PRACTICA 1:

El **objetivo** es empezar a familiarizarse con el lenguaje en R.

Introducción:

https://youtu.be/SMExZk9j7qs

Conceptos:

1. **Almacenaje de variables**

   Podemos utilizar R como un entorno temporal de trabajo, es decir que podemos ir agregando y guardando datos (resultados de operaciones) y objetos (datos con diferentes atributos).

   Un dato u objeto, puede ser creado con el operador “asignar” el cual se define con una flecha con el signo menos y el símbolo “>” o “<” dependiendo de la dirección en que asigna el objeto:

   n<-3
   n
   "maluma"->n
   n

   En el caso de que queramos guardar, por ejemplo, el resultado de una operación matemática o de una función:

   x = 10+2 * Podemos nombrar el resultado, así como los datos y objetos, como queramos o nos pidan. **Ej:** Alfredo <- 10+2

   Estas variables serán almacenadas temporalmente en la memoria de la consola.

2. **Generar Datos Al Azar:**

   Para generar datos al azar en la consola de R utilizaremos:

   - La función **“rnorm(n)”.** Nos permitirá obtener datos aleatorios a partir de una población teórica inicial con una distribución normal, de media 0 y varianza 1:

     ```
     > n=20
     > rnorm(n)
     [1] -1.28895315 -0.41791120  1.26323690  0.21927108  1.08827109
     0.99319983  1.05437685 -1.85937209
     [9] -0.34923681  0.60767086 -0.18680053 -0.31729906  0.09278316
     1.01654443 -1.77832251 -0.98396990
     [17]  1.33568280 -1.29489469  0.15665927  0.22317495
     ```

   - La función **“runif(n)”**. Nos permitirá obtener datos aleatorios entre [0,1]

```
> n=5
> runif(n)
[1] 0.9386272 0.6977254 0.6740465 0.6801067
[5] 0.9628415
```

* n en este caso significa el número de valores al azar que queremos crear

3. **Funciones De Memoria:**

- **Función “ls ()”:** nos permite crear una simple lista de datos en memoria, mostrando solo los nombres asignados a los datos guardados:

```
> j<-"josema"; x=23; -234->jamón; w=2
> ls()
[1] "j"     "jamón" "w"     "x"
```

Podemos listar sólo aquellos objetos que nos interesen y tengan una característica peculiar. Para ello utilizaremos la **opción de “pattern” (pat)**:

```
> ls(pat="j")
[1] "j"     "jamon"
```

- **Función “ls.str ()”:** muestra todos los detalles de los objetos en memoria:

```
> ls.str()
j :  chr "josema"
jamon :  num -234
w :  num 2
x :  num 23
```

Podemos establecer un nivel de detalle de los objetos de memoria, así evitando la visualización de todos los detalles. Se da con la opción “max.level”

La opción de pattern y ls.str () se usan de la misma manera. Y si queremos borrar datos utilizaremos la **función “rm ()”** (dentro del paréntesis irá el objeto o dato a eliminar)

4. **Operaciones Basicas:**

- **Suma:** +
- **Resta:** -
- **Producto:** *

- **Cociente:** /
- **Potencias:** ** (para elevar)

5. **Simbolos y Funciones:**

   - **Simbolos:**
     - **<** Menor
     - **>**Mayor
     - **<=** Menor o igual
     - **>=** Mayor o igual
     - **!=** Distinto
     - **==** Igualdad lógica
     - **exp(1)** número e
     - **Xe4** igual a X x 104
   - **Funciones:**
     - **log(x)** logaritmo neperiano
     - **log10(x)** logaritmo en base 10
     - **log2(x)** logaritmo en base 2
     - **log (x, base)** logaritmo en cualquier base
     - **exp(x)** función exponencial
     - **sin(x)** seno
     - **cos(x)** coseno
     - **tan(x)** tangente
     - **abs(x)** valor absoluto
     - **sqrt(x)** raíz cuadrada
     - **factorial(x)** factorial
     - **choose(n,x)** binomio de Newton n sobre x

6. **Generar una secuencia:**

   Para crear una secuencia de n valores entre el punto a y el punto b, se utilizará el comando “seq ()”: x=seq(a,b,length)

Vectores y Matrices:

- **Vectores:** almacenan datos numéricos u objetos. Para ello usaremos **“c”** para indicar que es un vector, y en el paréntesis introducimos los valore:

  ```
  > v=c(3,"pablo", 23)
  > v
  [1] "3"     "pablo"   "23"
  ```

  El **producto escalar** es con “%*%”

- **Matrices:** se usan para describir sistemas lineales, ecuaciones…. Para definirla, primero tendremos que definir el numero de filas, con “**nrow”**, y de columnas, con **“ncol”**.

Para introducir datos de vectores, creado anteriormente, utilizaremos los comandos **“cbind”** (columna) y **“rbind”** (filas)

```
> A=matrix(v, cbind=5, rbind = 5)
> A
     [,1]    [,2]    [,3]    [,4]    [,5]
[1,] "3"     "23"    "pablo" "3"     "23"
[2,] "pablo" "3"     "23"    "pablo" "3"
[3,] "23"    "pablo" "3"     "23"    "pablo"
[4,] "3"     "23"    "pablo" "3"     "23"
[5,] "pablo" "3"     "23"    "pablo" "3"
```

Errores Más Comunes:

Los errores más comunes detectados fueron:

- Poner x->2, cuando es x<-2
- Uso de corchetes en lugar de paréntesis
- Escribir mal la palabra “length”
- No poner comillas, al introducir un nombre de un objeto