

Actividad 3 Solución

Ejercicio 1

Cargar la librería `tidyverse` y con los datos `mpg` usar las funciones que vimos de `dplyr` para transformar los datos de la siguiente forma.

1. Seleccionar el conjunto de autos del 2008 que tienen `hwy` mayor a 31. ¿Cuántos autos son?

```
library(tidyverse)
mpg |>
  filter(year == 2008, hwy > 31) |>
  print(n = 5)
```

```
# A tibble: 6 x 11
  manufacturer model  displ  year   cyl trans      drv   cty   hwy fl  class
  <chr>         <chr>  <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
1 honda      civic    1.8   2008     4 manual(m~ f     26    34 r   subc~
2 honda      civic    1.8   2008     4 auto(15) f     25    36 r   subc~
3 honda      civic    1.8   2008     4 auto(15) f     24    36 c   subc~
4 nissan     altima    2.5   2008     4 manual(m~ f     23    32 r   mids~
5 toyota     corolla    1.8   2008     4 manual(m~ f     28    37 r   comp~
# i 1 more row
```

En este caso son solamente 6 autos y es sencillo contar pero podemos usar `summarise` o `count` para contarlos.

```
mpg |>
  filter(year == 2008, hwy > 31) |>
  summarise(n = n())
```

```
# A tibble: 1 x 1
      n
  <int>
1     6
```

2. Seleccionar el conjunto de autos que tienen `cty` igual a 9 ó `hwy` es mayor a 90. ¿Cuántos autos son? En este caso estoy pidiendo una condición que implica unión (no intersección como el anterior), los autos que cumplan *al menos* una de las condiciones deben ser incluidos. Para esto uso `|` para filtrar, noten que no hay autos con `hwy > 90` en la tabla.

```
mpg |> filter(cty == 9 | hwy > 90) |>
  select(cty, hwy)
```

```
# A tibble: 5 x 2
   cty  hwy
  <int> <int>
1     9   12
2     9   12
3     9   12
4     9   12
5     9   12
```

```
mpg |> filter(cty == 9 | hwy > 90) |>
  summarise(n = n())
```

```
# A tibble: 1 x 1
      n
  <int>
1     5
```

3. Selecciona la variable `displ` y `hwy`

Aquí usamos el verbo `select` para elegir las variables que se piden.

```
mpg |>
  select(displ, hwy) |>
  print(n = 5)
```

```
# A tibble: 234 x 2
  displ  hwy
  <dbl> <int>
```

```

1  1.8    29
2  1.8    29
3  2      31
4  2      30
5  2.8    26
# i 229 more rows

```

4. Seleccioná desde la variable `trans` hasta la variable `hwy`

Similar al anterior, si uso `select(trans:hwy)` obtengo lo que se pide

```

mpg|>
  select(trans:hwy) |>
  print(n = 5)

```

```

# A tibble: 234 x 4
  trans      drv      cty      hwy
  <chr>    <chr> <int> <int>
1 auto(l5)  f        18     29
2 manual(m5) f        21     29
3 manual(m6) f        20     31
4 auto(av)  f        21     30
5 auto(l5)  f        16     26
# i 229 more rows

```

5. Produce un dataframe: marca, modelo, año, cantidad de cilindros y rendimiento en ciudad. Únicamente para los autos, toyota, camry.

```

mpg |>
filter(manufacturer == 'toyota' & model == 'camry') |>
select(manufacturer, model, year, cyl, cty) |>
  dim()

```

6. Calcula el rendimiento promedio en ciudad para cada marca (`manufacturer`). ¿Cuál es el mejor rendimiento en ciudad? y el peor ?

```

mpg |>
  group_by(manufacturer) |>
  summarise(rend.ciudad = mean(cty) ) |>
  filter(rend.ciudad == min(rend.ciudad) | rend.ciudad == max(rend.ciudad))

```

```
# A tibble: 2 x 2
  manufacturer rend.ciudad
  <chr>         <dbl>
1 honda         24.4
2 lincoln       11.3
```

```
# ó también
# mpg |>
#   group_by(manufacturer) |>
#   summarise(rend.ciudad = mean(cty)) |>
#   summarise(minimo = min(rend.ciudad), maximo = max(rend.ciudad))
```

7. Para cada marca: calcula el rendimiento promedio en ciudad, el error estandar de la media, y el rango

```
mpg |>
  group_by(manufacturer) |>
  summarise(rend.mn = mean(cty),
            rend.se = sd(cty)/sqrt(n()),
            rend.rg = max(cty) - min(cty))
```

```
# A tibble: 15 x 4
  manufacturer rend.mn rend.se rend.rg
  <chr>         <dbl>  <dbl>  <int>
1 audi         17.6    0.465     6
2 chevrolet     15    0.671    11
3 dodge        13.1    0.409     9
4 ford         14     0.383     7
5 honda        24.4    0.648     7
6 hyundai      18.6    0.401     5
7 jeep         13.5    0.886     8
8 land rover   11.5    0.289     1
9 lincoln      11.3    0.333     1
10 mercury     13.2    0.25      1
11 nissan       18.1    0.950    11
12 pontiac     17     0.447     2
13 subaru      19.3    0.244     3
14 toyota      18.5    0.694    17
15 volkswagen  20.9    0.877    19
```

8. Mejora en el rendimiento: calcular el rendimiento promedio para cada marca, distinguiendo antes y después de 2004

```
# Forma 1
mpg |>
  mutate(
    rend.antes = ifelse(year < 2005, cty, NA),
    rend.despues = ifelse(year > 2004, cty, NA)
  ) |>
  group_by(manufacturer) |>
  summarise(
    rend.mn.antes = mean(rend.antes, na.rm = TRUE),
    rend.mn.despues = mean(rend.despues, na.rm = TRUE)
  )
```

```
# Forma 2
mpg |>
  group_by(manufacturer) |>
  summarise(
    rend.mn.antes = mean(cty[year < 2005]),
    rend.mn.despues = mean(cty[year > 2004])
  )
```

```
# A tibble: 15 x 3
  manufacturer rend.mn.antes rend.mn.despues
  <chr>          <dbl>          <dbl>
1 audi           17.1           18.1
2 chevrolet      15.1           14.9
3 dodge          13.4           13.0
4 ford           13.9           14.1
5 honda          24.8           24
6 hyundai        18.3           18.9
7 jeep           14.5           13.2
8 land rover     11            12
9 lincoln        11            12
10 mercury       13.5           13
11 nissan         17.7           18.4
12 pontiac        17            17
13 subaru         19            19.5
14 toyota         18.2           19.1
15 volkswagen     21.2           20.5
```

Camino más corto

```
mpg %>%
  group_by(manufacturer, antes_2004 = year < 2005) %>%
  summarise(rend.mn = mean(cty)) %>%
  pivot_wider(data = .,
              names_from = antes_2004,
              values_from = rend.mn,
              names_glue = "antes_2004_{ifelse(antes_2004, 'si', 'no')}")
```

```
# A tibble: 15 x 3
# Groups:   manufacturer [15]
  manufacturer antes_2004_no antes_2004_si
  <chr>           <dbl>           <dbl>
1 audi            18.1             17.1
2 chevrolet       14.9             15.1
3 dodge           13.0             13.4
4 ford            14.1             13.9
5 honda           24              24.8
6 hyundai         18.9             18.3
7 jeep            13.2             14.5
8 land rover      12              11
9 lincoln         12              11
10 mercury        13              13.5
11 nissan          18.4             17.7
12 pontiac         17              17
13 subaru          19.5             19
14 toyota          19.1             18.2
15 volkswagen      20.5             21.2
```

9. Calcular el rendimiento promedio en carretera (hwy), para 3 marcas seleccionadas aleatoriamente y ordena el resultado de menor a mayor

```
set.seed(123)
mpg |>
  filter(manufacturer %in% sample(unique(manufacturer), 3)) |>
  group_by(manufacturer) |>
  summarise(rend.mn = mean(cty))
```

```
# A tibble: 3 x 2
  manufacturer rend.mn
  <chr>         <dbl>
1 dodge        13.1
```

```
2 toyota      18.5
3 volkswagen  20.9
```

10. Crear una nueva variable que transforme hwy (millas por galón) en litros/100 km

Primero tenés que saber que la conversión de galones a litros es 1 gallón = 3.78541 litros, y para convertir de millas a kilómetros es 1 milla = 1.60934 km. Ahora, podés crear la variable con `mutate()`

```
mpg_inc <- mpg |> mutate(litkm = (1/hwy)*(3.78541 /1.60934)*100 )
mpg_inc
```

```
# A tibble: 234 x 12
  manufacturer model      displ  year   cyl trans drv      cty   hwy fl      class
  <chr>          <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
1 audi          a4         1.8  1999     4 auto~ f      18    29 p    comp~
2 audi          a4         1.8  1999     4 manu~ f      21    29 p    comp~
3 audi          a4         2    2008     4 manu~ f      20    31 p    comp~
4 audi          a4         2    2008     4 auto~ f      21    30 p    comp~
5 audi          a4         2.8  1999     6 auto~ f      16    26 p    comp~
6 audi          a4         2.8  1999     6 manu~ f      18    26 p    comp~
7 audi          a4         3.1  2008     6 auto~ f      18    27 p    comp~
8 audi          a4 quattro  1.8  1999     4 manu~ 4      18    26 p    comp~
9 audi          a4 quattro  1.8  1999     4 auto~ 4      16    25 p    comp~
10 audi         a4 quattro  2    2008     4 manu~ 4      20    28 p    comp~
# i 224 more rows
# i 1 more variable: litkm <dbl>
```