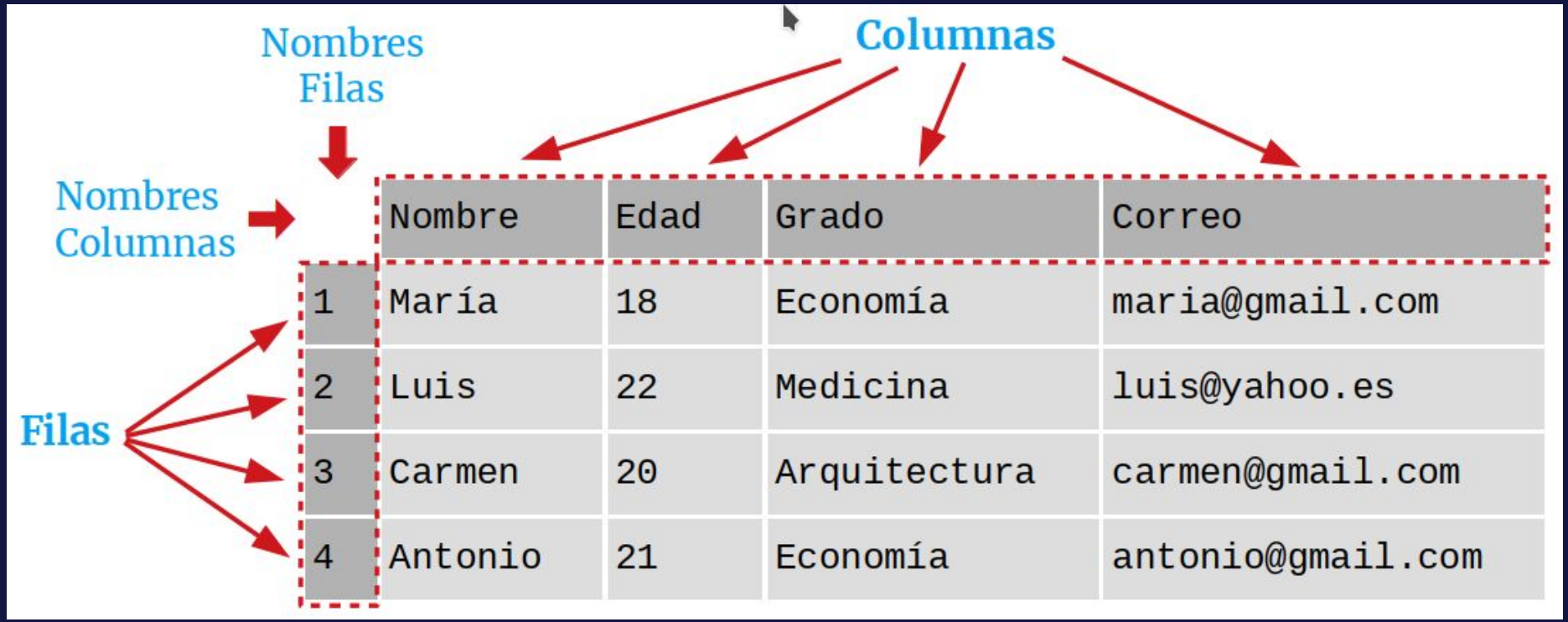
Introducción a la Ciencia de Datos con Python

¿Qué vimos en la #clase10?

- Pandas



	Nombre	Edad	Grado	Correo
1	María	18	Economía	maria@gmail.com
2	Luis	22	Medicina	luis@yahoo.es
3	Carmen	20	Arquitectura	carmen@gmail.com
4	Antonio	21	Economía	antonio@gmail.com

Valores faltantes

Son aquellos que no constan debido a cualquier acontecimiento, como por ejemplo errores en la transcripción de los datos o la ausencia de disposición a responder a ciertas cuestiones de una encuesta.

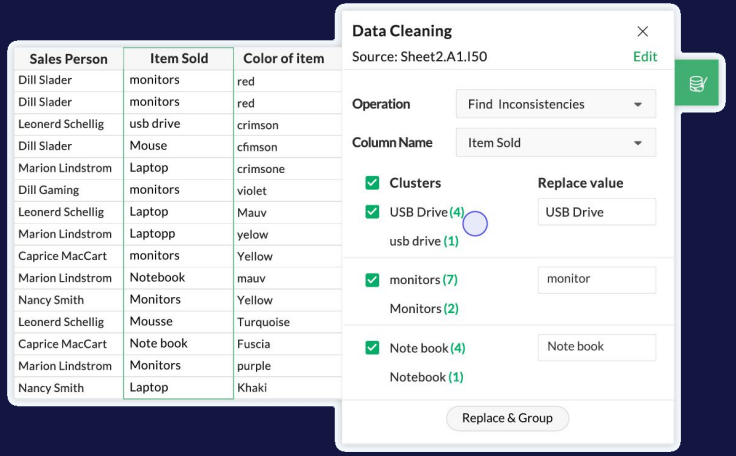
Los datos pueden faltar de manera aleatoria o no aleatoria.

Los datos faltantes aleatorios pueden perturbar el análisis de datos dado que disminuyen el tamaño de las muestras y en consecuencia la potencia de las pruebas de contraste de hipótesis. Los datos faltantes no aleatorios ocasionan, además, disminución de la representatividad de la muestra.

¿ Por qué faltan los valores?

Los valores faltantes existen por dos razones:

- Extracción de los datos
- Recolección de los datos



```
print(pd.isnull(df.body))  # chequeo los valores faltantes en la col de label body
```

```
print(pd.isnull(df.body))  # chequeo los valores faltantes en la col de label body
print()
print(pd.notnull(df.body))  # chequeo los valores no faltantes en la col de label body
```

```
print(pd.isnull(df.body).values)  # arreglo con los valores faltantes de la col de label body
```

```
print(pd.isnull(df.body).values.sum())  # cantidad de valores faltantes
```

¿Qué se hace con los valores faltantes?

- Borrar toda la fila o toda la columna donde falta el valor

```
# borro los datos
print(df.dropna(axis=0, how='all'))  # se borra una fila si TODAS las col tienen valores faltantes
print(df.dropna(axis=0, how='any'))  # se borra una fila si ALGUNA de las col tienen valores faltantes
```

- Cómputo de los valores faltantes

```
# reemplazo TODOS los valores faltantes por 0
print(df.fillna(0))

# reemplazo TODOS los valores faltantes por "desconocido"
print(df.fillna('Desconocido'))
```

Pero... recordemos que no nos interesa el df completo sino, más bien, las series que forman cada columna. Por lo tanto, la forma correcta de reemplazar valores es hacerlo dependiendo de la columna que estemos analizando.

```python
# reemplazo valores según columna

print(df['body'].fillna(0))
print(df['home.dest'].fillna('Desconocido'))
print(df['age'].fillna(df['age'].mean()))
```

IMPORTANTE: estamos imprimiendo pero NO guardando, para modificar finalmente el df hay que reasignar la variable df, y para extraer la info, hay que asignarlo a una variable.

Variables dummies

Variables ficticias para variables categóricas.

```
# Variables dummies

print(df.sex)  # variable categórica

dummy_sex = pd.get_dummies(df.sex, prefix='sex')
print(dummy_sex)

df = df.drop(['sex'], axis=1)
# print(df.sex)
df = pd.concat([df, dummy_sex], axis=1)
print(df)
```

merge()

Métodos para unir dfs

```
df1 = pd.DataFrame({'employee': ['Bob', 'Jake', 'Lisa', 'Sue'], 'group': ['Accounting', 'Engineering', 'Engineering', 'HR']})
df2 = pd.DataFrame({'employee': ['Lisa', 'Bob', 'Jake', 'Sue'], 'hire_date': [2004, 2008, 2012, 2014]})

print(df1)
print()
print(df2)
```

```
# Unimos los dos df con merge

df3 = pd.merge(df1, df2)
print(df3)
print()
```

Tablas pivote o dinámicas

Es un tipo especial de tablas en las que es posible resumir de forma dinámica el contenido del df.

```
sex = df.groupby('sex')[['survived']].mean()
print(sex)

sex = df.groupby(['sex', 'pclass'])[['survived']].aggregate('mean')  # vertical
print(sex)
sex = df.groupby(['sex', 'pclass'])[['survived']].aggregate('mean').unstack()  # horizontal: correcto
print(sex)

# Pivot table

sex = df.pivot_table('survived', index='sex', columns='pclass')
print(sex)
```

```
age = pd.cut(df['age'], [0, 18, 80])
sex = df.pivot_table('survived', ['sex', age], 'pclass')
print(sex)
```

```
sex = df.pivot_table(index='sex', columns='pclass', aggfunc={'survived': sum, 'fare': 'mean'})
print(sex)
```

```
sex = df.pivot_table('survived', index='sex', columns='pclass', margins=True)
print(sex)
```

Métodos similares a los de Python

len()	lower()	translate()	islower()
ljust()	upper()	startswith()	isupper()
rjust()	find()	endswith()	isnumeric()
center()	rfind()	isalnum()	isdecimal()
zfill()	index()	isalpha()	split()
strip()	rindex()	isdigit()	rsplit()
rstrip()	capitalize()	isspace()	partition()
lstrip()	swapcase()	istitle()	rpartition()