Leading Causes of Death in USA

1) Introduction

Data Management is a process of acquiring, validating, storing, manipulating and analysing data. It is an essential practise as we are creating and consuming data at unprecedented rates.

The objective of this project is to implement different Data Management techniques including exploratory data analysis, data cleaning and manipulation, in order to analyse the data and gather useful insights using 'Pandas' package available in Python to answer analytical questions related to the data.

2) Data

We are using two publically available datasets for this project, containing data related to population and causes of death in USA.

1) NCHS_-_Leading_Causes_of_Death__United_States
2) nst-est2018-01

The first dataset contains information on the number of deaths and its causes in USA. It has national as well as state level information for the period 1999 to 2016. The second datasets contains the information on the population of USA as a whole, population data for each state and each of four census regions for the period 2010 to 2018.

3) Research Questions

We are going to address the following questions in this project.

1) Are Americans facing increasing, decreasing or study likelihood of death?
2) What are the four leading causes of deaths for Americans?
3) Do individual states show the same four leading causes of death?
4) Are there year by year changes in the four leading causes of deaths nationwise?

4) Code, Explanation and Analysis

Preliminary steps:

Importing the packages:

In [1]:
```
import pandas as pd
import numpy as np
```

Reading the first data file, converting it into pandas dataframe and displaying the first 2 observations.

In [2]:
```
mydir = 'C:/RIT/Spring 2021\BANA 680 - Data Management for Business Analytics/Assignmen
file = mydir + 'NCHS_-_Leading_Causes_of_Death__United_States.csv'
```

```
NCHS = pd.read_csv(file)

#Convert the file to dataframe
NCHS_df = pd.DataFrame(NCHS)

#Read first 2 records
NCHS_df.head(2)
```

Out[2]:

	Year	113 Cause Name	Cause Name	State	Deaths	Age-adjusted Death Rate
0	2012	Nephritis, nephrotic syndrome and nephrosis (N...	Kidney disease	Vermont	21	2.6
1	2016	Nephritis, nephrotic syndrome and nephrosis (N...	Kidney disease	Vermont	30	3.7

In order to understand the data structure, its size and variables, we execute the following commands:

In [3]:
```
print('Data Structure:', '\n', NCHS_df.dtypes)
print('\n', 'Count of Rows and Columns:', '\n', NCHS_df.shape)
print('\n', 'Name of columns:', '\n', list(NCHS_df))
```

```
Data Structure:
 Year                        int64
113 Cause Name              object
Cause Name                  object
State                       object
Deaths                       int64
Age-adjusted Death Rate    float64
dtype: object

 Count of Rows and Columns:
 (10296, 6)

 Name of columns:
 ['Year', '113 Cause Name', 'Cause Name', 'State', 'Deaths', 'Age-adjusted Death Rate']
```

To further check the uniques values for the columns and check if there are any null or missing values, following commands are executed.

In [4]:
```
print('Unique Years:', NCHS_df.Year.nunique())
print('Unique States:', NCHS_df.State.nunique())
print('Unique Causes of Death:', len(pd.unique(NCHS_df['Cause Name'])))

print('\n', 'Count of null or missing values for each column:', '\n', NCHS_df.isnull().
```

```
Unique Years: 18
Unique States: 52
Unique Causes of Death: 11

 Count of null or missing values for each column:
 Year                       0
113 Cause Name              0
Cause Name                  0
State                       0
Deaths                      0
Age-adjusted Death Rate     0
dtype: int64
```

Reading the second data file relating to population data and converting it into Pandas dataframe.

On displaying the dataframe, it is seen that it is quite messy and requires cleaning and adjustments in order to use it. We will do that later, as and when required.

In [5]:
```
file2 = mydir + 'nst-est2018-01.xlsx'
pop_data = pd.read_excel(file2)

pop_df = pd.DataFrame(pop_data)
#pop_df.head(3)
```

Q1: Are Americans face increasing, decreasing or steady likelihood of death?

For this problem, we need to look at the yearly death count for USA as whole and ignore the data for individual states. Also, since the population differs for each year, we need to standarized the death count for each year in order to be compared fairly.

Steps:

1) We will consider the first dataframe for the death causes i.e. NCHS_df. On observing the df, we can see that it has already aggregated the number of deaths and it's causes for USA as a whole, along with state wise count. So we will just extract the records with 'United States' as string in column 'State' as it represents the data for USA as whole.

2) Next we need to exclude the records for 'All causes' in the column 'Cause Name' as they are the aggregate for all causes of deaths for all states. So we will just keep records for 10 causes of death for United States.

3) We need the total number of deaths per year in order to analyze the death trend over the years.

4) We know that population differs for each year. So in order to have fair comparision of death count over the years, we need to standarized the death variable. For this, we will extract the population data from the second df i.e. pop_df and use it to normalize the death count over the years.

5) The final goal to analyze the death trend is to compute the number of deaths per year and then normalize it by dividing it by population for the year so that the effect of population is ignored.

In [6]:
```
# creating a data frame with state - US .
df_US = NCHS_df[NCHS_df['State'].isin(['United States'])]

# From the above df, creating a new df with just 'All causes' in column 'Cause Name'
drop_cause = df_US['Cause Name'].isin(['All causes'])

#Filtered dataframe. Contains records for 10 causes of death for United States for the
df_US_f = (df_US[-drop_cause])
```

We can check if dataframe, for the unique states and death causes by executing the following command

In [7]:
```
print('Unique State:', df_US_f.State.nunique())
print('Unique causes of death:', len(pd.unique(df_US_f['Cause Name'])))
```

```
Unique State: 1
```

```
Unique causes of death: 10
```

2) Next step is to compute the total number of deaths per year. For this we will use groupby function to group the df by year and then agg function to get the total number of deaths for each year. The output will contain a record for the number of deaths for each year regardless of the causes for USA.

In [8]:
```
US_totdeaths = df_US_f.groupby('Year').agg({'Deaths': sum})
US_totdeaths.head(2) #Printing out first 2 records to verify the output
```

Out[8]:

Year	Deaths
1999	1905826
2000	1902194

3) The structure of the population dataset is: each year as different column and the population for states as rows. Now in our filtered df, we have a single column for Years. So we need to extract the population for each year for the USA as whole, in a vertical form in order to be consistent with our df structure. We will execute the following command for the same. Here we are not concern with the index values, as it is just an intermediary step.

In [9]:
```
pop_us = pd.DataFrame({'Year' : pop_df.iloc[2, 3:], 'Population' : pop_df.iloc[3, 3:]})
pop_us.head(2) # printing first 2 records to verify the output.
```

Out[9]:

	Year	Population
Unnamed: 3	2010	309326085
Unnamed: 4	2011	3.1158e+08

4) Merging the 2 dataframes using inner join on Year, since that is the common column in both the dataframes. Also, there is a difference in number of years in both the files. We are therefore using 'inner join' to keep the records common in both the files, as only those could be used for proper analysis. Further we add a new column to the merged df by calculating death per million to normalize the death variable for better comparision.

In [10]:
```
merged_us = pd.merge(US_totdeaths, pop_us, on = 'Year', how = 'inner')

# Adding new column DPM
merged_us['DPM'] = (merged_us['Deaths']/merged_us['Population'])*1000000
merged_us
```

Out[10]:

	Year	Deaths	Population	DPM
0	2010	1852349	309326085	5988.34
1	2011	1869321	3.1158e+08	5999.49
2	2012	1876588	3.13874e+08	5978.79
3	2013	1910311	3.16058e+08	6044.18
4	2014	1938408	3.18386e+08	6088.22

	Year	Deaths	Population	DPM
5	2015	2013017	3.20743e+08	6276.11
6	2016	2034119	3.23071e+08	6296.19

Conclusion:

From the above output we can see that death rate per million is gradually increasing from 2010 to 2016. So we can conclude that the Americans are facing increasing likelihood of death.

Q2: Four leading causes of death for Americans?

Here again we are dealing with the records for US as whole. The individual state records are ignored. We need to extract the four leading causes of death for US.

- For this purpose, we require a df having 'State' as only 'United States' and exclude the records for 'All causes' in column: 'Cause Name'. We will use the already created the df for this while addressing the Q1 above i.e. df_US_f.
- Next we need to group the data by column: 'Cause Name' and aggregate the total number of deaths for each cause over the years. Once this is done, we can extract the 4 causes having the highest death count for US.
- The final output will contain the 4 leading causes for death for US and the total count of death for each cause.

In [16]:
```
#Applying groupby and agg functions on df_US_f to get total number of deaths for
#each cause for US and extracting top 4 causes
death_cause_US = df_US_f.groupby('Cause Name').agg({'Deaths':
                                                    sum})['Deaths'].nlargest(10).reset_

death_cause_US #Printing the output to verify
```

Out[16]:

	Cause Name	Deaths
0	Heart disease	11575183
1	Cancer	10244536
2	Stroke	2580140
3	CLRD	2434726
4	Unintentional injuries	2177884
5	Alzheimer's disease	1373412
6	Diabetes	1316379
7	Influenza and pneumonia	1038969
8	Kidney disease	807980
9	Suicide	649843

Conclusion:

The four leading causes of deaths for United States over the years are 'Heart disease', 'Cancer', 'Stroke' and 'CLRD'.

Q3: Do individual states show the same four leading causes of death?

Here we are required to determine if the individual states have the same leading causes of death as that for US as whole which we determined in Q2 above. Since there is no mention of the order of the leading causes of death, we are assuming that the order does not matter as long as the top causes of death are the same.

Steps:

1) Dataframe required here should not contain the records for US as whole in 'State' column and 'All causes' of death in 'Cause Name' column.

2) We then need to determine the top four leading causes of death for each state and then compare it with the 4 leading causes of death for US as whole over the years.

3) Determine if causes of death for individual states are same as that for country US.

1) Creating the required dataframe and verifying it.

In [12]:
```
#Using original df, creating a new df with 'State' column not containing 'United States
alls = NCHS_df[NCHS_df['State'].isin(['United States']) == False]

#Using the above alls df,creating a new df with'Cause Name' column not containing 'All
allstates = alls[alls['Cause Name'].isin(['All causes']) == False]

#Verifying the df allstates
print('Is United State in State?', allstates['State'].isin(['United States']).any())
print('Is All causes in Cause Name?', allstates['Cause Name'].isin(['All causes']
                                                                   ).any())
```

```
Is United State in State? False
Is All causes in Cause Name? False
```

2) Next we need to determine top 4 causes of death for each state and compare it with top 4 death cause for US. For this we use for loop and ifelse statement.

- We create a new df (state_grp) from earlier df (allstates) by grouping it on 'State' column.
- Then in a 'for loop', we use 'get_group' function for the groups in df: state_grp and save it in a new variable: 'states'. This will save all the 51 states in the df in 'states' variable so that it is looped by 'states'.
- We further group the df: 'states' by 'Cause Name' column and aggregate the 'Deaths' over the years and also extract the 4 highest leading causes of death for each state and store it in a new variable - 'top4causes'.
- Then inside the 'for loop' we use a 'ifelse' statement to compare the list of death causes saved in 'top4causes' with the death causes for US determined earlier in 'death_cause_US'. If the 2 lists contain same causes in any order, the result is saved as true.

- Finally we can print out the count of states having same causes of death and find the percentage to analyse.

In [13]:
```
state_grp = allstates.groupby(['State'])
result =[]
for State in state_grp.groups:
    states = state_grp.get_group(State)
    top4causes = states.groupby(['Cause Name']).agg({'Deaths':
                                                     sum})['Deaths'].nlargest(4).reset_
    if set(death_cause_US['Cause Name']) == set(top4causes['Cause Name']):
        result.append(True)
    else:
        result.append(False)

print('Number of states compared with US as whole, for top 4 leading causes of death:',
      len(result))
print('Total states have same leading death causes as that of US:',sum(result))
print('Percentage of states having same 4 leading death causes as that of US a whole:',
      (round(((sum(result))/(len(result)))*100)), '%')
```

```
Number of states compared with US as whole, for top 4 leading causes of death: 51
Total states have same leading death causes as that of US: 30
Percentage of states having same 4 leading death causes as that of US a whole: 59 %
```

Conclusion:

From the above output, we can see that out of total 51 states, 30 states have the same four leading causes of death as that of country US. i.e. 59% of states have the same 4 leading causes of death as that of US while 41% of states have different leading causes of death for all the years in aggregate.

Q4: Are there year by year changes in the four leading causes of death nationwide?

Here we are required to determine if the top four causes of death for US for each year are same as compared to the top four causes of death for US over the years.

Steps:

1) We require a dataframe containing records for only 'United States' in 'State' column and exculde the records having 'All causes' in 'Cause Name' column. Since we have already created the df for this earlier, we will use the same df i.e. df_US_f here.

2) Next we have to determine the top 4 causes of death for each year and compare with the top 4 causes for death for US for all years. For this, we will follow the same steps we did in Q3 part (2) above except instead of using groupby on column: 'Cause Name' as in Q3, here we will use groupby on column: 'Year'.

In [14]:
```
# groupby year and agg death
year_grp = df_US_f.groupby('Year')
output = []
year_list = []

for Year in year_grp.groups:
    year = year_grp.get_group(Year)
    top_causes = year.groupby(['Cause Name']).agg({'Deaths': sum})['Deaths'].nlargest(4
```

```
    if set(death_cause_US['Cause Name']) == set(top_causes['Cause Name']):
        (output.append(True))
    else:
        output.append(False)

print('Number of years compared for the top 4 leading causes of death with that of aggr
      len(output))
print('Number of years having same leading death causes as that of US for all the years
print('Percentage of years having same leading death causes as that of US for all years
      (round(((sum(output))/(len(output)))*100)),'%')
```

```
Number of years compared for the top 4 leading causes of death with that of aggregate of
all years for US: 18
Number of years having same leading death causes as that of US for all the years in aggr
egate: 14
Percentage of years having same leading death causes as that of US for all years in aggr
egate: 78 %
```

Conclusion:

From the above analysis, we can see that 14 out of 18 years compared, have same 4 leading causes of death for US yearwise as well as for aggregate of all years. I.e. 78% of the years compared have same top 4 causes of death. Also, if analyse the output further by printing the list of top 4 causes of deaths for each year for US, we can see that the causes of death have started varying slightly in recent years.

5) Summary

- In this project, we implemented data management techniques using Pandas on the two datasets, analysed the outputs and extracted valueable insights.
- Based on the analysis, we tried to addressed the questions stated and concluded that Americans are facing gradual increase in likelihood of deaths.
- We determined the top 4 causes of deaths faced by Americans nationwide as well as causes for individual states and year-wise.
- Finally concluded that for 78% of the years compared, the top 4 causes of death are same compared to that over the years while 30% of the states have same top causes of death as that for the entire nation.