




Introducción a R y Tidyverse I

Sesión 02

Laboratorio de Innovación en Salud

2022-09-29

 @healthinnovation
 @innovalab_imt
 innovalab.info

Contenidos

- Importación de datos de principales formatos
- Reconocimiento de variables (función glimpse)
- Anidación de funciones con pipe (`%>%`)
- Cálculo de variables (introducción a mutate)
- Contabilización y ordenamiento en base a variables (uso de count y arrange).
- Explicación de tidy data
- Modificación a la estructura de los datos mediante pivot longer y wider

Exploración competencial

Base de datos a utilizar

Usaremos una base de datos proporcionado por Gan et al. (2018) en su estudio:

Article | [Open Access](#) | [Published: 13 June 2018](#)

Efficacy of two different dosages of levofloxacin in curing *Helicobacter pylori* infection: A Prospective, Single-Center, randomized clinical trial

[Huo-Ye Gan](#) , [Tie-Li Peng](#), [You-Ming Huang](#), [Kai-Hua Su](#), [Lin-Li Zhao](#), [Li-Ya Yao](#) & [Rong-Jiao Yang](#)

[Scientific Reports](#) **8**, Article number: 9045 (2018) | [Cite this article](#)

21k Accesses | **9** Citations | **1** Altmetric | [Metrics](#)

Abstract

Bismuth + proton pump inhibitor (PPI) + amoxicillin + levofloxacin is one of the bismuth quadruple therapy regimens widely used for the eradication of *H. pylori* infection. The recommended dosage of levofloxacin is 500 mg once daily or 200 mg twice daily to eradicate *H. pylori* infection. The aim of the present open-label, randomized control trial was to compare the effectiveness, safety, and compliance of different dosages of levofloxacin used to cure *Helicobacter pylori* infection. Eligible patients were randomly assigned to receive esomeprazole, amoxicillin, colloidal bismuth pectin and levofloxacin 500 mg once/day (group A) or levofloxacin 200 mg twice/day (group B) for 14 days. The primary outcome was the

Download PDF



Sections

Figures

References

[Abstract](#)

[Introduction](#)

[Materials and Methods](#)

[Results](#)

[Discussion](#)

[References](#)

[Author information](#)

[Ethics declarations](#)

[Additional information](#)

[Rights and permissions](#)

[About this article](#)

[Comments](#)





Base de datos a utilizar

Dicho estudio es un ensayo clínico aleatorizado (ECA) de 400 participantes en el que se compara 2 tipos de tratamientos para erradicar la infección por *Helicobácter Pylori*.

- **Grupo A** de tratamiento: Esomeprazol, amoxicilina, bismuto coloidal con pectina y levofloxacin 500mg una vez al día.
- **Grupo B** de tratamiento: Levofloxacin 200mg dos veces al día, durante un periodo de 14 días.

Randomized group	n
group A	200
group B	200

Importación de datos

Función	Tipo de archivo
<code>readxl::read_excel()</code>	
<code>readr::read_csv()</code>	
<code>haven::read_dta()</code>	
<code>haven::read_sav()</code>	

Importación de datos

Vamos a importar el archivo excel que tiene una extensión `.xlsx` y se encuentra dentro de la carpeta `data`.

Recuerda: Al importar el archivo debemos asignarlo a un objeto, para poder guardar la información. En este caso llamaremos a este objeto como `trial_data`.

```
trial_data <- readxl::read_excel("data/researchdata.xlsx")

trial_data
```

```
## # A tibble: 400 × 10
##   Patient numbe...1 Basel...2 Rando...3 Follo...4 adver...5 adver...6 Adver...7 Compl...8 Uncom...9 per p...*
##           <dbl> <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <chr>
## 1             1 Positi... group B Negati... No      NA      NA      Yes      NA      Yes
## 2             2 Positi... group A Negati... Yes    Fatigue Mild    Yes      NA      Yes
## 3             3 Positi... group B Negati... No      NA      NA      Yes      NA      Yes
## 4             4 Positi... group A Positi... Yes    Abdomi... Mild    Yes      NA      Yes
## 5             5 Positi... group A Negati... Yes    Drowsi... Mild    Yes      NA      Yes
## ...
```

Importación de datos 2


En algunos casos estaremos frente a base de datos que contengan cierta información en sus primeras filas que no sean relevantes como datos a considerar para el análisis. La mayor cantidad de veces tienen un propósito meramente informativo. Ejemplo:

researchdata2.xlsx

Home
 Insert
 Layout
 Formula
 Data
 Pivot Table
 Collaboration
 Protection
 View
 Plugins

Times New Roman
 11
 A⁺
 A⁻
 T
 B
 I
 U
 A₂
 A
 General
 Normal
 Neutral
 Bad

I18
 NA

	B	C	D	E	F	G	H	I
1	<div>  </div>							
2								
3								
4	Baseline ¹³ C-UBT	Randomized group	Follow-up ¹³ C-UBT (4 weeks after therapy)	adverse drug reactions	adverse drug reactions content	Adverse drug reaction classified	Complete the study	Uncompleted Reason
5	Positive	group B	Negative	No	NA	NA	Yes	NA
6	Positive	group A	Negative	Yes	Fatigue	Mild	Yes	NA
7	Positive	group B	Negative	No	NA	NA	Yes	NA
8	Positive	group A	Positive	Yes	Abdominal pain	Mild	Yes	NA
9	Positive	group A	Negative	Yes	Drowsiness	Mild	Yes	NA
10	Positive	group A	Negative	No	NA	NA	Yes	NA

Importación de datos 2

Situación	Problema	Solución
-----------	----------	----------

En estas situaciones tenemos 2 alternativas:

a) Editar el archivo en cuestión y eliminar las filas que no sean relevantes. b) Durante la importación indicar que se omitan las primeras filas o filtrarlas una vez se haya importado.

Siempre será una mejor opción manejar los cambios desde el código, ya que de esa manera mantenemos los archivos originales y además podemos tener un registro de los cambios realizados.

Para hacer esto usaremos el argumento `skip` dentro de la función `readxl::read_excel()`, que indicará la cantidad de filas que deseamos omitir en la importación.

¡Hazlo tú mismo!

Todas las importaciones requeridas serán asignadas en un objeto.

1. Realizar la importación del archivo excel
2. Realizar la importación del archivo csv
3. Realizar la importación del archivo spss

08:00

Función Glimpse

¿Para qué sirve? Ejemplo Glimpse

- Versión transpuesta de print
- Ayuda a visualizar la mayor cantidad de datos de muchas columnas.
- Muestra el nombre de la variable junto con una designación de tipo de variable.

Mediante la función:

```
glimpse()
```

Recordar importar el paquete tidyverse

```
library(tidyverse)
```

```
## — Attaching packages —————  
## ✓ ggplot2 3.3.6      ✓ purrr 0.3.4  
## ✓ tibble 3.1.8       ✓ stringr 1.4.1  
## ✓ tidyr 1.2.1        ✓ forcats 0.5.2  
## ✓ readr 2.1.2
```

Operador Pipe %>%

Queremos aplicar más de una función: El uso de funciones de forma anidada puede resultar ilegible o difícil de comprender.

```
tabla(formato(coeficiente(data)))
```

El operador %>% nos permite escribir una secuencia de operaciones de izquierda a derecha:

```
coeficiente(data) %>% formato() %>% tabla()
```

O mejor aún:

```
coeficiente(data) %>%  
  formato() %>%  
  tabla()
```

Útil para concatenar múltiples operaciones en dplyr

Atajo de Teclado

Cmd + Shift + M (Mac)

Ctrl + Shift + M (Windows/Linux)

Función `mutate()`

Con `mutate()` podemos realizar modificaciones en las variables. Por ej. sumar variables, o modificarlas de alguna manera (transformar a porcentaje, multiplicarlas por alguna constante, etc.). Estas modificaciones pueden realizarse:

- Creando una nueva variable a partir de otras ya existentes.
- Modificar una variable existente en la misma variable.

Ejemplo:

```
library(tidyverse)

df %>%
  mutate(
    New_var = var*2
  )
```

Explicación:

En una data llamada `df` se estaría aplicando la función `mutate` creando una variable llamada `New_var` a partir de otra variable llamada `var` que está siendo multiplicada por 2.

Uso práctico de mutate()

Reconocimiento de BD Janitor Rename Mutate I Mutate II

Para esta ejemplificación continuaremos usando la base de datos del ECA sobre la erradicación de la infección por *Helicobácter Pylori*.

```
trial_data ← readxl::read_excel("data/research_data/
trial_data")
```

```
## # A tibble: 400 × 10
##   Patient ...1 Basel...2 Rando...3 Follo...4 adver.
##           <dbl> <chr>      <chr>      <chr>      <chr>
## 1             1 Positi... group B Negati... No
## 2             2 Positi... group A Negati... Yes
## 3             3 Positi... group B Negati... No
## 4             4 Positi... group A Positi... Yes
## 5             5 Positi... group A Negati... Yes
## 6             6 Positi... group A Negati... No
## 7             7 Positi... group B Negati... No
```

```
glimpse(trial_data)
```

```
## Rows: 400
## Columns: 10
## $ `Patient number`
## $ `Baseline 13C-UBT`
## $ `Randomized group`
## $ `Follow-up13C-UBT (4 weeks after therapy)`
## $ `adverse drug reactions`
## $ `adverse drug reactions content`
## $ `Adverse drug reaction classified`
## $ `Complete the study`
## $ `Uncompleted Reason`
```

Uso de slice_max()

Anteriormente conocido como `top_n()`, la función `slice_max()`, permite seleccionar cuantos casos (filas) se especifique en base a una variable numérica.

```
trial_data %>%  
  count(randomized_group,  
         follow_4_weeks) %>%  
  slice_max(order_by = n,  
            n = 2)
```

```
## # A tibble: 2 × 3  
##   randomized_group follow_4_weeks     n  
##   <chr>           <chr>         <int>  
## 1 group B        Negative        159  
## 2 group A        Negative        155
```

¡Hazlo tú mismo!

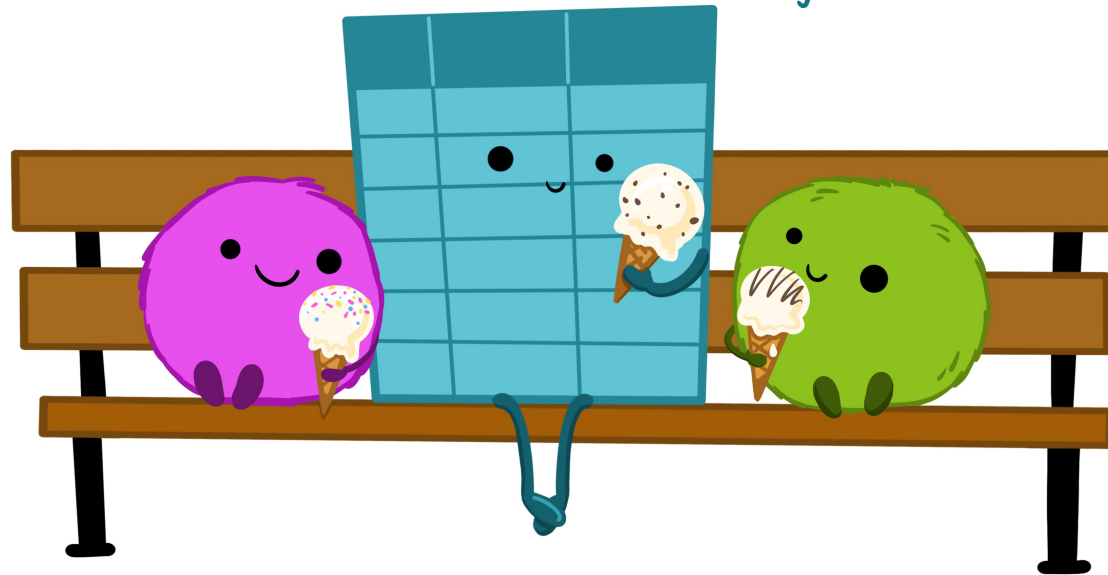


1. Generar una tabla de frecuencias para conocer cuantos de los que completaron el estudio, presentaron reacciones adversas al tratamiento
2. Complementar la tabla del anterior punto agregando la variable del grupo de tratamiento al que pertenece.

08:00

Introducción a tidydata

make friends with tidy data.



Introducción a tidydata

Tener datos ordenados (**tidydata**) significa tener una BD con estructuras adecuadas, donde cada registro representa una fila, cada variable representa una columna y cada celda contiene una simple medida (**Hadley Wickham, 2014**).

“**TIDY DATA** is a standard way of mapping the meaning of a dataset to its structure.”

—HADLEY WICKHAM

In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement

each column a variable

id	name	color
1	floof	gray
2	max	black
3	cat	orange
4	donut	gray
5	merlin	black
6	panda	calico

each row an observation

Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

Uso de pivot_wider()

De forma análoga, la función `pivot_wider()` hará exactamente lo contrario, pasar de una data que se encuentre ordenada (**tidy data**), a una data ancha. Para esta función se tendrá que indicar los argumentos: `names_from` y `values_from`.

```
covid19_tidy %>%
```

```
  pivot_wider(
    names_from = "fecha",
    values_from = "fallecidos"
  )
```

```
## # A tibble: 289 × 983
##   Province/S...1 Count...2 1/22/...3 1/23/...4 1/24/...5 1/25/...6
##   <chr>          <chr>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 <NA>          Afghan...      0        0        0        0
## 2 <NA>          Albania      0        0        0        0
## 3 <NA>          Algeria      0        0        0        0
## 4 <NA>          Andorra      0        0        0        0
## 5 <NA>          Angola      0        0        0        0
## 6 <NA>          Antarc...      0        0        0        0
## 7 <NA>          Antigu...      0        0        0        0
## 8 <NA>          Argent...      0        0        0        0
## 9 <NA>          Armenia      0        0        0        0
## 10 Australian ... Austra...      0        0        0        0
## #   with 270 more rows  977 more variables.
```



¡Hazlo tú mismo!

1. Replica los procedimientos observados en las diapositivas
2. Genera un objeto tidy (`pivot_longer`)
3. Genera un objeto wider (`pivot_wider`)

08:00



InnovaLab

Retroalimentación

¡Gracias!

✉ **imt.innovlab@oficinas-upch.pe**

🐙 **@healthinnovation**

🐦 **@innovalab_imt**

Estas diapositivas fueron creadas mediante el paquete `xaringan` y `xaringantheme`.