Tema IV: Manipulación de datos.

Maicel Monzón

Objetivos temáticos:

- Aprender a transformar y resumir datos.
- Transformación de datos

Contenido clave:

- Filtrar, seleccionar y ordenar datos con dplyr.
- Crear nuevas variables con mutate().
- Resúmenes estadísticos básicos con summarise().

Funciones clave de dplyr

- Filtrar o elegir las observaciones por sus valores (filter() — del inglés filtrar).
- Reordenar las filas (arrange() — del inglés organizar).
- Seleccionar las variables por sus nombres (select() — del inglés seleccionar).
- Crear nuevas variables con transformaciones de variables existentes (mutate() — del inglés mutar o transformar).
- Contraer muchos valores en un solo resumen (summarise() — del inglés resumir).

Funciones clave de dplyr

- Todas estas funciones se pueden usar junto con group_by() (del inglés agrupar por),

cambia el alcance de cada función para que actúe ya no sobre todo el conjunto de datos sino de grupo en grupo.

Todos los verbos funcionan de manera similar:

1. El primer argumento es un data frame.
2. Los argumentos posteriores describen qué hacer con el data frame usando los nombres de las variables (sin comillas).
3. El resultado es un nuevo data frame.

Filtrar filas con filter()

```
vuelos %>%
filter( mes == 1, dia == 1)
```

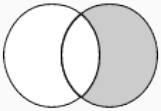
```
# A tibble: 842 x 19
    anio   mes   dia horario_salida salida_programada atraso_salida
   <int> <int> <int>          <int>              <int>         <dbl>
 1  2013     1     1            517                515             2
 2  2013     1     1            533                529             4
 3  2013     1     1            542                540             2
 4  2013     1     1            544                545            -1
 5  2013     1     1            554                600            -6
 6  2013     1     1            554                558            -4
 7  2013     1     1            555                600            -5
 8  2013     1     1            557                600            -3
 9  2013     1     1            557                600            -3
```

Filtrar filas con filter()

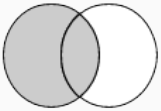
```
vuelos %>%
  filter( mes == 11 | mes == 12)
```

```
# A tibble: 55,403 x 19
    anio   mes   dia horario_salida salida_programada atraso_salida
   <int> <int> <int>          <int>             <int>         <dbl>
 1  2013    11     1              5              2359             6
 2  2013    11     1             35              2250           105
 3  2013    11     1            455               500            -5
 4  2013    11     1            539               545            -6
 5  2013    11     1            542               545            -3
 6  2013    11     1            549               600           -11
 7  2013    11     1            550               600           -10
 8  2013    11     1            554               600            -6
 9  2013    11     1            554               600            -6
```

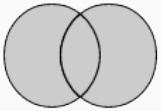
operadores lógicos



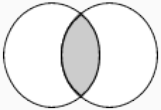
`y & !x`



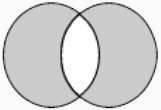
`x`



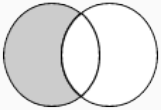
`x | y`



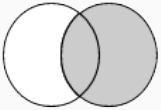
`x & y`



`xor(x, y)`



`x & !y`



`y`

Reordenar las filas con arrange()

```
vuelos %>%
  arrange( anio, mes, dia)
```

```
# A tibble: 336,776 x 19
    anio   mes   dia horario_salida salida_programada atraso_salida
   <int> <int> <int>          <int>             <int>         <dbl>
 1  2013     1     1            517               515             2
 2  2013     1     1            533               529             4
 3  2013     1     1            542               540             2
 4  2013     1     1            544               545            -1
 5  2013     1     1            554               600            -6
 6  2013     1     1            554               558            -4
 7  2013     1     1            555               600            -5
 8  2013     1     1            557               600            -3
 9  2013     1     1            557               600            -3
```

Seleccionar columnas con select()

Seleccionar columnas por nombre

```
vuelos %>%
  select( anio, mes, dia)
```

```
# A tibble: 336,776 x 3
    anio   mes   dia
   <int> <int> <int>
 1  2013     1     1
 2  2013     1     1
 3  2013     1     1
 4  2013     1     1
 5  2013     1     1
 6  2013     1     1
 7  2013     1     1
```

Seleccionar columnas con select()

Seleccionar todas las columnas entre anio y dia (incluyente)

```
vuelos %>%
  select(anio:dia)
```

```
# A tibble: 336,776 x 3
    anio   mes   dia
   <int> <int> <int>
 1  2013     1     1
 2  2013     1     1
 3  2013     1     1
 4  2013     1     1
 5  2013     1     1
 6  2013     1     1
 7  2013     1     1
```

Seleccionar columnas con select()

Seleccionar todas las columnas excepto aquellas entre anio en dia (incluyente)

```
vuelos %>%
  select( -(anio:dia))
```

```
# A tibble: 336,776 x 16
   horario_salida salida_programada atraso_salida horario_llegada
            <int>             <int>         <dbl>           <int>
 1            517               515             2             830
 2            533               529             4             850
 3            542               540             2             923
 4            544               545            -1            1004
 5            554               600            -6             812
 6            554               558            -4             740
 7            555               600            -5             913
```

Añadir nuevas variables con mutate()

```
vuelos_sml <- select(vuelos,
  anio:dia,
  starts_with("atraso"),
  distancia,
  tiempo_vuelo
)

mutate(vuelos_sml,
  ganancia = atraso_salida - atraso_llegada,
  velocidad = distancia / tiempo_vuelo * 60
)
```

```
# A tibble: 336,776 x 9
    anio   mes   dia atraso_salida atraso_llegada distancia tiempo_vuelo
   <int> <int> <int>         <dbl>          <dbl>     <dbl>        <dbl>
```

Resúmenes agrupados con summarise()

Se encarga de colapsar un data frame en una sola fila:

```
vuelos %>%
  summarise( atraso = mean(atraso_salida, na.rm = TRUE))
```

```
# A tibble: 1 x 1
  atraso
   <dbl>
1   12.6
```

Resúmenes agrupados con summarise()

```
vuelos %>%
  group_by(anio, mes, dia) %>%
  summarise(atraso = mean(atraso_salida, na.rm = TRUE))
```

```
# A tibble: 365 x 4
# Groups:   anio, mes [12]
    anio   mes   dia atraso
   <int> <int> <int>  <dbl>
 1  2013     1     1  11.5
 2  2013     1     2  13.9
 3  2013     1     3  11.0
 4  2013     1     4   8.95
 5  2013     1     5   5.73
 6  2013     1     6   7.15
 7  2013     1     7   5.42
```