

## Módulo 2 — Tidyverse I

`group_by()` y `summarise()`

CEPAL - Unidad de Estadísticas Sociales

2025-10-31

# Introducción

El objetivo de esta sección es **aprender a agrupar y resumir información** usando las funciones `group_by()` y `summarise()`.

Estas herramientas permiten obtener **estadísticas agregadas por categorías** o combinaciones de variables, dentro del flujo natural del *tidyverse*.

## Lectura de base de ejemplos

```
datos <- readRDS("data/base_personas_gasto.rds")
library(dplyr)
datos %>% mutate(log_ingreso = log(ingreso) ) %>%
  select(id_hogar, id_pers, ingreso, log_ingreso) %>% head(4)
```

id_hogar	id_pers	ingreso	log_ingreso
262	1	5391.527	8.592584
262	2	5391.527	8.592584
265	1	7077.083	8.864617
265	2	7105.557	8.868632

## Sintaxis general

`group_by()` define los grupos y `summarise()` resume las variables.

```
datos %>%  
  group_by(variable_grupo) %>%  
  summarise(  
    nueva_variable = funcion(variable_interés, na.rm = TRUE)  
  )
```

## Promedio del ingreso por área

```
datos %>%  
  group_by(area) %>%  
  summarise(media_ingreso = mean(ingreso, na.rm = TRUE))
```

area	media_ingreso
1	3677.658
2	2436.735

## Combinando select() y filter()

Integración: selección, filtrado y resumen en una misma secuencia.

```
datos %>%  
  select(area, sexo, ingreso) %>%  
  filter(ingreso > 0) %>%  
  group_by(area, sexo) %>%  
  summarise(media_ingreso = mean(ingreso), .groups = "drop")
```

area	sexo	media_ingreso
1	Hombre	3796.482
1	Mujer	3594.350
2	Hombre	2600.428
2	Mujer	2296.654

## Múltiples estadísticas

Se pueden generar **múltiples medidas resumen** en un solo paso.

```
datos %>%  
  group_by(niveduc_ee) %>%  
  summarise(  
    media_ingreso = mean(ingreso),  
    mediana_ingreso = median(ingreso),  
    sd_ingreso = sd(ingreso) )
```

niveduc_ee	media_ingreso	mediana_ingreso	sd_ingreso
1	2497.282	1840.673	2836.007
2	2553.615	2063.079	2480.526
3	2989.516	1844.828	11910.837
4	2598.814	1997.672	2479.123
5	3157.651	2319.757	4296.941
6	4443.027	3082.253	4880.921
7	9334.354	6082.638	14510.925
NA	2296.833	1853.527	1733.412

## Uso de across()

Permite aplicar funciones a **varias columnas simultáneamente**.

```
datos %>%  
  group_by(area) %>%  
  summarise(  
    across(c(ingreso, gasto), mean, .names = "media_{.col}")  
  )
```

area	media_ingreso	media_gasto
1	3677.658	3546.095
2	2436.735	2414.815

## Incorporando mutate() + ifelse()

Ejemplo de creación previa de una variable categórica antes del agrupamiento.

```
datos %>%  
  mutate(tipo = ifelse(ingreso > 10000, "Alto", "Bajo")) %>%  
  group_by(tipo) %>%  
  summarise(media_gasto = mean(gasto))
```

tipo	media_gasto
Alto	16693.711
Bajo	2660.821

## Agrupación tras case\_when()

Muestra cómo agrupar por clasificaciones múltiples.

```
datos %>%  
  mutate(  
    grupo_edad = case_when(  
      edad < 18 ~ "Menor",  
      edad >= 18 & edad < 60 ~ "Adulto",  
      TRUE ~ "Mayor"  
    )  
  ) %>%  
  group_by(grupo_edad) %>%  
  summarise(media_ingreso = mean(ingreso))
```

grupo_edad	media_ingreso
Adulto	3649.755
Mayor	3723.619
Menor	2648.083

## Mutate + summarise en secuencia

Crea una nueva variable antes de resumir por grupo.

```
datos %>%  
  filter(id_pers == 1) %>%  
  mutate(razon_hh = gasto_hh/ingreso_hh) %>%  
  group_by(niveduc_ee) %>%  
  summarise(media_razon_hh = mean(razon_hh, na.rm = TRUE))
```

niveduc_ee	media_razon_hh
1	1.0000000
2	1.0000000
3	1.0000000
4	0.9993295
5	0.9984701
6	0.9959953
7	0.9812437
NA	1.0000000

# Agrupaciones múltiples

Agrupación cruzada entre **varias dimensiones**.

```
datos %>%  
  group_by(area, sexo, etnia) %>%  
  summarise(media_ingreso = mean(ingreso), .groups = "drop")
```

area	sexo	etnia	media_ingreso
1	Hombre	0	3904.552
1	Hombre	1	2762.409
1	Hombre	99	1910.715
1	Mujer	0	3701.783
1	Mujer	1	2591.270
1	Mujer	99	2288.176
2	Hombre	0	2665.902
2	Hombre	1	2172.085
2	Mujer	0	2403.424
2	Mujer	1	1717.478

# Ordenar resultados

Ordena los resultados del resumen de forma descendente.

```
datos %>%  
  group_by(area, etnia) %>%  
  summarise(media_ingreso = mean(ingreso), .groups = "drop") %>%  
  arrange(desc(media_ingreso))
```

area	etnia	media_ingreso
1	0	3799.453
1	1	2670.701
2	0	2535.429
1	99	2018.561
2	1	1940.214

## Crear indicadores relativos

Conecta summarise() y mutate() para crear indicadores derivados.

```
datos %>%  
  group_by(area) %>%  
  summarise(  
    media_ingreso = mean(ingreso),  
    media_gasto = mean(gasto)  
  ) %>%  
  mutate(relacion = media_gasto / media_ingreso)
```

area	media_ingreso	media_gasto	relacion
1	3677.658	3546.095	0.9642263
2	2436.735	2414.815	0.9910043

## Cuantiles con across()

Calcula cuantiles o medianas de múltiples variables en un solo paso.

```
datos %>%  
  group_by(sexo) %>%  
  summarise(across(c(ingreso, gasto), ~ quantile(.x, 0.5)))
```

sexo	ingreso	gasto
Hombre	2266.667	2266.667
Mujer	2187.013	2187.013

# Conteo de observaciones

n() cuenta el número de registros por grupo.

```
datos %>%  
  group_by(pobreza) %>%  
  summarise(total_hogares = n())
```

pobreza	total_hogares
1	397
2	2597
3	16433

# Proporciones

Mide la distribución relativa de cada categoría.

```
datos %>%  
  group_by(sexo) %>%  
  summarise(prop = n() / nrow(datos))
```

sexo	prop
Hombre	0.4843774
Mujer	0.5156226

## Dispersión por grupo

Evalúa variación intra grupo.

```
datos %>%  
  group_by(pobreza) %>%  
  summarise(  
    var_ingreso = var(ingreso),  
    rango_ingreso = max(ingreso) - min(ingreso)  
  )
```

pobreza	var_ingreso	rango_ingreso
1	111500.1	1078.987
2	32298571.9	253749.913
3	33500457.9	253026.830

# Limpieza del agrupamiento

`ungroup()` elimina los grupos activos para posteriores análisis.

```
datos %>%  
  group_by(area) %>%  
  summarise(across(c(ingreso, gasto), mean)) %>%  
  ungroup()
```

area	ingreso	gasto
1	3677.658	3546.095
2	2436.735	2414.815

# Observación

Función	Propósito	Ejemplo
<code>group_by()</code>	Define agrupamientos	<code>group_by(area)</code>
<code>summarise()</code>	Resume valores	<code>summarise(mean(ingreso))</code>
<code>mutate()</code>	Crea nuevas variables	<code>mutate(tipo = ifelse(...))</code>
<code>across()</code>	Simplifica aplicación de funciones	<code>across(c(ingreso, gasto), mean)</code>