

# SAMPLING

Para hacer simple sistema

```
size = 5  
interval = h(df)//size  
df.iloc[:, interval]
```

→ No one so may still

Per fare un shuffle del dataset  $\rightarrow$  `shuffle = df.sample(frac=1)`

$$shuff = df.sample(frac=1)$$

Para mostrar la proporción de votos de cada categoría  $\rightarrow$

```
df_shat = df_graphy('rd').sample(...)
```

Si on veut de mieux la proposer, qu'envisage les mises par code coloré.

Lo mismo pero usando n

Si queremos hacerlo por pesos  $\rightarrow$

5. creates new column of pairs  
df.sample1....., weights='weights')

Clustering sampling (No uso todas las categorías solo algunas random)  
nº de categorías

scap = random.sample(df[0:13], k=N)  $\rightarrow$  categorical depends

$$df = df [df['relation(sup)']]$$

df-2.gapsy(...).sample(n=...)

→ Datos aleatorios de esas categorías

## Error relativo

$$\text{rel. err} = 100 * \text{abs}(\text{media de muestra} - \text{media total}) / \text{media total}$$

Per aplicarea districului este

→ Tab de los datos →

```
std::cbcf = 0;
```

→ Si es un master -

$$\boxed{\text{std}(\text{ddof} = 1)}$$

## Bootstrapping

Usamos los datos como una muestra y los aprovechamos para crear una población.

Per helios

```
dt.sample (frac = 1, replace = True)
```

## Intervento de confiança

$$(0.025 - 0.975)$$

Prede necesare con quantile

Other forms  $\rightarrow$

```
from scipy.stats import norm
lower = norm.ppf(0.025, loc = point_estimate, scale = std_err)
upper = " " (0.975, - - - - -)
```