# Data Cleaning Process in Pandas

**Before Cleaning Data, We Need to Understand the Data and Their Type and Try Identify the Issue/Assessment Manually and Programmatic**

**There Are 2 Type of Assessments**

- Manual - Looking Through the Data Manually in Excel or Google Sheet.
- Programmatic - By Using Pandas Function Such as Info (), Describe () Or Sample ()

**Manual:**

- **Review Your Dataset Carefully and Note Down Problems You Have Identify**
- **Check Number of Row and Columns.**
- **Any Columns Has Any Missing Values.**
- **Misspelling In String Columns**
- **Check Accuracy of Data Like Values in All Column Are Accurate**
- **Valid Data or Not (**New York -> Ny**)**
- **Any Negative Values in Any Columns Like (Weight, Height)**
- **Column Not Contains Multiple Value Which Is Inaccurate Format**
- **Check Missing Data, Etc.**

**Automatic/Programmatic Assessment**

- Head & Tail
- Sample
- Describe
- Info
- Isnull
- Duplicated
- Data type

**There Are 2 Step Involving in Assessment:**

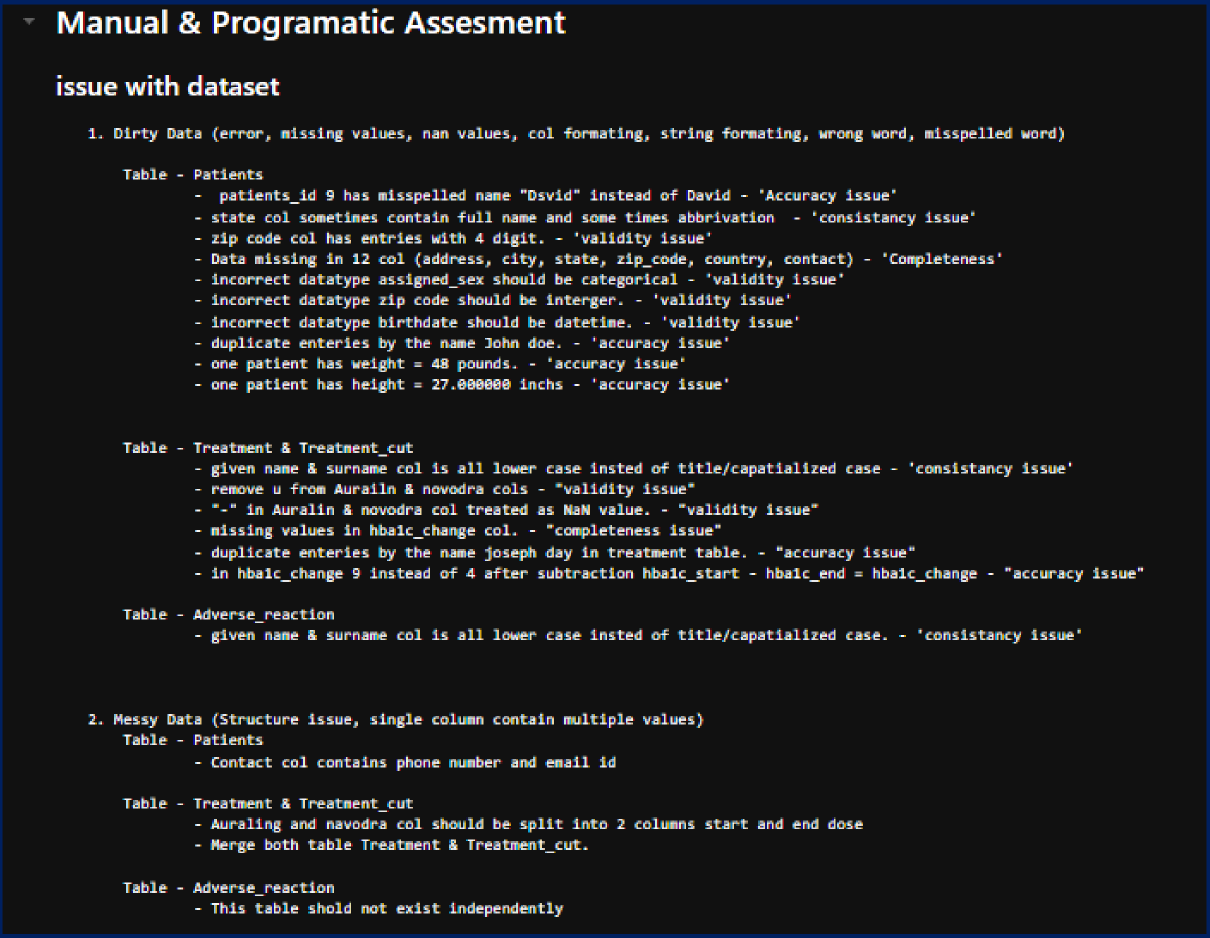
- **Discover** - find problem
- **Document** - write problem to remember it while cleaning data

**Once you discover the problem and create documentation now we need to labialised the problem to process the data cleaning as per the created path to resolve the issue in the data set for analysis. This process will help to clean data fast and safely.**

**There are 4 option which is help to sort your problem to clean your data to improve the data quality and arrange your data in Dirty Data or Messy Data.**

1. **Dirty Data**
   - (Error, Missing Values, Nan Values, Col Formatting, String Formatting, Wrong Word, Misspelled Word)
2. **Messy Data**
   - (Structure Issue, Single Column Contain Multiple Values)

# My manual Assessment and labelling



## Manual & Programatic Assesment

### issue with dataset

```
1. Dirty Data (error, missing values, nan values, col formating, string formating, wrong word, misspelled word)

    Table - Patients
            -  patients_id 9 has misspelled name "Dsvid" instead of David - 'Accuracy issue'
            - state col sometimes contain full name and some times abbrivation  - 'consistancy issue'
            - zip code col has entries with 4 digit. - 'validity issue'
            - Data missing in 12 col (address, city, state, zip_code, country, contact) - 'Completeness'
            - incorrect datatype assigned_sex should be categorical - 'validity issue'
            - incorrect datatype zip code should be interger. - 'validity issue'
            - incorrect datatype birthdate should be datetime. - 'validity issue'
            - duplicate enteries by the name John doe. - 'accuracy issue'
            - one patient has weight = 48 pounds. - 'accuracy issue'
            - one patient has height = 27.000000 inchs - 'accuracy issue'


    Table - Treatment & Treatment_cut
            - given name & surname col is all lower case insted of title/capatialized case - 'consistancy issue'
            - remove u from Aurailn & novodra cols - "validity issue"
            - "-" in Auralin & novodra col treated as NaN value. - "validity issue"
            - missing values in hba1c_change col. - "completeness issue"
            - duplicate enteries by the name joseph day in treatment table. - "accuracy issue"
            - in hba1c_change 9 instead of 4 after subtraction hba1c_start - hba1c_end = hba1c_change - "accuracy issue"

    Table - Adverse_reaction
            - given name & surname col is all lower case insted of title/capatialized case. - 'consistancy issue'



2. Messy Data (Structure issue, single column contain multiple values)
    Table - Patients
            - Contact col contains phone number and email id

    Table - Treatment & Treatment_cut
            - Auraling and navodra col should be split into 2 columns start and end dose
            - Merge both table Treatment & Treatment_cut.

    Table - Adverse_reaction
            - This table shold not exist independently
```

## Data Quality Dimensions:

1. Completeness - Is Data Missing?
2. Validity - Is Data Invalid (Negative Height, Duplicate Patients)
3. Accuracy - Data Is Valid But Not Accurate (Weight - 1kg )
4. Consistency - Both Valid & Accurate But Written Differently ( New York -> Ny )

## Order Of Severity (Who Is Dangerous Problem In Among All 4 Data Quality Dimensions)

$$\begin{tikzcd}
\boxed{\text{Completene}} & \boxed{\text{Validity}} \arrow[l] & \boxed{\text{Accuracy}} \arrow[l] & \boxed{\text{Consistency}} \arrow[l]
\end{tikzcd}$$

**After assigning label to your problems now follow the below process to clean your data in recommended Data cleaning orders as per the label given to your problems.**

## Data cleaning order

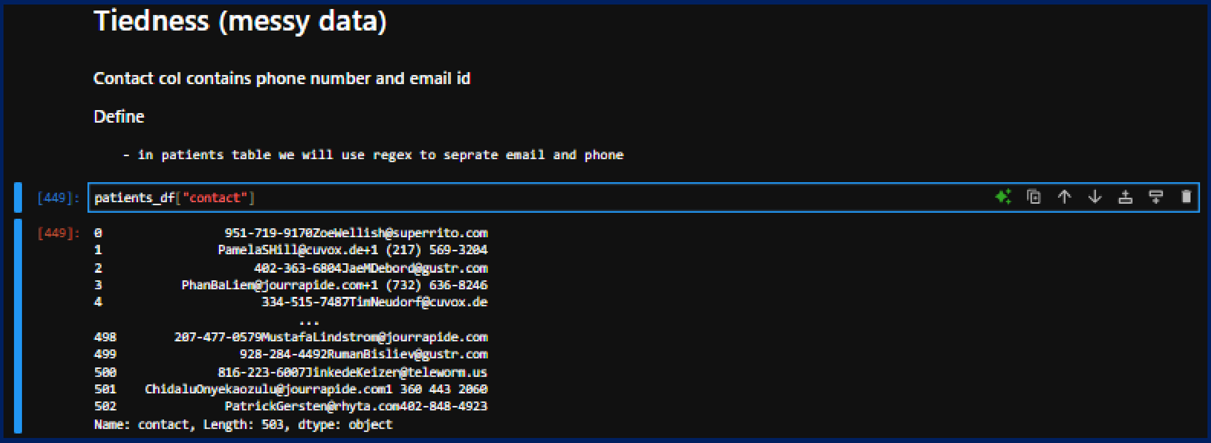
1. Quality - Completeness
2. Tidiness
3. Quality - Validity
4. Quality - Accuracy
5. Quality - consistency

## Step involve in data cleaning

- Define - solution of problem you have found
- Code - write code of the solution
- Test - check code is working to solve the problem.

**Always make sure to create a copy your pandas data frame before starts the cleaning process** *

## **Define**: Write how you solve the problem



### Tiedness (messy data)

Contact col contains phone number and email id

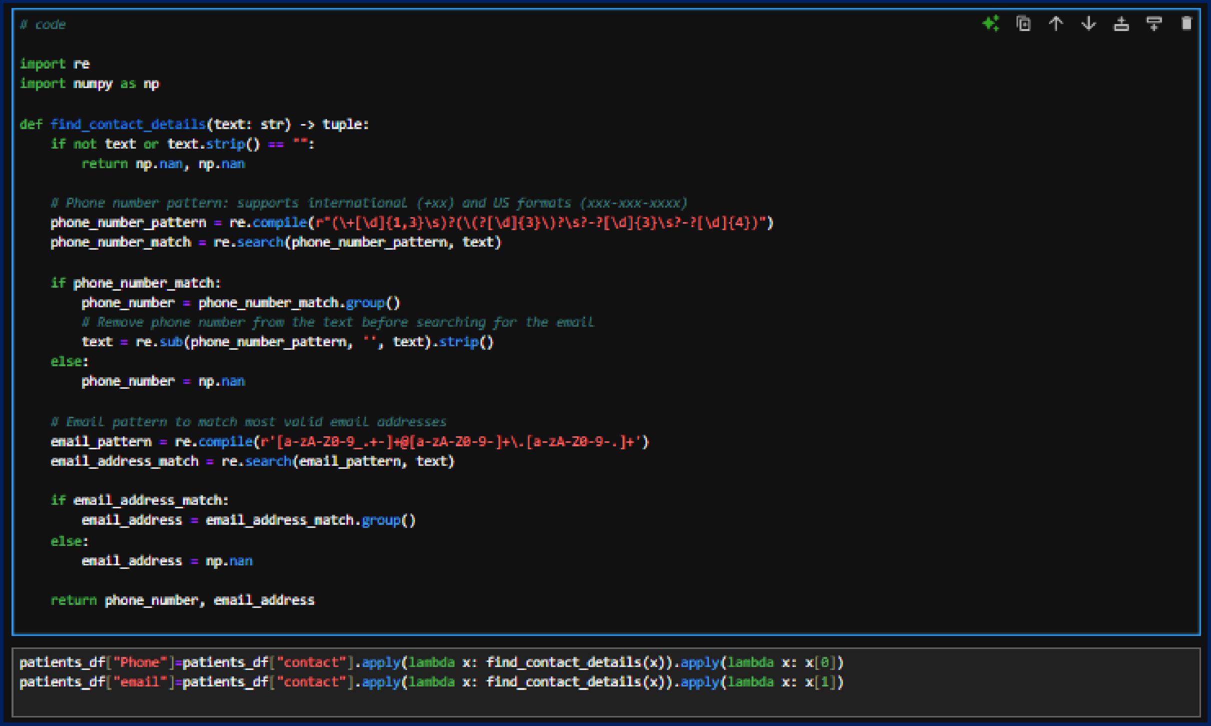
Define

```
    - in patients table we will use regex to seprate email and phone
```

```
[449]: patients_df["contact"]
```

```
[449]: 0                951-719-9170ZoeWellish@superrito.com
       1               PamelaSHill@cuvox.de+1 (217) 569-3204
       2                   402-363-6804JaeMDebord@gustr.com
       3          PhanBaLiem@jourrapide.com+1 (732) 636-8246
       4                    334-515-7487TimNeudorf@cuvox.de
                                     ...
       498      207-477-0579MustafaLindstrom@jourrapide.com
       499                928-284-4492RumanBisliev@gustr.com
       500             816-223-6007JinkedeKeizer@teleworm.us
       501    ChidaluOnyekaozulu@jourrapide.com1 360 443 2060
       502              PatrickGersten@rhyta.com402-848-4923
       Name: contact, Length: 503, dtype: object
```

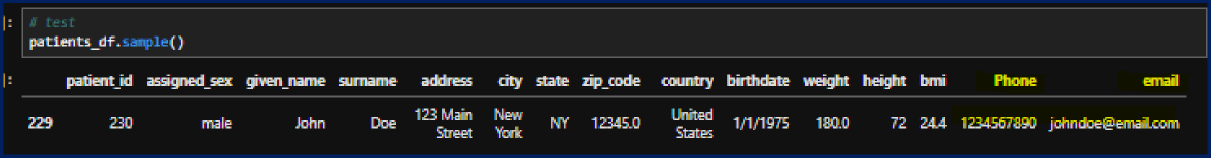
**Code**: write your code



```python
# code

import re
import numpy as np

def find_contact_details(text: str) -> tuple:
    if not text or text.strip() == "":
        return np.nan, np.nan

    # Phone number pattern: supports international (+xx) and US formats (xxx-xxx-xxxx)
    phone_number_pattern = re.compile(r"(\+[\d]{1,3}\s)?(\(?[\d]{3}\)?\s?-?[\d]{3}\s?-?[\d]{4})")
    phone_number_match = re.search(phone_number_pattern, text)

    if phone_number_match:
        phone_number = phone_number_match.group()
        # Remove phone number from the text before searching for the email
        text = re.sub(phone_number_pattern, '', text).strip()
    else:
        phone_number = np.nan

    # Email pattern to match most valid email addresses
    email_pattern = re.compile(r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+')
    email_address_match = re.search(email_pattern, text)

    if email_address_match:
        email_address = email_address_match.group()
    else:
        email_address = np.nan

    return phone_number, email_address
```

```python
patients_df["Phone"]=patients_df["contact"].apply(lambda x: find_contact_details(x)).apply(lambda x: x[0])
patients_df["email"]=patients_df["contact"].apply(lambda x: find_contact_details(x)).apply(lambda x: x[1])
```

**Test**: test your output is worked or not as per the define solution



```python
# test
patients_df.sample()
```

| | patient_id | assigned_sex | given_name | surname | address | city | state | zip_code | country | birthdate | weight | height | bmi | Phone | email |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 229 | 230 | male | John | Doe | 123 Main Street | New York | NY | 12345.0 | United States | 1/1/1975 | 180.0 | 72 | 24.4 | 1234567890 | johndoe@email.com |

# Thanks You