**Semana 4**

**18. Expresiones Regulares**

En esta lección vamos a hablar de una herramienta importante, las expresiones regulares. Hasta ahora, en el lenguaje se nos han presentado muchos vectores numérios o matrices, y hemos visto vectores con cadenas de caracteres, muy importantes, que se usan para leer archivos con fechas. Hay otros tipos de cadenas, dándonos R algunas funciones que sirven para manipularlas.

* **La función *grep()***

La función *grep()* toma como parámetros un patrón y un valor x que es una cadena de caracteres. *Greg()* nos devuelve un vector numérico cuyos índices indican donde el patrón ha sido encontrado. Es sensible a mayúsculas y minúsculas. En este ejemplo, el patrón que buscamos es “Hola”.



El patrón puede o no ser encontrado. Le damos como vector secundario el vector donde queremos que lo analice. Como resultado, la función nos devuelve la posición 2, que indica el lugar en el que se encuentra la cadena que estamos buscando. Si se usara una H mayúscula, devuelve para este caso un resultado 0.

* **La función *nchar()***

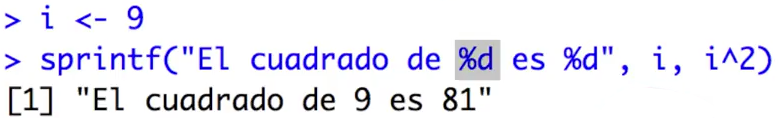
Esta función lo único que nos dice es el tamaño de la cadena de caracteres que tengamos. Si esa cadena está compuesta por elementos separados por comillas, nos devuelve el tamaño de cada uno de los elementos de ese vector.

* **La función *paste()***

Esta función concatena todas las cadenas que recibe como parámetros y nos devuelve una sola cadena. Estas cadenas son separadas por un espacio en blanco, aunque podemos cambiar lo con el parámetro *sep=””*.

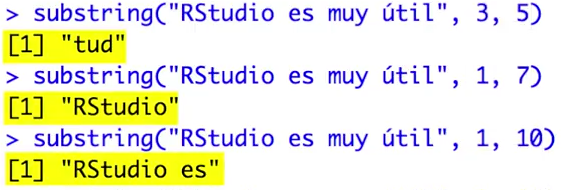
* **La función *sprintf()***

Esta ensambla una cadena y crea una nueva cadena con las partes que vayamos pidiendo. Si queremos formar una cadena que nos diera la información de un valor, podemos pedir que nos la dé con el valor numérico %d, indicando que es un valor decimal. Usamos comas para poner el nombre de las variables que queremos usar, siendo i y su cuadrado.



* **La función *substring()***

Lo que hace esta función es extraer una subcadena de la cadena que le pasamos, y los dos parámetros que nos devuelven son índice de inicio y término que queremos que extraiga dicha subcadena.



* **La función *strsplit()***

Subdivide una cadena en otras más pequeñas en función de un carácter de separación. Pasamos como segundo parámetro el elemento separador y se nos devuelve a cada uno de los elementos separados.

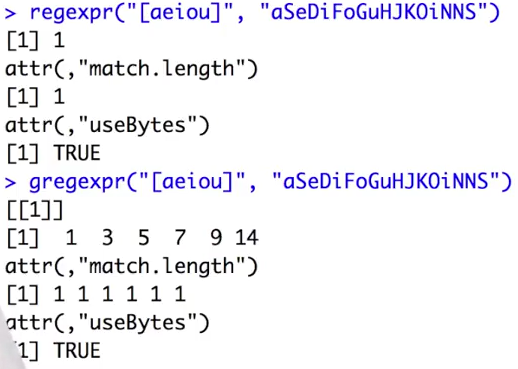
* **La función *regexpr()***

Recibe su nombre de *regular expression*, y nos pide un patrón o una expresión regular como primer parámetro y una cadena de texto como segundo. Esta función encuentra la posición del carácter de la primera instancia del patrón dentro del texto. Esta función tiene un comportamiento similar al anterior, pero ahora nos devuelve los índices donde se han encontrado el patrón que estamos buscando

* **Las expresiones regulares**

Las expresiones regulares o patrones son una especie de comodín o atajo para poder referirnos a una gran cantidad de cadenas. Una expresión regular es una secuencia de caracteres que definen la búsqueda de un patrón para hacer coincidencias de patrones con cadenas de caracteres. No se van a dar en profundidad, pero vamos a ver como R hace una ayuda con la función *help(regexp)*.

Hay libros completos que te enseñan a usar expresiones regulares. El uso de corchetes se puede hacer para buscar las letras en una cadena.



La primera función nos indica el índice donde los encuentra y la segunda todos los índices donde son encontradas.

El punto nos indica una función simple, lo que índica que cualquiera que se ponga valida esa expresión regular. Si tu pides *grep(“a.i”,c())*, lo que le estás pidiendo al programa es que busque la letra a, que a continuación haya cualquier otra letra y que después esté la i. Esto nos indica en qué cadena vamos a encontrar dicho tipo de patrón.

Si queremos buscar un punto dentro de la misma cadena, lo que hay que hacer es sacar dicho punto deslizando la diagonal invertida dos veces, seguidas del punto.

* **Tarea de RStudio**

El símbolo | indica OR, es decir, puede ser Handel o Händel o Haendel si se incluye H(a|ä|ae)ndel.