**AUTHOR: Jiebin Zhu (Alex)**

**ABOUT THE DATA AND ITS SOURCE:** There are two separate datasets that were obtained as a CSV file for this project. One dataset is obtained from Kaggle (https://www.kaggle.com/fedesoriano/stroke-prediction-dataset) and the other one is obtained from the CDC website (https://www.cdc.gov/nchs/pressroom/sosmap/stroke_mortality/stroke.htm).

The dataset from Kaggle had listed its source as:
"**(Confidential Source)** - *Use only for educational purposes*
If you use this dataset in your research, please credit the author."
I believe that the data here is really sensitive if it's tied by to the author then it can be reversed engineered into identifying the individual that's recorded in the dataset. The dataset has also recently been updated as the last updated date was Jan 26, 2021. It's safe to assume that the dataset is accurate and very recent for its usability here.

The dataset from the CDC is very easy to understand as it's separated by years and states. It contains only 3 other fields has the death rate per 100,000 total population, death count, and the sub url to access the state specific data.

A third dataframe will be created from the information that's obtained within the CDC Dataset. It's called the "rankerDF" and it contains the ranks for the deaths caused by "strokes" and "heart disease" per state. This is gathered through web scraping using the sub urls provided in the CDC Dataset.

Here's a list of all those attributes with its dataset:

**Kaggle Dataset**

| FIELD | DESCRIPTION | EXAMPLE |
|---|---|---|
| id | Unique identifier | 9046 |
| gender | Male, Female, or Other (only 1 entry has Other) | Male |
| age | Age of the patient | 67 |
| hypertension | 0 if the patient doesn't have hypertension, 1 if the patient has hypertension | 0 |
| heart_disease | 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease | 1 |
| ever_married | If the patient was ever married or not; Yes or No | Yes |
| work_type | "children", "Govt_jov", "Never_worked", "Private" or "Self-employed" | Private |
| Residence_type | "Rural" or "Urban" | Urban |
| avg_glucose_level | average glucose level in blood | 228.69 |
| bmi | body mass index | 36.6 |
| smoking_status | "formerly smoked", "never smoked", "smokes" or "Unknown";<br>"Unknown" in smoking_status means that the information is unavailable for this patient | formerly smoked |
| stroke | 1 if the patient had a stroke or 0 if not | 1 |

**CDC Dataset**

| FIELD | DESCRIPTION | EXAMPLE |
|---|---|---|
| YEAR | Year of the data | 2019 |
| STATE | Abbreviated state names | AL |
| RATE | Death rate based on stroke per 100,000 of total population | 51.5 |
| DEATHS | Death counts based on stroke | 3141 |
| URL | URL after the https://cdc.gov to obtain the state | /nchs/pressroom/states/alabama/al.htm |

**rankerDF**

| FIELD | DESCRIPTION | EXAMPLE |
|---|---|---|
| state | State abbreviation | AL |
| stroke_rank | Ranking of stroke as leading cause of death (1 being highest, 10 being lowest) | 4 |
| heart_disease_rank | Ranking of heart disease as leading cause of death (1 being highest, 10 being lowest) | 1 |

**DATA EXPLORATION & DATA CLEANING**

**Kaggle Dataset**

The Kaggle dataset here requires a little bit of cleaning to get it into a good condition for processing. First the entire column of "IDs" were removed within the CSV file itself because it doesn't provide any value for the data analysis itself. After browsing the dataset a bit longer in the CSV format within excel, I noticed two other things. One is that under the "Gender" column there is only one entry of "Other". The other thing is that only the "bmi" column contains some N/A values. For the "Gender" column containing only one entry of "Other", that was removed manually within the CSV file in excel. Then the file was saved as a "modified" CSV file, leaving the original data intact. Before, the "bmi" column can be cleaned up, the data has to be imported into a pandas Dataframe (called strokedDF) and remove the first row of data there. This was done because I assigned some custom column names for the dataset and didn't need the original one. Now tackling the "bmi" column, the N/A values were all filled in with 0s for since there 201 entries affected and it seemed like a lot of data to remove, thus this method would help keep those entries. The only caveat is that mean function would be skewed a bit here if used within this column, so if this were to be used then it should be followed with a median function to see if the 0s are outliers affecting the results hugely. Lastly, the columns that contain any sort of numeric values were converted into a numeric datatype so it can be used in calculations.

**CDC Dataset**

The CDC dataset itself is mostly very clean except for a few things. Nothing looks out of the ordinary except for the "commas" that it contains within some death counts that were recorded from the previous years before 2018. The CDC probably didn't have a structured way of storing this information back then so the commas would need to be removed after importing. The years that were recorded is very weird as well since the data goes from 2019, 2018, 2017, 2016, 2015,2014 and then skips all the way to 2005 without any of the years in between. No idea why this was the case, but certainly something worth pointing out. After loading the dataset into a pandas Dataframe (called cdc_strokeDF), the first row was removed due to the same reason as above, where I used custom column names for this and didn't need the original ones. Next, the sub_url column would be dropped only **AFTER** the **rankerDF** is created (explained more later on) since it's not longer needed. Then the last row if the dataset is dropped because it contained blank information. After that, the commas within "death_count" were all removed, and the death_rate and death_count columns can now be converted to a numeric datatype.

**rankerDF**

This dataframe is created via scraping of the information by using the sub_url from the CDC dataset. There's no cleaning required here because it's a tailored made dataset from the semi-structured data, as a lot of the work was already complete in the scraping process. This will be elaborated further later on.

All three of the dataframes here are now cleaned up and ready for analysis.

**DATA ANALYSIS**

There will be 4 questions (potentially 5 due to Q3, depending on how you view it) that would be answered with the analyses conducted from these three dataframes.

Q1: What is the average and median stroke death counts/death rates for all 50 states in all the different years listed?

**Unit of analysis:** years

**Comparison:** For each year, compute the median and the mean for the death rates and death counts for the 50 states

**Output:** 2 sets of tables containing the mean and median of death rates and death counts for each year.

Mean

| year | death_rate | death_count |
|---|---|---|
| 2005 | 48.938 | 2866.96 |
| 2014 | 36.858 | 2657.78 |
| 2015 | 37.482 | 2801.62 |
| 2016 | 37.138 | 2837.8 |
| 2017 | 37.456 | 2922.74 |
| 2018 | 36.808 | 2951.16 |
| 2019 | 36.318 | 2994.68 |

Median

| year | death_rate | death_count |
|---|---|---|
| 2005 | 48.75 | 2262 |
| 2014 | 36.75 | 1948.5 |
| 2015 | 37.3 | 1965.5 |
| 2016 | 37.25 | 2000 |
| 2017 | 37.5 | 2058 |
| 2018 | 36.8 | 2114 |
| 2019 | 36.2 | 2215 |

At first glance, it looks like average of the death counts are really high when comparing it to the median and might cause people to believe that there could be an outlier here. However, this doesn't seem like an one-off cases as all the average death counts were higher and roughly in the same range when comparing it to the median. The case here could be that states with a higher population would tend to report a higher amount of death here and this shouldn't be much of a surprise since there are a few states with really high population such as New York and California. Another thing that this really calls out is the death rate because it's getting lower and lower as the years progresses. 2005 was really high, but in 2019 the death rate dropped a lot comparing it to back then probably because of the advancement of medical technology.

Q2: What's the most common ranking for stroke and heart diseases as a top 10 leading cause of death in the US?

**Unit of analysis:** states

**Comparison:** For all the states, find the most common ranking among strokes and heart diseases as a leading cause of death

**Output:** 1 line of data containing the rank of strokes and rank of heart disease as a leading cause of death for all the states.

Ranking

| | stroke_rank | heart_disease_rank |
|---|---|---|
| 0 | 5 | 1 |

This was obtained by finding the mode of the two different rankings for stroke and heart diseases for all 50 states. The main datasets imported is for analyzing the stroke mortality and characteristics for predicting it. I noticed that "heart disease" was a factor in predicting stroke within the Kaggle dataset and decided to obtain an extra piece of information to see how its mortality could be compared to the stroke mortality. It's very shocking to see that the no.1 leading cause of death for the US is actually heart disease and not strokes. After seeing such high death rates for the strokes already (being ranked 5) it's scary to imagine where the heart disease numbers are at.

Q3: What's the total amount of male/female who has had a stroke before, based on their work types? How about the same thing for heart disease?

**Unit of analysis:** gender, job types

**Comparison:** For each gender and job types, count the number of strokes/heart disease.

**Output:** 2 sets of chart with the count of individuals who obtained 1 for each disease and each gender + job types combination.

| | stroke_count |
|---|---|
| ('Govt_job', 'Female') | 23 |
| ('Govt_job', 'Male') | 10 |
| ('Never_worked', 'Female') | 0 |
| ('Never_worked', 'Male') | 0 |
| ('Private', 'Female') | 76 |
| ('Private', 'Male') | 73 |
| ('Self-employed', 'Female') | 40 |
| ('Self-employed', 'Male') | 25 |
| ('children', 'Female') | 2 |
| ('children', 'Male') | 0 |

| | HD_count |
|---|---|
| ('Govt_job', 'Female') | 13 |
| ('Govt_job', 'Male') | 23 |
| ('Never_worked', 'Female') | 0 |
| ('Never_worked', 'Male') | 0 |
| ('Private', 'Female') | 68 |
| ('Private', 'Male') | 90 |
| ('Self-employed', 'Female') | 32 |
| ('Self-employed', 'Male') | 49 |
| ('children', 'Female') | 0 |
| ('children', 'Male') | 1 |

It looks like gender doesn't make a huge difference for strokes, but it does make a huge difference for heart diseases here. However, the common factor between the two high counts of strokes and heart diseases here is pertaining to the job type = Private. The other really low ones are the ones where job type = Never_worked. The only common thing I can infer from the information that can differentiate the job types is the stress level that could be involved. Private jobs are usually a lot more stressful, whereas the government jobs and Never_worked would be less stressful. This is a very important key point to highlight. Job Type= Private could be the most common characteristic of determining if a person would end up with a stroke/heart disease.

Q4: What is the median average glucose level for each gender varying on smoking status and residency type?

**NOTE**: There are “Unknown” responses here because the status could be unknown for many reasons such as the person is a toddler and the information was never recorded, and more.

**Unit of analysis:** gender, residency type, smoking status

**Comparison:** For each gender, job types, and smoking status calculate the median average glucose level for each set of combination.

**Output:** 1 table containing 16 rows of data on the different set of combinations of gender, residency type, and smoking status along with the corresponding median average glucose level

| | Med_avg_glucose_lvl |
|---|---|
| ('Rural', 'Unknown', 'Female') | 89.55 |
| ('Rural', 'Unknown', 'Male') | 92.815 |
| ('Rural', 'formerly smoked', 'Female') | 89.31 |
| ('Rural', 'formerly smoked', 'Male') | 97.59 |
| ('Rural', 'never smoked', 'Female') | 93.24 |
| ('Rural', 'never smoked', 'Male') | 94.78 |
| ('Rural', 'smokes', 'Female') | 94.71 |
| ('Rural', 'smokes', 'Male') | 95.465 |
| ('Urban', 'Unknown', 'Female') | 89.485 |
| ('Urban', 'Unknown', 'Male') | 90.13 |
| ('Urban', 'formerly smoked', 'Female') | 90.66 |
| ('Urban', 'formerly smoked', 'Male') | 96.04 |
| ('Urban', 'never smoked', 'Female') | 89.36 |
| ('Urban', 'never smoked', 'Male') | 92.87 |
| ('Urban', 'smokes', 'Female') | 93.23 |
| ('Urban', 'smokes', 'Male') | 90.95 |

My expectation was to see if smoking and the residency type would affect the glucose level since it could also be another effect of unhealthiness, but there’s no difference at all according to the dataset here. It’s pretty hard to obtain valuable information from this output because the median average glucose level is all within the common range of 90-110. Perhaps I should’ve compared it by people whose had or have a stroke versus people who don’t. Maybe the results would be more interesting there.

This marks the end of all the data analysis. The next section would contain a description of how the program works and the final conclusion of it.

**HOW THE PROGRAM WORKS**

First the two datasets that were obtained from Kaggle and CDC were imported into this. Then the program will create a dataframe to store that data. The fun part of the program begins in this next part because it's accessing the html of the 50 different states based on the sub_url information provided by the CDC file. Two functions were written for this process, but they are very similar in nature because they each will return a different rank only. What the function does here is obtain the 10 href tags for each "Leading Cause of Deaths" for each state in their html and then storing it into a list. Then its finding the index location of those tags to return the rank of strokes or heart diseases as a leading cause of death. The program will then loop this function 50 times, 1 time for each state, and then storing that rank information in a newly created "rankerDF". Since this process takes up a long time to complete because it's accessing the html 50 times, the rankerDF will then be saved as a CSV file in the local directory for reading it into the program when running the program again. The program will check if the "rankerDF.csv" is in the local directory and if it isn't it will go through that whole process, but only once, since it will then be saved into the local directory.

After all the dataframes are now created, then they will all go through the data cleaning steps identified above and be used to conduct analyses. The analyses will create new dataframes, but only for displaying the outputs cleanly.

**FINAL CONCLUSION**

The results of some questions were really shocking! All the analyses were conducted under the "Data Analysis" section, but I'd like to point out some important things that should be reiterated again.

1. Having the 2005 data point is actually really beneficial to using it for understanding why the death rate could be decreasing. There's 2 reasons thus could happen. Mathematically speaking, as population increases the death rate would be lowered if it's the same level of severity. However, I'd like to think positively and believe that it's the medical advancement that's lowering the death rates.
2. Watch out for your cardiovascular health! This is the nation's no.1 leading cause of death when being compared to the stroke (rank 5).
3. Job types = Private seem to have a huge correlation with having a stroke/heart disease so people should watch out if their current job type is a private job (mine is). This could be due the amount of stress that gets build up here, but more data is needed in this field for the future.

Overall, this program was very different by its nature considering that I needed to use the information provided in a structured data form (CDC data), use it to access a semi-structured data type (html), and then finally saving it into a structured data form (rankerDF). It was also the first time where I realized that it's really useful to create an "IF…ELSE" statement for checking if "rankerDF" actually exists or else I'd have to waste a minute each time I run the program. However, once all the information is collected and stored in as a pandas Dataframe, everything became very easy and familiar to work with again.