Section	Python Code	Explanation
Code Setup	import pandas as pd	Imports pandas for data manipulation.
Importing Data	nebraska_data = pd.read_csv('Nebraska_Violations_Formatted.csv')	Reads the CSV file into a DataFrame.
Importing Data	nebraska_data = pd.DataFrame(nebraska_data)	Ensures the data is in DataFrame format.
Importing Data	print(nebraska_data), print(summary), print(type_summary)	Prints the DataFrame and summaries.
Summarizing Data	nebraska_data.head(10)	Displays the first 10 rows.
Summarizing Data	nebraska_data.tail()	Displays the last 5 rows.
Summarizing Data	nebraska_data.columns	Lists the column names.
Grouping Data	grouped_data = nebraska_data.groupby(['pws_type', 'contaminant']).size	Groups by type and contaminant.
Grouping Data	summary = nebraska_data.groupby(['pws_type', 'contaminant']).size().res	Groups the data similarly.
Statistical Summary	stat_summary = summary.groupby('contaminant').agg(max=('n','max'), m	Calculates summary statistics.
Statistical Summary	type_summary = summary.groupby('contaminant').agg(max=('n','max'), m	Calculates summary statistics.
Statistical Summary	total_per_contaminant = summary.groupby('contaminant')['n'].sum()	Calculates total violations.
Statistical Summary	summary = summary.merge(total_per_contaminant, on='contaminant')	Merges total violations.
Statistical Summary	summary['percent_of_total'] = (summary['n'] / summary['total']) * 100	Calculates the percentage of violations.
Statistical Summary	summary_pivot = summary.pivot(index='contaminant', columns='pws_typ	Pivots the table for better readability.
Filtering Data	summary_pivot.to_csv('percentages.csv')	Exports the pivoted table.
Filtering Data	active_systems = nebraska_data[nebraska_data['activity_status'] == 'Activ	Filters for active systems.
Filtering Data	active_systems2 = nebraska_data.query('activity_status == "Active"')	Alternative filter for active systems.
Filtering Data	subset0 = nebraska_data[(nebraska_data['activity_status'] == 'Active') & (	Filters for active systems and specific violations.
Filtering Data	filter1 = nebraska_data	Initial filter.
Filtering Data	filter2 = filter1[filter1['population'] <= 10000]	Filters for populations <= 10,000.
Filtering Data	group1 = filter2.groupby(['pws_type', 'primary_source', 'contaminant']).si	Groups the filtered data.
Filtering Data	analysis1 = group1.groupby(['pws_type', 'primary_source', 'contaminant']	Summarizes the grouped data.
Filtering Data	small_pop_data = nebraska_data[nebraska_data['pop_category'] == 'sma	Extracts small population systems.
Defining Functions	def categorize(x):	Defines a function to categorize populations.
Defining Functions	if x < 1000: return 'small'	Categorizes small populations.
Defining Functions	elif 1000 <= x <= 10000: return 'medium' else: return 'large'	Categorizes medium and large populations.