Curso Primeros Pasos en R

Clase 4: Manipulación y otras estructuras de datos

Profesora: Ana María Alvarado

Pontificia Universidad Católica de Chile

Noviembre 2021

Clase 4: Manipulación y otras estructuras de datos

- Paquete janitor
- Paquete stringr
- Fechas
- Paquete lubridate
- Paquete skim
- Paquete naniar
- Factores
- Taller práctico

Paquete `janitor`

Paquete `janitor`

El paquete `janitor` contiene funciones útiles para la limpieza y manejo de bases de datos sucias, algunas de sus funciones son:

`clean_names()` Permite limpiar los nombres de las variables de una base de datos, realizando algunas modificaciones:

- Transforma todo a minúsculas.
- Reemplaza estapios por "_".
- Elimina tildes.
- Convierte % a "percent".
- ¡entre otros!

```
clean_names(nombre_base)
```

`remove_empty()` Permite remover las filas/columnas vacías de una base de datos.

```
remove_empty(nombre_base, which=c('rows','cols'))
```

Paquete `stringr`

Paquete `stringr`

El paquete `stringr` contiene múltiples funciones que nos permiten trabajar con stirngs de manera sencilla, estas pueden ser muy útiles para cuando queremos detectar, cambiar o eliminar patrones en vectores de tipo character, alguna de estas funciones son:

- `str_detect(objeto,pattern)` : Detecta si el patrón indicado existe o no en cada elemento del objeto, retorna TRUE o FALSE.
- `str_extract(objeto,pattern)` : Extrae de cada elemento la primera vez que el patrón se cumple.
- `str_remove(objeto, pattern)` : Elimina de cada elemento la primera ocurrencia del patrón.
- `str_replace(objeto, pattern, replacement)` : Reemplaza la primera ocurrencia del patrón por un patrón de reemplazo.

También existen los comandos `str_extract_all()`, `str_remove_all()` ,`str_replace_all()` que realizan las mismas acciones anteriores, pero para todas las ocurrencias dentro de los elementos.

Estos patrones pueden estar dados por expresiones regulares, para más información con respecto a estas pueden ejecutar el comando `help(regex)`.

Fechas

Fechas (clase `date`)

En R la clase de archivo `Date` nos permite definir objetos y variables de bases de datos en un formato fecha. El comando `as.Date` nos permite transformar objetos y variables como fecha, para eso se tiene que configurar un formato con ayuda de los siguientes códigos:

Código	Descripción
%Y	Año en 4 dígitos (1982)
%y	Año en 2 dígitos (82)
%m	Mes en 2 dígitos (01)
%d	Día del mes en 2 dígitos (13)
%A	Día de semana (Wednesday)
%a	Día de la semana abreviado (Wed)
%B	Mes (January)
%b	Mes abreviado (Jan)

A continuación presentamos tres formas distintas de definir el mismo objeto en formato fecha:

```
as.Date("1982-01-13")
as.Date("Jan-13-82", format = "%b-%d-%y")
as.Date("13 January, 1982", format = "%d %B, %Y")
```

Paquete `lubridate`

Paquete `lubridate`

El paquete `lubridate` nos permite trabajar de manera eficiente con variables tipo fecha, es parte de tidyverse pero hay que instalarlo de manera independiente.

Las funciones de este paquete son buenas para realizar operaciones entre fechas, una funcionalidad no incluida en R base.

Puede definir codificar como fecha utilizando la funciones, `dmy()`,`mdy()`,`ymd()`, entre otras permutaciones dependiendo del formato.

```
dmy('10011997')
mdy("July 4th, 2000")
ymd(20170131)
```

Puede obtener una componente de una fecha utilizando las funciones `date()`, `year()`, `month()`, `day()`, entre otras.

```
month(mdy("July 4th, 2000"))
```

Web del paquete `lubridate`.

Paquete `skimr`

Paquete `skimr`

El paquete `skimr` contiene funciones que nos permiten resumir bases de datos de una manera más completa, esto gracias principalmente a la función `skim()`, que hace un resumen detallado de los objetos, este puede ser aplicado a vectores o a bases de datos, y entrega cosas distintas dependiento del tipo de objeto:

`skim()` aplicado a vectores.

- **Numérica:** hace un resumen de cuantiles y media, además de un histograma representativo de la distribución.
- **Categórica:** hace un resumen de las frecuencias, cuenta el número de categorías, etc.

`skim()` aplicado a bases de datos.

- **Resumen de datos:** Una tabla que contiene las dimensiones de la base y la frecuencia de cada tipo de columna.
- **Resumen de cada tipo de columnas:** Agrupa en secciones los distintos tipos de variables y calcula para cada una los resúmenes respectivos.

La función `skimr()` también puede aplicarse a matrices.

Funciones adicionales `skimr`

- `partition()`: Nos entrega como resultado una lista de R, en donde cada elemento corresponde a un tipo de variable de la base de datos
- `yank()`: Nos permite seleccionar qué tipo de variable queremos ver, funciona similar a la función `select()` de `dplyr`.
- `to_long()` : Transforma el resultado de `skim()` a una base de datos de 4 columnas.
- `focus()` : Nos permite extraer una versión reducida de `skim()`, permitiendo seleccionar las medidas que queremos observar en nuestro resultado.

Para más información y ejemplos del paquete `skimr` visite el siguiente **link**.

Paquete `naniar`

Paquete `naniar`

El paquete `naniar` contiene numerosas herramientas para explorar los valores faltantes (NA) de nuestras bases de datos, contando con herramientas de conteo, visualización, entre otras. La siguiente sección está basada en la web **Exploring missing values in naniar** de Allison Horst.

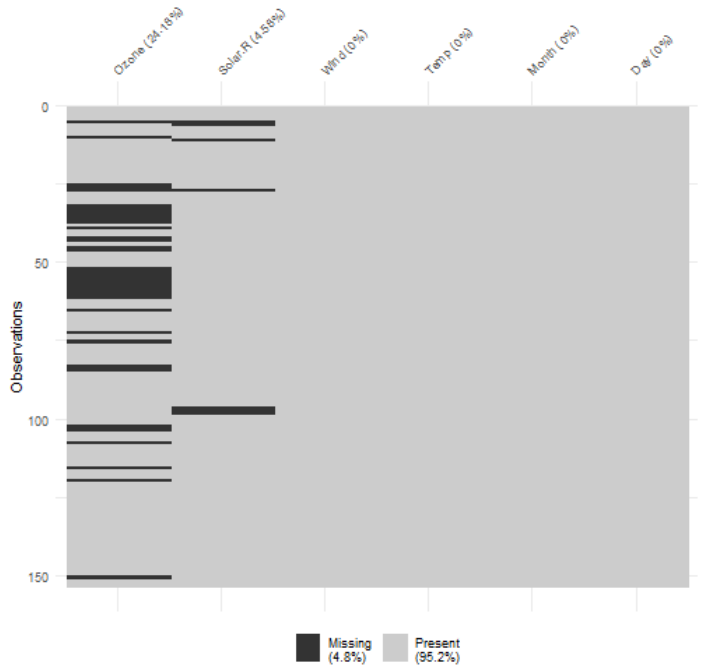
Funciones de conteos de NA

- `n_miss()`: Retorna el total de NA en un data frame o columna.
- `n_complete()`: Retorna el número de valores completos.
- `prop_miss()/pct_miss()`: Proporción o porcentaje de valores `NA`
- `miss_var_summary()`: Tabla de resumen de los `NA` por variable.
- `miss_case_table()`: Tabla de resumen de los `NA` por fila.

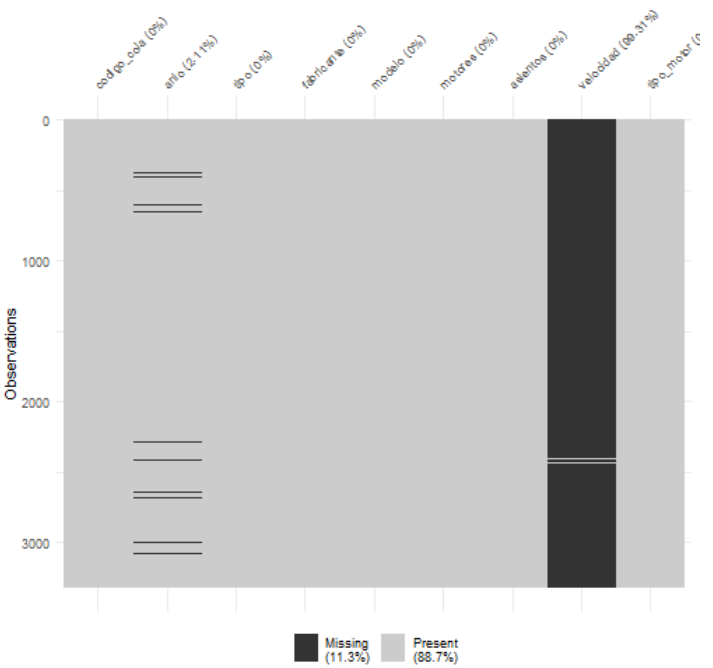
Funciones de visualización de NA

La función `vis_miss()` genera un mapa de calor con todas las variables, pintando de color negro los valores faltantes, por ejemplo:

```
vis_miss(airquality)
```



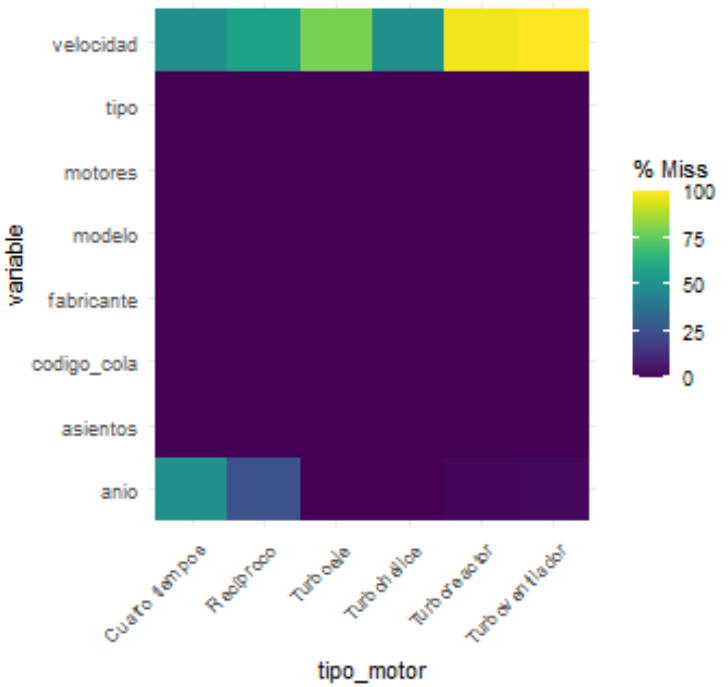

```
vis_miss(datos::aviones)
```



Funciones de visualización de NA

La función `gg_miss_fct()` genera un mapa de calor segregando por una variable categórica, lo que permite mostrar los cruces entre estas categorías y el número de NA:

```
gg_miss_fct(datos::aviones, fct = tipo_motor)
```



Factores

Factores

Un **factor** es una variable categórica con un número finito de valores. En R los factores se utilizan habitualmente para realizar clasificaciones de los datos, estableciendo su pertenencia a los grupos o categorías determinados por los niveles del factor.

Para definir un factor en R, se puede utilizar la función `factor`, el cual recibe como argumentos el vector de categorías, además, pueden definirse los argumentos `levels` y `labels` para definir el orden de los niveles del factor y sus nombres respectivamente.

```
genero <- c("M","H","M","M","M","H","M","M","H","H")
factor(genero)
```

```
##  [1] M H M M M H M M H H
## Levels: H M
```

Se códifica la el vector genero como factor, creando los niveles H y M.

```
factor(genero, levels = c("M","H"), labels = c("Mujer","Hombre"))
```

```
##  [1] Mujer  Hombre Mujer  Mujer  Mujer  Hombre Mujer  Mujer  Hombre Hombre
## Levels: Mujer Hombre
```

Taller Práctico

Taller Práctico

El Servicio de Evaluación Ambiental SEA es la institución encargada de autorizar la operación de proyectos en Chile, que podrían tener impactos potenciales en la salud de la población o el medio ambiente.

Cuando una empresa quiere llevar a cabo un proyecto de una magnitud relativamente grande, debe presentar un requisito a la SEA para evaluar el correcto y seguro funcionamiento de este proyecto. Si un proyecto es determinado como dañino para el medio ambiente o la población, la SEA puede denegar el permiso ambiental y, por lo tanto, la puesta en marcha de un proyecto.

Desde el inicio de la SEA en 1997, más de 25 mil proyectos han sido evaluados por esta institución, por lo que la base de datos de la SEA contiene una gran cantidad de registros, que pueden ser útiles para analizar.

Los datos del archivo “SEA_projects.xlsx” fueron extraídos el 15 de marzo de 2021 desde la página oficial del SEA, sabiendo que esta información es pública.

Descripción variables base `SEA_projects.xlsx`

Nombre	Descripción
name	Nombre del proyecto.
type	Tipo de proceso de evaluación. Los proyectos pueden presentar una declaración de impacto ambiental (DIA) o un estudio de impacto ambiental (EIA).
region	Región donde se está llevando a cabo el proyecto.
owner	Empresa dueña del proyecto.
typology	Código de tipología de proyecto en función de su sector.
investment	Monto de la inversión en millones de dólares estadounidenses.
entry_date	Fecha en la que el proyecto ingresa al proceso SEA.
state	El estado actual de la evaluación del proyecto a fecha de revisión.
qualification_date	Fecha en la que se emitió la resolución final de la SEA.

Taller Práctico

1.- Remueva las columnas y filas que solo contengan datos perdidos.

2.- Verifique que el formato de las variables y realice los cambios de formatos que sean necesarios.

3.- Obtenga un resumen de la base de datos, ¿En promedio cuantos millones de dólares se invierten en los proyectos de Chile?

4.- Realice un estudio de los datos faltantes (Na's) en las distintas variables, ¿Qué proporción de Na's se tiene para las distintas variables?

5.- Realice un análisis gráfico de los Na's de las distintas variables segregado por los distintos estado de la variable estado. ¿A que se deben estos resultados?, comente.

Referencias y material complementario

Paquete `janitor`

- Ejemplos de funciones del paquete `janitor`.

Paquete `stringr`

- Cheatsheet `stringr` (En español).

Paquete `lubridate`

- Cheatsheet `lubridate`

¡Gracias!

Ana María Alvarado Celis
amalvara@uc.cl

Maite Vergara - Esteban Rucán
maite.vergara@uc.cl - errucan@uc.cl