On inspecting the data:

A screenshot of a computer

AI-generated content may be incorrect.

1.There was no set datatype for each of the fields

* facility\_id had no meaningful identification to it, it is the primary key for the dataset, what is the fromat for the this field ?
* record line 2 & 3 are duplicates
* change the dataset fields headers to capitals letters
* remove line 2 or 3, under facility\_name place (St.) at the front and remove the “()”
* for facility\_type, standardize the word, if “Community Health Ctr.” -> Health Center, if Hosp. -> Hospital
* for capacity it would be renamed Bed\_Capacity, and the records would be a numerical datatype.
* For the region, all “ST.” must be “St.” with a space after the full stop, remove “parish” and common letters after the first letter,
* For licence\_issue\_date & inspection\_date , standard dd-mm-yy style
* For gps\_location, standardize coordinate xx.xxxxx, yy.yyyyy
* For remarks, if blank “-”
* Remove emojies from Facility Name, Facility type, and remarks
* Harris \nNewLine to be removed from facility name

Profile of the Datase:tcleaned\_health\_registry

A screenshot of a computer

AI-generated content may be incorrect.

#cleaning up facility names

import re

def clean\_facility\_name(name):

    if pd.isna(name):

        return name

    # Remove text in parentheses

    name = re.sub(r"\s\*\([^)]\*\)", "", name)

    # Remove emojis

    name = re.sub(r"[^\w\s.,'-]", "", name)

    # Clean extra whitespace

    return name.strip()

df["Facility Name"] = df["Facility Name"].apply(clean\_facility\_name)

# Dictionary of common abbreviations and their replacements

replacements = {

    r"\bHosp\.?\b": "Hospital",           # Hosp. or Hosp

    r"\bHlth Ctr\.?\b": "Health Center",  # Hlth Ctr. or Hlth Ctr

    r"\bCtr\.?\b": "Center",              # Ctr.

    r"\bMed\.?\b": "Medical",             # Med. (if applicable)

    r"\bPLC\b": "",                       # Remove "PLC" if not needed

}

def expand\_abbreviations(name):

    if pd.isna(name):

        return name

    for pattern, replacement in replacements.items():

        name = re.sub(pattern, replacement, name, flags=re.IGNORECASE)

    return name.strip()

df["Facility Name"] = df["Facility Name"].apply(expand\_abbreviations)

display(df.head(10))