

Actividad Individual 8

Natalia da Silva

12 de Abril

Esta actividad tiene que quedar disponible en su repositorio de GitHub con el resto de las actividades y tareas del curso. Asegurate que tanto lucía como yo seamos colaboradoras de tu proyecto Tareas_STAT_NT creado hace dos semanas. Recordar seleccionar en en opciones de proyecto, codificación de código UTF-8. Si trabajás en una misma computadora en el Lab todas tus actividades deben estar en un proyecto de RStudio en el disco T. Recordar que para que tengas la última version de tu repositorio debes hacer pull a tu repositorio para no generar inconsistencias y antes de terminar subir tus cambios con `commit` y `push`.

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.

```
ggplot2::mpg %>%  
  filter(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 manu~ f     26    34 r   subc~  
2 honda         civic    1.8  2008     4 auto~ f     25    36 r   subc~  
3 honda         civic    1.8  2008     4 auto~ f     24    36 c   subc~  
4 nissan        altima   2.5  2008     4 manu~ f     23    32 r   mids~  
5 toyota        corol~   1.8  2008     4 manu~ f     28    37 r   comp~  
6 toyota        corol~   1.8  2008     4 auto~ f     26    35 r   comp~
```

¿Cuántos autos son?

```
ggplot2::mpg %>%  
  filter(year == 2008, hwy > 31) %$%  
  cat("Son", dim(.)[1], "autos")
```

Son 6 autos

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

```
mpg %>%  
  filter(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	dakot~	4.7	2008	8	auto~	4	9	12	e	pick~
2	dodge	duran~	4.7	2008	8	auto~	4	9	12	e	suv
3	dodge	ram 1~	4.7	2008	8	auto~	4	9	12	e	pick~
4	dodge	ram 1~	4.7	2008	8	manu~	4	9	12	e	pick~
5	jeep	grand~	4.7	2008	8	auto~	4	9	12	e	suv

```
mpg %>%
  filter(cty == 9 | hwy > 90) %$%
  cat("Son", dim(.)[1], "autos")
```

Son 5 autos

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

```
select(mpg, 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
```

4. Seleccioná 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. 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)
```

```
# 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. 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(prom_ciudad = mean(cty, na.rm = TRUE)) %>%
  arrange(prom_ciudad)
```

```
# A tibble: 15 x 2
  manufacturer prom_ciudad
    <chr>         <dbl>
1 lincoln          11.3
2 land rover       11.5
3 dodge            13.1
4 mercury          13.2
5 jeep             13.5
6 ford             14
7 chevrolet        15
8 pontiac           17
9 audi             17.6
10 nissan           18.1
11 toyota           18.5
12 hyundai          18.6
13 subaru           19.3
14 volkswagen       20.9
15 honda            24.4
```

```
mpg %>%
  group_by(manufacturer) %>%
  summarise(prom_ciudad = mean(cty, na.rm = TRUE)) %>%
  arrange(prom_ciudad) %>%
  .[1,1] %$%
  cat("El peor es", as.character(.))
```

El peor es lincoln

```
mpg %>%
  group_by(manufacturer) %>%
  summarise(prom_ciudad = mean(cty, na.rm = TRUE)) %>%
  arrange(desc(prom_ciudad)) %>%
  .[1,1] %$%
  cat("El mejor es", as.character(.))
```

El mejor es honda

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

```
mpg %>%
  group_by(manufacturer) %>%
  summarise(prom_ciudad = mean(cty, na.rm = TRUE),
            st_err_med = var(cty, na.rm = TRUE) / n(),
            rango = max(cty) - min(cty))
```

```
# A tibble: 15 x 4
  manufacturer prom_ciudad st_err_med rango
  <chr>         <dbl>     <dbl> <int>
1 audi          17.6       0.217     6
2 chevrolet     15        0.450    11
3 dodge         13.1       0.167     9
4 ford          14        0.147     7
5 honda         24.4       0.420     7
6 hyundai       18.6       0.161     5
7 jeep          13.5       0.786     8
8 land rover    11.5       0.0833    1
9 lincoln       11.3       0.111     1
10 mercury      13.2       0.0625    1
11 nissan        18.1       0.903    11
12 pontiac      17        0.2       2
13 subaru       19.3       0.0597    3
14 toyota       18.5       0.482    17
15 volkswagen   20.9       0.769    19
```

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

```
mpg %>%
  mutate(antes = if_else(year < 2004, "Antes 2004", "Después 2004")) %>%
  group_by(manufacturer, antes) %>%
  summarise(media = mean(cty, na.rm = TRUE))
```

```
# A tibble: 30 x 3
# Groups:   manufacturer [15]
  manufacturer antes      media
  <chr>         <chr>     <dbl>
1 audi          Antes 2004  17.1
2 audi          Después 2004 18.1
3 chevrolet     Antes 2004  15.1
```

```

4 chevrolet    Después 2004  14.9
5 dodge        Antes 2004   13.4
6 dodge        Después 2004 13.0
7 ford         Antes 2004   13.9
8 ford         Después 2004 14.1
9 honda        Antes 2004   24.8
10 honda       Después 2004  24
# ... with 20 more rows

```

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

```

mpg %>%
  filter(manufacturer %in% base::sample(unique(mpg$manufacturer), size = 3)) %>%
  group_by(manufacturer) %>%
  summarise(prom_hwy = mean(hwy, na.rm = TRUE)) %>%
  arrange(prom_hwy)

```

```

# A tibble: 3 x 2
  manufacturer prom_hwy
  <chr>         <dbl>
1 lincoln      17
2 mercury      18
3 volkswagen   29.2

```

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(litros_cada_100_kms = 100 * (hwy / 3.74541 * 1.60934)^(-1)) %>%
  select(manufacturer, model, year, hwy, litros_cada_100_kms)

```

```

# A tibble: 234 x 5
  manufacturer model    year  hwy litros_cada_100_kms
  <chr>         <chr>   <int> <int>         <dbl>
1 audi         a4      1999   29           8.03
2 audi         a4      1999   29           8.03
3 audi         a4      2008   31           7.51
4 audi         a4      2008   30           7.76
5 audi         a4      1999   26           8.95
6 audi         a4      1999   26           8.95
7 audi         a4      2008   27           8.62
8 audi         a4 quattro 1999   26           8.95
9 audi         a4 quattro 1999   25           9.31
10 audi        a4 quattro 2008   28           8.31
# ... with 224 more rows

```