In-Place APIs Misused

Context

Data structures can be manipulated in mainly two different approaches:
1) by applying the changes to a copy of the data structure and leaving the original object intact.
2) by changing the existing data structure (also known as in-place).

Problem

Some methods can adopt in-place by default, while others return a copy. If the developer assumes an in-place approach, he will not assign the returned value to any variable. Hence, the operation will be executed, but it will not affect the final outcome.

For example, when using the Pandas library, the developer may not assign the result of `df.dropna()` to a variable. He may assume that this API will make changes on the original `DataFrame` and not set the in-place parameter to be `True` either. The original `DataFrame` will not be updated in this way.

Solution

We suggest developers to check for each Pandas or Numpy API call, **if the result is really collected in a variable.**

Existing Stage	**Effect**
Data Cleaning	Error-prone

Examples at next page.

Example

```
Python


### Pandas
import pandas as pd
df = pd.DataFrame([-1])
- df.abs()
+ df = df.abs()
### NumPy
import numpy as np
zhats = [2, 3, 1, 0]
- np.clip(zhats, -1, 1)

+ zhats = np.clip(zhats, -1, 1)
```

Examples of Pandas API that do not In Place operations:

DataFrame and Series Methods:
- *df.drop()*
- *df.rename()*
- *df.sort_values()*
- *df.fillna()*
- *df.replace()*
- *df.drop_duplicates()*
- *df.reset_index()*
- *df.set_index()*

Aggregation and Transformation:
- *df.groupby()*
- *df.agg()*
- *df.transform()*
- *df.abs()*

Merging and Joining:
- *pd.merge()*
- *df.join()*
- *df.concat()*

Reshaping:
- *df.pivot()*
- *df.melt()*
- *df.stack()*
- *df.unstack()*

Examples of Numpy API that do not in place operations:

Array Creation and Manipulation:
- *np.copy()*

- *np.reshape()*
- *np.transpose()*
- *np.flatten()*
- *np.ravel()*
- *np.concatenate()*
- *np.stack()*
- *np.split()*
- *np.clip()*

Mathematical Operations:

- *np.add()*
- *np.subtract()*
- *np.multiply()*
- *np.divide()*
- *np.power()*
- *np.sqrt()*
- *np.exp()*
- *np.log()*

Statistical Functions:

- *np.mean()*
- *np.median()*
- *np.std()*
- *np.var()*
- *np.sum()*
- *np.prod()*
- *np.min()*
- *np.max()*

Trigonometric Functions:

- *np.sin()*
- *np.cos()*
- *np.tan()*
- *np.arcsin()*
- *np.arccos()*
- *np.arctan()*