Ejercicios Prácticos: Tidyverse Avanzado

Pivots y Joins

FCE-UBA

Abril 2025

Ejercicio 1: Análisis de automóviles con pivot_longer

En este ejercicio trabajaremos con el conocido dataset `mtcars`, que contiene datos sobre diferentes modelos de automóviles.

```
# Cargar y examinar los datos
mtcars_tbl <- as_tibble(mtcars, rownames = "modelo")
glimpse(mtcars_tbl)
```

```
## Rows: 32
## Columns: 12
## $ modelo <chr> "Mazda RX4", "Mazda RX4 Wag", "Datsun 710", "Hornet 4 Drive", "~
## $ mpg    <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 19.2, 17.~
## $ cyl    <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, ~
## $ disp   <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.7, 140.8, ~
## $ hp     <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180, 180, 1~
## $ drat   <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3.92, 3.9~
## $ wt     <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.190, 3.150, ~
## $ qsec   <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.00, 22.90, ~
## $ vs     <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, ~
## $ am     <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, ~
## $ gear   <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3, ~
## $ carb   <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1, 2, 1, 1, ~
```

Tareas:

1. Transforma el dataset a formato "largo" dejando solo `modelo` como identificador y convirtiendo todas las demás columnas en pares variable-valor.
2. Filtra el dataset resultante para mostrar solo las variables relacionadas con el rendimiento: `mpg` (millas por galón), `hp` (caballos de fuerza) y `qsec` (tiempo en el cuarto de milla).
3. Crea una visualización que muestre estos tres indicadores de rendimiento para los 5 modelos más rápidos (menor `qsec`).

Ejercicio 2: Análisis de pingüinos con pivot_wider

Para este ejercicio utilizaremos el dataset `penguins` del paquete `palmerpenguins`, que contiene medidas de diferentes especies de pingüinos.

```
data(penguins)
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

Tareas:

1. Crea un resumen que calcule la media de la longitud del pico (`bill_length_mm`) para cada combinación de especie (`species`) e isla (`island`).
2. Transforma este resumen a formato "ancho" usando `pivot_wider()`, con las islas como columnas y las especies como filas.
3. Añade una columna que calcule la diferencia entre la longitud del pico más grande y más pequeña para cada especie (entre islas).
4. Interpreta brevemente los resultados: ¿hay diferencias significativas en la longitud del pico entre islas para la misma especie?

Ejercicio 3: Análisis internacional con pivot_longer y pivot_wider

Utilizaremos el dataset `gapminder` que contiene datos socioeconómicos de países a lo largo del tiempo.

```
data(gapminder)
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~
```

Tareas:

1. Filtra los datos para quedarte solo con el año más reciente (2007).
2. Crea un nuevo dataframe con países como filas y las variables `lifeExp` (esperanza de vida), `pop` (población) y `gdpPercap` (PIB per cápita) como columnas.
3. Usa `pivot_longer()` para transformar este dataset a formato largo.
4. Utiliza `pivot_wider()` para crear un dataset donde las filas sean continentes, las columnas sean las tres variables mencionadas, y los valores sean los promedios de cada variable por continente.
5. ¿Qué continente tiene la mayor esperanza de vida promedio? ¿Y el mayor PIB per cápita?

Ejercicio 4: Joins con datos de vuelos

Trabajaremos con los datasets del paquete `nycflights13`, que contiene información sobre vuelos desde aeropuertos de Nueva York en 2013.

```
# Examinar las tablas a utilizar
glimpse(flights)
```

```
## Rows: 336,776
## Columns: 19
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2~
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, ~
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, ~
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1~
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 849,~
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 851,~
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1~
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "~
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4~
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394~
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",~
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",~
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149, 1~
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 733, ~
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6~
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0~
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0~
```

```
glimpse(airlines)
```

```
## Rows: 16
## Columns: 2
## $ carrier <chr> "9E", "AA", "AS", "B6", "DL", "EV", "F9", "FL", "HA", "MQ", "O~
## $ name    <chr> "Endeavor Air Inc.", "American Airlines Inc.", "Alaska Airline~
```

```r
glimpse(airports)
```

```
## Rows: 1,458
## Columns: 8
## $ faa   <chr> "04G", "06A", "06C", "06N", "09J", "0A9", "0G6", "0G7", "0P2", "~
## $ name  <chr> "Lansdowne Airport", "Moton Field Municipal Airport", "Schaumbur~
## $ lat   <dbl> 41.13047, 32.46057, 41.98934, 41.43191, 31.07447, 36.37122, 41.4~
## $ lon   <dbl> -80.61958, -85.68003, -88.10124, -74.39156, -81.42778, -82.17342~
## $ alt   <dbl> 1044, 264, 801, 523, 11, 1593, 730, 492, 1000, 108, 409, 875, 10~
## $ tz    <dbl> -5, -6, -6, -5, -5, -5, -5, -5, -5, -8, -5, -6, -5, -5, -5, -5, ~
## $ dst   <chr> "A", "A", "A", "A", "A", "A", "A", "A", "U", "A", "A", "U", "A",~
## $ tzone <chr> "America/New_York", "America/Chicago", "America/Chicago", "Ameri~
```

Tareas:

1. Usa `inner_join()` para combinar la tabla `flights` con la tabla `airlines` y obtener el nombre completo de la aerolínea para cada vuelo.
2. Utiliza `left_join()` para añadir la información de los aeropuertos de origen.
3. Realiza un `anti_join()` para encontrar aerolíneas que aparecen en la tabla `airlines` pero que no operaron ningún vuelo en la base de datos.
4. Usa una combinación de joins para crear un resumen que muestre, para cada aerolínea, el número total de vuelos, el retraso promedio en la salida y el porcentaje de vuelos cancelados.

Ejercicio 5: Análisis económico con joins

Utilizaremos datos simulados inspirados en los conjuntos de datos de Wooldridge para análisis económico.

```r
# Crear dos datasets relacionados: trabajadores y empresas
set.seed(456)

# Dataset de trabajadores
trabajadores <- wage_data %>%
  mutate(
    id_trabajador = row_number(),
    id_empresa = sample(1:50, n(), replace = TRUE),
    sector = sample(c("Manufactura", "Servicios", "Tecnologia", "Salud", "Educacion"),
                    n(), replace = TRUE)
  ) %>%
  select(id_trabajador, id_empresa, wage, educ, exper, female, married, sector)

# Dataset de empresas
empresas <- tibble(
  id_empresa = 1:50,
  nombre_empresa = paste0("Empresa_", LETTERS[1:50]),
  tamano = sample(c("Pequena", "Mediana", "Grande"), 50, replace = TRUE),
  antiguedad = sample(1:50, 50, replace = TRUE),
  cotiza_bolsa = sample(c(TRUE, FALSE), 50, replace = TRUE, prob = c(0.3, 0.7))
)
```

```r
# Mostrar los primeros registros de ambas tablas
head(trabajadores)
```

id_trabajador	id_empresa	wage	educ	exper	female	married	sector
1	45	3.10	11	2	1	0	Tecnologia
2	37	3.24	12	22	1	1	Salud
3	35	3.00	11	2	0	0	Educacion
4	38	6.00	8	44	0	1	Salud
5	21	5.30	12	7	0	1	Manufactura
6	27	8.75	16	9	0	1	Servicios

```r
head(empresas)
```

id_empresa	nombre_empresa	tamano	antiguedad	cotiza_bolsa
1	Empresa_A	Pequena	25	FALSE
2	Empresa_B	Mediana	26	FALSE
3	Empresa_C	Pequena	12	FALSE
4	Empresa_D	Mediana	39	FALSE
5	Empresa_E	Grande	26	FALSE
6	Empresa_F	Mediana	40	TRUE

Tareas:

1. Utiliza `inner_join()` para combinar los datos de trabajadores y empresas.
2. Calcula el salario promedio por tamaño de empresa y por sector.
3. Con `anti_join()`, identifica si hay empresas que no tienen trabajadores en la muestra.
4. Usa `semi_join()` para filtrar solo los trabajadores que trabajan en empresas grandes.
5. Crea una visualización que muestre la relación entre educación (`educ`), experiencia (`exper`) y salario (`wage`) diferenciando por tamaño de empresa.

Ejercicio 6: Pivots avanzados con datos de gapminder

Continuaremos trabajando con el dataset `gapminder` pero ahora utilizando técnicas más avanzadas de pivot.

```r
# Volver a cargar los datos
data(gapminder)
```

Tareas:

1. Selecciona solo los años 1997 y 2007 del dataset.
2. Crea un resumen con el promedio de esperanza de vida y PIB per cápita por continente y año.
3. Utiliza `pivot_wider()` con múltiples columnas de valores para crear un dataset donde:
    - Las filas sean continentes

- Las columnas sean combinaciones de variable (lifeExp, gdpPercap) y año (1997, 2007)
- Crea nombres de columnas como "lifeExp_1997", "lifeExp_2007", etc.

4. Añade columnas que calculen la variación porcentual de cada indicador entre 1997 y 2007.
5. ¿Qué continente experimentó el mayor crecimiento en PIB per cápita? ¿Y en esperanza de vida?

Ejercicio 7: Análisis de ventas con pivots y joins

En este ejercicio trabajaremos con tablas relacionadas que contienen información sobre ventas, productos, clientes y tiendas.

```
# Examinar las tablas
head(productos)
```

id_producto	nombre	categoria	precio_unitario	stock
1	Producto_1	Hogar	43.73	134
2	Producto_2	Juguetes	852.31	176
3	Producto_3	Juguetes	240.78	164
4	Producto_4	Electronica	696.70	182
5	Producto_5	Hogar	476.65	151
6	Producto_6	Electronica	701.02	197

```
head(ventas)
```

id_venta	id_producto	id_cliente	id_tienda	fecha	cantidad
1	28	118	5	2023-04-06	8
2	34	98	8	2023-11-01	6
3	72	88	4	2023-05-07	9
4	9	125	3	2023-10-27	8
5	74	136	9	2023-07-17	8
6	33	193	2	2023-10-11	8

```
head(clientes)
```

id_cliente	nombre	ciudad	segmento
1	Cliente_1	Cordoba	Estandar
2	Cliente_2	Rosario	Estandar
3	Cliente_3	Rosario	Estandar
4	Cliente_4	Cordoba	Estandar
5	Cliente_5	Buenos Aires	Estandar
6	Cliente_6	La Plata	Premium

```
head(tiendas)
```

id_tienda	nombre	ubicacion	tamano
1	Tienda_A	Oeste	Mediana
2	Tienda_B	Oeste	Mediana
3	Tienda_C	Centro	Grande
4	Tienda_D	Centro	Grande
5	Tienda_E	Oeste	Grande
6	Tienda_F	Centro	Grande

Tareas:

1. Utiliza los joins adecuados para crear un dataset completo que combine información de ventas, productos, clientes y tiendas.
2. Calcula el monto total (cantidad × precio unitario) para cada venta y agrega esta información al dataset.
3. Crea un resumen que muestre las ventas totales por categoría de producto y por ciudad del cliente.
4. Transforma este resumen usando `pivot_wider()` para crear una matriz donde las filas sean las categorías de productos y las columnas sean las ciudades.
5. Identifica qué categoría de producto es la más vendida en cada ciudad.

Ejercicio 8: Joins múltiples y análisis de ventas

Continuamos con el análisis de las ventas, pero ahora nos enfocaremos en técnicas más avanzadas de joins.

```
# Crear datos adicionales: promociones
promociones <- tibble(
  id_promocion = 1:20,
  id_categoria = sample(c("Electronica", "Hogar", "Ropa", "Deportes", "Juguetes"), 20, replace = TRUE),
  fecha_inicio = sample(seq(as.Date("2023-01-01"), as.Date("2023-12-01"), by = "month"), 20, replace = T
  duracion_dias = sample(c(7, 14, 30), 20, replace = TRUE),
  descuento_pct = sample(c(10, 15, 20, 25, 30), 20, replace = TRUE)
)

# Añadir información de promoción a algunas ventas
ventas <- ventas %>%
  mutate(id_promocion = sample(c(NA, 1:20), n(), replace = TRUE, prob = c(0.7, rep(0.3/20, 20))))

head(promociones)
```

id_promocion	id_categoria	fecha_inicio	duracion_dias	descuento_pct
1	Ropa	2023-05-01	14	30
2	Electronica	2023-07-01	7	25
3	Electronica	2023-11-01	30	25
4	Deportes	2023-11-01	7	30
5	Hogar	2023-02-01	14	20
6	Hogar	2023-07-01	7	10

Tareas:

1. Realiza múltiples joins para combinar todas las tablas: ventas, productos, clientes, tiendas y promociones.
2. Calcula el precio final de cada venta considerando los descuentos de las promociones cuando corresponda.
3. Utiliza `anti_join()` para identificar ventas que no tienen promociones asociadas.
4. Crea un análisis que compare las ventas con promoción vs. sin promoción por categoría de producto.
5. Usa `pivot_wider()` para crear un resumen que muestre el impacto de las promociones en las ventas por categoría.

Ejercicio 9: Pivots complejos con datos anidados

En este ejercicio exploraremos técnicas avanzadas de pivots con estructuras de datos más complejas.

```r
# Crear un dataset con jerarquías y múltiples variables
evaluaciones <- tibble(
  curso = rep(c("Matematicas", "Fisica", "Economia", "Estadistica"), each = 40),
  estudiante = rep(paste0("Estudiante_", 1:10), times = 16),
  periodo = rep(rep(c("2023-1", "2023-2", "2024-1", "2024-2"), each = 10), times = 4),
  nota_teoria = runif(160, 4, 10) %>% round(1),
  nota_practica = runif(160, 4, 10) %>% round(1),
  asistencia_pct = runif(160, 60, 100) %>% round(0)
)

head(evaluaciones)
```

curso	estudiante	periodo	nota_teoria	nota_practica	asistencia_pct
Matematicas	Estudiante_1	2023-1	4.3	5.9	63
Matematicas	Estudiante_2	2023-1	4.3	10.0	95
Matematicas	Estudiante_3	2023-1	7.0	10.0	75
Matematicas	Estudiante_4	2023-1	9.1	9.2	75
Matematicas	Estudiante_5	2023-1	7.6	7.5	68
Matematicas	Estudiante_6	2023-1	9.2	6.0	72

Tareas:

1. Crea una nota final calculada como 60% de la nota de teoría y 40% de la nota de práctica.
2. Transforma el dataset para tener una columna de "tipo_evaluacion" y otra de "valor", donde tipo_evaluacion puede ser "teoria", "practica" o "asistencia".
3. Utiliza `pivot_wider()` para crear un dataset donde las filas sean combinaciones de curso y estudiante, y las columnas sean las evaluaciones en los diferentes períodos.
4. Crea un resumen de la evolución de las notas promedio por curso a lo largo de los períodos.
5. Transforma este resumen utilizando técnicas avanzadas de pivots para visualizar la tendencia temporal de manera efectiva.

Anexo: creación de datos

Datos ejercicio 5

```r
# Paquetes para datos de economia
# Si no los tienes instalados, puedes usar:
# install.packages(c("wooldridge", "AER"))

if(!require(wooldridge)) { # Si no lo tienen instalado se simula
  # Crear un dataset similar al de Wooldridge wage1
  set.seed(123)
  n <- 500
  wage_data <- tibble(
    wage = rnorm(n, mean = 950, sd = 400),
    educ = sample(8:18, n, replace = TRUE),
    exper = sample(1:50, n, replace = TRUE),
    female = sample(0:1, n, replace = TRUE, prob = c(0.6, 0.4)),
    married = sample(0:1, n, replace = TRUE)
  )
} else {
  library(wooldridge)
  data("wage1")
  wage_data <- as_tibble(wage1)
}
```

Datos ejercicio 7

```r
# Datos de ventas
set.seed(321)
productos <- tibble(
  id_producto = 1:100,
  nombre = paste0("Producto_", 1:100),
  categoria = sample(c("Electronica", "Hogar", "Ropa", "Deportes", "Juguetes"), 100, replace = TRUE),
  precio_unitario = runif(100, 10, 1000) %>% round(2),
  stock = sample(0:200, 100, replace = TRUE)
)

ventas <- tibble(
  id_venta = 1:1000,
  id_producto = sample(1:100, 1000, replace = TRUE),
  id_cliente = sample(1:200, 1000, replace = TRUE),
  id_tienda = sample(1:10, 1000, replace = TRUE),
  fecha = sample(seq(as.Date("2023-01-01"), as.Date("2023-12-31"), by = "day"), 1000, replace = TRUE),
  cantidad = sample(1:10, 1000, replace = TRUE)
)

clientes <- tibble(
  id_cliente = 1:200,
  nombre = paste0("Cliente_", 1:200),
  ciudad = sample(c("Buenos Aires", "Cordoba", "Rosario", "Mendoza", "La Plata"), 200, replace = TRUE),
```

```r
  segmento = sample(c("Estandar", "Premium", "VIP"), 200, replace = TRUE, prob = c(0.7, 0.2, 0.1))
)

tiendas <- tibble(
  id_tienda = 1:10,
  nombre = paste0("Tienda_", LETTERS[1:10]),
  ubicacion = sample(c("Centro", "Norte", "Sur", "Este", "Oeste"), 10, replace = TRUE),
  tamano = sample(c("Pequena", "Mediana", "Grande"), 10, replace = TRUE)
)
```