

Tablas de frecuencia con la librería dplyr de R

Felipe Andres Martínez Vera

1 Crear un cuaderno nuevo

Para crear un nuevo cuaderno de Colab basado en R use el este link: <https://colab.research.google.com/#create=true&language=r>

2 Cargar el archivo que contiene los datos a Colab

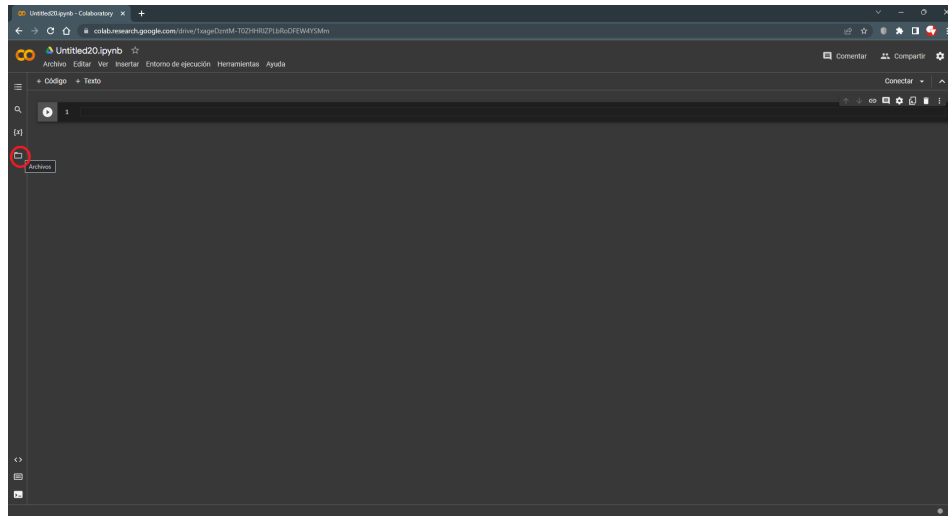


Figure 1: Cargar el archivo a Colab

3 Cargar las librerías que se van a utilizar

En este caso requerimos la librería *dplyr*.

dplyr hace parte del paquete *tidyverse*. Al cargar *tidyverse* se cargan 8 librerías dentro de las que se encuentra *dplyr*. Por practicidad se prefiere cargar *tidyverse* completa en lugar de cargar las librerías individuales.

```
library(tidyverse)
```

```
-- Attaching packages ----- tidyverse 1.3.2 --
v ggplot2 3.4.0      v purrr   1.0.1
v tibble  3.1.8      v dplyr   1.1.0
v tidyr   1.3.0      v stringr 1.5.0
v readr   2.1.3      v forcats 1.0.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
```

```
options(scipen = 9999)
```

4 Importar los datos a R

Dado que ya cargamos el paquete *tidyverse* usaremos la función `read_csv(){readr}` en lugar de `read.csv(){base}`. La función `read_csv()` es mas eficiente que su homóloga del paquete “base”.

Al usar `read_csv()`, como primer argumento se debe pasar la ruta del archivo que se desea importar.

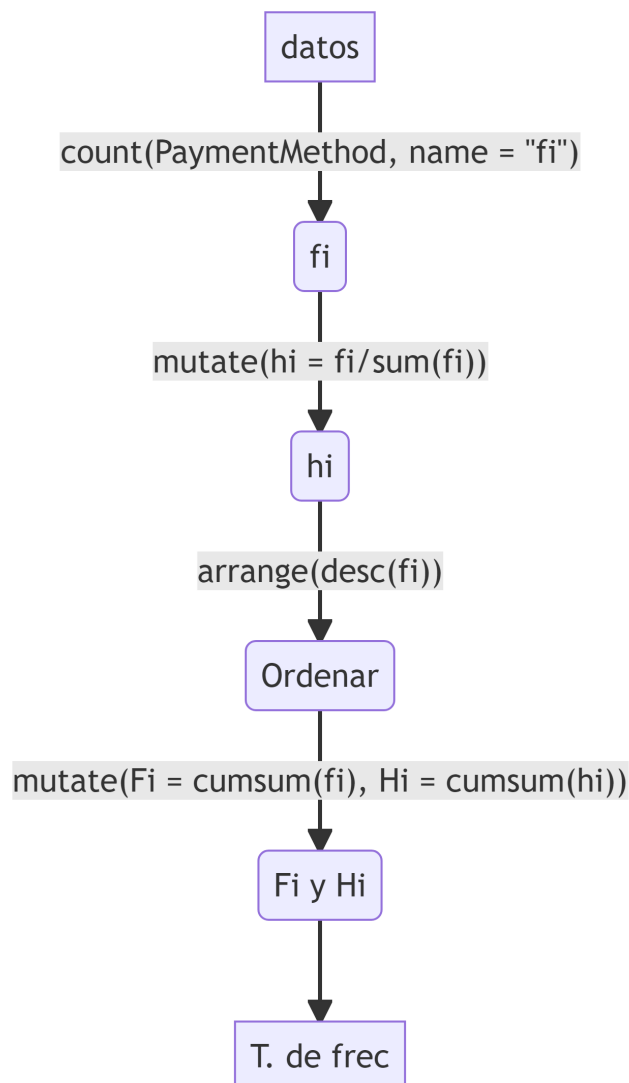
```
datos = read_csv("C:/GitHub/Estadistica-I-2023-I/Tablas_de_frecuencias/Clientes_Telcomunicacion.csv")
head(datos)
```

```
# A tibble: 6 x 21
  custom~1 gender Senio~2 Partner Depen~3 tenure Phone~4 Multi~5 Inter~6 Onlin~7
  <chr>    <chr>    <dbl> <chr>    <chr>    <dbl> <chr>    <chr>    <chr>    <chr>
1 7590-VH~ Female      0 Yes     No          1 No     No pho~ DSL     No
2 5575-GN~ Male        0 No     No         34 Yes    No     DSL     Yes
3 3668-QP~ Male        0 No     No          2 Yes    No     DSL     Yes
4 7795-CF~ Male        0 No     No         45 No     No pho~ DSL     Yes
5 9237-HQ~ Female      0 No     No          2 Yes    No     Fiber ~ No
6 9305-CD~ Female      0 No     No          8 Yes    Yes    Fiber ~ No
# ... with 11 more variables: OnlineBackup <chr>, DeviceProtection <chr>,
#   TechSupport <chr>, StreamingTV <chr>, StreamingMovies <chr>,
```

```
# Contract <chr>, PaperlessBilling <chr>, PaymentMethod <chr>,  
# MonthlyCharges <dbl>, TotalCharges <dbl>, Churn <chr>, and abbreviated  
# variable names 1: customerID, 2: SeniorCitizen, 3: Dependents,  
# 4: PhoneService, 5: MultipleLines, 6: InternetService, 7: OnlineSecurity
```

i Para tener en cuenta

A partir de este momento trabajaremos con la variable “PaymentMethod” que indica el método de pago que utiliza el cliente. Sin embargo, los procedimientos aplicados pueden ser replicados con cualquier otra variable cualitativa.



5 Obtención de las frecuencias absolutas

💡 Recuerde

Frecuencia absoluta La frecuencia de una clase (categoría o conjunto de valores) es el número de veces que la clase fue observada.

Para obtener las frecuencias de cada uno de los valores únicos de la variable usaremos la función `count()` de la librería `dplyr`.

Al usar la función `count()` debemos indicar como primer argumento la variable (columna) para la cual queremos obtener las frecuencias y en el argumento `name` podremos definir el nombre de la columna que tendrá las frecuencias absolutas. Si no se define el argumento `name` la columna con las frecuencias absolutas se llamará "n".

```
tab_freq = datos %>%  
  count(PaymentMethod, name = "fi")  
tab_freq
```

```
# A tibble: 5 x 2  
  PaymentMethod      fi  
  <chr>          <int>  
1 Bank transfer (automatic) 1558  
2 Credit card (automatic) 1537  
3 Electronic check      2387  
4 Mailed check          1621  
5 <NA>                  4
```

1. Cargar los datos importados
2. Calcular las frecuencias absolutas

6 Obtención de las frecuencias relativas

💡 Recuerde

Frecuencia relativa La frecuencia relativa de una clase (categoría o conjunto de valores) es el número de veces que la clase fue observada dividido entre el total de observaciones.

Para obtener las frecuencias relativas de cada una de las clases usaremos la función `mutate()` de la librería `dplyr`.

La función *mutate()* nos permite crear una nueva columna o modificar una columna existente en un dataframe. Al usar *mutate()* debemos indicar el nombre de la columna que se desea crear o modificar, introducir el signo =, e indicar la formula que se debe emplear para calcular el valor de esta.

```
tab_freq = datos %>%
  count(PaymentMethod, name = "fi") %>%
  mutate(hi = fi/sum(fi))
tab_freq
```

```
# A tibble: 5 x 3
  PaymentMethod      fi      hi
  <chr>          <int>   <dbl>
1 Bank transfer (automatic) 1558 0.219
2 Credit card (automatic) 1537 0.216
3 Electronic check      2387 0.336
4 Mailed check          1621 0.228
5 <NA>                   4 0.000563
```

7 Cálculo de las frecuencias acumuladas

Recuerde

Frecuencia absoluta acumulada La frecuencia acumulada de una clase (categoría o conjunto de valores) es el número total de veces que la clase considerada, junto con las clases anteriores, fueron observadas.

Frecuencia relativa acumulada La frecuencia relativa acumulada de una clase (categoría o conjunto de valores) es la suma de las frecuencias relativas anteriores e inclusive la clase en cuestión.

Para obtener las frecuencias acumuladas de cada una de las clases utilizaremos nuevamente la función *mutate()* de la librería *dplyr*.

```
tab_freq = datos %>%
  count(PaymentMethod, name = "fi") %>%
  mutate(hi = fi/sum(fi)) %>%
  mutate(Fi = cumsum(fi), Hi = cumsum(hi))
tab_freq
```

```
# A tibble: 5 x 5
```

	PaymentMethod	fi	hi	Fi	Hi
	<chr>	<int>	<dbl>	<int>	<dbl>
1	Bank transfer (automatic)	1558	0.219	1558	0.219
2	Credit card (automatic)	1537	0.216	3095	0.435
3	Electronic check	2387	0.336	5482	0.771
4	Mailed check	1621	0.228	7103	0.999
5	<NA>	4	0.000563	7107	1

i Para tener en cuenta

La acumulación de las frecuencias se ve afectada por el orden que tengan las clases en el dataframe. Por eso antes de calcular las frecuencias acumuladas es necesario verificar que las clases tienen el orden deseado.

Ahora ordenaremos las clases por su frecuencia absoluta en orden descendente (empezando con la clase de mayor frecuencia y terminando con la de menor frecuencia) y volveremos a calcular las frecuencias acumuladas.

Para ordenar el dataframe de acuerdo a los valores de la columna “fi” utilizaremos la función *arrange()* de la librería *dplyr*.

```
tab_freq = datos %>%
  count(PaymentMethod, name = "fi") %>%
  mutate(hi = fi/sum(fi)) %>%
  arrange(desc(fi)) %>%
  mutate(Fi = cumsum(fi), Hi = cumsum(hi))
tab_freq
```

A tibble: 5 x 5

	PaymentMethod	fi	hi	Fi	Hi
	<chr>	<int>	<dbl>	<int>	<dbl>
1	Electronic check	2387	0.336	2387	0.336
2	Mailed check	1621	0.228	4008	0.564
3	Bank transfer (automatic)	1558	0.219	5566	0.783
4	Credit card (automatic)	1537	0.216	7103	0.999
5	<NA>	4	0.000563	7107	1

8 ¡Para terminar!

1. ¿Que orden tenían inicialmente las clases en la tabla de frecuencias? ¿Que orden tienen las clases en la última tabla de frecuencias? ¿En cuál de los dos casos tienen más

significado las frecuencias acumuladas?

2. Interprete los siguientes valores de la última tabla de frecuencias.

- a) La frecuencia absoluta (f_i) de la clase Mailed check
- b) La frecuencia relativa (h_i) de la clase Credit card (automatic)
- c) La frecuencia acumulada (F_i) de la clase Bank transfer (automatic)
- d) La frecuencia relativa acumulada (H_i) de la clase Bank transfer (automatic).