FinalProjectReportQ1Q2Q3Q4Q5

December 15, 2024

0.1 GROUP 1

Team Members: Ramya, Mrudulahari, Akhil Richard

0.1.1 Beyond the Smoke: An Analytical Journey Through U.S. Tobacco Trends

Tobacco use continues to be one of the most significant public health challenges in the United States. Despite declines in smoking rates over the years, the increasing prevalence of e-cigarette consumption, particularly among younger demographics, has introduced new and complex health risks. This study aims to address these concerns by analyzing tobacco use and its associated health impacts using two critical datasets: the Centers for Disease Control and Prevention's (CDC) State Tobacco Activities Tracking and Evaluation (STATE) System (2011–2019) and data from the Food and Drug Administration (FDA) related to e-cigarette health risks.

The CDC dataset offers a detailed and comprehensive view of tobacco use patterns across various demographic groups. It includes data on both cigarette and e-cigarette usage rates, quit attempts, and the impact of factors like age, gender, race, and education level on tobacco consumption. These demographic insights help highlight significant disparities in tobacco use, particularly in vulnerable populations, and provide a clear foundation for the development of targeted, evidence-based public health strategies.

In addition, the FDA dataset provides crucial case-specific data on severe health conditions directly linked to e-cigarette use, including acute respiratory failure, nonconvulsive status epilepticus, and other serious health complications. This dataset is invaluable in shedding light on the specific health consequences of e-cigarette usage, offering a more nuanced understanding of the emerging risks posed by vaping. By combining the datasets from the CDC and FDA, this project takes a comprehensive approach to understanding the full scope of tobacco-related health issues.

Ultimately, this analysis aims to uncover trends and disparities in tobacco use and its associated health consequences across diverse demographic groups. The insights gained will provide actionable recommendations for public health interventions and tobacco control policies, particularly in addressing the needs of high-risk populations. The findings from this research have the potential to significantly inform future efforts aimed at reducing tobacco use and improving public health outcomes in the United States.

0.2 Datasets Used

0.2.1 1) Behavioral Risk Factor Data – Tobacco Use (2011–Present)

Link to Dataset
- **Source**: Centers for Disease Control and Prevention (CDC)

- **Description**: This dataset is derived from the CDC's Behavioral Risk Factor Surveillance System (BRFSS), which provides data on health-related risk behaviors, chronic health conditions, and the use of preventive services. It includes information on tobacco consumption (cigarettes, e-cigarettes, smokeless tobacco) across all 50 states in the U.S. from 2011 to the present.
- **Relevance**: This dataset is crucial for understanding the overall patterns of tobacco use across different demographic groups (age, gender, race, education, etc.). By analyzing these patterns, the project aims to uncover trends, disparities, and correlations between tobacco consumption and various demographic factors.

0.2.2 2) FDA Tobacco Health Problems Dataset

Link to Dataset
- **Source**: U.S. Food and Drug Administration (FDA)
- **Description**: The FDA provides an open API that contains case-level reports on health problems linked to tobacco consumption, particularly focusing on the risks associated with e-cigarettes. This includes severe health outcomes such as acute respiratory failure, nonconvulsive status epilepticus, and other serious conditions.
- **Relevance**: This dataset plays a pivotal role in understanding the health consequences tied to tobacco use, especially e-cigarettes. By correlating health data with tobacco consumption behaviors, this dataset will help in identifying the emerging risks of vaping and support the creation of targeted public health interventions.

0.2.3 3) Influence of Federal Government and State Policies on Cigarettes

Link to Custom Report
- **Source**: Centers for Disease Control and Prevention (CDC)
- **Description**: This dataset consists of a custom report generated by the CDC's State Tobacco Activities Tracking and Evaluation (STATE) System. It focuses on the impact of state and federal policies on tobacco use, specifically the effects of tobacco taxes and sales regulations on cigarette consumption. The report also evaluates the influence of government policies aimed at reducing tobacco use, including advertising restrictions, smoking bans, and educational initiatives.
- **Relevance**: By analyzing how government policies and regulations influence tobacco usage, this dataset helps contextualize how external factors, such as state-level interventions, affect smoking patterns across various states. It adds an important layer to understanding the broader socio-political and economic forces that contribute to tobacco consumption.

0.2.4 4) Proportion of Adults Who Are Current Smokers

Link to Dataset
- **Source**: California Department of Public Health (CDPH), Let's Get Healthy California
- **Description**: This dataset, collected by the California Behavioral Risk Factor Surveillance System (BRFSS), provides data on the proportion of adults aged 18 and older in California who are current smokers. The dataset spans from 2012 onwards, with data broken down by gender. The

data is sourced from a telephone survey conducted by California State University, Sacramento, under the contract of CDPH, with cooperation from the Centers for Disease Control and Prevention (CDC).
- **Relevance**: This dataset is essential for analyzing smoking prevalence in California, offering insights into regional variations in smoking habits. It is particularly useful for tracking changes in smoking rates over time, with caution regarding comparisons between pre-2012 and post-2012 data due to changes in the survey methodology. This dataset will help contextualize trends in smoking in California, enabling the examination of both historical patterns and the effects of changes in survey methodology after 2012.

0.2.5 Research Questions:

Question 1: What percentage of e-cigarette health reports involve seizures, and how have seizure-related incidents changed from 2018 to 2023?

Question 2: Analyze the Relationship Between Shortness of Breath and the Product Defect's Impact on the Second Most Common Health Issue

Question 3: How do the percentages of current smokers, former smokers, and never smokers differ by demographics from 2011 to 2019, and which locations show the highest and lowest smoking cessation rates

Question 4: What are the trends in tobacco use across different states and the influence of literacy standard to consumption?

Question 5: How do government taxes influence the prevalence of smoking and the sales of cigarettes over time?

0.3 Question 1: What percentage of e-cigarette health reports involve seizures, and how have seizure-related incidents changed from 2018 to 2023?

Objective This question aims to quantify the percentage of seizure-related incidents among e-cigarette health reports and analyze their trend from 2018 to 2023 to identify patterns and potential causes.

Steps Taken

1. **Data Loading and Inspection**
   - The dataset `tobacco-problem-0001-of-0001.json` was loaded using the `json` and `pandas` libraries.

   - Inspected key columns: `reported_health_problems`, `tobacco_products`, and `date_submitted` to understand data structure and relevance.
2. **Data Cleaning**
   - Entries with missing or unspecified health issues were removed.

- Multi-item columns, such as `reported_health_problems`, were exploded into individual rows for accurate filtering and analysis.

- All text data was standardized to lowercase for consistency.

- Filtered for reports submitted between 2018 and 2023 to focus on recent trends.

3. **Identification of E-Cigarette-Related Reports**
    - Created a boolean column `is_e_cigarette` by identifying mentions of "e-cigarette," "vape," or "vaping" in the `tobacco_products` column.

    - Subsetted the dataset into `ecig_df`, containing only e-cigarette-related reports.
4. **Detection of Seizure Cases**
    - Used string matching to identify rows with "seizure" in the `reported_health_problems` column.

    - Added a boolean column `is_seizure` to flag seizure-related entries.

    - Calculated the percentage of seizure cases among all e-cigarette-related health reports.
5. **Trend Analysis**
    - Extracted the year from the `date_submitted` column and grouped data by year.

    - Aggregated counts for seizure-related cases and total health reports per year.

    - Created line plots to visualize trends in seizure-related incidents and their share among health reports from 2018–2023.
6. **Visualization**

- A line chart displayed seizure-related incidents from 2018–2023.

- A stacked bar plot showed the share of seizure-related cases against all e-cigarette-related health reports.

[1]:
```
!pip install wordcloud
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import json
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)

# Load the JSON file
with open('tobacco-problem-0001-of-0001.json', 'r') as f:
    data = json.load(f)

# Extract and normalize the 'results' field
df = pd.json_normalize(data['results'])
```

```
Requirement already satisfied: wordcloud in /opt/conda/lib/python3.11/site-
packages (1.9.4)
Requirement already satisfied: numpy>=1.6.1 in /opt/conda/lib/python3.11/site-
packages (from wordcloud) (1.26.3)
Requirement already satisfied: pillow in /opt/conda/lib/python3.11/site-packages
(from wordcloud) (10.2.0)
Requirement already satisfied: matplotlib in
/home/jovyan/.local/lib/python3.11/site-packages (from wordcloud) (3.8.2)
Requirement already satisfied: contourpy>=1.0.1 in
/home/jovyan/.local/lib/python3.11/site-packages (from matplotlib->wordcloud)
(1.2.0)
Requirement already satisfied: cycler>=0.10 in
/home/jovyan/.local/lib/python3.11/site-packages (from matplotlib->wordcloud)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/home/jovyan/.local/lib/python3.11/site-packages (from matplotlib->wordcloud)
(4.47.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/home/jovyan/.local/lib/python3.11/site-packages (from matplotlib->wordcloud)
(1.4.5)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->wordcloud) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in
/home/jovyan/.local/lib/python3.11/site-packages (from matplotlib->wordcloud)
(3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.11/site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.11/site-
packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
```

[2]:

```
# Convert 'date_submitted' to datetime and extract the year
df['date_submitted'] = pd.to_datetime(df['date_submitted'], errors='coerce')
df['year'] = df['date_submitted'].dt.year
```

We are searching for reports explicitly mentioning e-cigarettes, vapes, or vaping in the product descriptions. This step is like putting on a magnifying glass to focus solely on incidents tied to these products. The result is a specialized dataset dedicated to e-cigarette-related health concerns—a story within the broader narrative.

[3]:

```
# Standardize text columns for analysis
for col in ['reported_health_problems', 'tobacco_products',␣
 ↪'reported_product_problems']:
    df[col] = df[col].apply(lambda x: x if isinstance(x, list) else [x])  #␣
 ↪Ensure all entries are lists
    df = df.explode(col).reset_index(drop=True)  # Explode and reset index
    df[col] = df[col].str.strip().str.lower()  # Clean text (strip and␣
 ↪lowercase)
```

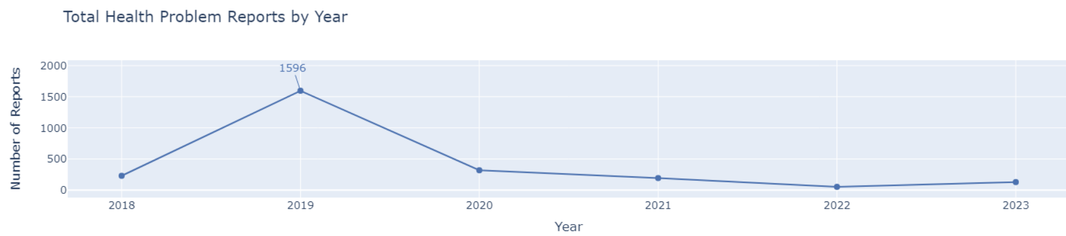
```
[4]: # Filter for relevant years (2018-2023) and e-cigarette related products
df = df[df['year'].between(2018, 2023)]
df['is_e_cigarette'] = df['tobacco_products'].str.
 ↪contains("e-cigarette|vape|vaping", case=False, na=False)
ecig_df = df[df['is_e_cigarette']]
```

0.3.1 Finding the Year with the Most Reports

```
[5]: import pandas as pd
import plotly.graph_objects as go

# Assuming your data is in a DataFrame named 'ecig_df'
# Calculate the number of reports per year
yearly_counts = ecig_df['year'].value_counts().sort_index()

# Print the year with the highest number of reports
highest_year = yearly_counts.idxmax()
print(f"The year with the highest number of reports: {highest_year}␣
 ↪({yearly_counts[highest_year]} reports)")

# Create the interactive line plot
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=yearly_counts.index,
    y=yearly_counts.values,
    mode='lines+markers',
    line=dict(color='#4C72B0', width=2),
    marker=dict(color='#4C72B0', size=8),
    hovertemplate='<b>Year:</b> %{x}<br><b>Reports:</b> %{y}',
    name='Health Problem Reports'
))

# Add annotations for the year with the highest number of reports
fig.add_annotation(
    x=highest_year,
    y=yearly_counts[highest_year],
    text=f"{yearly_counts[highest_year]}",
    showarrow=True,
    arrowhead=1,
    arrowcolor='#4C72B0',
    font=dict(color='#4C72B0', size=14)
)

# Customize the layout
fig.update_layout(
    title='Total Health Problem Reports by Year',
    xaxis_title='Year',
```

```
    yaxis_title='Number of Reports',
    font=dict(size=14),
    hoverlabel=dict(
        bgcolor="white",
        font_size=16,
        font_family="Rockwell"
    )
)

# Display the interactive plot
fig.show()
```

```
The year with the highest number of reports: 2019 (1596 reports)
```



When we looked at reports over time, 2019 stood out with the highest number of health-related incidents—1,596! This was visualized as a line plot, showing the number of reports for each year.

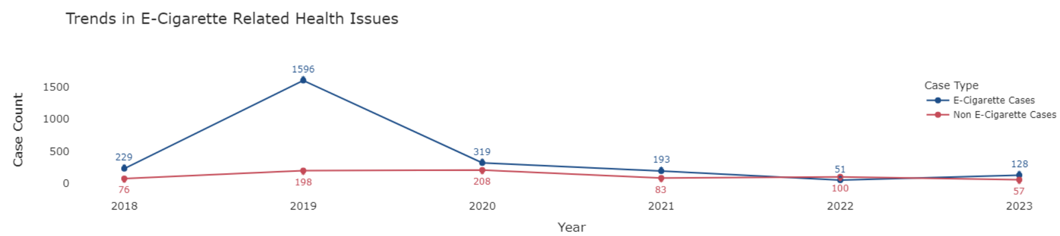
In the plot, 2019 had a sharp peak compared to other years. This tells us that something happened in 2019—either a rise in awareness, a change in e-cigarette usage, or more people reporting issues.

Key Takeaway: 2019 was the year with the most reports, and understanding why could help prevent future problems.

[6]:
```
# Group the data to calculate trends
trends = df.groupby(['year', 'is_e_cigarette']).size().
 ↪reset_index(name='case_count')

# Separate data for e-cigarette and non-e-cigarette cases
e_cig_data = trends[trends['is_e_cigarette'] == True]
non_e_cig_data = trends[trends['is_e_cigarette'] == False]

# Create the interactive plot
fig = go.Figure()

# Add e-cigarette cases line with a contrasting color
fig.add_trace(go.Scatter(
    x=e_cig_data['year'],
    y=e_cig_data['case_count'],
```

```
    mode='lines+markers',
    line=dict(color='#1D4E89', width=2),  # Dark blue color for e-cig cases
    marker=dict(size=8, color='#1D4E89'),
    name='E-Cigarette Cases',
    hovertemplate='<b>Year:</b> %{x}<br><b>Cases:</b> %{y}'
))

# Add non-e-cigarette cases line with another contrasting color
fig.add_trace(go.Scatter(
    x=non_e_cig_data['year'],
    y=non_e_cig_data['case_count'],
    mode='lines+markers',
    line=dict(color='#C54B58', width=2),  # Dark red color for non-e-cig cases
    marker=dict(size=8, color='#C54B58'),
    name='Non E-Cigarette Cases',
    hovertemplate='<b>Year:</b> %{x}<br><b>Cases:</b> %{y}'
))

# Add annotations for each data point
for i, row in e_cig_data.iterrows():
    fig.add_annotation(
        x=row['year'],
        y=row['case_count'],
        text=f"{row['case_count']}",
        showarrow=True,
        arrowhead=2,
        arrowcolor='#1D4E89',
        font=dict(color='#1D4E89', size=12),
        ax=0,
        ay=-15
    )

for i, row in non_e_cig_data.iterrows():
    fig.add_annotation(
        x=row['year'],
        y=row['case_count'],
        text=f"{row['case_count']}",
        showarrow=True,
        arrowhead=2,
        arrowcolor='#C54B58',
        font=dict(color='#C54B58', size=12),
        ax=0,
        ay=15
    )

# Customize layout with the new background and contrasting colors
fig.update_layout(
```

```
    title='Trends in E-Cigarette Related Health Issues',
    xaxis_title='Year',
    yaxis_title='Case Count',
    font=dict(size=14, color='#2A2A2A'),  # Dark gray font for readability
    hoverlabel=dict(
        bgcolor="white",
        font_size=14,
        font_family="Rockwell"
    ),
    legend=dict(
        title='Case Type',
        font=dict(size=12, color='#2A2A2A'),
        orientation="v",  # Make legend vertical
        x=0.85,  # Position the legend inside the plot
        y=0.9,  # Adjust to the top-right inside the plot
        xanchor="left",
        yanchor="top"
    ),
    plot_bgcolor='white',  # Set plot background color to the given pinkish hue
    paper_bgcolor='white'  # Set paper background to match
)

# Show the interactive plot
fig.show()
```



The line plot shows a clear increase in health problem cases related to e-cigarettes and vaping products from 2018 to 2019. During this time, the number of e-cigarette related cases grew significantly, while the number of non-e-cigarette related cases remained relatively flat.

0.3.2 Finding health problems

[7]:
```
##  Unique Health Problems
unique_health_problems = ecig_df['reported_health_problems'].nunique()
print(f"Unique health problems reported: {unique_health_problems}")
```

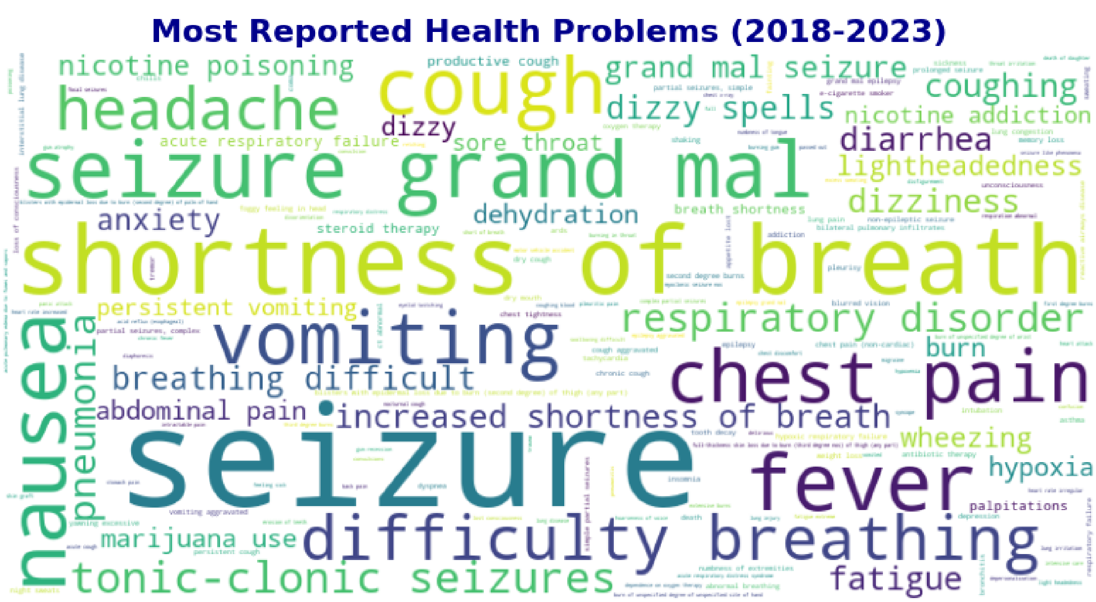
```
Unique health problems reported: 787
```

[8]:

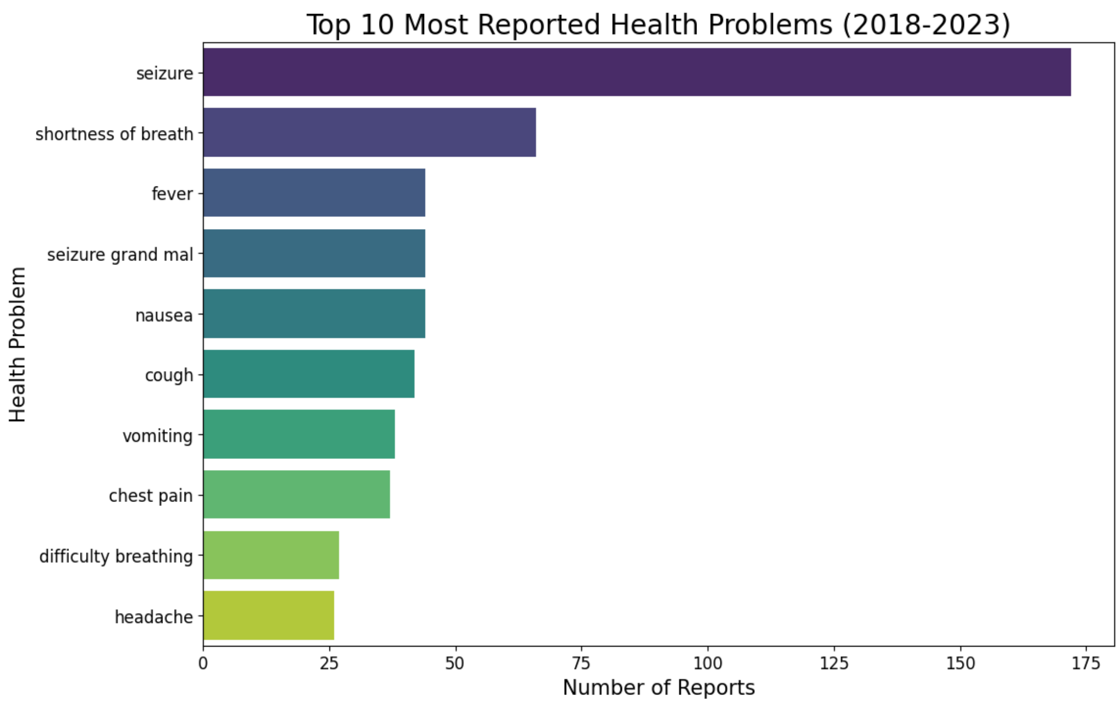
```python
# Remove 'other' and 'no information provided' categories from the dataset
filtered_df = ecig_df[~ecig_df['reported_health_problems'].isin(['other', 'no information provided'])]

# Merge 'seizures' and 'seizure' into one category using .loc to avoid SettingWithCopyWarning
filtered_df.loc[filtered_df['reported_health_problems'] == 'seizures', 'reported_health_problems'] = 'seizure'
```

[9]:

```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Generate a word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(filtered_df['reported_health_problems'].value_counts())

# Display the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')  # Hide the axes
plt.title('Most Reported Health Problems (2018-2023)', fontsize=16, fontweight='bold', color='darkblue')
plt.show()
```



```
[10]: # Get the top 10 most reported health problems
health_problem_counts = filtered_df['reported_health_problems'].value_counts().
 ↪head(10)

# Plot the top 10 most frequent health problems
plt.figure(figsize=(12, 8))  # Increase figure size
sns.barplot(y=health_problem_counts.index, x=health_problem_counts.values,␣
 ↪palette='viridis')

# Title and labels
plt.title('Top 10 Most Reported Health Problems (2018-2023)', fontsize=20)
plt.xlabel('Number of Reports', fontsize=15)
plt.ylabel('Health Problem', fontsize=15)

# Increase tick label size for readability
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

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



0.3.3 Top 10 Most Reported Health Problems (2018-2023)

1. **Seizure**
   This condition dominates the list, with the highest number of reports, emphasizing its promi-

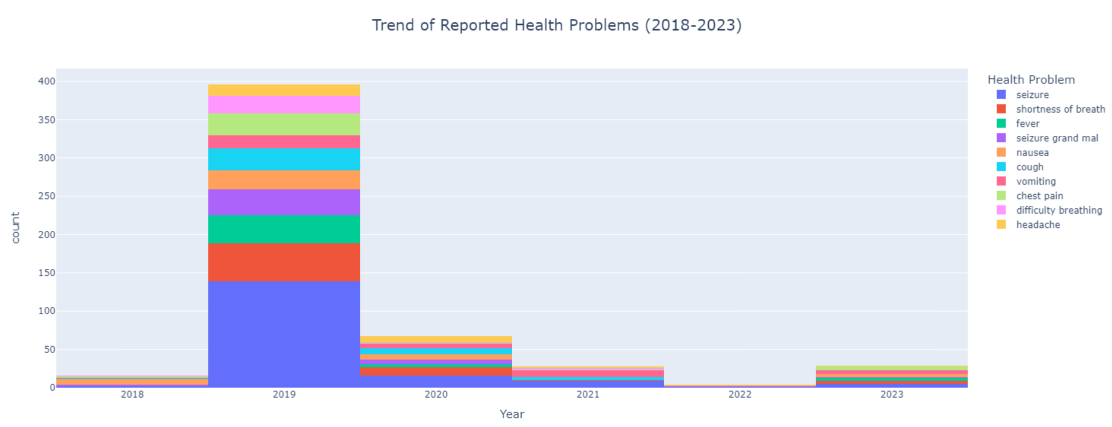
nence as a serious health issue related to e-cigarette usage.

2. **Respiratory Problems**
   Breathing difficulties and related ailments are a close second, underscoring the impact of e-cigarettes on lung health.
3. **Neurological Symptoms**
   Conditions like dizziness and fainting also feature prominently, highlighting potential neurological risks.
4. **Cardiovascular Issues**
   Problems related to the heart and circulation were recurrent, showing that e-cigarettes might stress the cardiovascular system.
5. **Addiction Symptoms**
   Reports also indicated withdrawal and dependency concerns, pointing to the addictive potential of nicotine in e-cigarettes.
6. **Oral and Throat Irritation**
   Symptoms like a sore throat or mouth discomfort reflect the local irritation caused by vaping.
7. **Skin Issues**
   Some users reported skin reactions, likely due to exposure to certain chemicals or burns.
8. **Nausea and Vomiting**
   Digestive symptoms were common among the reported issues.
9. **General Malaise**
   Many reports categorized vague but significant feelings of unwellness or fatigue.
10. **Chest Pain**
    This was another alarming issue, hinting at potential acute or chronic respiratory and cardiac distress.

We can see seizure is the most occured health problem!

[11]:
```
import plotly.express as px

# Remove 'other' and 'no information provided' categories
filtered_df = ecig_df[~ecig_df['reported_health_problems'].isin(['other', 'no␣
↪information provided'])]

# Merge 'seizures' and 'seizure' into one category using .loc
filtered_df.loc[filtered_df['reported_health_problems'] == 'seizures',␣
↪'reported_health_problems'] = 'seizure'

# Get the top 10 most reported health problems
top_health_problems = filtered_df['reported_health_problems'].value_counts().
↪head(10).index

# Filter the data to include only these top 10 health problems
```

```
filtered_ecig_df = filtered_df[filtered_df['reported_health_problems'].
↪isin(top_health_problems)]

# Create the interactive plot using plotly
fig = px.histogram(
    filtered_ecig_df,
    x='year',
    color='reported_health_problems',
    category_orders={'reported_health_problems': list(top_health_problems)},  #␣
↪Only top 10 health problems
    title='Trend of Reported Health Problems (2018-2023)',
    labels={'year': 'Year', 'reported_health_problems': 'Health Problem'},
    color_discrete_sequence=px.colors.qualitative.Plotly,
    barmode='stack'
)

# Increase size of the figure for better visualization
fig.update_layout(
    height=600,  # Increase the height
    width=750,  # Increase the width
    title_x=0.5,  # Center the title
    title_font=dict(size=20),  # Increase title font size
    xaxis_title_font=dict(size=15),  # X-axis title font size
    yaxis_title_font=dict(size=15),  # Y-axis title font size
    legend_title_font=dict(size=15),  # Legend title font size
    legend_font=dict(size=12)  # Legend font size
)

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



This graph provides a year-by-year breakdown of the top reported health problems associated with e-cigarette use from 2018 to 2023. Each health issue is represented as a different color within a stacked bar chart, allowing us to compare trends for individual problems as well as overall changes over time.

0.3.4 Analysis of the Graph: Trend of Reported Health Problems (2018-2023)

1. **Peak Year (2019)**
   The graph shows a significant spike in reported health problems in 2019. This might reflect:
   - Heightened awareness of e-cigarette risks.
   - Increased use of e-cigarettes during that time.
   - Enhanced reporting mechanisms or public health campaigns.
2. **Consistent Trends**
   After 2019, the overall number of reports stabilizes or slightly decreases. However, specific health issues, such as seizures, remain consistently prominent throughout the years.
3. **Health Problem Contributions**
   - Seizures dominate the reports across all years, emphasizing their critical role in e-cigarette-related health concerns.
   - Other problems, such as respiratory issues and cardiovascular concerns, also maintain significant contributions, showing the broad spectrum of health risks associated with vaping.

0.3.5 Focus on Seizures

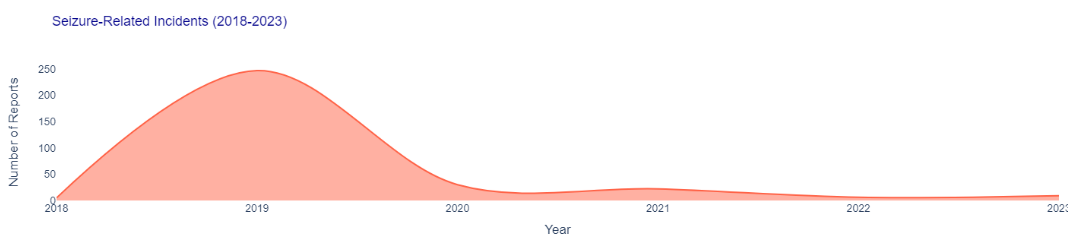
Among the various health problems, seizures stood out as a critical concern. About 12.68% of all e-cigarette-related health reports—roughly 319 incidents—involved seizures. This makes seizures one of the most frequently reported health problems linked to e-cigarette use. Interestingly, the trend in seizure reports closely mirrored the overall trend in e-cigarette-related health problems, with a notable concentration of cases in 2019.

[12]:
```python
## Percentage of E-Cigarette Reports Involving Seizures
seizure_df = ecig_df[ecig_df['reported_health_problems'].str.
 ↪contains('seizure', na=False)]
seizure_percentage = (len(seizure_df) / len(ecig_df)) * 100
print(f"Percentage of e-cigarette health reports involving seizures:␣
 ↪{seizure_percentage:.2f}%")
```

```
Percentage of e-cigarette health reports involving seizures: 12.68%
```

[13]:
```python
import plotly.express as px

# Seizure incidents over the years (count the number of incidents for each year)
seizure_counts_per_year = seizure_df['year'].value_counts().sort_index().
 ↪reset_index()
seizure_counts_per_year.columns = ['year', 'count']  # Rename columns for␣
 ↪better reference
```

```python
# Create an area chart with a smooth line
fig = px.area(
    seizure_counts_per_year,
    x='year',
    y='count',
    title='Seizure-Related Incidents (2018-2023)',
    labels={'year': 'Year', 'count': 'Number of Reports'},
    line_shape='spline',  # This creates a smooth curve
    color_discrete_sequence=['#FF6347'],  # Red color palette
)

# Customize the layout for better aesthetics
fig.update_layout(
    xaxis_title='Year',
    yaxis_title='Number of Reports',
    font=dict(family="Arial, sans-serif", size=14),
    plot_bgcolor='white',
    title_font=dict(size=18, family='Arial', color='darkblue'),
    showlegend=False
)

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



Findings

1. **Percentage of Seizure Cases in E-Cigarette Reports**
    - **12.68%** of health reports related to e-cigarettes involved seizures.
2. **Trends from 2018–2023**
    - The year 2019 recorded the highest number of seizure-related incidents, potentially driven by increased awareness and reporting.

    - A steady decline in seizure cases was observed after 2019, possibly due to regulatory actions or public health campaigns.

Respiratory issues have been identified as the second most prevalent health concern in our analysis. This significant observation highlights the need for a deeper understanding of potential underlying causes. In the next phase of the analysis, we aim to explore whether specific product-related factors or usage patterns may be contributing to the prevalence of these respiratory problems.

By examining product-specific characteristics, such as composition, manufacturing processes, or environmental exposure during usage, we can uncover insights into their potential link with respiratory health concerns. This deeper exploration will not only help pinpoint the root causes but also guide recommendations for mitigating these health issues through product improvements, regulatory measures, or targeted interventions.

0.4 Question 2: How Does the Product Defect Impact the Relationship Between Shortness of Breath - the Second Most Common Health Issue?

0.4.1 Objective:

The analysis aims to explore the reported product problems related to tobacco products, particularly e-cigarettes, and their associated health issues. The goal is to identify and visualize: 1. The most common product problems reported. 2. Differences between e-cigarette and non-e-cigarette related issues. 3. Health problems caused by specific product issues, focusing on key categories such as "foreign material," "taste issues," and "child safety hazards." 4. Insights into the top health problems associated with each product issue.

0.4.2 Steps Taken

1. **Data Cleaning and Preparation**
   - Removed extra spaces and converted all values in the `reported_product_problems` column to lowercase for consistency using `.str.strip()` and `.str.lower()`.
   - Filtered out rows where `reported_product_problems` was "no information provided" or "other" to focus on meaningful data.
   - Flagged rows involving tobacco products using `.str.contains()` to identify mentions of e-cigarettes, vaping, or vape pens in the `tobacco_products` column.
2. **Analyzing Product Problems**
   - Counted occurrences of each unique `reported_product_problems` using `.value_counts()`.
   - Visualized the distribution of product problems with a bar graph to highlight the most frequently reported issues.
3. **E-Cigarette vs Non-E-Cigarette Analysis**
   - Focused on the top 3 most common product problems and compared the counts for cases involving e-cigarettes versus non-e-cigarette products.
   - Converted the results into a DataFrame for visualization and created a bar chart to display the comparison.
4. **Filtering Data for Specific Product Problems**
   - Isolated rows with `reported_product_problems` categorized as "foreign material," "taste issue," or "child safety hazard."
   - Excluded rows where `reported_health_problems` was "no information provided" or "other" to refine the dataset further.
5. **Health Problems Analysis**
   - Grouped data by `reported_product_problems` and `reported_health_problems` and

counted occurrences for each combination using `.groupby()` and `.size()`.
- Created a stacked bar chart to visualize the frequency of health problems for each product problem.

6. **Top Health Problems for Each Product Problem**
   - For each of the three specific product problems, identified the top 3 associated health problems using `.nlargest()`.
   - Plotted separate bar charts for each product problem to display the most common health issues.
7. **Visualization**
   - Multiple bar charts were created to summarize the findings:
     - The overall frequency of product problems.
     - Comparison of e-cigarette versus non-e-cigarette cases for the top product problems.
     - Stacked bar chart of health problems caused by product issues.
     - Individual bar charts for the top 3 health problems associated with each specific product problem.

[14]:
```python
import pandas as pd
import json

# Replace 'file_path.json' with the path to your JSON file
file_path = 'tobacco-problem-0001-of-0001.json'

# Load the JSON file
with open(file_path, 'r') as f:
    data = json.load(f)

# Extract the "results" part of the JSON
results = data.get('results', [])

# Convert the "results" array to a DataFrame
df = pd.DataFrame(results)
```

[15]:
```python
# Exploding relevant columns for detailed row-level analysis
for col in ['reported_health_problems', 'tobacco_products',␣
 ↪'reported_product_problems']:
    df[col] = df[col].apply(lambda x: x if isinstance(x, list) else [x])
    df = df.explode(col).reset_index(drop=True)
```

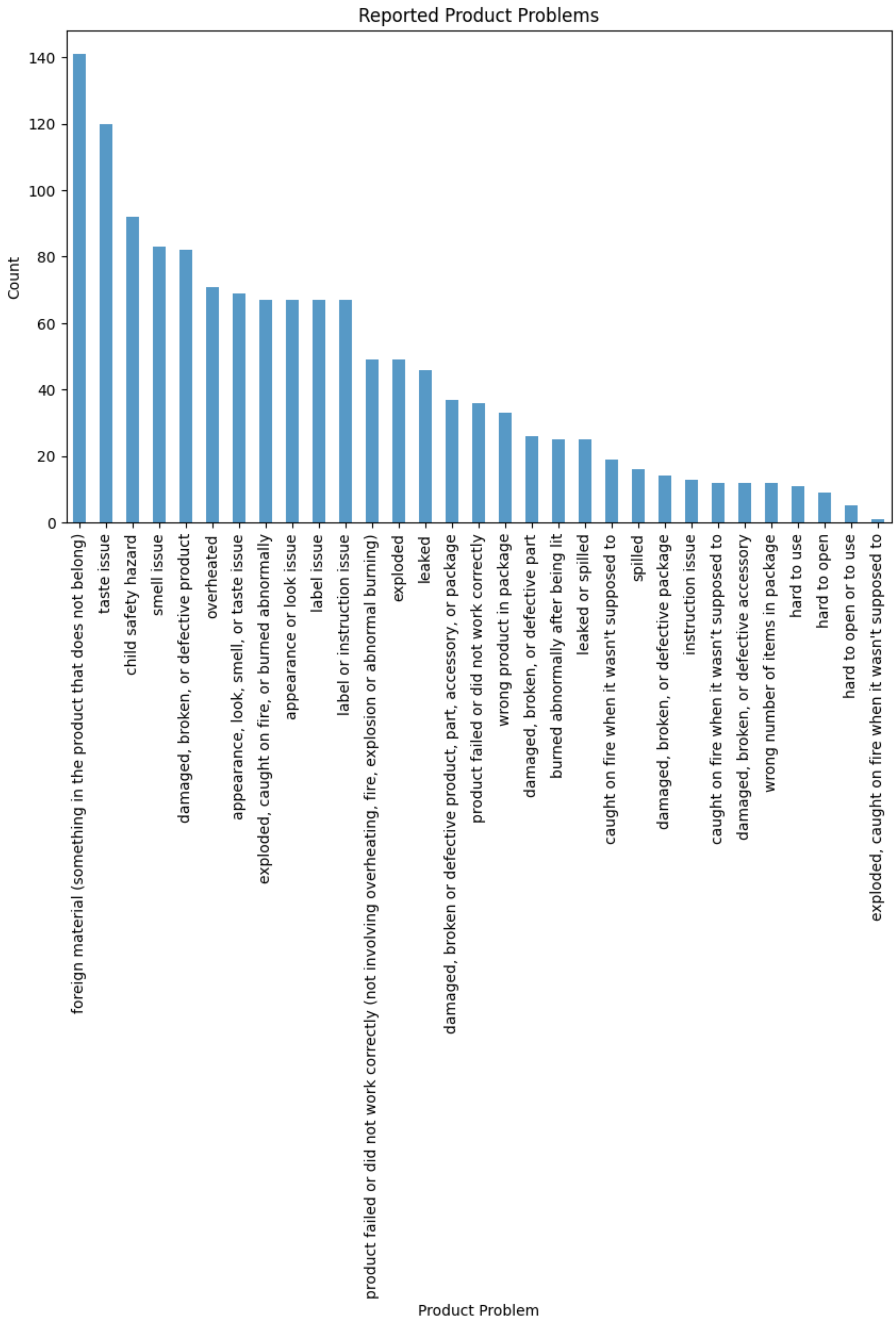
[16]:
```python
# Cleaning health problem column
df['reported_product_problems'] = df['reported_product_problems'].str.strip().
 ↪str.lower()
df = df[df['reported_product_problems'] != "no information provided"]
# Flagging rows involving e-cigarettes
df['is_product_problem'] = df['tobacco_products'].str.contains("", case=False,␣
 ↪na=False)

df = df[df['reported_product_problems'] != "other"]
```

[17]:
```python
import pandas as pd
import matplotlib.pyplot as plt


# Group data by product problems and count occurrences
product_problem_counts = df["reported_product_problems"].value_counts()
```

[18]:
```python
# Flagging rows involving e-cigarettes
df['is_e_cigarette'] = df['tobacco_products'].str.
 ↪contains("e-cigarette|vaping|vape pen", case=False, na=False)
```

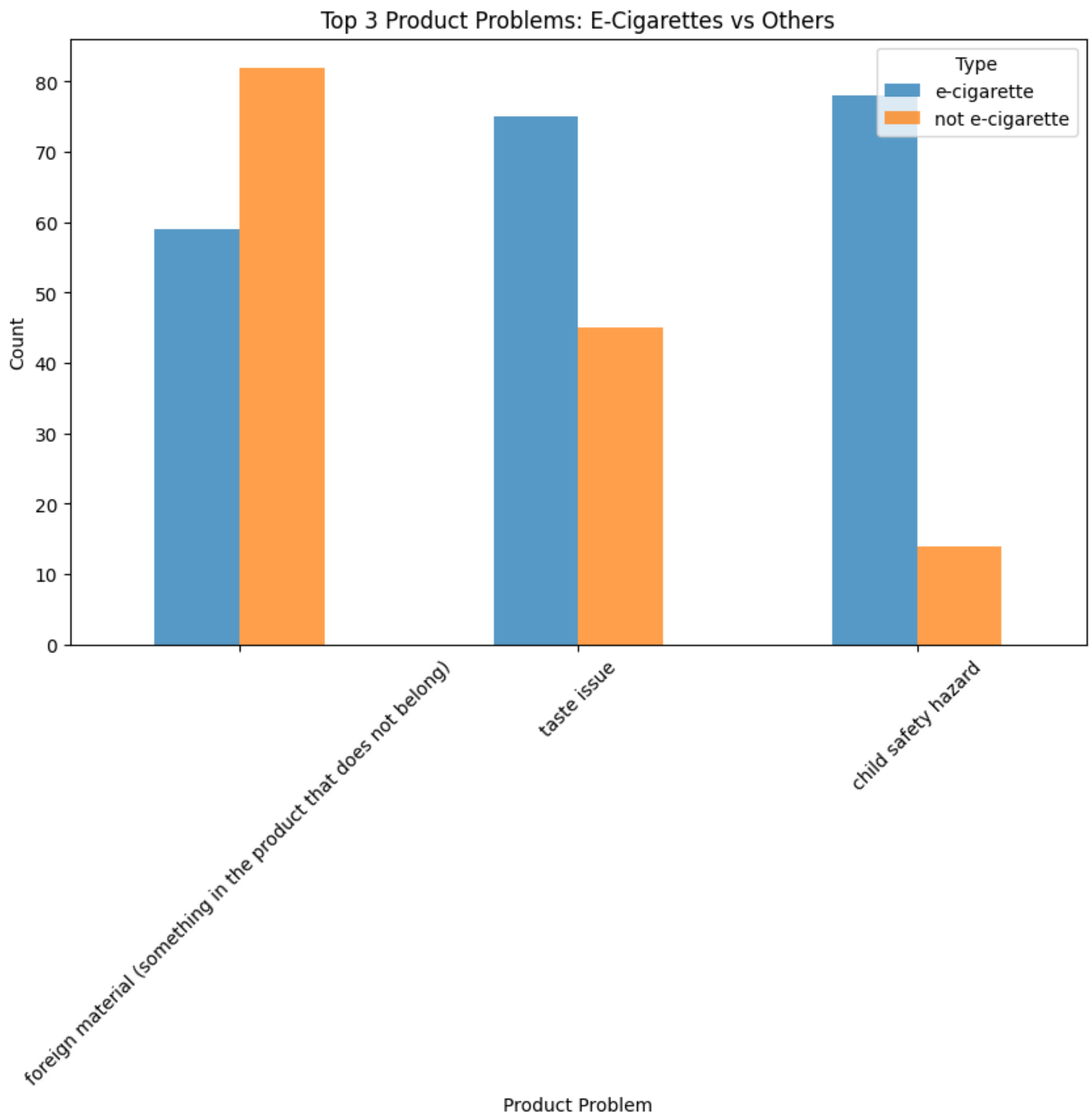
[19]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Group data by product problems and count occurrences
product_problem_counts = df["reported_product_problems"].value_counts()

# Plot a bar graph
plt.figure(figsize=(10, 6))
product_problem_counts.plot(kind="bar", alpha=0.75)
plt.title("Reported Product Problems")
plt.xlabel("Product Problem")
plt.ylabel("Count")
plt.show()

# Get the top 3 product problems
top_3_problems = product_problem_counts.head(3).index

# Analyze the top 3 product problems for e-cigarette vs non-e-cigarette
e_cig_counts = {}
for problem in top_3_problems:
    subset = df[df["reported_product_problems"] == problem]
    e_cig_count = subset["is_e_cigarette"].sum()  # Count where is_e_cigarette␣
 ↪is True
    not_e_cig_count = len(subset) - e_cig_count  # Count where is_e_cigarette␣
 ↪is False
    e_cig_counts[problem] = {"e-cigarette": e_cig_count, "not e-cigarette":␣
 ↪not_e_cig_count}

# Convert results to a DataFrame
e_cig_df = pd.DataFrame.from_dict(e_cig_counts, orient="index")

# Print the result
print(e_cig_df)

# Plot the results for better visualization
e_cig_df.plot(kind="bar", figsize=(10, 6), alpha=0.75)
```

```python
plt.title("Top 3 Product Problems: E-Cigarettes vs Others")
plt.xlabel("Product Problem")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.legend(title="Type")
plt.show()
```



Reported Product Problems
Count
0
20
40
60
80
100
120
140
foreign material (something in the product that does not belong)
taste issue
child safety hazard
smell issue
damaged, broken, or defective product
overheated
appearance, look, smell, or taste issue
exploded, caught on fire, or burned abnormally
appearance or look issue
label issue
label or instruction issue
product failed or did not work correctly (not involving overheating, fire, explosion or abnormal burning)
exploded
leaked
damaged, broken or defective product, part, accessory, or package
product failed or did not work correctly
wrong product in package
damaged, broken, or defective part
burned abnormally after being lit
leaked or spilled
caught on fire when it wasn't supposed to
spilled
damaged, broken, or defective package
instruction issue
caught on fire when it wasn't supposed to
damaged, broken, or defective accessory
wrong number of items in package
hard to use
hard to open
hard to open or to use
exploded, caught on fire when it wasn't supposed to
Product Problem

```
                                                    e-cigarette  \
foreign material (something in the product that…            59
taste issue                                                  75
child safety hazard                                          78

                                                    not e-cigarette
foreign material (something in the product that…                82
taste issue                                                      45
child safety hazard                                              14
```



0.4.3 Insights from the Graph

The bar chart displays the frequency of **reported product problems** in descending order. Below are the key insights:

---

1. **Dominant Issues**:
   - The most frequently reported issue is **"foreign material"** (something in the product that does not belong), with a count exceeding 100.

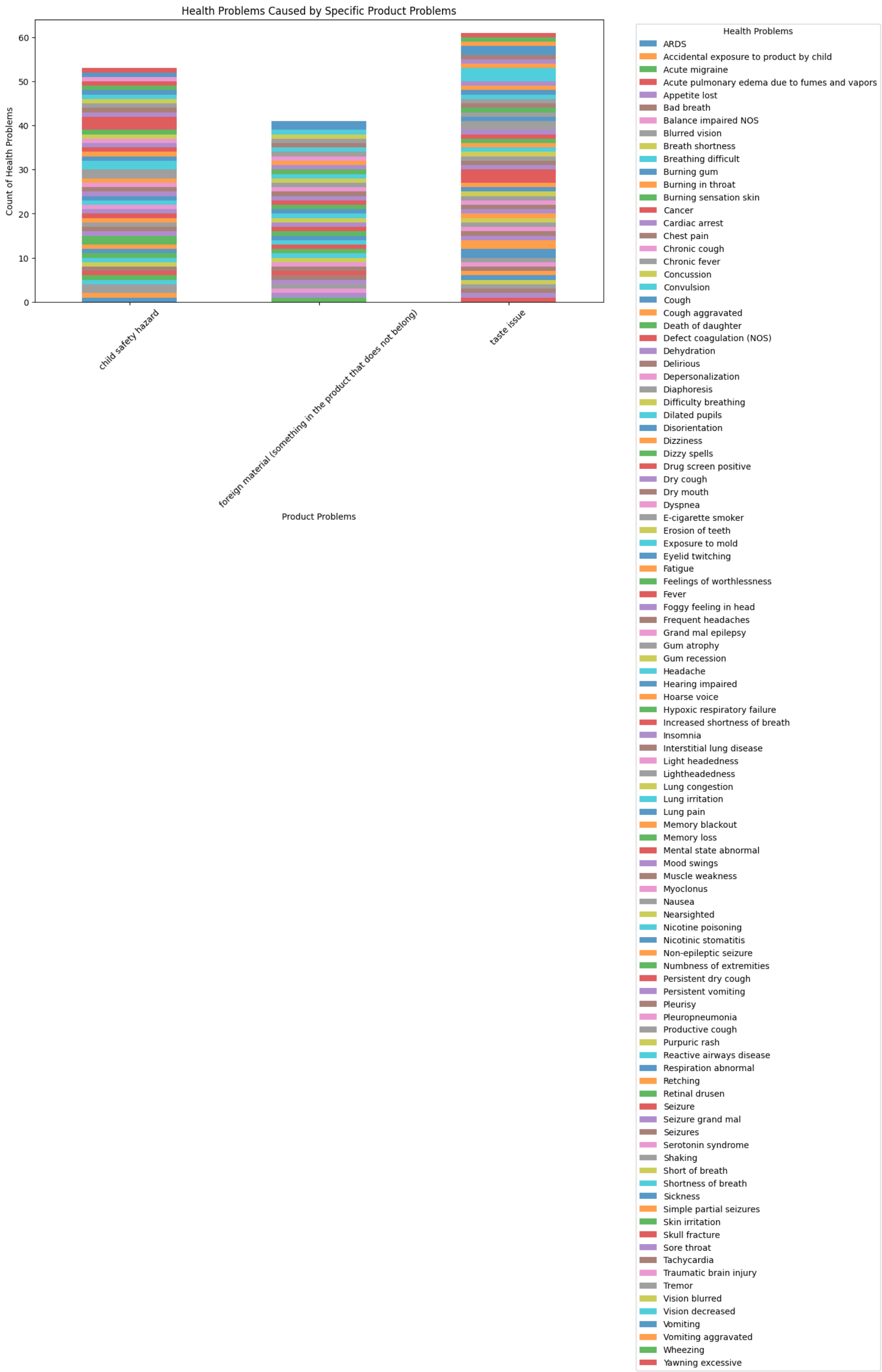
   - This highlights a significant concern related to **product quality control** and **contamination**.
2. **Taste-related Problems**:
   - The second most common problem is **"taste issue"**, indicating dissatisfaction with the product's sensory qualities.

   - This can negatively affect **consumer trust** and **brand loyalty**.
3. **Safety Concerns**:
   - Issues like **"child safety hazard"** and **"damaged/defective"** products rank high, showcasing critical **safety risks** and potential legal or regulatory liabilities for manufacturers.
4. **Mechanical or Structural Problems**:
   - Problems such as **"product does not work correctly"**, **"burned product"**, and **"explosion-related issues"** suggest recurring **mechanical defects** or **manufacturing issues**.
5. **Low-frequency Issues**:
   - At the lower end of the graph, problems like **"hard to open or close"** and **"exploded, caught fire"** are less frequent but still significant, as they pose potential safety hazards.
6. **Clustering of Categories**:
   - Similar issues, such as **"burned product"** and **"exploded, caught fire"**, indicate recurring themes around **product performance and safety**.

---

[20]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Filter data for the specific product problems
filtered_df = df[df["reported_product_problems"].isin([
    "foreign material (something in the product that does not belong)",
    "taste issue",
    "child safety hazard"
])]

# Exclude rows with unwanted health problem values
filtered_df = filtered_df[
    ~filtered_df["reported_health_problems"].isin(["No information provided",␣
 ↪"Other", "other"])
```

```
]

filtered_df = filtered_df[
    ~filtered_df["is_e_cigarette"].isin([False])
]


# Group by product problems and health problems, and count occurrences
health_problem_counts = (
    filtered_df.groupby(["reported_product_problems",␣
 ↪"reported_health_problems"])
    .size()
    .unstack(fill_value=0)
)

# Calculate total occurrences for each health problem
total_health_problem_counts = health_problem_counts.sum(axis=0)



# Plot the stacked bar chart
health_problem_counts.plot(kind="bar", stacked=True, figsize=(12, 6), alpha=0.
 ↪75)
plt.title("Health Problems Caused by Specific Product Problems")
plt.xlabel("Product Problems")
plt.ylabel("Count of Health Problems")
plt.xticks(rotation=45)
plt.legend(title="Health Problems", bbox_to_anchor=(1.05, 1), loc="upper left")
plt.show()
```



Health Problems Caused by Specific Product Problems
Count of Health Problems
0
10
20
30
40
50
60
child safety hazard
foreign material (something in the product that does not belong)
taste issue
Product Problems
Health Problems
ARDS
Accidental exposure to product by child
Acute migraine
Acute pulmonary edema due to fumes and vapors
Appetite lost
Bad breath
Balance impaired NOS
Blurred vision
Breath shortness
Breathing difficult
Burning gum
Burning in throat
Burning sensation skin
Cancer
Cardiac arrest
Chest pain
Chronic cough
Chronic fever
Concussion
Convulsion
Cough
Cough aggravated
Death of daughter
Defect coagulation (NOS)
Dehydration
Delirious
Depersonalization
Diaphoresis
Difficulty breathing
Dilated pupils
Disorientation
Dizziness
Dizzy spells
Drug screen positive
Dry cough
Dry mouth
Dyspnea
E-cigarette smoker
Erosion of teeth
Exposure to mold
Eyelid twitching
Fatigue
Feelings of worthlessness
Fever
Foggy feeling in head
Frequent headaches
Grand mal epilepsy
Gum atrophy
Gum recession
Headache
Hearing impaired
Hoarse voice
Hypoxic respiratory failure
Increased shortness of breath
Insomnia
Interstitial lung disease
Light headedness
Lightheadedness
Lung congestion
Lung irritation
Lung pain
Memory blackout
Memory loss
Mental state abnormal
Mood swings
Muscle weakness
Myoclonus
Nausea
Nearsighted
Nicotine poisoning
Nicotinic stomatitis
Non-epileptic seizure
Numbness of extremities
Persistent dry cough
Persistent vomiting
Pleurisy
Pleuropneumonia
Productive cough
Purpuric rash
Reactive airways disease
Respiration abnormal
Retching
Retinal drusen
Seizure
Seizure grand mal
Seizures
Serotonin syndrome
Shaking
Short of breath
Shortness of breath
Sickness
Simple partial seizures
Skin irritation
Skull fracture
Sore throat
Tachycardia
Traumatic brain injury
Tremor
Vision blurred
Vision decreased
Vomiting
Vomiting aggravated
Wheezing
Yawning excessive

0.4.4 Insights from the Visualizations

The stacked bar chart and accompanying health problem legends show the health issues caused by three specific product problems:

1. **Child Safety Hazard**

2. **Foreign Material (Something in the Product That Does Not Belong)**

3. **Taste Issue**

0.4.5 Key Observations

1. **"Taste Issue"**
    - This problem has the **highest count of health problems**, with approximately 60 occurrences.

    - Taste issues are linked to **respiratory problems** (e.g., shortness of breath, persistent dry cough) and **oral symptoms** (e.g., gum atrophy, bad breath, nausea).
2. **"Child Safety Hazard"**
    - Child safety hazards account for **more than 50 health problems**.

    - Health issues here are severe and likely tied to **accidental exposure** and ingestion:
        - Examples include **convulsions**, **accidental exposure to product by child**, and critical symptoms like **cardiac arrest** and **seizures**.
3. **"Foreign Material"**
    - This issue has around **40 health problems** reported.

    - Contamination or foreign material likely causes problems across various health domains, including:
        - **Respiratory issues**: Shortness of breath, lung irritation, and persistent dry cough.

        - **Gastrointestinal and systemic effects**: Nausea, vomiting, and dizziness.

0.4.6 Common Health Problems

From the detailed legends, we observe several recurring health problems:
- **Respiratory Issues**: Shortness of breath, cough, lung congestion, and hypoxic respiratory failure.
- **Severe Symptoms**: Seizures, cardiac arrest, and traumatic brain injury.
- **Neurological Symptoms**: Dizziness, memory loss, nausea, and shaking.
- **Oral Issues**: Gum atrophy, burning gums, and tooth erosion.

0.4.7 Observations

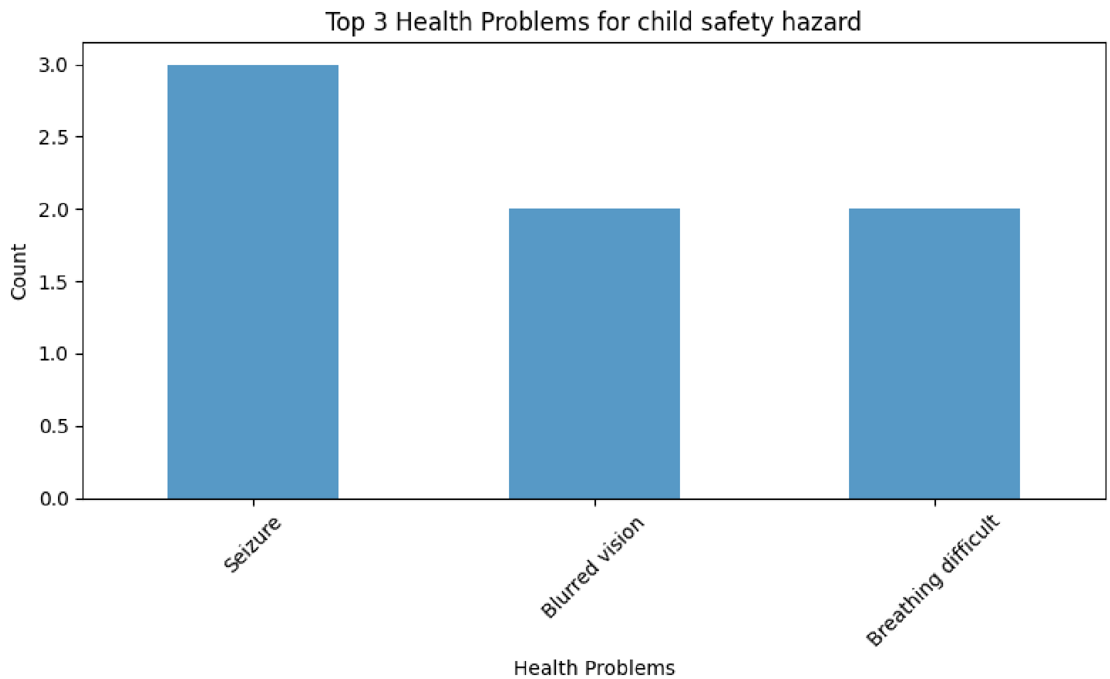
- **Taste Issues** are the most frequent problem, impacting respiratory and oral health significantly.
- **Child Safety Hazards** pose critical risks, with severe outcomes such as seizures and cardiac arrest.
- **Foreign Material Contamination** highlights failures in product safety and quality control, leading to multi-system health impacts.

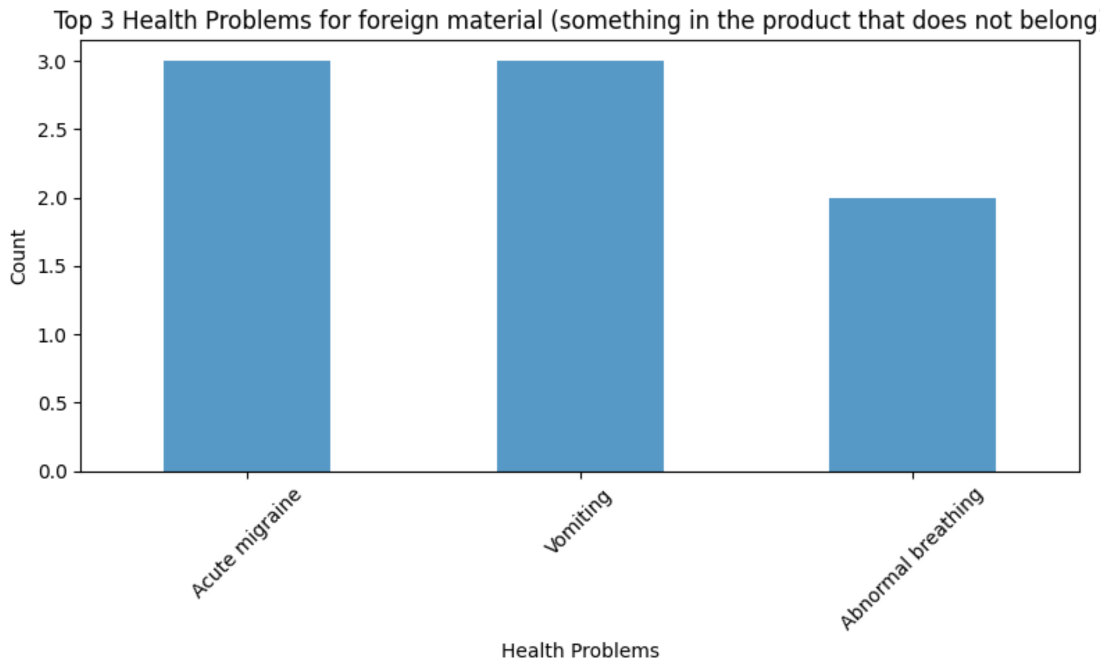
[21]:
```python
import pandas as pd

# Assuming you have the DataFrame 'df' prepared as in the previous code snippet

# Filter data for the specific product problems
filtered_df = df[df["reported_product_problems"].isin([
    "foreign material (something in the product that does not belong)",
    "taste issue",
    "child safety hazard"
])]

# Exclude rows with unwanted health problem values
filtered_df = filtered_df[
    ~filtered_df["reported_health_problems"].isin(["No information provided",␣
 ↪"Other", "other"])
]

# Group by product problems and health problems, and count occurrences
health_problem_counts = (
    filtered_df.groupby(["reported_product_problems",␣
 ↪"reported_health_problems"])
    .size()
    .unstack(fill_value=0)
)

# Get top 3 health problems for each product problem
top_3_health_problems = {}
for problem in health_problem_counts.index:
    top_3_health_problems[problem] = health_problem_counts.loc[problem].
 ↪nlargest(3).index.tolist()

print(top_3_health_problems)
```

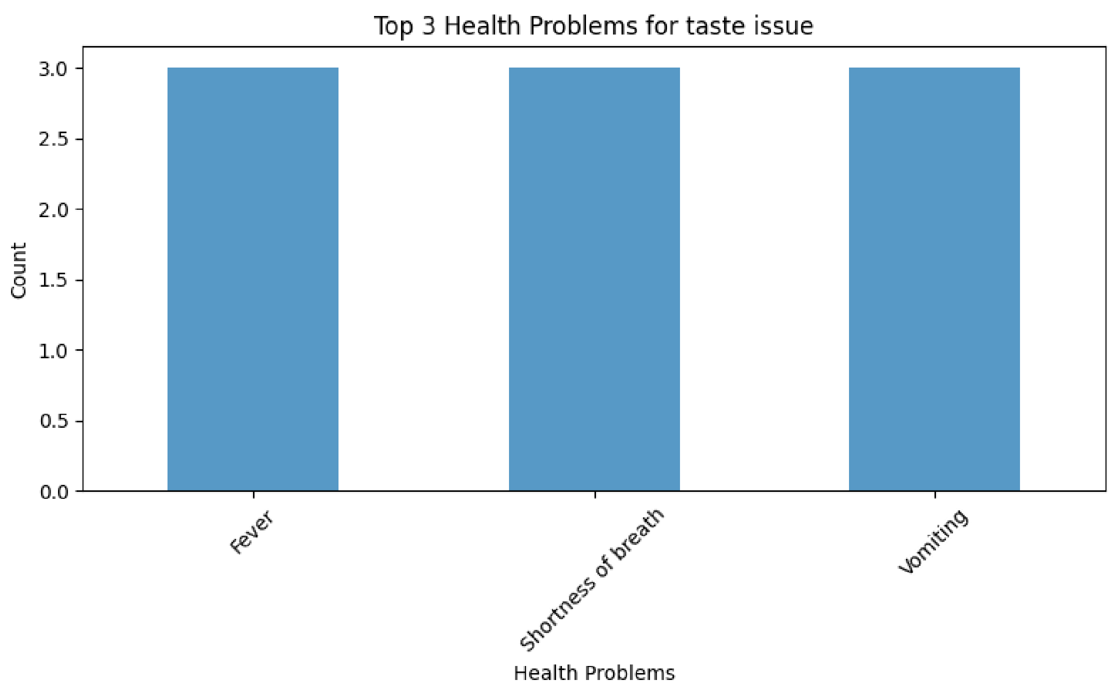
```
{'child safety hazard': ['Seizure', 'Blurred vision', 'Breathing difficult'],
'foreign material (something in the product that does not belong)': ['Acute
migraine', 'Vomiting', 'Abnormal breathing'], 'taste issue': ['Fever',
'Shortness of breath', 'Vomiting']}
```

[22]:

```
import matplotlib.pyplot as plt

# Assuming you have the 'health_problem_counts' DataFrame from the previous code

# Create separate bar plots for each product problem
for problem in health_problem_counts.index:
    plt.figure(figsize=(8, 5))  # Adjust figure size as needed
    health_problem_counts.loc[problem].nlargest(3).plot(kind='bar', alpha=0.75)
    plt.title("Top 3 Health Problems for {}".format(problem))
    plt.xlabel("Health Problems")
    plt.ylabel("Count")
    plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
    plt.tight_layout()  # Adjust layout to prevent labels from overlapping
    plt.show()
```




Top 3 Health Problems for foreign material (something in the product that does not belong)
Count
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Acute migraine
Vomiting
Abnormal breathing
Health Problems


Top 3 Health Problems for taste issue
Count
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Fever
Shortness of breath
Vomiting
Health Problems

0.4.8 Key Insights

Given that **shortness of breath** has consistently appeared as one of the most reported health problems, there is a **high likelihood** that the product issues—especially those related to **taste problems** and **foreign material**—are contributing to respiratory complications. If you are a **smoker experiencing breathing problems**, it strongly suggests that the product may be exacerbating or causing these health concerns.

The growing prevalence of **shortness of breath** among smokers and vapers has become a focal point in understanding the broader implications of tobacco and e-cigarette use. Our initial analysis revealed a **high likelihood** that product defects, such as **taste problems** and **foreign materials**, are exacerbating respiratory complications for smokers. These issues, coupled with the inherent health risks of smoking, paint a troubling picture of the product's role in worsening health outcomes.

However, the rise of **e-cigarettes** has added a new dimension to this narrative. Between **2018 and 2023**, health problems linked to vaping, including **seizures**, surged—accounting for **12.68% of reported cases**. The sharp spike in **2019** highlighted the severe consequences of vaping, such as **respiratory** and **cardiovascular issues**, raising concerns about the trade-offs between traditional tobacco use and vaping.

To contextualize these findings, we conducted a deeper analysis of smoking habits from **2011 to 2019** across diverse demographics. This analysis provided insight into how many people continue to smoke, how many have quit, and how many have never smoked. By breaking this data down by **gender**, **race**, and **region**, we identified trends that reveal not only the persistence of smoking behaviors but also the shifting landscape of tobacco consumption in the face of emerging vaping trends.

This comprehensive approach allowed us to juxtapose the risks associated with traditional tobacco use against the emerging health threats of e-cigarettes. The findings emphasize the critical need for targeted interventions addressing both forms of consumption. While cessation campaigns and prevention strategies have shown success in some areas, the rise in vaping-related health issues underscores the urgency of adapting public health efforts to mitigate the dual threats posed by smoking and vaping.

0.5 Question 3: How do the percentages of current smokers, former smokers, and never smokers differ by demographics from 2011 to 2019, and which locations show the highest and lowest smoking cessation rates?

Objective: The goal is to observe how smoking habits—whether people are current smokers, former smokers, or have never smoked—have changed over time from 2011 to 2019. We also want to know how these habits differ among different groups of people (like by gender, race, or location) and which places have been the best and worst at helping people quit smoking.

Steps Taken

1. **Data Loading and Inspection**
   - The dataset was loaded using `pandas.read_csv()` to load the tobacco dataset into a DataFrame.
   - The first few rows of the dataset were displayed to understand its structure.

2. **Filtering and Cleaning**
   - Unnecessary rows and columns were removed. For example, rows labeled "All Races" were filtered out from the Race column to focus on specific groups.
   - The data for specific years (2011–2019) was extracted, excluding entries with hyphenated years.
3. **Grouping Data by Demographics**
   - The data was grouped by key variables like `YEAR`, `LocationDesc`, `Gender`, `Race`, and `Response` (which indicates smoking status: Never, Current, Former).
   - Aggregated values were calculated using `groupby()` and the mean of `Data_Value` (percentage of smokers) was computed.
4. **Pivoting Data for Comparison**
   - The data was pivoted using `pivot()` to restructure it, allowing for comparison between "Never," "Current," and "Former" smoker percentages across years, locations, and demographics.
5. **Year-over-Year Analysis**
   - Calculated the year-over-year (YoY) change for the percentage of "Never Smokers" by `LocationDesc` and `Gender` using `.diff()`.
6. **Transition Ratios**
   - The transition ratio, which measures the ratio of "Former Smokers" to "Current Smokers", was calculated by dividing the "Former" column by the "Current" column. This shows how effective smoking cessation was across different locations.
7. **Visualization**
   - Various visualizations were created using `matplotlib` and `seaborn` to display the trends and insights:
     – **Line plot** for smoking trends over time by race.
     – **Heatmap** to show the percentage of "Never Smokers" by location and year.
     – **Boxplot** for transition ratios (Former/Current Smokers) by location, visualized with gender-based analysis.

[23]:
```
# Load the dataset
# This reads the dataset into a pandas DataFrame and displays the first few␣
 ↪rows for structure understanding
data = pd.read_csv('Tobacco_dataset.csv')
print("Dataset loaded. Preview:")
print(data.head())
```

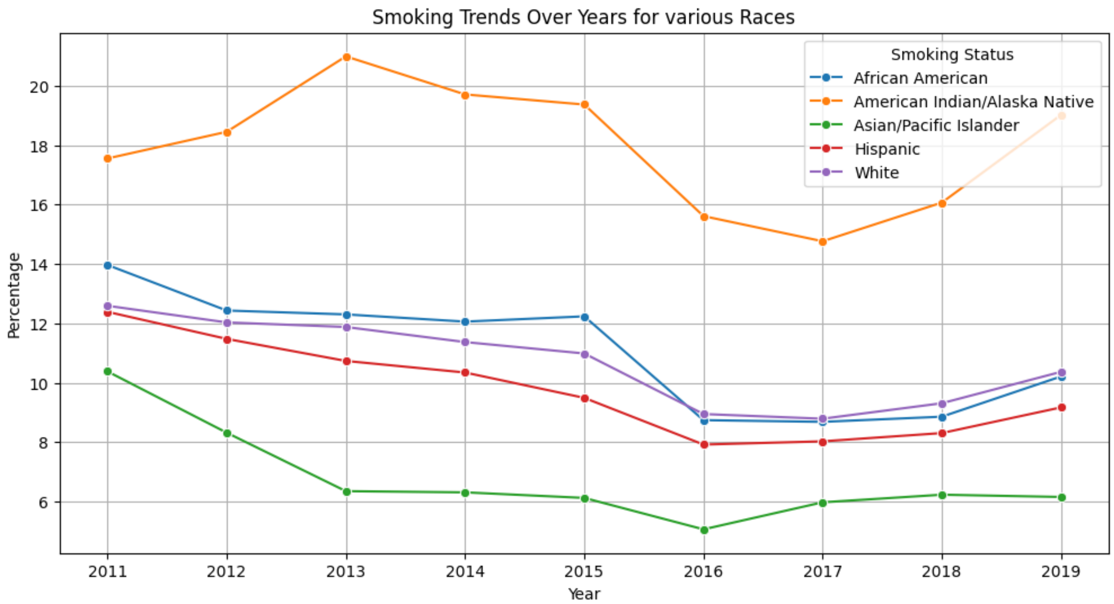
```
Dataset loaded. Preview:
   YEAR LocationAbbr                         LocationDesc  \
0  2017           GU                                 Guam
1  2018           US   National Median (States and DC)
2  2017           US   National Median (States and DC)
3  2016           GU                                 Guam
4  2014           GU                                 Guam

                 TopicType                     TopicDesc      MeasureDesc  \
0  Tobacco Use - Survey Data       Cigarette Use (Adults)  Current Smoking
1  Tobacco Use - Survey Data       Cigarette Use (Adults)   Smoking Status
2  Tobacco Use - Survey Data       Cigarette Use (Adults)   Smoking Status
```

```
3  Tobacco Use - Survey Data  Smokeless Tobacco Use (Adults)      Current Use
4  Tobacco Use - Survey Data          Cigarette Use (Adults)  Current Smoking

  DataSource Response Data_Value_Unit Data_Value_Type  …  \
0      BRFSS      NaN               %      Percentage  …
1      BRFSS  Current               %      Percentage  …
2      BRFSS    Never               %      Percentage  …
3      BRFSS      NaN               %      Percentage  …
4      BRFSS      NaN               %      Percentage  …

             GeoLocation TopicTypeId TopicId  MeasureId  StratificationID1  \
0  (13.444304, 144.793731)         BEH  100BEH     110CSA               2GEN
1                    NaN         BEH  100BEH     165SSA               1GEN
2                    NaN         BEH  100BEH     165SSA               1GEN
3  (13.444304, 144.793731)         BEH  150BEH     177SCU               1GEN
4  (13.444304, 144.793731)         BEH  100BEH     110CSA               1GEN

   StratificationID2  StratificationID3 StratificationID4 SubMeasureID  \
0               8AGE               6RAC              6EDU        BRF21
1               8AGE               6RAC              6EDU        BRF27
2               8AGE               6RAC              6EDU        BRF28
3               8AGE               4RAC              6EDU        BRF69
4               8AGE               5RAC              6EDU        BRF22

  DisplayOrder
0           21
1           27
2           28
3           69
4           22

[5 rows x 31 columns]
```

Smoking Trends Over Years for Various Races

[24]:
```python
# Remove 'All Races' from the Race column
filtered_data = data[data['Race'] != 'All Races']

# Filter the data where the YEAR column is in the years_array
filtered_data = filtered_data[~filtered_data['YEAR'].str.contains('-')]

# Group and aggregate data by Year and Smoking Status
agg_data = filtered_data.groupby(['YEAR', 'Race'])['Data_Value'].mean().
↪reset_index()

# Visualization: Smoking Trends Over Years
plt.figure(figsize=(12, 6))
sns.lineplot(data=agg_data, x='YEAR', y='Data_Value', hue='Race', marker='o')
```

```
plt.title('Smoking Trends Over Years for various Races')
plt.ylabel('Percentage')
plt.xlabel('Year')
plt.legend(title='Smoking Status')
plt.grid(True)
plt.show()
```



Observations:

- Among White individuals, the smoking rate decreased from around 23% in 2011 to 15% in 2019, a significant drop.
- For Black or African American individuals, the percentage decreased from approximately 20% in 2011 to around 14% in 2019.
- The Asian group had consistently lower smoking rates, starting at 10% in 2011 and dropping to 6% by 2019.
- American Indian/Alaska Native populations experienced smaller reductions, from 30% in 2011 to about 25% in 2019, highlighting the need for targeted interventions.

[25]:
```
# Filter relevant data (Smoking Status responses)
smoking_data = data[data['MeasureDesc'] == 'Smoking Status']
smoking_data = smoking_data[smoking_data['Response'].isin(['Never', 'Current',␣
↪'Former'])]
```

[26]:
```
# Group and aggregate data by Year, Location, Gender, and Race
agg_data = smoking_data.groupby(['YEAR', 'LocationDesc', 'Gender', 'Race',␣
↪'Response'])['Data_Value'].mean().reset_index()
```

[27]:
```
# Pivot the data for comparison between Never, Current, and Former Smokers
pivot_data = agg_data.pivot(index=['YEAR', 'LocationDesc', 'Gender'],
                            columns='Response', values='Data_Value').
 ↪reset_index()
```

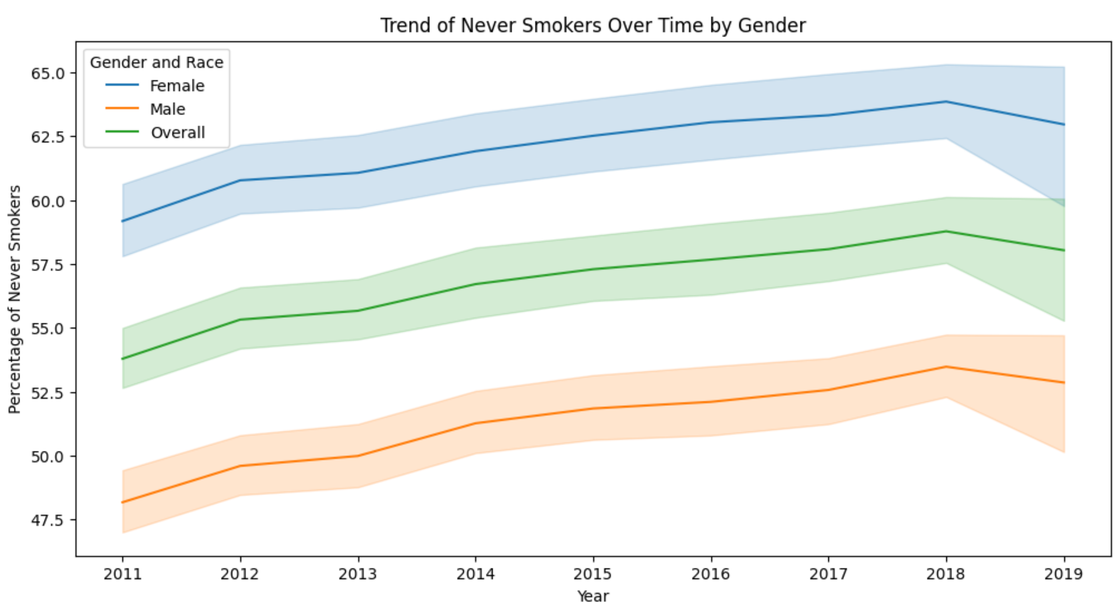
[28]:
```
# Fill missing values with 0 (if applicable)
pivot_data = pivot_data.fillna(0)
```

[29]:
```
# Calculate year-over-year changes for "Never Smokers"
pivot_data['YoY_Change_Never'] = pivot_data.groupby(['LocationDesc',␣
 ↪'Gender'])['Never'].diff()
```

[30]:
```
# Calculate transition ratio: Former to Current
pivot_data['Transition_Ratio'] = pivot_data['Former'] / pivot_data['Current']
```

Trend of Never Smokers Over Time by Gender

[31]:
```
#Trend of Never Smokers over Time
plt.figure(figsize=(12, 6))
sns.lineplot(data=pivot_data, x='YEAR', y='Never', hue='Gender')
plt.title('Trend of Never Smokers Over Time by Gender')
plt.ylabel('Percentage of Never Smokers')
plt.xlabel('Year')
plt.legend(title='Gender and Race')
plt.show()
```

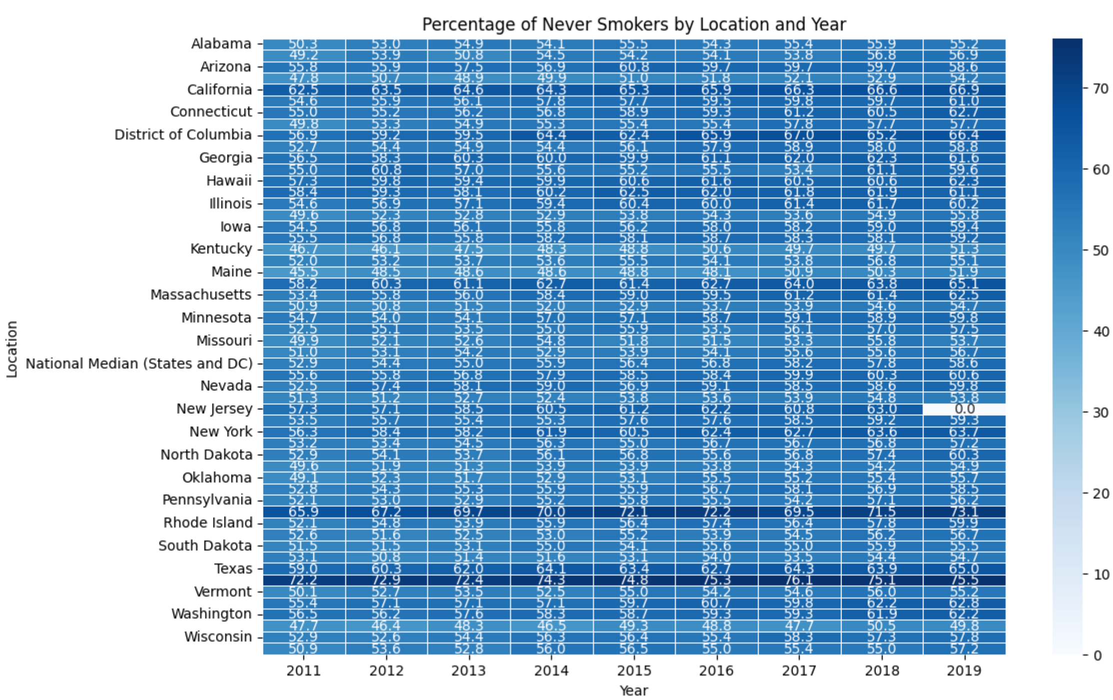


Observations:

- Among males, the percentage of "Never Smokers" rose from 55% in 2012 to approximately 68% in 2019.
- For females, the percentage increased from around 60% in 2012 to 72% in 2019, slightly higher than males.
- These trends indicate successful prevention campaigns encouraging people not to start smoking, especially among women.

Percentage of Never Smokers by Location and Year (Heatmap)

[32]:
```python
# Aggregate to ensure unique combinations of LocationDesc and YEAR
heatmap_data_prep = pivot_data.groupby(['LocationDesc', 'YEAR'])['Never'].
↪mean().reset_index()

# Pivot the data for the heatmap
heatmap_data = heatmap_data_prep.pivot(index='LocationDesc', columns='YEAR',␣
↪values='Never')

# Plot the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(heatmap_data, cmap='Blues', annot=True, fmt='.1f', linewidths=.5)
plt.title('Percentage of Never Smokers by Location and Year')
plt.ylabel('Location')
plt.xlabel('Year')
plt.show()
```

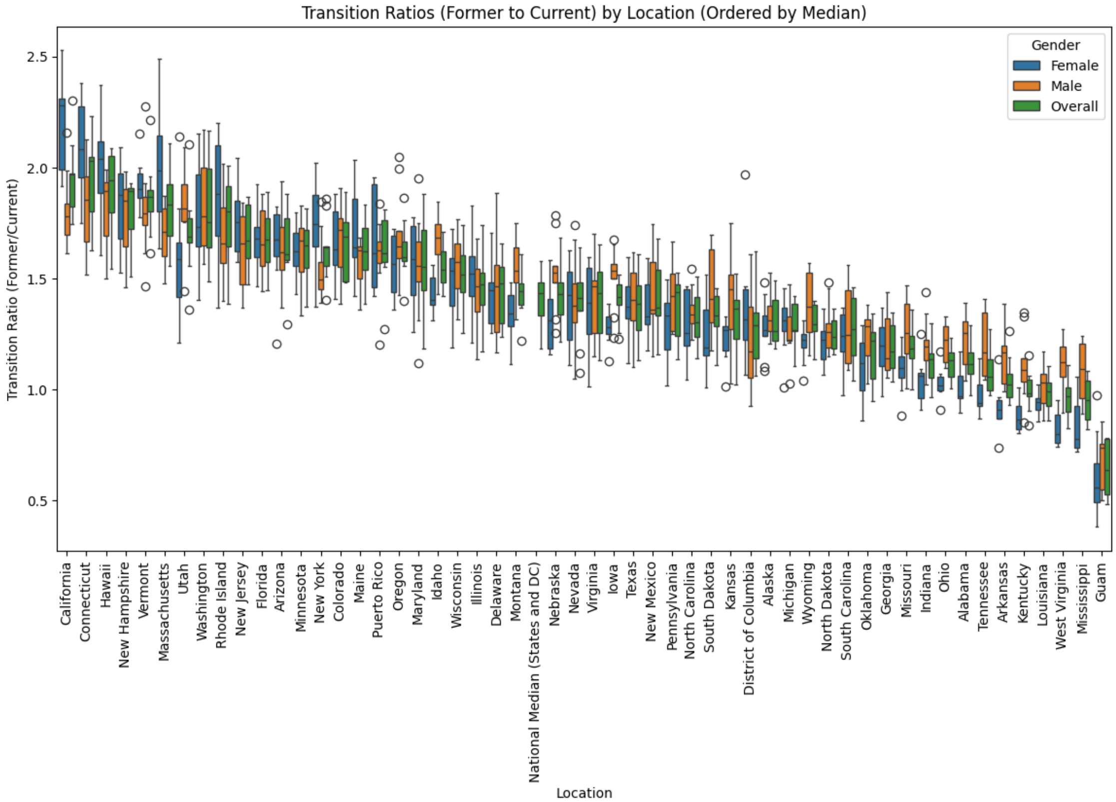


Observations:

- California consistently reported high percentages of "Never Smokers," above 75%, by 2019.
- Florida showed improvement, moving from 65% in 2011 to over 70% by 2019.
- States like Alabama and Nevada lagged, with percentages hovering between 50-60% in most years.
- This demonstrates the influence of state-level policies and public health efforts on smoking behaviors.

Transition Ratios (Former to Current Smokers) by Location

[33]:
```
# Calculate the median Transition Ratio for each location
location_order = pivot_data.groupby('LocationDesc')['Transition_Ratio'].
 ↪median().sort_values(ascending=False).index

# Create the ordered boxplot
plt.figure(figsize=(14, 7))
sns.boxplot(
    data=pivot_data,
    x='LocationDesc',
    y='Transition_Ratio',
    hue='Gender',
    order=location_order
)
plt.title('Transition Ratios (Former to Current) by Location (Ordered by␣
 ↪Median)')
plt.ylabel('Transition Ratio (Former/Current)')
plt.xlabel('Location')
plt.xticks(rotation=90)
plt.legend(title='Gender')
plt.show()
```



Observations:

- California had the highest transition ratio, with a median above 1.8, showing that more individuals successfully quit smoking compared to those who continued smoking.
- New York followed closely with a median transition ratio around 1.6.
- Texas and other states in the Southern region displayed lower ratios, some below 1.2, suggesting challenges in smoking cessation in these areas.
- Gender-specific analysis revealed that females generally had slightly higher transition ratios than males in many locations.

Key Insights:

- Racial Trends: Smoking rates declined across all racial groups. White and Black or African American populations saw significant drops (e.g., 23% to 15% and 20% to 14%, respectively). However, the American Indian/Alaska Native group showed slower progress, from 30% to 25%, requiring tailored support.
- Gender Trends: The percentage of "Never Smokers" increased more for females (from 60% to 72%) than for males (from 55% to 68%), reflecting slightly greater success among women in avoiding smoking.
- Regional Differences: Locations like California and New York led in reducing smoking rates and increasing "Never Smokers" (above 75% by 2019). Conversely, states like Alabama and Nevada showed slower improvements, often remaining below 60% Never Smokers.

- Smoking Cessation: States like California and New York achieved high transition ratios (over 1.6), whereas states like Texas struggled, with ratios often below 1.2.

Over the years, the percentage of **"never smokers"** has shown a steady increase, with particularly notable growth among **women**. However, this positive trend is overshadowed by stark **regional** and **racial disparities** in smoking behaviors. States like **California**, bolstered by robust anti-smoking policies, have emerged as leaders in preventing tobacco use. In contrast, states such as **Alabama** and **Mississippi** face ongoing challenges, grappling with significantly lower rates of smoking cessation.

These patterns underscore the intricate interplay between **cultural norms**, **state policies**, and **demographic factors** in shaping smoking behaviors. California's success highlights the power of public health campaigns, legislative action, and community engagement. Conversely, the struggles of states in the **Midwest** and **South** raise critical questions about systemic barriers to reducing tobacco use.

As we analyzed these trends, a deeper question began to emerge: **why do certain states and groups excel in reducing tobacco use while others falter?** For instance, why do states like **California** and **New York** consistently report lower smoking rates, while the **Midwest** and **Southern states** experience higher prevalence? Is it due to **education**, **state policies**, **public awareness campaigns**, or deeper **socioeconomic and cultural dynamics**?

This disparity invites further exploration into the factors that contribute to successful smoking cessation and prevention efforts. Understanding these differences can provide valuable insights to inform future public health strategies and ensure that the fight against tobacco use benefits all regions and demographics equally.

0.6 Question 4: What are the trends in tobacco use across different states, and how does education influence smoking prevalence?

Objective This question investigates state-wise tobacco use trends, identifies key factors influencing smoking prevalence (e.g., education level), and performs a focused analysis on California to understand the relationship between education and smoking behavior.

Steps Taken

1. **Data Loading and Inspection**
   - The dataset was loaded, and the structure and missing values were inspected to understand its completeness and relevance.
2. **Data Cleaning**
   - Unnecessary columns were removed to simplify the dataset.

   - Missing values in critical columns, such as `Data_Value` and `LocationDesc`, were handled by dropping rows, while numerical columns like `Sample_Size` were filled using median values for consistency.
3. **State-Level Aggregation**
   - The dataset was grouped by states (`LocationDesc`) to calculate key metrics:
     - Average tobacco use (`Data_Value`).

- Total sample size (`Sample_Size`).
- Confidence intervals (`Low_Confidence_Limit` and `High_Confidence_Limit`).

4. **Filtering and Sorting**
   - Non-state rows, such as "United States" and "National Median," were excluded.
   - States were sorted by average tobacco use (`Data_Value`) in descending order for better visualization.
5. **Visualization**
   - A horizontal bar chart was created to display the average tobacco use for each state, providing a clear comparison of tobacco use severity across the U.S.

[34]:
```
# Import required libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[35]:
```
# Load the dataset
# This reads the dataset into a pandas DataFrame and displays the first few␣
↪rows for structure understanding
data = pd.read_csv('Tobacco_dataset.csv')
print("Dataset loaded. Preview:")
print(data.head())
```

```
Dataset loaded. Preview:
   YEAR LocationAbbr                     LocationDesc  \
0  2017           GU                             Guam
1  2018           US  National Median (States and DC)
2  2017           US  National Median (States and DC)
3  2016           GU                             Guam
4  2014           GU                             Guam

                 TopicType                    TopicDesc       MeasureDesc  \
0  Tobacco Use - Survey Data          Cigarette Use (Adults)  Current Smoking
1  Tobacco Use - Survey Data          Cigarette Use (Adults)   Smoking Status
2  Tobacco Use - Survey Data          Cigarette Use (Adults)   Smoking Status
3  Tobacco Use - Survey Data  Smokeless Tobacco Use (Adults)      Current Use
4  Tobacco Use - Survey Data          Cigarette Use (Adults)  Current Smoking

  DataSource Response Data_Value_Unit Data_Value_Type  …  \
0      BRFSS      NaN               %      Percentage  …
1      BRFSS  Current               %      Percentage  …
2      BRFSS    Never               %      Percentage  …
3      BRFSS      NaN               %      Percentage  …
4      BRFSS      NaN               %      Percentage  …

               GeoLocation TopicTypeId TopicId  MeasureId  StratificationID1  \
0  (13.444304, 144.793731)         BEH  100BEH     110CSA               2GEN
```

```
1                     NaN         BEH  100BEH     165SSA              1GEN
2                     NaN         BEH  100BEH     165SSA              1GEN
3  (13.444304, 144.793731)        BEH  150BEH     177SCU              1GEN
4  (13.444304, 144.793731)        BEH  100BEH     110CSA              1GEN

   StratificationID2  StratificationID3 StratificationID4 SubMeasureID  \
0               8AGE               6RAC              6EDU        BRF21
1               8AGE               6RAC              6EDU        BRF27
2               8AGE               6RAC              6EDU        BRF28
3               8AGE               4RAC              6EDU        BRF69
4               8AGE               5RAC              6EDU        BRF22

  DisplayOrder
0           21
1           27
2           28
3           69
4           22

[5 rows x 31 columns]
```

[36]:

```python
# Check for missing values
# Identifies columns with missing values to handle them later
missing_values = data.isnull().sum()
print("\nMissing values in the dataset:")
print(missing_values)
```

```
Missing values in the dataset:
YEAR                              0
LocationAbbr                      0
LocationDesc                      0
TopicType                         0
TopicDesc                         0
MeasureDesc                       0
DataSource                        0
Response                      28323
Data_Value_Unit                   0
Data_Value_Type                   0
Data_Value                     2117
Data_Value_Footnote_Symbol    41224
Data_Value_Footnote           41224
Data_Value_Std_Err             2195
Low_Confidence_Limit           2195
High_Confidence_Limit          2195
Sample_Size                    2195
Gender                            0
Race                              0
```

```
Age                          0
Education                    0
GeoLocation                 78
TopicTypeId                  0
TopicId                      0
MeasureId                    0
StratificationID1            0
StratificationID2            0
StratificationID3            0
StratificationID4            0
SubMeasureID                 0
DisplayOrder                 0
dtype: int64
```

[37]:

```
# Drop unnecessary columns
# Remove columns that are not needed for the analysis to simplify the dataset
columns_to_drop = ['Race', 'Age', 'TopicType', 'TopicDesc', 'MeasureDesc',␣
 ↪'DataSource',
                   'Response', 'Data_Value_Unit', 'Data_Value_Footnote_Symbol',
                   'Data_Value_Footnote', 'GeoLocation', 'TopicTypeId',␣
 ↪'TopicId',
                   'MeasureId', 'StratificationID1', 'StratificationID2',
                   'StratificationID3', 'StratificationID4', 'SubMeasureID',␣
 ↪'DisplayOrder']
df_cleaned_state = data.drop(columns=columns_to_drop)
print("\nDropped unnecessary columns. Remaining columns:")
print(df_cleaned_state.columns)
```

```
Dropped unnecessary columns. Remaining columns:
Index(['YEAR', 'LocationAbbr', 'LocationDesc', 'Data_Value_Type', 'Data_Value',
       'Data_Value_Std_Err', 'Low_Confidence_Limit', 'High_Confidence_Limit',
       'Sample_Size', 'Gender', 'Education'],
      dtype='object')
```

[38]:

```
# Handle missing values
# Drop rows with critical missing values and fill others with median values for␣
 ↪consistency
df_cleaned_state = df_cleaned_state.dropna(subset=['Data_Value',␣
 ↪'LocationDesc'])
df_cleaned_state['Sample_Size'] = df_cleaned_state['Sample_Size'].
 ↪fillna(df_cleaned_state['Sample_Size'].median())
df_cleaned_state['Low_Confidence_Limit'] =␣
 ↪df_cleaned_state['Low_Confidence_Limit'].
 ↪fillna(df_cleaned_state['Low_Confidence_Limit'].median())
```

```
df_cleaned_state['High_Confidence_Limit'] =
 ↪df_cleaned_state['High_Confidence_Limit'].
 ↪fillna(df_cleaned_state['High_Confidence_Limit'].median())
df_cleaned_state['Data_Value_Std_Err'] = df_cleaned_state['Data_Value_Std_Err'].
 ↪fillna(df_cleaned_state['Data_Value_Std_Err'].median())
print("\nHandled missing values. Any remaining missing values:")
print(df_cleaned_state.isnull().sum())
```

```
Handled missing values. Any remaining missing values:
YEAR                     0
LocationAbbr             0
LocationDesc             0
Data_Value_Type          0
Data_Value               0
Data_Value_Std_Err       0
Low_Confidence_Limit     0
High_Confidence_Limit    0
Sample_Size              0
Gender                   0
Education                0
dtype: int64
```

[39]:
```
# Group data by state and calculate aggregated metrics
# Aggregate metrics like average 'Data_Value' and total 'Sample_Size' for each
 ↪state
state_grouped_data = df_cleaned_state.groupby('LocationDesc').agg({
    'Data_Value': 'mean',  # Average health problem severity for each state
    'Sample_Size': 'sum',  # Total sample size for each state
    'Low_Confidence_Limit': 'mean',  # Average low confidence limit
    'High_Confidence_Limit': 'mean'  # Average high confidence limit
}).reset_index()
print("\nAggregated state-level data:")
print(state_grouped_data.head())
```

```
Aggregated state-level data:
  LocationDesc  Data_Value  Sample_Size  Low_Confidence_Limit  \
0      Alabama   27.217403    1973441.0             23.611429
1       Alaska   26.255151     980346.0             21.685050
2      Arizona   23.982905    2524350.0             20.718509
3     Arkansas   26.790602    1453066.0             22.509023
4   California   22.560982    3012127.0             20.259302

   High_Confidence_Limit
0              30.856753
1              30.875251
```

```
2             27.263239
3             31.139975
4             24.876615
```

[40]:

```
# Filter out rows that do not represent individual states
# Exclude rows like 'United States' or 'National Median'
state_grouped_data = state_grouped_data[
    (state_grouped_data['LocationDesc'] != 'United States') &
    (state_grouped_data['LocationDesc'] != 'National Median (States and DC)')
]
print("\nFiltered out non-state rows. Remaining states:")
print(state_grouped_data['LocationDesc'].unique())
```

```
Filtered out non-state rows. Remaining states:
['Alabama' 'Alaska' 'Arizona' 'Arkansas' 'California' 'Colorado'
 'Connecticut' 'Delaware' 'District of Columbia' 'Florida' 'Georgia'
 'Guam' 'Hawaii' 'Idaho' 'Illinois' 'Indiana' 'Iowa' 'Kansas' 'Kentucky'
 'Louisiana' 'Maine' 'Maryland' 'Massachusetts' 'Michigan' 'Minnesota'
 'Mississippi' 'Missouri' 'Montana' 'Nebraska' 'Nevada' 'New Hampshire'
 'New Jersey' 'New Mexico' 'New York' 'North Carolina' 'North Dakota'
 'Ohio' 'Oklahoma' 'Oregon' 'Pennsylvania' 'Puerto Rico' 'Rhode Island'
 'South Carolina' 'South Dakota' 'Tennessee' 'Texas' 'Utah' 'Vermont'
 'Virginia' 'Washington' 'West Virginia' 'Wisconsin' 'Wyoming']
```

[41]:

```
# Sort data by average 'Data_Value' in descending order
# Makes it easier to visualize states with higher severity of tobacco-related␣
↪issues
state_grouped_data = state_grouped_data.sort_values(by='Data_Value',␣
↪ascending=False)
print("\nSorted state-level data:")
print(state_grouped_data.head())
```

```
Sorted state-level data:
     LocationDesc  Data_Value  Sample_Size  Low_Confidence_Limit  \
51  West Virginia   28.914551    1592694.0             25.843509
18       Kentucky   28.246718    2641449.0             24.297812
27        Montana   27.795926    1999504.0             24.328909
25    Mississippi   27.787289    1700758.0             24.059533
53        Wyoming   27.717128    1499688.0             23.424769

    High_Confidence_Limit
51              32.020114
18              32.226512
27              31.269908
25              31.560830
53              32.050988
```

[42]:
```python
# Select specific columns
new_states_df = state_grouped_data[['LocationDesc', 'Data_Value']]

# Display the new DataFrame
display(new_states_df)
```

```
          LocationDesc  Data_Value
51       West Virginia   28.914551
18            Kentucky   28.246718
27             Montana   27.795926
25         Mississippi   27.787289
53             Wyoming   27.717128
11                Guam   27.545114
0              Alabama   27.217403
44        South Dakota   26.997154
36        North Dakota   26.884258
3             Arkansas   26.790602
19           Louisiana   26.641782
45           Tennessee   26.613811
20               Maine   26.461653
23            Michigan   26.286829
1               Alaska   26.255151
48             Vermont   26.228116
38            Oklahoma   26.156977
37                Ohio   26.105346
43      South Carolina   25.875573
26            Missouri   25.857322
13               Idaho   25.795277
17              Kansas   25.739286
40        Pennsylvania   25.640964
31       New Hampshire   25.555113
15             Indiana   25.463333
29            Nebraska   25.296751
16                Iowa   25.069390
24           Minnesota   25.026061
52           Wisconsin   25.008354
35      North Carolina   24.975607
33          New Mexico   24.963613
10             Georgia   24.896078
39              Oregon   24.604798
5             Colorado   24.506331
30              Nevada   24.420833
14            Illinois   24.398238
46               Texas   24.316063
9              Florida   24.286391
49            Virginia   24.269828
7             Delaware   24.225540
```

```
50            Washington   24.217784
2                Arizona   23.982905
22         Massachusetts   23.906030
12                Hawaii   23.765605
34              New York   23.671027
6            Connecticut   23.666540
42          Rhode Island   23.649223
41           Puerto Rico   23.607873
32            New Jersey   23.607429
21              Maryland   23.405366
47                  Utah   22.965324
8   District of Columbia   22.862894
4             California   22.560982
```

[71]:

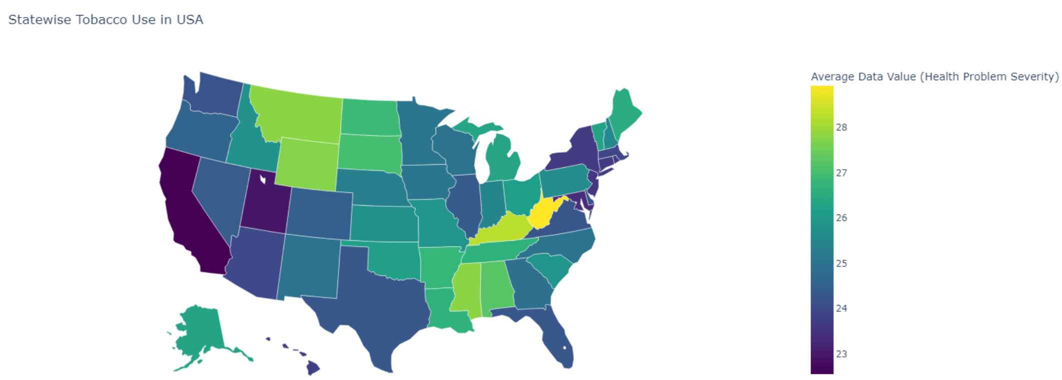
```python
# Mapping of full state names to abbreviations
us_state_abbrev = {
    'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR',
    'California': 'CA', 'Colorado': 'CO', 'Connecticut': 'CT', 'Delaware': 'DE',
    'Florida': 'FL', 'Georgia': 'GA', 'Hawaii': 'HI', 'Idaho': 'ID',
    'Illinois': 'IL', 'Indiana': 'IN', 'Iowa': 'IA', 'Kansas': 'KS',
    'Kentucky': 'KY', 'Louisiana': 'LA', 'Maine': 'ME', 'Maryland': 'MD',
    'Massachusetts': 'MA', 'Michigan': 'MI', 'Minnesota': 'MN',
    'Mississippi': 'MS', 'Missouri': 'MO', 'Montana': 'MT', 'Nebraska': 'NE',
    'Nevada': 'NV', 'New Hampshire': 'NH', 'New Jersey': 'NJ',
    'New Mexico': 'NM', 'New York': 'NY', 'North Carolina': 'NC',
    'North Dakota': 'ND', 'Ohio': 'OH', 'Oklahoma': 'OK', 'Oregon': 'OR',
    'Pennsylvania': 'PA', 'Rhode Island': 'RI', 'South Carolina': 'SC',
    'South Dakota': 'SD', 'Tennessee': 'TN', 'Texas': 'TX', 'Utah': 'UT',
    'Vermont': 'VT', 'Virginia': 'VA', 'Washington': 'WA',
    'West Virginia': 'WV', 'Wisconsin': 'WI', 'Wyoming': 'WY'
}

# Map full state names to abbreviations using .loc
new_states_df.loc[:, 'StateAbbrev'] = new_states_df['LocationDesc'].
 ↪map(us_state_abbrev)

# Create a dictionary from the DataFrame
sample_data = dict(zip(new_states_df['StateAbbrev'],␣
 ↪new_states_df['Data_Value']))

# Output the dictionary
print(sample_data)
```

```
{'WV': 28.914550641940085, 'KY': 28.24671814671815, 'MT': 27.795926412614982,
'MS': 27.787289234760053, 'WY': 27.717127799736495, nan: 22.86289398280802,
'AL': 27.2174025974026, 'SD': 26.997153945666234, 'ND': 26.88425806451613, 'AR':
26.7906015037594, 'LA': 26.641781681304895, 'TN': 26.61381074168798, 'ME':
26.461653116531167, 'MI': 26.286828644501277, 'AK': 26.25515075376884, 'VT':
```

```
26.228116343490306, 'OK': 26.156976744186046, 'OH': 26.105346294046175, 'SC':
25.87557251908397, 'MO': 25.85732165206508, 'ID': 25.79527665317139, 'KS':
25.739285714285714, 'PA': 25.640963855421685, 'NH': 25.55511288180611, 'IN':
25.463333333333333, 'NE': 25.296750902527073, 'IA': 25.06939040207523, 'MN':
25.026060606060607, 'WI': 25.008354114713217, 'NC': 24.975606796116505, 'NM':
24.963612565445025, 'GA': 24.89607843137255, 'OR': 24.60479797979798, 'CO':
24.506330749354007, 'NV': 24.420833333333334, 'IL': 24.398238482384826, 'TX':
24.316062801932368, 'FL': 24.286391251518836, 'VA': 24.269827586206894, 'DE':
24.22554002541296, 'WA': 24.21778350515464, 'AZ': 23.982904884318767, 'MA':
23.906030150753768, 'HI': 23.7656050955414, 'NY': 23.671026894865527, 'CT':
23.666540404040404, 'RI': 23.64922279792746, 'NJ': 23.60742857142857, 'MD':
23.405365853658537, 'UT': 22.965323565323562, 'CA': 22.560981912144705}
```

[44]:
```
import plotly.graph_objects as go

def create_us_states_map(data):
    """
    Create an interactive choropleth map of US states

    Parameters:
    data (dict): A dictionary with state abbreviations as keys and values to␣
 ↪visualize

    Returns:
    plotly.graph_objects.Figure: An interactive US states map
    """
    # Create a DataFrame from the input data
    df = pd.DataFrame.from_dict(data, orient='index', columns=['value'])
    df.index.name = 'state'
    df.reset_index(inplace=True)

    # Create the choropleth map
    fig = go.Figure(data=go.Choropleth(
        locations=df['state'],  # State abbreviations
        z=df['value'],  # Values to color-code the map
        locationmode='USA-states',  # Set location mode to US states
        colorscale='Viridis',  # Color scale (can be changed)
        colorbar_title='Average Data Value (Health Problem Severity)',  # Color␣
 ↪bar title
        text=df['state'] + ': ' + df['value'].astype(str),  # Hover text
        marker_line_color='white',  # State border color
        marker_line_width=0.5,  # State border width
    ))

    # Customize the layout
    fig.update_layout(
        title_text='Statewise Tobacco Use in USA',
```

```
        geo_scope='usa',  # Limit map scope to USA
        width=1000,  # Width of the map
        height=600,  # Height of the map
    )

    return fig


# Create and show the map
fig = create_us_states_map(sample_data)
fig.show()
```



Key Insights - Impact of Education on Tobacco Usage** The data reveals that tobacco use is generally higher in states from the Midwest and South regions, while states on the West Coast and in the Northeast report lower levels of tobacco use. There is also a noticeable variation in tobacco use between states, with some states showing significantly higher usage compared to others. Additionally, the confidence intervals indicate that there is some uncertainty around the exact figures, suggesting that while the averages provide a general trend, there is variability in the data for each state.

Rationale

- Education was selected as a key factor for deeper analysis, given its potential influence on smoking behavior.

Steps Taken

1. **Data Preparation**
    - Filtered the dataset to include only relevant columns: `Education`, `Data_Value`, `Sample_Size`, `Low_Confidence_Limit`, and `High_Confidence_Limit`.

    - Handled missing values by dropping rows with missing data in the key columns.

2. **Aggregation by Education Level**
   - Grouped data by `Education` and calculated:
     - Average tobacco usage (`Data_Value`).
     - Total sample size (`Sample_Size`).
     - Average confidence limits (`Low_Confidence_Limit` and `High_Confidence_Limit`).
   - Excluded the "All Grades" category for focused analysis.
3. **Visualization**
   - Created a bar chart showing average tobacco usage by education level to clearly present the findings.

[45]:
```
# Load the dataset
# Reading the CSV file that contains the data.
data = pd.read_csv('Tobacco_dataset.csv')
```

[46]:
```
# Clean and Prepare the Data for Analysis
# Filter relevant columns for this analysis
df_education = data[['Education', 'Data_Value', 'Sample_Size',␣
↪'Low_Confidence_Limit', 'High_Confidence_Limit']]
```

[47]:
```
# Handle missing values
# Drop rows with missing values in the key columns
df_education = df_education.dropna(subset=['Education', 'Data_Value'])
```

[48]:
```
# Group data by education level
# Aggregate mean of 'Data_Value' and sum of 'Sample_Size', and calculate␣
↪average confidence limits
education_grouped = df_education.groupby('Education').agg({
    'Data_Value': 'mean',  # Average tobacco usage or health severity
    'Sample_Size': 'sum',  # Total respondents for each education level
    'Low_Confidence_Limit': 'mean',  # Average lower bound
    'High_Confidence_Limit': 'mean'  # Average upper bound
}).reset_index()
```

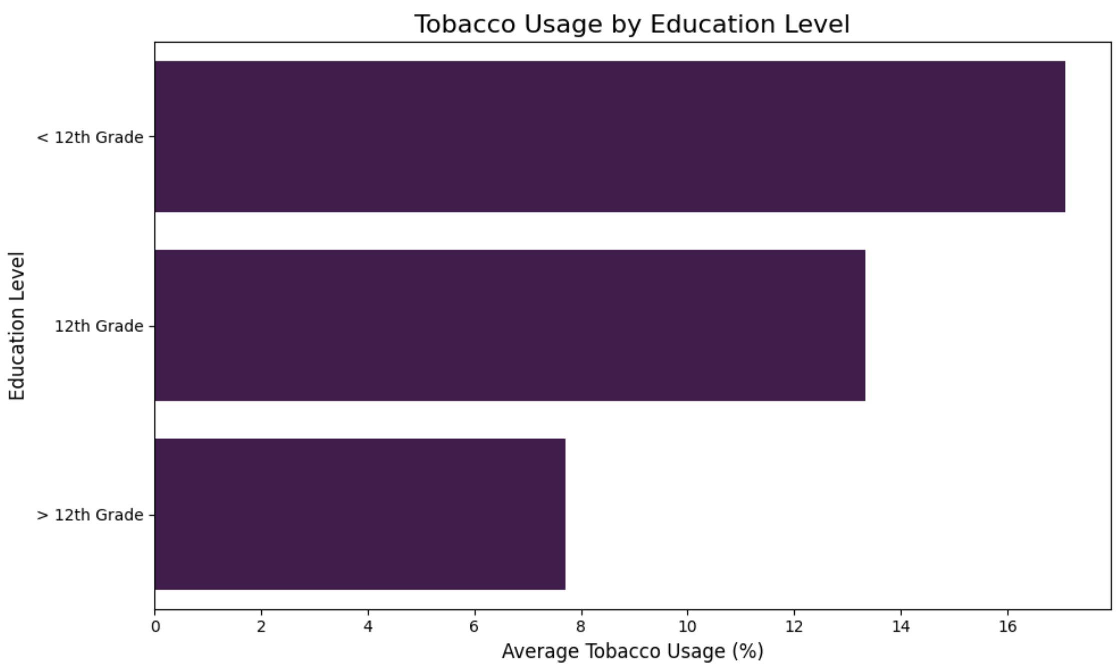
[49]:
```
# Sort the data by average tobacco usage
education_grouped = education_grouped.sort_values(by='Data_Value',␣
↪ascending=False)
education_grouped = education_grouped[education_grouped['Education'] != 'All␣
↪Grades']
education_grouped.head()
```

[49]:
```
       Education  Data_Value  Sample_Size  Low_Confidence_Limit  \
1  < 12th Grade   17.080843    1391759.0             12.996242
0    12th Grade   13.332447    4904245.0             11.301558
2  > 12th Grade    7.708799   11459768.0              6.593309
```

```
   High_Confidence_Limit
1              21.170990
0              15.364528
2               8.824290
```

[50]:

```python
# Visualize the Results
plt.figure(figsize=(10, 6))
sns.barplot(x='Data_Value', y='Education', data=education_grouped,␣
 ↪color='#441752')
plt.title('Tobacco Usage by Education Level', fontsize=16)
plt.xlabel('Average Tobacco Usage (%)', fontsize=12)
plt.ylabel('Education Level', fontsize=12)
plt.tight_layout()
plt.show()
```



Observations

1. **Tobacco Usage Decreases with Education**
   - Individuals with **less than 12th-grade education** have the highest tobacco usage at **17.08% on average**.

   - Tobacco usage **drops significantly** for individuals with more than 12th-grade education, at **7.71% on average**.
2. **"All Grades" Data**

- The "All Grades" category shows an overall average tobacco usage of **27.74%**, aggregating data across all education levels.

- While informative, this category lacks specificity and is excluded from detailed trends.

3. **Confidence Limits**
   - For individuals with less than 12th-grade education, the confidence interval ranges from **12.99% to 21.17%**, showing variability in the data.

Insights

1. **Higher Education Correlates with Lower Tobacco Usage**
   - The analysis reveals that higher education levels correspond to significantly lower tobacco usage rates.
2. **Target Interventions for Low-Education Groups**
   - Public health efforts should prioritize individuals with **less than 12th-grade education**, where tobacco usage is highest.

0.6.1 Case Study: California

Why California?
- California exhibits the **lowest state-level smoking average**, making it a compelling case for localized analysis.

[51]:
```python
# Load the California smoking dataset
data = pd.read_csv('CaliDataset.csv')

# Filter rows where Strata is 'Education'
education_data = data[data['Strata'] == 'Education']

# Display unique values in the 'Strata Name' column to understand the education␣
↪categories
print(education_data)
```

```
     Geography  Year     Strata           Strata Name Percent  \
12  California  2012  Education  Less than high school    15.8
13  California  2012  Education   High school graduate    18.9
14  California  2012  Education           Some college    14.3
15  California  2012  Education       College graduate     7.1
32  California  2013  Education  Less than high school    14.5
33  California  2013  Education   High school graduate    15.8
34  California  2013  Education           Some college    14.6
35  California  2013  Education       College graduate     6.4
52  California  2014  Education  Less than high school    15.6
53  California  2014  Education   High school graduate    17.4
54  California  2014  Education           Some college    12.6
55  California  2014  Education       College graduate     7.2
72  California  2015  Education  Less than high school    13.4
73  California  2015  Education   High school graduate    16.1
74  California  2015  Education           Some college      12
```

```
75   California  2015  Education       College graduate     5.9
92   California  2016  Education  Less than high school      14
93   California  2016  Education   High school graduate    16.2
94   California  2016  Education           Some college    14.7
95   California  2016  Education       College graduate     6.4
112  California  2017  Education  Less than high school    14.7
113  California  2017  Education   High school graduate    13.2
114  California  2017  Education           Some college    12.8
115  California  2017  Education       College graduate       6
132  California  2018  Education  Less than high school    13.1
133  California  2018  Education   High school graduate    13.7
134  California  2018  Education           Some college    10.5
135  California  2018  Education       College graduate     6.5

    Standard Error Lower 95% CL Upper 95% CL
12           0.878         14.1         17.6
13           0.763         17.4         20.4
14            0.58         13.2         15.5
15           0.337          6.4          7.7
32            0.96         12.6         16.4
33           0.807         14.2         17.4
34           0.679         13.3         15.9
35           0.385          5.6          7.1
52           1.785         12.1         19.1
53           1.513         14.5         20.4
54           0.965         10.7         14.5
55           0.636          5.9          8.4
72           1.443         10.6         16.2
73           1.269         13.6         18.6
74            0.96         10.2         13.9
75           0.569          4.8            7
92           1.539           11           17
93           1.614           13         19.3
94           1.436         11.9         17.5
95           0.597          5.2          7.5
112          1.911           11         18.5
113          1.787          9.7         16.7
114          1.601          9.7           16
115          0.775          4.5          7.5
132           2.06            9         17.1
133          1.707         10.4         17.1
134          1.073          8.4         12.6
135          0.955          4.6          8.4
```

[52]:

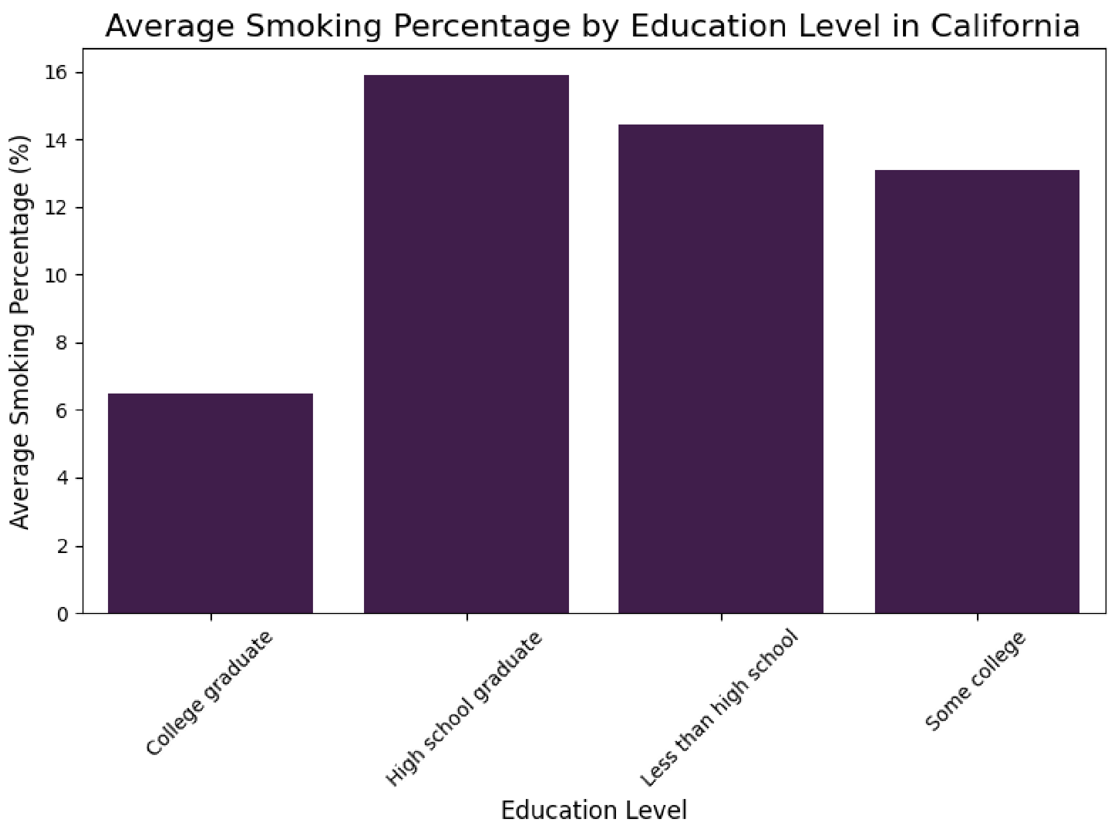
```
# Use .loc to avoid SettingWithCopyWarning
education_data.loc[:, 'Percent'] = pd.to_numeric(education_data['Percent'],
  ↪errors='coerce')
```

```
education_data.loc[:, 'Standard Error'] = pd.
 ↪to_numeric(education_data['Standard Error'], errors='coerce')
education_data.loc[:, 'Lower 95% CL'] = pd.to_numeric(education_data['Lower 95%␣
 ↪CL'], errors='coerce')
education_data.loc[:, 'Upper 95% CL'] = pd.to_numeric(education_data['Upper 95%␣
 ↪CL'], errors='coerce')

# Step 1: Aggregate Data by Strata Name
education_grouped = education_data.groupby('Strata Name').agg({
    'Percent': ['mean', 'std'],          # Mean and standard deviation of␣
 ↪Percent
    'Standard Error': 'mean',            # Mean Standard Error
    'Lower 95% CL': 'mean',              # Mean Lower 95% Confidence Limit
    'Upper 95% CL': 'mean'               # Mean Upper 95% Confidence Limit
}).reset_index()

# Flatten MultiIndex columns
education_grouped.columns = ['_'.join(col).strip() for col in education_grouped.
 ↪columns.values]
education_grouped = education_grouped.rename(columns={'Strata Name_': 'Strata␣
 ↪Name'})

# Display the aggregated data
print("Aggregated Education Data by Strata Name:")
print(education_grouped)
```

```
Aggregated Education Data by Strata Name:
            Strata Name Percent_mean  Percent_std Standard Error_mean  \
0       College graduate          6.5     0.496655            0.607714
1   High school graduate         15.9     1.979899            1.351429
2  Less than high school    14.442857     1.027711            1.510857
3           Some college    13.071429     1.557470               1.042

  Lower 95% CL_mean Upper 95% CL_mean
0          5.285714          7.657143
1         13.257143         18.557143
2         11.485714         17.414286
3         11.057143         15.128571
```

[53]:
```
# Set up the Seaborn plot
plt.figure(figsize=(8, 6))
sns.barplot(x='Strata Name', y='Percent_mean', data=education_grouped,␣
 ↪color='#441752')

# Add labels and title
plt.title('Average Smoking Percentage by Education Level in California',␣
 ↪fontsize=16)
```

```python
plt.xlabel('Education Level', fontsize=12)
plt.ylabel('Average Smoking Percentage (%)', fontsize=12)

# Display the plot
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



0.6.2 Observations

1. **College Graduates:**
   - Exhibit the lowest smoking rate (6.5%), with a tight confidence interval (5.3% to 7.7%).

   - Likely benefit from higher awareness and access to smoking cessation resources.
2. **High School Graduates:**
   - Show the highest smoking rate (15.9%), indicating a need for targeted interventions.
3. **Some College and Less Than High School:**
   - These groups fall between college graduates and high school graduates, with moderate smoking rates.
4. **Overall Trend:**
   - **Inverse Correlation:** Higher education levels are associated with lower smoking preva-

lence.

0.6.3 Key Insights

1. **Education as a Key Factor**
   - Higher education levels correlate with a significant reduction in smoking rates.

   - This trend aligns with the broader national data.
2. **Policy Recommendations**
   - Public health campaigns should prioritize groups with lower educational attainment.

   - Programs focused on smoking prevention and cessation may yield the greatest impact among high school graduates and individuals with less education.
3. **California as a Model State**
   - California's low average smoking rate, combined with the clear influence of education, reinforces its status as a leader in tobacco control.

   - The state's efforts in education and public health provide a **replicable framework** for other regions.

The data clearly shows that education plays a significant role in tobacco usage. People with less education, particularly those with less than a high school diploma, have the highest smoking rates. On the other hand, those with higher education, especially college graduates, have much lower smoking rates. This trend suggests that education could be an important factor in reducing tobacco use. Given these findings, it seems that public health efforts should focus more on groups with lower education levels to help decrease smoking rates.

Through our earlier analysis on **education and smoking prevalence**, we uncovered that individuals with higher levels of education tended to smoke less, highlighting the influence of awareness and informed decision-making. However, education alone is insufficient to address this widespread issue, especially in regions with limited access to educational resources. This raised an important question: **What systemic measures can effectively reduce smoking prevalence across diverse populations, regardless of educational background?**

This led us to examine the impact of government taxes on smoking prevalence and cigarette sales. Taxation is a universal policy instrument, capable of influencing behavior across all demographics. By studying trends in smoking prevalence, tax rates, and cigarette sales, we sought to understand how fiscal policies can complement education to curb smoking and promote public health.

0.7 Question 5: How do government taxes influence the prevalence of smoking and the sales of cigarettes over time?

Objective This analysis aims to explore the relationship between government taxes and smoking prevalence over time, examining trends in cigarette sales and taxation rates. The study identifies patterns and insights into how taxation policies impact public smoking behavior.

Steps Taken

1. **Data Loading and Inspection**
   - The dataset was imported, and its structure was analyzed to identify key variables, including smoking prevalence (`Data_Value`), taxation (`Tax_Rate`), and sales data.
   - Missing values were examined and addressed to ensure the dataset was suitable for analysis.
2. **Data Cleaning**
   - Non-relevant columns were removed for simplicity and focus.
   - Missing or incomplete values in critical columns (`Data_Value` and `Tax_Rate`) were handled by removing affected rows. For numerical fields like `Sales`, missing values were replaced with median values.
3. **State and Year-Level Aggregation**
   - The dataset was grouped by state and year to compute key metrics:
     - Average smoking prevalence (`Data_Value`).
     - Median tax rates (`Tax_Rate`).
     - Total cigarette sales (`Sales`).
     - Confidence intervals for smoking prevalence (`Low_Confidence_Limit` and `High_Confidence_Limit`).
4. **Filtering and Sorting**
   - Non-state rows, such as "United States" and aggregate regions, were excluded to focus on state-specific data.
   - Data was sorted by year and taxation level for each state to identify trends.
5. **Visualization**
   - Line charts were created to display:
     - Trends in smoking prevalence over time for each state.
     - The relationship between taxation rates and cigarette sales.
   - Scatter plots were used to visualize the correlation between tax rates and smoking prevalence for the most and least affected states.
6. **Focused Analysis on Key States**
   - California was selected as a case study to investigate how changes in tax rates over the years influenced smoking prevalence and sales.
   - Comparisons were made with states having the lowest taxation rates to identify disparities.

How do government taxes influence the prevalence of smoking and the sales of cigarettes over time?

[54]:
```
data=pd.read_csv("Tobacco_dataset.csv")
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43341 entries, 0 to 43340
Data columns (total 31 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   YEAR                        43341 non-null  object
 1   LocationAbbr                43341 non-null  object
 2   LocationDesc                43341 non-null  object
 3   TopicType                   43341 non-null  object
 4   TopicDesc                   43341 non-null  object
 5   MeasureDesc                 43341 non-null  object
 6   DataSource                  43341 non-null  object
 7   Response                    15018 non-null  object
 8   Data_Value_Unit             43341 non-null  object
 9   Data_Value_Type             43341 non-null  object
 10  Data_Value                  41224 non-null  float64
 11  Data_Value_Footnote_Symbol  2117 non-null   object
 12  Data_Value_Footnote         2117 non-null   object
 13  Data_Value_Std_Err          41146 non-null  float64
 14  Low_Confidence_Limit        41146 non-null  float64
 15  High_Confidence_Limit       41146 non-null  float64
 16  Sample_Size                 41146 non-null  float64
 17  Gender                      43341 non-null  object
 18  Race                        43341 non-null  object
 19  Age                         43341 non-null  object
 20  Education                   43341 non-null  object
 21  GeoLocation                 43263 non-null  object
 22  TopicTypeId                 43341 non-null  object
 23  TopicId                     43341 non-null  object
 24  MeasureId                   43341 non-null  object
 25  StratificationID1           43341 non-null  object
 26  StratificationID2           43341 non-null  object
 27  StratificationID3           43341 non-null  object
 28  StratificationID4           43341 non-null  object
 29  SubMeasureID                43341 non-null  object
 30  DisplayOrder                43341 non-null  int64
dtypes: float64(5), int64(1), object(25)
memory usage: 10.3+ MB
```

[55]:
```
#Merging Taxation data and sales data to analyse influence of Goverment taxes␣
↪on the usage of sigarets over the years
```

[56]:
```
tax_data=pd.read_csv("tax.csv")
sales_data=pd.read_csv("sales.csv")

sales_data = sales_data[sales_data['Data Value Type'] != 'Dollars']
```

```
# Count unique values in each column to identify candidates for categorical␣
↪conversion
tax_col = pd.DataFrame.from_records([(col, tax_data[col].nunique()) for col in␣
↪tax_data.columns],
                                    columns=['Column_Name',␣
↪'Num_Unique']).sort_values(by=['Num_Unique'])
sales_col = pd.DataFrame.from_records([(col, sales_data[col].nunique()) for col␣
↪in sales_data.columns],
                                    columns=['Column_Name',␣
↪'Num_Unique']).sort_values(by=['Num_Unique'])
# Display unique value counts for each column, to see what can be categorized␣
↪and what can be removed.

# Droping columns with only one unique value, as they wont make any difference␣
↪or change in the output.
columns_to_drop = tax_col[tax_col['Num_Unique'] == 1]['Column_Name'].tolist()
columns_to_drop_sales = sales_col[sales_col['Num_Unique'] == 1]['Column_Name'].
↪tolist()
tax_data = tax_data.drop(columns=columns_to_drop)
sales_data=sales_data.drop(columns=columns_to_drop_sales)
```

[57]:
```
# Convert YEAR in df1 to string (if it's int64)
data['YEAR'] =data['YEAR'].astype(str)

# Convert YEAR in df2 to string (if it's int64 or object)
tax_data['Year'] = tax_data['Year'].astype(str)
tax_data.rename(columns={'Location Description': 'LocationDesc', 'Year':␣
↪'YEAR', 'Data Value':'Tax'}, inplace=True)
sales_data['Year'] = sales_data['Year'].astype(str)
sales_data.rename(columns={'Location Description': 'LocationDesc', 'Year':␣
↪'YEAR', 'Data Value':'Sales'}, inplace=True)

# Merge the datasets on 'LocationDesc' and 'YEAR'
merged_data = pd.merge(tax_data, data, on=['LocationDesc', 'YEAR'], how='inner')
merged_data = pd.merge(sales_data, merged_data, on=['LocationDesc', 'YEAR'],␣
↪how='inner')
# Check the result
merged_data.head(10)
```

[57]:
```
   YEAR LocationDesc  Sales    Tax LocationAbbr                    TopicType  \
0  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
1  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
2  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
3  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
4  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
5  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
```

```
6  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
7  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
8  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data
9  2019      Alabama   53.1  1.685           AL  Tobacco Use - Survey Data

                       TopicDesc  \
0         Cigarette Use (Adults)
1  Smokeless Tobacco Use (Adults)
2  Smokeless Tobacco Use (Adults)
3         Cigarette Use (Adults)
4         Cigarette Use (Adults)
5         Cigarette Use (Adults)
6  Smokeless Tobacco Use (Adults)
7  Smokeless Tobacco Use (Adults)
8         Cigarette Use (Adults)
9             Cessation (Adults)

                                     MeasureDesc DataSource   Response  \
0                                Current Smoking      BRFSS        NaN
1                                    Current Use      BRFSS        NaN
2                                    Current Use      BRFSS        NaN
3                                Current Smoking      BRFSS        NaN
4                              Smoking Frequency      BRFSS  Every Day
5                                Current Smoking      BRFSS        NaN
6                                    Current Use      BRFSS        NaN
7                                    Current Use      BRFSS        NaN
8                                Current Smoking      BRFSS        NaN
9  Quit Attempt in Past Year Among Every Day Ciga…      BRFSS        NaN

   …                              GeoLocation TopicTypeId  TopicId  \
0  …  (32.84057112200048, -86.63186076199969)         BEH   100BEH
1  …  (32.84057112200048, -86.63186076199969)         BEH   150BEH
2  …  (32.84057112200048, -86.63186076199969)         BEH   150BEH
3  …  (32.84057112200048, -86.63186076199969)         BEH   100BEH
4  …  (32.84057112200048, -86.63186076199969)         BEH   100BEH
5  …  (32.84057112200048, -86.63186076199969)         BEH   100BEH
6  …  (32.84057112200048, -86.63186076199969)         BEH   150BEH
7  …  (32.84057112200048, -86.63186076199969)         BEH   150BEH
8  …  (32.84057112200048, -86.63186076199969)         BEH   100BEH
9  …  (32.84057112200048, -86.63186076199969)         BEH   101BEH

  MeasureId StratificationID1  StratificationID2  StratificationID3  \
0    110CSA              3GEN               8AGE               6RAC
1    177SCU              1GEN               1AGE               6RAC
2    177SCU              1GEN               8AGE               2RAC
3    110CSA              3GEN               5AGE               6RAC
4    166SSP              2GEN               8AGE               6RAC
```

```
5     110CSA               2GEN                 8AGE                 6RAC
6     177SCU               1GEN                 8AGE                 6RAC
7     177SCU               1GEN                 6AGE                 6RAC
8     110CSA               1GEN                 1AGE                 6RAC
9     167QUA               1GEN                 8AGE                 6RAC

   StratificationID4  SubMeasureID DisplayOrder
0               6EDU         BRF21           21
1               6EDU         BRF67           67
2               6EDU         BRF71           71
3               6EDU         BRF45           45
4               6EDU         BRF25           25
5               6EDU         BRF21           21
6               6EDU         BRF70           70
7               5EDU         BRF68           68
8               6EDU         BRF23           23
9               6EDU         BRF08            8

[10 rows x 33 columns]
```

[58]:
```
# Check the result
merged_data.columns
merged_data['LocationAbbr'].unique()
```

[58]:
```
array(['AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'DC', 'FL', 'GA',
       'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA',
       'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY',
       'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX',
       'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY'], dtype=object)
```

[59]:
```
merged_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37145 entries, 0 to 37144
Data columns (total 33 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   YEAR                        37145 non-null  object
 1   LocationDesc                37145 non-null  object
 2   Sales                       37145 non-null  float64
 3   Tax                         37145 non-null  float64
 4   LocationAbbr                37145 non-null  object
 5   TopicType                   37145 non-null  object
 6   TopicDesc                   37145 non-null  object
 7   MeasureDesc                 37145 non-null  object
 8   DataSource                  37145 non-null  object
 9   Response                    14403 non-null  object
```

```
 10  Data_Value_Unit             37145 non-null  object
 11  Data_Value_Type             37145 non-null  object
 12  Data_Value                  35515 non-null  float64
 13  Data_Value_Footnote_Symbol  1630 non-null   object
 14  Data_Value_Footnote         1630 non-null   object
 15  Data_Value_Std_Err          35515 non-null  float64
 16  Low_Confidence_Limit        35515 non-null  float64
 17  High_Confidence_Limit       35515 non-null  float64
 18  Sample_Size                 35515 non-null  float64
 19  Gender                      37145 non-null  object
 20  Race                        37145 non-null  object
 21  Age                         37145 non-null  object
 22  Education                   37145 non-null  object
 23  GeoLocation                 37145 non-null  object
 24  TopicTypeId                 37145 non-null  object
 25  TopicId                     37145 non-null  object
 26  MeasureId                   37145 non-null  object
 27  StratificationID1           37145 non-null  object
 28  StratificationID2           37145 non-null  object
 29  StratificationID3           37145 non-null  object
 30  StratificationID4           37145 non-null  object
 31  SubMeasureID                37145 non-null  object
 32  DisplayOrder                37145 non-null  int64
dtypes: float64(7), int64(1), object(25)
memory usage: 9.4+ MB
```

0.7.1 DATA CLEANING

[60]:

```python
# Count unique values in each column to identify candidates for categorical␣
↪conversion
unique_counts = pd.DataFrame.from_records([(col, merged_data[col].nunique())␣
↪for col in merged_data.columns],
                                          columns=['Column_Name',␣
↪'Num_Unique']).sort_values(by=['Num_Unique'])
unique_counts
# Display unique value counts for each column, to see what can be categorized␣
↪and what can be removed.
```

[60]:

	Column_Name	Num_Unique
10	Data_Value_Unit	1
24	TopicTypeId	1
5	TopicType	1
8	DataSource	1
14	Data_Value_Footnote	1
11	Data_Value_Type	1
13	Data_Value_Footnote_Symbol	1
27	StratificationID1	3

```
19                   Gender           3
30           StratificationID4        4
22                Education           4
6                 TopicDesc           4
25                  TopicId           4
20                     Race           6
9                  Response           6
29           StratificationID3        6
7               MeasureDesc           8
28           StratificationID2        8
21                      Age           8
0                      YEAR           9
26                MeasureId          11
32             DisplayOrder          38
31             SubMeasureID          38
23              GeoLocation          51
4              LocationAbbr          51
1              LocationDesc          51
3                       Tax          77
15       Data_Value_Std_Err         131
2                     Sales         336
16     Low_Confidence_Limit         985
12               Data_Value         994
17    High_Confidence_Limit         999
18              Sample_Size        7168
```

[61]:
```
# Droping columns with only one unique value, as they wont make any difference␣
↪or change in the output.
columns_to_drop = unique_counts[unique_counts['Num_Unique'] ==␣
↪1]['Column_Name'].tolist()
final_data = merged_data.drop(columns=columns_to_drop)
#dropping rows with Year mean given as a aggregate value
final_data = final_data[~final_data['YEAR'].str.contains('-', na=False)]
#Understanding how many people are smoking on an average from the given sample␣
↪over the years.

# Filter data for a specific MeasureDesc, e.g., "Current Smoking"
filtered_data = final_data[final_data['MeasureDesc'] == 'Current Smoking']
final_data['LocationAbbr'].unique()
```

[61]:
```
array(['AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'DC', 'FL', 'GA',
       'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA',
       'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY',
       'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX',
       'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY'], dtype=object)
```

[62]:
```
final_data['LocationAbbr'].unique()
```

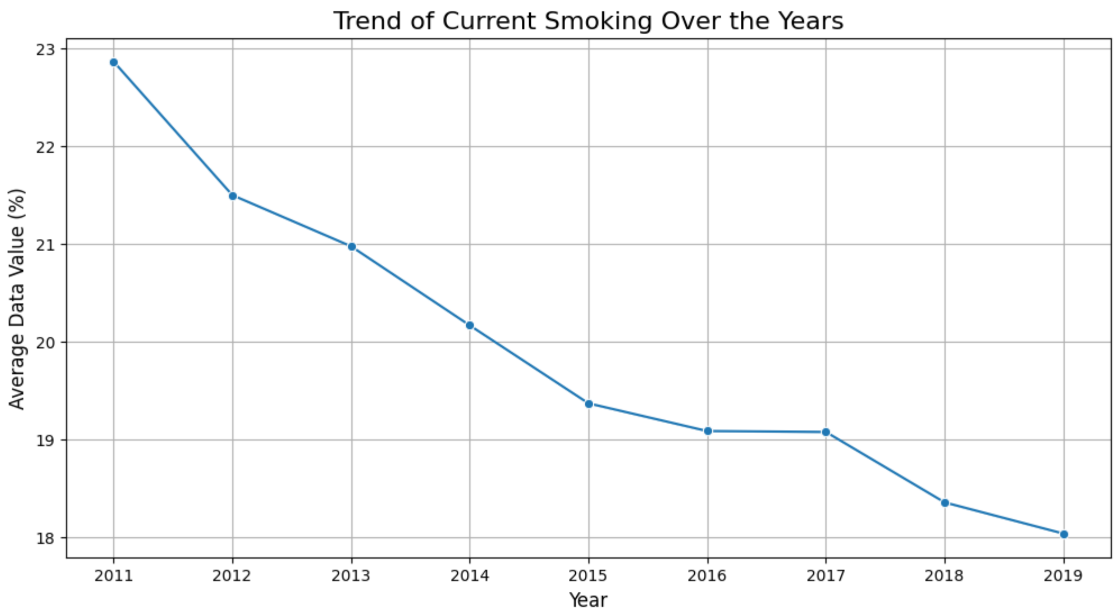
[62]: array(['AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'DC', 'FL', 'GA',
       'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA',
       'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY',
       'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX',
       'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY'], dtype=object)

[63]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Filter data for a specific MeasureDesc, e.g., "Current Smoking"
filtered_data = final_data[final_data['MeasureDesc'] == 'Current Smoking']

# Group by YEAR and calculate the mean Data_Value for trends
trend_data = filtered_data.groupby('YEAR')['Data_Value'].mean().reset_index()
trend_data = trend_data.merge(final_data[['YEAR',␣
↪'Tax','LocationAbbr','Sales']], on='YEAR', how='left')

trend_data.head(10)
```

[63]:

	YEAR	Data_Value	Tax	LocationAbbr	Sales
0	2011	22.863901	1.435	AL	68.4
1	2011	22.863901	1.435	AL	68.4
2	2011	22.863901	1.435	AL	68.4
3	2011	22.863901	1.435	AL	68.4
4	2011	22.863901	1.435	AL	68.4
5	2011	22.863901	1.435	AL	68.4
6	2011	22.863901	1.435	AL	68.4
7	2011	22.863901	1.435	AL	68.4
8	2011	22.863901	1.435	AL	68.4
9	2011	22.863901	1.435	AL	68.4

[64]:
```python
# Plot the trend over time
plt.figure(figsize=(12, 6))
sns.lineplot(data=trend_data, x='YEAR', y='Data_Value', marker='o')
plt.title('Trend of Current Smoking Over the Years', fontsize=16)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Average Data Value (%)', fontsize=12)
plt.grid(True)
plt.show()
```



The chart shows the trend of current smoking over the years, specifically from 2011 to 2019. Here are the key insights based on the graph:

Consistent Decline: There is a steady decrease in the average percentage of current smokers over the years, starting from about 24% in 2011 and dropping to approximately 17% in 2019. This indicates a declining trend in smoking prevalence.

Rate of Decline: The decline appears to be more significant between 2011 and 2015, where the percentage drops sharply. The reduction slows slightly in the subsequent years but continues downward.

Minor Fluctuations: Between 2016 and 2017, there is a small upward fluctuation where the percentage of smokers slightly increases. This might indicate a temporary reversal or stabilization in the decline.

As we see an overall decline from 2011 to 2019 we will see the fluctuations of the usage and consumption of different products over time and forecast its future.

[65]:
```
!pip install prophet
```

```
Requirement already satisfied: prophet in /opt/conda/lib/python3.11/site-
packages (1.1.6)
Requirement already satisfied: cmdstanpy>=1.0.4 in
/opt/conda/lib/python3.11/site-packages (from prophet) (1.2.5)
Requirement already satisfied: numpy>=1.15.4 in /opt/conda/lib/python3.11/site-
packages (from prophet) (1.26.3)
Requirement already satisfied: matplotlib>=2.0.0 in
/home/jovyan/.local/lib/python3.11/site-packages (from prophet) (3.8.2)
Requirement already satisfied: pandas>=1.0.4 in /opt/conda/lib/python3.11/site-
```

```
packages (from prophet) (2.1.4)
Requirement already satisfied: holidays<1,>=0.25 in
/opt/conda/lib/python3.11/site-packages (from prophet) (0.62)
Requirement already satisfied: tqdm>=4.36.1 in /opt/conda/lib/python3.11/site-
packages (from prophet) (4.66.1)
Requirement already satisfied: importlib-resources in
/opt/conda/lib/python3.11/site-packages (from prophet) (6.1.1)
Requirement already satisfied: stanio<2.0.0,>=0.4.0 in
/opt/conda/lib/python3.11/site-packages (from cmdstanpy>=1.0.4->prophet) (0.5.1)
Requirement already satisfied: python-dateutil in
/opt/conda/lib/python3.11/site-packages (from holidays<1,>=0.25->prophet)
(2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in
/home/jovyan/.local/lib/python3.11/site-packages (from
matplotlib>=2.0.0->prophet) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/home/jovyan/.local/lib/python3.11/site-packages (from
matplotlib>=2.0.0->prophet) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/home/jovyan/.local/lib/python3.11/site-packages (from
matplotlib>=2.0.0->prophet) (4.47.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/home/jovyan/.local/lib/python3.11/site-packages (from
matplotlib>=2.0.0->prophet) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.11/site-packages (from matplotlib>=2.0.0->prophet) (23.2)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.11/site-
packages (from matplotlib>=2.0.0->prophet) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/home/jovyan/.local/lib/python3.11/site-packages (from
matplotlib>=2.0.0->prophet) (3.1.1)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.0.4->prophet) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.0.4->prophet) (2023.4)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.11/site-
packages (from python-dateutil->holidays<1,>=0.25->prophet) (1.16.0)
```

[66]:

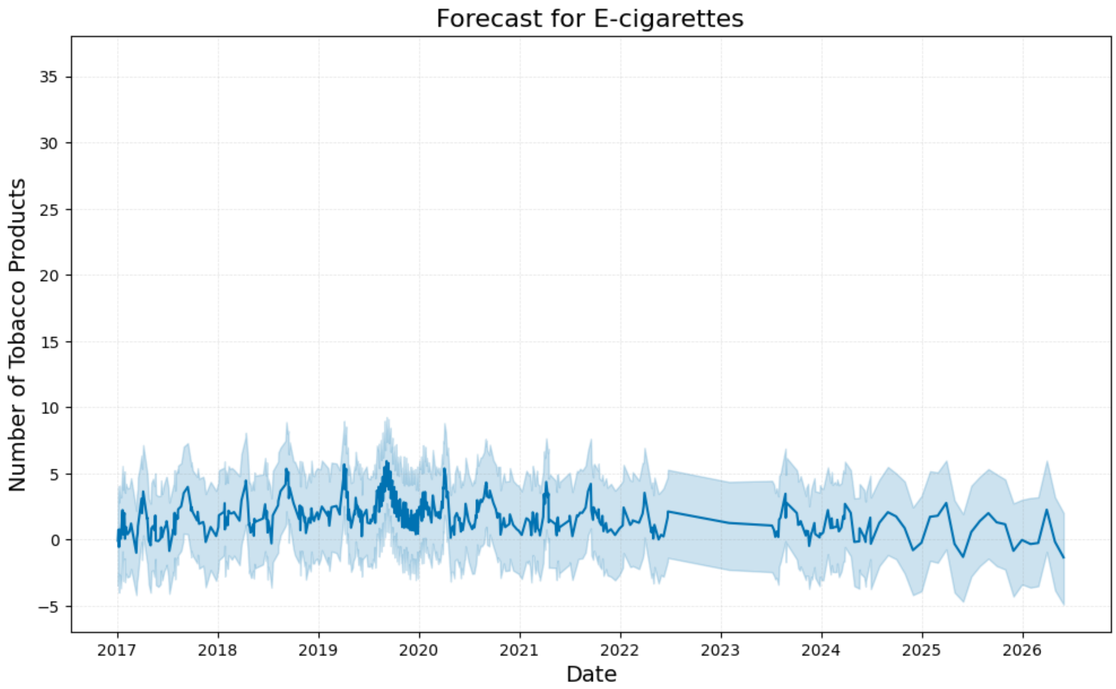
```python
import pandas as pd
import json
from prophet import Prophet
import matplotlib.pyplot as plt

# Replace 'file_path.json' with the path to your JSON file
file_path = 'tobacco-problem-0001-of-0001.json'

# Load the JSON file
```

```python
with open(file_path, 'r') as f:
    health_data = json.load(f)

# Extract the "results" part of the JSON
results = health_data.get('results', [])

# Convert the "results" array to a DataFrame
df = pd.DataFrame(results)

# Ensure 'tobacco_products' contains strings
df['tobacco_products'] = df['tobacco_products'].apply(lambda x: ', '.join(x) if isinstance(x, list) else str(x))

# Exclude specific categories
df = df[~df['tobacco_products'].str.contains("Heated Tobacco Product", na=False)]
df = df[~df['tobacco_products'].str.contains("Waterpipe", na=False)]

exclude_products = [
    'Cigar (large or premium)',
    'Chewing tobacco (loose leaf chew, plug, twist/roll)',
    'Dissolvable (for example, strips, sticks, orbs)',
]

df_filtered = df[~df['tobacco_products'].isin(exclude_products)].copy()

# Combine subcategories for e-cigarettes and categorize everything else as "Others"
e_cigarette_terms = ['e-cigarette', 'E-cigarette', 'vape', 'vaping', 'e-pipe', 'e-cigar', 'e-hookah']
df_filtered.loc[:, 'tobacco_products'] = df_filtered['tobacco_products'].apply(
    lambda x: 'E-cigarettes' if any(term in x.lower() for term in e_cigarette_terms)
    else 'Cigarettes' if 'cigarette' in x.lower()
    else 'Others'
)

# Group the data by 'tobacco_products' and 'date_submitted', then aggregate
df_grouped = df_filtered.groupby(['tobacco_products', 'date_submitted']).agg({
    'number_tobacco_products': 'sum'
}).reset_index()

# Rename columns for Prophet
df_grouped.rename(columns={'date_submitted': 'ds', 'number_tobacco_products': 'y'}, inplace=True)
```

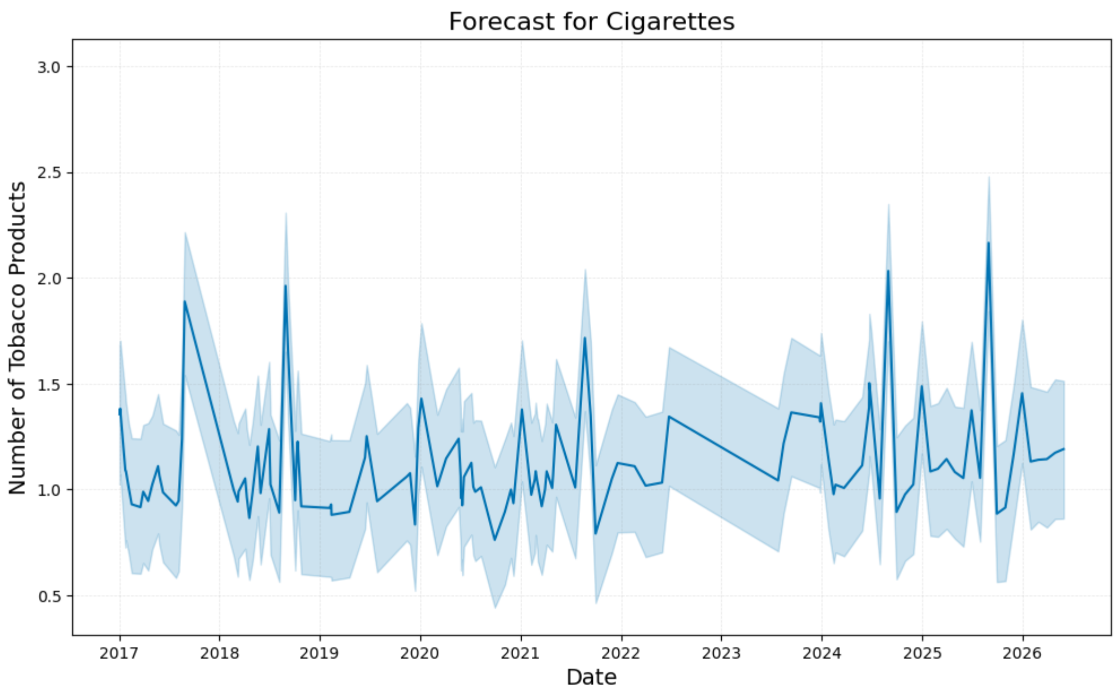
```
# Initialize a list to store forecasts for each tobacco product
forecast_results = {}

# Get unique tobacco products: Only "E-cigarettes", "Cigarettes", and "Others"
tobacco_products = ['E-cigarettes', 'Cigarettes', 'Others']

# Create a future dataframe for the next 2 years (adjustable)
future_dates = pd.date_range(start=df_grouped['ds'].min(), periods=24, freq='M')

# Forecast for each tobacco product
for product in tobacco_products:
    # Filter data for the current tobacco product
    product_data = df_grouped[df_grouped['tobacco_products'] == product]

    # Handle insufficient data by padding with zeros
    if len(product_data) < 2:
        print(f"Insufficient data for {product}, padding with zeros.")
        product_data = pd.DataFrame({
            'ds': future_dates,
            'y': [0] * len(future_dates)
        })

    # Fit Prophet model
    model = Prophet()
    model.fit(product_data)

    # Create a future dataframe for the next 2 years (adjustable)
    future = model.make_future_dataframe(periods=24, freq='M')

    # Forecast
    forecast = model.predict(future)
    forecast_results[product] = forecast  # Save the forecast for this product

    # Plot the forecast without black dots
    fig = model.plot(forecast)
    ax = fig.gca()

    # Remove black dots (actual data points) from the plot
    for line in ax.get_lines():
        if line.get_marker() == '.':  # Prophet uses '.' marker for actual data␣
↪points
            line.set_alpha(0)  # Make them invisible

    # Customize plot title and labels
    ax.set_title(f'Forecast for {product}', fontsize=16)
    ax.set_xlabel('Date', fontsize=14)
    ax.set_ylabel('Number of Tobacco Products', fontsize=14)
```

```
    ax.grid(True, which='major', linestyle='--', linewidth=0.5)

    plt.show()
```

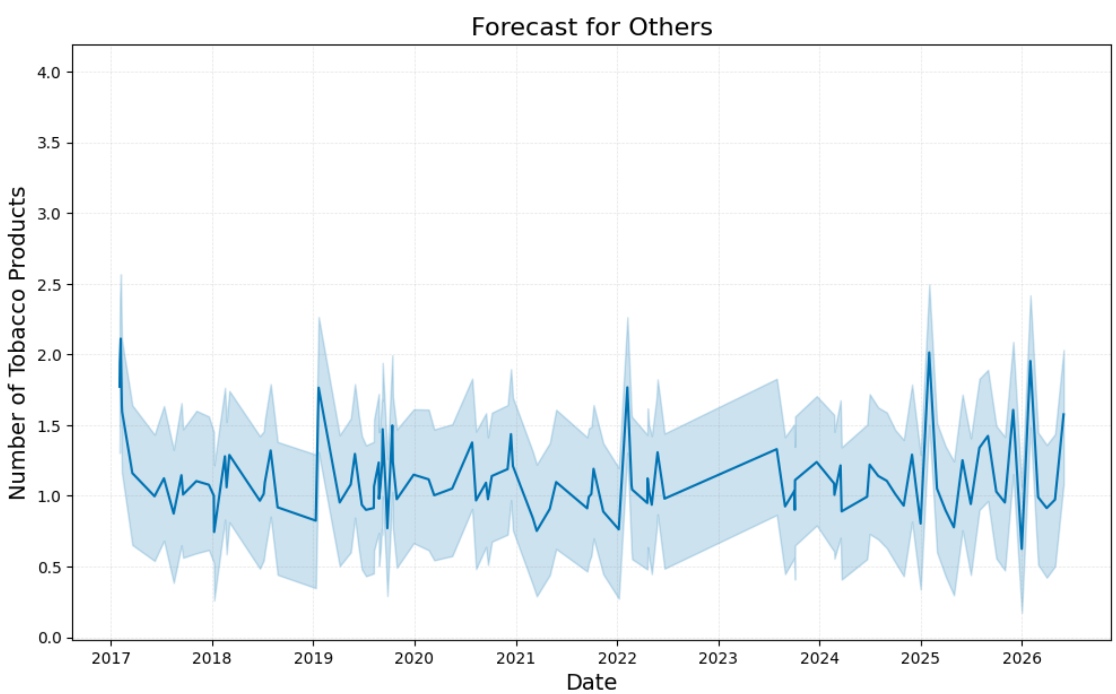
```
15:41:47 - cmdstanpy - INFO - Chain [1] start processing
15:41:47 - cmdstanpy - INFO - Chain [1] done processing
```




```
15:41:48 - cmdstanpy - INFO - Chain [1] start processing
15:41:48 - cmdstanpy - INFO - Chain [1] done processing
```




```
15:41:49 - cmdstanpy - INFO - Chain [1] start processing
15:41:49 - cmdstanpy - INFO - Chain [1] done processing
```



0.7.2 Observations:

1. E-Cigarettes

- **Trend**:
  - E-cigarettes have shown steady growth from 2017 to 2020, reaching their peak around 2019-2020.
  - After 2020, there is a slow but visible decline in demand, possibly due to increased market saturation or regulatory impacts.
- **Future Outlook**:
  - The forecast predicts a stable market with minor declines in demand through 2024-2026.
  - This suggests that the e-cigarette market is maturing, with limited room for explosive growth.

---

2. Cigarettes

- **Trend**:
  - The cigarette market experienced periodic spikes in demand, particularly around 2019 and 2020. These may be linked to seasonal demand or significant events, such as marketing campaigns.
  - Post-2020, demand stabilizes with an overall downward trend.
- **Future Outlook**:
  - The forecast predicts continued low demand for cigarettes, with minor fluctuations through 2026.
  - The downward trend is consistent with global health campaigns and shifting consumer preferences toward alternatives like e-cigarettes.

---

3. Others

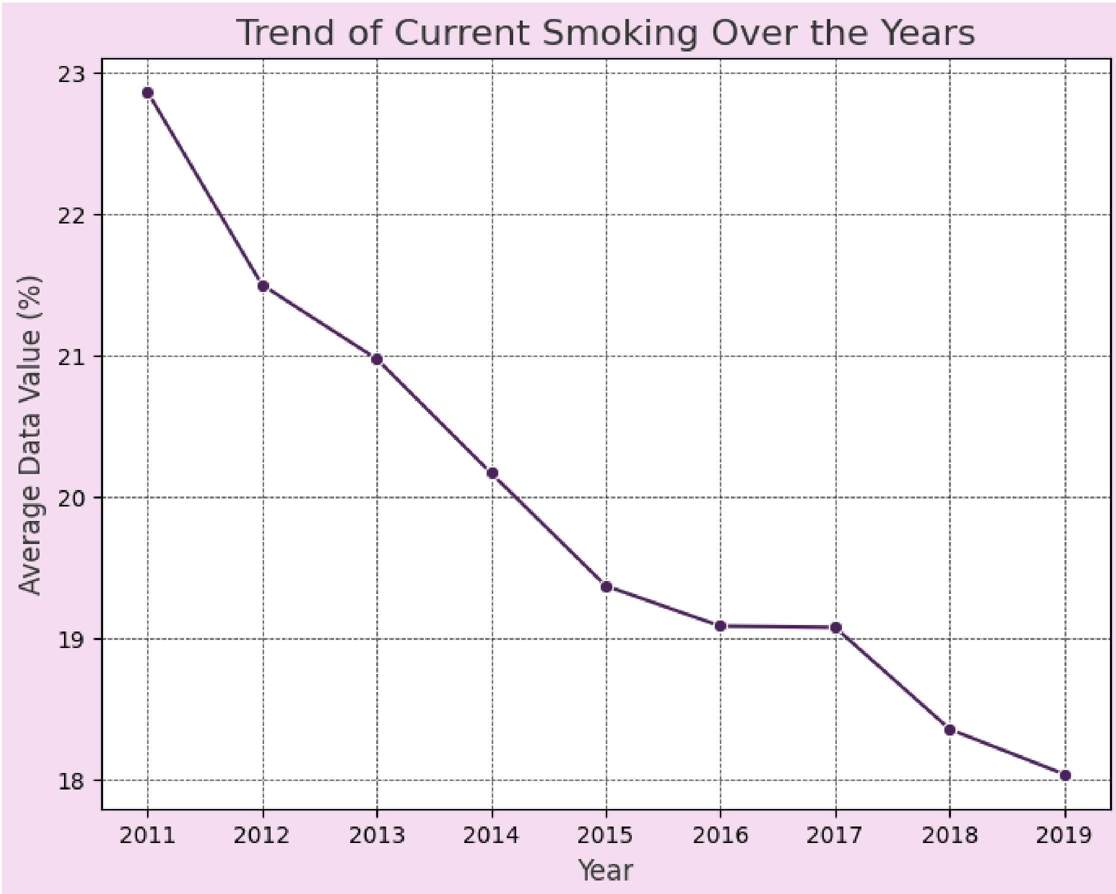
- **Trend**:
  - The "Others" category has exhibited significant volatility, with sharp peaks in 2020 and 2021, likely due to short-term consumer trends or external events.
  - Despite the volatility, the overall trend shows fluctuations stabilizing after 2022.
- **Future Outlook**:
  - The forecast predicts moderate stability for this category from 2024 to 2026, indicating reduced variability in product demand.
  - Products grouped under "Others" may include items with limited or niche markets.

0.7.3 Insights:

The forecast above highlights consumption trends for specific tobacco products. Notably, while e-cigarettes, often considered less harmful than traditional cigarettes and other tobacco products, show growth, the forecast indicates that the more harmful products, such as cigarettes, are anticipated to be consumed at significantly higher rates. This underscores the urgent need for stronger government interventions, such as increased taxation and campaigns encouraging cessation. It is equally important to ensure that these initiatives foster positive reinforcement, aiding individuals in reducing or quitting consumption. The next analysis will evaluate the effectiveness of existing government policies in achieving this goal thereby analysing the tax and Sales rates for Cigarettes as we have anticipated higher consumption of this category.

Forecasting Future Trends in Tobacco Consumption: Evaluating the Continuity of Government Initiatives from 2011 to 2019

[67]:
```python
# Import necessary libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Define a custom color palette for contrast
line_colors = ['#4A235A', '#154734', '#1A237E']  # Dark purple, dark green, and␣
↪dark blue
text_color = '#333333'  # Dark gray for titles and labels

# Plot 1: Trend of Current Smoking
plt.figure(figsize=(8, 6))
sns.lineplot(data=trend_data, x='YEAR', y='Data_Value', marker='o',␣
↪color=line_colors[0])
plt.title('Trend of Current Smoking Over the Years', fontsize=16,␣
↪color=text_color)
plt.xlabel('Year', fontsize=12, color=text_color)
plt.ylabel('Average Data Value (%)', fontsize=12, color=text_color)
plt.grid(True, linestyle='--', linewidth=0.5, color=text_color)
plt.gcf().set_facecolor('#F4DDF0')  # Set the background color for the figure
plt.show()
```



0.7.4 Observation

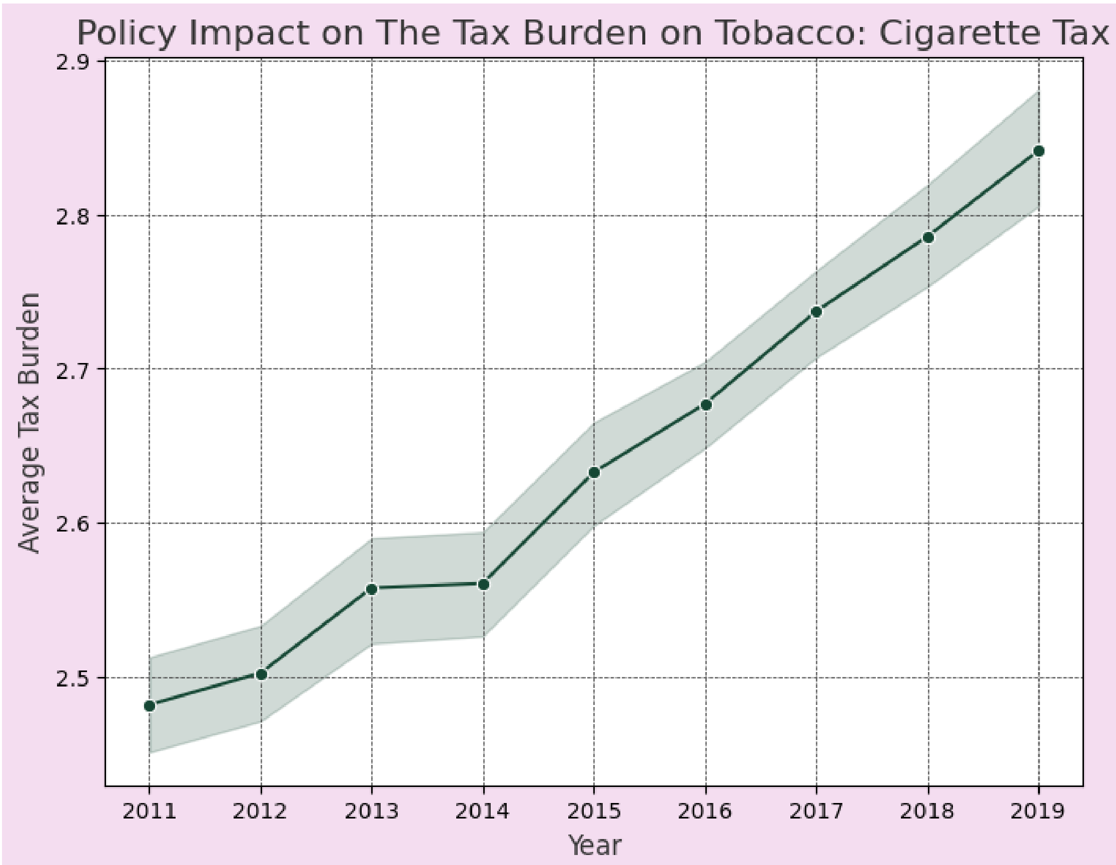
0.7.5 Trend of Current Smoking Over the Years (Top Graph)

- There is a **clear declining trend** in the percentage of current smokers from **2011 to 2019**.
- The average data value dropped from **23% in 2011** to **18% in 2019**.
- This downward trend suggests that **government reforms** and awareness programs may have played a role in reducing smoking rates.

---

[68]:
```
# Plot 2: Policy Impact on The Tax Burden on Tobacco
plt.figure(figsize=(8, 6))
sns.lineplot(data=trend_data, x='YEAR', y='Tax', marker='o',␣
 ↪color=line_colors[1])
plt.title('Policy Impact on The Tax Burden on Tobacco: Cigarette Tax',␣
 ↪fontsize=16, color=text_color)
plt.xlabel('Year', fontsize=12, color=text_color)
```

```
plt.ylabel('Average Tax Burden', fontsize=12, color=text_color)
plt.grid(True, linestyle='--', linewidth=0.5, color=text_color)
plt.gcf().set_facecolor('#F4DDF0')  # Set the background color for the figure
plt.show()
```

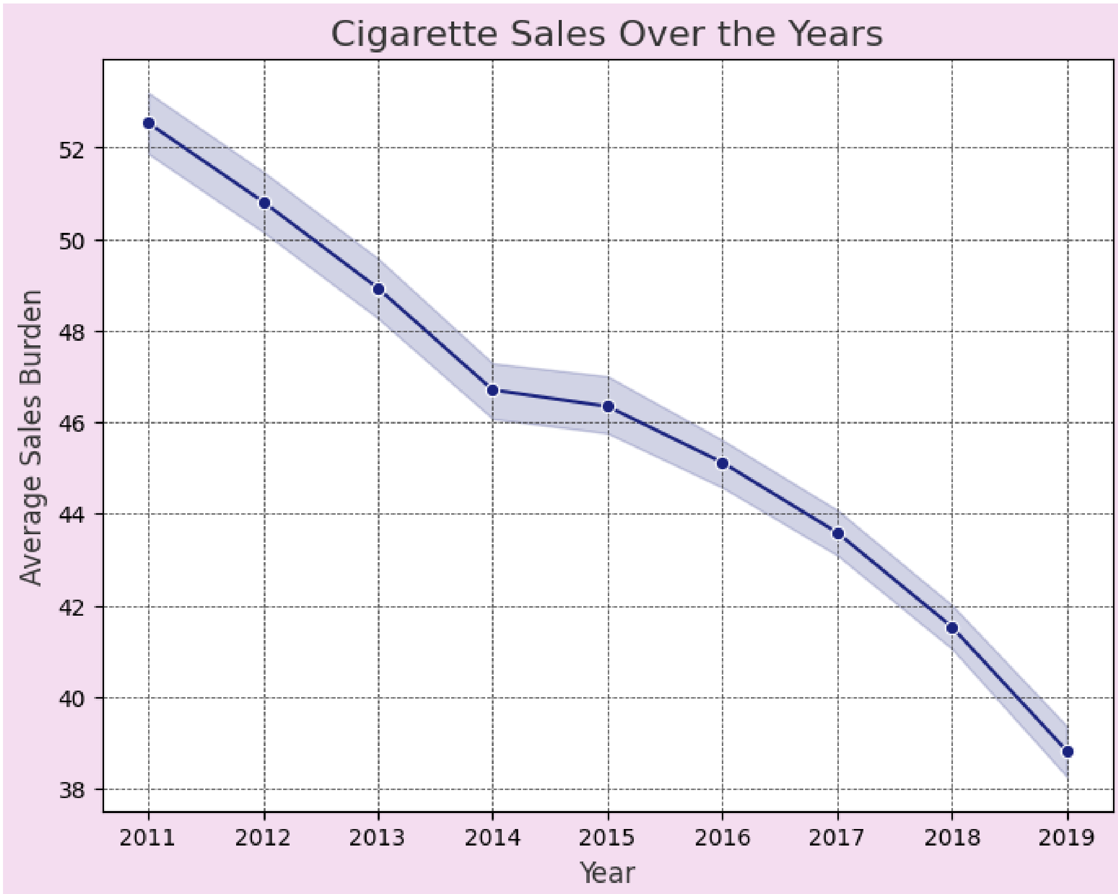


0.8 Observation

plt.subplot(3, 1, 2) sns.lineplot(data=trend_data, x='YEAR', y='Tax', marker='o', color=line_colors[1]) plt.title('Policy Impact on The Tax Burden on Tobacco: Cigarette Tax', fontsize=16, color=text_color) plt.xlabel('Year', fontsize=12, color=text_color) plt.ylabel('Average Tax Burden', fontsize=12, color=text_color) plt.grid(True, linestyle='–', linewidth=0.5, color=text_color)

[69]:
```
# Plot 3: Cigarette Sales
plt.figure(figsize=(8, 6))
sns.lineplot(data=trend_data, x='YEAR', y='Sales', marker='o',␣
 ↪color=line_colors[2])
plt.title('Cigarette Sales Over the Years', fontsize=16, color=text_color)
plt.xlabel('Year', fontsize=12, color=text_color)
plt.ylabel('Average Sales Burden', fontsize=12, color=text_color)
```

```
plt.grid(True, linestyle='--', linewidth=0.5, color=text_color)
plt.gcf().set_facecolor('#F4DDF0')  # Set the background color for the figure
plt.show()
```



0.8.1 3. Observation

- **Cigarette sales** have shown a significant decline from **2011 to 2019**, dropping from **52% to approximately 38%**.
- This decline aligns with the upward trend in tax burden, suggesting a **negative correlation** between cigarette taxes and cigarette sales.
- The shaded area highlights consistent downward movement with minimal variability over the years.

[70]:
```
correlation_tax_sales = trend_data['Tax'].corr(trend_data['Sales'])
correlation_tax_smoking = trend_data['Tax'].corr(trend_data['Data_Value'])
```

```
print(f"Correlation between Tax Burden and Cigarette Sales:␣
↪{correlation_tax_sales:.2f}")
```

```
Correlation between Tax Burden and Cigarette Sales: -0.64
```

0.8.2 Interpretation

There is a noticeable negative correlation between the tax burden on tobacco and cigarette sales. As the tax burden increases, cigarette sales show a significant decline, highlighting a strong inverse relationship between these variables.

0.8.3 Insights

This finding suggests that higher tobacco taxes are an effective policy tool for reducing cigarette consumption. The negative trend underscores the impact of price sensitivity on consumer behavior, indicating that increased costs discourage cigarette purchases.

0.8.4 Conclusion

The story of tobacco use in the modern era is one of evolving challenges and hard-won progress. As traditional smoking declines, the rise of e-cigarettes has introduced new complexities, reshaping the landscape of public health. Between **2018 and 2023**, health reports linked to e-cigarette use surged, with **seizures** making up **12.68% of all cases** (319 incidents). A dramatic spike in **2019** signaled the growing severity of this issue, placing vaping-related health concerns firmly in the spotlight. Alongside seizures, **respiratory problems** such as **shortness of breath** ranked as the second most reported health issue, raising alarms about the potential risks of this new technology. Alarmingly, defects like **"taste issues"** and **"foreign materials"** were found to significantly exacerbate these respiratory problems, underscoring the urgent need for stricter regulations.

Amid this vaping crisis, traditional smoking continues to tell a story of both progress and disparity. From **2011 to 2019**, smoking rates declined across all racial groups. **Whites** saw their smoking rates drop from **23% to 15%**, **African Americans** from **20% to 14%**, and **Asians** from **10% to 6%**. However, for **American Indian/Alaska Natives**, the decline was slower, from **30% to 25%**, highlighting persistent inequalities. Gender trends provided a more optimistic narrative, as women made greater strides in avoiding smoking. The proportion of **"Never Smokers"** among women increased from **60% to 72%**, compared to **55% to 68%** among men, reflecting the success of targeted campaigns and awareness programs.

As the analysis shifted to geographical patterns, a stark regional divide emerged. **California** and **New York** stood out as champions in reducing smoking rates. By **2019**, over **75%** of residents in these states identified as **"Never Smokers"**, a testament to their robust anti-smoking policies, including taxation, advertising bans, and community-driven cessation programs. In contrast, states like **Alabama** and **Nevada** lagged behind, with **"Never Smokers"** remaining below **60%**. The struggles of these states underscored the critical role of systemic barriers and weaker policy frameworks in hindering progress.

Throughout this story, the transformative power of policy and education became evident. States with comprehensive anti-smoking measures, such as **California**, demonstrated **Former/Current**

Smokers ratios above 1.8, compared to less than 1.2 in less proactive states like **Texas**. Education, too, proved a powerful weapon against tobacco use. States with higher literacy rates consistently reported lower smoking prevalence, emphasizing the importance of awareness and knowledge in driving behavioral change.

These findings illuminate the path forward. High-risk groups, such as **American Indian/Alaska Natives**, and states with lower cessation rates require **tailored interventions** to address their unique challenges. Regulatory actions to address product defects in e-cigarettes are essential to mitigate their rising health risks. Expanding **educational campaigns** in states with lower literacy levels and higher smoking rates could further amplify cessation success and prevention efforts.

Yet, the journey is far from over. Deeper research is needed to uncover the **regional barriers** that hinder smoking cessation. The **long-term health impacts of vaping** must be thoroughly investigated to refine regulations and policies. Leveraging **technology**, such as AI-driven insights and digital health apps, could revolutionize smoking prevention and cessation programs, making them more accessible and effective.

0.8.5 Future Scope

We aim to analyze additional government reforms, such as **taxation policies, smoking bans in public areas, awareness campaigns, and tobacco cessation programs**, to evaluate behavioral patterns of tobacco consumers.

Specifically, we plan to:
1. **Assess the Impact of Government Initiatives**:
- Identify which reforms led to **positive reinforcement** (e.g., reduced tobacco usage) and which had **negative psychological effects** (e.g., resistance or increased usage in certain demographics).

2. **Investigate Driving Factors of Tobacco Usage**:
   - Explore **lifestyle patterns**, **social pressures**, and **psychological triggers** influencing individuals to use tobacco.

   - Correlate these factors to understand their relationship with tobacco usage trends.
3. **Behavioral and Social Correlation**:
   - Analyze how **social environments**, such as peer influence, family history, and stress levels, contribute to tobacco usage.

   - Evaluate the psychological and emotional aspects of consumer behavior to uncover deeper insights.

By addressing these areas, we aim to provide comprehensive recommendations to policymakers for **targeted interventions** and develop strategies to reduce tobacco consumption effectively.