**Introducción R tidyverse: Trabajo práctico final**

**Creación del proyecto y carga de datos**

1. Crear un proyecto de R a partir de un nuevo directorio y llamarlo **tp\_final\_Apellido**.
2. Dentro del nuevo proyecto crear la carpeta **raw**, que usaremos para guardar los datos crudos y descargar los datasets correspondientes al periodo 2020-2024 del sitio: <http://datos.salud.gob.ar/dataset/vigilancia-de-dengue-y-zika>. Guardarlos en la carpeta **raw** y renombrarlos como: **msal\_dengue\_zika\_202X**, donde 202X representa el año de reporte de los casos (Por ejemplo “msal\_dengue\_zika\_2020”).
3. Generar un nuevo script con el nombre **script\_tp\_final\_Apellido** que contenga la siguiente documentación y secciones:

A screenshot of a computer

Description automatically generated

1. Cargar los paquetes que consideren necesarios para la limpieza, exploración y representación de los datos.
2. Importen los datos crudos a R siguiendo el formato:

**msal\_202X <- import(“raw/msal\_dengue\_zika\_202X.xlsx”)**

Tengan en cuenta que los archivos descargados pueden tener distintas extensiones (.xlsx, .xls, .csv) y configuraciones regionales (encoding). En caso de que al importar el dataset aparezcan caracteres extraños en las palabras con acentos o ñ, modifiquen el comando anterior de la siguiente forma:

**msal\_202X <- read\_csv2(“raw/msal\_dengue\_zika\_202X.xlsx”,**

**locale = locale(encoding = "WINDOWS-1252"))**

Una forma rápida de encontrar el archivo es posicionando el cursor en “raw/” y apretar la tecla TAB para que aparezca la lista de archivos. También pueden ejecutar el siguiente comando en la consola: list.files("raw").

**Limpieza y unión de datos**

1. Crear nuevos objetos para las versiones limpias de cada base según lo aprendido en clase. Recuerden como primer paso usar la función **clean\_names()** para eliminar espacios, acentos y caracteres especiales. Explorar los nuevos nombres usando **names(nombre\_del\_dataset)**.
2. Renombrar manualmente las columnas que contengan los nombres de provincias, departamentos y grupos etarios como **provincia, departamento, grupo\_edad**, la columna del año como **periodo**, las semanas epidemiológicas como **semana\_epi** y el número de casos como **casos**.
3. Modificar las variables **departamento** y **grupo\_edad** para que todos los nombres estén en mayúsculas (pueden usar comandos de R base o de tidyverse) y sin acentos ni caracteres especiales (**stringi::stri\_trans\_general(id = "Latin-ASCII")**).
4. Usar los comandos **skim()** y **tabyl()** para explorar los valores ausentes e inconsistencias en las columnas, prestar particularmente atención a **departamento** y **grupo\_edad**.
5. Reemplazar todos los valores extraños que hayan encontrado en las variables **departamento** y **grupo\_edad** por NAs.
6. Crear la variable **grupo\_edad\_cat** que contenga los siguientes niveles: “0-9 años”, “10-19 años”, “20-34 años”, “35-65 años”, y “65+ años”.
7. Filtrar los casos de dengue para provincia de Buenos Aires y retener solo las columnas **departamento, periodo, semana\_epi, grupo\_edad\_cat** y **casos.**
8. Descartar las filas con valores ausentes (NAs).
9. Una vez completado el proceso para todas las bases, unirlas en el objeto **msal\_dengue**, exportarla como “**msal\_dengue\_2020\_2024.xlsx**” y limpiar el Environment.

**Tablas de resumen, gráficos y mapas**

1. Importar la base limpia y generar tablas de resumen para el número de casos por las distintas variables del dataset.
2. Generar un nuevo objeto llamado **casos\_dengue** que contenga el número total de casos por departamento, periodo, semana epidemiológica y grupo etario.
3. Realizar los gráficos exploratorios que consideren necesarios.
4. La carpeta comprimida **clima\_PBA.zip** contiene un shapefile con datos sobre la elevación, temperatura media anual y precipitaciones para la provincia de Buenos Aires. Siguiendo el ejemplo de la clase final, generar mapas coropléticos del número de casos de dengue por departamento de PBA, estratificanado por años y/o grupos etarios.