Cardiovascular Dashboard

Environment Setup

In [1]:

```python
import pandas as pd
import numpy as np
import panel as pn
pn.extension('tabulator')

import hvplot.pandas
import holoviews as hv
hv.extension('bokeh')

from holoviews.selection import link_selections
from holoviews import opts
```

Data

Explore and clean data

In [2]:

```python
df = pd.read_csv('https://chronicdata.cdc.gov/api/views/kztq-p2jf/rows.csv?accessType=DOWNLOAD&api_foundry=true')

pd.set_option('display.max_rows', 20)
pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', 100)

df = df.dropna(subset = ['Data_Value'])
df
```

Out[2]:

	row_id	Year	LocationAbbr	LocationDesc	DataSource	PriorityArea1	PriorityArea2	PriorityArea3	PriorityArea4	Category	Topic	Ir
0	NVSS~2003~32~NV005~AGE08~Crude	2003	NV	Nevada	NVSS	None	None	None	None	Cardiovascular Diseases	Heart Failure	Rate r an adul
1	NVSS~2007~59~NV001~AGE08~Crude	2007	US	United States	NVSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease	Rate cardio r an adul
2	NVSS~2012~59~NV001~RAC07~Crude	2012	US	United States	NVSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease	Rate cardio r an adul
3	NVSS~2011~59~NV001~GEN02~Crude	2011	US	United States	NVSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease	Rate cardio r an adul
4	NVSS~2010~59~NV001~AGE05~Crude	2010	US	United States	NVSS	None	None	None	None	Cardiovascular Diseases	Major Cardiovascular Disease	Rate cardio r an adul
...	...	...	...	...	...	...	...	...	...	...	...	
158075	NVSS~2012~15~NV008~GEN01~Crude	2012	HI	Hawaii	NVSS	None	None	None	None	Cardiovascular Diseases	Stroke	hem r an adul
158076	NVSS~2015~06~NV008~AGE04~Crude	2015	CA	California	NVSS	None	None	None	None	Cardiovascular Diseases	Stroke	hem r an adul
158077	NVSS~2013~06~NV008~RAC01~Age-Standardized	2013	CA	California	NVSS	None	None	None	None	Cardiovascular Diseases	Stroke	hem r an adul
158078	NVSS~2012~19~NV008~GEN01~Crude	2012	IA	Iowa	NVSS	None	None	None	None	Cardiovascular Diseases	Stroke	hem r an adul
158079	NVSS~2013~04~NV008~OVR01~Age-Standardized	2013	AZ	Arizona	NVSS	None	None	None	None	Cardiovascular Diseases	Stroke	hem r an adul

129093 rows × 30 columns

In [3]:
```
df['Break_Out_Category'].unique()
```

Out[3]:
```
array(['Age', 'Race', 'Gender', 'Overall'], dtype=object)
```

In [4]:
```
race = df[(df['Break_Out_Category'] == 'Race')]
race['Break_Out'].unique()
```

Out[4]:
```
array(['Other', 'Non-Hispanic Black', 'Hispanic', 'Non-Hispanic White'],
      dtype=object)
```

In [5]:
```
# Replace original variables with more accurate terms for easier readability

disease = df.replace(['Diseases of the Heart (Heart Disease)','Acute Myocardial Infarction (Heart Attack)'], ['Diseases of the Heart','Acute Myocardial Infarc
race = df.replace(['Non-Hispanic Black','Non-Hispanic White'], ['Black','White'], inplace=True)
```

```
df['Topic'].unique()
```

Out[5]:

```
array(['Heart Failure', 'Major Cardiovascular Disease', 'Stroke',
       'Diseases of the Heart', 'Coronary Heart Disease',
       'Acute Myocardial Infarction'], dtype=object)
```

In [6]:

```
df['Break_Out'].unique()
```

Out[6]:

```
array(['75+', 'Other', 'Female', '45-64', '25-44', '65+', '35+', '18-24',
       'Male', 'Black', 'Hispanic', 'Overall', 'White'], dtype=object)
```

In [ ]:

```
age = df[(df['Break_Out_Category'] == 'Age')]
age = list(age['Break_Out'].unique())
```

In [7]:

```
idf = df.interactive()
```

Widgets

Create widgets to allow interactivity for the dashboard

In [8]:

```
# "Select State" widget - removed US and DC to ensure only states were used
menu_items =list(df['LocationDesc'].unique())
menu_items.remove('United States')
menu_items.remove('Washington, DC')
menu_items.sort()
state_select = pn.widgets.Select(name='State', options=menu_items)

# "Select Year" widget
Year=list(df['Year'].unique())
Year.sort()
year_select = pn.widgets.Select(name='Year', options=Year)

# "Select Disease" widget
disease_states = df['Topic'].unique()
disease_items = list(disease_states)
disease_select = pn.widgets.Select(name='Disease', options=disease_items)
```

Out[8]:

Creating separate visual elements

Disease State Timeline

In [11]:

```
disease_pipeline = (
    idf[
        (idf.Year <= 2018) &
        (idf.Topic.isin(disease_states)) &
        (idf.LocationDesc == 'United States') &
        (idf.Break_Out_Category == 'Overall') &
        (idf.Data_Value_TypeID == 'AgeStdz')
    ]
    .groupby(['Topic','Year'])['Data_Value'].sum()
    .to_frame()
    .reset_index()
    .sort_values(by='Year')
    .reset_index(drop=True)
)
```

In [12]:

```
disease_pipeline
```

Out[12]:

In [13]:

```
disease_state_timeline = disease_pipeline.hvplot(frame_width=800,
                                                 frame_height=175,
                                                 x='Year',
                                                 y='Data_Value',
                                                 by='Topic',
                                                 ylabel='Incidence Rate Per 100,000',
                                                 line_width=1.5,
                                                 color=['pink','red','crimson','indianred','firebrick','darkred'],
                                                 title="CV Disease in USA (2000-2018)")
disease_state_timeline
```

Out[13]:

Disease Rate by State Bar Chart

In [15]:

```
disease_states_bar_pipeline = (
    idf[
        (idf.LocationDesc == state_select) &
        (idf.Topic.isin(disease_states)) &
        (idf.Break_Out_Category == 'Overall') &
        (idf.Data_Value_TypeID == 'AgeStdz')
    ]
    .groupby(['LocationDesc','Topic'])['Data_Value'].mean()
    .to_frame()
    .reset_index()
    .sort_values(by='LocationDesc')
    .reset_index(drop=True)
)
```

In [16]:

```
disease_states_bar_pipeline
```

Out[16]:

In [17]:

```
disease_bar_chart = disease_states_bar_pipeline.hvplot(kind='bar',
                                                       frame_width=800,
                                                       frame_height=175,
                                                       x='LocationDesc',
                                                       y='Data_Value',
                                                       by='Topic',
                                                       xlabel='',
                                                       ylabel='Incidence Rate Per 100,000',
                                                       title="Average Rates By State (2000-2018)")
disease_bar_chart = disease_bar_chart.opts(cmap='reds', shared_axes=False)
disease_bar_chart
```

Out[17]:

Gender Bar Chart

In [18]:

```
gender = df[(df['Break_Out_Category'] == 'Gender')]
gender = list(gender['Break_Out'].unique())
gender
```

Out[18]:

```
['Female', 'Male']
```

In [19]:

```
gender_pipeline = (
    idf[
        (idf.Year == year_select) &
        (idf.LocationDesc == state_select) &
        (idf.Topic == disease_select) &
        (idf.Break_Out.isin(gender)) &
        (idf.Data_Value_TypeID == 'AgeStdz')
    ]
    .groupby(['Year','LocationDesc','Topic','Break_Out'])['Data_Value'].mean()
    .to_frame()
    .reset_index()
    .sort_values(by='LocationDesc')
    .reset_index(drop=True)
)
```

In [20]:

```
gender_pipeline
```

Out[20]:

In [21]:

```
gender_bar_chart = gender_pipeline.hvplot.barh(
                                          frame_width=300,
                                          frame_height=175,
                                          x='Break_Out',
                                          y='Data_Value',
                                          color='Data_Value',
                                          cmap=['maroon','lavenderblush'],
                                          alpha=0.85,
                                          xlabel='Gender',
                                          ylabel='Incidence Rate Per 100,000',
                                          title="Average Rates By Gender")
gender_bar_chart
```

Out[21]:

Age Bar Chart

In [22]:

```
age_pipeline = (
    idf[
        (idf.Year == year_select) &
        (idf.LocationDesc == state_select) &
        (idf.Topic == disease_select) &
        (idf.Break_Out.isin(age))
    ]
    .groupby(['Year','LocationDesc','Topic','Break_Out'])['Data_Value'].mean()
    .to_frame()
    .reset_index()
    .sort_values(by=['Year','LocationDesc','Break_Out'])
    .reset_index(drop=True)
)
```

In [23]:

```
age_pipeline
```

Out[23]:

In [24]:

```
age_bar_chart = age_pipeline.hvplot.barh(
                                          frame_width=300,
                                          frame_height=175,
                                          x='Break_Out',
                                          y='Data_Value',
                                          color='Data_Value',
                                          cmap='reds',
                                          alpha=0.75,
                                          xlabel='Age',
                                          ylabel='Incidence Rate Per 100,000',
                                          title="Average Rates By Age")
age_bar_chart
```

Out[24]:

Race Bar Chart

In [25]:

```
race = ['Other', 'Black', 'Hispanic', 'White']

race_pipeline = (
    idf[
        (idf.Year == year_select) &
        (idf.LocationDesc == state_select) &
        (idf.Topic == disease_select) &
        (idf.Break_Out.isin(race)) &
        (idf.Data_Value_TypeID == 'AgeStdz')
    ]
    .groupby(['Year','LocationDesc','Topic','Break_Out'])['Data_Value'].mean()
    .to_frame()
    .reset_index()
    .sort_values(by=['Year','LocationDesc','Break_Out'])
    .reset_index(drop=True)
)
```

In [26]:

```
race_pipeline
```

Out[26]:

In [27]:

```
race_bar_chart = race_pipeline.hvplot.barh(
                                      frame_width=300,
                                      frame_height=175,
                                      x='Break_Out',
                                      y='Data_Value',
                                      color='Data_Value',
                                      cmap='reds',
                                      xlabel='Race',
                                      ylabel='Incidence Rate Per 100,000',
                                      title="Average Rates By Race")
race_bar_chart
```

Out[27]:

Dashboard Creation

Linking all separate elements together to create the final interactive dashboard

In [28]:

```
link = link_selections.instance()

linked = link(gender_bar_chart + age_bar_chart + race_bar_chart).opts(shared_axes=False)
linked
```

Out[28]:

In [29]:

```
widgets = pn.WidgetBox(year_select, state_select, disease_select, margin=5)

template = pn.template.FastListTemplate(
    title='Cardiovascular Disease in USA',
    sidebar=[pn.pane.Markdown('## Visualization of incidence rates of cardiovascular diseases that occurred in the United States from 2000-2018. Data retrieve
             pn.pane.Markdown('## Settings'),
             widgets,
             pn.pane.Markdown('###### Dashboard created by Thanh-Trang Tran')],
    main=[pn.Row(pn.Column(disease_state_timeline.panel(width=1000))),
          pn.Row(pn.Column(disease_bar_chart.panel(width=1000))),
          pn.Row(pn.Column(linked.panel(width=370)))],
    header_background='#8B0A50',
)

#template.show()
template.servable();
```

```
Launching server at http://localhost:50470
```

In [ ]: