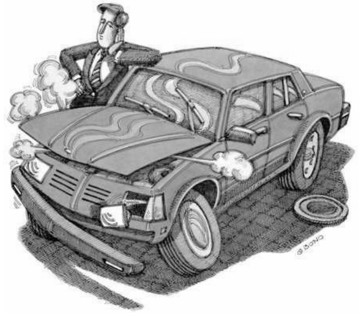
# Good Programming Practices
# Defensive Programming

Analytics Vidhya

# Defensive Programming

Car Requires Maintenance

- Check individual parts for Malfunction
  - Testing and debugging



- Take Preventive Measures
  - Defensive Programming

## Defensive Programming

# Defensive Programming

1. Preventive measures to avoid or easily detect bugs in future

# Defensive Programming

1. Preventive measures to avoid or easily detect bugs in future

2. What to do when code encounters some unexpected condition

## Defensive Programming

1. Preventive measures to avoid or easily detect bugs in future
2. What to do when code encounters some unexpected condition

- Document constraints and specifications
- Document assumptions behind code design
- Identify boundary conditions for the code

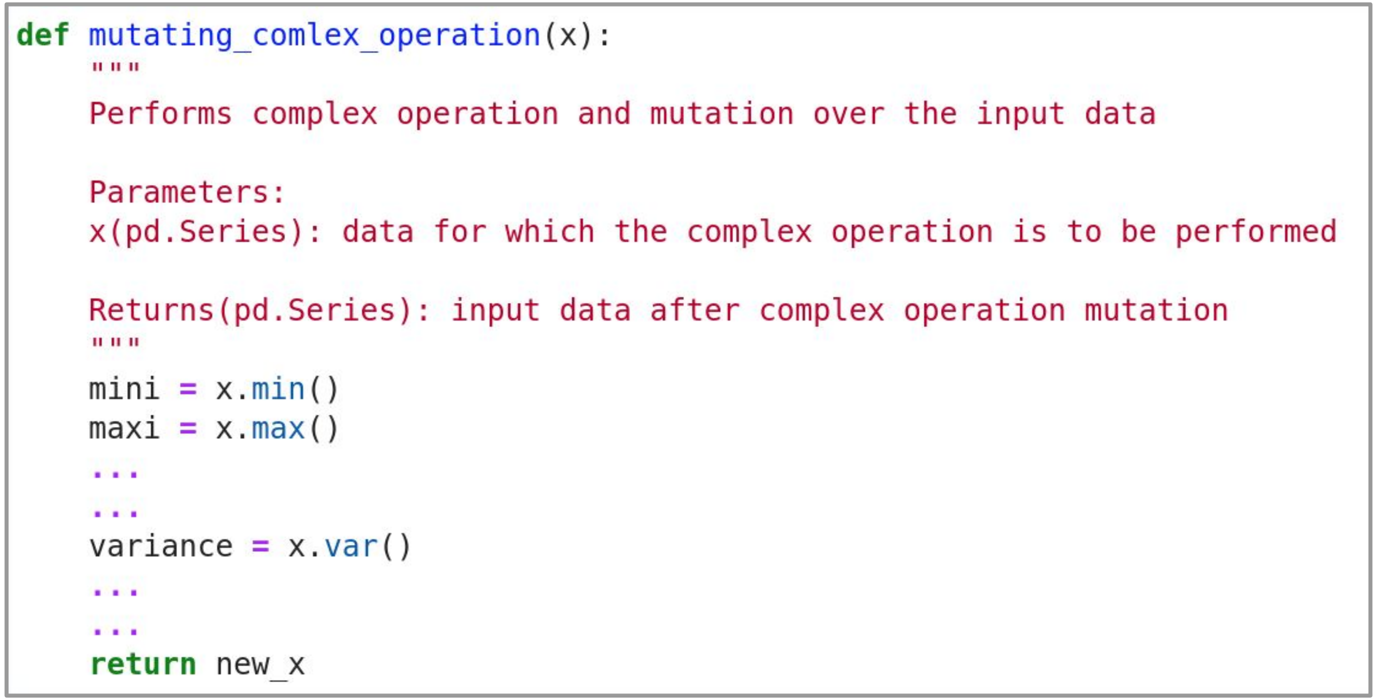
# Defensive Programming

1. Preventive measures to avoid or easily detect bugs in future
2. What to do when code encounters some unexpected condition

- Document constraints and specifications
- Document assumptions behind code design
- Identify boundary conditions for the code

Enforced during Implementation

## Defensive Programming



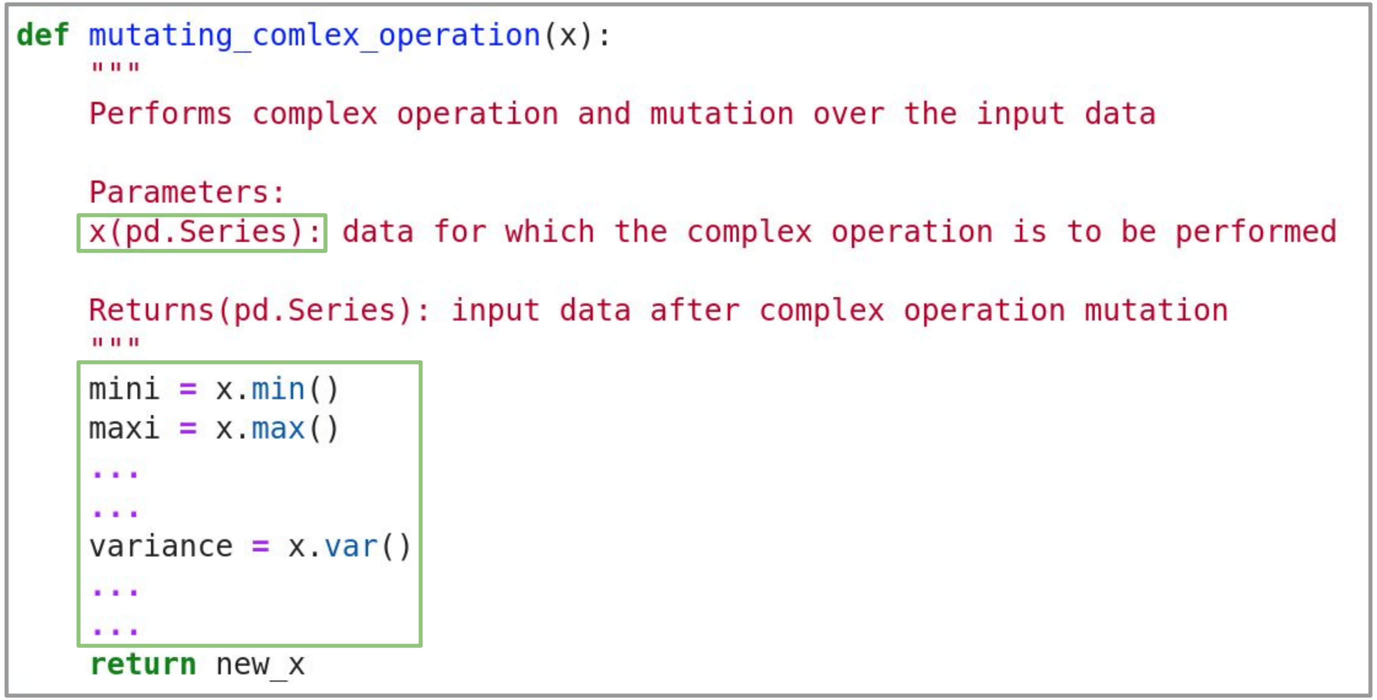
```python
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

# Defensive Programming

```
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

Analytics Vidhya

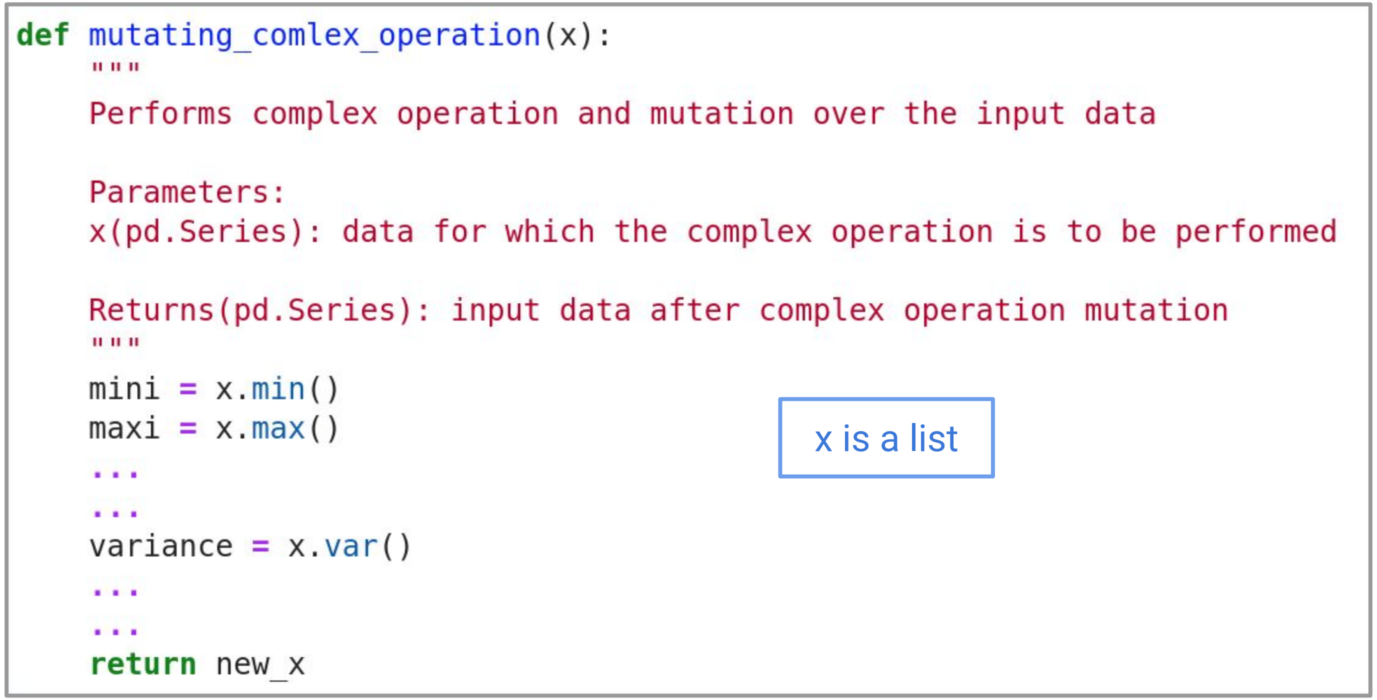
# Defensive Programming



```python
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

Analytics Vidhya

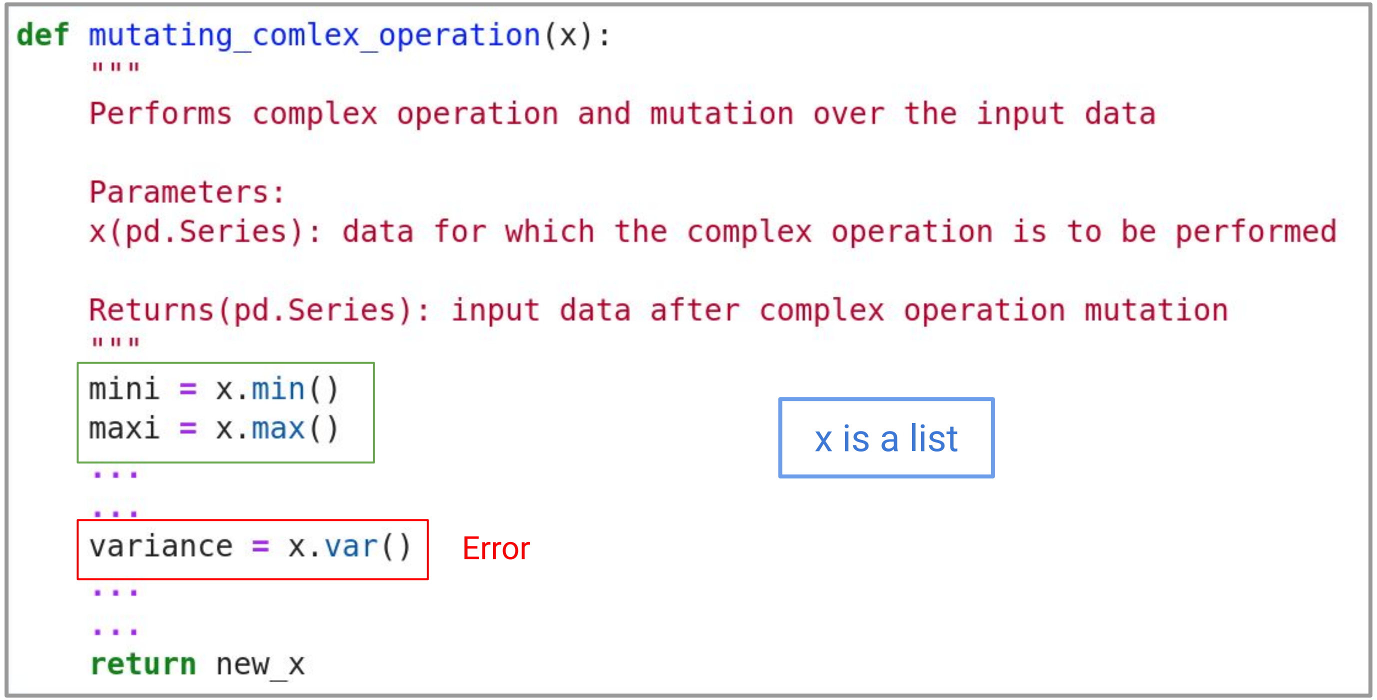
# Defensive Programming



```python
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

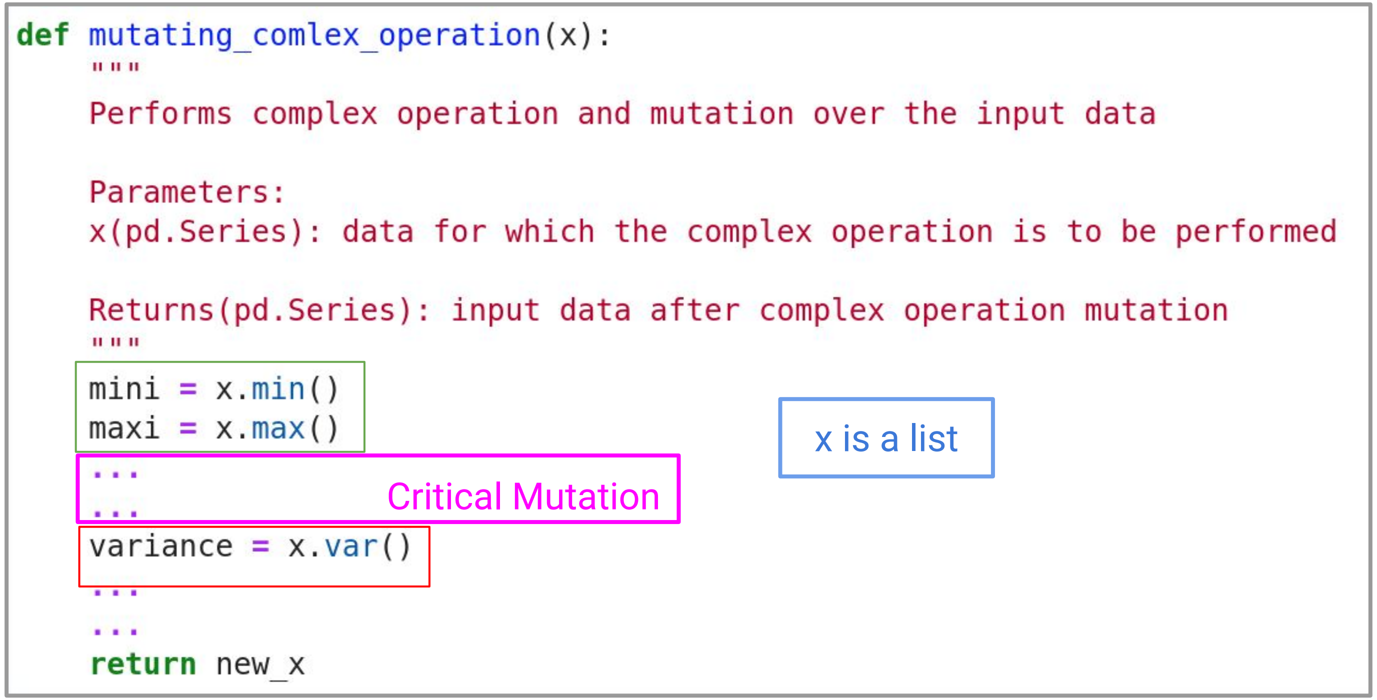
x is a list

# Defensive Programming



```
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

# Defensive Programming

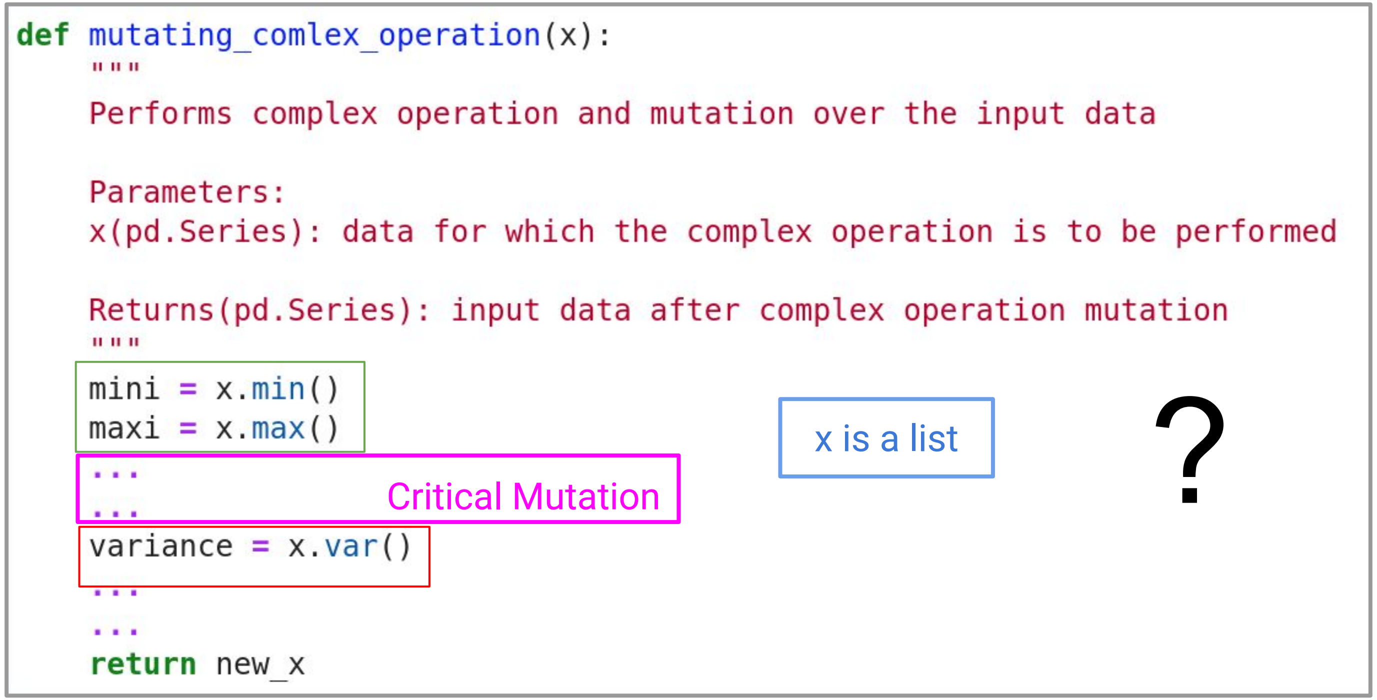


```
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

Critical Mutation

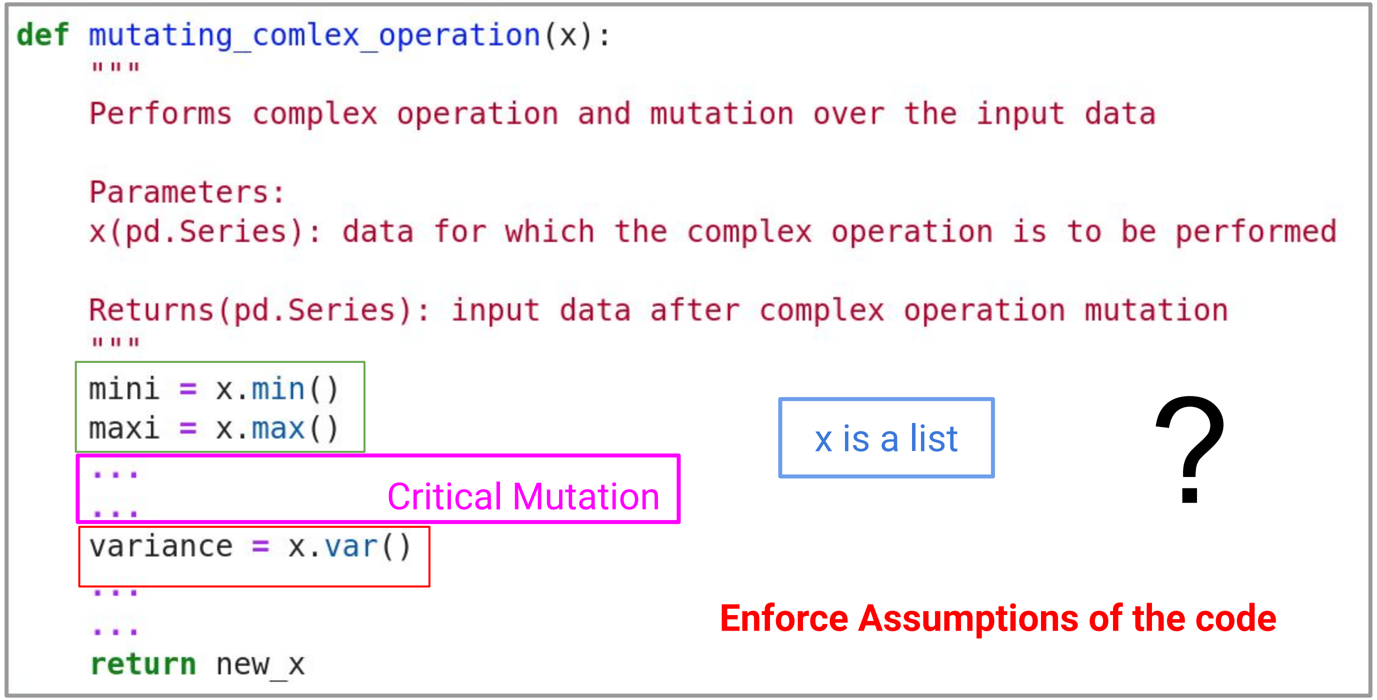
x is a list

# Defensive Programming



```python
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

# Defensive Programming



```python
def mutating_comlex_operation(x):
    """
    Performs complex operation and mutation over the input data

    Parameters:
    x(pd.Series): data for which the complex operation is to be performed

    Returns(pd.Series): input data after complex operation mutation
    """
    mini = x.min()
    maxi = x.max()
    ...
    ...
    variance = x.var()
    ...
    ...
    return new_x
```

Critical Mutation

x is a list

?

**Enforce Assumptions of the code**

Thank You

Analytics Vidhya