ENES
JURIQUILLA

CURSO INTER-SEMESTRAL
INTRODUCCIÓN A R

PUBLICO: PROFESORES, INVESTIGADORES Y ESTUDIANTES
REQUISITOS: NINGUNO
DURACIÓN DEL CURSO: 20 HRS
FECHAS: 17-21 DE AGOSTO 2021

Profesores:
Dr. Alberto Prado
Dr. Ulises Olivares Pinto

MANEJO Y ANÁLISIS DE DATOS
2 HORAS

- Repaso data.frame
- Manejo de fechas
- Ordenar filas
- Transformaciones del formato
  - Transposición
- Familia de funciones **apply**
  - **apply()**
  - **sapply()**
  - **tapply()**
- Transformaciones del formato
  - Formato ancho vs formato largo: Librería reshape2
- Manejo de datos faltantes
  - na.rm = TRUE
  - complete.obs

Data frame: Es una estructura bi-dimensional de datos, dónde las filas son las observaciones y las columnas son las variables.

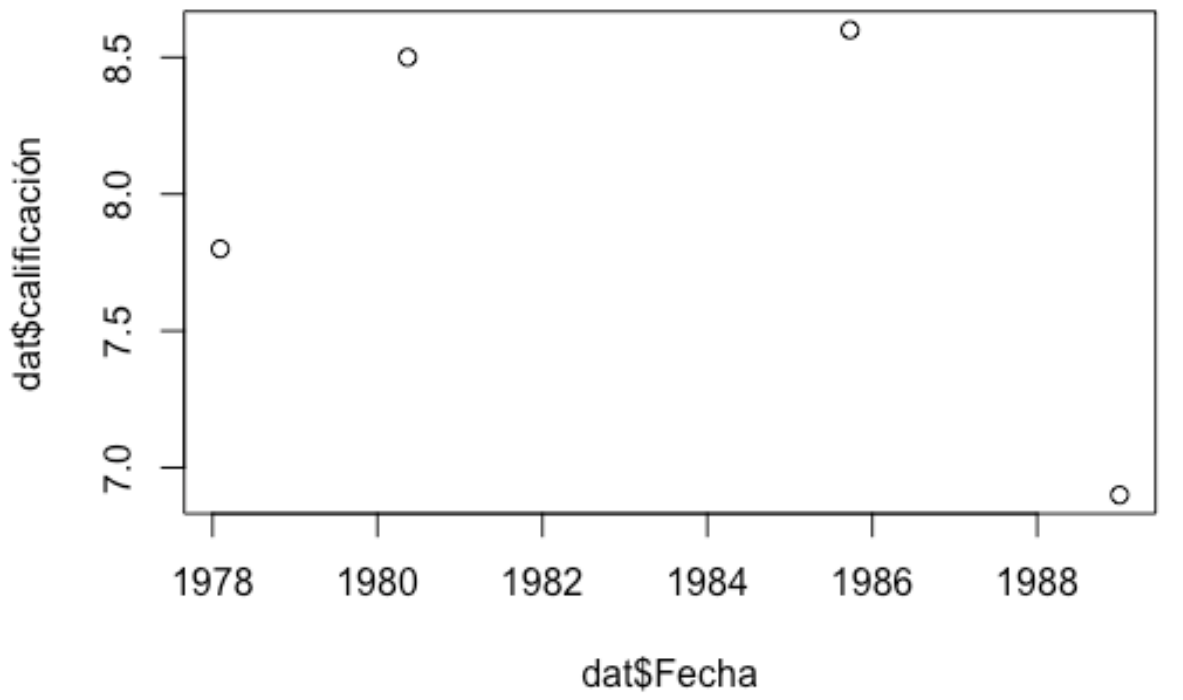
```
dat <- data.frame (nombres=( "Juan","Ines","Pablo"), calificación = c(8.6, 7.8, 6.9), Fecha= c("1985-09-24", "1978-02-04", "1988-12-31")

dat
```

Variable nominal	Variable numérica	Variable Fecha
Juan	8.6	1985-09-24
Ines	7.8	1978-02-04
Pablo	6.9	1988-12-31

Manejo de fechas

Las fechas las le R por default como YYYY-mm-dd.
La función **as.Date()** convierte las secuencias de caracteres en fechas

```
dat$Fecha <- as.Date(dat$Fecha)
plot(dat$calificación~dat$Fecha)
```



Existe la posibilidad de especificar el formato de la fecha usando el argumento **tryFormats = "%d/%m/%y"**

Ordenar filas

La función **order()** ordena un vector.

```
X <- c(3,1,2)
order(X)
1,2,3
```

Si se le especifica **order(-X)** o **order(X, decreasing =T)**

```
3,2,1
```

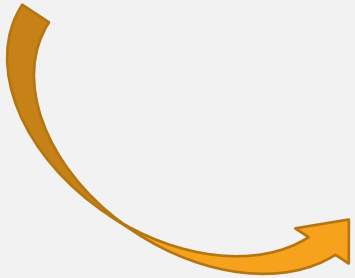
Las filas de un data.frame se pueden ordenar de la siguiente manera:

```
dat[ order(dat$nombre), ]
```

Transposición

	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Luis	1	2	3	4	5	6	7	8	9	10	11	12
Yolanda	4	8	12	16	20	24	28	32	36	40	44	48
Jacinto	16	32	48	64	80	96	112	128	144	160	176	192
Barabara	64	128	192	256	320	384	448	512	576	640	704	768
Conchita	256	512	768	1024	1280	1536	1792	2048	2304	2560	2816	3072
Miguel	1024	2048	3072	4096	5120	6144	7168	8192	9216	10240	11264	12288

t(data.frame)



	Luis	Yolanda	Jacinto	Barabara	Conchita	Miguel
Enero	1	4	16	64	256	1024
Febrero	2	8	32	128	512	2048
Marzo	3	12	48	192	768	3072
Abril	4	16	64	256	1024	4096
Mayo	5	20	80	320	1280	5120
Junio	6	24	96	384	1536	6144
Julio	7	28	112	448	1792	7168
Agosto	8	32	128	512	2048	8192
Septiembre	9	36	144	576	2304	9216
Octubre	10	40	160	640	2560	10240
Noviembre	11	44	176	704	2816	11264
Diciembre	12	48	192	768	3072	12288

Familia apply

La familia **apply** es una de las librerías básicas de R que permite manipular segmentos de los datos de matrices, data frames o listas de manera repetitiva.

Actúan sobre los datos ejecutando la función llamada (Ej de funciones: **mean(), sum()**).

apply()

apply(data.frame, MARGIN, FUN, ...)

¿A qué datos le voy a aplicar la función?

¿Por filas (1) o por coliumnas (2)?

Función a aplicar

Argumentos adicionales a pasar a la función.
Ej: ¿Qué hacer de los datos faltantes?

Ejercicio 1

- Cargar datos “DataApply.csv”
- Cambiar nombre de las filas por una amalgama de la referencia individual y el método con la función **paste()**
- Eliminar columna “RefIndiv”
- Agregar una columna del promedio de los 5 evaluadores
- Crear un nuevo data.frame con al promedio de calificación de cada evaluador

sapply & tapply

sapply()

Funciona muy parecido a apply por columnas pero puede manejar data.frames, listas o vectores.

```
sapply(concurso[,3:7], mean, na.rm=T)
```

tapply()

Crea resúmenes de los datos de acuerdo a un factor

```
tapply(concurso$Examinateur1, concurso$Animal, mean, na.rm=T)
```

Formato Ancho

	Enero	Febrero	Marzo
Luis	1	2	3
Yolanda	4	8	12
Jacinto	16	32	48

Formato Largo

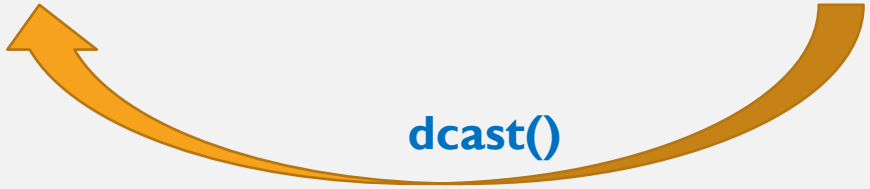
	Variables	Valor
Luis	Enero	1
Luis	Febrero	2
Luis	Marzo	3
Yolanda	Enero	4
Yolanda	Febrero	8
Yolanda	Marzo	12
Jacinto	Enero	16
Jacinto	Febrero	32
Jacinto	Marzo	48

Formato Ancho

	Enero	Febrero	Marzo
Luis	1	2	3
Yolanda	4	8	12
Jacinto	16	32	48

Formato Largo

	Variables	Valor
Luis	Enero	1
Luis	Febrero	2
Luis	Marzo	3
Yolanda	Enero	4
Yolanda	Febrero	8
Yolanda	Marzo	12
Jacinto	Enero	16
Jacinto	Febrero	32
Jacinto	Marzo	48



```
melt(data,  id.vars=c("Variable1","Variable2"))
```

Variables que se van a conservar como columnas

Ejercicio 2:

- Descargar la librería **reshape2**
- Cambiar el formato de "concurso" a formato largo usando la función **melt()**
  - No incluir la columna promedio
- Hacer una tabla con el promedio de cada examinador usando **tapply()**
- Usar la opción **list(factor1, factor2)** para obtener los promedios de cada examinador por cada tipo de animal.
- Repetir el paso anterior incorporando el método.

Ejercicio 3:

- Descargar los datos **calabazas.csv**
- Checar estructura
- Son datos de dos variedades de calabazas (variable: cult) sembradas en tres fechas diferentes
- Saca el promedio de cada vitamina C para cada una de las variedades para cada fecha de siembra

Ejercicio 4:

- Descargar los datos **airquality**
- Checar estructura
- Convertir columnas 1:4 a numérico
- Transformar a formato largo conservando el mes y el día como columnas
- Sacar el promedio de cada mes para cada una de las variables atmosféricas