**Semana 3**

**13. Graficación con el sistema base de gráficos**

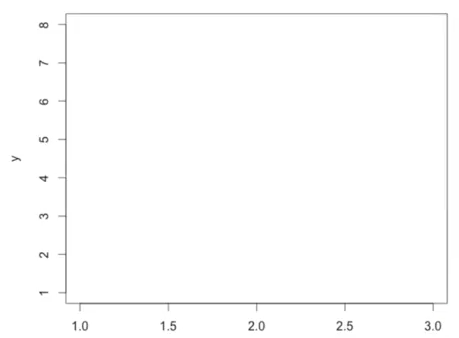
En esta lección cubriremos el sistema base de gráficos de R, que cuenta con 2 tipos de funciones; las funciones de alto nivel que generan gráficas preestablecidas, y las de bajo nivel, que nos permiten construir gráficas elemento a elemento.

Como primera función, veremos la función *plot()*, de alto nivel y que genera gráficas de distintos tipos de dispersión, que constrastan puntos x frente a y en dos dimensiones. Esta función puede suponer el punto inicial de otras más complejas, colocando sobre ésta puntos, leyenda, texto, otros tipos de gráfica… Es una función genérica, dependiendo del objeto de entrada, es decir, podemos construir una gráfica cogiendo un elemento de un vector y un elemento de otro. Puede cambiar el tipo de carácter con el que va a graficar los puntos, como el parámetro *pch=”X”*, que lo que hace es cambiar el punto con el que vamos a representar.



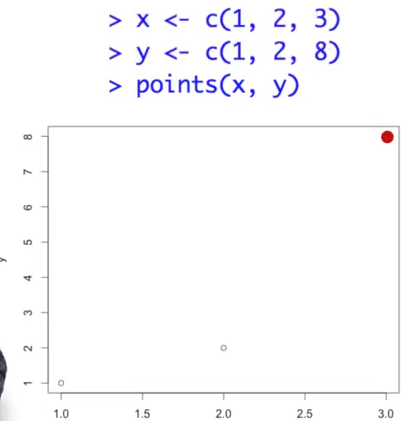
Cambiamos solamente el carácter por una X en lugar de poner un punto. Podemos construir nuestra gráfica vacía e ir agregando los elementos que queremos paso por paso. Para generar una gráfica vacía.



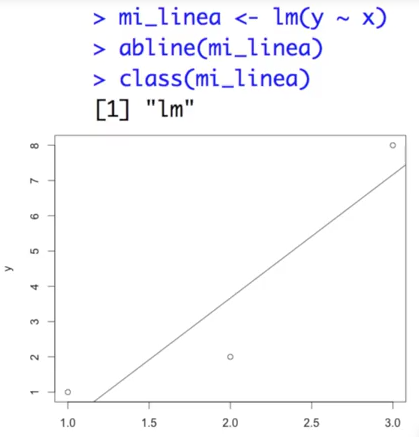


Los dos primeros elementos definen el tamaño de los ejes. Representamos una gráfica “n”, que indica a R que es una gráfica sin puntos, y usamos los parámetros *xlab* y *ylab* para colocar la etiqueta de los ejes.

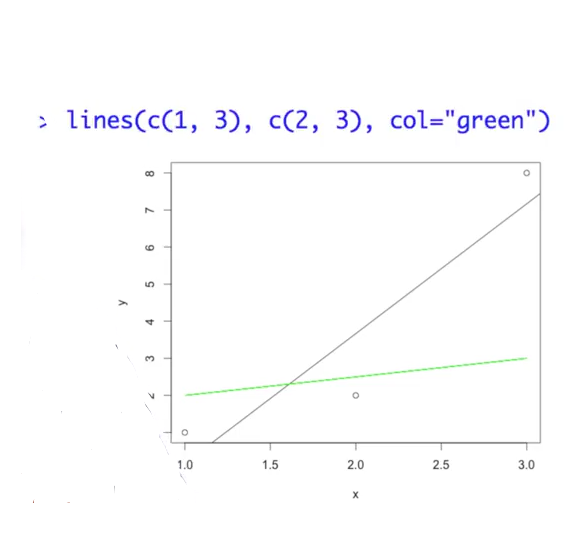
Ahora añadiremos algunos puntos sobre esta gráfica, generando los vectores x e y, usando la función *point()*, que va cogiendo y los va colocando punto a punto.



Una vez hecho esto, si queremos generar una línea con un ajuste lineal, podemos usar la función *lm()*, que recibe los puntos y los intenta ajustar a un modelo de regresión lineal. Podemos guardar este modelo en un objeto, y con la función *abline()* podemos añadir al gráfico dicha línea.

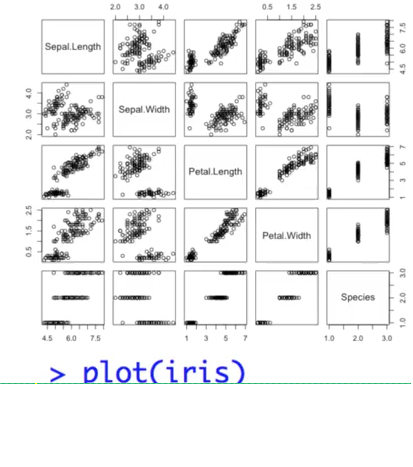


Si mandamos a llamar al objeto *mi\_linea*, se puede ver que se guarda la fórmula y los coeficientes del modelo de regresión lineal que se mandó. Es importante ver cómo se suan las fórmulas, y nos dice cómo queremos que los puntos se guarden. El objeto *class()* guarda el modelo lineal. Si queremos graficar otra línea dentro del mismo modelo, podemos usar la función *lines()*, que recibe un punto inicial y un punto final y traza la línea entre los dos puntos. Al llamar las líneas con dos vectores, podemos generar la línea que va desde el punto indicado al otro. Si queremos diferenciar la línea, podemos pintarla con un color particular, por lo que para ello se representa con el parámetro *col=”green”*.

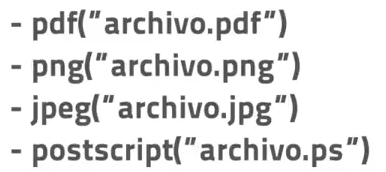


El color debe ir en inglés.

*Plot()* es una función genérica, pero si en lugar de punto por punto le incluimos un datagrame iris que nos da varios valores de medida de las flores del iris y si se incluye, todos estos valores son representados unos contra otros de manera conjunta.



Para guardar nuestras gráficas, podemos usar la función *Export*, que permitirá guardar la imagen o como un archivo como imagen o incluso guardarlo en la memoria. También podemos guardar nuestras gráficas con diversas funciones.



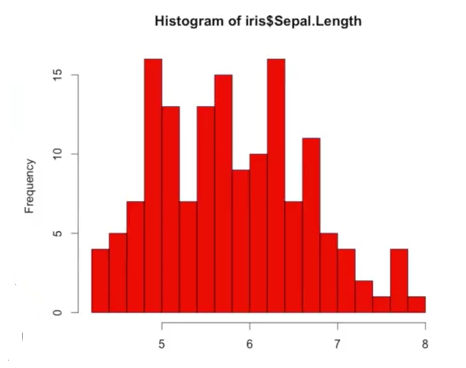
Resumiendo, ahora tenemos algunas funciones para hacer gráficos en R, como *plot(), points(), lines()* y *abline()*, además de que tenemos funciones para guardar nuestras gráficas.

* **Algunas funciones gráficas de alto nivel**

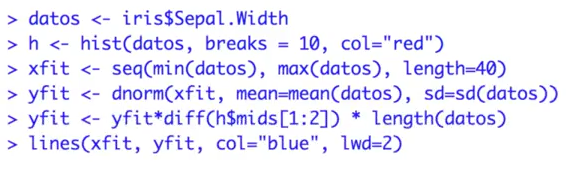
A continuación, revisaremos algunas funciones para hacer funciones de alto nivel, es decir gráficas preestablecidas que nos dan mucha información. Estas gráficas pueden ser de puntos, de histogramas, de densidades…

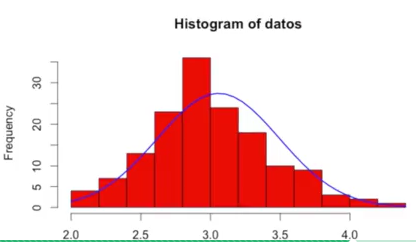
Una de las primeras funciones que veremos son los histogramas mediante la función *hist()*, que recibe un valor y lo representa punto por punto. Si no queremos que represente frecuencias, usamos el parámetro *freq=FALSE* para que represente probabilidades, en lugar de frecuencias. EL color de las barras se hace con el parámetro *col=”red”*. Cuando lo utilizamos así, lo representa de la siguiente manera:





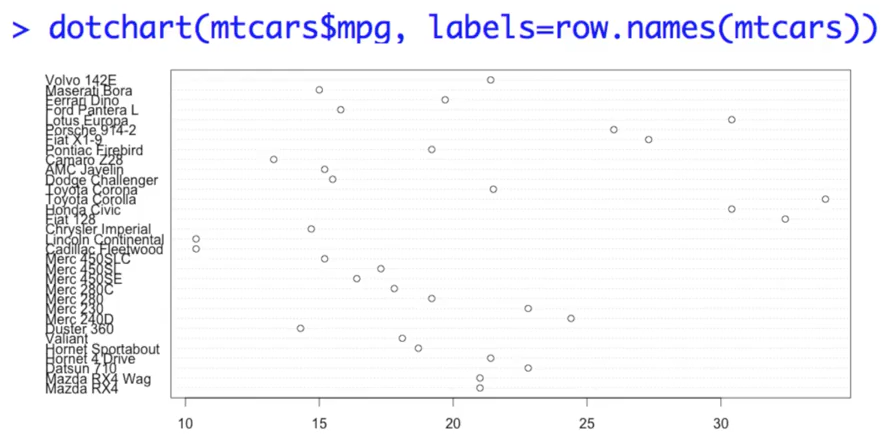
También es posible representar el número de cajas con el parámetro *breaks()*. Como el histograma no nos puede dar una visión exacta de cómo funcionan los datos, podemos colocar una línea de densidad con el programa.





Esta función nos regresa los valores de la densidad de los valores que estamos observando. Una manera más definitiva de ver la densidad es usar una gráfica de densidad, con la función *density()*, indicando un vector numérico de entrada, y una vez introducido, esta nos regresa un ajuste de manera no paramétrica. Este regreso de la función *density()* puede ser usado por *plot()*. Cuando mandamos a llamar a *plot()*, lo que observamos es que la gráfica no es totalmente suave-

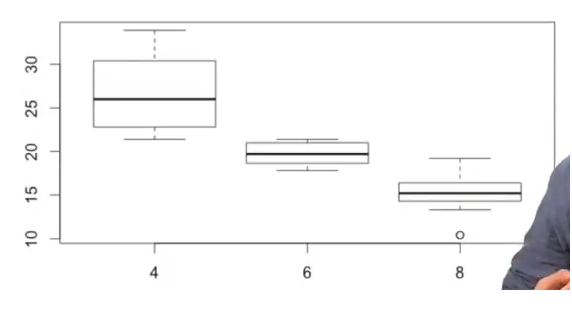
Las gráficas de puntos nos representan un nombre y su valor, por lo que solo tienen que tener un vector de nombres y un vector de valores. Se puede usar la función *dotchart()* para representarla. Se pueden usar Datasets. Si mandamos llamar la función podemos generar una gráfica de puntos.



La función boxplots genera una gráfica de caja. Nos dicen como se comportan los datos unos respectos a otros. En este caso usamos el archivo *mtcars* otra vez. En el primer argumento de la función, indicamos las columnas que vamos a representar.



La gráfica que nos devuelve es la siguiente:



Los valores de dispersión están representados. Todo esto depende de los modelos que estemos observando. Podemos pasarle parámetros para cambiar los títulos y las etiquetas de los ejes. En resumen, una boxplot genera una síntesis gráfica de cómo se están comportando un grupo de datos.

* **Tarea de RStudio**

Para ver los gráficos que puede hacer R, escribe el comando *demo(graphics)*.

*Type* sirve para especificar qué tipo de gráfica debe trazar. Los valores que puede tomar son:

* “p” para puntos
* “l” para líneas
* “b” para ambas (líneas y puntos)
* “c” para la parte de líneas que se muestra usando “b”
* “o” para ambas (líneas y puntos) sobrepuestas
* “h” para histograma como líneas verticales (o de alta densidad)
* “s” para escalonado
* “S” para otro tipo de escalonamiento
* “n” para no graficar

Con la función *abline()* puedes representar una línea recta en la gráfica. Con h y v puedes hacer rejillas en la misma. Si deseas poner una cuadrícula en tu gráfica, usa la función *grid()*.