Review the Mortality Notebook

*In this exercise, you’ll run the cells in the Mortality Notebook. This Notebook includes all the examples that are in this chapter, plus a few variations. As you run each cell, be sure that you understand what it does. To help you understand it, you may want to change some of the parameters to see how that changes the results.*

## Open the Notebook and run the cells that get the data

1. Start JupyterLab and open the Notebook named lab\_2-1\_mortality that is in Blackboard
2. Use the Kernel menu to restart the kernel and clear all outputs.
3. Run the cells that import the Pandas module and get the data.// *import pandas as pd*
4. Run the cells that save and restore the DataFrame. //

*mortality\_url = "https://data.cdc.gov/api/views/v6ab-adf5/rows.csv?accessType=DOWNLOAD"*

*mortality\_data = pd.read\_csv(mortality\_url)*

*mortality\_data.to\_pickle('mortality\_data.pkl')*

*mortality\_data = pd.read\_pickle('mortality\_data.pkl')*

## Run the cells that examine and clean the data

1. Run the cells that use various techniques to display the data in the DataFrame

*mortality\_data*

*mortality\_data.head(5)*

*mortality\_data.tail(3)*

*display(mortality\_data)*

1. Run the cells that display the DataFrame attributes.

*mortality\_data.values*

1. Run the cells that change and display the column names.

*print("Index: ", mortality\_data.index)*

*print("Columns:", mortality\_data.columns)*

*print("Size: ", mortality\_data.size)*

*print("Shape: ", mortality\_data.shape)*

*mortality\_data.columns = mortality\_data.columns.str.replace(" ", "")*

*print(mortality\_data.columns)*

1. Run the cells that use the info() and nunique() method, and note how the changes to the column names are reflected in the results.

*mortality\_data.info()*

*mortality\_data.nunique()*

1. Run the cells that use the describe() method, and note how the T attribute changes the display.

*mortality\_data.describe()*

*mortality\_data.describe().T*

1. Run the cells that save and restore the cleaned DataFrame.

*mortality\_data.to\_pickle('mortality\_cleaned.pkl')*

*mortality\_data = pd.read\_pickle('mortality\_cleaned.pkl')*

*mortality\_data.head()*

## Run the cells that access the data

1. Run the cells that access columns and note the difference in the output when you use dot notation or brackets to access a single column and a list to access multiple columns.
2. Run the cells that access rows and note the use of the query() method.
3. Run the cells that access rows and columns using the query() method along with dot notation or brackets, and note the difference in the output between the first two cells and the next two cells.
4. Run all the cells that use the loc[ ] and iloc[ ] accessors to access subsets of rows and columns, and note the use of lists and slices.

Run the cells that prepare the data

1. Run the cells that sort the data and note the difference in the sequence of the data for the last three cells.
2. Run the cells that apply statistical methods and note the use of dot notation and brackets for accessing columns.
3. Run the cells that uses column arithmetic to add a column named MeanCentered to a DataFrame, modify the data in the DeathRate column, and display the results.
4. Run the cell that modifies the string data in the AgeGroup column. Then, run the cells that save and restore the prepared data.

## Run the cells that shape the data

1. Run the cells that set different indexes for the DataFrame and note that the index must be reset before a new one can be set.
2. Run the cells that pivot the data and note that both of these create a new DataFrame named mortality\_wide. As a result, the second DataFrame replaces the first one. Note too that all the other columns are pivoted if you don’t specify a values parameter.
3. In the cells that melt the wide DataFrame, note that the first cell saves the wide DataFrame as an Excel file, and the second one imports the Excel file back into the DataFrame. That’s just an easy way to create a wide DataFrame that can be used to demonstrate the use of the melt() method.
4. In the cell that contains the melt() method, note that this method creates a new DataFrame named mortality\_long, but it melts just two of the wide columns into the AgeGroup and DeathRate columns. To melt all four, you can delete the value\_vars parameter and run the cell again.
5. Run the cells that melt the data and then the cells that save and restore the wide DataFrame.

## Run the cells that analyze the data

1. Run the cells that group and aggregate the data and note how the functions are applied to the columns in each group.
2. Run the cells that visualize the data, and note how the plot() method is chained to the preceding methods.