A guide to comparing map, applymap and apply

Comparing map, applymap and apply

The major differences between map, applymap and apply are:

Definition of the function:

- map is defined on Series **only**
- applymap is defined on DataFrames **only**
- apply is defined on **both**

Input arguments:

- map accepts dict, Series, or **callable**
- applymap and apply accept **callable** only

Behaviour:

- map works elementwise for **Series**
- applymap works elementwise for **DataFrames**
- apply also works elementwise but is suited to more **complex** operations and aggregation. The behaviour and return value depends on the function.

Use case (the most important difference):

- map is meant for mapping values from one domain to another, so is optimised for performance, e.g.,

```
df['A'].map({1:'a', 2:'b', 3:'c'})
```

- applymap is good for elementwise transformations across multiple rows/columns, e.g.,

```
df[['A', 'B', 'C']].applymap(str.strip)
```

- apply is for applying any function that cannot be vectorised, e.g.,

```
df['sentences'].apply(nltk.sent_tokenize)
```

NB: apply is generally slow.

Summarising

Criteria	map	applymap	apply
Defined on Series?	Yes	No	Yes
Defined on DataFrame?	No	Yes	Yes
Argument	dict, Series, or callable[1]	callable[2]	Callable
Elementwise?	Yes	Yes	Yes
Aggregation?	No	Yes	N/A
Use Case	Transformation/mapping[3]	Transformation	More complex functions
Returns	series	DataFrame	scalar, Series, or DataFrame[4]

Did you know:

1.map when passed a dictionary/Series will map elements based on the keys in that dictionary/Series. Missing values will be recorded as NaN in the output.

2.applymap in more recent versions has been optimised for some operations. You will find applymap slightly faster than apply in some cases. One suggestion is to test them both and use whatever works better.

3.map is optimised for elementwise mappings and transformation. Operations that involve dictionaries or Series will enable pandas to use faster code paths for better performance.

4.Series.apply returns a scalar for aggregating operations, Series otherwise. Similarly for DataFrame.apply.