# Pandas Handling Wrong Data

Sometimes, a dataset can have inaccurate entries due to reasons such as human errors during data input, sourcing data from unreliable places, etc.

This can significantly undermine the quality and reliability of the data analysis performed on it.

Let's take a DataFrame containing data about students of an all-boys elementary school.

```python Name Age Gender Standard 0 John 8 M 3 1 Michael 9 M 4 2 Tom 7 M 12 3 Alex 80 F 3 4 Ryan 100 M 5

In the above DataFrame, we can see a few obvious mistakes like:

- The ages of two students are listed as **80** and **100**, which is too old for primary school students.
- The gender of **Alex** is marked as **F**. Since this is an all-boys school, it is obviously a mistake.
- **Tom** is listed as being in the 12th standard, which is simply not possible in an elementary school context. We can handle such wrong data in the following ways:

1. Replace Individual Values
2. Replace Values Based on a Condition
3. Remove Wrong Values

## Replace Individual Values

We can see that the value **F** for `Gender` column is an obvious mistake. Let's replace **F** with **M** to rectify the error

```python
import pandas as pd

# create dataframe
data = {
    'Name': ['John', 'Michael', 'Tom', 'Alex', 'Ryan'],
    'Age': [8, 9, 7, 80, 100],
    'Gender': ['M', 'M', 'M', 'F', 'M'],
    'Standard': [3, 4, 12, 3, 5]
}
df = pd.DataFrame(data)

# replace F with M
df.loc[3, 'Gender'] = 'M'

print(df)
```

```
      Name   Age Gender  Standard
0     John    8      M          3
```
```

```
1  Michael    9      M          4
2      Tom    7      M         12
3     Alex   80      M          3
4     Ryan  100      M          5
```

In this example, we replaced the wrong value with the right value using `df.loc[]`.

While this method is effective for small datasets, it becomes tedious as the size of the dataset grows.

## Replace Values Based on a Condition

In cases where the values need to satisfy a particular condition, we can iterate through the values to check if the condition is satisfied and to make changes accordingly.

For example, the maximum age of students in this elementary school is **14**. But there are two students whose ages are **80** and **100**. This looks like a typo where an additional zero has been added unintentionally.

Let's fix the error.

```
import pandas as pd

# create dataframe
data = {
    'Name': ['John', 'Michael', 'Tom', 'Alex', 'Ryan'],
    'Age': [8, 9, 7, 80, 100],
    'Gender': ['M', 'M', 'M', 'M', 'M'],
    'Standard': [3, 4, 12, 3, 5]
}
df = pd.DataFrame(data)

# replace values based on conditions
for i  in df.index:
    age_val = df.loc[i, 'Age']
    if (age_val > 14) and (age_val%10 == 0):
        df.loc[i, 'Age'] = age_val/10

print(df)
```

```
      Name  Age Gender  Standard
0     John    8      M         3
1  Michael    9      M         4
2      Tom    7      M        12
3     Alex    8      M         3
4     Ryan   10      M         5
```

In the above example, we replaced ages greater than **14** which are a multiple of **10** by removing a zero from the original values.

We checked if the age is a multiple of **10** because it is highly likely that a higher age which is a multiple of **10** is a typo.

## Remove Wrong Values

Some values can't be corrected. For example, the value **12** in `Standard` doesn't make any sense as this is an elementary school. But we can't also correct it because we don't know the right standard.

So we need to remove this whole row and any other row with such values.

```
import pandas as pd

# create dataframe
data = {
    'Name': ['John', 'Michael', 'Tom', 'Alex', 'Ryan'],
    'Age': [8, 9, 7, 8, 10],
    'Gender': ['M', 'M', 'M', 'M', 'M'],
    'Standard': [3, 4, 12, 3, 5]
}
df = pd.DataFrame(data)

# remove mistaken values
for i in df.index:
    if df.loc[i,'Standard'] > 8:
        df.drop(i, inplace=True)

print(df)
```

```
      Name  Age Gender  Standard
0     John    8      M         3
1  Michael    9      M         4
3     Alex    8      M         3
4     Ryan   10      M         5
```

In this example, we removed the row(s) containing values greater than 8 in the `Standard` column.