stringr

Juan Pablo Delzo

> Paquete en ***R*** que tiene como objectivo editar, transformar y extraer información respecto a las variables cadenas en un dataframe

In [1]:
```
library(stringr)
```

In [2]:
```
#Ejemplo
df<-read.csv('string.csv')
df
```

Company	Slogan	Address
Emard-Weimann	monetize virtual functionalities	13029 Canary Trail
Halvorson, Cremin and Tremblay	empower killer markets	51 Sommers Hill
Davis, Rutherford and Reilly	disintermediate killer communities	0 Sage Way
Harvey-Ernser	harness end-to-end eyeballs	95 Maple Wood Road
Muller, Jakubowski and Kuphal	integrate rich users	697 Lakeland Road
Green and Sons	redefine scalable infrastructures	3016 Claremont Court
Oberbrunner, Bode and Casper	optimize magnetic applications	3 Hermina Drive
Schumm-Kertzmann	optimize out-of-the-box technologies	7605 Atwood Drive
Adams, Satterfield and Kemmer	leverage collaborative convergence	9 2nd Plaza
Strosin and Sons	transition 24/365 e-services	9030 Lunder Street

In [3]:
```
library(stringr)
```

str_detect

Función boleana, responde si existe una sub-cadena en la cadena

In [4]:
```
str_detect(df$Company,'Inc')
```

```
FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
```

str_which

Selecciona el índice de la instancia donde contenga la sub-cadena

In [5]:
```
str_which(df$Address,'Roa')
```

```
4  5
```

str_count

- Determina la longitud de cada cadena

```
In [6]: str_count(df$Slogan)
```

```
32   22   34   27   20   33   30   36   34   28
```

- Determina el número de veces que se repite la sub-cadena

```
In [7]: str_count(df$Company,'a')
```

```
2   3   2   1   3   1   2   1   3   1
```

str_locate

Determina el punto inicial y final de la ubicación de la sub-cadena en cada instancia

```
In [8]: str_locate(df$Address,'Road')
```

```
start   end
   NA    NA
   NA    NA
   NA    NA
   15    18
   14    17
   NA    NA
   NA    NA
   NA    NA
   NA    NA
   NA    NA
```

str_sub

Selecciona sub-cadena en función de la posición inicial y final definida

```
In [9]: str_sub(df$Company,start=2,end=10)
```

```
'mard-Weim'   'alvorson,'   'avis, Rut'   'arvey-Ern'   'uller, Ja'   'reen and '   'berbrunne'
'chumm-Ker'   'dams, Sat'   'trosin an'
```

str_subset

Extrae las cadenas que contienen las sub-cadena definida

```
In [10]: str_subset(df$Company,'and')
```

```
'Halvorson, Cremin and Tremblay'   'Davis, Rutherford and Reilly'   'Muller, Jakubowski and Kuphal'
'Green and Sons'   'Oberbrunner, Bode and Casper'   'Adams, Satterfield and Kemmer'
'Strosin and Sons'
```

str_match

Aparece la sub-cadena solo si la sub-cadena se encuentra dentro de la instancia. En caso contrario, solo aparece `NA`

In [11]:
```
str_match(df$Slogan,'-ed')
```

```
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
```

str_length

Cuenta el número de caracteres en cada instancia

In [12]:
```
str_length(df$Address)
```

```
18   15   10   18   17   20   15   17   11   18
```

str_sub

Reemplaza la sub-cadena en función de la posición inicial y final definida

In [13]:
```
str_sub(df$Slogan,1,3)<- '741'
df$Slogan
```

```
'741etize virtual functionalities'   '741ower killer markets'   '741intermediate killer communities'
'741ness end-to-end eyeballs'   '741egrate rich users'   '741efine scalable infrastructures'
'741imize magnetic applications'   '741imize out-of-the-box technologies'
'741erage collaborative convergence'   '741nsition 24/365 e-services'
```

str_replace

Reemplaza el caracter a otro definido

In [14]:
```
str_replace(df$Company,'a','@')
```

```
'Em@rd-Weimann'   'H@lvorson, Cremin and Tremblay'   'D@vis, Rutherford and Reilly'
'H@rvey-Ernser'   'Muller, J@kubowski and Kuphal'   'Green @nd Sons'
'Oberbrunner, Bode @nd Casper'   'Schumm-Kertzm@nn'   'Ad@ms, Satterfield and Kemmer'
'Strosin @nd Sons'
```

str_to_lower

Cambia a minúsculas toda la cadena

```
In [15]: str_to_lower(df$Company)
```

```
'emard-weimann'   'halvorson, cremin and tremblay'   'davis, rutherford and reilly'   'harvey-ernser'
'muller, jakubowski and kuphal'   'green and sons'   'oberbrunner, bode and casper'
'schumm-kertzmann'   'adams, satterfield and kemmer'   'strosin and sons'
```

str_to_upper

Cambia a mayúsculas toda la cadena

```
In [16]: str_to_upper(df$Company)
```

```
'EMARD-WEIMANN'   'HALVORSON, CREMIN AND TREMBLAY'
'DAVIS, RUTHERFORD AND REILLY'   'HARVEY-ERNSER'
'MULLER, JAKUBOWSKI AND KUPHAL'   'GREEN AND SONS'
'OBERBRUNNER, BODE AND CASPER'   'SCHUMM-KERTZMANN'
'ADAMS, SATTERFIELD AND KEMMER'   'STROSIN AND SONS'
```

str_to_title

Convierte el primer caracter despues de un espacio en blanco a mayúscula

```
In [17]: str_to_title(df$Company)
```

```
'Emard-Weimann'   'Halvorson, Cremin And Tremblay'   'Davis, Rutherford And Reilly'
'Harvey-Ernser'   'Muller, Jakubowski And Kuphal'   'Green And Sons'
'Oberbrunner, Bode And Casper'   'Schumm-Kertzmann'   'Adams, Satterfield And Kemmer'
'Strosin And Sons'
```

str_c

Une dos cadenas

```
In [18]: str_c(df$Company,df$Address,sep = ':')
```

```
'Emard-Weimann:13029 Canary Trail'   'Halvorson, Cremin and Tremblay:51 Sommers Hill'
'Davis, Rutherford and Reilly:0 Sage Way'   'Harvey-Ernser:95 Maple Wood Road'
'Muller, Jakubowski and Kuphal:697 Lakeland Road'   'Green and Sons:3016 Claremont Court'
'Oberbrunner, Bode and Casper:3 Hermina Drive'   'Schumm-Kertzmann:7605 Atwood Drive'
'Adams, Satterfield and Kemmer:9 2nd Plaza'   'Strosin and Sons:9030 Lunder Street'
```

str_dup

Repite el número de veces la concatenación de una misma cadena sin ningún espacio entre ellos

```
In [19]: str_dup(df$Company,2)
```

```
'Emard-WeimannEmard-Weimann'
'Halvorson, Cremin and TremblayHalvorson, Cremin and Tremblay'
'Davis, Rutherford and ReillyDavis, Rutherford and Reilly'   'Harvey-ErnserHarvey-Ernser'
'Muller, Jakubowski and KuphalMuller, Jakubowski and Kuphal'   'Green and SonsGreen and Sons'
'Oberbrunner, Bode and CasperOberbrunner, Bode and Casper'
'Schumm-KertzmannSchumm-Kertzmann'
'Adams, Satterfield and KemmerAdams, Satterfield and Kemmer'
'Strosin and SonsStrosin and Sons'
```

str_sort

Ordena todo el set de cadenas por orden alfabético en función del primer caracter

```
In [20]: str_sort(df$Company)
```

```
'Adams, Satterfield and Kemmer'   'Davis, Rutherford and Reilly'   'Emard-Weimann'
'Green and Sons'   'Halvorson, Cremin and Tremblay'   'Harvey-Ernser'
'Muller, Jakubowski and Kuphal'   'Oberbrunner, Bode and Casper'   'Schumm-Kertzmann'
'Strosin and Sons'
```

str_split

Rompe la cadena de acuerdo al caracter establecido convertiendolo a lista

```
In [21]: df$Company_list=str_split(df$Company,'[ -]')
         #Se está rompiendo la cadena en función de un espacio en blanco (' ') o '-'
         df
```

Company	Slogan	Address	Company_list
Emard-Weimann	741etize virtual functionalities	13029 Canary Trail	Emard , Weimann
Halvorson, Cremin and Tremblay	741ower killer markets	51 Sommers Hill	Halvorson,, Cremin , and , Tremblay
Davis, Rutherford and Reilly	741intermediate killer communities	0 Sage Way	Davis, , Rutherford, and , Reilly
Harvey-Ernser	741ness end-to-end eyeballs	95 Maple Wood Road	Harvey, Ernser
Muller, Jakubowski and Kuphal	741egrate rich users	697 Lakeland Road	Muller, , Jakubowski, and , Kuphal
Green and Sons	741efine scalable infrastructures	3016 Claremont Court	Green, and , Sons
Oberbrunner, Bode and Casper	741imize magnetic applications	3 Hermina Drive	Oberbrunner,, Bode , and , Casper
Schumm-Kertzmann	741imize out-of-the-box technologies	7605 Atwood Drive	Schumm , Kertzmann
Adams, Satterfield and Kemmer	741erage collaborative convergence	9 2nd Plaza	Adams, , Satterfield, and , Kemmer
Strosin and Sons	741nsition 24/365 e-services	9030 Lunder Street	Strosin, and , Sons

Ejemplo:
Se desea crear un atributo del primer nombre de la empresa.
Para ello se deberá utilizar un bucle que reemplaze la posicion 1 en cada lista de `Company_list` a toda la lista y luego cambiar el nombre del atributo a `Company_firstname`

In [22]:
```
for(index in 1:dim(df)[1]){df$Company_list[index]<- df$Company_list[[index]][1]}
colnames(df)[4]<- 'Company_firstname'
#Definiendo  al atributo que corresponde a una matriz columna
df$Company_firstname <- as.vector(df$Company_firstname)
#Limpiando las comas adjuntas a los nombres principales
df$Company_firstname <- str_replace(df$Company_firstname,',','')
df
```

Company	Slogan	Address	Company_firstname
Emard-Weimann	741etize virtual functionalities	13029 Canary Trail	Emard
Halvorson, Cremin and Tremblay	741ower killer markets	51 Sommers Hill	Halvorson
Davis, Rutherford and Reilly	741intermediate killer communities	0 Sage Way	Davis
Harvey-Ernser	741ness end-to-end eyeballs	95 Maple Wood Road	Harvey
Muller, Jakubowski and Kuphal	741egrate rich users	697 Lakeland Road	Muller
Green and Sons	741efine scalable infrastructures	3016 Claremont Court	Green
Oberbrunner, Bode and Casper	741imize magnetic applications	3 Hermina Drive	Oberbrunner
Schumm-Kertzmann	741imize out-of-the-box technologies	7605 Atwood Drive	Schumm
Adams, Satterfield and Kemmer	741erage collaborative convergence	9 2nd Plaza	Adams
Strosin and Sons	741nsition 24/365 e-services	9030 Lunder Street	Strosin