Data Cleaning Project Documentation

# 1. Introduction

This document outlines the steps taken to clean a messy dataset using Python and Pandas. The dataset contains columns such as Year, Month, State, Location, Food, Ingredient, Species, Serotype/Genotype, Status, and Illness. The goal of the cleaning process is to ensure data consistency, handle missing values, standardize formats, and prepare the data for analysis.

Data Source: Kaggle

# 2. Initial Data Loading and Inspection

The first step involves loading the dataset and inspecting it for general structure and issues.

import pandas as pd  
  
# Load the dataset  
df = pd.read\_csv("your\_file.csv")  
  
# Inspect the dataset  
print(df.head())  
print(df.info())  
print(df.describe(include='all'))

# 3. Column Name Standardization

Standardizing column names to ensure uniform formatting (e.g., lowercase, no spaces).

df.columns = df.columns.str.strip().str.lower().str.replace(' ', '\_')

# 4. Handling Missing Values

Identifying and managing missing values in key columns.

print(df.isnull().sum())  
  
df = df.dropna(subset=['state', 'food', 'illness'])  
df['ingredient'] = df['ingredient'].fillna('Unknown')

# 5. Data Type Fixes and Date Conversion

Converting 'year' and 'month' columns into a proper datetime format.

df['year'] = df['year'].astype(str).str.extract(r'(\d{4})')  
df['month'] = df['month'].str.strip().str.title()  
df['date'] = pd.to\_datetime(df['month'] + ' ' + df['year'], errors='coerce')

# 6. Standardizing Categorical Values

Ensuring consistency in categorical text columns.

df['status'] = df['status'].str.strip().str.title()  
df['illness'] = df['illness'].str.strip().str.title()  
df['state'] = df['state'].str.strip().str.upper()

# 7. Splitting Composite Columns

Splitting 'serotype/genotype' into separate columns.

df[['serotype', 'genotype']] = df['serotype/genotype'].str.split('/', n=1, expand=True)

# 8. Removing Duplicates

Removing duplicate records to avoid redundancy.

df = df.drop\_duplicates()

# 9. Saving the Cleaned Dataset

Saving the cleaned dataset to a new CSV file.

df.to\_csv("cleaned\_output.csv", index=False)