

Exercise: An Analysis of Epileptic and Neurological Mortality

Summary

Dangerous environmental neurotoxins are on the rise throughout the United States. It is theorized that neurotoxins play a key role in the rising levels of neurological deaths in the United States. Researchers have found a causal link to the weakening of one's seizure threshold because of neurotoxins and the development of epilepsy, a neurological disorder. This can raise health concerns if the quantity of individuals dying from neurological and epileptic deaths is increasing rapidly, as well as the rate at which it happens. It is also of health concern if these deaths are concentrated in certain states throughout the United States. This exercise analyses that issue for all states in the United States.

Input Data

There are two input files, and each gives the mortality levels associated with neurological disorders in general, and epileptic disorders on their own. Each file is in text (TXT) format with death counts throughout the years. File `Underlying Cause of Death, 1999-2020.txt` shows the death and population counts for overall neurological disorders from years 1999 to 2020 and organized by all states for which data was reported to the Centers for Disease Control and Prevention (CDC). File `Underlying Cause of Death, 1999-2020 Epilepsy.txt` shows the death and population counts for epileptic disorders from years 1999 to 2020 and organized by all states for which data was reported to the Centers for Disease Control and Prevention (CDC).

Deliverables

The deliverable for this assignment is a script, `read_cdc.py`, that generates a range of figures using Pandas and Matplotlib, as well as mortality calculations.

Instructions

1. Import `pandas` and `matplotlib.pyplot`, and `seaborn` as `sns`.
2. Create a new blank dataframe for holding the neurological mortality data by setting variable `raw` to the result of calling `pd.DataFrame()` with no arguments.
3. Print the total columns of data in this dataframe by calling `len()` on `raw`.
4. Print the "Notes" in `raw`.
5. Create a dataframe called `has_notes` to indicate the "Notes" in `raw` are `False` and do not belong.
6. Create a dataframe called `neuro` with the indication of the "Notes" not belonging.
7. Drop the "Note" column in `raw` from the `neuro` dataframe.
8. Drop the "Year" subset in `neuro` from the `neuro` dataframe.
9. Print the updated total columns of data in the `neuro` dataframe by calling `len()` on `neuro`.
10. Create a new blank dataframe for holding the epileptic mortality data by setting variable `epilepsy` to the result of calling `pd.DataFrame()` with no arguments.
11. Print the total columns of data in this dataframe by calling `len()` on `epilepsy`.

12. Print the “Notes” in epilepsy.
13. Create a dataframe called `has_notes` to indicate the “Notes” in epilepsy are False and do not belong.
14. Create a dataframe called `epi` with the indication of the “Notes” not belonging.
15. Drop the “Note” column in epilepsy from the `epi` dataframe.
16. Drop the “Year” subset in `epi` from the `epi` dataframe.
17. Print the updated total columns of data in the `epi` dataframe by calling `len()` on `epi`.
18. Create a dictionary called `new_columns_neuro` to rename the rows in the `neuro` data frame. Use “State Code” for the State Code row, “`neuro_year_code`” for the “Year Code”, “`neuro_crude_rate`” for the Crude Rate, “`neuro_pop`” for the Population, and “`neuro_deaths`” for the Deaths. This will make it easier to decipher between columns once this dataframe is merged.
19. Rename the columns in `neuro` using the dictionary keys produced above using the `rename` call with the argument `columns=new_columns_neuro`.
20. Create a dictionary called `new_columns_epi` to rename the rows in the `epi` data frame. Use “State Code” for the State Code row, “`epi_year_code`” for the “Year Code”, “`epi_crude_rate`” for the Crude Rate, “`epi_pop`” for the Population, and “`epi_deaths`” for the Deaths. This will make it easier to decipher between columns once this dataframe is merged.
21. Rename the columns in `epi` using the dictionary keys produced above using the `rename` call with the argument `columns=new_columns_epi`.
22. Now merge on the epileptic data by setting `neuro_epi` to the result of calling `.merge()` on `neuro` with arguments `pop`, `on=["State", "Year"]`, `how="left"`, `validate='1:1'`, and `indicator=True`.
23. Drop the “`neuro_crude_rate`”, “`epi_crude_rate`”, and “`epi_state_code`” from the `neuro_epi` dataframe as they will not be crucial to this analysis.
24. Calculate the rate at which neurological deaths are happening and add it to the `neuro_epi` dataframe using “`neuro_death_rate`” as the column name. Divide the “`neuro_deaths`” column in the `neuro_epi` dataframe by the “`neuro_pop`” column in `neuro_epi`, per one million individuals to control for population.
25. Calculate the rate at which epileptic deaths are happening and add it to the `neuro_epi` dataframe using “`epi_death_rate`” as the column name. Divide the “`epi_deaths`” column in the `neuro_epi` dataframe by the “`epi_pop`” column in `neuro_epi`, per one million individuals to control for population.
26. Calculate an epileptic to neurological death rate ratio and add it to the `neuro_epi` dataframe using “`epi_to_neuro_ratio`” as the column name. Divide the “`epi_death_rate`” column in the `neuro_epi` dataframe by the “`neuro_death_rate`” column in `neuro_epi`.
27. Create a new dataframe called `grouped` to the data in `neuro_epi` by “State” and then add it to a dataframe called `average` that includes the average neurological and epileptic death rates for each state.
28. Create a new dataframe called `grouped_death_counts` to the data in `neuro_epi` by “State” and then add it to a dataframe called `average_death_counts` that includes the average

neurological and epileptic deaths for each state.

29. Set `plt.rcParams['figure.dpi']` to 300 to improve the resolution of an upcoming figure and use `sns.set_theme(style="white")` to set its theme.
30. Create a scatterplot to visualize epilepsy death rates on neurological death rates. Use the `.plot.scatter()` call with arguments `x="neuro_death_rate"` and `y="epi_death_rate"`. Title it "Epilepsy Death Rates Scattered on Neurological Death Rates". Save the figure as **"neuro_epi_death_rate_scatter.png"**.
31. Create a scatterplot to visualize epilepsy deaths on neurological deaths. Use the `.plot.scatter()` call with arguments `x="neuro_deaths "` and `y="epi_deaths "`. Title it "Epilepsy Death Counts Scattered on Neurological Death Counts". Save the figure as **"neuro_epi_death_counts_scatter.png"**.
32. Create a linegraph to visualize average neurological death rates. Use the `sns.lineplot()` call with arguments `data=neuro_epi`, `x="Year"`, and `y="neuro_death_rate"`. Title it "Average Neurological Death Rate Linegraph". Save the figure as **"neuro_death_rate_line_graph.png"**.
33. Create a linegraph to visualize average epileptic death rates. Use the `sns.lineplot()` call with arguments `data=neuro_epi`, `x="Year"`, and `y="epi_death_rate"`. Title it "Average Epilepsy Death Rate Linegraph". Save the figure as **"epi_death_rate_line_graph.png"**.
34. Create a linegraph to visualize average neurological death counts. Use the `sns.lineplot()` call with arguments `data=neuro_epi`, `x="Year"`, and `y="neuro_deaths "`. Title it "Average Neurological Death Count Linegraph". Save the figure as **"neuro_death_count_line_graph.png"**.
35. Create a linegraph to visualize average epileptic death counts. Use the `sns.lineplot()` call with arguments `data=neuro_epi`, `x="Year"`, and `y="epi_deaths "`. Title it "Average Epileptic Death Count Linegraph". Save the figure as **"epi_death_count_line_graph.png"**.
36. Now create a figure with two panels in a row by setting `fig, (ax1,ax2)` to the result of calling `plt.subplots(1,2)`.
37. Next, call `.plot.hist()` on the "neuro_death_rate" column of average with the argument `ax=ax1` to put the histogram in the left panel.
38. Use the `.set_title()` method of `ax1` to set its title to "Cases of Neurological Deaths".
39. Now call `.plot.hist()` on the "epi_death_rate" column of average with the argument `ax=ax2`. Then set its title to "Cases of Epilepsy Deaths".
40. Tighten the figure's layout using `.tight_layout()` and save the figure **"death_frequency.png"**.
41. Run summary statistics on the "neuro_death_rate" column of average and use the `sort_values()` call and print average.
42. Run summary statistics on the "epi_death_rate" column of average and use the `sort_values()` call and print average.
43. Run summary statistics on the "neuro_deaths" column of average `_death_counts` and use the `sort_values()` call and print `average_death_counts`.
44. Run summary statistics on the "epi_deaths" column of average `_death_counts` and use the `sort_values()` call and print `average_death_counts`.

Results

1. Top three states with neurological mortality:
 - California
 - Texas
 - Florida
2. Top three states with epileptic mortality:
 - California
 - Texas
 - Michigan
3. Top three ranks neurological mortality rates:
 - North Dakota
 - South Dakota
 - Washington
4. Top three ranks of epileptic mortality rates:
 - South Dakota
 - Rhode Island
 - Montana
5. Overall death counts and death rates are increasing at alarming rates for both neurological and epileptic mortalities across the United States.
6. There seems to be a positive correlation between epileptic deaths and neurological deaths, as an increase in epileptic deaths is indicating an even higher number of neurological deaths across states.
7. States such as California, Texas, and South Dakota should shift their policy and investigative strategies towards initiatives to combat such disorders, as there might be high levels of neurotoxins in their environments.

Tips

- If you run the code all at once instead of doing so individually by cell, Spyder might output the plots incorrectly for reasons out of my knowledge.