.between() method
Dealling with Duplicates

.between()

.between()

.between() is a **method**

- It is used on a pandas Series.
- It checks if the values in the Series fall within a given range.
- It is inclusive
- Returns a boolean Series.

```
df["age"].between(18, 30)
```

- The Boolean Series can be used to filter rows

```
df[df["date"].between("2023-02-01", "2023-03-31")]
```

- It can be used with numbers, dates and strings.
- It doesn't support non-orderable types like objects or booleans

Data Filtering

Dealing With Duplicate Values

What is .duplicated()?

A pandas method used to identify duplicate rows in a DataFrame or Series.

Returns a boolean Series where:

True indicates a duplicate row.

False indicates a unique row.

Dealing With Duplicate Values

Key Parameters:

subset: Specify columns to consider for duplication. (Default: All columns)

keep: Controls which duplicate is marked as False:

'first' (default): Marks duplicates except the first occurrence.

'last': Marks duplicates except the last occurrence.

False: Marks all duplicates as True.

Dealing With Duplicate Values

Creating a Boolean Series:

```
df["first_name"].duplicated()
```

```
619    False
872     True
74     False
535    False
898    False
       ...
550    False
540    False
675    False
476    False
783     True
Name: first_name, Length: 1000, dtype: bool
```

Dealing With Duplicate Values

Filtering with duplicated() to see the duplicated rows

```
df[df.duplicated()]
```

	first_name	last_name	salary	start_date	gender	remote	team
1	Haleigh	Calderhead	334473	2020-05-09	NaN	True	management
2	Haleigh	Calderhead	334473	2020-05-09	NaN	True	management
3	Jaime	Gianneschi	253523	2020-09-02	Bigender	False	marketing

Dealing With Duplicate Values

What is .drop_duplicates()?

- A **pandas method** used to remove duplicate rows in a DataFrame or Series.
- Creates a new DataFrame by default or modifies the original if inplace=True.

Key Parameters:

- **subset**: Specify columns to consider for duplication. (Default: All columns)
- **keep**: Controls which duplicate to keep:
  - '**first**' (default): Keeps the first occurrence.
  - '**last**': Keeps the last occurrence.
  - **False**: Removes all duplicates.
- **inplace**: If True, modifies the DataFrame directly.

Dealing With Duplicate Values

Feature	.duplicated()	.drop_duplicates()
Purpose	Identify duplicates	Remove duplicates
Returns	Boolean Series	DataFrame or Series
Modifies Original?	No	Yes (if inplace=True)
Custom Subset?	Yes (via subset parameter)	Yes (via subset parameter)
Keep Which Rows?	Configurable via keep	Configurable via keep

Workflow Tip:

1. Use .duplicated() to count duplicates: df.duplicated().sum()
2. Use .drop_duplicates() to clean up duplicates if needed: df.drop_duplicates()