

Tema III - Datos Ordenados con tidyr

Desafío 3: Vitamina A – Claridad en la Oscuridad

Dr. Maicel Monzón Pérez

2025-03-24

Índice

La Vitamina A, un Micronutriente fundamental	1
Objetivo de la práctica	2
Instrucciones	2
Materiales	3
Tarea 1: Convertir de ancho a largo con pivot_longer	3
Tarea 2: Separar columnas combinadas con separate	4
Tarea 3: Unir columnas con unite	4
Tarea 4: Convertir de largo a ancho con pivot_wider	5
Tarea 5: Resolver datos desordenados múltiples	6
Conclusión	7
Desafío completado: Claridad en la Oscuridad	8

La Vitamina A, un Micronutriente fundamental

La vitamina A (*retinol*, *ácido retinoico*) es un micronutriente clave para:

- **Visión nocturna** : Contribuye a la síntesis de melanina en la retina

- **Sistema inmunológico** : Refuerza la resistencia a infecciones
- **Crecimiento celular** : Esencial para el desarrollo de tejidos y órganos Por su parte, la deficiencia :
- *Puede causar ceguera nocturna ,*
- *Síndrome diarreico*
- *Susceptibilidad a las infecciones*

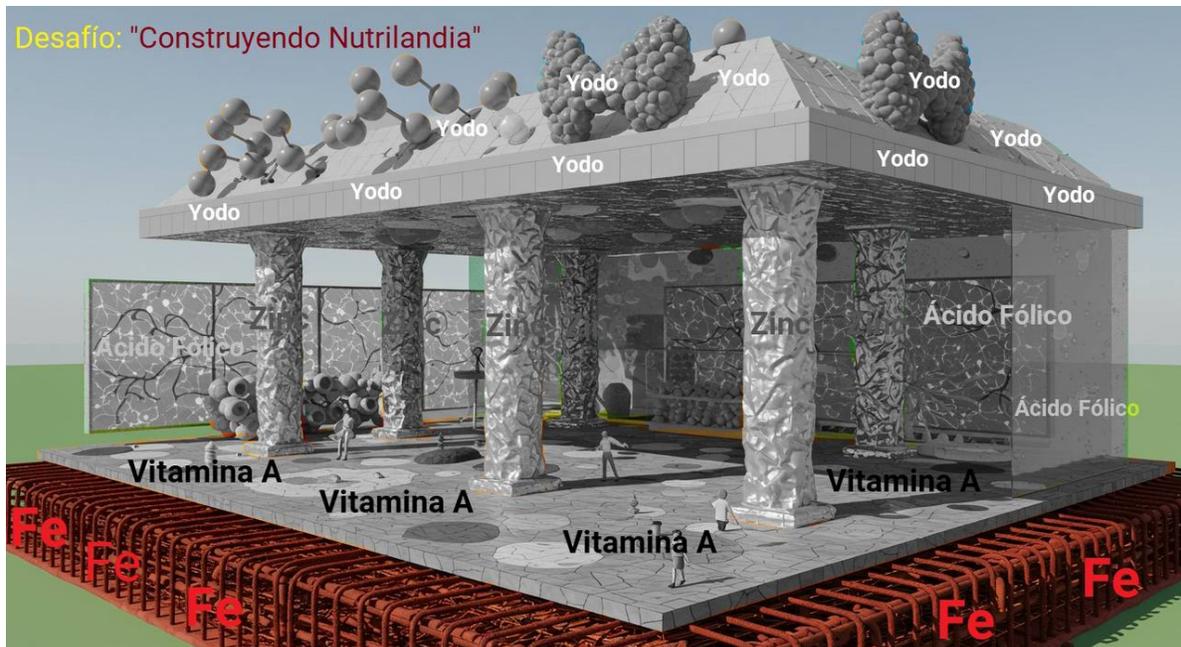
En contextos de la **modelación de dietas**, es crítico **organizar los datos** de manera clara para identificar áreas con déficit de vitamina A. Las vitaminas potencian el metabolismo celular, los datos “**tidy**” potencian la **eficiencia analítica**.

Objetivo de la práctica

1. Aplica principios de datos *tidy* (cada variable en una columna, cada observación en una fila).
2. Usar *tidyr* para **transformar** datos “**ancho**” a “**largo**” y viceversa.
3. **Separar** y **unir** columnas combinadas.

Instrucciones

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



Materiales

- Datos de ejemplo para importar:
 - nutricion_fortificacion.csv

TIPS: use la función `view(nutricion_fortificacion)` y `glimpse(nutricion_fortificacion)` para ver la estructura del dataframe

Antes debemos importar los datos

Tarea 1: Convertir de ancho a largo con `pivot_longer`

Las vitaminas están en columnas separadas. Necesitamos una columna Nutriente y otra Valor. Realice la transformación apropiada guarde el dataframe con el nombre `nutricion_largo`.

Solución:

```
library(tidyr)

nutricion_largo <- pivot_longer(
  data = nutricion_fortificacion,
  cols = c(Vitamina_A, Vitamina_D, Hierro),
  names_to = "Nutriente",
  values_to = "Valor"
)
# view(nutricion_largo)
```

Tarea 2: Separar columnas combinadas con separate

La columna `Alimento_Comida` combina el nombre del alimento y su tipo (ej: “Arroz_Industrial”). Realice la transformación apropiada guarde el dataframe con el nombre *nutricion_separada*.

Solución:

```
nutricion_separada <- separate(
  data = nutricion_largo,
  col = Alimento_Comida,
  into = c("Alimento", "Tipo"),
  sep = "_"
)
# view(nutricion_separada)
```

Tarea 3: Unir columnas con unite

La columna `Ciudad_Pais` está combinada (ej: “Quito_Ecuador”). Separarla en Ciudad y País, luego unir en Ubicación. Realice la transformación apropiada guarde el dataframe con el nombre *nutricion_unida*.

Solución:

```

# Primero separar Ciudad y País
nutricion_separada <- separate(
  data = nutricion_separada,
  col = Ciudad_País,
  into = c("Ciudad", "País"),
  sep = "_"
)

# Luego unir en una sola columna
nutricion_unida <- unite(
  data = nutricion_separada,
  col = "Ubicación",
  Ciudad, País,
  sep = ", ",
  remove = TRUE
)

```

Tarea 4: Convertir de largo a ancho con `pivot_wider`

Crear un formato ancho para comparar promedios de vitamina A por país y tipo de alimento. Realice la transformación apropiada guarde el dataframe con el nombre *nutricion_ancho*.

Nota: Debe realizar previamente la selección de un subconjunto, hay múltiples maneras de hacerlo, en el tema IV estudiaremos una que resulta muy práctica con la función **filter** de la biblioteca **dplyr**. Por el momento, inténtelo con la información recibida hasta este punto con **Base R** (selección basada en condiciones lógicas).

Tip: Puede consultar en el buscador <https://rseek.org/> maneras de hacer **subsetting** también en las **hojas de trucos**.

Bibliografía:

<https://cran.r-project.org/doc/manuals/r-release/R-intro.html#Index-vectors>

Solución:

```

# Filtrar solo vitamina A
vitamina_a <- nutricion_unida[nutricion_unida$Nutriente=="Vitamina_A",]

# Convertir a ancho
nutricion_ancho <- pivot_wider(

```

```

data = vitamina_a,
id_cols = c(Ubicación, Tipo),
names_from = Nutriente,
values_from = Valor
)

```

Tarea 5: Resolver datos desordenados múltiples

El dataset `nutricion_fortificacion` tiene:

- Columnas combinadas (ej: `Alimento_Comida`).
- Datos en formato ancho para vitaminas.
- Valores faltantes en `Vitamina_A`.

Realice la transformación apropiada para obtener el siguiente dataframe:

ID_Hogar	Alimento	Tipo	Fecha_Ing_Capturada	Ubicación	Nutriente	Valor
1	Arroz	Industrial	2023-08-13	Santiago, Chile	Vitamina_A	131.0
2	Arroz	Industrial	2023-08-13	Santiago, Chile	Vitamina_D	NA
3	Arroz	Industrial	2023-08-13	Santiago, Chile	Hierro	2.9
4	Arroz	Industrial	2023-07-25	Lima, Perú	Vitamina_A	457.0
5	Arroz	Industrial	2023-07-25	Lima, Perú	Vitamina_D	243.0
6	Arroz	Industrial	2023-07-25	Lima, Perú	Hierro	1.7
7	Arroz	Industrial	2023-12-14	Santiago, Chile	Vitamina_A	418.0
8	Arroz	Industrial	2023-12-14	Santiago, Chile	Vitamina_D	144.0
9	Arroz	Industrial	2023-12-14	Santiago, Chile	Hierro	0.7
10	Arroz	Industrial	2023-10-16	Lima, Perú	Vitamina_A	451.0

Solución:

```

# Separar columnas combinadas
nutricion <- separate(
  data = nutricion_fortificacion,
  col = Ciudad_Pais,
  into = c("Ciudad", "País"),
  sep = "_"
) %>%
  separate(

```

```

    col = Alimento_Comida,
    into = c("Alimento", "Tipo"),
    sep = "_"
  )

# Convertir vitaminas a formato largo
nutricion_largo <- pivot_longer(
  data = nutricion,
  cols = c(Vitamina_A, Vitamina_D, Hierro),
  names_to = "Nutriente",
  values_to = "Valor"
)

# Eliminar NA en Vitamina_A y unir Ubicación
nutricion_final <- unite(
  data = nutricion_largo,
  col = "Ubicación",
  Ciudad, País,
  sep = ", ",
  remove = TRUE
)

```

Conclusión

- ¿Por qué es importante usar `pivot_longer` y `pivot_wider`?

Respuesta: Permiten transformar datos entre formatos ancho/largo para facilitar análisis específicos (ej: comparar nutrientes).

- ¿Qué problema soluciona `separate` y `unite`?

Respuesta: `separate` descompone columnas combinadas, mientras que `unite` combina columnas para crear identificadores únicos (ej: "Ciudad, País").

Desafío completado: Claridad en la Oscuridad

