**Instrucciones básicas**

Existe un comando que genera valores uniformemente en el rango que nosotros ponemos y donde n es la cantidad de números aleatorios que se quieren obtener.



*Generación de números aleatorios*

A veces, en los ejercicios nos piden que generar números aleatorios dentro de un rango, y para ello utilizamos diferentes comandos en función de cómo queremos que sean esos números. Todos los comandos deben almacenarse en un vector, es decir poner el comando c() y dentro las estructuras expuestas a continuación.

**Racionales**

Para generar números racionales, es decir, con decimales, utilizamos la estructura de runif(). Este comando genera n números aleatorios en el rango que establecemos.



Es importante poner primero el número más bajo del rango establecido.

Si lo que queremos generar es números distribuidos uniformemente, se quisiera que estos números siguieran una distribución normal (de media 0 y varianza 1).



(n es la cantidad de números aleatorios que se quieren obtener)

**Naturales**

Para generar números naturales, es decir, sin decimales, el comando utilizado será:



Si escribimos replace=T, indicamos que los números se pueden repetir en el intervalo, en cambio si ponemos replace=F, indicamos que ningún numero se puede repetir.

*Acceder a la memoria*

Acceder a la memoria significa acceder a las variables que se han ido almacenando en el programa de R, a veces puede resultar complicado encontrar algo si tu programa es muy largo. Por lo que hay un comando que facilita esa búsqueda:

Icono

Descripción generada automáticamente

Sin embargo, esta estructura solo nos muestra el nombre de las variables y no sus valores, entonces en el caso de querer saber el valor asociado a ese dato se utilizará:

Logotipo

Descripción generada automáticamente

Más allá de poder ver todas las variables con sus valores podemos también listar todas las que contengan un determinado carácter y también aquellas que empiecen por un determino carácter (añadiendo un ^ al comando).

Imagen que contiene Polígono

Descripción generada automáticamente

Así como podemos ver todo lo almacenado, también podemos borrar elementos si ya no nos son útiles o son indiferentes en el programa. Su sintaxis sería:

