**Readme for group 3**

**An overview of the project and its purpose:**

This project examines US Healthcare Costs, and creating visualizations in the trends of Medical Inflation. The dataset focuses on the analysis of medicare and medicaid across the US between the years of 1980-2020, encompassing the cost of health care goods and services by state. The estimates in this dataset were used to help determine the role that health care costs have on a state’s population including size, aging, changes in disease incidence, available services, their usage, and pricing.

**Dataset:**

This project uses National Health Expenditure data from the website for Medicare and Medicaid, last updated in September 2023, and last accessed in February 2024:

<https://www.cms.gov/data-research/statistics-trends-and-reports/national-health-expenditure-data/state-provider>

**Dataset Interactions**

These instructions allow for the creation of the dataset for use to create a health spending choropleth map, a plot of healthcare spending by category over time, top states per each category over time, and bokeh descriptive tables.

**\* Download the dataset at the following link:**

<https://www.cms.gov/data-research/statistics-trends-and-reports/national-health-expenditure-data/state-provider>

*\* Dependencies:*

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

import pathlib as path

\**Import CSV*

aggregate **=** pd**.**read\_csv('Resources/PROV\_US\_AGGREGATE20.CSV')

med\_data **=** pd**.**DataFrame(aggregate)

med\_data**.**head()

*\*Rename columns: Region Number, region name, state, average annual growth*

med\_data.rename(columns = {'Region\_Number': 'Region(#)', 'REGION\_NAME': 'Region', 'State\_Name': 'State'}, inplace = True)

med\_data.head()

*\*Remove ‘Y’ from the year columns*

*med\_data.rename(columns = {'Y1980': '1980', 'Y1981': '1981',*

*}, inplace = True)*

*med\_data.head()*

*\*Rename columns*

*med\_data.rename(columns = {'Average\_Annual\_Percent\_Growth': 'Avg An Growth(%)'}, inplace = True)*

*med\_data*

*\*Drop columns not needed*

list = ['Code', 'Region(#)', 'Region']

med\_data = med\_data.drop(list, axis = "columns")

med\_data.head()

\**Filter 'Group' column to just show state*

med\_data.dropna(subset=['State'], inplace = True)

med\_data

*\*Export to csv*

med\_data.to\_csv("Resources/med\_data.csv", index=False)

*\*Import dependencies and create an instance of MongoClient*

from pymongo import MongoClient

from pprint import pprint

*\*Create an instance of MongoClient*

mongo = MongoClient(port=#####)

*\*Assign the database to a variable name*

db = mongo['project\_3']

*\*Review the collections in the new database*

print(db.list\_collection\_names())

*\*Assign the collection to a variable*

med\_data = db['med\_data']

*\*Print an entry to verify*

pprint(db.med\_data.find\_one())

*\*Print all entries*

query = {'Group': 'State'}

med\_data\_all = med\_data.find(query)

med\_data\_all\_df = pd.DataFrame(med\_data\_all)

med\_data\_all\_df.head()

*\*Print column names*

print("Column Names:")

print(med\_data\_all\_df.columns)

*\** *New order of columns*

*new\_order = ['\_id', 'State', 'Group', 'Item', '1980', '1981', '1982', '1983',*

*\*Create a new DataFrame with columns in the desired order*

med\_data\_all\_df\_reorder = med\_data\_all\_df[new\_order]

med\_data\_all\_df\_reorder.head()

*\*Drop columns we don't need*

list = ['\_id', 'Group']

med\_data\_all\_df\_reorder = med\_data\_all\_df\_reorder.drop(list, axis ="columns")

med\_data\_all\_df\_reorder.head()

*\*Reshape the DataFrame into long format*

med\_data\_long\_df = pd.melt(med\_data\_all\_df\_reorder, id\_vars=['State','Item'], var\_name='Year', value\_name='Spending')

med\_data\_long\_df.head()

*\*Unique items*

*unique\_items = med\_data\_long\_df["Item"].unique()*

for item in unique\_items:

print(item)

*\*Drop all the years and just show average growth rate*

list = ['1980', '1981', '1982', '1983', '1984', '1985',

'\_id', 'Group']

med\_data\_all\_df\_2 = med\_data\_all\_df.drop(list, axis = "columns")

med\_data\_all\_df\_2.head()

*\*New order of columns*

new\_order = ['State', 'Item', 'Avg An Growth(%)']

*\*Create a new DataFrame with columns in the desired order*

med\_data\_all\_df\_2 = med\_data\_all\_df\_2[new\_order]

med\_data\_all\_df\_2.head()

######################################################

***Choropleth map***

*\*Dependencies*

import plotly.express as px

import pandas as pd

import json

*\*Load GeoJSON for US states*

with open('Resources/us-states.json') as f:

geojson\_data = json.load(f)

*\*Define df*

df = med\_data\_all\_df\_2

*\*Make list of items*

items = df['Item'].unique().tolist()

*\*Create dropdown menu buttons*

item\_buttons = []

for item in items:

args = [

{

'z': [df[df['Item'] == item]['Avg An Growth(%)'].astype(float).tolist()],

'locations': [df[df['Item'] == item]['State'].tolist()],

'text': [df[df['Item'] == item]['Item'].tolist()]

},

]

button = {

'args': args,

'label': item,

'method': 'update'

}

item\_buttons.append(button)

*\*Create initial choropleth map for the first item*

selected\_item = items[0]

filtered\_data = df[df['Item'] == selected\_item]

result\_df = filtered\_data[['State', 'Avg An Growth(%)']]

fig = px.choropleth(

data\_frame = result\_df,

geojson = geojson\_data,

locations = 'State',

featureidkey = 'properties.name',

color = 'Avg An Growth(%)',

color\_continuous\_scale = 'Greens',

scope = 'usa',

labels = {'Avg An Growth(%)': 'Average Annual Growth (%)'},

)

*\*Add title*

fig.update\_layout(title\_text = 'Choropleth Map for Healthcare Spending by State')

*\*Create dropdown menu*

dropdown\_menu = [

{

'buttons': item\_buttons,

'direction': 'down',

'showactive': True,

'x': 0.1,

'xanchor': 'left',

'y': 1.04,

'yanchor': 'top'

}

]

*\*Update layout with dropdown menu*

fig.update\_layout(

updatemenus = dropdown\_menu

)

*\*Save the figure as an HTML file*

fig.write\_html('health\_spending\_choropleth.html')

##################################################

**matplotlib.pyplot as plt – repeat for California, Texas, Florida, New York, Pennsylvania, South Dakota, North Dakota, Alaska, Wyoming, Vermont**

*\*Filter for California (Population Rank = 1)*

california\_data = med\_data\_long\_df[med\_data\_long\_df['State'] == 'California']

*\*Plot figure size*

plt.figure(figsize=(12, 6))

*\*Unique categories*

categories = california\_data['Item'].unique()

*\*Loop through categories*

for category in categories:

category\_data = california\_data[california\_data['Item'] == category]

category\_data = category\_data.sort\_values('Year')

plt.plot(category\_data['Year'], category\_data['Spending'], label=category, marker='o')

*\*Chart title and labels*

plt.title('Healthcare Spending by Category Over Time in California', fontsize=16)

plt.xlabel('Year', fontsize=14)

plt.ylabel('Spending (in Millions)', fontsize=14)

*\*Show every other year on x-axis*

plt.xticks(category\_data['Year'].unique()[::2])

*\*Adding a legend*

plt.legend()

*\*Show the plot*

plt.show()

#####################################################

***\*The top 10 states per each category:***

*\*Options:*

# Personal Health Care (Millions of Dollars)

# Hospital Care (Millions of Dollars)

# Physician & Clinical Services (Millions of Dollars)

# Other Professional Services (Millions of Dollars)

# Dental Services (Millions of Dollars)

# Other Health, Residential, and Personal Care (Millions of Dollars)

# Home Health Care (Millions of Dollars)

# Nursing Home Care (Millions of Dollars)

# Prescription Drugs (Millions of Dollars)

# Durable Medical Products (Millions of Dollars)

# Other Non-durable Medical Products (Millions of Dollars)

#-----

*\*Personal Health Care (Millions of Dollars) Top 10 States*

import matplotlib.pyplot as plt

*\*Filter df for item*

healthcare\_item\_data = med\_data\_long\_df[med\_data\_long\_df['Item'] == 'Personal Health Care (Millions of Dollars)']

*\*Summing up the spending for each state and selecting the top 10 states for total spending (1980-2020)*

total\_spending\_by\_state = healthcare\_item\_data.groupby('State')['Spending'].sum().nlargest(10)

top\_states = total\_spending\_by\_state.index.tolist()

print(top\_states)

*\*Setting the figure size*

plt.figure(figsize=(15, 8))

*\*Plotting each of the top states with a loop*

*for state in top\_states:*

state\_data = healthcare\_item\_data[healthcare\_item\_data['State'] == state]

state\_data = state\_data.sort\_values('Year')

plt.plot(state\_data['Year'], state\_data['Spending'], label=state, linewidth=2, marker='o') # Adjusted line width and added markers

*\*Show every other year on x-axis*

plt.xticks(state\_data['Year'].unique()[::2])

*\*Adding title and labels*

plt.title('Top 10 States in Personal Health Care (Millions of Dollars) Over Time', fontsize=16)

plt.xlabel('Year', fontsize=14)

plt.ylabel('Spending (in Millions)', fontsize=14)

*\*Adding a legend*

plt.legend()

# Show the plot

plt.show()

#####################################################

***\*Creating the Bokeh models – repeat for each state, California, Texas, Florida, New York Pennsylvania,***

*\*Dependencies*

import pandas as pd

from bokeh.plotting import show

from bokeh.models import ColumnDataSource, DataTable, TableColumn

*\*Create df variable to represent med\_data\_long\_df filtered on state by name*

df = med\_data\_long\_df[med\_data\_long\_df['State'] == 'California']

*\*Run summary statistics*

stats = df.describe().reset\_index()

print(stats.reset\_index())

*\*Create source and column variables for DataTable*

source = ColumnDataSource(stats)

columns = [TableColumn(field=col, title=col) for col in stats.columns]

*\*Print to show Bokeh DataTable*

data\_table = DataTable(source=source, columns=columns, width=800, height=400, index\_position=None)

show(data\_table)

*\*Filter by multiple states using list comprehension*

filtered\_df = med\_data\_long\_df[med\_data\_long\_df['State'].isin(['California', 'Vermont', 'Alaska', 'North Dakota'])]

*\*Get summary statistics*

stats = filtered\_df.describe().reset\_index()

print(stats) # No need for reset\_index() here

#####################################################

**Ethical considerations made in the project:**

This project utilizes de-identified, publicly available data that contains no personal health information. No persons were further contacted regarding this dataset.

**References for the data source(s)**

Centers for Medicare & Medicaid Services (2022). *Health Expenditures by State of Provider*. Retrieved February 2024 at <https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/NationalHealthExpendData/Downloads/provider-state-estimates.zip>

**References for any code**

# https://docs.bokeh.org/en/3.0.2/

# https://plotly.com/python/dropdowns/

# https://stackoverflow.com/questions/61750811/dropdown-menu-for-plotly-choropleth-map-plots

# https://community.plotly.com/t/how-to-modify-points-drawn-on-map-using-a-dropdown-menu/63807

# intersphinx\_mapping = {'bokeh': ('https://docs.bokeh.org/en/latest/', Plotting)}

# intersphinx\_mapping = {'bokeh': ('https://docs.bokeh.org/en/latest/', Models)}