- -> notebooks from this lecture: https://github.com/ine-rmotr-curriculum/data-cleaning-rmotr-freecodecamp
- -> data cleaning
- -> manipulating data with pandas <- previous
- -> now we are fixing the data
- ***-> data cleaning***
    - ***-> finding missing data <- the first step***
        - when something is missing from the dataset -> e.g there is a car without a price
        - we can drop these records, or fill the values with the average values of the sales data
        - if the value is important, we might need to find its actual value -> for example calling the vendor for the data, the company that soled it
    - ***-> when there are invalid values***
        - -> if there is a string in the column
        - -> increasing the complexity
        - -> if we have values which are ridiculous (for example an age of a customer being 170 in the dataset)
            - -> these are values which are unrealistic
            - -> but for example an age in the dataset still being a number
            - -> sometimes you can't always judge if the value is valid or not
            - -> the domain of the value -> everything being valid or not
    - ***-> functions with pandas***
        - -> missing values
        - -> this is related to the way numpy works
        - -> NaN <- for a missing / null value
            - -> none type
        - -> is null
        - -> is na
        - -> is null and is na
        - -> null and na are the same in pandas
        - -> is null is favoured
        - -> not na is the opposite of null
        - -> not na of 3 is true, for example
        - <u>-> 'truthy' <- something which is a true statement</u>
        - -> these work with entire series / values
        - -> which values in the series are null or not null
        - -> we can also calculate the sum of all the null values and all the not null values
        - -> we can get a result which is the summary of all of the not null values
        - -> to get the summary of all of the not null values
        - -> booleans are integers in Python
        - <u>-> every true value counts as 1 and every false value counts as a 0</u>
            - -> this is for a series
            - -> we are asking fo for the amount of null values we have
            - -> this can be used to filter the values with a series
            - -> both dataframes are for series
            - -> both funcitons also work as methods
                - -> <u>s.isnull</u>
            - -> drop na is another example of this
            - -> we are missing /e xcluding all of the missing values in the dataframe
            - <u>-> all the methods are immutable <- we aren't actually changing the original series</u>
            - -> there is a new series which is returned
- ***-> question***

What will the following code print out?

```
import pandas as pd
import numpy as np

s = pd.Series(['a', 3, np.nan, 1, np.nan])

print(s.notnull().sum())
```

```
3 <- This one
```

```
0     True
1     True
2    False
3     True
4    False
dtype: bool
```

```
0    False
1    False
2     True
3    False
4     True
dtype: bool
```