Manipulacion y transformacion de datos con dplyr: Parte 1

En esta oportunidad nos vamos a enfocar en como usar RStudio y como usar `dplyr`

Carga de librerias

Primero instalamos y cargamos las librerias que requerimos.

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("gapminder")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gapminder)
```

Ahora visualizemos el conjunto de datos con el que vamos a trabajar

```
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

Seleccionando un subconjunto de los datos

```
canada <- gapminder[241:252, ]
canada
```

```
## # A tibble: 12 x 6
##    country continent  year lifeExp      pop gdpPercap
##    <fct>   <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Canada  Americas   1952    68.8 14785584    11367.
##  2 Canada  Americas   1957    70.0 17010154    12490.
##  3 Canada  Americas   1962    71.3 18985849    13462.
##  4 Canada  Americas   1967    72.1 20819767    16077.
##  5 Canada  Americas   1972    72.9 22284500    18971.
##  6 Canada  Americas   1977    74.2 23796400    22091.
##  7 Canada  Americas   1982    75.8 25201900    22899.
##  8 Canada  Americas   1987    76.9 26549700    26627.
##  9 Canada  Americas   1992    78.0 28523502    26343.
## 10 Canada  Americas   1997    78.6 30305843    28955.
## 11 Canada  Americas   2002    79.8 31902268    33329.
## 12 Canada  Americas   2007    80.7 33390141    36319.
```

La función `filter`

La funcion filter nos ayuda a filtrar por filas un Dataframe.

Filtremos por expectativa de vida

```
filter(gapminder, lifeExp < 29)
```

```
## # A tibble: 2 x 6
##   country     continent  year lifeExp     pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>   <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8 8425333      779.
## 2 Rwanda      Africa     1992    23.6 7290203      737.
```

Filtremos por pais y por año

```
filter(gapminder, country == "Argentina", year > 1979)
```

```
## # A tibble: 6 x 6
##   country   continent  year lifeExp      pop gdpPercap
##   <fct>     <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Argentina Americas   1982    69.9 29341374     8998.
## 2 Argentina Americas   1987    70.8 31620918     9140.
## 3 Argentina Americas   1992    71.9 33958947     9308.
## 4 Argentina Americas   1997    73.3 36203463    10967.
## 5 Argentina Americas   2002    74.3 38331121     8798.
## 6 Argentina Americas   2007    75.3 40301927    12779.
```

Filtermos dos paises a la vez usando un "truco"

```
filter(gapminder, country %in% c("Colombia", "Peru", "Ecuador"))
```

```
## # A tibble: 36 x 6
##    country  continent  year lifeExp      pop gdpPercap
##    <fct>    <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Colombia Americas   1952    50.6 12350771     2144.
##  2 Colombia Americas   1957    55.1 14485993     2324.
```

```
##  3 Colombia Americas   1962    57.9 17009885     2492.
##  4 Colombia Americas   1967    60.0 19764027     2679.
##  5 Colombia Americas   1972    61.6 22542890     3265.
##  6 Colombia Americas   1977    63.8 25094412     3816.
##  7 Colombia Americas   1982    66.7 27764644     4398.
##  8 Colombia Americas   1987    67.8 30964245     4903.
##  9 Colombia Americas   1992    68.4 34202721     5445.
## 10 Colombia Americas   1997    70.3 37657830     6117.
## # ... with 26 more rows
```

Este operador `%in%` es usado para identificar elementos que pertenecen a un vector o un Dataframe

Nuestros ejemplos anteriores muestran que hacer esto `canada <- gapminder[241:252, ]` es en general una mala idea ya que no hace de forma explícita las decisiones que se toman para hacer el subconjunto. Es mucho mejor hacer esto

```
filter(gapminder, country == "Canada")
```

```
## # A tibble: 12 x 6
##    country continent  year lifeExp      pop gdpPercap
##    <fct>   <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Canada  Americas   1952    68.8 14785584    11367.
##  2 Canada  Americas   1957    70.0 17010154    12490.
##  3 Canada  Americas   1962    71.3 18985849    13462.
##  4 Canada  Americas   1967    72.1 20819767    16077.
##  5 Canada  Americas   1972    72.9 22284500    18971.
##  6 Canada  Americas   1977    74.2 23796400    22091.
##  7 Canada  Americas   1982    75.8 25201900    22899.
##  8 Canada  Americas   1987    76.9 26549700    26627.
##  9 Canada  Americas   1992    78.0 28523502    26343.
## 10 Canada  Americas   1997    78.6 30305843    28955.
## 11 Canada  Americas   2002    79.8 31902268    33329.
## 12 Canada  Americas   2007    80.7 33390141    36319.
```

Operador Pipe %>%

Se importa desde el paquete magrittr

```
gapminder %>% head()
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

Lo anterior es equivalente a escribir `head(gapminder)`. El operador pipe toma lo que esta en la izquierda y lo usa en lo que esta definido en derecha. Lo usa como el primer **argumento**.

Ahora bien, ¿Cómo especificamos un argumento dentro de la funcion?

```
gapminder %>% head(3)
```

```
## # A tibble: 3 x 6
```

```
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
```

Funcion `select()`

Con esta función podemos crear un subconjunto del Dtaframe por columnas o variables

```
select(gapminder, year, lifeExp)
```

```
## # A tibble: 1,704 x 2
##     year lifeExp
##    <int>   <dbl>
##  1  1952    28.8
##  2  1957    30.3
##  3  1962    32.0
##  4  1967    34.0
##  5  1972    36.1
##  6  1977    38.4
##  7  1982    39.9
##  8  1987    40.8
##  9  1992    41.7
## 10  1997    41.8
## # ... with 1,694 more rows
```

```
gapminder %>%
  select(year, lifeExp) %>%
  head(4)
```

```
## # A tibble: 4 x 2
##    year lifeExp
##   <int>   <dbl>
## 1  1952    28.8
## 2  1957    30.3
## 3  1962    32.0
## 4  1967    34.0
```

```
gapminder %>%
  filter(country == "Chile") %>%
  select(year, lifeExp)
```

```
## # A tibble: 12 x 2
##     year lifeExp
##    <int>   <dbl>
##  1  1952    54.7
##  2  1957    56.1
##  3  1962    57.9
##  4  1967    60.5
##  5  1972    63.4
##  6  1977    67.1
##  7  1982    70.6
##  8  1987    72.5
##  9  1992    74.1
## 10  1997    75.8
## 11  2002    77.9
```

```
## 12  2007    78.6
```