Data Analytics in Healthcare

July 4, 2024

1 Import Libraries

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Set visualization styles
sns.set(style="whitegrid")
```

2 Load the Data

[3]:
```
df = pd.read_csv("D:\\new project\\dataset.csv")

df.head()
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_14024\1151330334.py:1: DtypeWarning:
Columns (10) have mixed types. Specify dtype option on import or set
low_memory=False.
  df = pd.read_csv("D:\\new project\\dataset.csv")
```

[3]:

	YearStart	YearEnd	LocationAbbr	LocationDesc	DataSource	Topic	\
0	2014	2014	AR	Arkansas	SEDD; SID	Asthma	
1	2018	2018	CO	Colorado	SEDD; SID	Asthma	
2	2018	2018	DC	District of Columbia	SEDD; SID	Asthma	
3	2017	2017	GA	Georgia	SEDD; SID	Asthma	
4	2010	2010	MI	Michigan	SEDD; SID	Asthma	

	Question	Response	DataValueUnit	DataValueType	…	\
0	Hospitalizations for asthma	NaN	NaN	Number	…	
1	Hospitalizations for asthma	NaN	NaN	Number	…	
2	Hospitalizations for asthma	NaN	NaN	Number	…	
3	Hospitalizations for asthma	NaN	NaN	Number	…	
4	Hospitalizations for asthma	NaN	NaN	Number	…	

	LocationID	TopicID	QuestionID	DataValueTypeID	StratificationCategoryID1	\
0	5	AST	AST3_1	NMBR	GENDER	

```
1          8      AST     AST3_1            NMBR                    OVERALL
2         11      AST     AST3_1            NMBR                    OVERALL
3         13      AST     AST3_1            NMBR                     GENDER
4         26      AST     AST3_1            NMBR                       RACE

   StratificationID1 StratificationCategoryID2 StratificationID2  \
0               GENM                        NaN               NaN
1                OVR                        NaN               NaN
2                OVR                        NaN               NaN
3               GENF                        NaN               NaN
4                HIS                        NaN               NaN

   StratificationCategoryID3  StratificationID3
0                        NaN                NaN
1                        NaN                NaN
2                        NaN                NaN
3                        NaN                NaN
4                        NaN                NaN

[5 rows x 34 columns]
```

3 Initial Data Exploration

[3]:
```python
# Display first few rows
print(df.head())

# Display last few rows
print(df.tail())

# Get DataFrame information
print(df.info())

# Get descriptive statistics
print(df.describe())

# Get column names
print(df.columns)

# Get shape of DataFrame
print(df.shape)
```

```
   YearStart  YearEnd LocationAbbr          LocationDesc DataSource   Topic  \
0       2014     2014           AR              Arkansas  SEDD; SID  Asthma
1       2018     2018           CO              Colorado  SEDD; SID  Asthma
2       2018     2018           DC  District of Columbia  SEDD; SID  Asthma
3       2017     2017           GA               Georgia  SEDD; SID  Asthma
4       2010     2010           MI              Michigan  SEDD; SID  Asthma
```

```
                     Question  Response DataValueUnit DataValueType  …  \
0  Hospitalizations for asthma       NaN           NaN        Number  …
1  Hospitalizations for asthma       NaN           NaN        Number  …
2  Hospitalizations for asthma       NaN           NaN        Number  …
3  Hospitalizations for asthma       NaN           NaN        Number  …
4  Hospitalizations for asthma       NaN           NaN        Number  …

  LocationID  TopicID QuestionID DataValueTypeID  StratificationCategoryID1  \
0          5      AST     AST3_1            NMBR                     GENDER
1          8      AST     AST3_1            NMBR                    OVERALL
2         11      AST     AST3_1            NMBR                    OVERALL
3         13      AST     AST3_1            NMBR                     GENDER
4         26      AST     AST3_1            NMBR                       RACE

   StratificationID1 StratificationCategoryID2 StratificationID2  \
0               GENM                       NaN               NaN
1                OVR                       NaN               NaN
2                OVR                       NaN               NaN
3               GENF                       NaN               NaN
4                HIS                       NaN               NaN

   StratificationCategoryID3  StratificationID3
0                        NaN                NaN
1                        NaN                NaN
2                        NaN                NaN
3                        NaN                NaN
4                        NaN                NaN

[5 rows x 34 columns]
         YearStart  YearEnd LocationAbbr  LocationDesc DataSource  \
1185671       2020     2020           WY       Wyoming      BRFSS
1185672       2020     2020           WY       Wyoming      BRFSS
1185673       2017     2017           IA          Iowa      BRFSS
1185674       2020     2020           WY       Wyoming      BRFSS
1185675       2019     2019           RI  Rhode Island      BRFSS

                Topic                                          Question  \
1185671      Diabetes  Dilated eye examination among adults aged >= 1…
1185672  Older Adults  Proportion of older adults aged >= 65 years wh…
1185673     Arthritis  Activity limitation due to arthritis among adu…
1185674      Diabetes   Diabetes prevalence among women aged 18-44 years
1185675     Arthritis  Activity limitation due to arthritis among adu…

         Response DataValueUnit            DataValueType  … LocationID  \
1185671       NaN             %  Age-adjusted Prevalence  …         56
1185672       NaN             %         Crude Prevalence  …         56
1185673       NaN             %  Age-adjusted Prevalence  …         19
```

```
1185674       NaN             %         Crude Prevalence  …         56
1185675       NaN             %         Crude Prevalence  …         44

         TopicID QuestionID DataValueTypeID  StratificationCategoryID1  \
1185671      DIA     DIA7_0      AGEADJPREV                       RACE
1185672      OLD     OLD3_1         CRDPREV                       RACE
1185673      ART     ART2_1      AGEADJPREV                       RACE
1185674      DIA     DIA2_2         CRDPREV                       RACE
1185675      ART     ART2_1         CRDPREV                    OVERALL

         StratificationID1 StratificationCategoryID2 StratificationID2  \
1185671               WHT                        NaN               NaN
1185672               WHT                        NaN               NaN
1185673               HIS                        NaN               NaN
1185674               HIS                        NaN               NaN
1185675               OVR                        NaN               NaN

         StratificationCategoryID3  StratificationID3
1185671                        NaN                NaN
1185672                        NaN                NaN
1185673                        NaN                NaN
1185674                        NaN                NaN
1185675                        NaN                NaN

[5 rows x 34 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1185676 entries, 0 to 1185675
Data columns (total 34 columns):
 #   Column                     Non-Null Count    Dtype
---  ------                     --------------    -----
 0   YearStart                  1185676 non-null  int64
 1   YearEnd                    1185676 non-null  int64
 2   LocationAbbr               1185676 non-null  object
 3   LocationDesc               1185676 non-null  object
 4   DataSource                 1185676 non-null  object
 5   Topic                      1185676 non-null  object
 6   Question                   1185676 non-null  object
 7   Response                   0 non-null        float64
 8   DataValueUnit              1033553 non-null  object
 9   DataValueType              1185676 non-null  object
 10  DataValue                  806942 non-null   object
 11  DataValueAlt               804578 non-null   float64
 12  DataValueFootnoteSymbol    393710 non-null   object
 13  DatavalueFootnote          393710 non-null   object
 14  LowConfidenceLimit         682380 non-null   float64
 15  HighConfidenceLimit        682380 non-null   float64
 16  StratificationCategory1    1185676 non-null  object
 17  Stratification1            1185676 non-null  object
```

```
 18  StratificationCategory2    0 non-null        float64
 19  Stratification2            0 non-null        float64
 20  StratificationCategory3    0 non-null        float64
 21  Stratification3            0 non-null        float64
 22  GeoLocation                1175510 non-null  object
 23  ResponseID                 0 non-null        float64
 24  LocationID                 1185676 non-null  int64
 25  TopicID                    1185676 non-null  object
 26  QuestionID                 1185676 non-null  object
 27  DataValueTypeID            1185676 non-null  object
 28  StratificationCategoryID1  1185676 non-null  object
 29  StratificationID1          1185676 non-null  object
 30  StratificationCategoryID2  0 non-null        float64
 31  StratificationID2          0 non-null        float64
 32  StratificationCategoryID3  0 non-null        float64
 33  StratificationID3          0 non-null        float64
dtypes: float64(13), int64(3), object(18)
memory usage: 307.6+ MB
None
          YearStart       YearEnd  Response  DataValueAlt  LowConfidenceLimit  \
count  1.185676e+06  1.185676e+06       0.0  8.045780e+05       682380.000000
mean   2.015103e+03  2.015643e+03       NaN  1.005325e+03           50.264623
std    3.320259e+00  3.001197e+00       NaN  1.880433e+04           89.004848
min    2.001000e+03  2.001000e+03       NaN  0.000000e+00            0.000000
25%    2.013000e+03  2.013000e+03       NaN  1.610000e+01           11.000000
50%    2.015000e+03  2.016000e+03       NaN  4.000000e+01           28.500000
75%    2.018000e+03  2.018000e+03       NaN  7.600000e+01           56.300000
max    2.021000e+03  2.021000e+03       NaN  2.925456e+06         2541.600000

       HighConfidenceLimit  StratificationCategory2  Stratification2  \
count        682380.000000                      0.0              0.0
mean             61.873881                      NaN              NaN
std             100.104303                      NaN              NaN
min               0.000000                      NaN              NaN
25%              16.300000                      NaN              NaN
50%              41.000000                      NaN              NaN
75%              71.100000                      NaN              NaN
max            3530.500000                      NaN              NaN

       StratificationCategory3  Stratification3  ResponseID    LocationID  \
count                      0.0              0.0         0.0  1.185676e+06
mean                       NaN              NaN         NaN  3.078907e+01
std                        NaN              NaN         NaN  1.750972e+01
min                        NaN              NaN         NaN  1.000000e+00
25%                        NaN              NaN         NaN  1.700000e+01
50%                        NaN              NaN         NaN  3.000000e+01
75%                        NaN              NaN         NaN  4.500000e+01
max                        NaN              NaN         NaN  7.800000e+01
```

```
       StratificationCategoryID2  StratificationID2  \
count                        0.0                0.0
mean                         NaN                NaN
std                          NaN                NaN
min                          NaN                NaN
25%                          NaN                NaN
50%                          NaN                NaN
75%                          NaN                NaN
max                          NaN                NaN

       StratificationCategoryID3  StratificationID3
count                        0.0                0.0
mean                         NaN                NaN
std                          NaN                NaN
min                          NaN                NaN
25%                          NaN                NaN
50%                          NaN                NaN
75%                          NaN                NaN
max                          NaN                NaN
Index(['YearStart', 'YearEnd', 'LocationAbbr', 'LocationDesc', 'DataSource',
       'Topic', 'Question', 'Response', 'DataValueUnit', 'DataValueType',
       'DataValue', 'DataValueAlt', 'DataValueFootnoteSymbol',
       'DatavalueFootnote', 'LowConfidenceLimit', 'HighConfidenceLimit',
       'StratificationCategory1', 'Stratification1', 'StratificationCategory2',
       'Stratification2', 'StratificationCategory3', 'Stratification3',
       'GeoLocation', 'ResponseID', 'LocationID', 'TopicID', 'QuestionID',
       'DataValueTypeID', 'StratificationCategoryID1', 'StratificationID1',
       'StratificationCategoryID2', 'StratificationID2',
       'StratificationCategoryID3', 'StratificationID3'],
      dtype='object')
(1185676, 34)
```

[23]:
```
# List unique values in a specific column
print(df['Topic'].unique())
```

```
['Asthma' 'Cancer' 'Chronic Kidney Disease'
 'Chronic Obstructive Pulmonary Disease' 'Cardiovascular Disease'
 'Diabetes' 'Disability' 'Reproductive Health' 'Alcohol' 'Arthritis'
 'Tobacco' 'Nutrition, Physical Activity, and Weight Status'
 'Mental Health' 'Older Adults' 'Oral Health' 'Overarching Conditions'
 'Immunization']
```

4 Dealing with Missing Values

[4]:
```
# Check for missing values
df.isnull().sum()
```

[4]:
```
YearStart                       0
YearEnd                         0
LocationAbbr                    0
LocationDesc                    0
DataSource                      0
Topic                           0
Question                        0
Response                  1185676
DataValueUnit              152123
DataValueType                   0
DataValue                  378734
DataValueAlt               381098
DataValueFootnoteSymbol    791966
DatavalueFootnote          791966
LowConfidenceLimit         503296
HighConfidenceLimit        503296
StratificationCategory1         0
Stratification1                 0
StratificationCategory2   1185676
Stratification2           1185676
StratificationCategory3   1185676
Stratification3           1185676
GeoLocation                 10166
ResponseID                1185676
LocationID                      0
TopicID                         0
QuestionID                      0
DataValueTypeID                 0
StratificationCategoryID1       0
StratificationID1               0
StratificationCategoryID2 1185676
StratificationID2         1185676
StratificationCategoryID3 1185676
StratificationID3         1185676
dtype: int64
```

[4]:
```
# Drop columns with all missing values and rows with missing DataValue
df = df.drop(columns=['Response', 'StratificationCategory2', 'Stratification2',␣
 ↪'StratificationCategory3', 'Stratification3', 'ResponseID',␣
 ↪'StratificationCategoryID2', 'StratificationID2',␣
 ↪'StratificationCategoryID3', 'StratificationID3'])
df = df.dropna(subset=['DataValue'])
```

```
df['DataValue'] = pd.to_numeric(df['DataValue'], errors='coerce')
df = df.dropna(subset=['DataValue'])

# Convert YearStart and YearEnd to datetime type
df['YearStart'] = pd.to_datetime(df['YearStart'], format='%Y')
df['YearEnd'] = pd.to_datetime(df['YearEnd'], format='%Y')

# Handle missing values in LowConfidenceLimit and HighConfidenceLimit
df['LowConfidenceLimit'] = df['LowConfidenceLimit'].
 ↪fillna(df['LowConfidenceLimit'].mean())
df['HighConfidenceLimit'] = df['HighConfidenceLimit'].
 ↪fillna(df['HighConfidenceLimit'].mean())
```

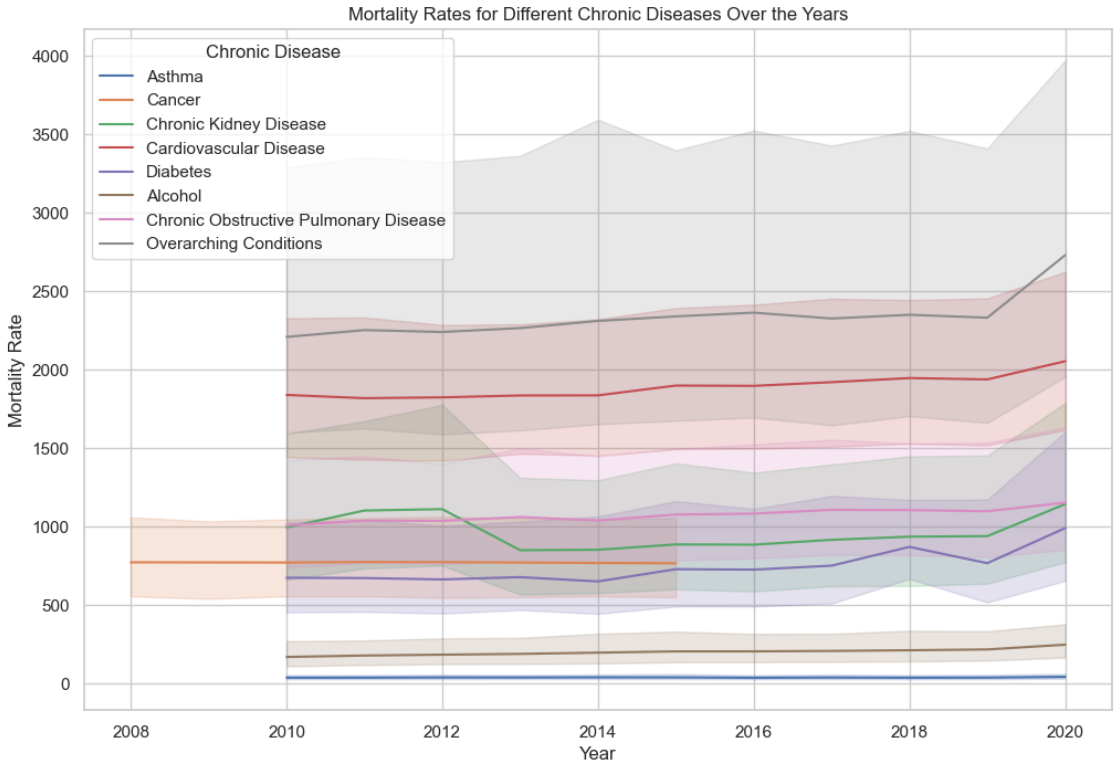
[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 804578 entries, 0 to 1185675
Data columns (total 24 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   YearStart                  804578 non-null  datetime64[ns]
 1   YearEnd                    804578 non-null  datetime64[ns]
 2   LocationAbbr               804578 non-null  object
 3   LocationDesc               804578 non-null  object
 4   DataSource                 804578 non-null  object
 5   Topic                      804578 non-null  object
 6   Question                   804578 non-null  object
 7   DataValueUnit              699340 non-null  object
 8   DataValueType              804578 non-null  object
 9   DataValue                  804578 non-null  float64
 10  DataValueAlt               804578 non-null  float64
 11  DataValueFootnoteSymbol    14976 non-null   object
 12  DatavalueFootnote          14976 non-null   object
 13  LowConfidenceLimit         804578 non-null  float64
 14  HighConfidenceLimit        804578 non-null  float64
 15  StratificationCategory1    804578 non-null  object
 16  Stratification1            804578 non-null  object
 17  GeoLocation                794819 non-null  object
 18  LocationID                 804578 non-null  int64
 19  TopicID                    804578 non-null  object
 20  QuestionID                 804578 non-null  object
 21  DataValueTypeID            804578 non-null  object
 22  StratificationCategoryID1  804578 non-null  object
 23  StratificationID1          804578 non-null  object
dtypes: datetime64[ns](2), float64(4), int64(1), object(17)
memory usage: 153.5+ MB
```

5 Mortality Rates for Different Chronic Diseases Over the Years

[5]:
```
# Filter the dataset for questions related to mortality
mortality_df = df[df['Question'].str.contains('mortality', case=False,␣
↪na=False)]
```

[48]:
```
plt.figure(figsize=(12, 8))
sns.lineplot(data=mortality_df, x='YearStart', y='DataValue', hue='Topic')
plt.title('Mortality Rates for Different Chronic Diseases Over the Years')
plt.xlabel('Year')
plt.ylabel('Mortality Rate')
plt.legend(title='Chronic Disease')
plt.show()
```



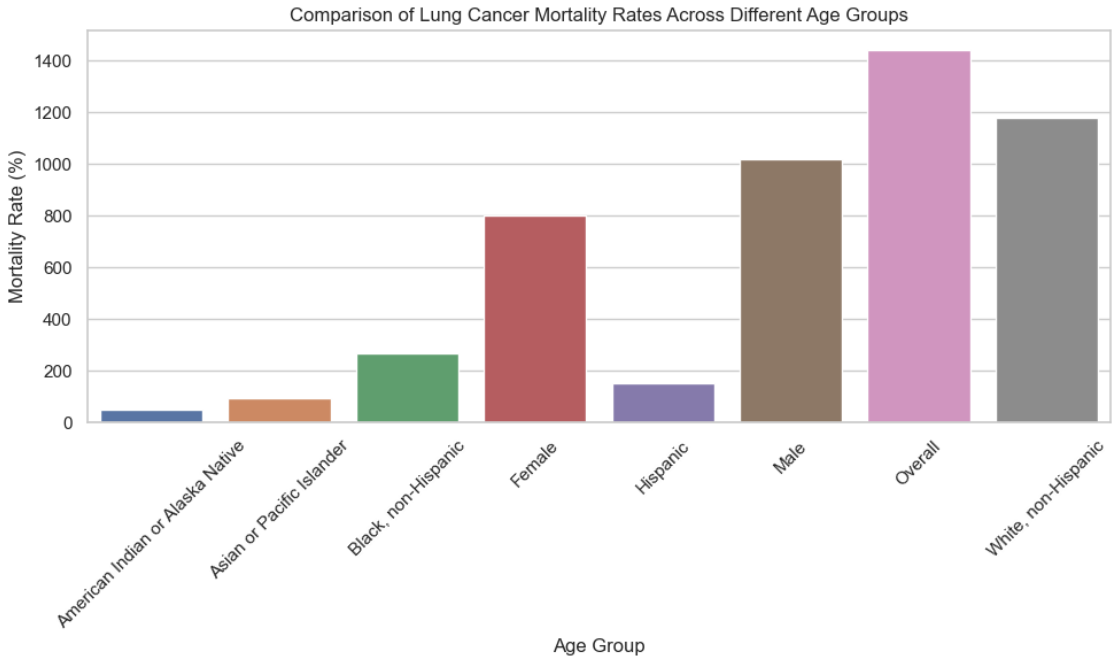
6 Geographic Variations in Cancer Mortality Rates

[8]:
```
# Filter data for lung cancer mortality
lung_cancer_mortality = mortality_df[mortality_df['Topic'] == 'Cancer']

# Group by age groups and calculate average mortality rates
```

```
mortality_by_cat = lung_cancer_mortality.
  ↪groupby('Stratification1')['DataValueAlt'].mean().reset_index()

# Plotting comparison of mortality rates between age groups
plt.figure(figsize=(10, 6))
sns.barplot(x='Stratification1', y='DataValueAlt', data=mortality_by_cat)
plt.title('Comparison of Lung Cancer Mortality Rates Across Different Age␣
  ↪Groups')
plt.xlabel('Age Group')
plt.ylabel('Mortality Rate (%)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



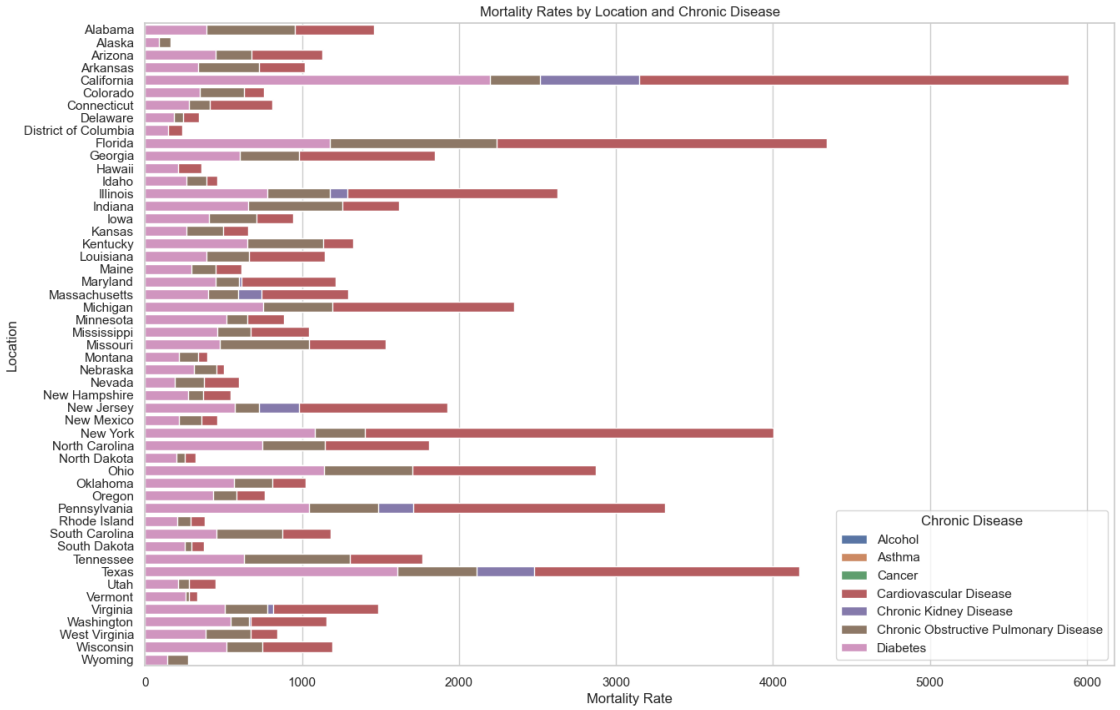
7 Disease Mortality Rates by Location

[6]:
```
# Ensure that we have relevant data for chronic diseases
relevant_conditions = ['Asthma', 'Cancer', 'Chronic Kidney Disease',
       'Cardiovascular Disease', 'Diabetes', 'Alcohol',
       'Chronic Obstructive Pulmonary Disease']
filtered_df = mortality_df[mortality_df['Topic'].isin(relevant_conditions)]

relevant_location = filtered_df[filtered_df['LocationDesc'] != 'United States']
```

```
# Aggregate data by location and chronic disease
location_mortality = relevant_location.groupby(['LocationDesc',␣
 ↪'Topic'])['DataValue'].mean().reset_index()

# Plot mortality rates by location
plt.figure(figsize=(15, 10))
sns.barplot(data=location_mortality, x='DataValue', y='LocationDesc',␣
 ↪hue='Topic', dodge=False)
plt.title('Mortality Rates by Location and Chronic Disease')
plt.xlabel('Mortality Rate')
plt.ylabel('Location')
plt.legend(title='Chronic Disease')
plt.show()
```

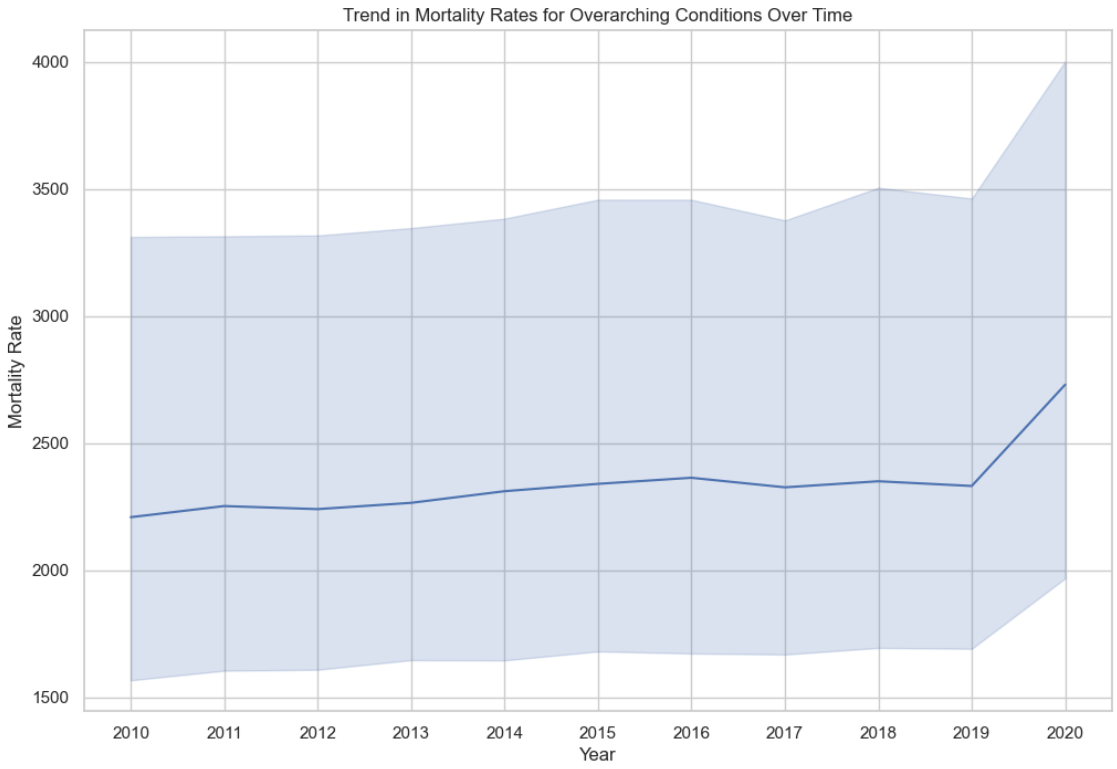


8 Trend in Mortality Rates for Overarching Conditions Over Time

[83]:
```
# Plot the trend in mortality rates for a specific disease (e.g., Diabetes)
diabetes_mortality = mortality_df[mortality_df['Topic'] == 'Overarching␣
 ↪Conditions']

plt.figure(figsize=(12, 8))
sns.lineplot(data=diabetes_mortality, x='YearStart', y='DataValue')
plt.title('Trend in Mortality Rates for Overarching Conditions Over Time')
```

```
plt.xlabel('Year')
plt.ylabel('Mortality Rate')
plt.show()
```

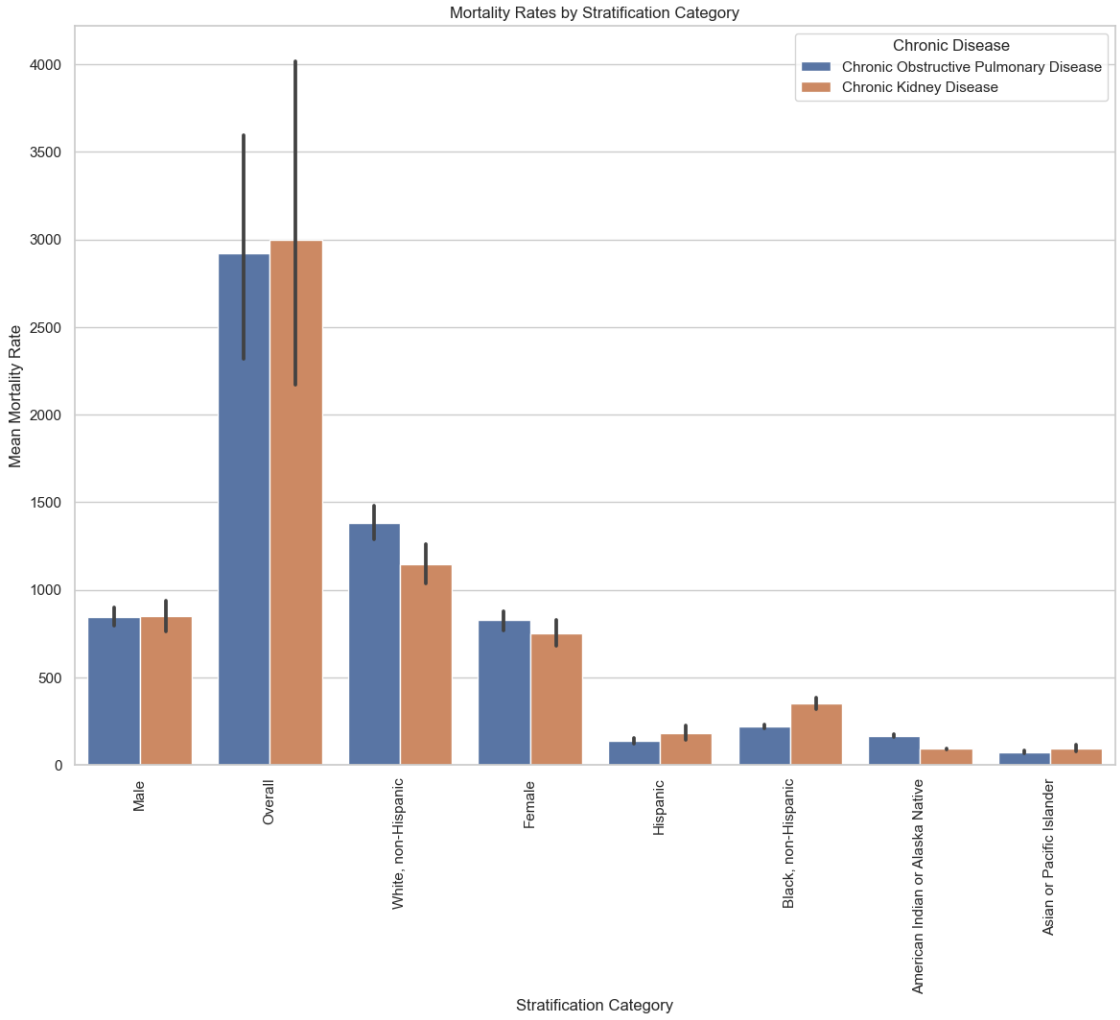


9 Mortality Rates by Stratification Category

[81]:
```
# Plot mortality rates by stratification categories (e.g., gender, race)

topic_order = ['Chronic Obstructive Pulmonary Disease', 'Chronic Kidney␣
 ↪Disease']

plt.figure(figsize=(14, 10))
sns.barplot(data=mortality_df, x='Stratification1', y='DataValue', hue='Topic',␣
 ↪hue_order=topic_order)
plt.title('Mortality Rates by Stratification Category')
plt.xlabel('Stratification Category')
plt.ylabel('Mean Mortality Rate')
plt.xticks(rotation=90)
plt.legend(title='Chronic Disease')
plt.show()
```

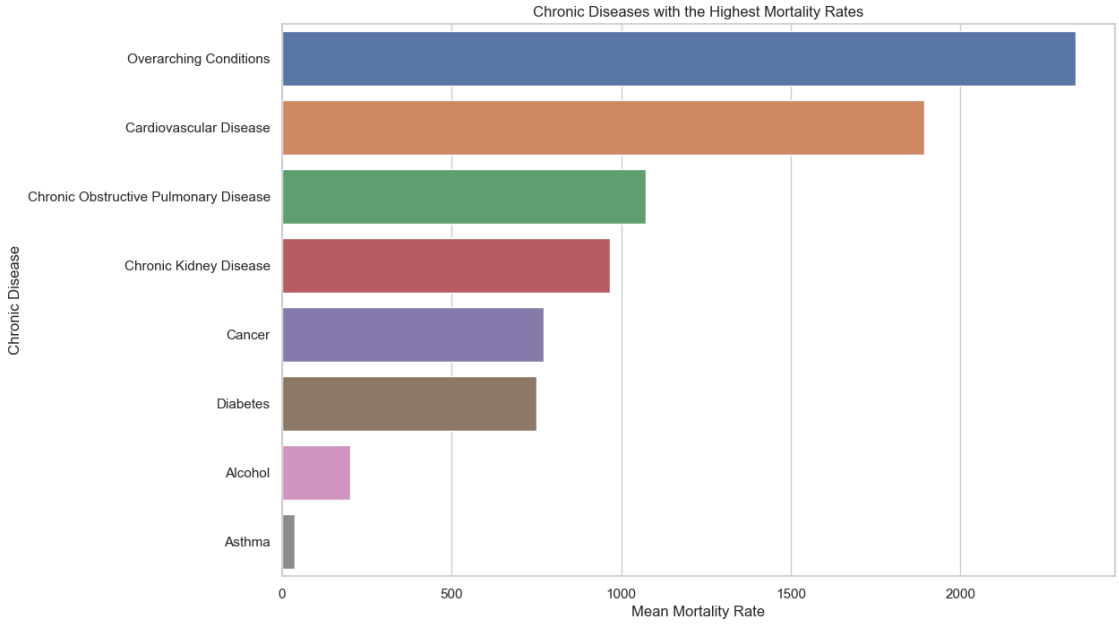


10 Chronic Diseases with the Highest Mortality Rates

[84]:
```
# Aggregate data to find the mean mortality rate for each chronic disease
mean_mortality = mortality_df.groupby('Topic')['DataValue'].mean().reset_index()

# Sort values to find the highest mortality rates
mean_mortality = mean_mortality.sort_values(by='DataValue', ascending=False)

# Plot the chronic diseases with the highest mortality rates
plt.figure(figsize=(12, 8))
sns.barplot(data=mean_mortality, x='DataValue', y='Topic')
plt.title('Chronic Diseases with the Highest Mortality Rates')
plt.xlabel('Mean Mortality Rate')
plt.ylabel('Chronic Disease')
plt.show()
```

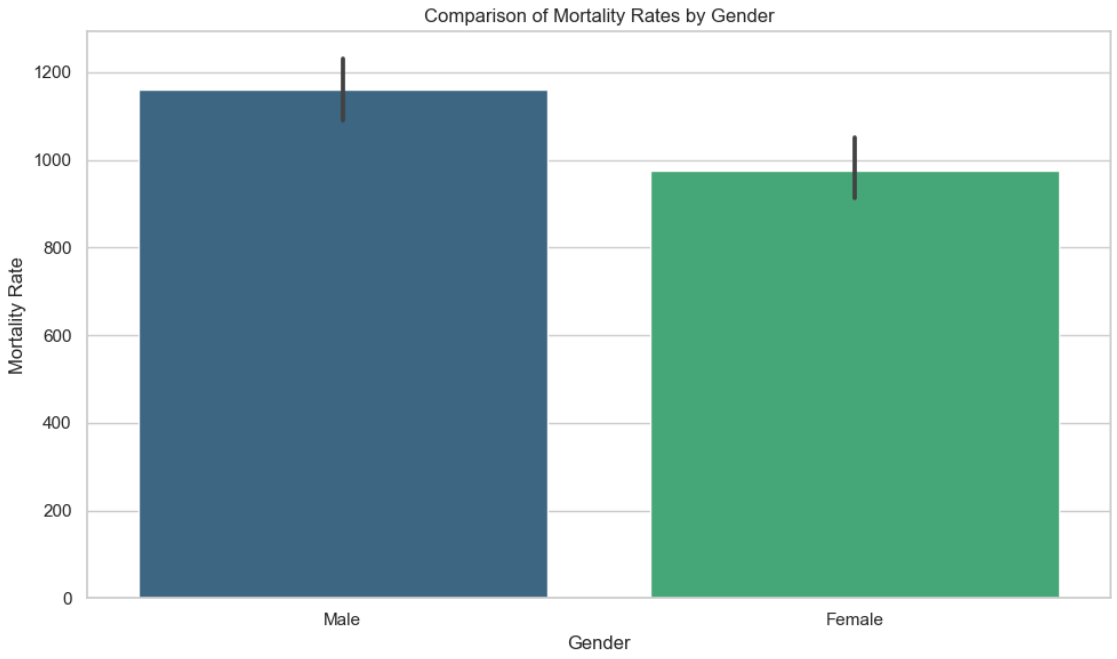


11 Comparison of Cancer Mortality Rates by Gender

[90]:
```
mortality_gender =  mortality_df[mortality_df['StratificationCategory1'] ==␣
 ↪'Gender']

# Plotting
plt.figure(figsize=(10, 6))
sns.barplot(x='Stratification1', y='DataValue', data=mortality_gender,␣
 ↪palette='viridis')
plt.title('Comparison of Mortality Rates by Gender')
plt.xlabel('Gender')
```

```
plt.ylabel('Mortality Rate')
plt.tight_layout()
plt.show()
```

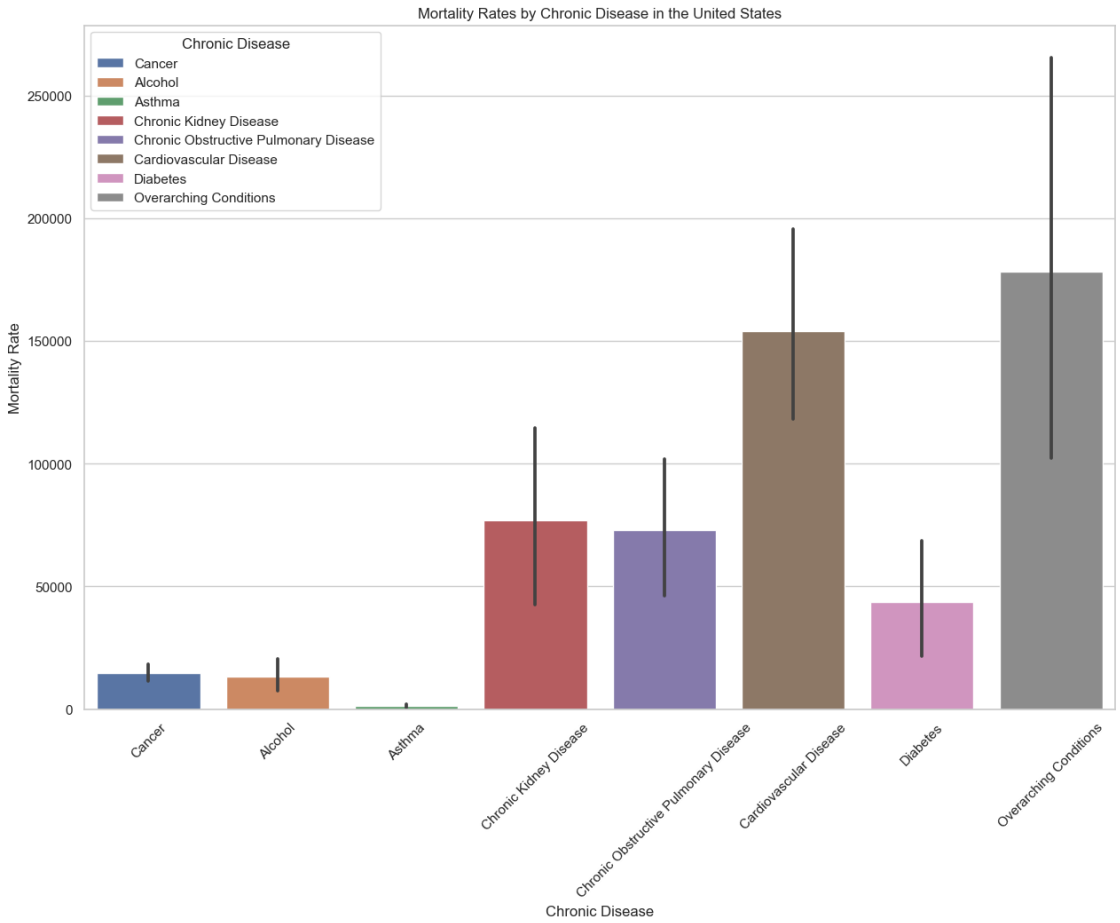


12 Mortality Rates by Chronic Disease in the United States

[12]:
```
# Filter data to include only United States
filtered_df_us = mortality_df[mortality_df['LocationDesc'] == 'United States']

# Set the figure size directly within sns.catplot
plt.figure(figsize=(15, 10))
sns.barplot(data=filtered_df_us, x='Topic', y='DataValue', hue='Topic',␣
 ↪dodge=False)
plt.title('Mortality Rates by Chronic Disease in the United States')
plt.xlabel('Chronic Disease')
plt.ylabel('Mortality Rate')
plt.legend(title='Chronic Disease', loc='upper left')
plt.xticks(rotation=45)

# Show the plot
plt.show()
```



13 Correlation Matrix of Mortality Rates for Chronic Diseases

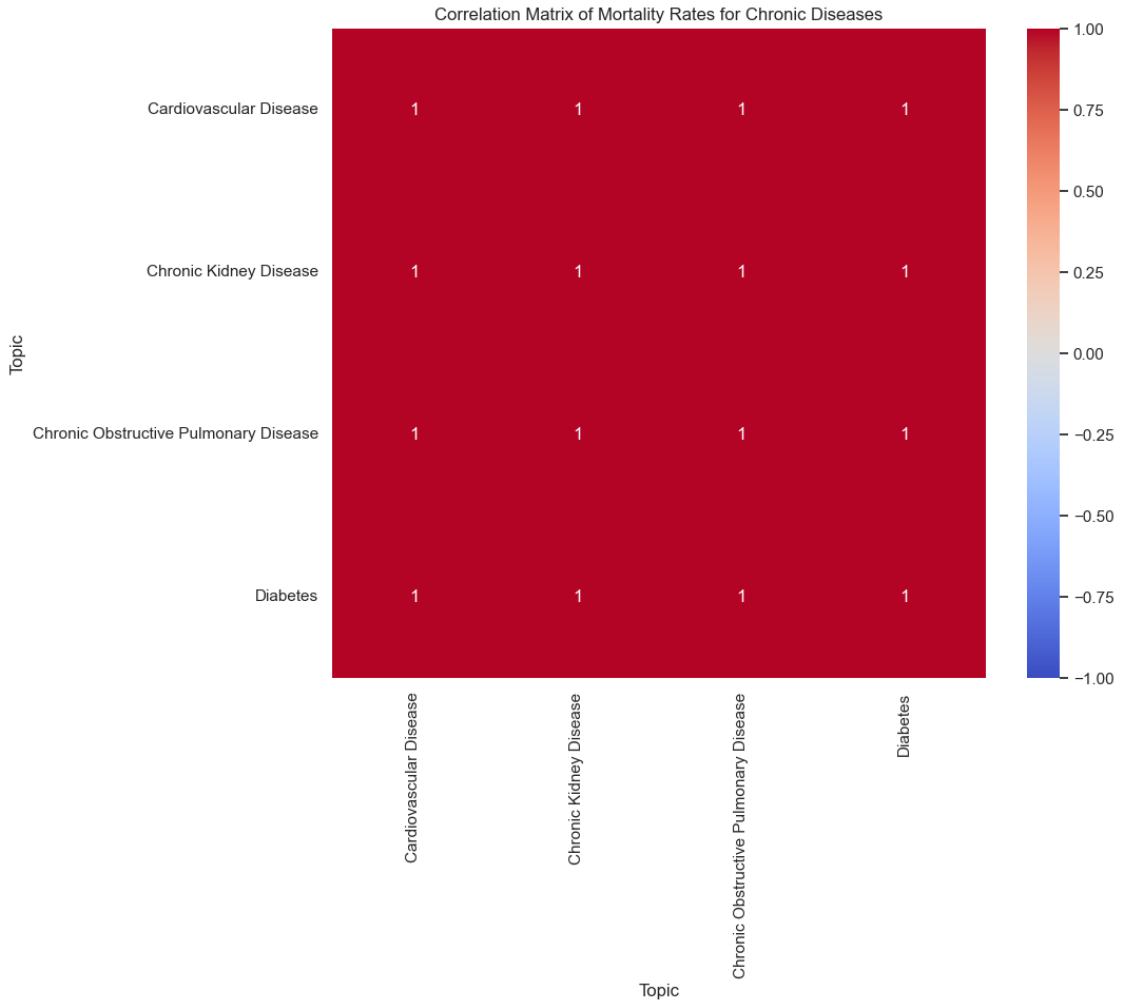
[9]:
```
# Filter dataset for mortality rates of various chronic diseases
chronic_diseases = ['Diabetes', 'Cardiovascular Disease', 'Chronic Obstructive␣
 ↪Pulmonary Disease',
                    'Chronic Kidney Disease']
chronic_mortality = mortality_df[mortality_df['Topic'].isin(chronic_diseases)]

# Pivot the dataset
pivot_df = chronic_mortality.pivot_table(index='LocationDesc', columns='Topic',␣
 ↪values='DataValue', aggfunc=np.mean)

# Calculate the correlation matrix
correlation_matrix = pivot_df.corr()

# Plot the heatmap
plt.figure(figsize=(10, 8))
```

```
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix of Mortality Rates for Chronic Diseases')
plt.show()
```



Based on our comprehensive analysis, we observed the following: - **Missing Values:** Handled missing values appropriately to ensure the integrity of the dataset. - **Outliers:** Identified and analyzed outliers using boxplots and Z-score method. - **Data Preparation:** Performed necessary data preparation steps such as date conversion and feature engineering. - **EDA:** Conducted exploratory data analysis to understand the data distribution, relationships, and trends. - **Visualization:** Created various visualizations to present the findings effectively to stakeholders.

[ ]: