Agregar y resumir

HR Analytics: Teoría y Práctica

http://pablohaya.com/contact

09/2018

Preparación

Leemos el conjunto de datos que ya habíamos limpiado anteriormente.

```
library(here)
library(tidyverse)
load(here("data","join_dataset_clean.RData"))
head(df)
```

```
## # A tibble: 6 x 28
##   EmployeeName EmployeeNumber State Zip   DOB           Age
##   <chr>        <chr>          <fct> <fct> <date>      <int>
## 1 King, Janet  1001495124     MA    01902 1954-09-21     63
## 2 Albert, Mic~ 1501072311     MA    02169 1968-10-10     49
## 3 Bozzi, Char~ 1303054580     MA    01901 1970-03-10     47
## 4 Butler, Web~ 1110029990     MA    02169 1983-08-09     34
## 5 Dunn, Amy    1409070147     MA    01731 1973-11-28     44
## 6 Gray, Eliji~ 1307060077     MA    01752 1981-07-11     36
```

Agregando

La función `summarise()` agrega todos los valores de una columna según la función que indiquemos generando una nueva tabla.

```
df %>% summarise(mean(PayRate))
```

```
## # A tibble: 1 x 1
##   `mean(PayRate)`
##             <dbl>
## 1            23.2
```

Es posible incluir más de un función para realizar distintos resumenes.

```
df %>% summarise(mean(PayRate),
                 median(PayRate),
                 var(PayRate))
```

```
## # A tibble: 1 x 3
##   `mean(PayRate)` `median(PayRate)` `var(PayRate)`
##             <dbl>             <dbl>          <dbl>
## 1            23.2                22           85.0
```

o incluir dos o más variables distintas en el resumen.

```
df %>% summarise(mean(PayRate), mean(Age))
```

```
## # A tibble: 1 x 2
##   `mean(PayRate)` `mean(Age)`
##             <dbl>       <dbl>
## 1            23.2        39.2
```

También es posible personalizar el nombre de las columnas de la nueva tabla.

```
df %>% summarise(media=mean(PayRate),
                 mediana=median(PayRate),
                 var=var(PayRate))
```

```
## # A tibble: 1 x 3
##   media mediana   var
##   <dbl>   <dbl> <dbl>
## 1  23.2      22  85.0
```

Group by

La función group_by() permite crear grupos sobre los que realizar la agregación. Aparentemente no modifica la tabla aunque internamente queda preparada para las siguentes operaciones.

```
by_sex <- df %>% group_by(Sex)
by_sex
```

```
## # A tibble: 209 x 28
## # Groups:   Sex [2]
##    EmployeeName EmployeeNumber State Zip   DOB          Age
##    <chr>        <chr>          <fct> <fct> <date>     <int>
##  1 King, Janet  1001495124     MA    01902 1954-09-21    63
##  2 Albert, Mic~ 1501072311     MA    02169 1968-10-10    49
##  3 Bozzi, Char~ 1303054580     MA    01901 1970-03-10    47
##  4 Butler, Web~ 1110029990     MA    02169 1983-08-09    34
##  5 Dunn, Amy    1409070147     MA    01731 1973-11-28    44
```

Agregando por grupos

La función `summarise()` realiza la operación indicada para cada uno de los grupos especificados por `group_by()`. El resultado es una tabla que resume los resultados en cada fila para cada uno de los grupos.

```
df %>% group_by(Sex) %>%
       summarise(PayRate = mean(PayRate))
```

```
## # A tibble: 2 x 2
##   Sex    PayRate
##   <fct>    <dbl>
## 1 Female    22.8
## 2 Male      23.9
```

Tal como hemos visto es posible incluir más de una función y/o columna sobre los que realizar el resumen.

```
df %>% group_by(Sex) %>%
       summarise(PayRate = mean(PayRate), Age = mean(Age))
```

```
## # A tibble: 2 x 3
##   Sex    PayRate   Age
##   <fct>    <dbl> <dbl>
## 1 Female    22.8  39.6
## 2 Male      23.9  38.7
```

Contar el número de elementos en un grupo

La funció `n()` permite contar el número de elementos por cada grupo.

```
df %>% group_by(Sex) %>%
       summarise(count = n())
```

```
## # A tibble: 2 x 2
##   Sex    count
##   <fct>  <int>
## 1 Female   127
## 2 Male      82
```

Incluyendo más de una variable en la función `group_by()` se realizan las agrupaciones para todas las combinaciones posibles de valores.

```
df %>% group_by(Sex, Age) %>%
       summarise(count = n())
```

```
## # A tibble: 65 x 3
## # Groups:   Sex [?]
##    Sex      Age count
##    <fct>  <int> <int>
##  1 Female    25     2
##  2 Female    26     1
##  3 Female    27     4
##  4 Female    28     5
##  5 Female    29     4
##  6 Female    30     4
##  7 Female    31    11
##  8 Female    32     5
##  9 Female    33     6
```

Resumir el pasado (Window Functions)

`tidyverse` incluye varias funciones que permiten resumir empleando los valores pasados (`cumsum()`, `cummean()`, `lag()`, `lead()`...).

Vamos a calcular la plantilla para cada año, y el crecimiento respecto al año anterior. Primeramente calculamos el número de contratados acumulado año a año.

```
df_c <- df %>% group_by(year=format(DateofHire, "%Y")) %>%
            summarise(count = n()) %>%
            mutate(contratacion=cumsum(count)) %>%
            select(year, contratacion)
df_c
```

```
## # A tibble: 10 x 2
##    year  contratacion
##    <chr>        <int>
##  1 2007             2
```

A continuación, calculamos el acumulado del número de bajas.

```
df_e <- df %>% group_by(year=format(DateofTermination, "%Y"))
               summarise(count = n()) %>%
               mutate(despidos=cumsum(count)) %>%
               filter(!is.na(year)) %>%
               select(year, despidos)
df_e
```

```
## # A tibble: 7 x 2
##   year  despidos
##   <chr>    <int>
## 1 2010         2
## 2 2011        16
## 3 2012        33
## 4 2013        46
## 5 2014        56
## 6 2015        72
## 7 2016        83
```

Finalmente, juntamos las dos tablas anteriores, calculamos la plantilla para cada año, y el crecimiento.

```
library(scales)
df_c %>%
    full_join(df_e, by="year") %>%
    mutate(despidos=ifelse(is.na(despidos), 0, despidos)) %>%
    mutate(plantilla=contratacion-despidos) %>%
    mutate(plantilla_1=lag(plantilla)) %>%
    mutate(crecimiento=(plantilla-plantilla_1)/plantilla) %>%
    mutate(crecimiento=percent(crecimiento)) %>%
    select(year, plantilla, crecimiento)
```

```
## # A tibble: 10 x 3
##    year  plantilla crecimiento
##    <chr>     <dbl> <chr>
##  1 2007          2 NA%
##  2 2008          4 50.0%
##  3 2009          9 55.6%
```