

Actividad Individual 8

Ángela Vieyto 5.487.839-8

Entrega 17/5/2021

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.

```
library(tidyverse)
```

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

```
filter(mpg, year == 2008 & hwy > 31)
```

```
## # 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(~ f    26    34 r    subcom~
## 2 honda        civic    1.8  2008    4 auto(15) f    25    36 r    subcom~
## 3 honda        civic    1.8  2008    4 auto(15) f    24    36 c    subcom~
## 4 nissan        altima   2.5  2008    4 manual(~ f    23    32 r    midsize
## 5 toyota        corol~   1.8  2008    4 manual(~ f    28    37 r    compact
## 6 toyota        corol~   1.8  2008    4 auto(14) f    26    35 r    compact
```

Hay 6 autos del 2008 con un rendimiento en carretera mayor a 31 millas por galón de combustible.

2. Seleccionar el conjunto de autos que tienen `cty` igual a 9 o `hwy` es mayor a 90. ¿Cuántos autos son?

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

```
## # A tibble: 5 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 dodge        dakota pi~  4.7  2008    8 auto(~ 4    9    12 e    pick~
## 2 dodge        durango 4~  4.7  2008    8 auto(~ 4    9    12 e    suv
## 3 dodge        ram 1500 ~  4.7  2008    8 auto(~ 4    9    12 e    pick~
## 4 dodge        ram 1500 ~  4.7  2008    8 manua~ 4    9    12 e    pick~
## 5 jeep         grand che~  4.7  2008    8 auto(~ 4    9    12 e    suv
```

Hay 5 autos con un rendimiento en ciudad de 9 millas por galón de combustible, pero no hay ningún auto con un rendimiento en carretera mayor a 90 millas por galón de combustible.

3. Seleccionar las variables `displ` y `hwy`.

```
select(mpg, c(displ, hwy))
```

```
## # 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
## 6 2.8 26
## 7 3.1 27
## 8 1.8 26
## 9 1.8 25
## 10 2 28
## # ... with 224 more rows
```

Comentario: No es necesario concatenar

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

```
select(mpg, trans:hwy)
```

```
## # 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
## 6 manual(m5) f      18    26
## 7 auto(av)   f      18    27
## 8 manual(m5) 4      18    26
## 9 auto(l5)   4      16    25
## 10 manual(m6) 4      20    28
## # ... with 224 more rows
```

5. Producir un `data.frame`: marca, modelo, año, cantidad de cilindros y rendimiento en ciudad. Únicamente para los autos, toyota, camry.

Opción 1

```
filter(select(mpg, c(manufacturer, model, year, cyl, cty)),
        manufacturer == "toyota" & model == "camry")
```

```
## # A tibble: 7 x 5
##   manufacturer model  year   cyl   cty
##   <chr>        <chr> <int> <int> <int>
## 1 toyota      camry  1999     4    21
## 2 toyota      camry  1999     4    21
## 3 toyota      camry  2008     4    21
## 4 toyota      camry  2008     4    21
## 5 toyota      camry  1999     6    18
## 6 toyota      camry  1999     6    18
## 7 toyota      camry  2008     6    19
```

Opción 2

```
mpg %>%
  filter(manufacturer == "toyota" & model == "camry") %>%
  select(c(manufacturer, model, year, cyl, cty))
```

```
## # A tibble: 7 x 5
##   manufacturer model  year   cyl   cty
```

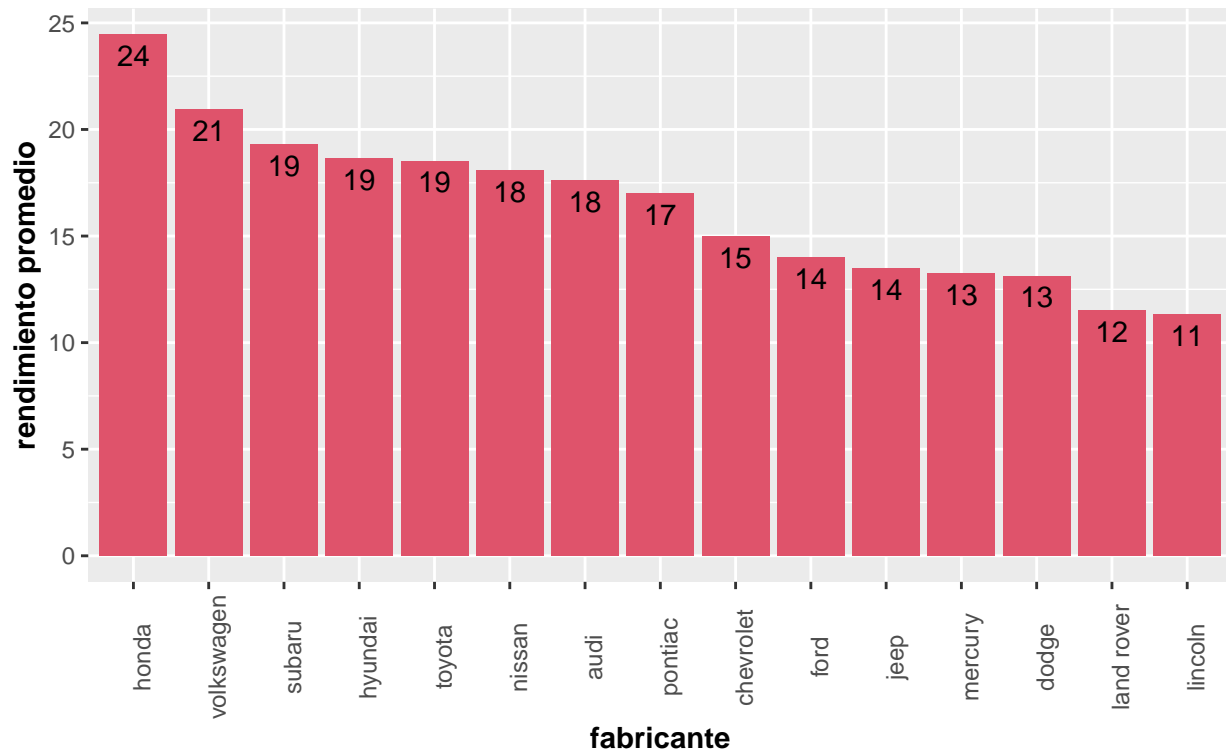
```
##   <chr>      <chr> <int> <int> <int>
## 1 toyota    camry  1999    4    21
## 2 toyota    camry  1999    4    21
## 3 toyota    camry  2008    4    21
## 4 toyota    camry  2008    4    21
## 5 toyota    camry  1999    6    18
## 6 toyota    camry  1999    6    18
## 7 toyota    camry  2008    6    19
```

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

```
mpg %>%
  group_by(manufacturer) %>%
  summarise(mean_cty = mean(cty)) %>%
  ggplot(aes(x = reorder(manufacturer, -mean_cty), y = mean_cty)) +
  geom_col(fill = "#e41a1c") +
  labs(x = "fabricante",
       y = "rendimiento promedio",
       title = "Rendimiento promedio en ciudad (en millas por galón)",
       subtitle = "1999 - 2008") +
  geom_text(aes(label = round(mean_cty, 0)), vjust = 1.5) +
  theme(plot.title = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        axis.text.x = element_text(angle = 90))
```

Rendimiento promedio en ciudad (en millas por galón)

1999 – 2008



Mientras que Honda presenta el mejor rendimiento promedio en ciudad (ascendiendo a 24 millas por galón),

Lincoln presenta el peor rendimiento (ascendiendo a 11 millas por galón).

Comentario: Excelente!!

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

Opción 1 - dos líneas por fabricante

```
mpg %>%
  group_by(manufacturer) %>%
    summarise(mean_cty = mean(cty),
              sd_cty = sd(cty),
              range_cty = range(cty))

## # A tibble: 30 x 4
## # Groups:   manufacturer [15]
##   manufacturer mean_cty sd_cty range_cty
##   <chr>          <dbl> <dbl>    <int>
## 1 audi          17.6   1.97     15
## 2 audi          17.6   1.97     21
## 3 chevrolet     15     2.92     11
## 4 chevrolet     15     2.92     22
## 5 dodge         13.1   2.49      9
## 6 dodge         13.1   2.49     18
## 7 ford          14     1.91     11
## 8 ford          14     1.91     18
## 9 honda         24.4   1.94     21
## 10 honda        24.4   1.94     28
## # ... with 20 more rows
```

Opción 2 - una línea por fabricante

```
mpg %>%
  group_by(manufacturer) %>%
    summarise(mean_cty = mean(cty),
              sd_cty = sd(cty),
              min_cty = min(cty),
              max_cty = max(cty))

## # A tibble: 15 x 5
##   manufacturer mean_cty sd_cty min_cty max_cty
##   * <chr>          <dbl> <dbl>    <int>    <int>
## 1 audi          17.6   1.97     15      21
## 2 chevrolet     15     2.92     11      22
## 3 dodge         13.1   2.49      9      18
## 4 ford          14     1.91     11      18
## 5 honda         24.4   1.94     21      28
## 6 hyundai       18.6   1.50     16      21
## 7 jeep          13.5   2.51      9      17
## 8 land rover    11.5   0.577    11      12
## 9 lincoln       11.3   0.577    11      12
## 10 mercury      13.2   0.5      13      14
## 11 nissan        18.1   3.43     12      23
## 12 pontiac       17     1       16      18
## 13 subaru       19.3   0.914    18      21
## 14 toyota       18.5   4.05     11      28
## 15 volkswagen   20.9   4.56     16      35
```

Comentario: Falta calcular el rango max - min. En el primer casos se duplican las filas lo cual es un error. Y

es el error estándar de la media no el desvío estándar

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

```
# Opción 1 - dos líneas por fabricante, una por cada año (1999 y 2008)
```

```
mpg %>%  
  group_by(manufacturer, year) %>%  
  summarise(mean_cty = mean(cty))
```

```
## # A tibble: 30 x 3  
## # Groups:   manufacturer [15]  
##   manufacturer year mean_cty  
##   <chr>         <int>   <dbl>  
## 1 audi          1999    17.1  
## 2 audi          2008    18.1  
## 3 chevrolet     1999    15.1  
## 4 chevrolet     2008    14.9  
## 5 dodge         1999    13.4  
## 6 dodge         2008    13.0  
## 7 ford          1999    13.9  
## 8 ford          2008    14.1  
## 9 honda         1999    24.8  
## 10 honda        2008     24  
## # ... with 20 more rows
```

```
# Opción 2 - dos líneas por fabricante, según el año sea mayor o menor a 2004
```

```
mpg %>%  
  group_by(manufacturer, year < 2004) %>%  
  summarise(mean_cty = mean(cty))
```

```
## # A tibble: 30 x 3  
## # Groups:   manufacturer [15]  
##   manufacturer 'year < 2004' mean_cty  
##   <chr>         <lgl>         <dbl>  
## 1 audi          FALSE          18.1  
## 2 audi          TRUE           17.1  
## 3 chevrolet     FALSE          14.9  
## 4 chevrolet     TRUE           15.1  
## 5 dodge         FALSE          13.0  
## 6 dodge         TRUE           13.4  
## 7 ford          FALSE          14.1  
## 8 ford          TRUE           13.9  
## 9 honda         FALSE           24  
## 10 honda        TRUE           24.8  
## # ... with 20 more rows
```

```
# Opción 3 - una línea por fabricante
```

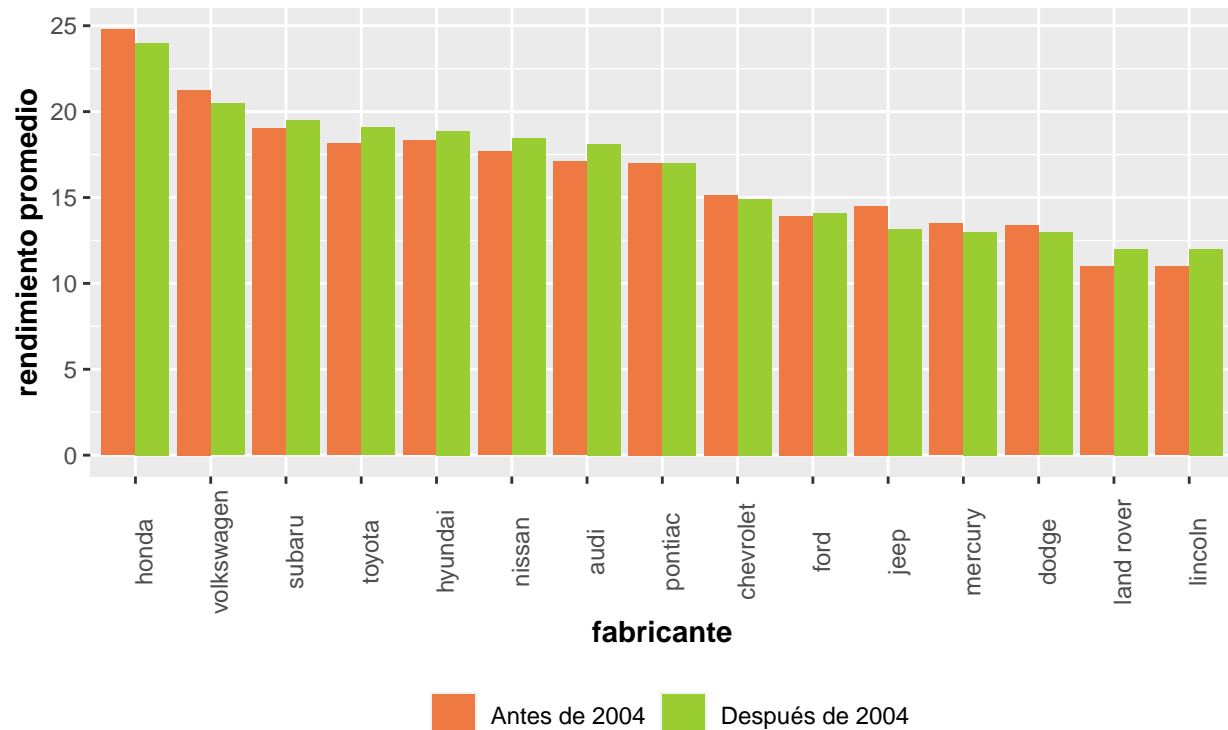
```
mpg %>%  
  mutate(cty,  
    cty_before2004 = ifelse(year < 2004, cty, NA),  
    cty_after2004 = ifelse(year >= 2004, cty, NA)) %>%  
  group_by(manufacturer) %>%  
  summarise(mean_cty = mean(cty),  
    mean_cty_before2004 = mean(cty_before2004, na.rm = TRUE),  
    mean_cty_after2004 = mean(cty_after2004, na.rm = TRUE))
```

```
## # A tibble: 15 x 4
##   manufacturer mean_cty mean_cty_before2004 mean_cty_after2004
## * <chr>          <dbl>          <dbl>          <dbl>
## 1 audi            17.6            17.1            18.1
## 2 chevrolet       15             15.1            14.9
## 3 dodge           13.1            13.4            13.0
## 4 ford            14             13.9            14.1
## 5 honda           24.4            24.8            24
## 6 hyundai         18.6            18.3            18.9
## 7 jeep            13.5            14.5            13.2
## 8 land rover      11.5            11             12
## 9 lincoln         11.3            11             12
## 10 mercury        13.2            13.5            13
## 11 nissan           18.1            17.7            18.4
## 12 pontiac        17             17             17
## 13 subaru         19.3            19             19.5
## 14 toyota         18.5            18.2            19.1
## 15 volkswagen     20.9            21.2            20.5
```

```
mpg %>%
  group_by(manufacturer, year) %>%
  summarise(mean_cty = mean(cty)) %>%
  ggplot() +
  geom_col(aes(x = reorder(manufacturer, -mean_cty), y = mean_cty, fill = year > 2004),
    position = "dodge") +
  labs(x = "fabricante",
    y = "rendimiento promedio",
    fill = NULL,
    title = "Rendimiento promedio en ciudad (en millas por galón)",
    subtitle = "1999 - 2008") +
  scale_fill_manual(labels = c("Antes de 2004", "Después de 2004"),
    values = c("FALSE" = "sienna2", "TRUE" = "olivedrab3")) +
  theme (plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.text.x = element_text(angle = 90),
    legend.position = "bottom")
```

Rendimiento promedio en ciudad (en millas por galón)

1999 – 2008



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

```
mpg %>%
  filter(manufacturer == sample(manufacturer, 3)) %>%
  group_by(manufacturer) %>%
  summarise(mean_hwy = mean(hwy)) %>%
  arrange(mean_hwy)
```

```
## # A tibble: 3 x 2
##   manufacturer mean_hwy
##   <chr>          <dbl>
## 1 chevrolet      20
## 2 hyundai       27.4
## 3 honda         31.7
```

10. Crear una nueva variable que transforme hwy (millas por galón) en litros/100 km. Primero tenés que saber la conversión de galones a litros y de millas a kilómetros.

1 gallón = 3.78541 litros

1 milla = 1.60934 km

```
mpg %>%
  mutate(hwy, l100km = (3.78541/1.60934)/hwy)
```

```
## # 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 q~    1.8 1999      4 manu~ 4      18 26 p      comp~
## 9 audi      a4 q~    1.8 1999      4 auto~ 4      16 25 p      comp~
## 10 audi     a4 q~    2    2008      4 manu~ 4      20 28 p      comp~
## # ... with 224 more rows, and 1 more variable: l100km <dbl>
```

Comentario: Excelente trabajo nuevamente!!