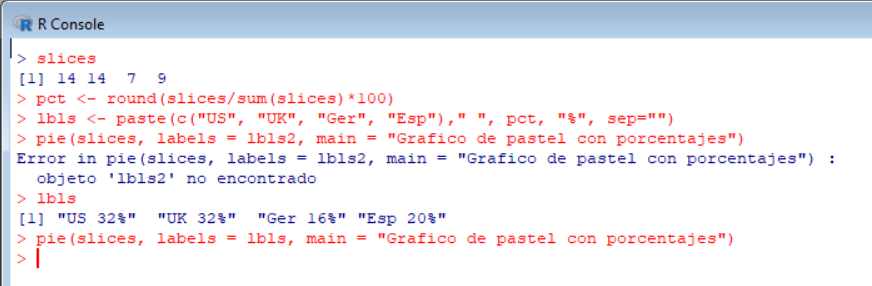
Un gráfico simple seria implementado de la siguiente forma:



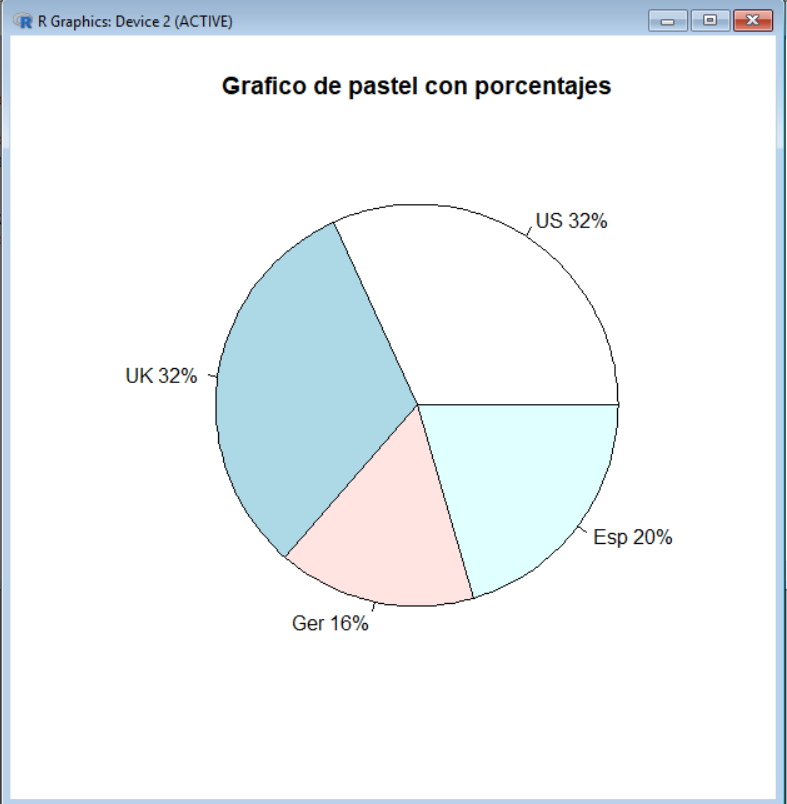
Lo cual tendría como resultado:



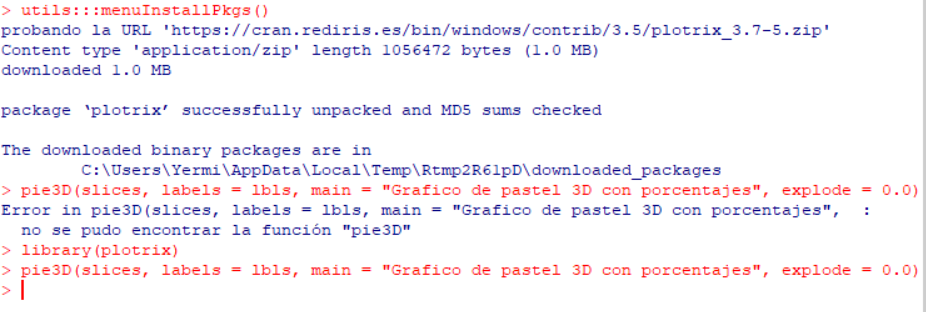
Si queremos añadir porcentajes al gráfico lo haríamos de la siguiente manera:



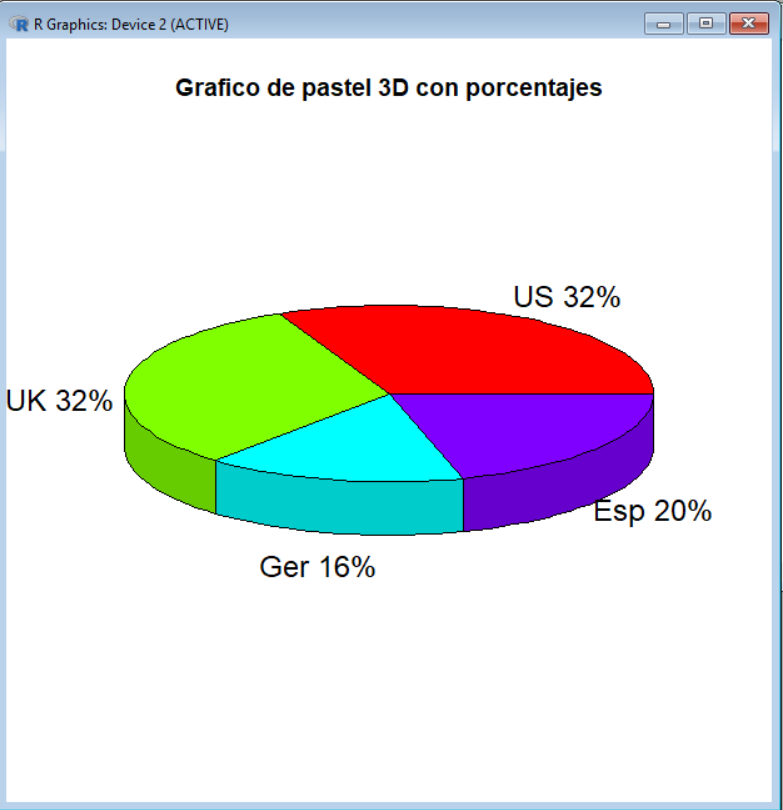
Y el resultado sería el siguiente:



Finalmente, si queremos realizar este gráfico en 3D, tendríamos que instalar el paquete de plotrix, seguidamente llamar a la librería plotrix, y después lanzar la sentencia que lo ejecuta.



El resultado sería el siguiente:

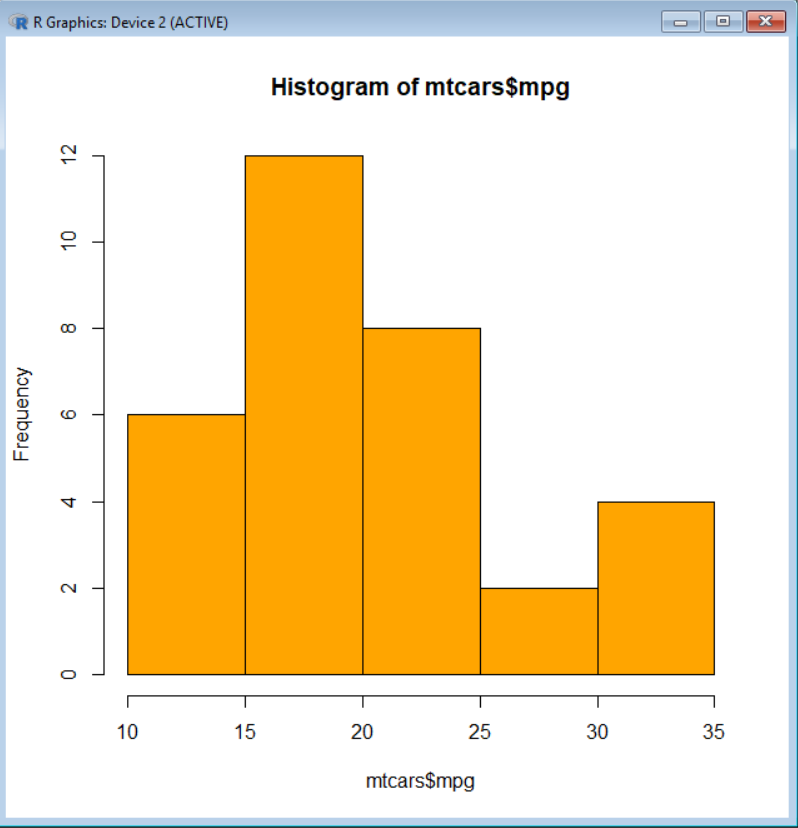


### **Histograma**

Un histograma es un gráfico de la representación de distribuciones de frecuencias, en él hacemos uso de rectángulos dentro de unas coordenadas. Para hacer este tipo de gráfico en R haremos uso del siguiente código:



Como resultado obtendremos un histograma de la siguiente forma:

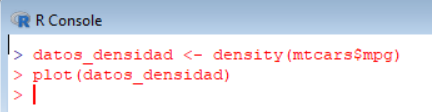


Este tipo de gráfico no puede ser representado en 3D y el histograma esta creado sobre un vector básico que nos ofrece R para observarlo.

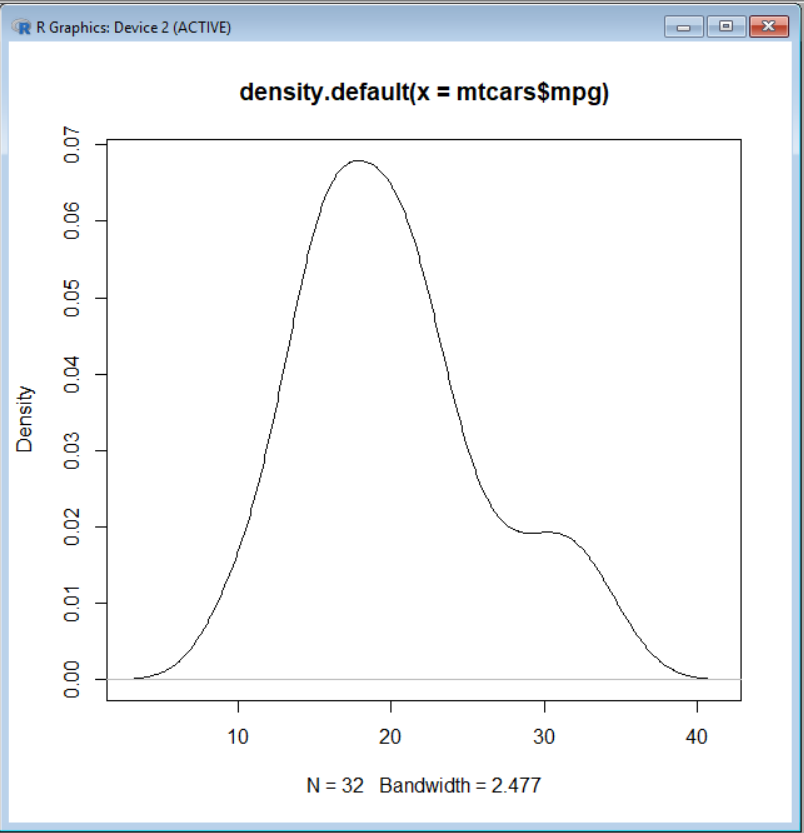
### **Gráfico de densidad de Kernel**

En este tipo de gráficos visualizamos la distribución de datos en un intervalo o periodo de tiempo continuo. Este gráfico es muy similar al histograma solamente que con este utilizamos un suavizado para trazar valores, de esta forma tenemos distribuciones más suaves.

Para elaborar este tipo de gráfico en R utilizamos el siguiente código, el cual nos servirá de ejemplo:



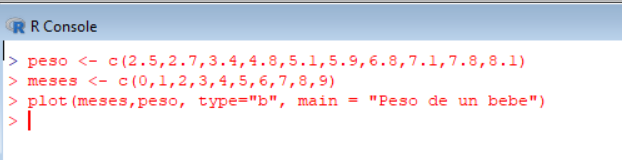
El resultado visual es el siguiente:



### **Gráfico de línea**

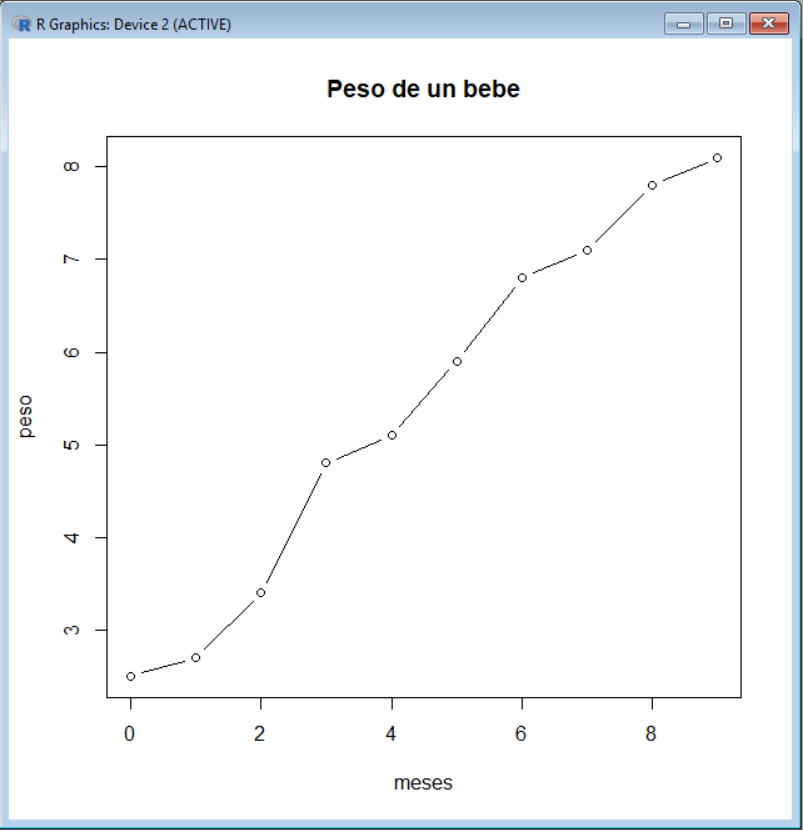
El gráfico de línea es un gráfico muy utilizado tanto en el ámbito escolar como en el ámbito de la estadística. En este tipo de gráficos señalamos con un punto los datos que nos dan y posteriormente unimos esos puntos con una línea fina.

Para representar este tipo de gráfico haremos uso del siguiente código:



En este caso definimos 2 variables, peso que seria la variable “y” y meses que sería la variable “x”, como ambos parámetros tienen un vector de la misma longitud (10), se representa un punto donde ambos coinciden y se unen con una línea, como ya he explicado anteriormente.

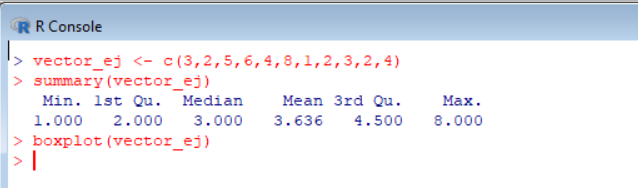
El resultado de ese código daría la siguiente gráfica:



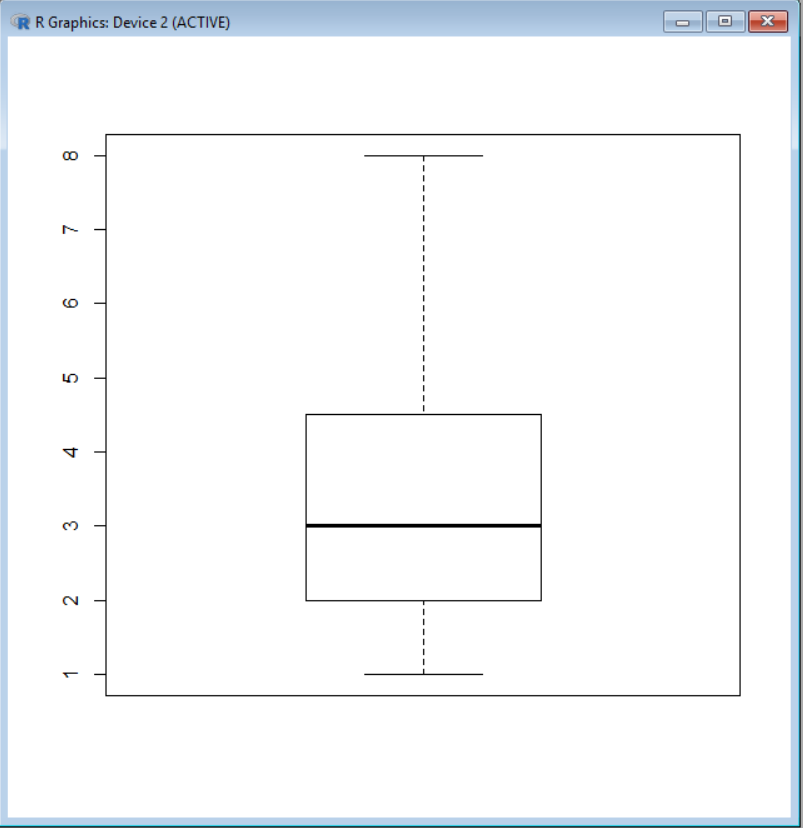
### **Diagrama de cajas**

El diagrama de cajas quizás es de los menos conocidos de todos los que hemos hablado, en ellos tratamos de representar gráficamente una serie de datos numéricos a través de sus cuartiles. De esta forma podemos ver fácilmente los cuartiles, la mediana, etc.

Para la representación en R de un diagrama de cajas usaremos este código:



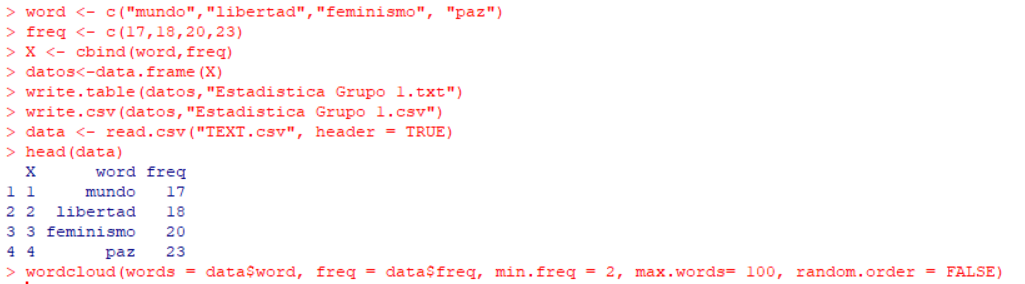
Le incluimos un vector, después sacamos los datos sobre ella y finalmente la representamos:



Como podemos ver, este gráfico representa los valores del vector que hemos introducido, de este grafico podemos sacar fácilmente que el mínimo es 1, el máximo es 8, el valor 2 es el primer cuartil (25% de los datos), el valor 3 es el segundo cuartil o mediana (50% de los datos) y el 4,5 sería el tercer cuartil (75% de los datos).

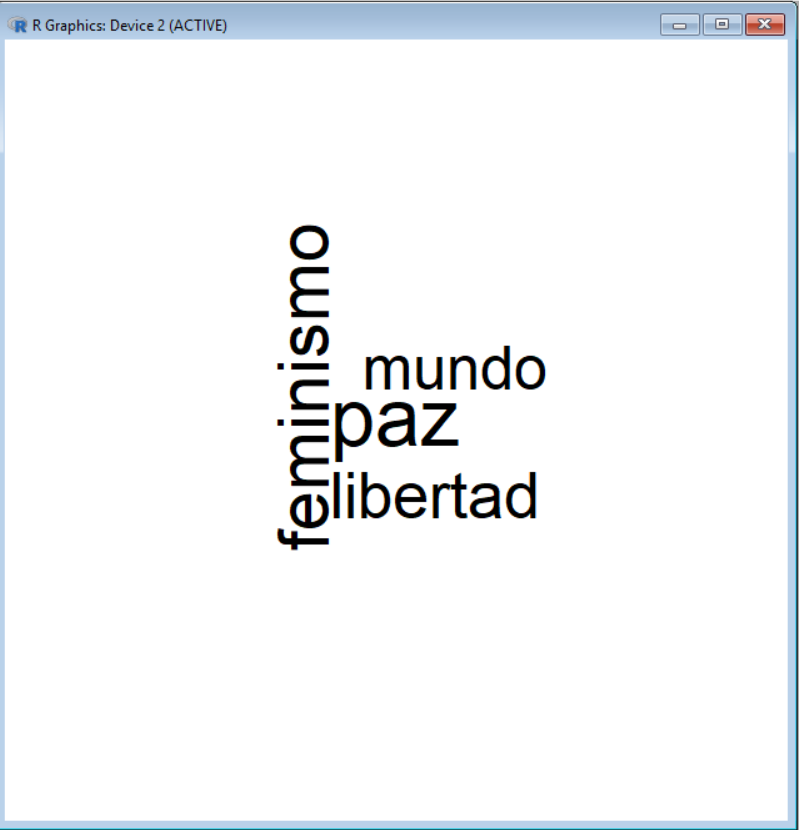
# Nube de palabras

Este tipo de gráfico es básicamente artístico a diferencia de todos los anteriores, tratamos de crear una imagen con un conjunto de palabras. El código que hemos utilizado en R es el siguiente:



Como podemos observar, hemos creado un CSV con dos columnas que son “Word” y “freq”. En la primera columna hemos metido las palabras que queríamos que apareciesen y en la segunda con la frecuencia con la que la queríamos. Además, hemos tenido que importar el paquete de “cloudword” ya que no viene preinstalado en R.

El resultado ha sido el siguiente:



Este tipo de gráfico ha sido de los más costosos ya que ha habido que hacer un CSV desde R porque no servía exportarlo del Excel.

## 4.3 Documentación de pruebas

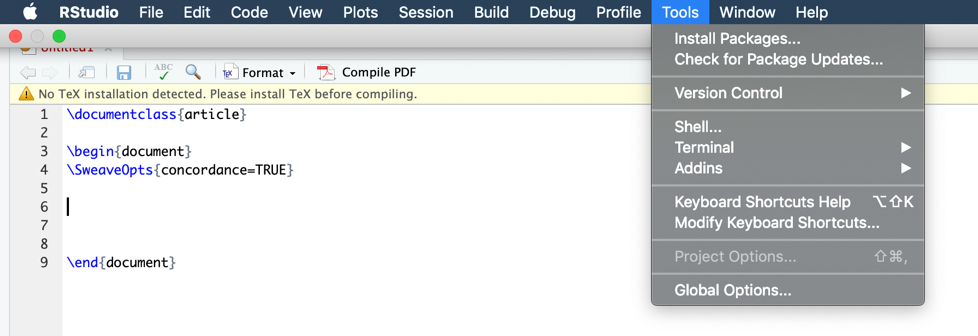
A continuación, vamos a ver los requisitos que se cumplen y los que no con el prototipo de visualización de datos con Ggplot2.

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ.** | **DESCRIPCIÓN** | **√/**X | **Observaciones** |
| RF01 | Se deberán mostrar diferentes niveles de granularidad. | X | No se puede hacer en Ggplot2 |
| RF02 | Se deberá poder aumentar el nivel de abstracción. | X | - |
| RF03 | Se deberán poder mostrar dependencias y relaciones entre los datos de forma clara. | **√** | Se ven claramente las dependencias en los gráficos |
| RF04 | Se permitirá hacer zoom. | **√** | - |
| RF05 | Se mostrará información adicional al pasar el ratón por encima de elementos. | X | No se muestra información adicional, es una imagen estática que se genera |
| RF06 | Implementar checkboxes para filtrar la información se desee mostrar. | X | No se puede hacer con Ggplot2 |
| RF07 | Se podrá analizar y mostrar los datos de diferentes maneras, entre ellas en *conjuntos.* | **√** |  |
| RF08 | Se podrá analizar y mostrar los datos de diferentes maneras, entre ellas en *clusters.* | **√** | Se puede hacer visualización de clusters. Pero no se ha hecho en este prototipo concreto |
| RF09 | Se podrá analizar y mostrar los datos de diferentes maneras, entre ellas *ordenados.* | **√** | A través de comandos el usuario puede cambiar la configuración predefinida |
| RF10 | Se podrán comparar cambios cada cierto tiempo y entre versiones diferentes del software. | **√** | Nos debemos meter a ver las características de cada versión en internet. |
| RNF01 | Se podrá cambiar el color y la representación de los datos. | **√** | Con (theme) podremos cambiar la configuración de colores |
| RNF02 | Se mostrarán los datos de una manera sencilla que facilite la comprensión. | **√** | Predomina la simplicidad |
| RNF03 | El framework utilizara el lenguaje de programación R | **√** | Ggplot2 es una librería de visualización de datos en R |
| RNF04 | El framework será open source | **√** | Ggplot2 es Open Source |

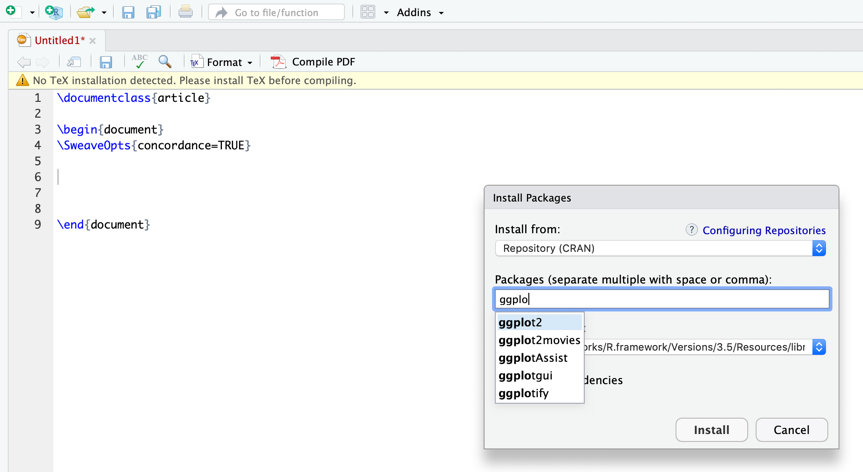
## 4.4 Documentación de instalación

Para la realización de el prototipo vamos a hacer uso de Rstudio y más concretamente, la librería Ggplot2 para la realización de las gráficas para su posterior comparación.

A continuación, vamos a explicar la instalación de la librería en cuestión. Lo primero de todo será instalar y abrir la aplicación de Rstudio, y después de esto nos deberemos dirigir al apartado de Tool e install packages como aparece en la captura inferior.



Nos aparecerá la pantalla emergente que vemos en la captura inferior, y deberemos introducir por teclado el nombre de la librería que queremos incluir. Después de esto instalamos y vemos por la consola si se ha instalado correctamente.







A partir de aquí ya podemos empezar a trabajar.

## 4.5 Manual de usuario

Ggplot2 es una librería de R. En el punto anterior hemos explicado su instalación. Esta librería se usa a través de la consola de RStudio.

Para hacer los diversos gráficos que hemos explicado anteriormente, solo debemos introducir el código correspondiente al gráfico que queremos implementar. Toda esta información aparece en el apartado de implementación de el prototipo.

Gráfico de barras: Con los siguientes comandos podemos hacer un gráfico de barras coloreados.

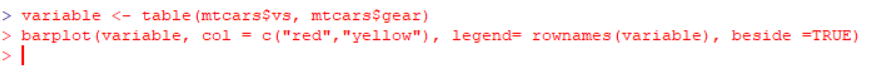
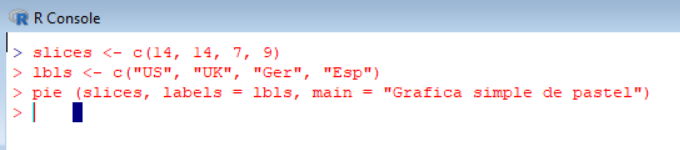


Gráfico pastel: dentro de las variables slices y lbls vamos a introducir los datos.



Histograma:



Gráfico de densidad Kernel:

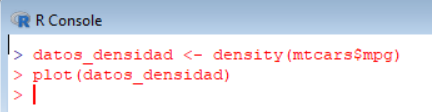


Gráfico de líneas:

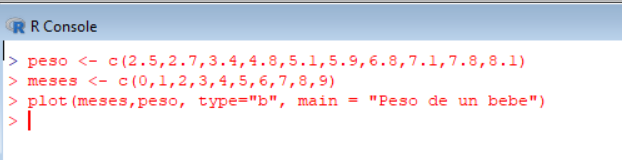
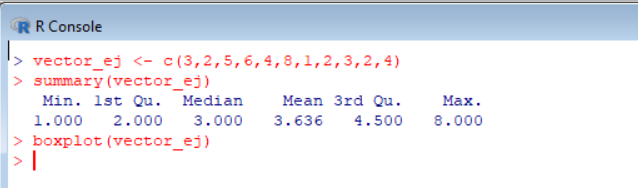


Gráfico de caja y bigotes:



Nube de palabras: Si queremos que aparezca otra nube de palabras, las deberemos cambiar dentro del código.

