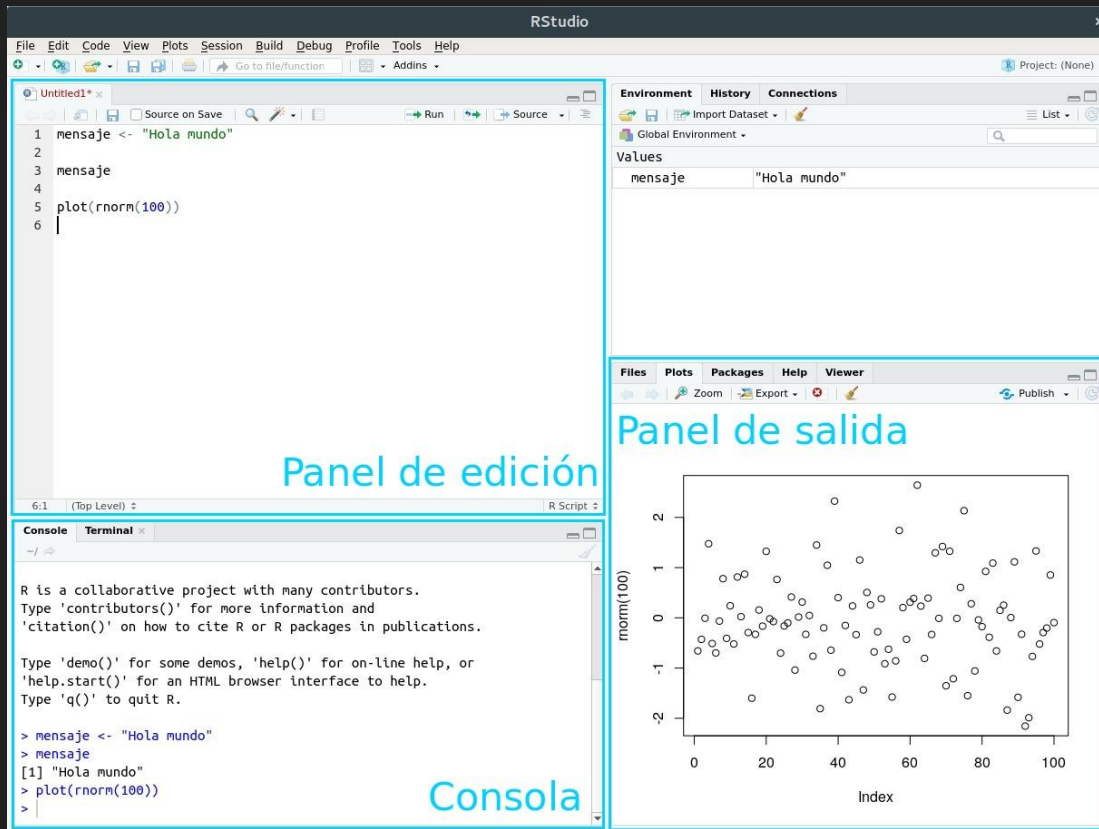


Intro a R

Clase 1: R base

Características de R

- Es un lenguaje de programación orientado al análisis y visualización de datos
- Actualizaciones muy frecuentes (dos o tres por año)
- **Funcionalidad modular:** hay un conjunto de funciones básicas al cual se le van agregando diferentes paquetes con funcionalidades específicas
- Free software → su uso e instalación es gratis
- Para trabajar con este lenguaje es útil una interfaz gráfica, en este caso **RStudio**



- Se escriben comandos en el panel de edición y aparecen en la consola.

Interfaz de RStudio

Tipos de objeto

- Usamos los comandos para crear **objetos**, que pueden ser:
 - **Valores**
 - **Conjuntos de valores (dataframes, listas)**
 - **Funciones**

Tipos de objeto

- Para guardar un objeto puedo usar el símbolo `=` o `<-`
 - Por convención, se usa la flecha. A la izquierda siempre va el nombre que le vamos a asignar al objeto, y a la derecha, el valor.

```
a <- 4 #Esto es un objeto con valor numérico
```

```
b <- "Cuatro" #Esto es un objeto con valor de texto
```

Tipos de objeto: valores

Los valores son la unidad mínima de los objetos en R. Existen cuatro tipos:

- `character`: valores alfanuméricos, es decir, letras, números y signos mezclados.
- `numeric`: valores numéricos, incluye decimales.
- `integer`: números enteros, no incluye decimales.
- `logical`: valores lógicos, un `TRUE` o `FALSE`

Tipos de objeto: vectores

- R puede almacenar un conjunto de valores como un mismo objeto en un elemento llamado **vector**.
- Se interpretan con el comando `c()`

```
vector_1 <- c(55, 65, 75, 85, 95, 105)
```

- Los vectores son de cierto tipo. Si mezclo distintos valores en un vector, R usará la coerción para convertir todos a un mismo tipo.

Tipos de objeto: factores

- Para almacenar valores categóricos se usa el objeto factor. Funcionan como un integer con una etiqueta asociada. Se puede generar factors ordenados con el argumento `levels`.
 - Por default, R usa el orden numérico ascendente u orden alfabético dependiendo el tipo de valor.

```
cat_ocup <- factor(c("Ocup", "Ocup", "Desoc", "Ocup", "Inact"),  
                  levels=c("Inact", "Desoc", "Ocup"))
```


Tipos de objeto: listas

- Las listas son una clase especial de vector: pueden contener elementos de cualquier tipo en su interior.

```
l <- list(c("A", "B", "C", "D"), c(1:15), TRUE)
```

- Son objetos muy importantes en R: la mayoría de los resultados de los modelos que se aplican en R devuelven como output un objeto que es una lista.

Tipos de objeto: dataframe

- Los dataframes son lo más parecido a lo que entendemos por base de datos. **Cada columna es una variable, y cada fila es una observación.**
- Para R son un **tipo especial de vector**, en la cual cada elemento del vector tiene la misma longitud.
- Cada observación está compuesta por datos que pueden ser de distinto tipo.

Tipos de objeto: dataframe

- La función básica para crear un dataframe a partir de dos vectores es `data.frame()`

```
### Creo los vectores
```

```
nombres <- c("Natalia Pérez", "Débora Fernández", "Julián  
López", "Eduardo Gómez", "Ana Pérez")
```

```
edades <- c(34, 65, 51, 36, 41)
```

```
#Creamos el dataframe y lo guardamos en el objeto "data"
```

```
data <- data.frame(nombres, edades, cat_ocup)
```

Indexado - Subsetting/Slicing

- La forma para navegar a través de los valores de un vector, una lista o un dataframe es a través de corchetes (`[]`).
 - En vectores uso un corchete simple
 - `cat_ocup[2]`
 - En listas, uso un corchete doble
 - `l[[1]]`
 - En dataframes, podemos seleccionar un valor entre corchetes usando el orden `[fila, columna]`
 - `data[4, 1]`

Tipos de objeto: funciones

- La función es una forma de encapsular una serie de operaciones que de otra manera nos llevarían más pasos y tiempo.
- Una función tiene un nombre y coloca entre paréntesis los argumentos.

```
fun<-function(argumentos){  
    #[se hace algún cómputo]  
}
```

Tipos de objeto: funciones

- Ciertas funciones también incluyen **parámetros**, las cuales agregan **especificaciones** que hacemos a la operación que realiza la función

```
fun<-function(argumentos,  
               parámetro = TRUE){  
  #[se hace algún cómputo]  
}
```

Tipos de objeto: funciones

- Por ejemplo, a la función `paste()` sirve para concatenar valores de texto. Le podemos pasar como **argumento** todos los **valores** que queremos, y con el **parámetro sep** indicamos el separador que van a tener.

```
paste("Esto", "es", "un ejemplo", sep = "_")
```

```
[1] "Esto_es_un ejemplo"
```

Veamos esto en R...