1. **Investigue cómo leer un archivoCSV desdeR.**

Cargar archivos de texto a variable

R tiene funciones genéricas para abrir este tipo de archivos en una tabla como `read.table(...)`. Esta función presume pocas cosas en la estructura de datos, por lo que permite especificar un montón de parámetros y nos brinda variabilidad (ver `help(read.table)`). Sin embargo, en general conocemos la estructura de nuestros datos (por ejemplo, la primera fila es el título de las columnas o es un archivo separado por comas). Por lo tanto, usaremos llamadas del estilo:

```
datos <- read.csv(file = 'nombre_de_archivo.csv')
```

Es común que las computadoras en español utilicen el separador `;` en vez de `,` para archivos `.csv`. En ese caso, podemos especificar:

```
datos <- read.csv(file = 'nombre_de_archivo.csv', sep = ';')
```

Un archivo separado por tabulaciones (`.txt`) puede leerse como:

```
datos <- read.table(file = 'nombre_de_archivo.txt', sep = '\t')
```

2. **Problemas comunes:**

```
> datos <- read.csv(file = 'C:\Users\Leo\Documents\Maestria 2023\Scripting\primary_results')
Error: '\U' used without hex digits in character string starting "'C:\U"
```

R error: '\u' used without hex digits in character string starting ""c:\u"

How to fix this error message.

Fixing this problem is actually quite easy. Even better you do have just have one solution but three. All three of these methods use a different way of avoiding this backslash problem.

```
# solution 1: '\u' used without hex digits in character string
starting ""c:\u"
x = read.csv("C:\\Users\\Bob\\Desktop\\problem.csv")
```

Investigue acerca de la función select() de la paquetería de dplyr.

select: Subset columns using their names and types

Description

Select (and optionally rename) variables in a data frame, using a concise mini-language that makes it easy to refer to variables based on their name (e.g. a:f selects all columns from a on the left to f

on the right). You can also use predicate functions like is.numeric to select variables based on their properties.

Overview of selection features

Tidyverse selections implement a dialect of R where operators make it easy to select variables:

: for selecting a range of consecutive variables.

! for taking the complement of a set of variables.

& and | for selecting the intersection or the union of two sets of variables.

c() for combining selections.

In addition, you can use selection helpers. Some helpers select specific columns:

everything(): Matches all variables.

last_col(): Select last variable, possibly with an offset.

These helpers select variables by matching patterns in their names:

starts_with(): Starts with a prefix.

ends_with(): Ends with a suffix.

contains(): Contains a literal string.

matches(): Matches a regular expression.

num_range(): Matches a numerical range like x01, x02, x03.