

# Tema IV: Manipulación de Datos

## Desafío 4: Ácido Fólico – Datos que Nutren

Dr. Maicel Monzón Pérez

2025-03-24

### Índice

<b>El Ácido Fólico, un protector de información genética (ADN)</b>	<b>2</b>
<b>Objetivo de la práctica</b>	<b>2</b>
<b>Instrucciones</b>	<b>2</b>
<b>Materiales</b>	<b>3</b>
<b>Tareas 1: Filtrar con una variable</b>	<b>3</b>
<b>Tareas 2: Obtener un subconjuntos con una condición lógica</b>	<b>4</b>
<b>Tareas 3: Empleo de múltiples funciones (group_by,summarise y arrange)</b>	<b>4</b>
<b>Tareas 4: Identificar subconjuntos con dos condiciones</b>	<b>5</b>
<b>Tareas 5: Calcular una nueva variable</b>	<b>5</b>
<b>Tareas 6: Selección de subconjuntos con una condición lógica</b>	<b>5</b>
<b>Tareas 7: Realizar resúmenes de datos con summarise()</b>	<b>6</b>
<b>Tareas 8: Combinar Funciones</b>	<b>6</b>
<b>Tareas 7: Crear un resumen por alimento:</b>	<b>6</b>
<b>Conclusiones</b>	<b>7</b>

## El Ácido Fólico, un protector de información genética (ADN)

El **ácido fólico** y **dplyr** comparten un rol fundamental como arquitectos de estructuras saludables. Así como el **folato** está implicado en la síntesis de ADN para transmitir información genética precisa, **mutate()** construye variables que mejoran la lógica del análisis.

La división celular organizada, guiada por el folato, es similar a **group\_by()** y **summarise()**, que **dividen datos en grupos independientes** y extraen patrones clave.

En la prevención, mientras el folato filtra errores en el desarrollo fetal (Ej. defectos del tubo neural), **filter()** actúa como sistema inmunológico analítico, eliminando datos corruptos o sesgados.

Ambos son catalizadores de procesos críticos: **el folato en el metabolismo celular, dplyr en el flujo de trabajo analítico**. Su poder radica en intervenciones precisas, **transformando materia prima (células o datos)** en sistemas funcionales y resilientes.

---

### Objetivo de la práctica

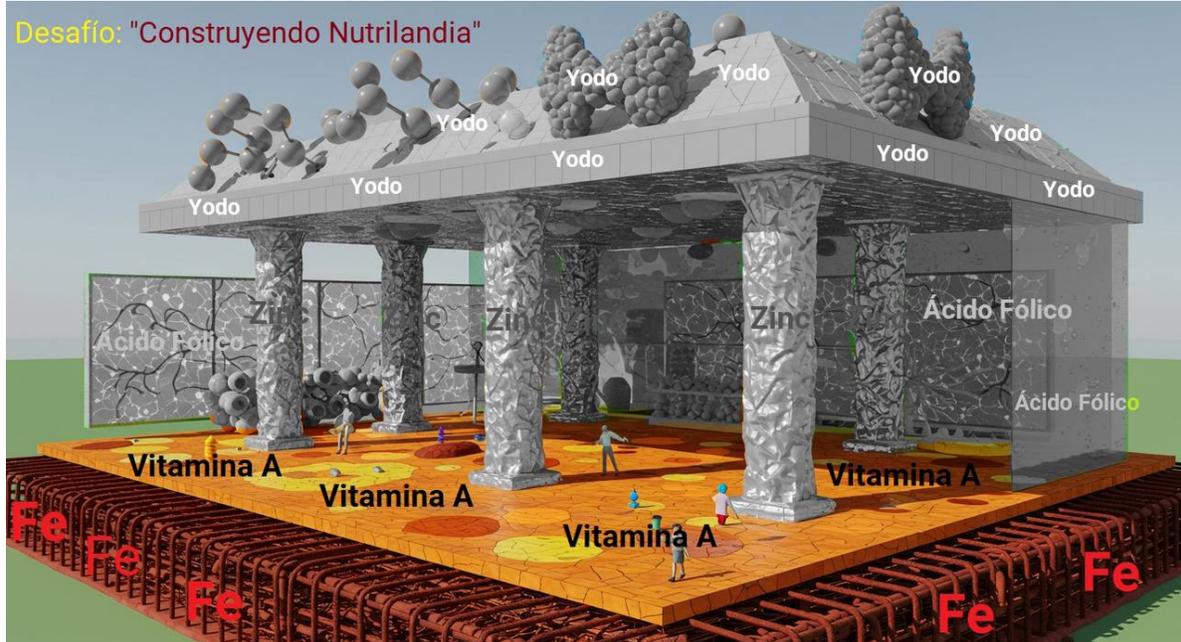
1. Importar datos desde múltiples formatos (CSV, Excel, bases de datos) usando herramientas como `readr`, sentando las bases para operaciones avanzadas.
2. Aplicar verbos clave (`mutate()`, `filter()`, `group_by()`) para transformar datos crudos en marcos limpios y listos para análisis.
3. Validar la calidad de los datos (valores faltantes, inconsistencias) mediante operaciones como `summarise()` y `count()`, asegurando que cada “bloque” del dataset cumpla estándares rigurosos.
4. Encadenar operaciones con el operador `%>%` para crear pipelines reproducibles, imitando los planos detallados de un proyecto de ingeniería.

### Instrucciones

- Completa todas las tareas para obtener **5 puntos**.
- Usa el formato **R Markdown** o **R script** para entregar tus respuestas.

### TIPS:

- use la función `view(nutricion_fortificacion)` y `glimpse(nutricion_fortificacion)` para ver la estructura del dataframe.
- consulte la ayuda con la función `?funcion_objetivo`, en internet en el sitio <https://rseek.org/>, busque las **vignette** de la biblioteca **dplyr**



## Materiales

- Datos de ejemplo para importar:
  - `datos_fortificacion.csv`
  - `nutricion_subset.csv` para solución de la clase anterior
- Recuerde convertir la variable `hierro_mg` de cualitativa por carácter usado como NA “-” a cuantitativa (tipo vector numérico)

## Tareas 1: Filtrar con una variable

¿Qué alimento tienen menos de 10 microgramos de hierro?

Filtrar alimentos con hierro > 10 mcg: `filter()`

### Solución:

```
# Filtrar hierro_mg > 10 mcg
alto_hierro <- datos_fortificacion %>%
  filter(hierro_mg > 10) %>%
  select(alimento, hierro_mg)
# view(alto_hierro)
```

## Tareas 2: Obtener un subconjuntos con una condición lógica

- Se obtiene un subconjunto datos para el análisis solo de la vitamina A como micronutriente (Nueva solución para ejercicio de la clase anterior)

### Solución:

```
# Filtrar hierro_mg > 10 mcg
Vitamina_A <- nutricion_subset %>%
  filter(Nutriente == "Vitamina_A")
# view(Vitamina_A) para ver el resultado
```

## Tareas 3: Empleo de múltiples funciones (group\_by, summarise y arrange)

“¿Qué alimentos podrían ser fortificados con hierro?”

- Agrupe por alimentos y calcule la media de aritmética de consumo de hierro por alimentos.
- Ordene los valores del promedio de manera descendente para encontrar los de menor valor.

### Solución:

```
#calculo de la media de MCG de Hierro por alimentos
alimentos_fortificar_promedio <- datos_fortificacion %>%
  group_by(alimento) %>% # para agrupar
  summarise(promedio_hierro=mean(hierro_mg)) %>%
  arrange(promedio_hierro) # orden
# note: los valores son simulados no se usaron datos reales
```

## Tareas 4: Identificar subconjuntos con dos condiciones

“Filtrar alimentos con zinc < 5 mcg Y consumen alimentos como la”Harina de trigo”. ¿Cuántos registros quedan?”

**TIPS:** utilice la función `nrow()` para identificar el número de filas del dataframe

```
alimentos_fortificar <- datos_fortificacion %>%  
  filter(hierro_mg<5,alimento=="Harina de trigo")  
# muestra el numero de filas  
nrow(alimentos_fortificar)
```

```
[1] 29
```

## Tareas 5: Calcular una nueva variable

- Calcular la relación hierro/zinc en cada alimento con la función `mutate()` .

TIP:

Crear una nueva variable `hierro_zinc_ratio`:

```
datos_fortificacion <- datos_fortificacion %>%  
  mutate(hierro_zinc_ratio = hierro_mg / zinc_mg)
```

## Tareas 6: Selección de subconjuntos con una condición lógica

-Calcular cuantos hombres adultos con déficit relativo de hierro si se usa un `hierro_zinc_ratio` menor o igual a **0.67**

```
hombres_deficit <- datos_fortificacion %>%  
  filter(hierro_zinc_ratio <= 0.67)  
nrow(hombres_deficit)
```

```
[1] 109
```

## Tareas 7: Realizar resúmenes de datos con summarise()

- Calcular promedios de zinc por región.

Solución:

```
resumen_zinc <- datos_fortificacion %>%
  group_by(region) %>%
  summarise(media_zinc = mean(zinc_mg, na.rm = TRUE))

# print(resumen_zinc)
```

## Tareas 8: Combinar Funciones

Filtrar alimentos con hierro < 8 mg Y zinc < 5 mg:

Solución:

```
alimentos_criticos <- datos_fortificacion %>%
  filter(hierro_mg < 8, zinc_mg < 5) %>%
  select(alimento, region, hierro_mg, zinc_mg)

# print(alimentos_criticos)
```

## Tareas 7: Crear un resumen por alimento:

Cree un resumen por alimentos para las variables hierro y zinc.

Solución:

```
resumen_nutricional <- datos_fortificacion %>%
  group_by(alimento) %>%
  summarise(
    media_hierro = mean(hierro_mg, na.rm = TRUE),
    media_zinc = mean(zinc_mg, na.rm = TRUE))

#print(resumen_nutricional)
```

## Conclusiones

Este ejercicio reforzó habilidades analíticas en un contexto de fortificación de alimentos a gran escala, demostrando cómo herramientas de dplyr permiten convertir datos crudos en conjuntos listos para el análisis de datos.

Desafío completado: Ácido Fólico – Datos que Nutren

