#data cleaning

import pandas as pd

import ace\_tools\_open as tools;

us\_counties\_path = 'uscounties.csv'

counties\_path = 'Counties.csv'

us\_counties = pd.read\_csv(us\_counties\_path)

counties = pd.read\_csv(counties\_path)

us\_counties.head(), counties.head()

merged\_data = pd.merge(counties, us\_counties, how='left', left\_on='fips', right\_on='county\_fips')

# Drop rows where the merge failed (missing values in `state\_name` after merge)

merged\_data\_cleaned = merged\_data.dropna(subset=['state\_name'])

# Add a `days\_since\_zero` column (calculate days from the earliest date)

merged\_data\_cleaned['date'] = pd.to\_datetime(merged\_data\_cleaned['date'])

day\_zero = merged\_data\_cleaned['date'].min()

merged\_data\_cleaned['days\_since\_zero'] = (merged\_data\_cleaned['date'] - day\_zero).dt.days

columns\_to\_drop = [

'county\_y', 'county\_ascii', 'county\_full', 'county\_fips',

'state\_id', 'state\_name'

]

merged\_data\_cleaned = merged\_data\_cleaned.drop(columns=columns\_to\_drop)

# Sort data by fips and date to simplify calculations

merged\_data\_cleaned = merged\_data\_cleaned.sort\_values(by=['fips', 'date'])

def calculate\_last\_week\_cases(df):

df['cases\_last\_week'] = df.groupby('fips')['cases'].diff(periods=7).fillna(0)

df['deaths\_last\_week'] = df.groupby('fips')['deaths'].diff(periods=7).fillna(0)

df.loc[df['days\_since\_zero'] < 7, 'cases\_last\_week'] = df['cases']

df.loc[df['days\_since\_zero'] < 7, 'deaths\_last\_week'] = df['deaths']

return df

# Apply the function

merged\_data\_cleaned = calculate\_last\_week\_cases(merged\_data\_cleaned)

merged\_data\_cleaned['cases\_per\_100k'] = (

merged\_data\_cleaned['cases\_last\_week'] / merged\_data\_cleaned['population'] \* 100000

).fillna(0)

merged\_data\_cleaned['deaths\_per\_100k'] = (

merged\_data\_cleaned['deaths\_last\_week'] / merged\_data\_cleaned['population'] \* 100000

).fillna(0)

def classify\_risk\_level(row):

if row['cases\_per\_100k'] > 100 or row['deaths\_per\_100k'] > 5:

return 'High'

elif row['cases\_per\_100k'] > 50 or row['deaths\_per\_100k'] > 2:

return 'Medium'

else:

return 'Low'

merged\_data\_cleaned['risk\_level'] = merged\_data\_cleaned.apply(classify\_risk\_level, axis=1)

# Display

tools.display\_dataframe\_to\_user(name="Final Cleaned Dataset with Risk Levels and Metrics", dataframe=merged\_data\_cleaned)

#Output

output\_path = 'clean\_covid\_case.csv'

merged\_data\_cleaned.to\_csv(output\_path, index=False)