

COMANDOS UTILES EN R

- 1) $V <- c(..., ..., ..., ...)$ → declarar un vector
- 2) $U * V$ → producto de componentes de los vectores
- 3) $U \% * \% V$ → producto escalar (es la suma de los productos de las componentes) **Se puede usar para calcular el numerador de la MEDIA de tablas de intervalos.**
- 4) $Sum(vector)$ → suma de componentes del vector
- 5) $Mean(vector)$ → promedio
- 6) $Summary(vector)$ → media, mediana, cuartiles, min, max
- 7) $quantile(vector\ de\ datos, prob = 0. q)$ → percentil q
- 8) **PROMEDIO DE TABLA DE INTERVALOS:**
X: VECTOR que contiene las MARCAS DE CLASE (xi)
f: VECTOR que contiene las FRECUENCIAS absolutas (fi)

$$promedio <- X \% * \% f / sum(f)$$

Se está calculando con ello: $promedio = \bar{x} = \frac{\sum_{i=1}^k (xi * fi)}{n}$

- 9) $IQR(vector)$: Rango intercuartílico (Cuartil 3 – Cuartil 1)
- 10) $Var(vector)$: varianza $(s^2 = \frac{\sum_{i=1}^k (xi - \bar{x})^2}{n-1})$
- 11) $sd(vector)$: desviación estándar $(s = \sqrt{varianza})$
- 12) **COEFICIENTE DE VARIACIÓN:**
 $CV <- (sd(vector) / mean(vector)) * 100$
Se está calculando: $CV = \frac{s}{\bar{x}} . 100\%$
- 13) **VARIANZA EN UNA TABLA:**
X: VECTOR que contiene las MARCAS DE CLASE (xi)
f: VECTOR que contiene las FRECUENCIAS absolutas (fi)

$$Dif <- X - promedio$$

$$VAR <- ((Dif * Dif) \% * \% f) / (sum(f) - 1)$$

NOTA:

Para hacer cálculos con los comandos 4, 5, 6, 7, 9, 10, 11 y 12, a partir de un archivo .csv en Rstudio, se debe cambiar **vector**, por **Nombre de archivo \$ Columna deseada**.

Es necesario invocar en cada proyecto:

```
library(readr)
library(plyr)
library(dplyr)
```

Luego de tener el Excel en RStudio:

```
read.csv("Utec_Alumnos.csv")
BD <- read.csv("Utec_Alumnos.csv")
```

Declarando al Excel con nombre BD, se puede leer en la derecha de RStudio, **el número de observaciones y número de variables.**

Pero también se puede usar: `dim(BD)`

También se puede usar: `nrow(BD$...)` y `ncol(BD$...)`

Se puede visualizar el tipo de variable y algunos de sus datos, mediante:

```
glimpse(BD)
```

Recordar que hay 5 tipos de datos aquí:

Numérico o Doble (*double*), entero (*integer*), complejo (*complex*), lógico (*logical*), carácter (*char*)

También se puede usar: `class(BD$...)`

Se puede seleccionar algunas columnas en una nueva variable (base de datos) usando:

```
BD1 <- select(BD, Apellido, Nombre, Ciclo, Edad)
```

Para filtrar datos, según alguna condición y guardar en nueva variable las observaciones que lo cumplen, se usa

```
BD2 <- filter(BD1, Edad > 20)
```

BD2 (para visualizar)

```
BD2 <- filter(BD1, Ciclo == 5)
```

BD2

Puede eliminarse todas las observaciones (filas) que tienen NA según alguna columna:

```
BD3 <- filter(BD1, !is.na(Ciclo))
```

BD3

```
BD31 <- filter(BD1, !is.na(Edad))
```

BD31

Para tener un conteo de los datos de una variable y ver su redacción

```
table(BD$Sexo)
```

```
> table(BD$Sexo)
```

	M	F	m	M	M	mM	NB
9	1	269	2	795	1	1	1

CAMBIO DE NOMBRE A COLUMNAS

Para cambiar de nombre a variables de un *data frame*

```
BDR <- rename(BD, periodo = PerÃ.odo, cÃ³digo = CÃ³digo)
```

BDR

```
=====
```

Modificando los datos mal escritos

Eliminar espacios en blanco de datos mal escritos:

Para eliminar los espacios en blanco utilizaremos el paquete *stringr*, el cual sirve para manipular caracteres y expresiones regulares; viene incluido en *tidyverse*.

```
library(stringr)
```

```
BDR$Sexo <- str_trim(BDR$Sexo, side = "both")
```

```
table(BDR$Sexo)
```

```
> table(BDR$Sexo)
```

	F	m	M	mM	NB
9	269	2	797	1	1

Cambiamos el tipo de letra a minúscula para homologar

todos los términos, sin que unos vengan en mayúsculas, otros en minúsculas o en formato de título.

```
BDR$Sexo <- str_to_lower(BDR$Sexo, locale = "es")
```

```
table(BDR$Sexo)
```

```
> BDR$Sexo <- str_to_lower(BDR$Sexo, locale = "es")
```

```
> table(BDR$Sexo)
```

	f	m	mm	nb
9	269	799	1	1

Palabras con acentos y sin acentos, con mismo significado, se remueven los acentos.

```
BDR$Sexo <- chartr('áéíóúñ','aeiouñ',BDR$Sexo)
```

```
table(BDR$Sexo)
```

	f	m	mm	nb
9	269	799	1	1

=====muy laborioso
Ahora unificamos las palabras con igual significado.



=====

Modificar los valores de una variable para corregir escritura:

Con transmute se crea nuevas columnas y se elimina la original

```
Sexomodif <- transmute(BDR, Sexo = case_when(Sexo == '~'no especifica',
Sexo == 'mm' ~ 'm', Sexo == 'nb' ~ 'desconocido', Sexo == 'm' ~ 'm', Sexo == 'f' ~ 'f'))
Sexomodif
table(Sexomodif)
```

```
> table(Sexomodif)
Sexomodif
desconocido      f      m no especifica
1          269    800          9
```

Creando un *data frame*. Ejemplo:

```
id <- c(1,4,3,NA,7,6,9,4,0,8)
```

```
texto <- c("a","b",NA,"NA","a","b","b","b","c","d")
```

```
valor <- c(2,8,7,5,1,9,4,3,7,2)
```

```
datos <- data.frame(id = id, texto = texto, valor = valor)
```

```
head(datos, n = 10)
```