

# Actividad 8

Luis Gagnevin

5/13/2021

## Ejercicio 1

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

```
library(tidyverse)
data(mpg)
```

1. Selecciona 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
```

En total son 6 vehiculos del 2008 con un hwy mayor a 31.

2. Seleccionar el conjunto de autos que tienen cty igual a 9 o hwy 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
```

Son en total 5 vehiculos con cty de 9 o hwy mayor a 90

3. Selecciona 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. Selecciona 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

```
select(filter(mpg, manufacturer=="toyota" & model=="camry"),
        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

Los agrupamos y luego hacemos un “Resumen” de los datos segun lo deseado. Posteriormente lo ordenamos y tenemos los valores hechos!

```
agrupacion<-mpg %>% group_by(manufacturer)
a<-agrupacion %>% summarise(promedio=mean(cty))
a<-a[order(-a$promedio),]
```

La marca con mejor rendimiento es: honda con 24.4444444444444 millas por galon

La marca con peor rendimiento es: lincoln con 11.3333333333333 millas por galon

**Comentario:** No es la forma de hacerlo, no hay necesidad de generar objetos intermedios. Si se quiere usar el texto bastaría con a nada más.

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

Ya tenemos la agrupacion por marca asi que no saltaremos ese paso!

Promedio por marca

```
agrupacion %>% summarise(promedio=mean(cty))
```

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

Error estandar de la media por marca en ciudad

```
agrupacion %>% summarise(error_estandar=sd(cty))
```

```
## # A tibble: 15 x 2
##   manufacturer error_estandar
##   * <chr>          <dbl>
## 1 audi             1.97
## 2 chevrolet        2.92
## 3 dodge            2.49
## 4 ford             1.91
## 5 honda            1.94
## 6 hyundai          1.50
## 7 jeep             2.51
```

```
## 8 land rover      0.577
## 9 lincoln         0.577
## 10 mercury        0.5
## 11 nissan          3.43
## 12 pontiac        1
## 13 subaru         0.914
## 14 toyota         4.05
## 15 volkswagen     4.56
```

Rango por marca de rendimiento en ciudad

```
k<-agrupacion %>% summarise(rango=range(cty))
k %>% summarise(rango_1=min(rango), rango_2=max(rango))
```

```
## # A tibble: 15 x 3
##   manufacturer rango_1 rango_2
##   * <chr>      <int>  <int>
## 1 audi         15      21
## 2 chevrolet    11      22
## 3 dodge        9      18
## 4 ford         11      18
## 5 honda        21      28
## 6 hyundai      16      21
## 7 jeep         9      17
## 8 land rover   11      12
## 9 lincoln      11      12
## 10 mercury     13      14
## 11 nissan       12      23
## 12 pontiac     16      18
## 13 subaru      18      21
## 14 toyota      11      28
## 15 volkswagen  16      35
```

**Comentario:** El calculo tiene que ser en el mismo chunk y en el mismo objeto, puedes (y es la idea en este ejercicio) hacer todo en el mismo summarise.

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

```
merge(filter(agrupacion, year<=2004) %>% summarise(rendimiento_promedio_antes_2004=mean(cty)),
filter(agrupacion, year>2004) %>% summarise(rendimiento_promedio_despues_2004=mean(cty)))
```

```
##   manufacturer rendimiento_promedio_antes_2004
## 1      audi      17.11111
## 2  chevrolet      15.14286
## 3    dodge      13.37500
## 4     ford      13.93333
## 5     honda      24.80000
## 6   hyundai      18.33333
## 7     jeep      14.50000
## 8  land rover      11.00000
## 9   lincoln      11.00000
## 10  mercury      13.50000
## 11   nissan      17.66667
## 12  pontiac      17.00000
```

```
## 13      subaru      19.00000
## 14      toyota      18.15000
## 15    volkswagen    21.25000
##      rendimiento_promedio_despues_2004
## 1      18.11111
## 2      14.91667
## 3      12.95238
## 4      14.10000
## 5      24.00000
## 6      18.87500
## 7      13.16667
## 8      12.00000
## 9      12.00000
## 10     13.00000
## 11     18.42857
## 12     17.00000
## 13     19.50000
## 14     19.07143
## 15     20.45455
```

**Comentario:** No es incorrecto pero (de nuevo) tiene que quedar todo en un objeto.

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

Para poder hacer esto, tenemos que tomar 3 aleatorios pero se repiten las marcas si no las unificamos por lo que primero debemos unificarlas.

```
marcas<- agrupacion %>% summarise(mean(cty))
marcas<-sample(marcas$manufacturer,3)
promedio2 <- agrupacion %>% filter(manufacturer==marcas) %>% summarise(promedio=mean(hwy))
promedio2[order(promedio2$promedio),]
```

```
## # A tibble: 3 x 2
##   manufacturer promedio
##   <chr>          <dbl>
## 1 ford          20.2
## 2 hyundai       27.2
## 3 volkswagen    29.2
```

**Comentario:** Revisar la solución. Si primero haces un summarise y luego filtras estas haciendo un calculo para toda la columna que luego no vas a usar. Primero filtras (reducís dimensión en filas) y luego computas. Por último, usar set.seed para reproducir resultado.

### 10. Crea una nueva variable que transforme hwy en litros/100km

```
head(merge(mpg,mpg %>% summarise("litros/100km"=hwy/235.214583)))
```

```
##   manufacturer model displ year cyl      trans drv  cty  hwy  fl   class
## 1      audi      a4   1.8 1999   4    auto(l5)  f   18   29   p compact
## 2      audi      a4   1.8 1999   4 manual(m5)  f   21   29   p compact
## 3      audi      a4   2.0 2008   4 manual(m6)  f   20   31   p compact
## 4      audi      a4   2.0 2008   4    auto(av)  f   21   30   p compact
## 5      audi      a4   2.8 1999   6    auto(l5)  f   16   26   p compact
```

```
## 6      audi      a4    2.8 1999    6 manual(m5)    f  18  26  p compact
##      litros/100km
## 1      0.1232917
## 2      0.1232917
## 3      0.1232917
## 4      0.1232917
## 5      0.1232917
## 6      0.1232917
```

**Comentario:** Si bien se llega a los resultados finales se debe mejorar en la forma de hacer los calculos y no generar objetos intermedios cuando no son necesarios. Además de computar en un mismo objeto en vez de generar varios.<sup>8</sup>