

# Club de lectura

"Efficient R programming", de Colin Gillespie y Robin Lovelace.

## MANIPULACIÓN

## EFICIENTE DE DATOS

*Capítulo 6.1 - 6.3*

**Fecha:** Miércoles, 08 de junio

**Hora:** 18:00 a 19:00 UTC-5

**Lugar:** [ZOOM](#)

**Inscripción:** [Meetup](#)



## Conferencistas:



Mary Jane  
Rivero Morales



Viviana Carolina  
Flórez Camacho

Organiza:



Semillero  
Unal

Colaboran:



R-Ladies Medellín +  
Barranquilla + Galápagos

# Orden del día



- 6.1. Consejos Generales.**
- 6.2. Marcos de datos eficientes con tibble()**
- 6.3. Ordenar datos con tidyr y expresiones regulares.**

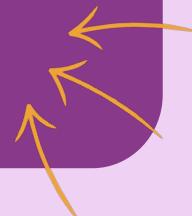


## 6. Carpintería eficiente de datos.



*Paquetes Necesarios:*

```
library("tibble")  
library("tidyr")  
library(stringr)  
library(readr)  
library(dplyr)  
library(broom)
```



## *6.1 Top cinco para una carpintería eficiente de datos*

- 1** El tiempo dedicado a preparar sus datos al principio puede ahorrar horas de frustración a largo plazo.
- 2** "Datos ordenados" proporciona un concepto para organizar los datos y el orden de paquetes proporciona algunas funciones para este trabajo.
- 3** La clase `data_frame` definida por el paquete `tibble` hace que los conjuntos de datos se impriman de forma eficiente y sea fácil trabajar con ellos

## 6.1 Top cinco para una carpintería eficiente de datos

- 4 **dplyr** proporciona funciones de procesamiento de datos rápidas e intuitivas; data.table tiene una velocidad inigualable para algunas aplicaciones de procesamiento de datos.
- 5 El operador **%>%** 'pipe' puede ayudar a clarificar los flujos de trabajo de procesamiento de datos complejos.



F•R•I•E•N•D•S

## 6.2 Marcos de datos eficientes con tibble

1

Al imprimir en la función Tibble se puede ver la clase de cada variable, mientras que con data.frame, no.

# A tibble: 6 × 4				
	Personaje	Profesion	Nacimiento	Sexo
1	Rachel Green	Ejecutiva	5-05-19	F
2	Ross Geller	Paleontólogo	18-10-19	M
3	Monica Geller	Chef	22-04-19	F
4	Joey Tribbiani	Actor	25-05-19	M
5	Chandler Bing	Analista de datos	8-04-19	M
6	Phoebe Buffay	Masajista	10-10-19	F

```
> df #con dataframe
#> #> #> Personaje      Profesion  as.Date.Nacimiento. Sexo
#> 1  Rachel Green    Ejecutiva   5-05-19     F
#> 2  Ross Geller    Paleontólogo 18-10-19     M
#> 3  Monica Geller   Chef        22-04-19     F
#> 4  Joey Tribbiani Actor       25-05-19     M
#> 5  Chandler Bing   Analista de datos 8-04-19     M
#> 6  Phoebe Buffay   Masajista  10-10-19     F
```



## 6.2 Marcos de datos eficientes con tibble

```
Console Terminal x Jobs x
R 4.2.0 - /cloud/project/ ✎
# A tibble: 6 x 4
  Personaje     Profesion
  <chr>        <chr>
1 Rachel Green Ejecutiva
2 Ross Geller  Paleontólogo
3 Monica Geller Chef
4 Joey Tribbiani Actor
5 Chandler Bing Analista de ...
6 Phoebe Buffay Masajista
# ... with 2 more variables:
#   Nacimiento <date>,
#   Sexo <chr>
> |
```



2

Los tibbles tienen un método de impresión en la consola refinado: solo muestran las primeras 10 filas y solo aquellas columnas que entran en el ancho de la pantalla.

```
> df #con dataframe
```

	Personaje	Profesion	as.Date.Nacimiento.	Sexo
1	Rachel Green	Ejecutiva	5-05-19	F
2	Ross Geller	Paleontólogo	18-10-19	M
3	Monica Geller	Chef	22-04-19	F
4	Joey Tribbiani	Actor	25-05-19	M
5	Chandler Bing	Analista de datos	8-04-19	M
6	Phoebe Buffay	Masajista	10-10-19	F

## 6.2 Marcos de datos eficientes con tibble

- ③ Podemos extraer información de los tibbles con herramientas como \$ y [[

```
df[,1]
```

```
[1] "Rachel Green"   "Ross Geller"  
[3] "Monica Geller" "Joey Tribbiani"  
[5] "Chandler Bing" "Phoebe Buffay"
```

```
my_tibble[,1]
```

```
# A tibble: 6 × 1  
Personaje  
<chr>  
1 Rachel Green  
2 Ross Geller  
3 Monica Geller  
4 Joey Tribbiani  
5 Chandler Bing  
6 Phoebe Buffay
```

# 6.3 Ordenar datos con *tidyverse* y expresiones regulares



Existen tres reglas interrelacionadas que hacen que un conjunto de datos sea ordenado:

- 1 Cada variable debe tener su propia columna.

Personaje	Profesión	Nacimiento	Sexo
Rachel Green	Ejecutiva	05-05-1971	F
Ross Geller	Paleontólogo	18-10-1967	M
Monica Geller	Chef	22-04-1969	F
Joey Tribbiani	Actor	25-05-1968	M
Chandler Bing	Analista de datos	08-04-1969	M
Phoebe Buffay	Masajista	10-10-1968	F

- 2 Cada observación forma una fila.

Personaje	Profesión	Nacimiento	Sexo
Rachel Green	Ejecutiva	05-05-1971	F
Ross Geller	Paleontólogo	18-10-1967	M
Monica Geller	Chef	22-04-1969	F
Joey Tribbiani	Actor	25-05-1968	M
Chandler Bing	Analista de datos	08-04-1969	M
Phoebe Buffay	Masajista	10-10-1968	F

- 3 Cada valor debe tener su propia celda.

Personaje	Profesión	Nacimiento	Sexo
Rache●Green	Ejecutiv●	05-0●1971	F●
Ross ●Geller	Paleontolog●	18-1●1967	M●
Monic●Geller	Chef●	22-0●1969	F●
Joey ●Tribbiani	Actor●	25-0●1968	M●
Chand●er Bing	Analista de datos	08-0●1969	M●
Phoeb●Buffay	Masajis●	10-1●1968	F●



## 6.3.1 De datos anchos a largos con pivot\_longer()

```
> my_tibble  
# A tibble: 6 × 4  
  Personaje Profesion Nacimiento Sexo  
  <chr>     <chr>      <chr>     <chr>  
1 Rachel Green Ejecutiva 05-05-1971 F  
2 Ross Geller Paleontologo 18-10-1967 M  
3 Monica Geller Chef 22-04-1969 F  
4 Joey Tribbiani Actor 25-05-1968 M  
5 Chandler Bing Analista de datos 08-04-1969 M  
6 Phoebe Buffay Masajista 10-10-1968 F
```



```
pivot_longer(data = my_tibble,  
             cols = c("Profesion", "Nacimiento", "Sexo"),  
             names_to = "Variables",  
             values_to = "Valor")
```



```
> data  
# A tibble: 18 × 3  
  Personaje Variables Valor  
  <chr>     <chr>     <chr>  
1 Rachel Green Profesion Ejecutiva  
2 Rachel Green Nacimiento 05-05-1971  
3 Rachel Green Sexo F  
4 Ross Geller Profesion Paleontologo  
5 Ross Geller Nacimiento 18-10-1967  
6 Ross Geller Sexo M  
7 Monica Geller Profesion Chef  
8 Monica Geller Nacimiento 22-04-1969  
9 Monica Geller Sexo F  
10 Joey Tribbiani Profesion Actor  
11 Joey Tribbiani Nacimiento 25-05-1968  
12 Joey Tribbiani Sexo M  
13 Chandler Bing Profesion Analista de datos  
14 Chandler Bing Nacimiento 08-04-1969  
15 Chandler Bing Sexo M  
16 Phoebe Buffay Profesion Masajista  
17 Phoebe Buffay Nacimiento 10-10-1968  
18 Phoebe Buffay Sexo F
```

# *pivot\_wider()*

```
> data
# A tibble: 18 × 3
  Personaje    Variables  Valor
  <chr>        <chr>      <chr>
1 Rachel Green Profesion Ejecutiva
2 Rachel Green Nacimiento 05-05-1971
3 Rachel Green Sexo       F
4 Ross Geller   Profesion Paleontologo
5 Ross Geller   Nacimiento 18-10-1967
6 Ross Geller   Sexo       M
7 Monica Geller Profesion Chef
8 Monica Geller Nacimiento 22-04-1969
9 Monica Geller Sexo       F
10 Joey Tribbiani Profesion Actor
11 Joey Tribbiani Nacimiento 25-05-1968
12 Joey Tribbiani Sexo       M
13 Chandler Bing Profesion Analista de datos
14 Chandler Bing Nacimiento 08-04-1969
15 Chandler Bing Sexo       M
16 Phoebe Buffay Profesion Masajista
17 Phoebe Buffay Nacimiento 10-10-1968
18 Phoebe Buffay Sexo       F
```

```
> my_tibble
# A tibble: 6 × 4
  Personaje    Profesion      Nacimiento Sexo
  <chr>        <chr>          <chr>      <chr>
1 Rachel Green Ejecutiva    05-05-1971 F
2 Ross Geller  Paleontologo 18-10-1967 M
3 Monica Geller Chef         22-04-1969 F
4 Joey Tribbiani Actor       25-05-1968 M
5 Chandler Bing  Analista de datos 08-04-1969 M
6 Phoebe Buffay Masajista    10-10-1968 F
```

```
pivot_wider(data = data,
            id_cols = 'Personaje',
            names_from = 'Variables',
            values_from = 'Valor')
```



## 6.3.2 Dividir variables con separate()

```
> my_tibble  
# A tibble: 6 × 4  
  Personaje    Profesion     Nacimiento Sexo  
  <chr>        <chr>       <chr>      <chr>  
1 Rachel Green Ejecutiva 05-05-1971 F  
2 Ross Geller  Paleontologo 18-10-1967 M  
3 Monica Geller Chef 22-04-1969 F  
4 Joey Tribbiani Actor 25-05-1968 M  
5 Chandler Bing Analista de datos 08-04-1969 M  
6 Phoebe Buffay Masajista 10-10-1968 F
```

```
# A tibble: 6 × 5  
  Nombre    Apellido Profesion     Nacimiento Sexo  
  <chr>    <chr>    <chr>       <chr>      <chr>  
1 Rachel   Green    Ejecutiva 05-05-1971 F  
2 Ross     Geller   Paleontologo 18-10-1967 M  
3 Monica   Geller   Chef 22-04-1969 F  
4 Joey     Tribbiani Actor 25-05-1968 M  
5 Chandler Bing Analista de datos 08-04-1969 M  
6 Phoebe   Buffay  Masajista 10-10-1968 F
```

1



```
my_tibble %>%  
  separate(Personaje,  
           into=c("Nombre", "Apellido"),  
           sep=" ")
```

2

## 6.3.3 Otras funciones de *tidy*

Funciones de Tidyr para ordenar los resultados de un modelo y facilitar su interpretación.

### ① tidy()

```
> summary(glmfit)

Call:
glm(formula = am ~ wt, family = "binomial", data = mtcars)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-2.11400 -0.53738 -0.08811  0.26055  2.19931 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) 12.040     4.510   2.670  0.00759 ** 
wt          -4.024     1.436  -2.801  0.00509 ** 
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 43.230  on 31  degrees of freedom
Residual deviance: 19.176  on 30  degrees of freedom
AIC: 23.176

Number of Fisher Scoring iterations: 6
```

```
glmfit <- glm(am ~ wt, mtcars, family = "binomial")

> tidy(glmfit)
# A tibble: 2 × 5
  term       estimate std.error statistic p.value
  <chr>        <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  12.0      4.51      2.67  0.00759
2 wt          -4.02      1.44      -2.80  0.00509
```

## 6.3.3 Otras funciones de *tidyR*



② `augment()`

```
glmfit <- glm(am ~ wt, mtcars, family = "binomial")
```

```
> augment(glmfit)
# A tibble: 32 × 9
  .rownames      am     wt .fitted .resid .std.resid   .hat .sigma .cooksdi
  <chr>        <dbl>  <dbl>  <dbl>  <dbl>    <dbl>  <dbl>  <dbl>  <dbl>
1 Mazda RX4       1    2.62   1.50   0.635    0.680  0.126  0.803  0.0184
2 Mazda RX4 Wag   1    2.88   0.471   0.985    1.04   0.108  0.790  0.0424
3 Datsun 710      1    2.32   2.70   0.360    0.379  0.0963  0.810  0.00394
4 Hornet 4 Drive   0    3.22  -0.897  -0.827   -0.860  0.0744  0.797  0.0177
5 Hornet Sportabout 0    3.44  -1.80  -0.553   -0.572  0.0681  0.806  0.00647
6 Valiant         0    3.46  -1.88  -0.532   -0.551  0.0674  0.807  0.00590
7 Duster 360       0    3.57  -2.33  -0.432   -0.446  0.0625  0.809  0.00348
8 Merc 240D        0    3.19  -0.796  -0.863   -0.897  0.0755  0.796  0.0199
9 Merc 230          0    3.15  -0.635  -0.922   -0.960  0.0776  0.793  0.0242
10 Merc 280         0    3.44  -1.80  -0.553   -0.572  0.0681  0.806  0.00647
# ... with 22 more rows
```

③ `glance()`

```
> glance(glmfit)
# A tibble: 1 × 8
  null.deviance df.null logLik  AIC   BIC deviance df.residual nobs
            <dbl>   <int>  <dbl>  <dbl>  <dbl>    <dbl>       <int> <int>
1        43.2      31  -9.59  23.2  26.1    19.2        30     32
```

## 6.3.4 Expresiones regulares

```
grepl(pattern = "Monica" ,x =data$Nombre)  
> grepl(pattern = "Monica" ,x =data$Nombre)  
[1] FALSE FALSE TRUE FALSE FALSE FALSE
```



base::grepl()

```
str_detect(string = data$Nombre,pattern = "Monica")  
> str_detect(string = data$Nombre,pattern = "Monica")  
[1] FALSE FALSE TRUE FALSE FALSE FALSE
```

```
# A tibble: 6 × 5  
  Nombre   Apellido Profesion      Nacimiento Sexo  
  <chr>     <chr>    <chr>        <chr>       <chr>  
1 Rachel    Green    Ejecutiva    05-05-1971 F  
2 Ross      Geller   Paleontologo 18-10-1967 M  
3 Monica    Geller   Chef         22-04-1969 F  
4 Joey      Tribbiani Actor       25-05-1968 M  
5 Chandler  Bing    Analista de datos 08-04-1969 M  
6 Phoebe    Buffay  Masajista    10-10-1968 F
```



stringr::str\_detect()

# Próximo Encuentro

"Efficient R programming", de Colin Gillespie y Robin Lovelace.

## MANIPULACIÓN EFICIENTE DE DATOS

*Capítulo 6.4 - 6.7*

**Fecha:** Martes, 21 de junio

**Hora:** 18:00 a 19:00 UTC-5

**Lugar:** [ZOOM](#)

**Inscripción:** [Meetup](#)



## Conferencistas:

Denisse  
Fierro Arcos



Danisse  
Carrascal Polo



Organiza:



Semillero  
Unal

Colaboran:



R-Ladies Medellín +  
Barranquilla + Galápagos

*Club de lectura*

# GRACIAS

Organiza:



Semillero  
Unal

Colaboran:



R-Ladies Medellín +  
Barranquilla + Galápagos