**Programación con R - Sesión 2**

**Sesión 02: Utilización de Scripts en R con aplicaciónes para el análisis de datos**

**PREWORK**

**Sesión 02: Utilización de Scripts en R con aplicaciónes para el análisis de datos**

**🎯 OBJETIVOS**

En este prework, aprenderás sobre la programación orientada a objetos, las funciones, condiciones y loops en R.

### 👨‍💻 **REQUISITOS**

- Cuenta de **GitHub**

- Git **Bash**

- **R** versión 3.6.2 o mayor

- **R Studio** versión 1.2.5033 o mayor

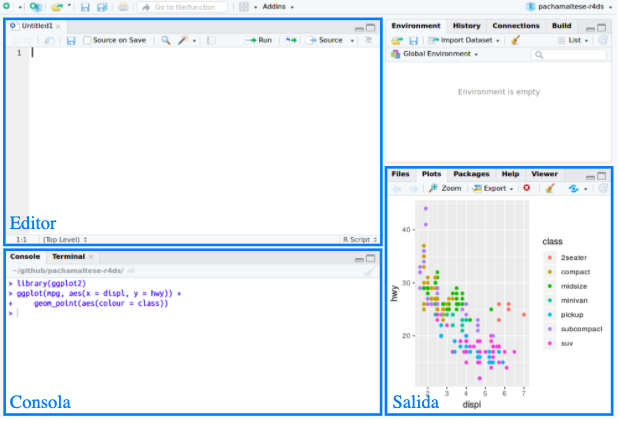
### 🚀 **DESARROLLO**

Ya que tienes una idea de qué es la programación orientada a objetos, en este prework aprenderás a utilizar Scripts y recordarás el concepto de funciones con algunos ejercicios.

**📝 ACTIVIDAD**

***Workflow* proceso de creación de Scripts:** Hasta ahora estuviste utilizando la consola para ejecutar código. Ese es un buen punto de partida, pero verás que se vuelve incómodo rápidamente a medida que creas gráficos de **ggplot2** y *pipes* en **dplyr** más complejos.

Para tener más espacio de trabajo, una buena idea es usar el editor de *script de R*. Ábrelo ya sea haciendo clic en el menú de Archivo (*File*), seleccionando Nuevo Archivo (*New File*), y luego Script de R (*R Script*), o bien, utilizando el atajo del teclado Cmd/Ctrl + Shift + N (en Windows). Ahora verás cuatro paneles:



El editor de R *script* es un excelente espacio para colocar el código que te importa. Continúa experimentando en la consola, pero una vez que has escrito un código que funciona y hace lo que quieres, colócalo en el editor de *script*. RStudio guardará automáticamente los contenidos del editor cuando salgas del programa, y los cargará automáticamente cuando vuelvas a abrirlo. De todas formas, es una buena idea que guardes los *scripts* regularmente y que los respaldes.

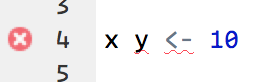
La clave para utilizar el editor de *script* efectivamente es memorizar uno de los atajos del teclado más importantes: Cmd/Ctrl + Enter. Esto ejecuta la expresión actual de R en la consola.

En lugar de correr expresión por expresión, también puedes ejecutar el *script* completo en un paso: Cmd/Ctrl + Shift + S. Hacer esto regularmente es una buena forma de verificar que has capturado todas las partes importantes de tu código en el *script*. Te recomendamos que siempre comiences tu *script* cargando los paquetes que necesitas. De este modo, si compartes tu código con otras personas, quienes lo utilicen pueden fácilmente ver qué paquetes necesitan instalar. Ten en cuenta, sin embargo, que nunca deberías incluir install.packages() (del inglés, *instalar paquetes*) o setwd() (del inglés *set working directory*, establecer directorio de trabajo) en un *script* que compartes.

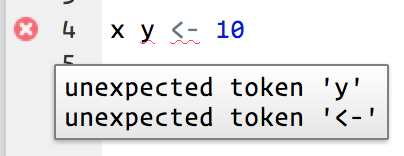
Además también puedes enviar algún script guardado en tu *working directory* a el ambiente de ejecución en memoria de R *(Environment) mediante la función source(“tuscript.R”) .*

**Diagnósticos de RStudio**

El editor de *script* resaltará errores de sintaxis con una línea roja serpenteante bajo el código y una cruz en la barra lateral:



Sitúate sobre la cruz para ver cuál es el problema:



RStudio te informará también sobre posibles problemas:

