

## PED - Tema 3: Eficiencia en la programación y gestión de datos

Contenido:

- Generación optimizada de datos para su procesamiento
- Vectorización
- Gestión del almacenamiento

### Generación optimizada de datos para su procesamiento

Bibliografía:

- Básica
  - Capítulo 3, apartado 3.1, sección List, Set, and Dict Comprehensions; apartado 3.2, sección Generators (estos contenidos se han tratado en el tema 1, pero para este tema debemos centrarnos en lo relativo a optimización en cuanto a tiempo de ejecución y uso de memoria) en Python for Data Analysis (2nd Edition)
  - Capítulo 3, apartado 3.1, sección List, Set, and Dict Comprehensions; apartado 3.2, sección Generators en Python for Data Analysis (3rd Edition)
  - Notebook con ejemplos suministrado por el Equipo Docente

### Vectorización

Bibliografía:

- Básica
  - Capítulo 4, apartado 4.1, sección Arithmetic with NumPy Array en Python for Data Analysis (2nd Edition)
  - Capítulo 4, apartado 4.1, sección Arithmetic with NumPy Array en Python for Data Analysis (3rd Edition)
  - Notebook con ejemplos suministrado por el Equipo Docente
  - Capítulo 4, apartados 4.2 y 4.5, donde se aprovecha la vectorización para realizar operaciones sobre arrays Data Analysis (2nd Edition)
  - Capítulo 4, apartados 4.3 y 4.6, donde se aprovecha la vectorización para realizar operaciones sobre arrays Data Analysis (3rd Edition)

### Gestión del almacenamiento

Para consultar la memoria

```
df.info(memory_usage="deep")
```

Para obtener el total de memoria ocupada por un DataFrame:

```
mem = df.memory_usage(deep=True).sum()
```

Bibliografía:

- Notebook `datosAImportar.csv` facilitado en la asignatura
- Notebook `datosAImportarConFechas.csv` suministrado en la asignatura