DATA CLEANING-1

In [ ]:
```
import numpy as np
import pandas as pd
```

Missing Data

What does "missing data" mean? What is a missing value? It depends on the origin of the data and the context it was generated. For example, for a survey, a Salary field with an empty value, or a number 0, or an invalid value (a string for example) can be considered "missing data". These concepts are related to the values that Python will consider "Falsy"

In [ ]:
```
falsy_values=(0, False, None, '', [], {})
```

In [ ]:
```
any(falsy_values)
```

Out[ ]:
```
False
```

In [ ]:
```
np.nan
```

Out[ ]:
```
nan
```

Be careful with np.nan-> it is virus like, anything that it touches becomes a NaN value.

In [ ]:
```
3+np.nan
```

Out[ ]:
```
nan
```

In [ ]:
```
a = np.array([1, 2, 3, np.nan, np.nan, 4])
```

In [ ]:
```
print(a.mean(),'\n',a.sum())
```

```
nan
 nan
```

In [ ]:
```
3 + None
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\Anushtup\AppData\Local\Temp\ipykernel_95976\3045745293.py in <module>
----> 1 3 + None

TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'
```

this gave an error because their types are different!!

In [ ]:
```
a = np.array([1, 2, 3, np.nan, np.nan, 4], dtype='float')
a
```

Out[ ]:
```
array([ 1.,  2.,  3., nan, nan,  4.])
```

In [ ]:
```
print(a.mean(),'\n', a.sum())
```

```
nan
 nan
```

numpy also has an infinity function which serves as a virus (just like np.nan)

In [ ]:
```
np.inf  #INFINITE!
```

Out[ ]:
```
inf
```

Checking for inf or nan

np.isnan() and np.isinf() and the joint operation is performed using np.isfinite()

In [ ]:
```
np.isnan(np.array([1, 2, 3, np.nan, np.nan, 4])
)
```

Out[ ]:
```
array([False, False, False,  True,  True, False])
```

In [ ]:
```
np.isinf(np.array([1, 2, 3, np.nan, np.nan, 4]))
```

Out[ ]:
```
array([False, False, False, False, False, False])
```

In [ ]:
```
np.isfinite(np.array([1, 2, 3, np.nan, np.nan, 4]))
```

Out[ ]:
```
array([ True,  True,  True, False, False,  True])
```

Filtering them out:

a[~np.isnan()]

In [ ]:
```
a[~np.isnan(a)]
```

Out[ ]:
```
array([1., 2., 3., 4.])
```