Heart Disease Across the US and Predictive Modeling

Dirk Hartog

Data 602 - Advanced Programming Final Project

Abstract

According to a 2024 report from the American Heart Association, heart disease has been the leading cause of death in the U.S. for 100 years. Awareness of such trends among the general population is an important part of preventative strategies to reduce the burden of heart disease. The data set I choose to work with includes data on mortality rates or percentages on cardiovascular disease in US counties between the years of 1999 - 2019. It comes from the National Vital Statistics System and the csv file can be found at Data.gov. The purpose of this analysis is to identify trends in mortality rates at the US county level and uncover any inequalities among races, genders and age groups. Various Python libraries and visualization techniques were used to extract the data and present findings. In general mortality rates have continued to decline over the past 20 years. Certain subgroups suffer higher mortality rates (Men, blacks (non hispanic), and those 65 years or older).

Part two explores the use of a data set from Kaggle to predict the presence or absence of heart disease. Understanding a model's accuracy can give insights into factors to monitor or address when collecting data for research and potentially monitor or target to prevent or slow onset of disease. Building a classification model using K-nearest neighbors we reach a moderate to high level of accuracy.

Further directions can be aimed at uncovering trends within subgroups from each county. Comparing different classification models is needed to determine the most accurate strategy and to deepen understanding of how the features available in this data influence the prediction of heart disease.

Research Question/Introduction

1. What were the trends in mortality rates from heart disease during the time frame in dataset?

2. What were differences in rates between race, gender, and age?

3. How accurately can we predict the presence or absence of heart disease?

Part 1:
Heart Disease Across the US

Identifying trends in mortality rates

Exploratory Data Analysis: Exploring the contents

- Large data set with 5,770,205 rows and 21 columns
- The data contains observations of heart disease mortality rates in 1,829 unique US counties from 1999 - 2019
- Mortality rate measurements and Missing values
  - Total Percent Change
  - Age-Standardized, Spatiotemporally Smoothed Rate
  - Approximately 41% (2,365,475 observations) of the rates were missing
  - Rates were not reported for all levels of Stratification
    - Ex. No rates were reported for specific races and specific sex categories (API and Male)
    - Rates among race categories were reported as **overall** rates for the “men and women”
  - Some counties did not report on all race

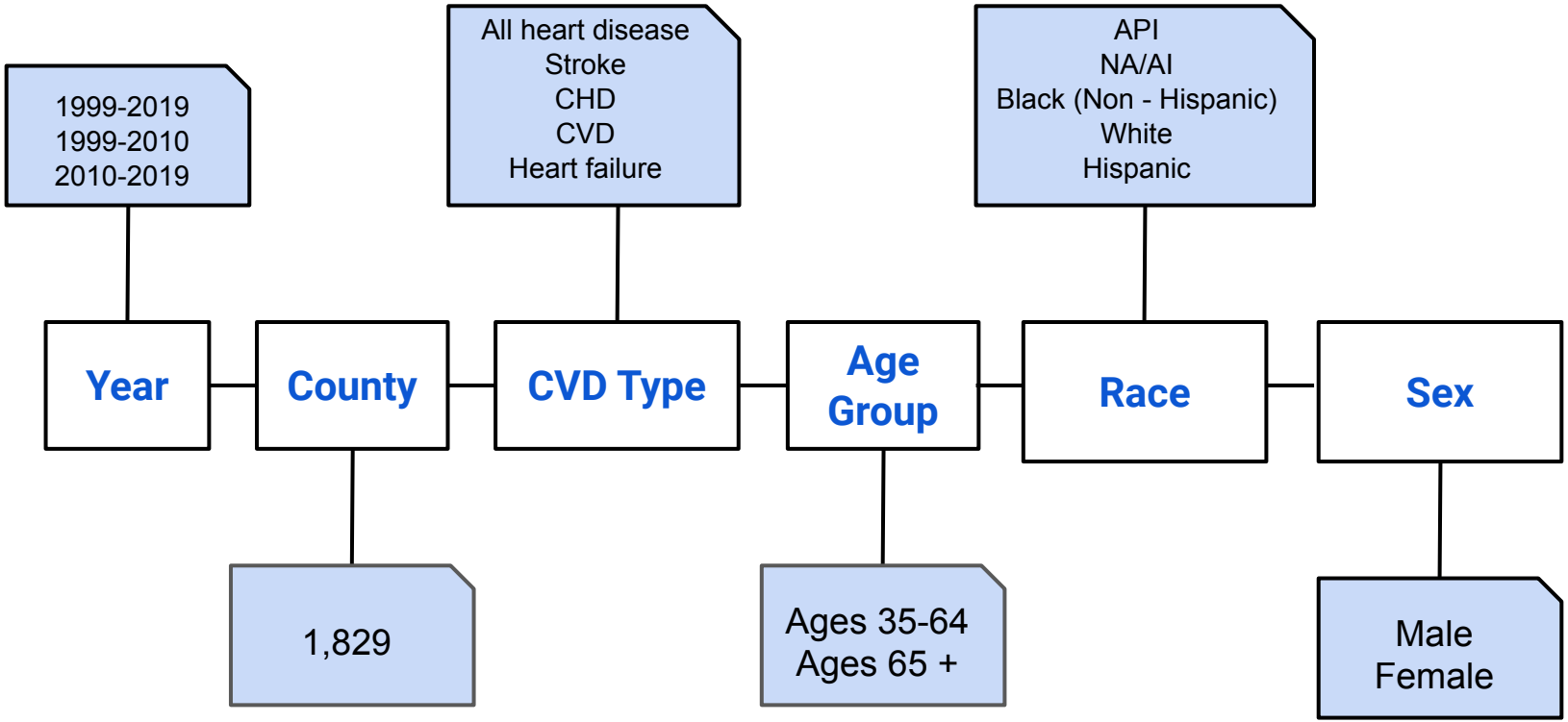
Libraries Used

```
# Pandas
import pandas as pd
# Datetime
from datetime import datetime as dt
# Numpy
import numpy as np
# Matplotlib
import matplotlib.pyplot as plt
# Seaborn
import seaborn as sns
```

```
heart_disease[heart_disease["Data_Value"].notna()].groupby(["LocationDesc",
                                                            "Year",
                                                            "Topic",
                                                            "Stratification1",
                                                            "Stratification2",
                                                           "Stratification3"])["Data_Value"].count().head(50)
```

LocationDesc	Year	Topic	Stratification1	Stratification2	Stratification3	
Abbeville	1999	All heart disease	Ages 35-64 years	Black (Non-Hispanic)	Overall	1
				Overall	Men	1
					Overall	1
					Women	1
				White	Overall	1
			Ages 65 years and older	Black (Non-Hispanic)	Overall	1
				Overall	Men	1
					Overall	1
					Women	1
				White	Overall	1
		All stroke	Ages 35-64 years	Black (Non-Hispanic)	Overall	1
				Overall	Men	1
					Overall	1
					Women	1
				White	Overall	1
			Ages 65 years and older	Black (Non-Hispanic)	Overall	1
				Overall	Men	1
					Overall	1
					Women	1
				White	Overall	1
		Cardiovascular disease (CVD)	Ages 35-64 years	Black (Non-Hispanic)	Overall	1
				Overall	Men	1
					Overall	1
					Women	1

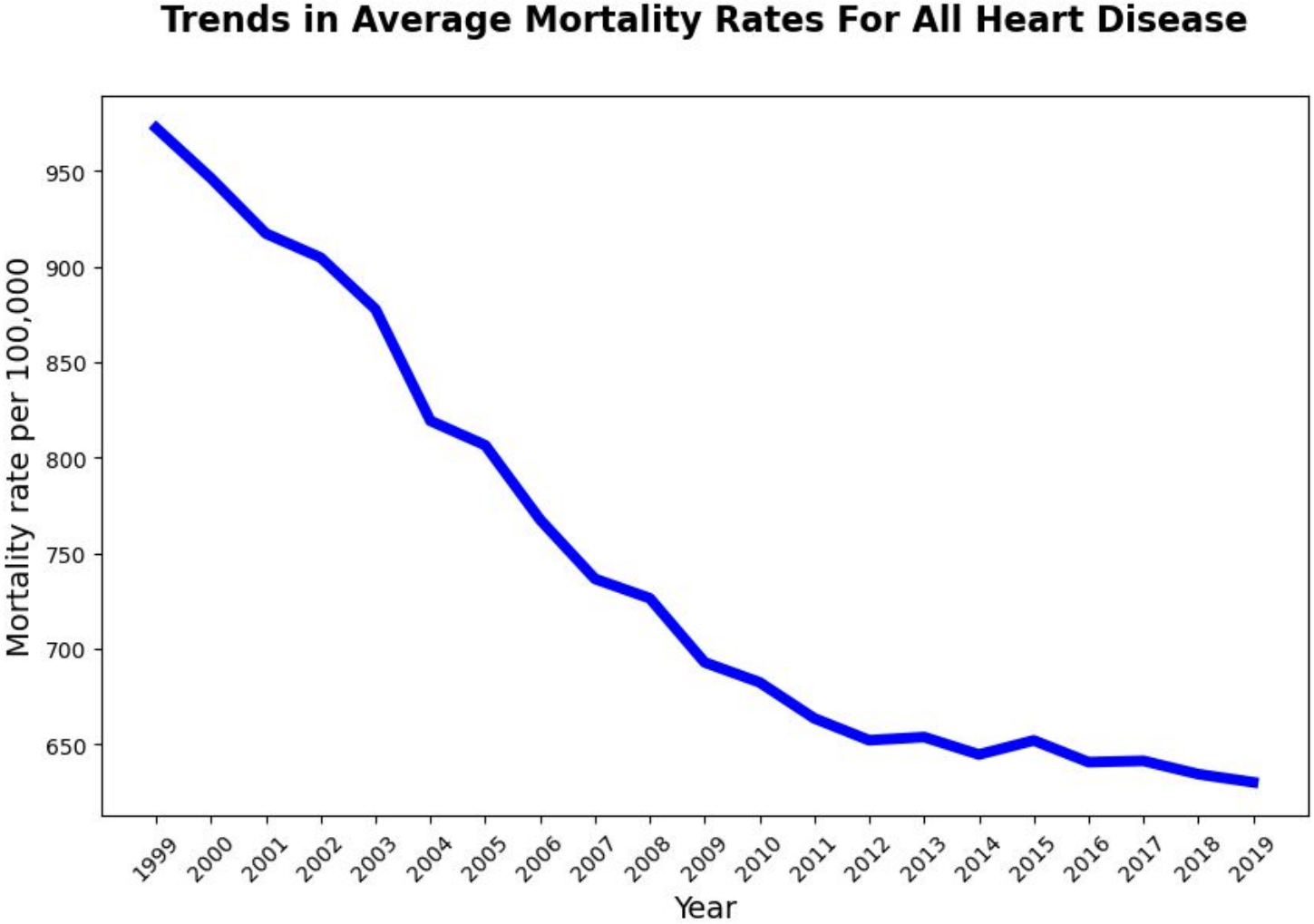
Exploratory Data Analysis: Data Stratification



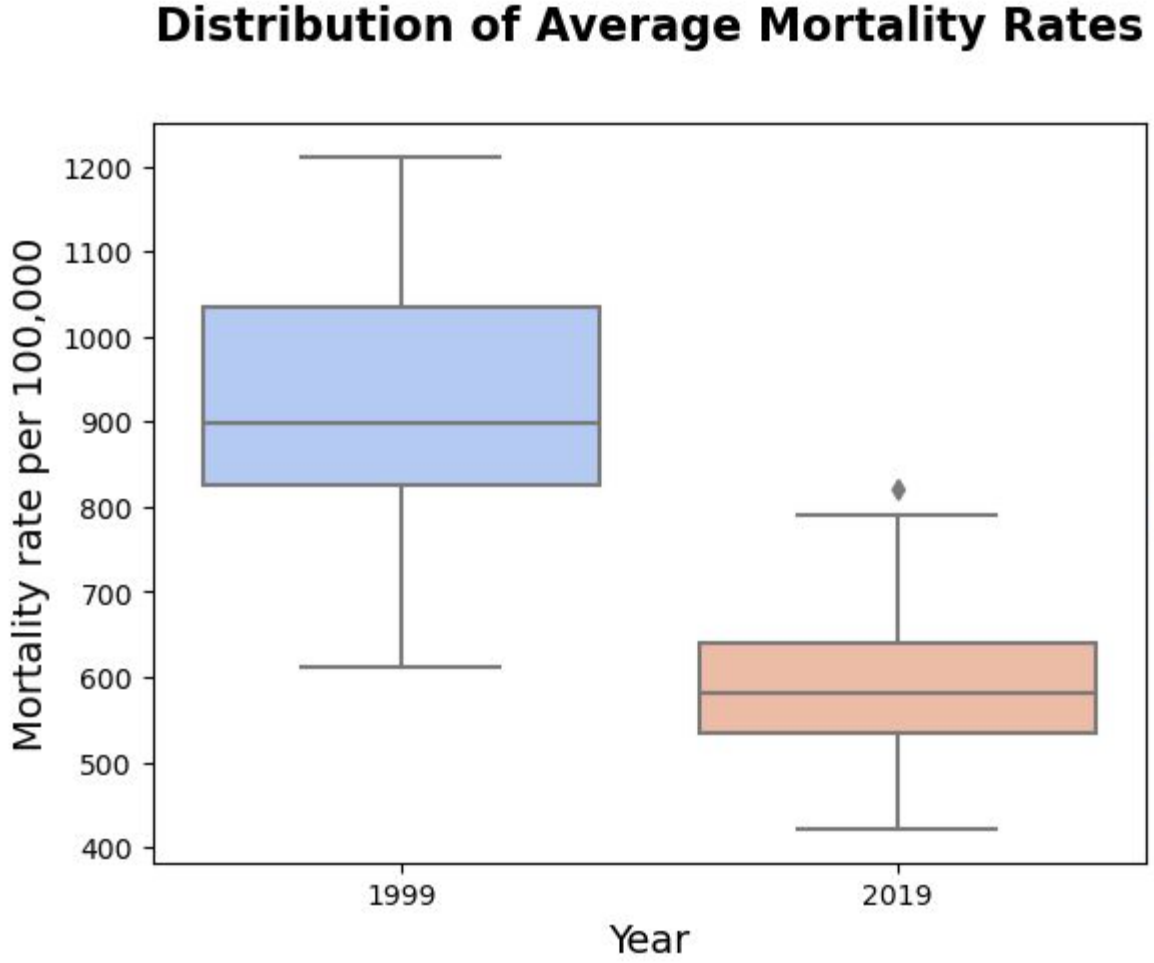
```
df_yearly_trends = heart_disease[(heart_disease["Topic"] == "All heart disease") &
                                 (heart_disease["Stratification3"] == "Overall") &
                                 (heart_disease["Stratification2"] == "Overall") &
                                 (heart_disease["Data_Value_Type"] =='Age-Standardized, Spatiotemporally Smoothed Rate')]


df_yearly_trends.head()
```

```
gb_yearly_trends = df_yearly_trends.groupby("Year")["Data_Value"].mean().reset_index()

gb_yearly_trends
```

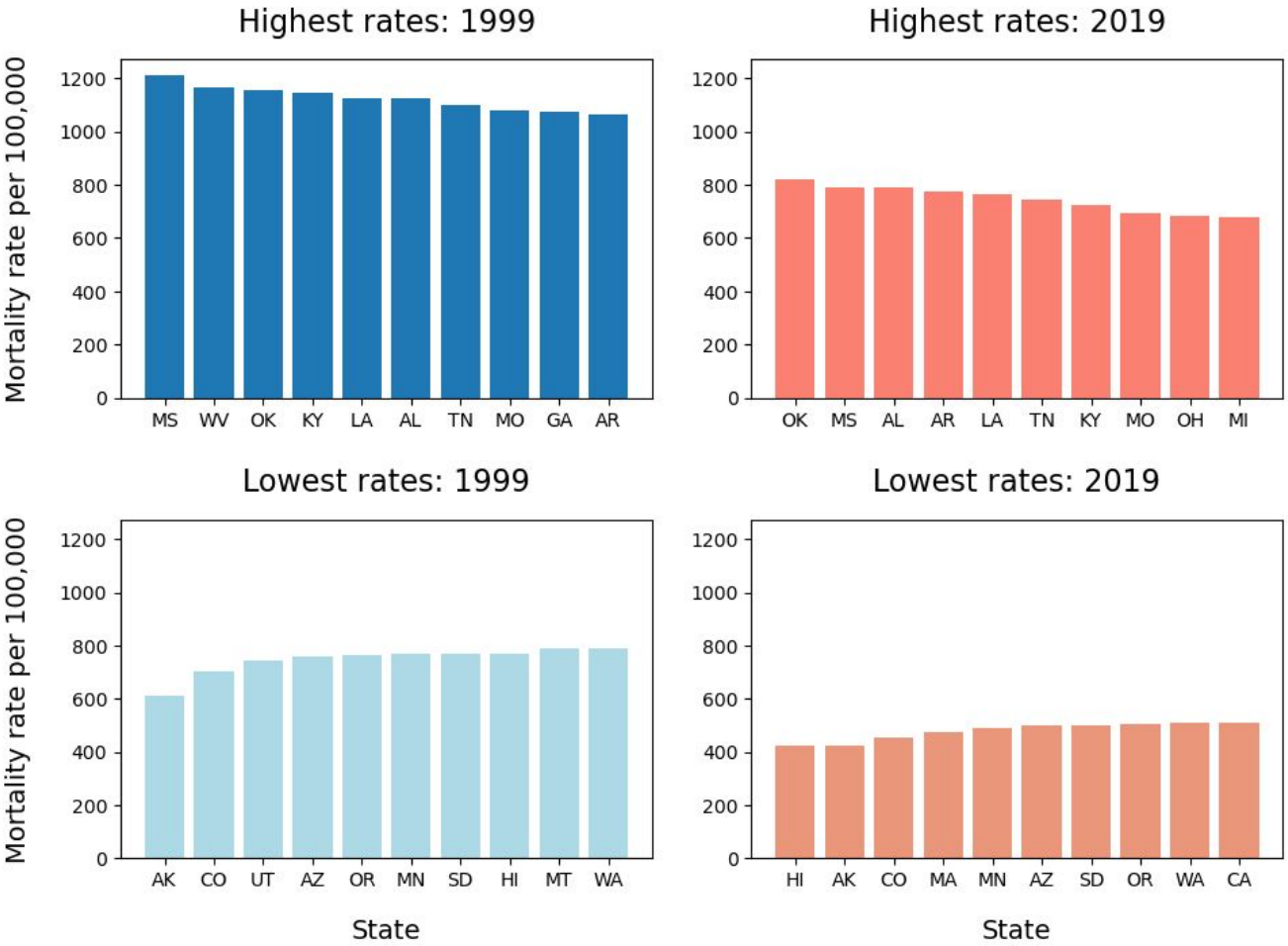
```
# Plot the trend in mortality rate for all heart disease from group by object above.

plt.figure(figsize=(10,6))
plt.plot(gb_yearly_trends["Year"], gb_yearly_trends["Data_Value"],markersize = 4, color = "blue", linewidth = 5)
plt.xticks(rotation = 45)
plt.title("Trends in Average Mortality Rates For All Heart Disease", size = 16, fontweight = "bold", pad = 30.0)
plt.ylabel(ylabel = "Mortality rate per 100,000", size = 14)
plt.xlabel(xlabel = "Year", size = 14)

plt.show()
```


Trends in Average Mortality Rates For All Heart Disease
Mortality rate per 100,000
950
900
850
800
750
700
650
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Year

```
# Filter to pull out the overall all heart disease mortality rates

overall_rate = heart_disease.copy()

overall_rate = overall_rate[(overall_rate["Topic"] == "All heart disease") &
      (overall_rate["Data_Value_Type"] == 'Age-Standardized, Spatiotemporally Smoothed Rate') &
      (overall_rate["Stratification3"] == "Overall") &
      (overall_rate["Stratification2"] == "Overall")]

# Change data type of year to integer to filter only the min and max year

overall_rate["Year"] = overall_rate["Year"].astype("int")

overall_rate = overall_rate[(overall_rate["Year"] == overall_rate["Year"].max()) | (overall_rate["Year"] == overall_rate["Year"


# Groupby year and location and get mean

overall_mean = overall_rate.groupby(["Year", "LocationAbbr"])["Data_Value"].mean()

# Create a box plot using plotly express to see range of data among the two years
overall_mean = overall_mean.reset_index()

overall_boxplot = sns.boxplot(data = overall_mean,
                          y = "Data_Value",
                          x = "Year",
                          hue = "Year",
                          palette = "coolwarm", dodge = False)
plt.title("Distribution of Average Mortality Rates", size = 16, fontweight = "bold", pad = 30.0)
plt.ylabel(ylabel = "Mortality rate per 100,000", size = 14)
plt.xlabel(xlabel = "Year", size = 14)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5), fontsize = 10)
plt.show(overall_boxplot)
```

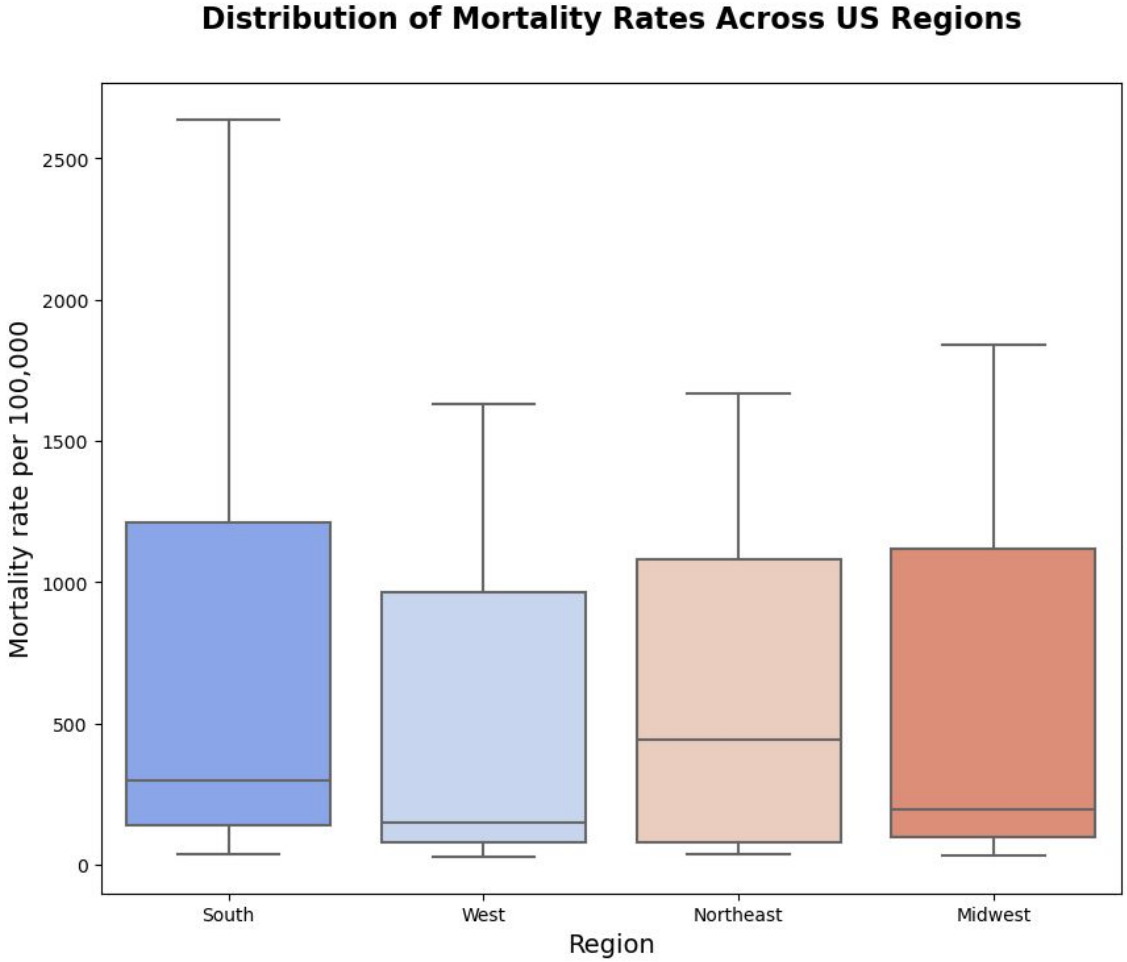

Distribution of Average Mortality Rates
Mortality rate per 100,000
1200
1100
1000
900
800
700
600
500
400
1999
2019
Year

Highest and Lowest Average All Heart Disease Mortality Rates



```
# Read in region/divisionlabeled data

url = "https://raw.githubusercontent.com/cphalpert/census-regions/master/us%20census%20bureau%20regions%20and%20divisions.csv"

regions_div = pd.read_csv(url)

# Filter for the state code, region and division
regions_div = regions_div[["State Code", "Region", "Division"]]

# Join heart disease df and region df on LocationAbbr == State Code
heart_disease = heart_disease.merge(regions_div, how = "left",
                                    left_on = "LocationAbbr", right_on = "State Code").drop("State Code", axis = 1)


region_dist_2019 = heart_disease[(heart_disease["Topic"] == "All heart disease") &
                                 (heart_disease["Year"] == "2019") &
                                 (heart_disease["Stratification3"] == "Overall") &
                                 (heart_disease["Stratification2"] == "Overall")]
```

```
plt.figure(figsize=(10,8))
region_boxplot = sns.boxplot(data = region_dist_2019,
                             x = "Region",
                             y = "Data_Value",
                             hue = "Region",
                             width = 0.8,
                             palette = "coolwarm",
                             dodge = False)
plt.title("Distribution of Mortality Rates Across US Regions", size = 16, fontweight = "bold", pad = 30.0)
plt.ylabel(ylabel = "Mortality rate per 100,000", size = 14)
plt.xlabel(xlabel = "Region", size = 14)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5), fontsize = 10)
plt.show(region_boxplot)
```

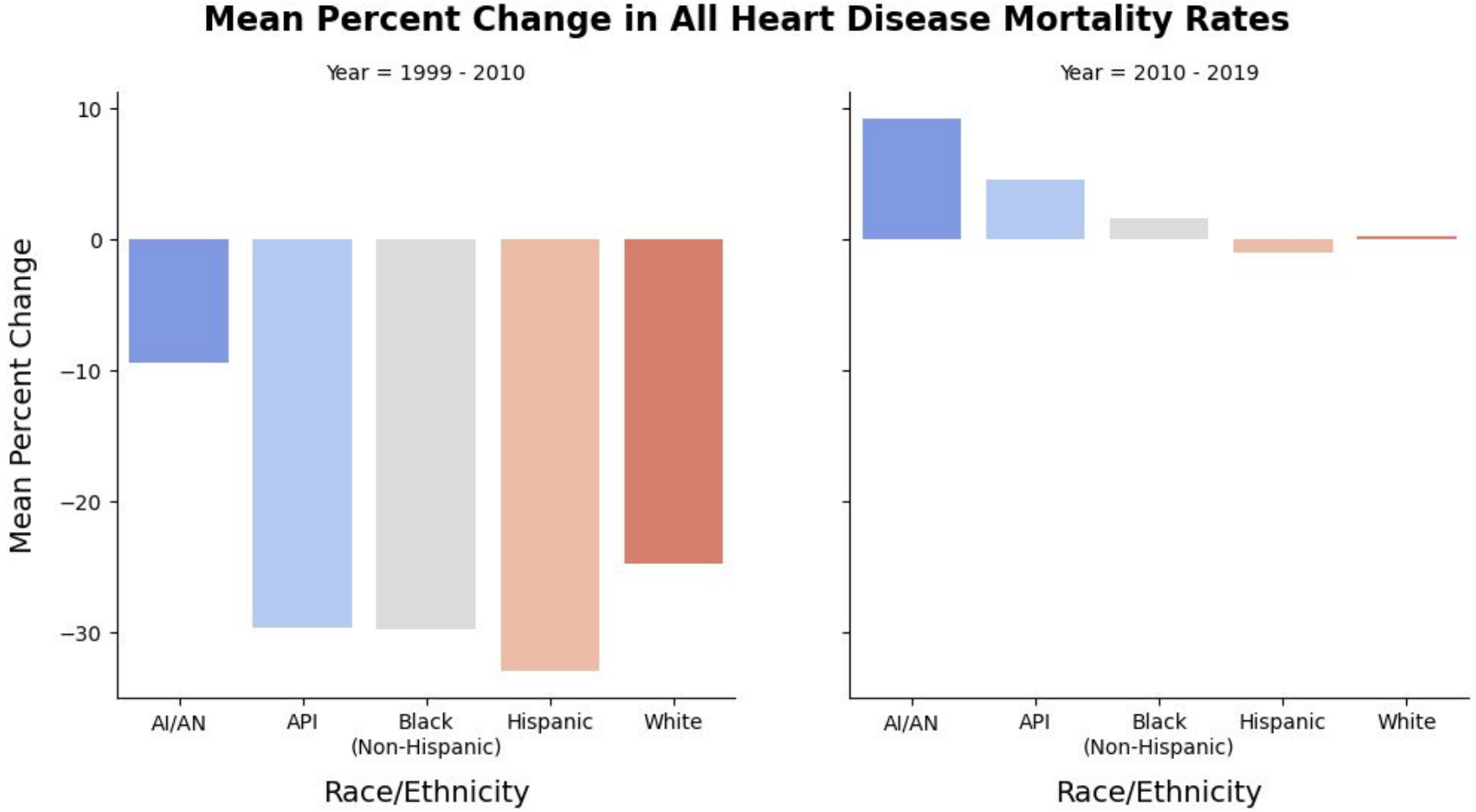

Distribution of Mortality Rates Across US Regions
Mortality rate per 100,000
2500
2000
1500
1000
500
0
South
West
Northeast
Midwest
Region

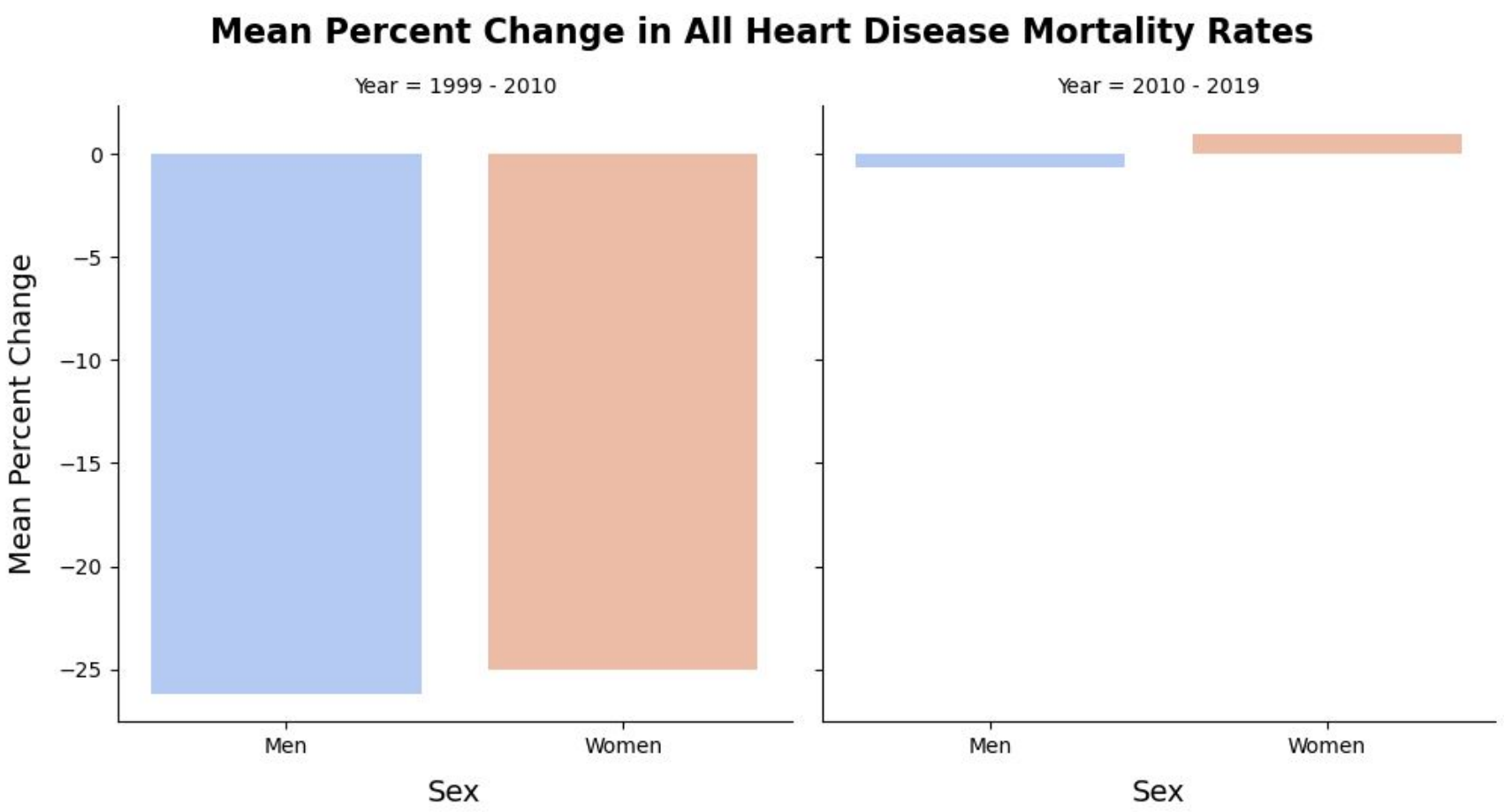
Rates Between Subgroups

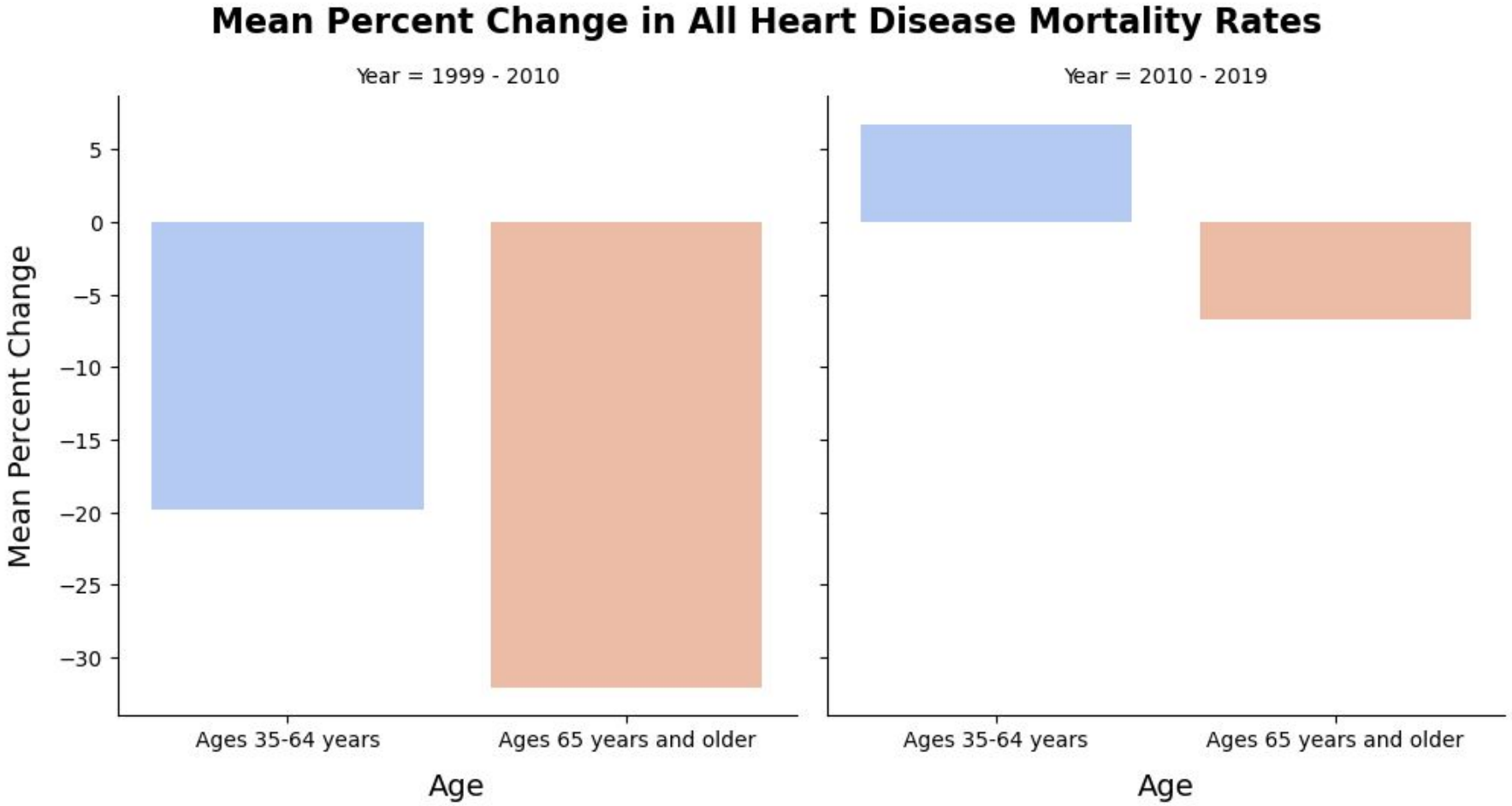
```
# Summary statistics by RACE/ETHNICITY and All heart disease

df1 = heart_disease[(heart_disease["Topic"] == "All heart disease") &
              (heart_disease["Data_Value_Type"] == "Total percent change") &
              (heart_disease["Stratification3"] == "Overall") &
              (heart_disease["Stratification2"] != "Overall")
             ].dropna(axis = 0, subset = "Data_Value")


race_ethnicity = df1.groupby(["Year", "Stratification2"])["Data_Value"].describe()

race_ethnicity
```

Year	Race	Mean	Stnd. Dev.	Min	25%	50%	75%	Max
1999 - 2010	American Indian/Alaska Native	**-9.414857**	23.330523	-61.2	-24.90	-12.1	1.600	105.2
	Asian/Pacific Islander	**-29.606522**	12.398880	-59.9	-37.05	-30.9	-23.200	69.1
	Black (Non-Hispanic)	**-29.705465**	13.409396	-67.1	-38.40	-30.9	-23.000	248.7
	Hispanic	**-32.861399**	13.076478	-72.2	-41.55	-34.1	-25.900	40.7
	White	**-24.690683**	13.270857	-60.7	-34.30	-27.1	-17.575	47.7
2010 - 2019	American Indian/Alaska Native	**9.157143**	23.641501	-46.8	-7.40	7.5	23.600	146.9
	Asian/Pacific Islander	**4.534353**	23.233030	-37.7	-13.70	1.0	19.200	147.0
	Black (Non-Hispanic)	**1.546256**	16.824125	-50.5	-10.30	0.2	11.900	82.9
	Hispanic	**-0.997285**	14.648169	-49.9	-11.70	-2.4	7.600	80.8
	White	**0.270210**	14.211688	-38.5	-9.40	-1.3	8.000	140.0

```
# Create bar chart visualizing the mean percent change between RACES/ETHNICITY for each time period

race = ["AI/AN", "API", "Black\n(Non-Hispanic)", "Hispanic", "White"]

race_ethnicity_plot = sns.catplot(data = race_ethnicity,
            x = "Stratification2",
            y = "mean",
            kind = "bar",
            col = "Year", palette = "coolwarm")

race_ethnicity_plot.set_xticklabels(labels = race, rotation = 0)
race_ethnicity_plot.fig.suptitle("Mean Percent Change in All Heart Disease Mortality Rates",
                                 y = 1.05, size = 16, fontweight = "bold")
race_ethnicity_plot.set_xlabels(label = "Race/Ethnicity", labelpad = 10.0, fontsize = 14)
race_ethnicity_plot.set_ylabels(label = "Mean Percent Change", labelpad = 10.0, fontsize = 14)
plt.show()
```


Mean Percent Change in All Heart Disease Mortality Rates
Year = 1999 - 2010
Year = 2010 - 2019
Mean Percent Change
10
0
−10
−20
−30
AI/AN
API
Black (Non-Hispanic)
Hispanic
White
Race/Ethnicity

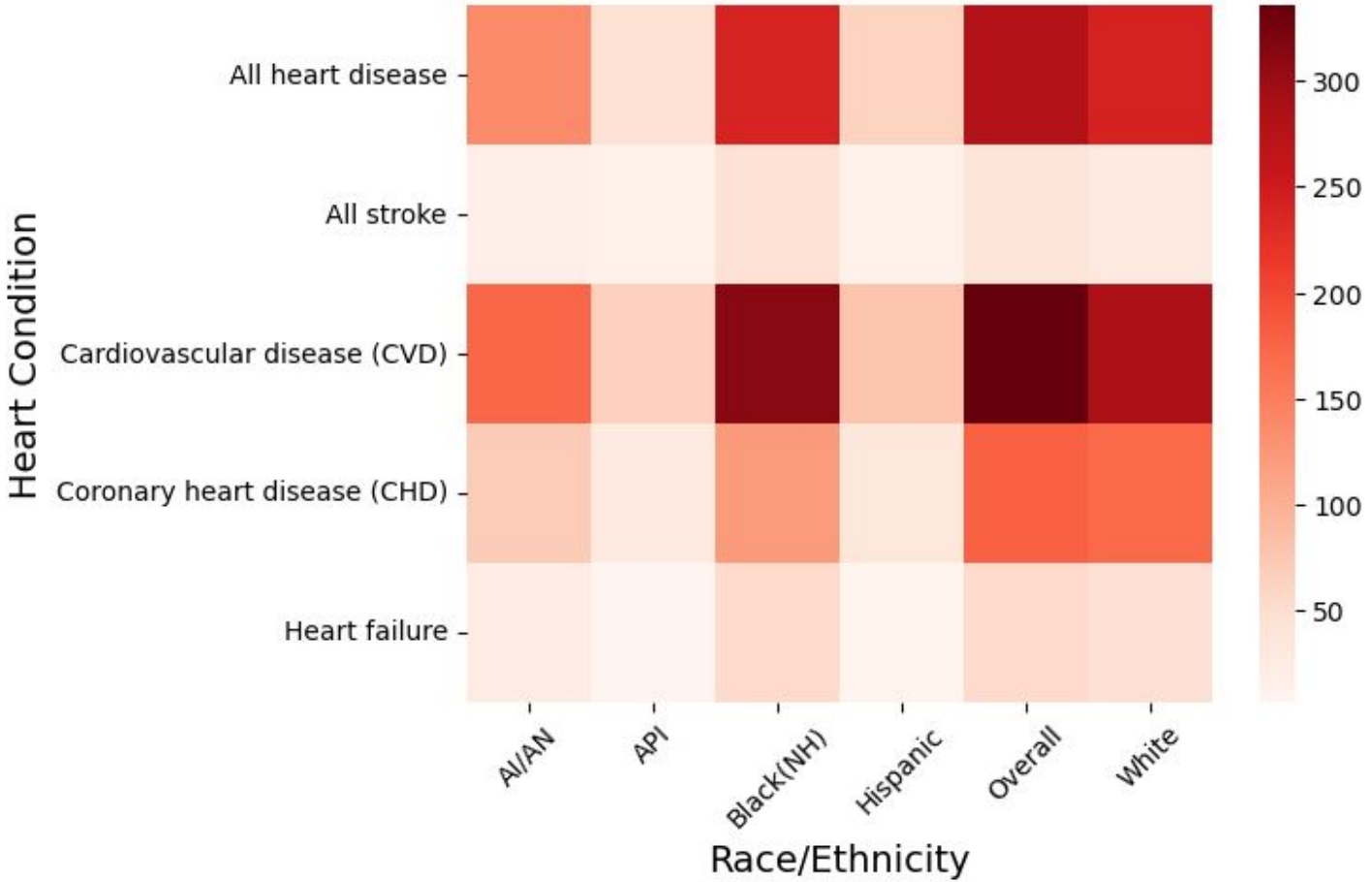

Mean Percent Change in All Heart Disease Mortality Rates
Year = 1999 - 2010
Year = 2010 - 2019
Mean Percent Change
0
−5
−10
−15
−20
−25
Men
Women
Sex
Men
Women
Sex


Mean Percent Change in All Heart Disease Mortality Rates
Year = 1999 - 2010
Year = 2010 - 2019
Mean Percent Change
5
0
−5
−10
−15
−20
−25
−30
Ages 35-64 years
Ages 65 years and older
Age
Ages 35-64 years
Ages 65 years and older
Age

Heart Condition	RACE					
	AI/Aative	API	Black (Non-Hispanic)	Hispanic	Overall	White
All heart disease	137.7	44.5	240.30	61.5	278.3	241.25
All stroke	18.7	13.3	44.90	13.4	38.3	27.45
Cardiovascular disease (CVD)	174.4	64.8	312.15	78.3	335.6	284.65
Coronary heart disease (CHD)	70.9	27.9	121.80	35.7	179.0	170.95
Heart failure	22.9	6.8	54.15	9.8	53.7	45.50

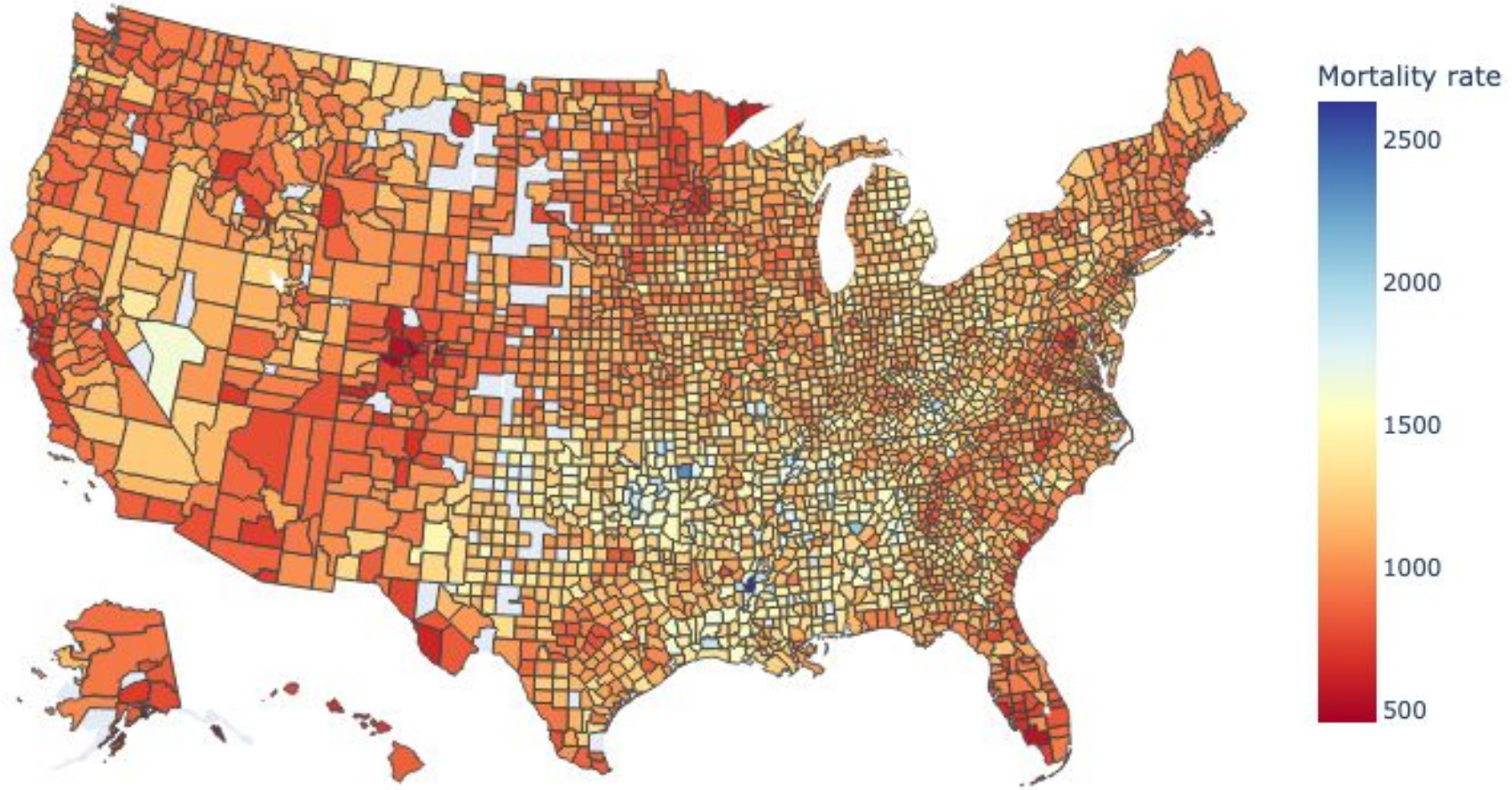
```
# Visualize mortality rates among the different races and topic

recent_trend = heart_disease[heart_disease["Year"] == "2019"]
recent_trend = recent_trend.groupby(["Stratification2", "Topic"])["Data_Value"].describe().reset_index()
recent_trend = recent_trend[["Stratification2", "Topic", "50%"]]
recent_trend = recent_trend.pivot(index="Topic", columns="Stratification2", values="50%")
recent_trend


sns.heatmap(recent_trend, xticklabels=["AI/AN", "API", "Black(NH)", "Hispanic", "Overall", "White"], cmap = "coolwarm
plt.ylabel(ylabel = "Heart Condition", fontsize = 14)
plt.xticks(rotation = 45)
plt.xlabel(xlabel = "Race/Ethnicity", fontsize = 14)
plt.title("2019 Mortality Rates By Race and Heart Condition", pad = 30.0, fontsize = 16, fontweight = "bold")
plt.show()
```

2019 Mortality Rates By Race and Heart Condition



```
from urllib.request import urlopen
import json

with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)

fig = px.choropleth(map_sixty_five_plus,
                    geojson=counties,
                    locations='LocationID_str',
                    color='Data_Value',
                    color_continuous_scale='rdylbu',
                    #range_color=(0, 12),
                    scope="usa",
                    labels={'Data_Value':'Mortality rate'},
                    title = "2019 US Mortality Rate by County"
                   )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

2019 Mortality Rates Across US Counties: Age 65 Year or Older



Conclusions

- Since 1999 and up until 2019 the average mortality rates from heart disease were decreasing across all the US.
- States with the highest and lowest average mortality rates remained largely unchanged when comparing data from 1999 and 2019.
- Among the 4 regions of the US the Northeast had the highest median mortality rate, where as the West had the lowest.
- Between 2010 and 2019 we saw that the mean total percent change was positive indicating an increase in mortality rates among white, black(non - Hispanic), API, AI/AN, women and those aged 35-64 years of age.
- The largest average total percent change in mortality rates was among the AI/AN.
- Black (non-Hispanic) individuals had the highest mortality rates due to stroke, CVD, and heart failure.

Discussions/Limitations

- Most recent data is about 5 years old so generalizing the findings to today is not appropriate.
- Taking the average mortality rates across all counties when looking at subgroups and years may not have been the best metric to generate comparisons.
- The analysis did reveal potential areas where inequalities might exist.
- Zooming on counties where mortality rates are highest among subgroups would be the next step in identifying where public health education might be targeted.
- The Age-Standardized Rates (ASRs) allow for meaningful comparisons of the prevalence or incidence of health-related events between different populations, subgroups or over time, while adjusting for differences in age structures. These rates can be compared to those of other countries if data exists.

Part 2:
Heart Disease Prediction

Building a KNN classification model

Libraries Used

```
import sklearn
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split, cross_val_score, KFold, GridSearchCV, RandomizedSearchCV
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, roc_auc_score
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
```

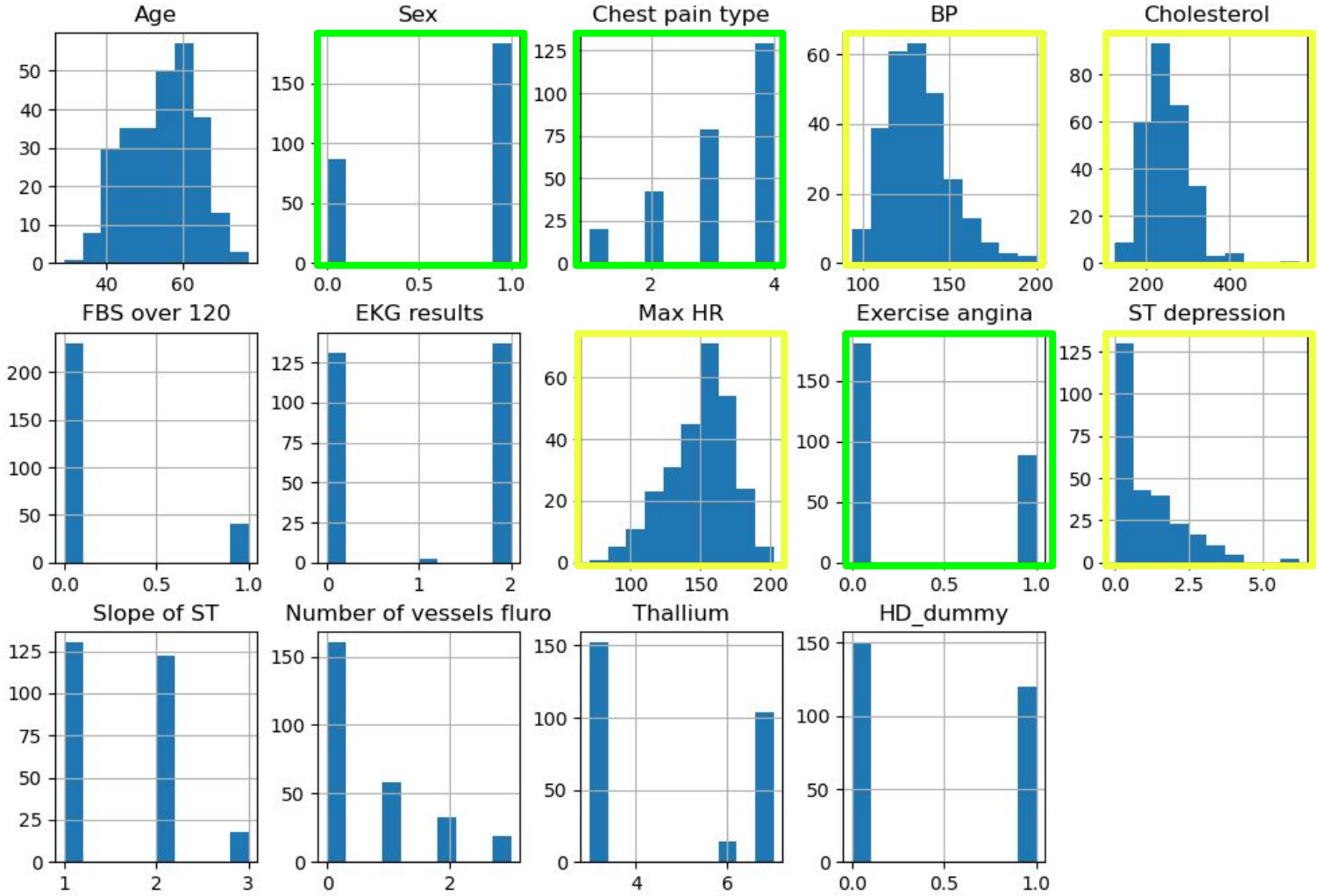
Introduction/Purpose

- Build a classification model to predict those who have heart disease using the features given in the data from kaggle

- The data contains 14 features including our target column "Heart Disease" that notes the "Absence" or "Presence" of heart disease.

Exploratory Data Analysis

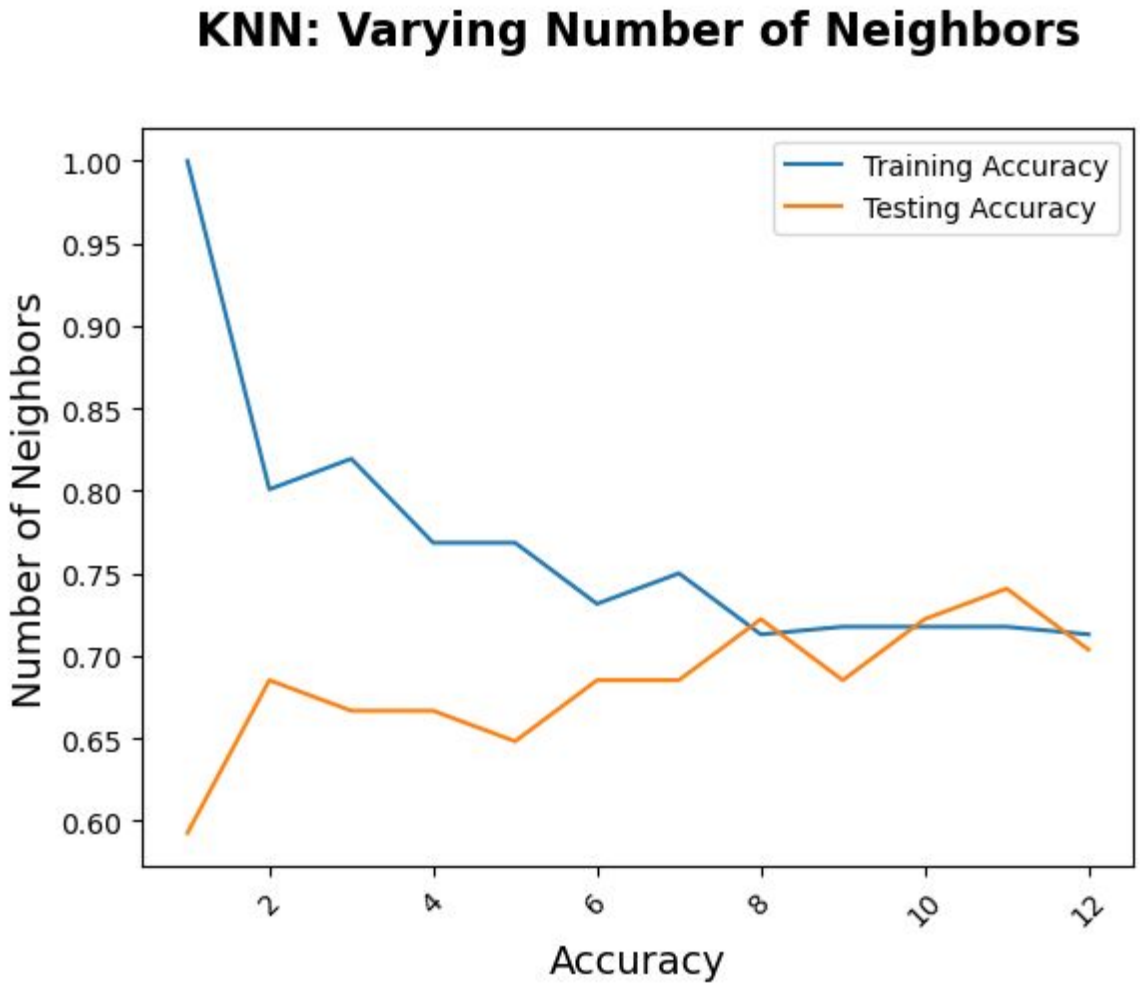
- No missing values and minimal to no data cleaning was necessary.
  - Categorical variables were already transformed into numeric representations except the target variable.
- There were 270 observations
- Average age of the individuals in this data set were 54 years old
- Approx. 67% were males
- Approx. 44% of the observations were labeled with the presence of heart disease


Age
Sex
Chest pain type
BP
Cholesterol
FBS over 120
EKG results
Max HR
Exercise angina
ST depression
Slope of ST
Number of vessels fluro
Thallium
HD_dummy

```
# Seperate our data into X and y variables and split the data into training and testing sets

X = hd_prediction.drop(["Heart Disease", "HD_dummy"], axis = 1).values
y = hd_prediction["HD_dummy"].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Create neighbors
neighbors = np.arange(1, 13) # 1 - 12
train_accuracies = {}
test_accuracies = {}

for neighbor in neighbors:

# Set up a KNN classifier
    knn = KNeighborsClassifier(n_neighbors=neighbor)

# Fit the model
    knn.fit(X_train, y_train)

    train_accuracies[neighbor] = knn.score(X_train, y_train)
    test_accuracies[neighbor] = knn.score(X_test, y_test)
```

```python
# Visualizing model complexity
import matplotlib.pyplot as plt

# Add a title
plt.title("KNN: Varying Number of Neighbors")

# Plot training accuracies
plt.plot(neighbors, train_accuracies.values(), label="Training Accuracy")

# Plot test accuracies
plt.plot(neighbors, test_accuracies.values(), label="Testing Accuracy")

plt.legend()
plt.xlabel("Number of Neighbors")
plt.ylabel("Accuracy")

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

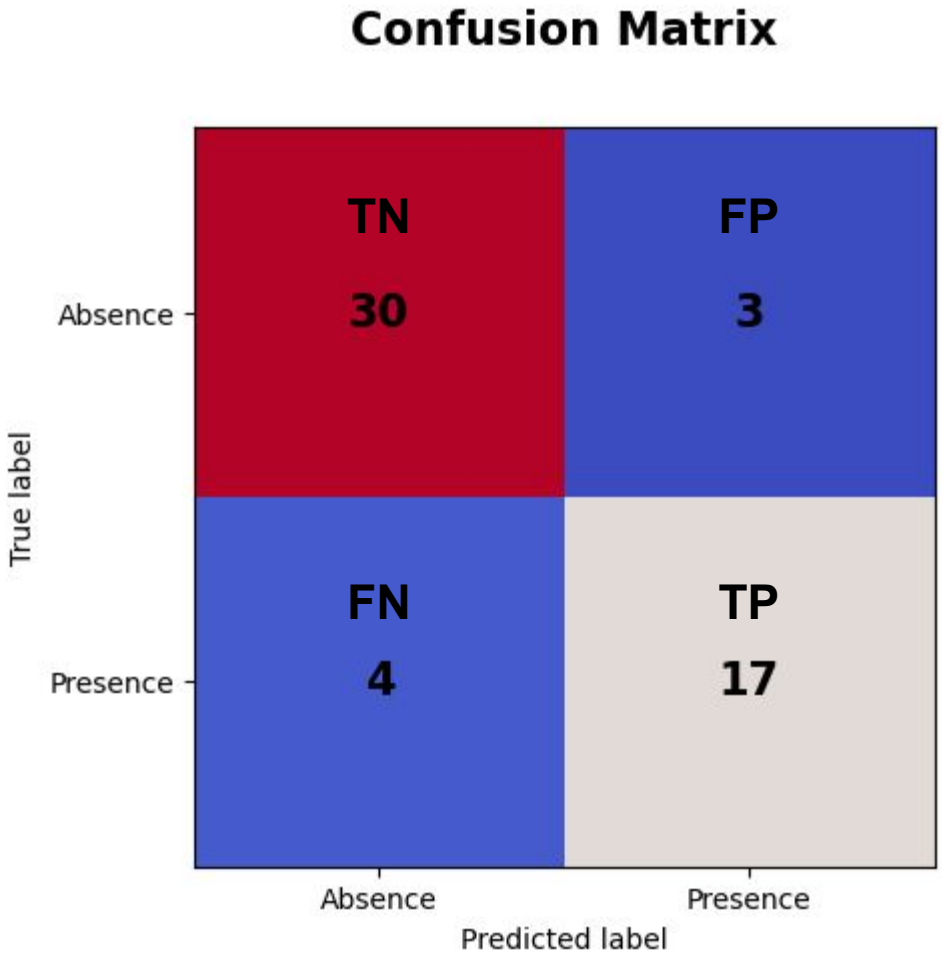

KNN: Varying Number of Neighbors
Training Accuracy
Testing Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Number of Neighbors
2
4
6
8
10
12
Accuracy

```
# Create a pipeline and pass that into a grid search to find the optimal parameters for our KNN model

# Build the steps
steps = [("scaler", StandardScaler()),
         ("KNN", KNeighborsClassifier())]

pipeline = Pipeline(steps)

#Create the parameter space
parameters = {"KNN__n_neighbors": np.arange(1, 13),
              'KNN__weights':['uniform','distance'],
              'KNN__p': np.arange(1,3),
              "KNN__algorithm":['auto','ball_tree','kd_tree','brute']}

# Instantiate the grid search object
cv = GridSearchCV(pipeline, param_grid = parameters)

# Fit to the training data
cv.fit(X_train, y_train)
print(cv.best_score_, "\n", cv.best_params_)
```

Best Score	***Best Parameters***
0.8476744186046512	**algorithm:** 'auto' **n_neighbors**: 11 **p**: 2 **weights**: 'uniform'

Build Pipeline with best parameters and scaled data

```
# With the optimal parameters found we can re run the KNN model and generate a confusion matrix and classification report

knn = KNeighborsClassifier(n_neighbors = 11, algorithm = "auto", p = 2, weights = "uniform")

steps = [("scaler", StandardScaler()),
         ("KNN", KNeighborsClassifier(n_neighbors = 11,
                                      algorithm = 'auto',
                                      p = 2, weights = 'uniform'))]
pipeline = Pipeline(steps)

knn_scaled = pipeline.fit(X_train, y_train)

y_pred = knn_scaled.predict(X_test)
print(knn_scaled.score(X_test, y_test))

cf_matrix = confusion_matrix(y_test, y_pred)

print(cf_matrix)

cr = classification_report(y_test, y_pred)
print(cr)
```



Precision	0.85
Recall (Sensitivity)	0.81
Specificity	0.91
Accuracy	0.87
F1 score	0.83

Discussion/Conclusions

- Using the classification model K-Nearest Neighbors resulted in good accuracy at 87% in predicting heart disease from the features that were included.
- The model is better at predicting negative or absent cases as the model's specificity is high at 91%.
- Depending on the population and potential intervention, our precision at 85% is good.
    - If the intervention was prescribing a consult with a dietician or an exercise specialist, identifying someone who has heart disease even with low precision might not be a bad thing. This might change if the intervention involves drugs or other invasive treatments where you might want higher accuracy.
- Comparison of other classification models may reveal even better accuracy and need would need to be tested and compared against this one.

Thank you!