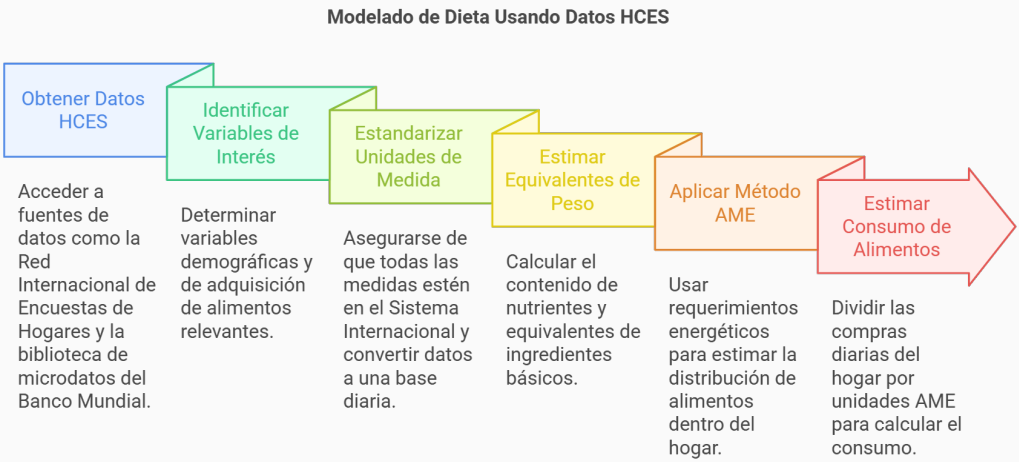
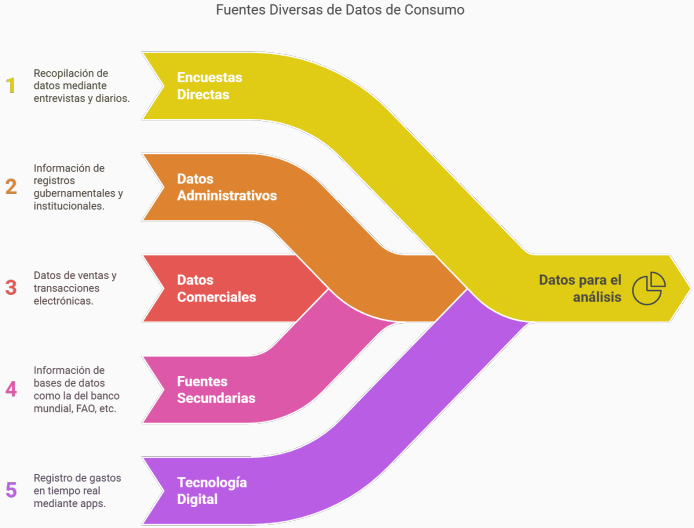
Tema III: Datos Ordenados.

Maicel Monzón

Para Modelar La Dieta Se Requiere Datos Precisos Y Fáciles De Analizar



Se abordarán métodos para estructurar datos desordenados.



Sumario

- ¿Qué son los Datos Ordenados?
- ¿Por qué son Importantes?
- Datos Desordenados Comunes
- Funciones Clave de la biblioteca `tidyr`

1. `pivot_longer()`
2. `pivot_wider()`
3. `separate()`
4. `unite()`

Analogía

"Todas las **familias felices se parecen** unas a otras, pero cada **familia infeliz lo es a su manera**." *León Tolstoy*

"Todos los **conjuntos de datos ordenados se parecen** unos a otros, pero cada conjunto de **datos desordenado lo es a su manera**" *Hadley Wickham*

tidydata: Un estándar para organizar datos de manera consistente.

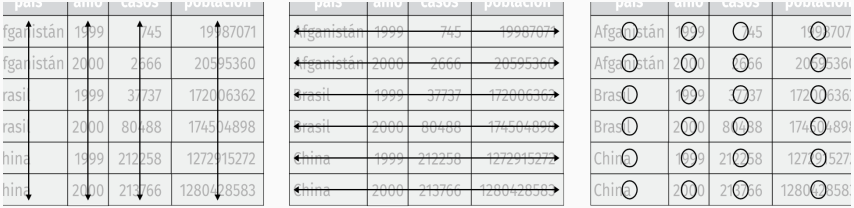
Existen **tres principios** que definen un conjunto de datos ordenados.

Table 1: conjunto de datos

Col_ID	var 1	var 2	var 3
obs 1	**valor** 1,1	**valor** 1,2	**valor** 1,3
obs ...	**valor** .,1	**valor** .,2	**valor** .,3
obs n	**valor** n,1	**valor** n,2	**valor** n,3

Principios de Tidydata (datos ordenados)

1. Cada **variable** debe tener su **propia columna**.
2. Cada **observación** debe tener su **propia fila**.
3. Cada **valor** debe tener su **propia celda**.



Ejemplo de datos ordenados

Ej. distribución de ingresos y miembros por hogares

```
# A tibble: 3 x 3
  ID_Hogar Ingreso Miembros
     <dbl>   <dbl>    <dbl>
1      101    8500        4
2      102   12000        4
3      102   12000        3
```

Ventajas de datos ordenados

- **Consistencia:** Facilita el aprendizaje.
- **Compatibilidad:** `dplyr`, `gtsummary`, y otros paquetes del tidyverse están diseñados para trabajar con datos ordenados.
- **Eficiencia:** Reduce errores y facilita la manipulación, visualización y análisis.

El enfoque facilita crear nuevas variables (percapita)

$$pct = Ingreso/Miembros$$

```
hogares_tidy %>%
  mutate(pct=Ingreso/Miembros)
```

```
# A tibble: 3 x 4
  ID_Hogar Ingreso Miembros   pct
     <dbl>   <dbl>    <dbl> <dbl>
1      101    8500        4  2125
2      102   12000        4  3000
3      102   12000        3  4000
```

Pero los datos puden estar desodenados

Orígenes del desorden de datos

Formas comunes de datos desordenados

- **Problema 1:** Una variable distribuida en múltiples columnas.
- **Problema 2:** Una observación dispersa en múltiples filas.
- **Problema 3:** Múltiples variables almacenadas en una columna.
- **Problema 4:** Diferentes tipos de datos almacenados en la misma columna.
- **Problema 5:** Variables almacenadas tanto en filas como en columnas.

Problema.1 La variable tiempo está distribuida en múltiples columnas

- Se requiere **pivotar** de **formato ancho** a **formato largo**

Datos desordenados

1: Variable distribuida en múltiples columnas

Alimento	Enero	Febrero	Marzo
Arroz	150	140	160
Pollo	100	90	110
Verduras	200	180	210

Formato ancho

Formato Largo

Datos ordenados

Alimento	Mes	Gramos
Arroz	Enero	150
Arroz	Febrero	140
Arroz	Marzo	160
Pollo	Enero	100
Pollo	Febrero	90
Pollo	Marzo	110
Verduras	Enero	200
Verduras	Febrero	180
Verduras	Marzo	210

Problema.2 La variable nutriente para un individuo específico está dispersa en múltiples filas

- Se requiere **pivotar** de **formato largo** a **formato ancho**

Datos desordenados

2: Una observación dispersa en múltiples filas.

Formato Largo

ID	nutriente	cantidad
101	Proteína	70
101	Carbohidratos	250
102	Proteína	60
102	Carbohidratos	300

Datos ordenados

ID	Proteína	Carbohidratos
101	70	250
102	60	300

Formato ancho

Problema.3 Múltiples variables almacenadas en una misma columna.

- Alimento y comida almacenadas en la variable **Alimento_Comida** separada por un guión requiere separar la variable.

Datos desordenados

3: Múltiples variables almacenadas en una columna.

Alimento_Comida	Gramos
Leche_Desayuno	200
Pan_Cena	50
Carne_Almuero	150

separar

Datos ordenados

Alimento	Comida	Gramos
Leche	Desayuno	200
Pan	Cena	50
Carne	Almuero	150

Problema 4: Diferentes tipos de datos almacenados en la misma columna.

Datos desordenados

4: Diferentes tipos de datos almacenados en la misma columna.

ID	año	mes	dia
1	2,023	10	26
2	2,023	11	15
3	2,024	1	1

unir

Datos ordenados

```
# A tibble: 3 × 2
     ID fecha
  <dbl> <chr>
1     1 2023-10-26
2     2 2023-11-15
3     3 2024-1-1
```

Introducción a `tidyr`

- **`tidyr`**: Un paquete clave en el **`tidyverse`** para ordenar datos desordenados.
- funciones principales
- pivot_longer()
- pivot_wider ()
- separate
- unite

funcion de `tidyr`: `pivot_longer()`

- **Función:** Convierte datos “anchos” a “largos”.
- **Uso:** Cuando los nombres de las columnas son valores, no variables.

Agumentos Clave de `pivot_longer()`

- `cols`: Columnas a pivotar. Selecciona las columnas a transformar.
- `names_to`: Nombre de la nueva columna para los nombres de las columnas originales. Define el nombre de la columna que contendrá los nombres de las columnas originales.
- `values_to`: Nombre de la nueva columna para los valores de las celdas. Define el nombre de la columna que contendrá los valores de las celdas.

funcion de `tidyr`: `pivot_wider()`

- **Función:** Convierte datos “largos” a “anchos”.
- **Uso:** Observación dispersa en filas.

Agumentos Clave de `pivot_wider()`

- `names_from`: Columna para nombres de nuevas columnas.
- `values_from`: Columna para llenar las nuevas columnas.

funcion de `tidyr`: `separate()`

- **Función:** Divide una columna en múltiples columnas
- **Uso:** Cuando una columna contiene múltiples variables combinadas

Agumentos Clave de `separate()`

- `cols` : La columna que se va a separar
- `into` : Vector de nombres para las nuevas columnas
- `sep`: El carácter o la posición donde se va a separar la columna. Ej “_”

funcion de `tidyr`: `unite()`

- **Función:** Combina múltiples columnas en una sola columna
- **Uso:** Cuando los componentes de una sola variable están dispersos en múltiples columnas

Agumentos Clave de `unite()`

- `cols` : El nombre de la nueva columna combinada
- `...` : Las columnas que se van a unir
- `sep`: El carácter separado

-`remove`: Eliminar las columnas de entrada