ISSUES

The data presented here is sourced from the Centers for Disease Control and Prevention and pertains to the rates of diabetes, obesity, and inactivity in U.S. counties for 2018. The dataset includes information about U.S. counties, their respective states, FIPS codes, the year of data collection, % Diabetes, % Obesity, and % Inactivity.

Further, we have obtained a dataset from the year 2018 containing information about states governed by either the Republican or Democratic party.

The questions we aim to address are as follows:

1. Is there a correlation between the percentages of diabetes, obesity, and inactivity?
2. Do counties or states have any patterns to help us understand where health issues are most common?
3. Is there a correlation between the governing political party and the health of its residents?
4. How do we consider and address individual metrics among the states, i.e., diabetes, obesity and inactivity?

FINDINGS

From the data by CDC of the year 2018, we were able to conclude the following:

1. We found Diabetes is more common when obesity and inactivity increases for the population.
2. The top 5 counties with the highest rates of these health issues (i.e., diabetes, obesity and inactivity) are all in Georgia.
3. Top 10 health-affected counties, 60% were Republican-led, and 40% were Democrat-led.
4. The state of Wyoming shows that when checking the indicators diabetes, obesity, and inactivity, it has better metrics for all its counties, implying a healthier population in Wyoming.
5. Of the 354 counties in the dataset, 138 are from Texas, comprising nearly 40%, potentially introducing bias in both the data and analyses due to its single-state focus.

DISCUSSIONS

1. On proper analysis of the data provided by the CDC health department, we can clearly say that diabetes as a disease is a dependent variable of obesity and inactivity, and a person among the population suffering from these two should be prevalent to suffer from diabetes. This we were able to confirm through analysis of obesity and inactivity, where we tried to predict diabetes percentage for a county, and it was approximately around the provided or the actual value.

2. After analysing the data with all three variables (diabetes, obesity, and inactivity), we found that we had complete data for 38 states, while 12 states had missing data for one or two variables. Among these 38 states, Georgia had the highest combined rates of diabetes, obesity, and inactivity. In fact, the top five counties with the highest percentages of these health issues were all located in Georgia.

3. We analysed the top 10 states with the highest combined percentages of diabetes, obesity, and inactivity, considering the ruling parties. Among these counties, 60% were governed by the Republican party in 2018, while 40% were governed by the Democratic party. This raises questions about potential differences in food lifestyles or the cost of healthy food in these counties that may have contributed to these health metrics. To elaborate, there were 247 counties led by Republicans and 147 led by Democrat.

4. In our general data analysis, focusing on common data points where we had all three parameters (Diabetes, Obesity, and Inactivity), some notable trends emerged. Texas exhibited the highest average diabetic percentage, followed by Georgia, while Wyoming had the lowest. For obesity, the highest rate was observed in Washington, with Wyoming having the lowest. Regarding physical inactivity, Nebraska had the highest percentage, while Wyoming had the lowest. Based on these findings, it's reasonable to conclude that Wyoming boasts the most favourable health metrics among the states considered in our analysis.

5. The dataset comprises 354 counties, and 138 of them belong to Texas, making up nearly 40% of the entire dataset. This composition raises concerns about potential bias in both the dataset and any subsequent analyses because the data primarily comes from a single state.

APPENDIX A:

1. We began by downloading data from the CDC website in Excel format and then imported it into Jupyter notebook.
2. We conducted analyses for each sheet in the Excel file and created corresponding graphs.
3. We noticed data inconsistencies; the Diabetes sheet had 3,142 rows and columns, Obesity had 363 rows, and Inactivity had over 1,370 rows. To address this, we merged all the Excel sheets into one using the FIPS column as a common identifier.
4. With our new dataset, we calculated summary statistics such as mean, median, skewness, kurtosis, and standard deviation to gain insights.
5. To explore patterns among counties or states, we counted the occurrences of each state in the dataset.
6. We obtained information about the political party rule in each state from this link [1] and merged it with our dataset using FIPS values. This allowed us to determine which party ruled each county in 2018 and count the states predominantly led by Republicans or Democrats.
7. We also summed %diabetes, %inactivity, and %obesity to identify the counties with the highest combined health issues and determine the states to which these counties belonged.
8. Next, we delved into various modelling techniques using the new dataset. We began with Multiple Linear Regression, exploring all three columns: %diabetes, %inactivity, and %obesity. We considered different combinations of independent and dependent variables and identified the best model with %inactivity and %obesity as independent variables and %diabetes as the dependent variable.
9. To assess this model, we applied various evaluation techniques like cross-validation, collinearity analysis, intercept and coefficient examination, and confidence interval estimation. Additionally, we performed the Breusch-Pagan test to check for heteroskedasticity in this modelling approach.
10. We also conducted Random Forest Regression in a manner similar to Linear Regression and analysed the results.
11. Further we applied Ridge Regression, yielding findings consistent with our previous analyses.
12. To mention, we also worked with logistic regression and tried to get a method where, we should be able to predict using binary values and tell whether the obesity or inactivity will be more than or less than a threshold percentage. With this, we can go ahead and get an answer for the missing data in binary values.

APPENDIX B:

Out of a total of 3,142 data values across the three parameters (obesity, inactivity, and diabetes), we focused on 354 common data points (about 11% of the total) where each county provided percentages for all three parameters.

These 354 datasets were initially analyzed individually, starting with diabetes data, where we observed an unusually high mean value in Texas, followed by Georgia and Illinois, while Wyoming recorded the lowest diabetes rates in the entire United States. You can also visualize this information in the attached graph below.

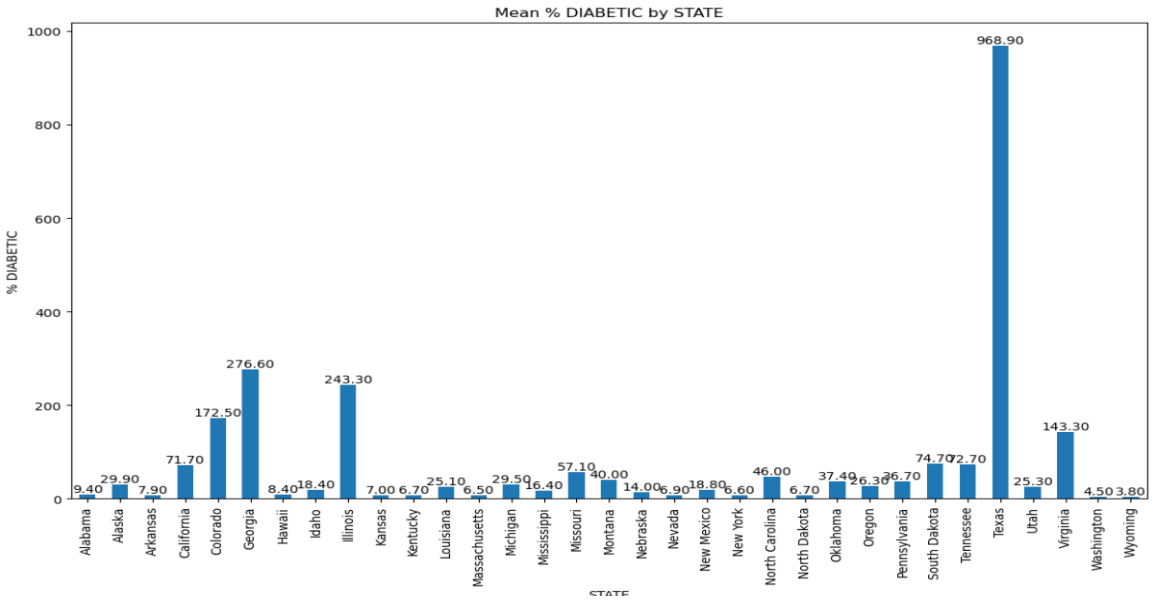

Fig: State wise distribution of Diabetes

When examining obesity rates individually, we found that Washington had the highest rate at 19.30, followed by Nebraska at 19.10, with Arkansas ranking third. In contrast, Wyoming had the lowest obesity rate. This information is also visualized in the attached graph below.

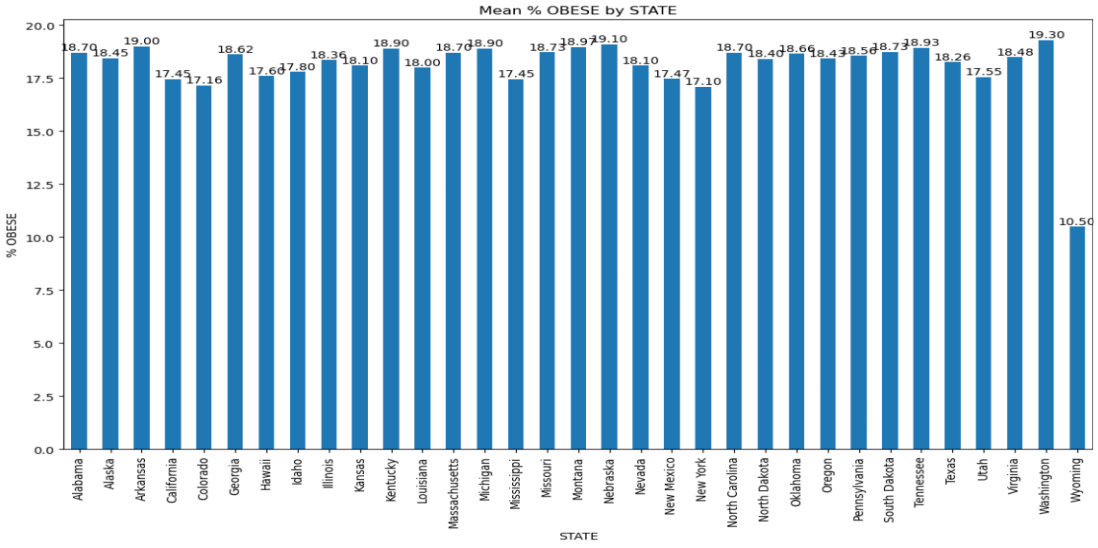

Fig: State wise distribution of Obesity

When examining the data specifically for inactivity, we found that Nebraska had the highest inactivity rate at 17.50, followed by two states at 17.20, namely Kentucky and New York, and North Dakota ranked third with 17.06% of the population being inactive. In contrast, Wyoming had the lowest percentage of inactive population, as depicted in the attached graph below.

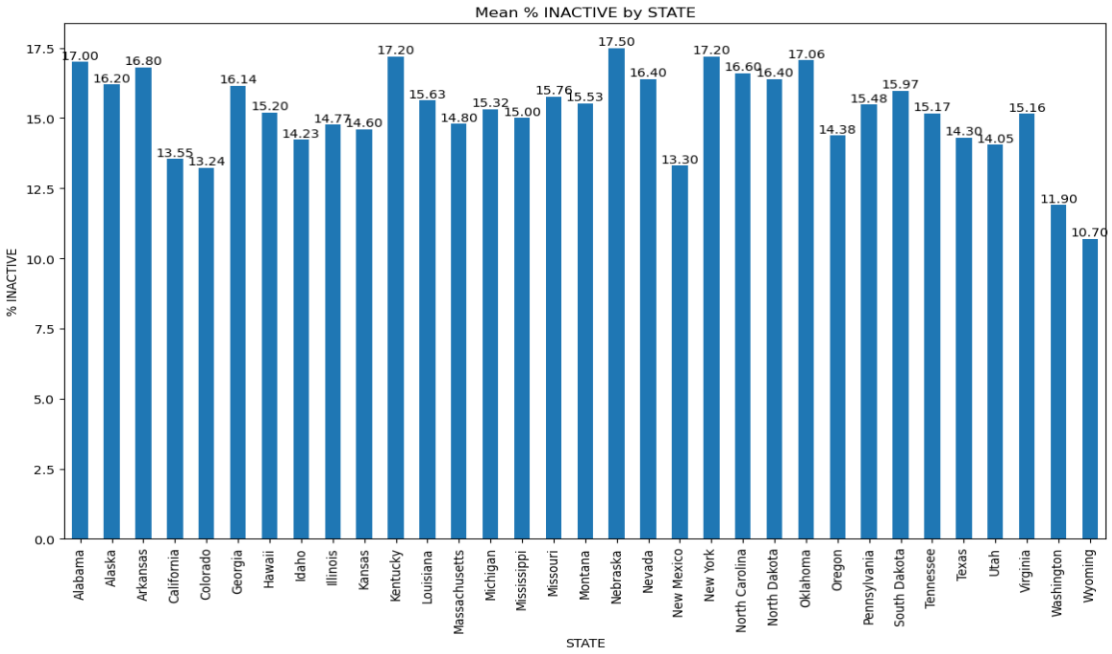

Fig: State wise distribution of Inactivity

Upon closer examination, we wanted to explore whether the political party in power had any bearing on our analysis. Primarily, we observed that either Republicans or Democrats held elected offices in the dataset. To investigate this aspect, we sourced political party data from a provided link (https://www.statista.com/statistics/1080003/political-party-identification-state-us/) and combined it with our analysis, summing up the parameters of diabetes, obesity, and inactivity.

In 2018, there were 207 counties under Republican governance and 147 led by Democrats, as shown in the graph. This also raised questions about the potential impact of food prices and state government policies on the health of county residents.

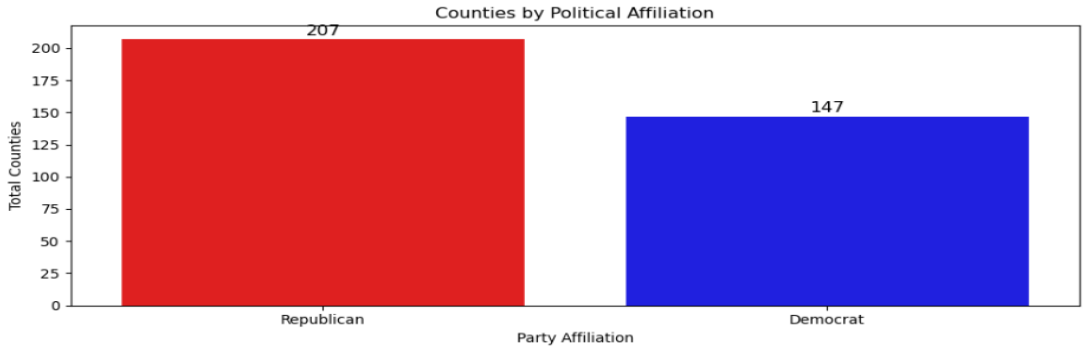

Fig: Political Party Wise Distribution

We also counted the respective states of each county in the dataset:

State	Count
Texas	138
Georgia	35
Illinois	34
Colorado	27
Virginia	19
California	11
South Dakc	10
Tennessee	10
Missouri	8
Montana	6
North Caro	6
Pennsylvar	5
Oklahoma	5
Alaska	4
Oregon	4
Michigan	4
Utah	4
New Mexic	3
Idaho	3
Louisiana	3
Mississippi	2
Nebraska	2
Washingto	1
Alabama	1
North Dakc	1
New York	1
Nevada	1
Massachus	1
Kentucky	1
Kansas	1
Hawaii	1
Arkansas	1
Wyoming	1

Fig: State wise distribution

When conducting analysis to identify the state with the highest overall health issue metrics, we found that Georgia had the highest metrics, indicating a high impact on its population. To reach this conclusion, we aggregated the metrics for the three parameters and sorted them, revealing that counties in Georgia had the highest affected population, as depicted in the results below:

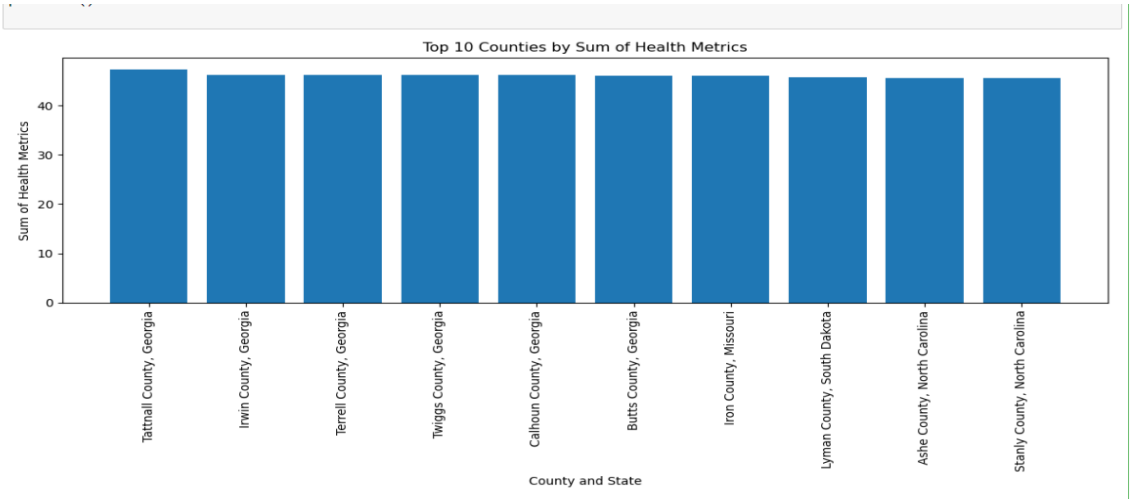

Fig: Metrics of Top 10 states

Moreover, we aimed to explore the relationships between the three variables, diabetes, obesity, and inactivity, by conducting various analyses. We employed multiple linear, random forest, and ridge regression techniques to investigate these connections. Our analyses yielded diverse outcomes depending on the chosen dependent variable—obesity, inactivity, or diabetes.

To achieve this, we followed a systematic approach in each regression model. Initially, we imported the data and defined the independent variables ('x') and the dependent variable ('y'). Subsequently, we split the data into training and testing datasets and trained the model using the training set. Following the model training, we evaluated its performance on the reserved testing data.

It's worth noting that we applied different regression techniques throughout the analysis. Remarkably, our findings were remarkably consistent when we treated diabetes as the dependent variable while investigating its relationship with obesity and inactivity as independent variables. We then visualized the best-fit model for the test data and computed metrics such as R2, mean square error, and absolute mean square error to assess prediction accuracy.

Furthermore, we conducted the Brusch-Pagan test on the multiple linear regression model to validate the accuracy of our analysis. This test helped ensure our findings' robustness and suitability for further interpretation.

The results of multiple linear regressions are attached below:

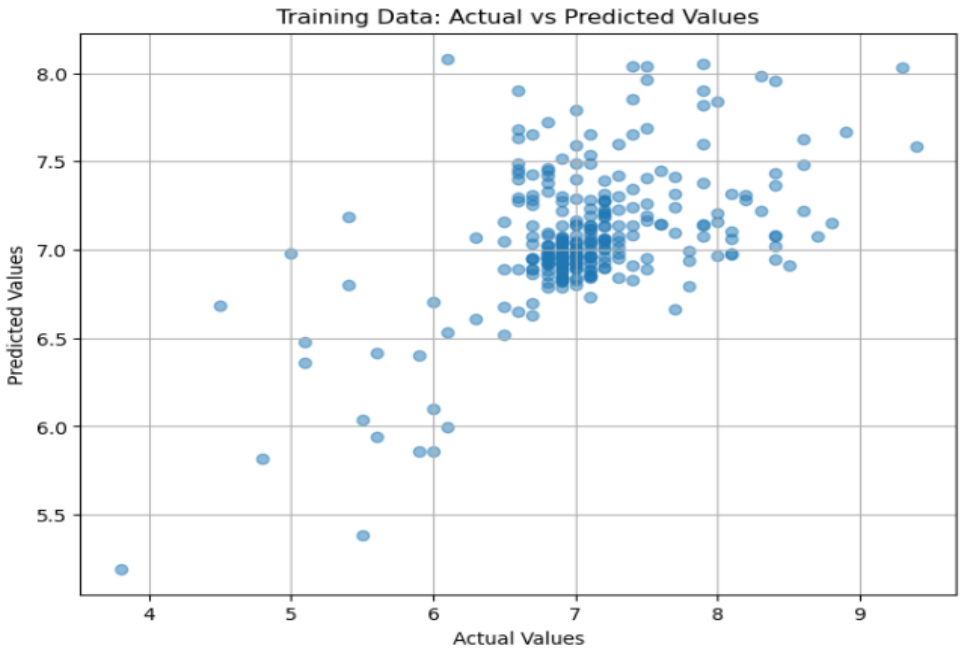

Fig: Plot of Multiple Linear Regression

Further, the results of the R^2, mean square error and additional values were as follows using the multiple linear regression model:

```
Cross-Validation MSE Scores: [0.6616154  0.49679872 0.54263043 0.6213607  0.65976824]
Mean Squared Error: 0.5964346993680371
Training MSE: 0.5831964900110125
Test MSE: 0.632505457320762
Mean R-squared (R²) or Accuracy Score: 0.36007498977268937
```

Fig: Results from Multiple Linear Regression

The results of random forest regression are attached below:

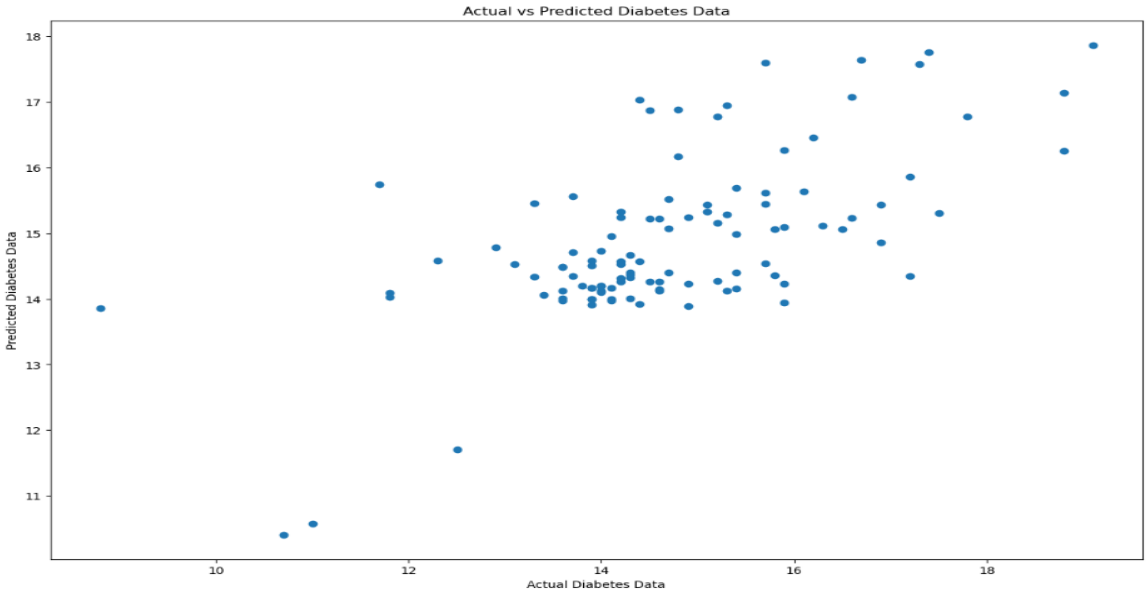

Fig: Plot of Random Forest Regression

Further, the results of the R^2, mean square error and additional values were as follows using the random forest regression model:

```
R2 Value 0.3855128769690096
Mean Squared Error 1.5941231152284219
Mean Absolute Error 0.9137392904343841
```

Fig:Results of Random Forest Regression

The results of ridge regression are attached below:

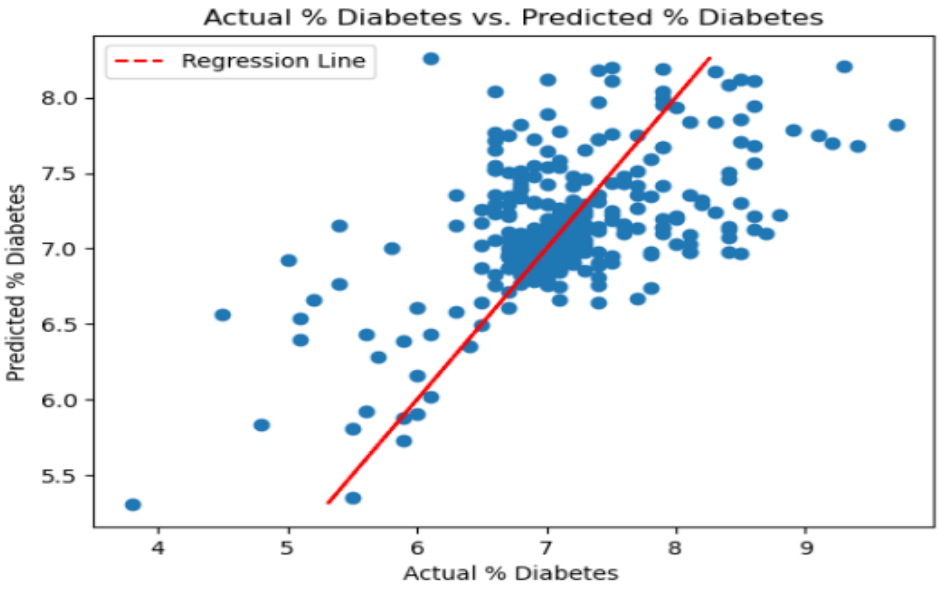

Fig: Plot of Ridge Regression

Further, the results of the R^2, mean square error and additional values were as follows using the ridge regression model:

```
VIF values:
       Variable         VIF
0         const  318.054491
1     % Obesity    1.287670
2  % Inactivity    1.287670
R-squared values for each fold: [-0.20292727 -0.53594962 -0.26711033 -0.34676252 -0.54544811]
Mean R-squared: -0.3796395709871226
Intercept: 1.6594416948999076
Coefficients: [0.11093297 0.2322351 ]
```

Fig: Result from Ridge Regression

After conducting the regression tests, we were able to predict diabetes percentages for specific states based on obesity and inactivity variables. Our prediction model consistently achieved an R-squared value ranging from 36% to 38%, indicating the reliability of our model compared to others. This suggests that diabetes is indeed influenced by obesity and inactivity. However, it's important to note that this analysis was limited to 38 states, as we had to exclude 89% of the counties due to missing data. Obtaining more county-level data would have been valuable. Additionally, incorporating factors like BMI, age range, pulse rates, and blood sugar could have enhanced the accuracy of our results, enabling a more comprehensive analysis.

We conducted a heatmap analysis to examine the pairwise correlations between parameters. The analysis revealed that counties with higher obesity rates also tended to have higher diabetic rates. Similarly, higher inactivity rates were associated with increased rates of diabetes and obesity. Additionally, obesity and inactivity exhibited a strong positive correlation. The results are visually presented below.

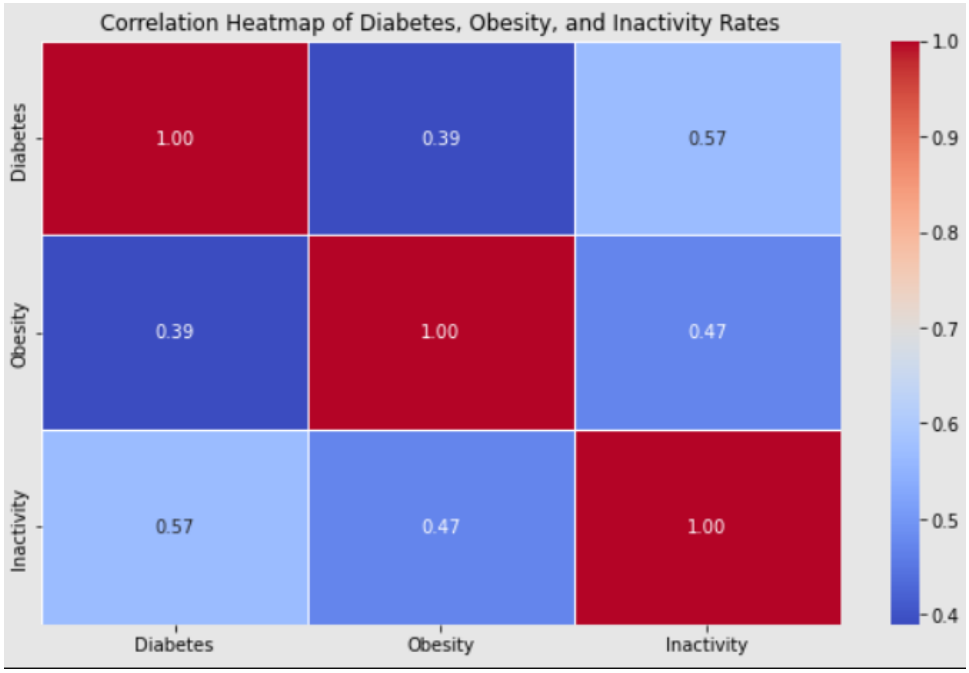

Fig: Heat Map Analysis of the data

We conducted an ANOVA analysis to validate our findings. For individual parameters, the p-values for diabetes, obesity, and inactivity ranged between 10^-16 to 10^-30, indicating statistical significance. However, the F-statistic values were notably high at 5.55, 6.89, and 9.90 for obesity, inactivity, and diabetes, respectively. These results suggest the need for state-specific interventions,

especially in states that deviate significantly from the national average, warranting larger data samples for accurate analysis.

Additionally, we attempted to create a logistic regression model to predict whether counties would have obesity rates above a certain threshold. Using binary responses derived from diabetes values with a threshold of 5.7, signifying an obesity percentage above 15%, we found that at least 70% of the county population in the dataset exceeded this threshold.

APPENDIX C:

We exclusively utilized the Python platform for all our coding tasks, primarily working within the Jupyter notebook or Anaconda Prompt environment. Our analysis heavily relied on several essential Python libraries, including pandas, numpy, matplotlib, seaborn, sklearn, statsmodels.api, and scipy.stats.

It's important to note that although our initial code for data importation, extraction, and distribution remained consistent, we employed distinct code segments for various purposes. Specifically, we used different code sections to conduct regression tests, analyze ANOVA, generate heat maps, perform geographical analysis, and examine political party-related data.

The codes in sequential workflows are as follows:

- To import the data set:

```
In [1]: import pandas as pd
        import numpy as np
        dataset="D:\cdcdata.xlsx"
        df=pd.read_excel(dataset)
        df.head()
```

Out[1]:

	Unnamed: 0	YEAR	FIPS	COUNTY	STATE	OBESE	DIABETIC	INACTIVE	PROB
0	0	2018	1011	Bullock County	Alabama	18.7	9.4	17.0	1
1	1	2018	2068	Denali Borough	Alaska	18.9	6.8	16.2	1
2	2	2018	2105	Hoonah-Angoon Census Area	Alaska	19.4	7.3	15.0	1
3	3	2018	2195	Petersburg Census Area	Alaska	17.2	9.2	17.8	1
4	4	2018	2230	Skagway Municipality	Alaska	18.3	6.6	15.8	1

Fig: Code to import dataset

- To extract the dataset:

```
In [ ]: import pandas as pd
        diabeties = pd.read_excel('Diabeties.xlsx')
        inactivity = pd.read_excel('Inactivity.xlsx')
        obesity = pd.read_excel('Obesity.xlsx')
        diabeties.head()
        diabeties.shape
        diabeties.nunique()
        inactivity.head()
        inactivity.shape
        inactivity.nunique()
        obesity.head()
        obesity.shape
        obesity.nunique()
        diabeties = diabeties.rename(columns={'STATEW': 'STATE'})
        inactivity = inactivity.rename(columns={'FIPDS':'FIPS'})
        diabeties.head()
        obesity_diabeties = pd.merge(obesity,diabeties,how='inner', on='FIPS')
        obesity_diabeties.head(10)
        obesity_diabeties.shape
        obesity_diabeties = obesity_diabeties.drop(['YEAR_y','COUNTY_y','STATE_y'], axis=1)
        obesity_diabeties_inactivity = pd.merge(obesity_diabeties,inactivity,how='inner', on='FIPS')
        obesity_diabeties_inactivity.head()
        obesity_diabeties_inactivity = obesity_diabeties_inactivity.drop(['YEAR','COUNTY','STATE'], axis=1)
        obesity_diabeties_inactivity = obesity_diabeties_inactivity.rename(columns={'YEAR_x': 'YEAR'})
        obesity_diabeties_inactivity = obesity_diabeties_inactivity.rename(columns={'COUNTY_x': 'COUNTY'})
        obesity_diabeties_inactivity = obesity_diabeties_inactivity.rename(columns={'STATE_x': 'STATE'})
        obesity_diabeties_inactivity.head()
        CDC = obesity_diabeties_inactivity.to_excel('CDC_FINAL_DATA.xlsx')
```

Fig: Code to extract common datasets

- To draw individual graphs:

```
import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame and 'STATE' and '% DIABETIC' are valid columns

# Group the data by 'STATE' and calculate the sum of '% DIABETIC'
grouped_data = df.groupby('STATE')['% DIABETIC'].sum()

# Set the figure size
plt.figure(figsize=(14, 8))

# Plot the grouped data as a bar plot
ax = grouped_data.plot(kind='bar')

# Add data labels to each bar
for i, v in enumerate(grouped_data):
    ax.text(i, v + 0.02, f'{v:.2f}', ha='center', va='bottom')

plt.xlabel('STATE')
plt.ylabel('% DIABETIC')
plt.title('Mean % DIABETIC by STATE')
plt.show()
```

Fig: Code for individual analysis

- For the purposes of calculating sum of % diabetes, % Inactivity and % obesity:

```
In [8]: import pandas as pd
        import matplotlib.pyplot as plt

        try:
            df = pd.read_excel(file_path)
        except FileNotFoundError:
            print(f"Error: The file '{file_path}' does not exist.")
            exit(1)
        except Exception as e:
            print(f"An error occurred while reading the Excel file: {e}")
            exit(1)

        df['Sum'] = df['% Diabetes'] + df['% Inactivity'] + df['% Obesity']

        sorted_df = df.sort_values(by='Sum', ascending=False)

        sorted_df['County'] = sorted_df['County'].astype(str)
        sorted_df['State'] = sorted_df['State'].astype(str)

        sorted_df['County_with_State'] = sorted_df['County'] + ', ' + sorted_df['State']

        # Select only the top 10 counties
        top_10_counties = sorted_df.head(10)

        # Visualize the top 10 counties as a bar chart
        plt.figure(figsize=(12, 6))
        plt.bar(top_10_counties['County_with_State'], top_10_counties['Sum'])
        plt.xlabel('County and State')
        plt.ylabel('Sum of Health Metrics')
        plt.title('Top 10 Counties by Sum of Health Metrics')
        plt.xticks(rotation=90)
        plt.tight_layout()
        plt.show()
```

Fig: Code to analyze percentage of Top 10 states

- To find the total count of states in the dataset :

```
In [ ]: state_column = df['State']

In [ ]: state_counts = state_column.value_counts()

In [ ]: state_counts_df = pd.DataFrame({'State': state_counts.index, 'Count': state_counts.values})

In [ ]: state_counts_df.to_excel('state_count.xlsx', index=False)
```

Fig: Code to work for state counts

- For the purpose of Political Party Analysis:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

file_path=(r'C:\Users\reach\Downloads\your_modified_file.xlsx')
try:
    df = pd.read_excel(file_path)
except FileNotFoundError:
    print(f"Error: The file '{file_path}' does not exist.")
    exit(1)
except Exception as e:
    print(f"An error occurred while reading the Excel file: {e}")
    exit(1)

republican_counties = df[df['Political Affiliation'] == 'Republican']
democrat_counties = df[df['Political Affiliation'] == 'Democrat']

total_republican = len(republican_counties)
total_democrat = len(democrat_counties)

plt.figure(figsize=(10, 4))
sns.barplot(x=['Republican', 'Democrat'], y=[total_republican, total_democrat], palette=['red', 'blue'])

plt.xlabel('Party Affiliation')
plt.ylabel('Total Counties')
plt.title('Counties by Political Affiliation')
plt.xticks(rotation=0)

for i, v in enumerate([total_republican, total_democrat]):
    plt.text(i, v + 1, str(v), ha='center', va='bottom', fontsize=12)

plt.tight_layout()

plt.savefig('county_affiliation_total.png', dpi=300)
plt.show()
```

Fig: Code for political party analysis

- For the purpose of Multiple Linear Regression,

```
In [12]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         from sklearn.linear_model import LinearRegression
         from sklearn.model_selection import cross_val_score
         from statsmodels.api import add_constant, OLS
         from statsmodels.stats.outliers_influence import variance_inflation_factor
         df_cleaned = df.dropna(subset=['% Inactivity'])
         [['% Inactivity', '% Obesity']]
         y = df_cleaned['% Diabetes']
         X_with_const = add_constant(X)
         vif = pd.DataFrame()
         vif["Variable"] = X_with_const.columns
         vif["VIF"] = [variance_inflation_factor(X_with_const.values, i) for i in range(X_with_const.shape[1])]
         print("VIF values:")
         print(vif)
         model = LinearRegression()
         k = 5  # You can adjust the number of folds (e.g., 5-fold cross-validation)
         cv_scores = cross_val_score(model, X, y, cv=k, scoring='r2')
         print("R-squared values for each fold:", cv_scores)
         print("Mean R-squared:", np.mean(cv_scores))
         model.fit(X, y)
         print("Intercept:", model.intercept_)
         print("Coefficients:", model.coef_)
         y_pred = model.predict(X)
         plt.scatter(y, y_pred)
         plt.xlabel("Actual % Diabetes Y")
         plt.ylabel("Predicted % Diabetes")
         plt.title("Actual % Diabetes vs. Predicted % Diabetes")
         regression_line = model.predict(X)
         plt.plot(y_pred, regression_line, color='red', linestyle='--', label='Regression Line')
         plt.legend()
         X = add_constant(X)
         model = OLS(y, X).fit()
         confidence_intervals = model.conf_int()
         p_values = model.summary()
         print("Confidence Intervals for coefficients:")
         print(confidence_intervals)
         print(p_values)
```

Fig: Code for Multiple Linear Regression

- For the purpose of Brusch Pagan test,

```
import pandas as pd
df = pd.read_excel('/Users/jisusingh/Downloads/MTH_522/merged_dataset.xlsx')

import statsmodels.api as sm
# Independent Variables
X = df[['% INACTIVE', '% OBESE']]
# Dependent Variables
y = df['% DIABETIC']

# Add a constant (intercept) to the independent variables
X = sm.add_constant(X)

# Fit a linear regression model
model = sm.OLS(y, X).fit()

# Getting the residuals
residuals = model.resid

import matplotlib.pyplot as plt
plt.scatter(model.fittedvalues, residuals)
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.axhline(y=0, color='r', linestyle='--')
plt.show()

from statsmodels.stats.diagnostic import het_breuschpagan

# Performing the Breusch-Pagan test
bp_test = het_breuschpagan(residuals, X)

# Extracting the test statistics and p-values
test_statistic = bp_test[0]
p_value = bp_test[1]

print(f"Breusch-Pagan Test Statistic: {test_statistic}")
print(f"P-value: {p_value}")

# Checking for the Heteroskedasticity
significance_level = 0.05
if p_value < significance_level:
    print("Heteroskedasticity is detected.")
else:
    print("No significant evidence of heteroskedasticity.")
```

Fig: Code for Brusch Pagan Test

- For the purpose of Random Forest Regression,

```
import pandas as pd
import numpy as np
dataset="D:\cdcdata.xlsx"
df=pd.read_excel(dataset)
df.head()
df=df.drop(['Unnamed: 0'],axis=1)
x=df.drop(['YEAR','FIPS','COUNTY','STATE','PROB','INACTIVE'],axis=1).values
y=df['INACTIVE'].values
print(x)
print(y)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=25)
from sklearn.ensemble import RandomForestRegressor
rf_regressor = RandomForestRegressor()
rf_regressor.fit(x_train, y_train)
rf_pred = rf_regressor.predict(x_test)
from sklearn.metrics import r2_score
final=r2_score(y_test,y_pred)
print(final)
import matplotlib.pyplot as plt
plt.figure(figsize=(15,10))
plt.scatter(y_test,y_pred)
plt.xlabel('Actual Diabetes Data')
plt.ylabel('Predicted Diabetes Data')
plt.title('Actual vs Predicted Diabetes Data')
pred_y_df=pd.DataFrame({'Actual Diabetes Value':y_test,'Predicted Diabetes Value':y_pred,'Difference':y_test-y_pred})
pred_y_df[0:50]
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
r2_value=r2_score(y_test,y_pred)
mean_squared_error_value=mean_squared_error(y_test,y_pred)
mean_absolute_error_value=mean_absolute_error(y_test,y_pred)
print("R2 Value",r2_value)
print("Mean Squared Error",mean_squared_error_value)
print("Mean Absolute Error",mean_absolute_error_value)
```

Fig: Code for Random Forest Regression

- To do Ridge Regression:

In [13]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import Ridge
from sklearn.model_selection import cross_val_score
from statsmodels.api import add_constant
from statsmodels.stats.outliers_influence import variance_inflation_factor
file_path=(r'C:\Users\reach\Downloads\your_modified_file.xlsx')

df = pd.read_excel(file_path)

df_cleaned = df.dropna(subset=['% Inactivity'])


X = df_cleaned[['% Obesity','% Inactivity' ]]
y = df_cleaned['% Diabetes']

X_with_const = add_constant(X)
vif = pd.DataFrame()
vif["Variable"] = X_with_const.columns
vif["VIF"] = [variance_inflation_factor(X_with_const.values, i) for i in range(X_with_const.shape[1])]
print("VIF values:")
print(vif)

ridge_model = Ridge(alpha=1.0)  # You can adjust the alpha parameter for regularization strength

k = 5
cv_scores = cross_val_score(ridge_model, X, y, cv=k, scoring='r2')

print("R-squared values for each fold:", cv_scores)
print("Mean R-squared:", np.mean(cv_scores))

ridge_model.fit(X, y)

print("Intercept:", ridge_model.intercept_)
print("Coefficients:", ridge_model.coef_)

# Make predictions using the Ridge model
y_pred = ridge_model.predict(X)

plt.scatter(y, y_pred)
plt.xlabel("Actual % Diabetes ")
plt.ylabel("Predicted % Diabetes")
plt.title("Actual % Diabetes vs. Predicted % Diabetes")

regression_line = ridge_model.predict(X)

plt.plot(y_pred, regression_line, color='red', linestyle='--', label='Regression Line')

plt.legend()

plt.show()
```

Fig: Code for Ridge Regression

- To generate the heat map:

In [13]:
```
# Importing Required Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.formula.api import ols
```

In [5]:
```
# Load the Excel Data into a DataFrame
file_path = 'your_modified_file.xlsx'  # Replace with file path
df = pd.read_excel(file_path)
```

In [6]:
```
# Data Cleaning - Remove Rows with Missing Values
df_clean = df.dropna().copy()

# Rename Columns for Easier Analysis
df_clean.rename(columns={'% Diabetes': 'Diabetes', '% Obesity': 'Obesity', '% Inactivity': 'Inactivity'}, inplace=True)
```

In [7]:
```
# Generate Correlation Heatmap
plt.figure(figsize=(10, 6))
corr = df_clean[['Diabetes', 'Obesity', 'Inactivity']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f', linewidths=.5)
plt.title('Correlation Heatmap of Diabetes, Obesity, and Inactivity Rates')
plt.show()
```

Fig: Code for Heat Map Analysis

- To generate the ANNOVA test,

```
# Conduct ANOVA Analysis
## ANOVA for Diabetes across different states
model_diabetes = ols('Diabetes ~ State', data=df_clean).fit()
anova_table_diabetes = sm.stats.anova_lm(model_diabetes, typ=2)
print("ANOVA Table for Diabetes:")
print(anova_table_diabetes)
```

Fig: Code for ANNOVA test

- For the purpose of Logistic Regression Testing:

```
#Define the independent and dependent variables
y= df['OBESE'] #dependent variable is Decision
x= df.drop(['OBESE'], axis=1)
# splitting the data
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size= 0.2)
```

```
#Implementing Logistic Regression using sklearn
mlog=LogisticRegression(x_train)
nlog=LogisticRegression(y_train)
from sklearn import tree
import statsmodels.formula.api as smfi
import matplotlib.pyplot as plt
model=(mlog,nlog)
print(model)
```

Fig: Code for Logistic Regression

REFERENCES:

[1]: (https://www.statista.com/statistics/1080003/political-party-identification-state-us/)

[2]: (https://gis.cdc.gov/grasp/diabetes/diabetesatlas-sdoh.html)