

# CDC Heart Disease Prediction

October 22, 2024

```
[1]: # Background: A Kaggle provided, from the CDC and is a major part of the
      ↳ Behavioral Risk Factor Surveillance System
      # (BRFSS), which conducts annual telephone surveys to collect data on the
      ↳ health status of U.S. residents.
      # https://www.kaggle.com/datasets/kamilpytlak/
      ↳ personal-key-indicators-of-heart-disease
```

```
[2]: # The goal of the notebook is to construct a logistic regression model to
      ↳ predict heart attacks
      # (the dependent variable - 'HadHeartAttack') based on the other variables as
      ↳ independet predictor variables.
```

```
[3]: # Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
```

```
[4]: # setup for multiple outputs from single cell
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'
```

```
[5]: # silence warnings
import warnings
warnings.filterwarnings('ignore')
import absl.logging as absl_logging
```

```
[6]: #####
      # SET AND VERIFY THE CURRENT WORKING DIRECTORY (CHANGE AS
      ↳ NEEDED)#####
      #####
      import os
```

```

root_directory = '/media/ijmg/SSD_FOUR_TB/IJMG_DATA_SCIENTIST/data sets/CDC_CHD/'
↪
os.chdir(root_directory)
print(os.getcwd() + '/')
print(root_directory)

```

```

/media/ijmg/SSD_FOUR_TB/IJMG_DATA_SCIENTIST/data sets/CDC_CHD/
/media/ijmg/SSD_FOUR_TB/IJMG_DATA_SCIENTIST/data sets/CDC_CHD/

```

```

[7]: #####
##      I. EXPLORATORY DATA ANALYSIS (EDA):
#####
# Data Visualization: As allowed by data, show:
# -- summary statistics
# -- histograms
# -- box plots
# -- outliers
# -- scatter plots
# -- correlation matrices

```

```

[8]: # Load the dataset from CSV file into a DataFrame and preview
source_df = pd.read_csv('heart_2022_with_nans.csv')
# Display the DataFrame and print dimensions
print("Number of rows:", source_df.shape[0])
print("Number of columns:", source_df.shape[1])
source_df.head()
source_df.tail()

```

Number of rows: 445132

Number of columns: 40

```

[8]:      State      Sex GeneralHealth PhysicalHealthDays MentalHealthDays \
0  Alabama  Female      Very good              0.0              0.0
1  Alabama  Female      Excellent              0.0              0.0
2  Alabama  Female      Very good              2.0              3.0
3  Alabama  Female      Excellent              0.0              0.0
4  Alabama  Female          Fair              2.0              0.0

                                LastCheckupTime PhysicalActivities \
0  Within past year (anytime less than 12 months ...              No
1                                NaN                      No
2  Within past year (anytime less than 12 months ...              Yes
3  Within past year (anytime less than 12 months ...              Yes
4  Within past year (anytime less than 12 months ...              Yes

      SleepHours RemovedTeeth HadHeartAttack ... HeightInMeters \
0           8.0          NaN          No ...          NaN
1           6.0          NaN          No ...          1.60

```

2	5.0	NaN	No ...	1.57
3	7.0	NaN	No ...	1.65
4	9.0	NaN	No ...	1.57

	WeightInKilograms	BMI	AlcoholDrinkers	HIVTesting	FluVaxLast12	\
0	NaN	NaN	No	No	Yes	
1	68.04	26.57	No	No	No	
2	63.50	25.61	No	No	No	
3	63.50	23.30	No	No	Yes	
4	53.98	21.77	Yes	No	No	

	PneumoVaxEver	TetanusLast10Tdap	\
0	No	Yes, received tetanus shot but not sure what type	
1	No	No, did not receive any tetanus shot in the pa...	
2	No	NaN	
3	Yes	No, did not receive any tetanus shot in the pa...	
4	Yes	No, did not receive any tetanus shot in the pa...	

	HighRiskLastYear	CovidPos
0	No	No
1	No	No
2	No	Yes
3	No	No
4	No	No

[5 rows x 40 columns]

[8]:

	State	Sex	GeneralHealth	PhysicalHealthDays	\
445127	Virgin Islands	Female	Good	0.0	
445128	Virgin Islands	Female	Excellent	2.0	
445129	Virgin Islands	Female	Poor	30.0	
445130	Virgin Islands	Male	Very good	0.0	
445131	Virgin Islands	Male	Very good	0.0	

	MentalHealthDays	LastCheckupTime	\
445127	3.0	Within past 2 years (1 year but less than 2 ye...	
445128	2.0	Within past year (anytime less than 12 months ...	
445129	30.0	5 or more years ago	
445130	0.0	Within past year (anytime less than 12 months ...	
445131	1.0	NaN	

	PhysicalActivities	SleepHours	RemovedTeeth	HadHeartAttack	...	\
445127	Yes	6.0	None of them	No	...	
445128	Yes	7.0	None of them	No	...	
445129	No	5.0	1 to 5	No	...	
445130	No	5.0	None of them	Yes	...	
445131	Yes	5.0	None of them	No	...	

	HeightInMeters	WeightInKilograms	BMI	AlcoholDrinkers	HIVTesting	\
445127	1.65	69.85	25.63	NaN	Yes	
445128	1.70	83.01	28.66	No	Yes	
445129	1.70	49.90	17.23	NaN	No	
445130	1.83	108.86	32.55	No	Yes	
445131	1.68	63.50	22.60	Yes	No	

	FluVaxLast12	PneumoVaxEver	\
445127	No	No	
445128	Yes	No	
445129	No	No	
445130	Yes	Yes	
445131	No	No	

	TetanusLast10Tdap	HighRiskLastYear	\
445127	No, did not receive any tetanus shot in the pa...	No	
445128	Yes, received tetanus shot but not sure what type	No	
445129	No, did not receive any tetanus shot in the pa...	No	
445130	No, did not receive any tetanus shot in the pa...	No	
445131	Yes, received tetanus shot but not sure what type	No	

	CovidPos
445127	Yes
445128	No
445129	No
445130	Yes
445131	No

[5 rows x 40 columns]

```
[9]: # Before any data visualizations, count then remove any missing values or NaNs
```

```
[10]: # Count missing (NaN) values for each column/variable
missing_values_df = source_df.isna().sum() # or df.isnull().sum()
print("Missing Values in Each Variable:\n")
print(missing_values_df)
print("\nMissing Values in Entire Dataframe:\n")
print(missing_values_df.sum())
```

Missing Values in Each Variable:

State	0
Sex	0
GeneralHealth	1198
PhysicalHealthDays	10927
MentalHealthDays	9067
LastCheckupTime	8308

PhysicalActivities	1093
SleepHours	5453
RemovedTeeth	11360
HadHeartAttack	3065
HadAngina	4405
HadStroke	1557
HadAsthma	1773
HadSkinCancer	3143
HadCOPD	2219
HadDepressiveDisorder	2812
HadKidneyDisease	1926
HadArthritis	2633
HadDiabetes	1087
DeafOrHardOfHearing	20647
BlindOrVisionDifficulty	21564
DifficultyConcentrating	24240
DifficultyWalking	24012
DifficultyDressingBathing	23915
DifficultyErrands	25656
SmokerStatus	35462
ECigaretteUsage	35660
ChestScan	56046
RaceEthnicityCategory	14057
AgeCategory	9079
HeightInMeters	28652
WeightInKilograms	42078
BMI	48806
AlcoholDrinkers	46574
HIVTesting	66127
FluVaxLast12	47121
PneumoVaxEver	77040
TetanusLast10Tdap	82516
HighRiskLastYear	50623
CovidPos	50764
dtype: int64	

Missing Values in Entire Dataframe:

902665

```
[11]: # Remove missing (NaN) values for each column/variable
source_df = source_df.dropna()
```

```
[12]: # Verify removal of missing (NaN) values with counts for each column/variable
missing_values_df = source_df.isna().sum() # or df.isnull().sum()
print("Missing Values in Each Variable:\n")
print(missing_values_df)
```

```
print("\nMissing Values in Entire Dataframe:\n")
print(missing_values_df.sum())
```

Missing Values in Each Variable:

State	0
Sex	0
GeneralHealth	0
PhysicalHealthDays	0
MentalHealthDays	0
LastCheckupTime	0
PhysicalActivities	0
SleepHours	0
RemovedTeeth	0
HadHeartAttack	0
HadAngina	0
HadStroke	0
HadAsthma	0
HadSkinCancer	0
HadCOPD	0
HadDepressiveDisorder	0
HadKidneyDisease	0
HadArthritis	0
HadDiabetes	0
DeafOrHardOfHearing	0
BlindOrVisionDifficulty	0
DifficultyConcentrating	0
DifficultyWalking	0
DifficultyDressingBathing	0
DifficultyErrands	0
SmokerStatus	0
ECigaretteUsage	0
ChestScan	0
RaceEthnicityCategory	0
AgeCategory	0
HeightInMeters	0
WeightInKilograms	0
BMI	0
AlcoholDrinkers	0
HIVTesting	0
FluVaxLast12	0
PneumoVaxEver	0
TetanusLast10Tdap	0
HighRiskLastYear	0
CovidPos	0

dtype: int64

Missing Values in Entire Dataframe:

0

```
[13]: # Display the DataFrame and print dimensions
print("Number of rows:", source_df.shape[0])
print("Number of columns:", source_df.shape[1])
source_df.head()
source_df.tail()
```

Number of rows: 246022

Number of columns: 40

```
[13]:      State      Sex GeneralHealth PhysicalHealthDays MentalHealthDays \
342 Alabama Female      Very good              4.0              0.0
343 Alabama Male      Very good              0.0              0.0
345 Alabama Male      Very good              0.0              0.0
346 Alabama Female      Fair              5.0              0.0
347 Alabama Female      Good              3.0             15.0

                                LastCheckupTime PhysicalActivities \
342 Within past year (anytime less than 12 months ...          Yes
343 Within past year (anytime less than 12 months ...          Yes
345 Within past year (anytime less than 12 months ...           No
346 Within past year (anytime less than 12 months ...          Yes
347 Within past year (anytime less than 12 months ...          Yes

      SleepHours      RemovedTeeth HadHeartAttack ... HeightInMeters \
342          9.0      None of them              No ...          1.60
343          6.0      None of them              No ...          1.78
345          8.0 6 or more, but not all              No ...          1.85
346          9.0      None of them              No ...          1.70
347          5.0          1 to 5              No ...          1.55

      WeightInKilograms      BMI AlcoholDrinkers HIVTesting FluVaxLast12 \
342          71.67  27.99              No              No          Yes
343          95.25  30.13              No              No          Yes
345         108.86  31.66              Yes              No          No
346          90.72  31.32              No              No          Yes
347          79.38  33.07              No              No          Yes

      PneumoVaxEver      TetanusLast10Tdap \
342          Yes          Yes, received Tdap
343          Yes Yes, received tetanus shot but not sure what type
345          Yes No, did not receive any tetanus shot in the pa...
346          Yes No, did not receive any tetanus shot in the pa...
347          Yes No, did not receive any tetanus shot in the pa...

      HighRiskLastYear CovidPos
```

342	No	No
343	No	No
345	No	Yes
346	No	Yes
347	No	No

[5 rows x 40 columns]

```
[13]:
```

	State	Sex	GeneralHealth	PhysicalHealthDays	\
445117	Virgin Islands	Male	Very good	0.0	
445123	Virgin Islands	Female	Fair	0.0	
445124	Virgin Islands	Male	Good	0.0	
445128	Virgin Islands	Female	Excellent	2.0	
445130	Virgin Islands	Male	Very good	0.0	

	MentalHealthDays	LastCheckupTime	\
445117	0.0	Within past 2 years (1 year but less than 2 ye...	
445123	7.0	Within past year (anytime less than 12 months ...	
445124	15.0	Within past year (anytime less than 12 months ...	
445128	2.0	Within past year (anytime less than 12 months ...	
445130	0.0	Within past year (anytime less than 12 months ...	

	PhysicalActivities	SleepHours	RemovedTeeth	HadHeartAttack	...	\
445117	Yes	6.0	None of them	No	...	
445123	Yes	7.0	None of them	No	...	
445124	Yes	7.0	1 to 5	No	...	
445128	Yes	7.0	None of them	No	...	
445130	No	5.0	None of them	Yes	...	

	HeightInMeters	WeightInKilograms	BMI	AlcoholDrinkers	HIVTesting	\
445117	1.78	102.06	32.28	Yes	No	
445123	1.93	90.72	24.34	No	No	
445124	1.68	83.91	29.86	Yes	Yes	
445128	1.70	83.01	28.66	No	Yes	
445130	1.83	108.86	32.55	No	Yes	

	FluVaxLast12	PneumoVaxEver	\
445117	No	No	
445123	No	No	
445124	Yes	Yes	
445128	Yes	No	
445130	Yes	Yes	

	TetanusLast10Tdap	HighRiskLastYear	\
445117	Yes, received tetanus shot but not sure what type	No	
445123	No, did not receive any tetanus shot in the pa...	No	
445124	Yes, received tetanus shot but not sure what type	No	



445128	Yes, received tetanus shot but not sure what type	No
445130	No, did not receive any tetanus shot in the pa...	No

	CovidPos
445117	No
445123	Yes
445124	Yes
445128	No
445130	Yes

[5 rows x 40 columns]

```
[14]: # COMMENT: At this point, the removal of rows holding 902665 missing values
#      resulted in a decrease in the datarame size
#      from
#      Number of rows: 445132
#      Number of columns: 40
#      to
#      Number of rows: 246022
#      Number of columns: 40
```

```
[15]: # Begin Exploratory Data Analysis and Visualizations
```

```
[16]: # Visualize unique categories for each column/variable
for column in source_df.columns:
    unique_categories = source_df[column].unique()
    print(f"Unique categories for column '{column}':")
    print(unique_categories)
    print()
```

Unique categories for column 'State':

```
['Alabama' 'Alaska' 'Arizona' 'Arkansas' 'California' 'Colorado'
 'Connecticut' 'Delaware' 'District of Columbia' 'Florida' 'Georgia'
 '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' 'Rhode Island' 'South Carolina'
 'South Dakota' 'Tennessee' 'Texas' 'Utah' 'Vermont' 'Virginia'
 'Washington' 'West Virginia' 'Wisconsin' 'Wyoming' 'Guam' 'Puerto Rico'
 'Virgin Islands']
```

Unique categories for column 'Sex':

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

Unique categories for column 'GeneralHealth':

```
['Very good' 'Fair' 'Good' 'Excellent' 'Poor']
```

Unique categories for column 'PhysicalHealthDays':  
[ 4. 0. 5. 3. 2. 25. 30. 15. 29. 8. 16. 20. 10. 9. 7. 1. 21. 6.  
27. 14. 12. 11. 13. 28. 17. 23. 24. 26. 18. 22. 19.]

Unique categories for column 'MentalHealthDays':  
[ 0. 15. 4. 25. 5. 30. 27. 3. 2. 1. 10. 20. 21. 6. 7. 8. 14. 9.  
12. 18. 29. 28. 17. 11. 16. 13. 26. 22. 24. 19. 23.]

Unique categories for column 'LastCheckupTime':  
['Within past year (anytime less than 12 months ago)'  
'5 or more years ago'  
'Within past 2 years (1 year but less than 2 years ago)'  
'Within past 5 years (2 years but less than 5 years ago)']

Unique categories for column 'PhysicalActivities':  
['Yes' 'No']

Unique categories for column 'SleepHours':  
[ 9. 6. 8. 5. 7. 10. 4. 12. 3. 18. 11. 2. 1. 16. 14. 15. 13. 20.  
24. 23. 19. 17. 22.]

Unique categories for column 'RemovedTeeth':  
['None of them' '6 or more, but not all' '1 to 5' 'All']

Unique categories for column 'HadHeartAttack':  
['No' 'Yes']

Unique categories for column 'HadAngina':  
['No' 'Yes']

Unique categories for column 'HadStroke':  
['No' 'Yes']

Unique categories for column 'HadAsthma':  
['No' 'Yes']

Unique categories for column 'HadSkinCancer':  
['No' 'Yes']

Unique categories for column 'HadCOPD':  
['No' 'Yes']

Unique categories for column 'HadDepressiveDisorder':  
['No' 'Yes']

Unique categories for column 'HadKidneyDisease':  
['No' 'Yes']

Unique categories for column 'HadArthritis':

['Yes' 'No']

Unique categories for column 'HadDiabetes':

['No' 'Yes' 'Yes, but only during pregnancy (female)'  
'No, pre-diabetes or borderline diabetes']

Unique categories for column 'DeafOrHardOfHearing':

['No' 'Yes']

Unique categories for column 'BlindOrVisionDifficulty':

['No' 'Yes']

Unique categories for column 'DifficultyConcentrating':

['No' 'Yes']

Unique categories for column 'DifficultyWalking':

['No' 'Yes']

Unique categories for column 'DifficultyDressingBathing':

['No' 'Yes']

Unique categories for column 'DifficultyErrands':

['No' 'Yes']

Unique categories for column 'SmokerStatus':

['Former smoker' 'Never smoked' 'Current smoker - now smokes every day'  
'Current smoker - now smokes some days']

Unique categories for column 'ECigaretteUsage':

['Never used e-cigarettes in my entire life' 'Use them some days'  
'Not at all (right now)' 'Use them every day']

Unique categories for column 'ChestScan':

['No' 'Yes']

Unique categories for column 'RaceEthnicityCategory':

['White only, Non-Hispanic' 'Black only, Non-Hispanic'  
'Other race only, Non-Hispanic' 'Multiracial, Non-Hispanic' 'Hispanic']

Unique categories for column 'AgeCategory':

['Age 65 to 69' 'Age 70 to 74' 'Age 75 to 79' 'Age 80 or older'  
'Age 50 to 54' 'Age 40 to 44' 'Age 60 to 64' 'Age 55 to 59'  
'Age 45 to 49' 'Age 35 to 39' 'Age 25 to 29' 'Age 30 to 34'  
'Age 18 to 24']

Unique categories for column 'HeightInMeters':

[1.6 1.78 1.85 1.7 1.55 1.63 1.75 1.68 1.83 1.52 1.88 1.5 1.73 1.65

1.8 1.57 1.91 1.47 1.42 1.22 1.93 2.01 1.96 1.98 1.45 1.35 1.76 2.03  
2.16 1.51 1.53 1.69 1.56 1.84 1.9 1.54 1.72 1.87 1.74 1.4 1.64 1.58  
1.62 1.79 1.67 1.46 1.89 1.61 1.3 1.37 2.13 2.06 2.11 0.91 2.26 2.18  
1.77 2.36 1.59 1.86 1.82 1.66 1.71 1.95 1.05 2.08 1.49 1.38 1.81 1.44  
1.48 1.19 1.32 1.24 1.07 1.04 1.27 1.1 1.92 1.2 2.24 1.12 1.03 0.97  
1.25 2.29 1.16 1.18 1.09 2.41 1. 1.17 1.08 1.43 1.14 1.02 2. 2.02  
0.95 2.34 2.21]

Unique categories for column 'WeightInKilograms':

[ 71.67 95.25 108.86 90.72 79.38 120.2 88. 74.84 78.02 63.5  
122.47 115.67 81.65 86.18 76.2 54.88 72.57 88.45 104.33 52.16  
68.04 65.77 56.7 94.8 123.83 50.8 68.95 113.4 83.91 77.56  
68.49 82.1 80.74 106.14 58.06 61.69 57.61 84.82 70.76 70.31  
91.63 102.06 48.08 61.23 109.77 99.79 58.97 110.68 64.86 111.13  
45.36 79.83 98.88 55.34 101.6 77.11 93.89 71.21 49.9 96.16  
163.29 120.66 97.52 88.9 44.91 85.73 83.46 92.99 132. 67.59  
92.08 73.48 107.5 107.95 91.17 74.39 64.41 62.6 46.72 103.42  
87.09 89.81 83.01 100.7 56.25 96.62 66.68 67.13 69.4 58.51  
78.93 95.71 63.05 49.44 127.01 145.15 122.02 107.05 126.55 117.03  
47.17 181.44 65.32 117.93 136.08 78.47 52.62 121.56 73.94 82.55  
106.59 59.87 110.22 62.14 51.71 93.44 54.43 85.28 59.42 66.22  
76.66 55.79 75.3 97.07 87.54 69.85 124.74 63.96 47.63 94.35  
97.98 89.36 92.53 101.15 149.69 129.27 84.37 195.04 99.34 114.31  
53.07 81.19 75.75 124.28 112.94 80.29 114.76 45.81 53.52 133.81  
51.26 158.76 60.78 46.27 72.12 131.54 127.91 53.98 98.43 130.63  
143.34 102.51 115.21 90.26 166.92 109.32 40.37 135.62 204.12 129.73  
127.46 138.35 105.69 119.75 48.53 140.61 105.23 139.25 126.1 135.17  
102.97 122.92 57.15 38.56 60.33 131.09 148.78 116.57 112.49 86.64  
112.04 172.37 133.36 118.84 50.35 103.87 111.58 121.11 113.85 73.03  
142.88 134.26 123.38 37.19 119.29 36.29 48.99 43.09 41.73 35.38  
104.78 144.24 167.83 149.23 37.65 86. 147.42 165.56 154.22 136.98  
108.41 155.58 206.38 148.32 42.18 44.45 90. 191.87 249.48 67.  
44. 40.82 156.49 53. 139.71 130.18 118.39 100. 151.95 165.11  
43.54 134.72 141.52 125.19 75. 250. 116.12 73. 100.24 74.  
200. 80. 82. 54. 66. 152.41 39.46 41.28 190.51 188.24  
59. 70. 170.1 46. 265. 168.74 190. 55. 93. 159.66  
78. 38.1 185.07 104. 183.7 125.65 68. 134. 130. 32.21  
143.79 137.89 179.17 105. 65. 32. 292.57 85. 72. 174.63  
50. 128.37 62. 87. 176.9 39.92 76. 128.82 58. 156.04  
121. 42.64 89. 146.96 146.06 171.46 227.25 29.48 190.06 161.03  
226.8 132.45 137.44 64. 56. 141.07 52. 63. 120. 83.  
57. 31.75 77. 96. 60. 115. 41. 150.59 272.16 48.  
39.01 95. 197.31 158.3 45. 94. 240.4 49. 157.85 108.  
185. 61. 34.02 132.9 84. 229.97 138.8 81. 79. 92.  
107. 155.13 208.65 69. 111. 110. 151.05 210. 140.16 35.83  
146.51 117.48 102. 125. 151.5 36.74 38. 135. 71. 147.87  
153.77 170. 91. 98. 192.32 186.88 118. 160.12 160. 170.55  
201.85 184.16 175.09 142.43 169. 166.01 180.53 196.41 162.39 40.]

```

171.91 195.95 136.53 153.31 159.21 164.2 219.99 141.97 173.27 34.47
213.19 276.24 199.58 215.46 217.72 175.99 200.03 230.88 33.57 185.52
103. 152.86 101. 160.57 150.14 157.4 145. 150. 163.75 191.42
174.18 164.65 256.28 205.48 192.78 161.48 178.26 179.62 144.7 205.02
178.72 154.68 166.47 177.81 200.49 231.79 238.14 227.7 273.52 211.83
223.62 197.77 189.15 185.97 250.38 183.25 181.89 222.26 231.33 180.08
202.76 180. 164. 156.94 114. 122. 161.93 137. 162.84 188.69
234.51 199.13 203.21 145.6 173.73 263.08 154. 239.04 177.35 224.98
117. 37. 97. 210.92 273.06 203.66 238.59 113. 224.53 169.64
146. 201.4 220. 34.93 254.01 212.73 176.45 184.61 124. 152.
233.6 193.23 205. 244.94 229.06 47. 167.38 99. 28.12 235.87
171. 212.28 180.98 169.19 175.54 30.84 116. 168.28 123. 186.43
172.82 182.8 217.27 182.34 246.3 30.39]

```

```

Unique categories for column 'BMI':
[27.99 30.13 31.66 ... 38.8 58.95 45.28]

```

```

Unique categories for column 'AlcoholDrinkers':
['No' 'Yes']

```

```

Unique categories for column 'HIVTesting':
['No' 'Yes']

```

```

Unique categories for column 'FluVaxLast12':
['Yes' 'No']

```

```

Unique categories for column 'PneumoVaxEver':
['Yes' 'No']

```

```

Unique categories for column 'TetanusLast10Tdap':
['Yes, received Tdap' 'Yes, received tetanus shot but not sure what type'
 'No, did not receive any tetanus shot in the past 10 years'
 'Yes, received tetanus shot, but not Tdap']

```

```

Unique categories for column 'HighRiskLastYear':
['No' 'Yes']

```

```

Unique categories for column 'CovidPos':
['No' 'Yes'
 'Tested positive using home test without a health professional']

```

```

[17]: # Visualize data types of each column/variable
      column_types = source_df.dtypes

      # Find columns with data type 'object' (strings), 'int' (integers), 'float'
      ↪ (float numbers)

```

```

string_columns = column_types[column_types == 'object'].index.tolist()
integer_columns = column_types[column_types == 'int'].index.tolist()
float_columns = column_types[column_types == 'float'].index.tolist()

# Display remaining categorical string variables for one-hot encoding
print("Columns holding data as strings:")
print(string_columns)
# Display numeric integer value (binary) variables
print("\nColumns holding data as numeric integers:")
print(integer_columns)
# Display numeric floating value variables
print("\nColumns holding data as numeric floats:")
print(float_columns)

```

Columns holding data as strings:

```

['State', 'Sex', 'GeneralHealth', 'LastCheckupTime', 'PhysicalActivities',
'RemovedTeeth', 'HadHeartAttack', 'HadAngina', 'HadStroke', 'HadAsthma',
'HadSkinCancer', 'HadCOPD', 'HadDepressiveDisorder', 'HadKidneyDisease',
'HadArthritis', 'HadDiabetes', 'DeafOrHardOfHearing', 'BlindOrVisionDifficulty',
'DifficultyConcentrating', 'DifficultyWalking', 'DifficultyDressingBathing',
'DifficultyErrands', 'SmokerStatus', 'ECigaretteUsage', 'ChestScan',
'RaceEthnicityCategory', 'AgeCategory', 'AlcoholDrinkers', 'HIVTesting',
'FluVaxLast12', 'PneumoVaxEver', 'TetanusLast10Tdap', 'HighRiskLastYear',
'CovidPos']

```

Columns holding data as numeric integers:

```

[]

```

Columns holding data as numeric floats:

```

['PhysicalHealthDays', 'MentalHealthDays', 'SleepHours', 'HeightInMeters',
'WeightInKilograms', 'BMI']

```

```

[18]: # Visualize summary statistics of numeric float variables
summary_statistics_df = source_df[float_columns].describe()
print(summary_statistics_df)

```

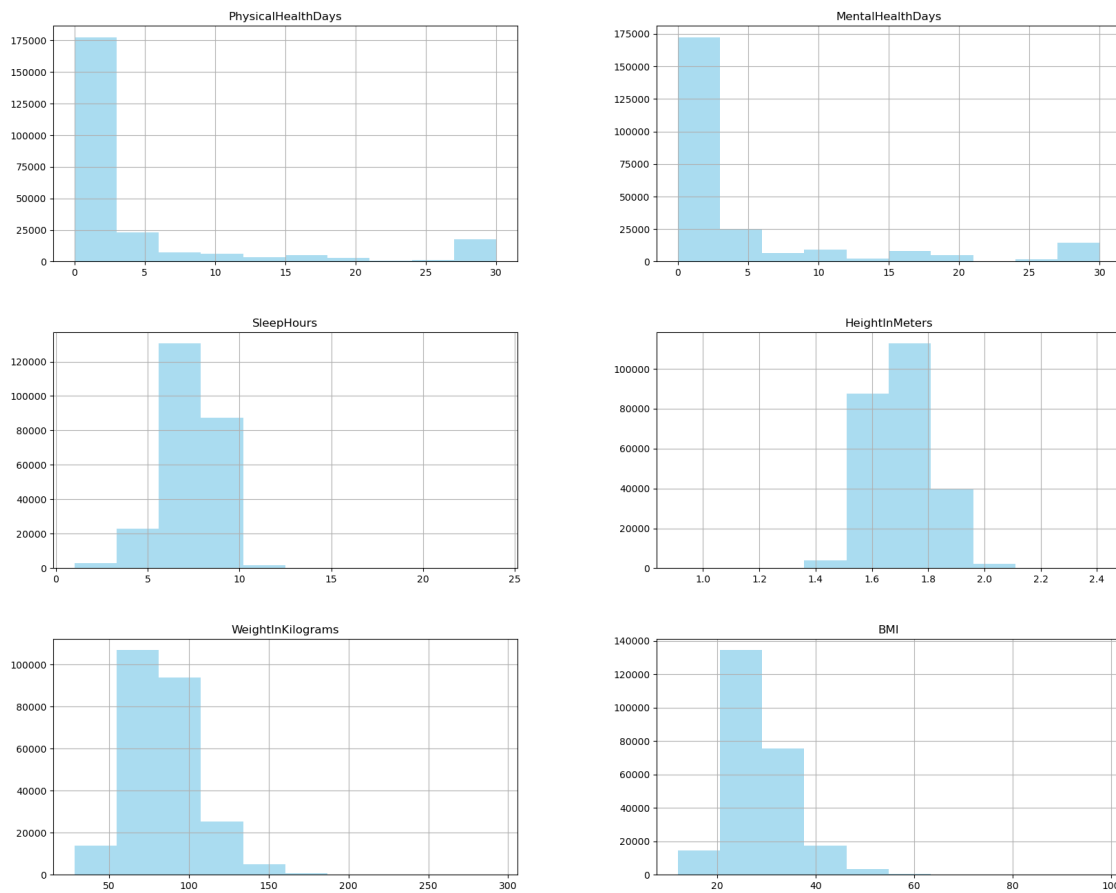
	PhysicalHealthDays	MentalHealthDays	SleepHours	HeightInMeters	\
count	246022.000000	246022.000000	246022.000000	246022.000000	
mean	4.119026	4.167140	7.021331	1.705150	
std	8.405844	8.102687	1.440681	0.106654	
min	0.000000	0.000000	1.000000	0.910000	
25%	0.000000	0.000000	6.000000	1.630000	
50%	0.000000	0.000000	7.000000	1.700000	
75%	3.000000	4.000000	8.000000	1.780000	
max	30.000000	30.000000	24.000000	2.410000	

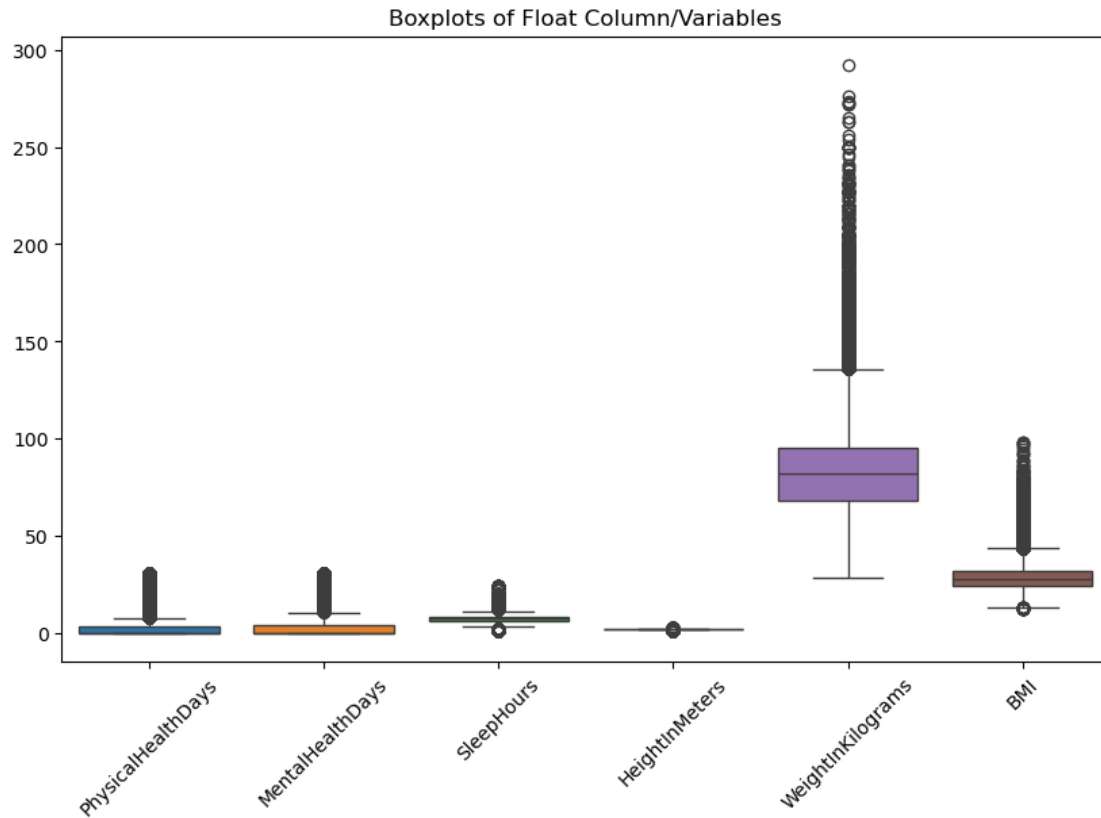
	WeightInKilograms	BMI
count	246022.000000	246022.000000

mean	83.615179	28.668136
std	21.323156	6.513973
min	28.120000	12.020000
25%	68.040000	24.270000
50%	81.650000	27.460000
75%	95.250000	31.890000
max	292.570000	97.650000

```
[19]: # Visualize histograms of numeric float variables
source_df[float_columns].hist(figsize=(20, 16) , color='skyblue', alpha=0.7)
plt.show();
```



```
[20]: # Visualize boxplots of numeric float variables
plt.figure(figsize=(10, 6));
sns.boxplot(data=source_df[float_columns]);
plt.title('Boxplots of Float Column/Variables');
plt.xticks(rotation=45); # Rotate x-axis labels for better readability
plt.show;
```

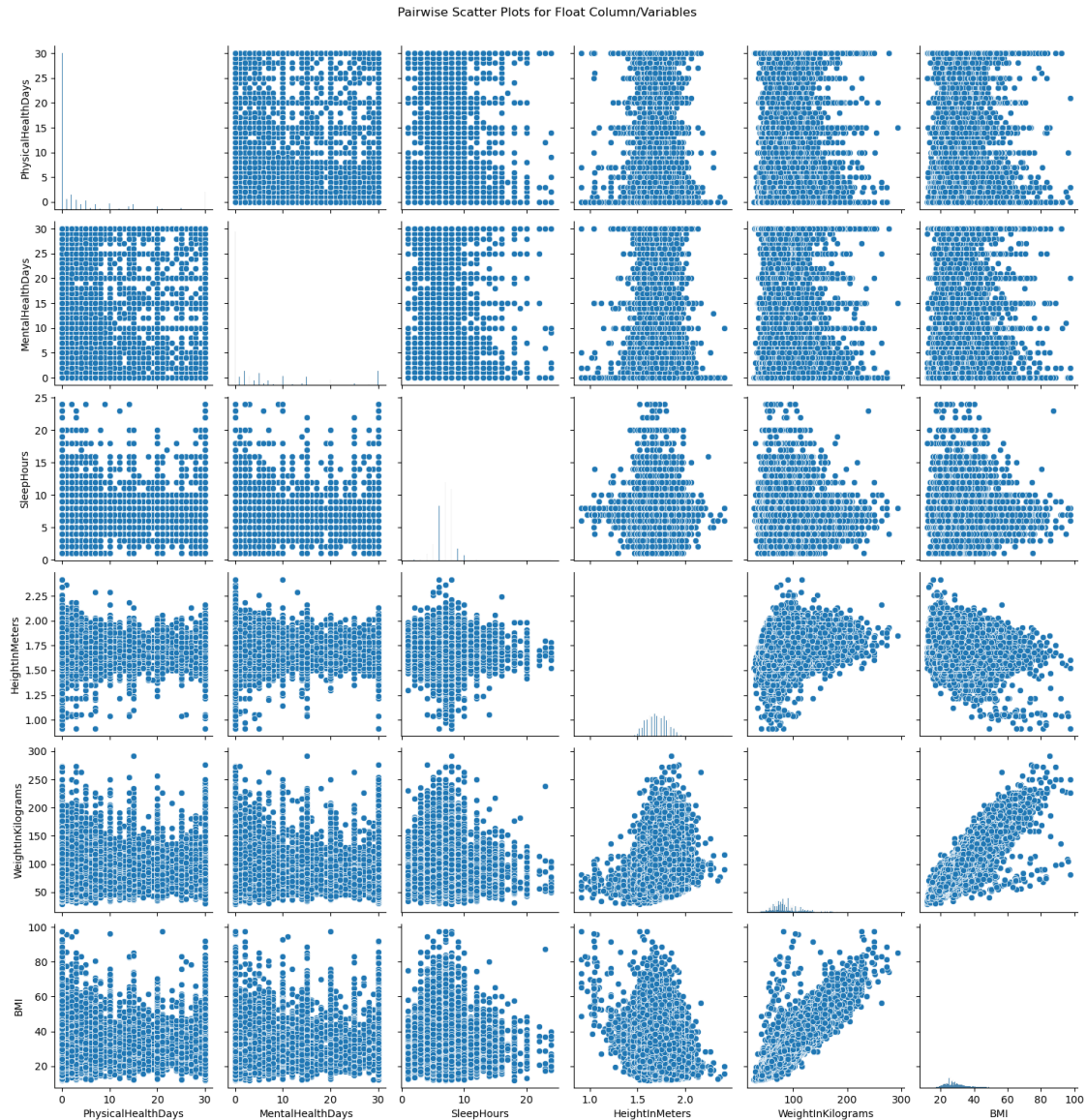


```
[21]: # COMMENT:
# Summary statistics, histograms, and boxplots all indicate the presence of
# outliers for the numeric float variables:
# 'PhysicalHealthDays', 'MentalHealthDays', 'SleepHours', 'HeightInMeters',
# 'WeightInKilograms', 'BMI'.
# No effort will be made to remove these outliers since they may be relevant
# in
# predicting the dependent variable, 'HadHeartAttack'.
```

```
[22]: # Visualize scatter plots of numeric float variables
plt.figure(figsize=(12, 8));
sns.pairplot(source_df[float_columns]);
plt.suptitle('Pairwise Scatter Plots for Float Column/Variables', y=1.02);
plt.show();
```

<Figure size 1200x800 with 0 Axes>





```
[23]: # COMMENT:
#      Scatter plots suggest positive correlations between:
#      'HeightInMeters' and 'BMI'
#      'WeightInKilograms' and 'BMI'
#      'WeightInKilograms' and 'HeightInMeters'
#      No other positive or negative correlations seem present
```

```
[24]: # Visualize correlation matrix
```

```
# COMMENT:
#      Given the large number of variables and anticipated difficulty in viewing
#      ↪ them
```

```
# all in the same correlation matrix, the attempt to visualize a correlation
# matrix will be made after the most relevant variables have been selected_
↳ using
# feature engineering.
```

```
[25]: #####
##      II.      FEATURE ENGINEERING:
#####
# Begin the process of
# -- making all variables numeric,
# -- normalize/scale all numeric float variables,
# -- selecting the best variables based on correlation matrix values
# -- adding any variable-to-variable interaction terms
```

```
[26]: # Make all column/variables binary with replacements of "1" or "0"
replacements = {
    'PhysicalActivities': {'Yes': 1, 'No': 0},
    'HadHeartAttack': {'Yes': 1, 'No': 0},
    'HadAngina': {'Yes': 1, 'No': 0},
    'HadStroke': {'Yes': 1, 'No': 0},
    'HadAsthma': {'Yes': 1, 'No': 0},
    'HadSkinCancer': {'Yes': 1, 'No': 0},
    'HadCOPD': {'Yes': 1, 'No': 0},
    'HadDepressiveDisorder': {'Yes': 1, 'No': 0},
    'HadKidneyDisease': {'Yes': 1, 'No': 0},
    'HadArthritis': {'Yes': 1, 'No': 0},
    'DeafOrHardOfHearing': {'Yes': 1, 'No': 0},
    'BlindOrVisionDifficulty': {'Yes': 1, 'No': 0},
    'DifficultyConcentrating': {'Yes': 1, 'No': 0},
    'DifficultyWalking': {'Yes': 1, 'No': 0},
    'DifficultyDressingBathing': {'Yes': 1, 'No': 0},
    'DifficultyErrands': {'Yes': 1, 'No': 0},
    'ChestScan': {'Yes': 1, 'No': 0},
    'gender': {'Yes': 1, 'No': 0},
    'AlcoholDrinkers': {'Yes': 1, 'No': 0},
    'HIVTesting': {'Yes': 1, 'No': 0},
    'FluVaxLast12': {'Yes': 1, 'No': 0},
    'PneumoVaxEver': {'Yes': 1, 'No': 0},
    'HighRiskLastYear': {'Yes': 1, 'No': 0}
}

# Apply replacements to multiple columns at once
source_df.replace(replacements, inplace=True)
```

```
[27]: # Verify changes. Print unique categories for each column/variable
for column in source_df.columns:
    unique_categories = source_df[column].unique()
```

```
print(f"Unique categories for column '{column}':")
print(unique_categories)
print()
```

Unique categories for column 'State':

```
['Alabama' 'Alaska' 'Arizona' 'Arkansas' 'California' 'Colorado'
 'Connecticut' 'Delaware' 'District of Columbia' 'Florida' 'Georgia'
 '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' 'Rhode Island' 'South Carolina'
 'South Dakota' 'Tennessee' 'Texas' 'Utah' 'Vermont' 'Virginia'
 'Washington' 'West Virginia' 'Wisconsin' 'Wyoming' 'Guam' 'Puerto Rico'
 'Virgin Islands']
```

Unique categories for column 'Sex':

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

Unique categories for column 'GeneralHealth':

```
['Very good' 'Fair' 'Good' 'Excellent' 'Poor']
```

Unique categories for column 'PhysicalHealthDays':

```
[ 4.  0.  5.  3.  2. 25. 30. 15. 29.  8. 16. 20. 10.  9.  7.  1. 21.  6.
 27. 14. 12. 11. 13. 28. 17. 23. 24. 26. 18. 22. 19.]
```

Unique categories for column 'MentalHealthDays':

```
[ 0. 15.  4. 25.  5. 30. 27.  3.  2.  1. 10. 20. 21.  6.  7.  8. 14.  9.
 12. 18. 29. 28. 17. 11. 16. 13. 26. 22. 24. 19. 23.]
```

Unique categories for column 'LastCheckupTime':

```
['Within past year (anytime less than 12 months ago)'
 '5 or more years ago'
 'Within past 2 years (1 year but less than 2 years ago)'
 'Within past 5 years (2 years but less than 5 years ago)']
```

Unique categories for column 'PhysicalActivities':

```
[1 0]
```

Unique categories for column 'SleepHours':

```
[ 9.  6.  8.  5.  7. 10.  4. 12.  3. 18. 11.  2.  1. 16. 14. 15. 13. 20.
 24. 23. 19. 17. 22.]
```

Unique categories for column 'RemovedTeeth':

```
['None of them' '6 or more, but not all' '1 to 5' 'All']
```

Unique categories for column 'HadHeartAttack':

```
[0 1]
```

Unique categories for column 'HadAngina':  
[0 1]

Unique categories for column 'HadStroke':  
[0 1]

Unique categories for column 'HadAsthma':  
[0 1]

Unique categories for column 'HadSkinCancer':  
[0 1]

Unique categories for column 'HadCOPD':  
[0 1]

Unique categories for column 'HadDepressiveDisorder':  
[0 1]

Unique categories for column 'HadKidneyDisease':  
[0 1]

Unique categories for column 'HadArthritis':  
[1 0]

Unique categories for column 'HadDiabetes':  
['No' 'Yes' 'Yes, but only during pregnancy (female)'  
 'No, pre-diabetes or borderline diabetes']

Unique categories for column 'DeafOrHardOfHearing':  
[0 1]

Unique categories for column 'BlindOrVisionDifficulty':  
[0 1]

Unique categories for column 'DifficultyConcentrating':  
[0 1]

Unique categories for column 'DifficultyWalking':  
[0 1]

Unique categories for column 'DifficultyDressingBathing':  
[0 1]

Unique categories for column 'DifficultyErrands':  
[0 1]

Unique categories for column 'SmokerStatus':

```
['Former smoker' 'Never smoked' 'Current smoker - now smokes every day'
 'Current smoker - now smokes some days']
```

Unique categories for column 'ECigaretteUsage':

```
['Never used e-cigarettes in my entire life' 'Use them some days'
 'Not at all (right now)' 'Use them every day']
```

Unique categories for column 'ChestScan':

```
[0 1]
```

Unique categories for column 'RaceEthnicityCategory':

```
['White only, Non-Hispanic' 'Black only, Non-Hispanic'
 'Other race only, Non-Hispanic' 'Multiracial, Non-Hispanic' 'Hispanic']
```

Unique categories for column 'AgeCategory':

```
['Age 65 to 69' 'Age 70 to 74' 'Age 75 to 79' 'Age 80 or older'
 'Age 50 to 54' 'Age 40 to 44' 'Age 60 to 64' 'Age 55 to 59'
 'Age 45 to 49' 'Age 35 to 39' 'Age 25 to 29' 'Age 30 to 34'
 'Age 18 to 24']
```

Unique categories for column 'HeightInMeters':

```
[1.6  1.78 1.85 1.7  1.55 1.63 1.75 1.68 1.83 1.52 1.88 1.5  1.73 1.65
 1.8  1.57 1.91 1.47 1.42 1.22 1.93 2.01 1.96 1.98 1.45 1.35 1.76 2.03
 2.16 1.51 1.53 1.69 1.56 1.84 1.9  1.54 1.72 1.87 1.74 1.4  1.64 1.58
 1.62 1.79 1.67 1.46 1.89 1.61 1.3  1.37 2.13 2.06 2.11 0.91 2.26 2.18
 1.77 2.36 1.59 1.86 1.82 1.66 1.71 1.95 1.05 2.08 1.49 1.38 1.81 1.44
 1.48 1.19 1.32 1.24 1.07 1.04 1.27 1.1  1.92 1.2  2.24 1.12 1.03 0.97
 1.25 2.29 1.16 1.18 1.09 2.41 1.  1.17 1.08 1.43 1.14 1.02 2.  2.02
 0.95 2.34 2.21]
```

Unique categories for column 'WeightInKilograms':

```
[ 71.67  95.25 108.86  90.72  79.38 120.2  88.  74.84  78.02  63.5
122.47 115.67  81.65  86.18  76.2  54.88  72.57  88.45 104.33  52.16
 68.04  65.77  56.7  94.8 123.83  50.8  68.95 113.4  83.91  77.56
 68.49  82.1  80.74 106.14  58.06  61.69  57.61  84.82  70.76  70.31
 91.63 102.06  48.08  61.23 109.77  99.79  58.97 110.68  64.86 111.13
 45.36  79.83  98.88  55.34 101.6  77.11  93.89  71.21  49.9  96.16
163.29 120.66  97.52  88.9  44.91  85.73  83.46  92.99 132.  67.59
 92.08  73.48 107.5 107.95  91.17  74.39  64.41  62.6  46.72 103.42
 87.09  89.81  83.01 100.7  56.25  96.62  66.68  67.13  69.4  58.51
 78.93  95.71  63.05  49.44 127.01 145.15 122.02 107.05 126.55 117.03
 47.17 181.44  65.32 117.93 136.08  78.47  52.62 121.56  73.94  82.55
106.59  59.87 110.22  62.14  51.71  93.44  54.43  85.28  59.42  66.22
 76.66  55.79  75.3  97.07  87.54  69.85 124.74  63.96  47.63  94.35
 97.98  89.36  92.53 101.15 149.69 129.27  84.37 195.04  99.34 114.31
 53.07  81.19  75.75 124.28 112.94  80.29 114.76  45.81  53.52 133.81
 51.26 158.76  60.78  46.27  72.12 131.54 127.91  53.98  98.43 130.63
143.34 102.51 115.21  90.26 166.92 109.32  40.37 135.62 204.12 129.73]
```

127.46	138.35	105.69	119.75	48.53	140.61	105.23	139.25	126.1	135.17
102.97	122.92	57.15	38.56	60.33	131.09	148.78	116.57	112.49	86.64
112.04	172.37	133.36	118.84	50.35	103.87	111.58	121.11	113.85	73.03
142.88	134.26	123.38	37.19	119.29	36.29	48.99	43.09	41.73	35.38
104.78	144.24	167.83	149.23	37.65	86.	147.42	165.56	154.22	136.98
108.41	155.58	206.38	148.32	42.18	44.45	90.	191.87	249.48	67.
44.	40.82	156.49	53.	139.71	130.18	118.39	100.	151.95	165.11
43.54	134.72	141.52	125.19	75.	250.	116.12	73.	100.24	74.
200.	80.	82.	54.	66.	152.41	39.46	41.28	190.51	188.24
59.	70.	170.1	46.	265.	168.74	190.	55.	93.	159.66
78.	38.1	185.07	104.	183.7	125.65	68.	134.	130.	32.21
143.79	137.89	179.17	105.	65.	32.	292.57	85.	72.	174.63
50.	128.37	62.	87.	176.9	39.92	76.	128.82	58.	156.04
121.	42.64	89.	146.96	146.06	171.46	227.25	29.48	190.06	161.03
226.8	132.45	137.44	64.	56.	141.07	52.	63.	120.	83.
57.	31.75	77.	96.	60.	115.	41.	150.59	272.16	48.
39.01	95.	197.31	158.3	45.	94.	240.4	49.	157.85	108.
185.	61.	34.02	132.9	84.	229.97	138.8	81.	79.	92.
107.	155.13	208.65	69.	111.	110.	151.05	210.	140.16	35.83
146.51	117.48	102.	125.	151.5	36.74	38.	135.	71.	147.87
153.77	170.	91.	98.	192.32	186.88	118.	160.12	160.	170.55
201.85	184.16	175.09	142.43	169.	166.01	180.53	196.41	162.39	40.
171.91	195.95	136.53	153.31	159.21	164.2	219.99	141.97	173.27	34.47
213.19	276.24	199.58	215.46	217.72	175.99	200.03	230.88	33.57	185.52
103.	152.86	101.	160.57	150.14	157.4	145.	150.	163.75	191.42
174.18	164.65	256.28	205.48	192.78	161.48	178.26	179.62	144.7	205.02
178.72	154.68	166.47	177.81	200.49	231.79	238.14	227.7	273.52	211.83
223.62	197.77	189.15	185.97	250.38	183.25	181.89	222.26	231.33	180.08
202.76	180.	164.	156.94	114.	122.	161.93	137.	162.84	188.69
234.51	199.13	203.21	145.6	173.73	263.08	154.	239.04	177.35	224.98
117.	37.	97.	210.92	273.06	203.66	238.59	113.	224.53	169.64
146.	201.4	220.	34.93	254.01	212.73	176.45	184.61	124.	152.
233.6	193.23	205.	244.94	229.06	47.	167.38	99.	28.12	235.87
171.	212.28	180.98	169.19	175.54	30.84	116.	168.28	123.	186.43
172.82	182.8	217.27	182.34	246.3	30.39]				

Unique categories for column 'BMI':  
[27.99 30.13 31.66 ... 38.8 58.95 45.28]

Unique categories for column 'AlcoholDrinkers':  
[0 1]

Unique categories for column 'HIVTesting':  
[0 1]

Unique categories for column 'FluVaxLast12':  
[1 0]

Unique categories for column 'PneumoVaxEver':  
[1 0]

Unique categories for column 'TetanusLast10Tdap':  
['Yes, received Tdap' 'Yes, received tetanus shot but not sure what type'  
 'No, did not receive any tetanus shot in the past 10 years'  
 'Yes, received tetanus shot, but not Tdap']

Unique categories for column 'HighRiskLastYear':  
[0 1]

Unique categories for column 'CovidPos':  
['No' 'Yes'  
 'Tested positive using home test without a health professional']

```
[28]: # In anticipation of one-hot encoding for remaining categorical string_
      ↪ variables,
      # check data types of each column.
      column_types = source_df.dtypes

      # Find columns with data type 'object' (strings), 'int' (integers), 'float'
      ↪ (float numbers)
      string_columns = column_types[column_types == 'object'].index.tolist()
      integer_columns = column_types[column_types == 'int'].index.tolist()
      float_columns = column_types[column_types == 'float'].index.tolist()

      # Display remaining categorical string variables for one-hot encoding
      print("Columns holding values as strings:")
      print(string_columns)
      # Display numeric integer value (binary) variables
      print("\nColumns holding values as integers:")
      print(integer_columns)
      # Display numeric floating value variables
      print("\nColumns holding values as floats:")
      print(float_columns)
```

Columns holding values as strings:  
['State', 'Sex', 'GeneralHealth', 'LastCheckupTime', 'RemovedTeeth',  
 'HadDiabetes', 'SmokerStatus', 'ECigaretteUsage', 'RaceEthnicityCategory',  
 'AgeCategory', 'TetanusLast10Tdap', 'CovidPos']

Columns holding values as integers:  
['PhysicalActivities', 'HadHeartAttack', 'HadAngina', 'HadStroke', 'HadAsthma',  
 'HadSkinCancer', 'HadCOPD', 'HadDepressiveDisorder', 'HadKidneyDisease',  
 'HadArthritis', 'DeafOrHardOfHearing', 'BlindOrVisionDifficulty',  
 'DifficultyConcentrating', 'DifficultyWalking', 'DifficultyDressingBathing',  
 'DifficultyErrands', 'ChestScan', 'AlcoholDrinkers', 'HIVTesting',

```
'FluVaxLast12', 'PneumoVaxEver', 'HighRiskLastYear']
```

Columns holding values as floats:

```
['PhysicalHealthDays', 'MentalHealthDays', 'SleepHours', 'HeightInMeters',  
'WeightInKilograms', 'BMI']
```

```
[29]: # Apply one-hot encoding to categorical column/variables holding values  
#      as strings  
encoded_df = pd.get_dummies(source_df, columns=string_columns)
```

```
[30]: # Verify changes. Check data types of each column  
column_types = encoded_df.dtypes  
  
# Find columns with data type 'object' (strings)  
string_columns = column_types[column_types == 'object'].index.tolist()  
integer_columns = column_types[column_types == 'int'].index.tolist()  
float_columns = column_types[column_types == 'float'].index.tolist()  
boolean_columns = column_types[column_types == 'bool'].index.tolist()  
  
print("Columns holding values as strings:")  
print(string_columns)  
print("\nColumns holding values as integers:")  
print(integer_columns)  
print("\nColumns holding values as floats:")  
print(float_columns)  
print("\nColumns holding values as booleans (True/False):")  
print(boolean_columns)
```

Columns holding values as strings:

```
[]
```

Columns holding values as integers:

```
['PhysicalActivities', 'HadHeartAttack', 'HadAngina', 'HadStroke', 'HadAsthma',  
'HadSkinCancer', 'HadCOPD', 'HadDepressiveDisorder', 'HadKidneyDisease',  
'HadArthritis', 'DeafOrHardOfHearing', 'BlindOrVisionDifficulty',  
'DifficultyConcentrating', 'DifficultyWalking', 'DifficultyDressingBathing',  
'DifficultyErrands', 'ChestScan', 'AlcoholDrinkers', 'HIVTesting',  
'FluVaxLast12', 'PneumoVaxEver', 'HighRiskLastYear']
```

Columns holding values as floats:

```
['PhysicalHealthDays', 'MentalHealthDays', 'SleepHours', 'HeightInMeters',  
'WeightInKilograms', 'BMI']
```

Columns holding values as booleans (True/False):

```
['State_Alabama', 'State_Alaska', 'State_Arizona', 'State_Arkansas',  
'State_California', 'State_Colorado', 'State_Connecticut', 'State_Delaware',  
'State_District of Columbia', 'State_Florida', 'State_Georgia', 'State_Guam',  
'State_Hawaii', 'State_Idaho', 'State_Illinois', 'State_Indiana', 'State_Iowa',
```



```

'State_Kansas', 'State_Kentucky', 'State_Louisiana', 'State_Maine',
'State_Maryland', 'State_Massachusetts', 'State_Michigan', 'State_Minnesota',
'State_Mississippi', 'State_Missouri', 'State_Montana', 'State_Nebraska',
'State_Nevada', 'State_New Hampshire', 'State_New Jersey', 'State_New Mexico',
'State_New York', 'State_North Carolina', 'State_North Dakota', 'State_Ohio',
'State_Oklahoma', 'State_Oregon', 'State_Pennsylvania', 'State_Puerto Rico',
'State_Rhode Island', 'State_South Carolina', 'State_South Dakota',
'State_Tennessee', 'State_Texas', 'State_Utah', 'State_Vermont', 'State_Virgin
Islands', 'State_Virginia', 'State_Washington', 'State_West Virginia',
'State_Wisconsin', 'State_Wyoming', 'Sex_Female', 'Sex_Male',
'GeneralHealth_Excellent', 'GeneralHealth_Fair', 'GeneralHealth_Good',
'GeneralHealth_Poor', 'GeneralHealth_Very good', 'LastCheckupTime_5 or more
years ago', 'LastCheckupTime_Within past 2 years (1 year but less than 2 years
ago)', 'LastCheckupTime_Within past 5 years (2 years but less than 5 years
ago)', 'LastCheckupTime_Within past year (anytime less than 12 months ago)',
'RemovedTeeth_1 to 5', 'RemovedTeeth_6 or more, but not all',
'RemovedTeeth_All', 'RemovedTeeth_None of them', 'HadDiabetes_No',
'HadDiabetes_No, pre-diabetes or borderline diabetes', 'HadDiabetes_Yes',
'HadDiabetes_Yes, but only during pregnancy (female)', 'SmokerStatus_Current
smoker - now smokes every day', 'SmokerStatus_Current smoker - now smokes some
days', 'SmokerStatus_Former smoker', 'SmokerStatus_Never smoked',
'ECigaretteUsage_Never used e-cigarettes in my entire life',
'ECigaretteUsage_Not at all (right now)', 'ECigaretteUsage_Use them every day',
'ECigaretteUsage_Use them some days', 'RaceEthnicityCategory_Black only, Non-
Hispanic', 'RaceEthnicityCategory_Hispanic', 'RaceEthnicityCategory_Multiracial,
Non-Hispanic', 'RaceEthnicityCategory_Other race only, Non-Hispanic',
'RaceEthnicityCategory_White only, Non-Hispanic', 'AgeCategory_Age 18 to 24',
'AgeCategory_Age 25 to 29', 'AgeCategory_Age 30 to 34', 'AgeCategory_Age 35 to
39', 'AgeCategory_Age 40 to 44', 'AgeCategory_Age 45 to 49', 'AgeCategory_Age 50
to 54', 'AgeCategory_Age 55 to 59', 'AgeCategory_Age 60 to 64', 'AgeCategory_Age
65 to 69', 'AgeCategory_Age 70 to 74', 'AgeCategory_Age 75 to 79',
'AgeCategory_Age 80 or older', 'TetanusLast10Tdap_No, did not receive any
tetanus shot in the past 10 years', 'TetanusLast10Tdap_Yes, received Tdap',
'TetanusLast10Tdap_Yes, received tetanus shot but not sure what type',
'TetanusLast10Tdap_Yes, received tetanus shot, but not Tdap', 'CovidPos_No',
'CovidPos_Testing positive using home test without a health professional',
'CovidPos_Yes']

```

```

[31]: # Verify that all variables are now some form of numeric:
# -- integer, binary 0 or 1
# -- float
# -- boolean, True "1"/False "0" (after one-hot encoding)
# Print unique categories for each column/variable
for column in encoded_df.columns:
    unique_categories = encoded_df[column].unique()
    print(f"Unique categories for column '{column}':")
    print(unique_categories)

```

```
print()
```

```
Unique categories for column 'PhysicalHealthDays':
```

```
[ 4.  0.  5.  3.  2. 25. 30. 15. 29.  8. 16. 20. 10.  9.  7.  1. 21.  6.
 27. 14. 12. 11. 13. 28. 17. 23. 24. 26. 18. 22. 19.]
```

```
Unique categories for column 'MentalHealthDays':
```

```
[ 0. 15.  4. 25.  5. 30. 27.  3.  2.  1. 10. 20. 21.  6.  7.  8. 14.  9.
 12. 18. 29. 28. 17. 11. 16. 13. 26. 22. 24. 19. 23.]
```

```
Unique categories for column 'PhysicalActivities':
```

```
[1 0]
```

```
Unique categories for column 'SleepHours':
```

```
[ 9.  6.  8.  5.  7. 10.  4. 12.  3. 18. 11.  2.  1. 16. 14. 15. 13. 20.
 24. 23. 19. 17. 22.]
```

```
Unique categories for column 'HadHeartAttack':
```

```
[0 1]
```

```
Unique categories for column 'HadAngina':
```

```
[0 1]
```

```
Unique categories for column 'HadStroke':
```

```
[0 1]
```

```
Unique categories for column 'HadAsthma':
```

```
[0 1]
```

```
Unique categories for column 'HadSkinCancer':
```

```
[0 1]
```

```
Unique categories for column 'HadCOPD':
```

```
[0 1]
```

```
Unique categories for column 'HadDepressiveDisorder':
```

```
[0 1]
```

```
Unique categories for column 'HadKidneyDisease':
```

```
[0 1]
```

```
Unique categories for column 'HadArthritis':
```

```
[1 0]
```

```
Unique categories for column 'DeafOrHardOfHearing':
```

```
[0 1]
```

```
Unique categories for column 'BlindOrVisionDifficulty':
```

[0 1]

Unique categories for column 'DifficultyConcentrating':

[0 1]

Unique categories for column 'DifficultyWalking':

[0 1]

Unique categories for column 'DifficultyDressingBathing':

[0 1]

Unique categories for column 'DifficultyErrands':

[0 1]

Unique categories for column 'ChestScan':

[0 1]

Unique categories for column 'HeightInMeters':

[1.6 1.78 1.85 1.7 1.55 1.63 1.75 1.68 1.83 1.52 1.88 1.5 1.73 1.65  
1.8 1.57 1.91 1.47 1.42 1.22 1.93 2.01 1.96 1.98 1.45 1.35 1.76 2.03  
2.16 1.51 1.53 1.69 1.56 1.84 1.9 1.54 1.72 1.87 1.74 1.4 1.64 1.58  
1.62 1.79 1.67 1.46 1.89 1.61 1.3 1.37 2.13 2.06 2.11 0.91 2.26 2.18  
1.77 2.36 1.59 1.86 1.82 1.66 1.71 1.95 1.05 2.08 1.49 1.38 1.81 1.44  
1.48 1.19 1.32 1.24 1.07 1.04 1.27 1.1 1.92 1.2 2.24 1.12 1.03 0.97  
1.25 2.29 1.16 1.18 1.09 2.41 1. 1.17 1.08 1.43 1.14 1.02 2. 2.02  
0.95 2.34 2.21]

Unique categories for column 'WeightInKilograms':

[ 71.67 95.25 108.86 90.72 79.38 120.2 88. 74.84 78.02 63.5  
122.47 115.67 81.65 86.18 76.2 54.88 72.57 88.45 104.33 52.16  
68.04 65.77 56.7 94.8 123.83 50.8 68.95 113.4 83.91 77.56  
68.49 82.1 80.74 106.14 58.06 61.69 57.61 84.82 70.76 70.31  
91.63 102.06 48.08 61.23 109.77 99.79 58.97 110.68 64.86 111.13  
45.36 79.83 98.88 55.34 101.6 77.11 93.89 71.21 49.9 96.16  
163.29 120.66 97.52 88.9 44.91 85.73 83.46 92.99 132. 67.59  
92.08 73.48 107.5 107.95 91.17 74.39 64.41 62.6 46.72 103.42  
87.09 89.81 83.01 100.7 56.25 96.62 66.68 67.13 69.4 58.51  
78.93 95.71 63.05 49.44 127.01 145.15 122.02 107.05 126.55 117.03  
47.17 181.44 65.32 117.93 136.08 78.47 52.62 121.56 73.94 82.55  
106.59 59.87 110.22 62.14 51.71 93.44 54.43 85.28 59.42 66.22  
76.66 55.79 75.3 97.07 87.54 69.85 124.74 63.96 47.63 94.35  
97.98 89.36 92.53 101.15 149.69 129.27 84.37 195.04 99.34 114.31  
53.07 81.19 75.75 124.28 112.94 80.29 114.76 45.81 53.52 133.81  
51.26 158.76 60.78 46.27 72.12 131.54 127.91 53.98 98.43 130.63  
143.34 102.51 115.21 90.26 166.92 109.32 40.37 135.62 204.12 129.73  
127.46 138.35 105.69 119.75 48.53 140.61 105.23 139.25 126.1 135.17  
102.97 122.92 57.15 38.56 60.33 131.09 148.78 116.57 112.49 86.64  
112.04 172.37 133.36 118.84 50.35 103.87 111.58 121.11 113.85 73.03]

142.88	134.26	123.38	37.19	119.29	36.29	48.99	43.09	41.73	35.38
104.78	144.24	167.83	149.23	37.65	86.	147.42	165.56	154.22	136.98
108.41	155.58	206.38	148.32	42.18	44.45	90.	191.87	249.48	67.
44.	40.82	156.49	53.	139.71	130.18	118.39	100.	151.95	165.11
43.54	134.72	141.52	125.19	75.	250.	116.12	73.	100.24	74.
200.	80.	82.	54.	66.	152.41	39.46	41.28	190.51	188.24
59.	70.	170.1	46.	265.	168.74	190.	55.	93.	159.66
78.	38.1	185.07	104.	183.7	125.65	68.	134.	130.	32.21
143.79	137.89	179.17	105.	65.	32.	292.57	85.	72.	174.63
50.	128.37	62.	87.	176.9	39.92	76.	128.82	58.	156.04
121.	42.64	89.	146.96	146.06	171.46	227.25	29.48	190.06	161.03
226.8	132.45	137.44	64.	56.	141.07	52.	63.	120.	83.
57.	31.75	77.	96.	60.	115.	41.	150.59	272.16	48.
39.01	95.	197.31	158.3	45.	94.	240.4	49.	157.85	108.
185.	61.	34.02	132.9	84.	229.97	138.8	81.	79.	92.
107.	155.13	208.65	69.	111.	110.	151.05	210.	140.16	35.83
146.51	117.48	102.	125.	151.5	36.74	38.	135.	71.	147.87
153.77	170.	91.	98.	192.32	186.88	118.	160.12	160.	170.55
201.85	184.16	175.09	142.43	169.	166.01	180.53	196.41	162.39	40.
171.91	195.95	136.53	153.31	159.21	164.2	219.99	141.97	173.27	34.47
213.19	276.24	199.58	215.46	217.72	175.99	200.03	230.88	33.57	185.52
103.	152.86	101.	160.57	150.14	157.4	145.	150.	163.75	191.42
174.18	164.65	256.28	205.48	192.78	161.48	178.26	179.62	144.7	205.02
178.72	154.68	166.47	177.81	200.49	231.79	238.14	227.7	273.52	211.83
223.62	197.77	189.15	185.97	250.38	183.25	181.89	222.26	231.33	180.08
202.76	180.	164.	156.94	114.	122.	161.93	137.	162.84	188.69
234.51	199.13	203.21	145.6	173.73	263.08	154.	239.04	177.35	224.98
117.	37.	97.	210.92	273.06	203.66	238.59	113.	224.53	169.64
146.	201.4	220.	34.93	254.01	212.73	176.45	184.61	124.	152.
233.6	193.23	205.	244.94	229.06	47.	167.38	99.	28.12	235.87
171.	212.28	180.98	169.19	175.54	30.84	116.	168.28	123.	186.43
172.82	182.8	217.27	182.34	246.3	30.39]				

Unique categories for column 'BMI':  
[27.99 30.13 31.66 ... 38.8 58.95 45.28]

Unique categories for column 'AlcoholDrinkers':  
[0 1]

Unique categories for column 'HIVTesting':  
[0 1]

Unique categories for column 'FluVaxLast12':  
[1 0]

Unique categories for column 'PneumoVaxEver':  
[1 0]

Unique categories for column 'HighRiskLastYear':  
[0 1]

Unique categories for column 'State\_Alabama':  
[ True False]

Unique categories for column 'State\_Alaska':  
[False True]

Unique categories for column 'State\_Arizona':  
[False True]

Unique categories for column 'State\_Arkansas':  
[False True]

Unique categories for column 'State\_California':  
[False True]

Unique categories for column 'State\_Colorado':  
[False True]

Unique categories for column 'State\_Connecticut':  
[False True]

Unique categories for column 'State\_Delaware':  
[False True]

Unique categories for column 'State\_District of Columbia':  
[False True]

Unique categories for column 'State\_Florida':  
[False True]

Unique categories for column 'State\_Georgia':  
[False True]

Unique categories for column 'State\_Guam':  
[False True]

Unique categories for column 'State\_Hawaii':  
[False True]

Unique categories for column 'State\_Idaho':  
[False True]

Unique categories for column 'State\_Illinois':  
[False True]

Unique categories for column 'State\_Indiana':  
[False True]

Unique categories for column 'State\_Iowa':  
[False True]

Unique categories for column 'State\_Kansas':  
[False True]

Unique categories for column 'State\_Kentucky':  
[False True]

Unique categories for column 'State\_Louisiana':  
[False True]

Unique categories for column 'State\_Maine':  
[False True]

Unique categories for column 'State\_Maryland':  
[False True]

Unique categories for column 'State\_Massachusetts':  
[False True]

Unique categories for column 'State\_Michigan':  
[False True]

Unique categories for column 'State\_Minnesota':  
[False True]

Unique categories for column 'State\_Mississippi':  
[False True]

Unique categories for column 'State\_Missouri':  
[False True]

Unique categories for column 'State\_Montana':  
[False True]

Unique categories for column 'State\_Nebraska':  
[False True]

Unique categories for column 'State\_Nevada':  
[False True]

Unique categories for column 'State\_New Hampshire':  
[False True]

Unique categories for column 'State\_New Jersey':  
[False True]

Unique categories for column 'State\_New Mexico':  
[False True]

Unique categories for column 'State\_New York':  
[False True]

Unique categories for column 'State\_North Carolina':  
[False True]

Unique categories for column 'State\_North Dakota':  
[False True]

Unique categories for column 'State\_Ohio':  
[False True]

Unique categories for column 'State\_Oklahoma':  
[False True]

Unique categories for column 'State\_Oregon':  
[False True]

Unique categories for column 'State\_Pennsylvania':  
[False True]

Unique categories for column 'State\_Puerto Rico':  
[False True]

Unique categories for column 'State\_Rhode Island':  
[False True]

Unique categories for column 'State\_South Carolina':  
[False True]

Unique categories for column 'State\_South Dakota':  
[False True]

Unique categories for column 'State\_Tennessee':  
[False True]

Unique categories for column 'State\_Texas':  
[False True]

Unique categories for column 'State\_Utah':  
[False True]

Unique categories for column 'State\_Vermont':  
[False True]

Unique categories for column 'State\_Virgin Islands':  
[False True]

Unique categories for column 'State\_Virginia':  
[False True]

Unique categories for column 'State\_Washington':  
[False True]

Unique categories for column 'State\_West Virginia':  
[False True]

Unique categories for column 'State\_Wisconsin':  
[False True]

Unique categories for column 'State\_Wyoming':  
[False True]

Unique categories for column 'Sex\_Female':  
[ True False]

Unique categories for column 'Sex\_Male':  
[False True]

Unique categories for column 'GeneralHealth\_Excellent':  
[False True]

Unique categories for column 'GeneralHealth\_Fair':  
[False True]

Unique categories for column 'GeneralHealth\_Good':  
[False True]

Unique categories for column 'GeneralHealth\_Poor':  
[False True]

Unique categories for column 'GeneralHealth\_Very good':  
[ True False]

Unique categories for column 'LastCheckupTime\_5 or more years ago':  
[False True]

Unique categories for column 'LastCheckupTime\_Within past 2 years (1 year but less than 2 years ago)':  
[False True]



Unique categories for column 'LastCheckupTime\_Within past 5 years (2 years but less than 5 years ago)':

[False True]

Unique categories for column 'LastCheckupTime\_Within past year (anytime less than 12 months ago)':

[ True False]

Unique categories for column 'RemovedTeeth\_1 to 5':

[False True]

Unique categories for column 'RemovedTeeth\_6 or more, but not all':

[False True]

Unique categories for column 'RemovedTeeth\_All':

[False True]

Unique categories for column 'RemovedTeeth\_None of them':

[ True False]

Unique categories for column 'HadDiabetes\_No':

[ True False]

Unique categories for column 'HadDiabetes\_No, pre-diabetes or borderline diabetes':

[False True]

Unique categories for column 'HadDiabetes\_Yes':

[False True]

Unique categories for column 'HadDiabetes\_Yes, but only during pregnancy (female)':

[False True]

Unique categories for column 'SmokerStatus\_Current smoker - now smokes every day':

[False True]

Unique categories for column 'SmokerStatus\_Current smoker - now smokes some days':

[False True]

Unique categories for column 'SmokerStatus\_Former smoker':

[ True False]

Unique categories for column 'SmokerStatus\_Never smoked':

[False True]

Unique categories for column 'ECigaretteUsage\_Never used e-cigarettes in my entire life':

[ True False]

Unique categories for column 'ECigaretteUsage\_Not at all (right now)':

[False True]

Unique categories for column 'ECigaretteUsage\_Use them every day':

[False True]

Unique categories for column 'ECigaretteUsage\_Use them some days':

[False True]

Unique categories for column 'RaceEthnicityCategory\_Black only, Non-Hispanic':

[False True]

Unique categories for column 'RaceEthnicityCategory\_Hispanic':

[False True]

Unique categories for column 'RaceEthnicityCategory\_Multiracial, Non-Hispanic':

[False True]

Unique categories for column 'RaceEthnicityCategory\_Other race only, Non-Hispanic':

[False True]

Unique categories for column 'RaceEthnicityCategory\_White only, Non-Hispanic':

[ True False]

Unique categories for column 'AgeCategory\_Age 18 to 24':

[False True]

Unique categories for column 'AgeCategory\_Age 25 to 29':

[False True]

Unique categories for column 'AgeCategory\_Age 30 to 34':

[False True]

Unique categories for column 'AgeCategory\_Age 35 to 39':

[False True]

Unique categories for column 'AgeCategory\_Age 40 to 44':

[False True]

Unique categories for column 'AgeCategory\_Age 45 to 49':

[False True]

Unique categories for column 'AgeCategory\_Age 50 to 54':  
[False True]

Unique categories for column 'AgeCategory\_Age 55 to 59':  
[False True]

Unique categories for column 'AgeCategory\_Age 60 to 64':  
[False True]

Unique categories for column 'AgeCategory\_Age 65 to 69':  
[ True False]

Unique categories for column 'AgeCategory\_Age 70 to 74':  
[False True]

Unique categories for column 'AgeCategory\_Age 75 to 79':  
[False True]

Unique categories for column 'AgeCategory\_Age 80 or older':  
[False True]

Unique categories for column 'TetanusLast10Tdap\_No, did not receive any tetanus shot in the past 10 years':  
[False True]

Unique categories for column 'TetanusLast10Tdap\_Yes, received Tdap':  
[ True False]

Unique categories for column 'TetanusLast10Tdap\_Yes, received tetanus shot but not sure what type':  
[False True]

Unique categories for column 'TetanusLast10Tdap\_Yes, received tetanus shot, but not Tdap':  
[False True]

Unique categories for column 'CovidPos\_No':  
[ True False]

Unique categories for column 'CovidPos\_Test positive using home test without a health professional':  
[False True]

Unique categories for column 'CovidPos\_Yes':  
[False True]

```
[32]: # Normalize/scale all numeric float variables
# Initialize scaler objects
standard_scaler = StandardScaler()
min_max_scaler = MinMaxScaler()

# Fit and transform selected columns using standard scaler
encoded_df[float_columns] = standard_scaler.
    ↪fit_transform(encoded_df[float_columns])

# Fit and transform selected columns using min-max scaler
encoded_df[float_columns] = min_max_scaler.
    ↪fit_transform(encoded_df[float_columns])
```

```
[33]: # Verify that all variables are now some form of numeric:
# -- integer, binary 0 or 1
# -- float (normalized/scaled between 0 and 1)
# -- boolean, True "1"/False "0" (after one-hot encoding)
# Print unique categories for each column/variable
for column in encoded_df.columns:
    unique_categories = encoded_df[column].unique()
    print(f"Unique categories for column '{column}':")
    print(unique_categories)
    print()
```

Unique categories for column 'PhysicalHealthDays':

```
[0.13333333 0.          0.16666667 0.1          0.06666667 0.83333333
 1.          0.5          0.96666667 0.26666667 0.53333333 0.66666667
 0.33333333 0.3          0.23333333 0.03333333 0.7          0.2
 0.9          0.46666667 0.4          0.36666667 0.43333333 0.93333333
 0.56666667 0.76666667 0.8          0.86666667 0.6          0.73333333
 0.63333333]
```

Unique categories for column 'MentalHealthDays':

```
[0.          0.5          0.13333333 0.83333333 0.16666667 1.
 0.9          0.1          0.06666667 0.03333333 0.33333333 0.66666667
 0.7          0.2          0.23333333 0.26666667 0.46666667 0.3
 0.4          0.6          0.96666667 0.93333333 0.56666667 0.36666667
 0.53333333 0.43333333 0.86666667 0.73333333 0.8          0.63333333
 0.76666667]
```

Unique categories for column 'PhysicalActivities':

```
[1 0]
```

Unique categories for column 'SleepHours':

```
[0.34782609 0.2173913  0.30434783 0.17391304 0.26086957 0.39130435
 0.13043478 0.47826087 0.08695652 0.73913043 0.43478261 0.04347826
 0.          0.65217391 0.56521739 0.60869565 0.52173913 0.82608696
 1.          0.9562174  0.7826087  0.69565217 0.91304348]
```

Unique categories for column 'HadHeartAttack':  
[0 1]

Unique categories for column 'HadAngina':  
[0 1]

Unique categories for column 'HadStroke':  
[0 1]

Unique categories for column 'HadAsthma':  
[0 1]

Unique categories for column 'HadSkinCancer':  
[0 1]

Unique categories for column 'HadCOPD':  
[0 1]

Unique categories for column 'HadDepressiveDisorder':  
[0 1]

Unique categories for column 'HadKidneyDisease':  
[0 1]

Unique categories for column 'HadArthritis':  
[1 0]

Unique categories for column 'DeafOrHardOfHearing':  
[0 1]

Unique categories for column 'BlindOrVisionDifficulty':  
[0 1]

Unique categories for column 'DifficultyConcentrating':  
[0 1]

Unique categories for column 'DifficultyWalking':  
[0 1]

Unique categories for column 'DifficultyDressingBathing':  
[0 1]

Unique categories for column 'DifficultyErrands':  
[0 1]

Unique categories for column 'ChestScan':  
[0 1]

Unique categories for column 'HeightInMeters':

```
[0.46      0.58      0.62666667 0.52666667 0.42666667 0.48
 0.56      0.51333333 0.61333333 0.40666667 0.64666667 0.39333333
 0.54666667 0.49333333 0.59333333 0.44      0.66666667 0.37333333
 0.34      0.20666667 0.68      0.73333333 0.7      0.71333333
 0.36      0.29333333 0.56666667 0.74666667 0.83333333 0.4
 0.41333333 0.52      0.43333333 0.62      0.66      0.42
 0.54      0.64      0.55333333 0.32666667 0.48666667 0.44666667
 0.47333333 0.58666667 0.50666667 0.36666667 0.65333333 0.46666667
 0.26      0.30666667 0.81333333 0.76666667 0.8      0.
 0.9      0.84666667 0.57333333 0.96666667 0.45333333 0.63333333
 0.60666667 0.5      0.53333333 0.69333333 0.09333333 0.78
 0.38666667 0.31333333 0.6      0.35333333 0.38      0.18666667
 0.27333333 0.22      0.10666667 0.08666667 0.24      0.12666667
 0.67333333 0.19333333 0.88666667 0.14      0.08      0.04
 0.22666667 0.92      0.16666667 0.18      0.12      1.
 0.06      0.17333333 0.11333333 0.34666667 0.15333333 0.07333333
 0.72666667 0.74      0.02666667 0.95333333 0.86666667]
```

Unique categories for column 'WeightInKilograms':

```
[0.16468141 0.25384761 0.30531291 0.23671772 0.19383626 0.34819437
 0.22643222 0.17666856 0.18869351 0.13378711 0.35677822 0.33106447
 0.20242012 0.21955001 0.18181131 0.10119115 0.1680847 0.22813386
 0.28818302 0.09090565 0.15095481 0.14237096 0.10807336 0.25214596
 0.36192097 0.0857629 0.15439592 0.32248062 0.21096616 0.18695406
 0.15265646 0.20412176 0.19897901 0.29502742 0.11321611 0.12694271
 0.11151446 0.21440726 0.16124031 0.15953867 0.24015882 0.27959917
 0.07547741 0.12520325 0.30875402 0.27101531 0.11665721 0.31219512
 0.13892985 0.31389677 0.06519191 0.19553791 0.26757421 0.10293061
 0.27785971 0.18525241 0.24870486 0.16294196 0.08235961 0.25728871
 0.51113632 0.34993382 0.26243146 0.22983551 0.06349026 0.21784836
 0.20926451 0.24530157 0.39281528 0.14925317 0.24186047 0.17152581
 0.30017016 0.30187181 0.23841936 0.17496691 0.13722821 0.13038382
 0.07033466 0.28474192 0.22299111 0.23327661 0.20756287 0.27445642
 0.10637171 0.25902817 0.14581206 0.14751371 0.15609756 0.11491775
 0.19213462 0.25558707 0.13208546 0.08062016 0.37394593 0.44254112
 0.35507657 0.29846852 0.37220647 0.33620722 0.0720363 0.57976933
 0.14066931 0.33961051 0.40824352 0.19039516 0.09264511 0.35333711
 0.17326527 0.20582341 0.29672906 0.1200605 0.31045566 0.12864436
 0.08920401 0.24700321 0.09948951 0.21614672 0.11835886 0.1440726
 0.18355077 0.10463226 0.17840802 0.26072982 0.22469276 0.15779921
 0.36536207 0.13552656 0.07377576 0.25044432 0.26417092 0.23157497
 0.24356211 0.27615806 0.45970883 0.38249196 0.21270562 0.63119682
 0.26931367 0.32592172 0.09434676 0.20068066 0.18010966 0.36362261
 0.32074116 0.19727737 0.32762337 0.06689355 0.0960484 0.39965967
 0.08750236 0.49400643 0.12350161 0.06863301 0.16638306 0.39107582
 0.37734922 0.09778786 0.26587257 0.38763471 0.43569673 0.28130081]
```

0.32932501	0.23497826	0.52486292	0.30705237	0.04632256	0.40650407
0.66553224	0.38423142	0.37564757	0.41682738	0.29332577	0.34649272
0.07717905	0.42537342	0.29158631	0.42023067	0.37050482	0.40480242
0.28304027	0.35847986	0.109775	0.03947816	0.12179996	0.38937417
0.45626773	0.33446776	0.31903952	0.22128947	0.31733787	0.54547173
0.39795803	0.34305162	0.08406126	0.28644356	0.31559841	0.35163547
0.32418227	0.16982416	0.43395727	0.40136132	0.36021932	0.0342976
0.34475326	0.03089431	0.07891851	0.05660805	0.05146531	0.0274532
0.28988467	0.43910002	0.52830403	0.45796937	0.03603706	0.21886935
0.45112498	0.51972017	0.47683872	0.41164681	0.30361127	0.48198147
0.67407828	0.45452827	0.05316695	0.0617508	0.23399508	0.61920968
0.83705804	0.14702212	0.06004916	0.0480242	0.48542258	0.09408206
0.42197013	0.38593307	0.34134997	0.27180942	0.46825487	0.51801853
0.0583097	0.40310078	0.42881452	0.36706372	0.17727359	0.83902439
0.33276612	0.16971072	0.27271696	0.17349215	0.64995273	0.19618075
0.20374362	0.09786349	0.14324069	0.46999433	0.04288145	0.04976366
0.61406693	0.60548308	0.11677066	0.15836642	0.53688788	0.06761202
0.89574589	0.53174513	0.6121384	0.10164492	0.24533938	0.49740972
0.18861789	0.0377387	0.59349593	0.28693515	0.58831537	0.36880318
0.15080355	0.40037814	0.38525241	0.01546606	0.43739837	0.41508792
0.57118548	0.29071658	0.13945926	0.01467196	1.	0.21508792
0.16592929	0.55401777	0.08273776	0.37908867	0.12811496	0.22265078
0.56260163	0.04462091	0.18105502	0.38079032	0.11298922	0.48372093
0.35121951	0.05490641	0.23021365	0.44938552	0.44598223	0.54203063
0.75299679	0.00514275	0.61236529	0.50259028	0.75129514	0.39451692
0.41338627	0.13567782	0.10542636	0.42711288	0.09030062	0.13189639
0.34743808	0.20752505	0.10920779	0.0137266	0.18483645	0.25668368
0.12055209	0.32853091	0.04870486	0.46311212	0.92282095	0.07517489
0.04117981	0.25290225	0.63978068	0.49226697	0.06383059	0.24912082
0.80272263	0.07895632	0.49056532	0.30206088	0.59323123	0.12433352
0.02231046	0.39621857	0.21130649	0.76328228	0.41852902	0.19996219
0.19239932	0.24155795	0.29827945	0.48027983	0.68266213	0.15458499
0.31340518	0.30962375	0.46485158	0.68776706	0.42367177	0.02915485
0.44768387	0.33790887	0.27937228	0.36634524	0.46655322	0.03259595
0.03736056	0.40415958	0.16214785	0.45282662	0.47513708	0.53650974
0.23777652	0.26424655	0.62091133	0.60034033	0.33987521	0.49914918
0.49869541	0.53858953	0.65694838	0.59005483	0.55575723	0.43225562
0.5327283	0.52142182	0.57632823	0.63637739	0.50773303	0.04492343
0.54373227	0.63463793	0.40994517	0.47339762	0.49570807	0.51457742
0.72554358	0.43051617	0.54887502	0.0240121	0.69982984	0.9382492
0.64836453	0.70841369	0.71695973	0.55916052	0.65006618	0.76672339
0.02060881	0.59519758	0.28315372	0.47169597	0.27559085	0.50085082
0.46141047	0.48886368	0.44197391	0.46088107	0.51287578	0.61750804
0.55231613	0.51627907	0.86277179	0.67067499	0.62265078	0.50429193
0.56774438	0.57288712	0.44083948	0.66893553	0.56948383	0.47857818
0.52316128	0.56604273	0.65180563	0.77016449	0.79417659	0.75469843
0.9279637	0.69468709	0.73927018	0.64152014	0.60892418	0.59689922
0.84046133	0.58661373	0.58147098	0.73412743	0.76842503	0.57462658

0.66038949 0.57432407 0.51382114 0.48712422 0.32474948 0.35500095  
0.50599357 0.41172244 0.50943468 0.60718472 0.78044999 0.64666289  
0.66209113 0.44424277 0.55061448 0.88848554 0.47600681 0.79757988  
0.56430327 0.74441293 0.33609378 0.03357913 0.26046512 0.69124598  
0.92622424 0.66379278 0.79587824 0.32096805 0.74271129 0.53514842  
0.44575534 0.65524674 0.7255814 0.02575156 0.85418794 0.69809038  
0.56089998 0.59175648 0.36256381 0.46844394 0.77700889 0.62435243  
0.6688599 0.81989034 0.75984118 0.07139346 0.52660238 0.26802798  
0. 0.78559274 0.54029117 0.69638873 0.57802987 0.53344678  
0.55745888 0.0102855 0.33231235 0.53000567 0.35878238 0.59863868  
0.54717338 0.58491208 0.71525808 0.58317262 0.82503309 0.00858385]

Unique categories for column 'BMI':

[0.18650006 0.2114913 0.22935887 ... 0.31274086 0.54805559 0.38841528]

Unique categories for column 'AlcoholDrinkers':

[0 1]

Unique categories for column 'HIVTesting':

[0 1]

Unique categories for column 'FluVaxLast12':

[1 0]

Unique categories for column 'PneumoVaxEver':

[1 0]

Unique categories for column 'HighRiskLastYear':

[0 1]

Unique categories for column 'State\_Alabama':

[ True False]

Unique categories for column 'State\_Alaska':

[False True]

Unique categories for column 'State\_Arizona':

[False True]

Unique categories for column 'State\_Arkansas':

[False True]

Unique categories for column 'State\_California':

[False True]

Unique categories for column 'State\_Colorado':

[False True]



Unique categories for column 'State\_Connecticut':  
[False True]

Unique categories for column 'State\_Delaware':  
[False True]

Unique categories for column 'State\_District of Columbia':  
[False True]

Unique categories for column 'State\_Florida':  
[False True]

Unique categories for column 'State\_Georgia':  
[False True]

Unique categories for column 'State\_Guam':  
[False True]

Unique categories for column 'State\_Hawaii':  
[False True]

Unique categories for column 'State\_Idaho':  
[False True]

Unique categories for column 'State\_Illinois':  
[False True]

Unique categories for column 'State\_Indiana':  
[False True]

Unique categories for column 'State\_Iowa':  
[False True]

Unique categories for column 'State\_Kansas':  
[False True]

Unique categories for column 'State\_Kentucky':  
[False True]

Unique categories for column 'State\_Louisiana':  
[False True]

Unique categories for column 'State\_Maine':  
[False True]

Unique categories for column 'State\_Maryland':  
[False True]

Unique categories for column 'State\_Massachusetts':  
[False True]

Unique categories for column 'State\_Michigan':  
[False True]

Unique categories for column 'State\_Minnesota':  
[False True]

Unique categories for column 'State\_Mississippi':  
[False True]

Unique categories for column 'State\_Missouri':  
[False True]

Unique categories for column 'State\_Montana':  
[False True]

Unique categories for column 'State\_Nebraska':  
[False True]

Unique categories for column 'State\_Nevada':  
[False True]

Unique categories for column 'State\_New Hampshire':  
[False True]

Unique categories for column 'State\_New Jersey':  
[False True]

Unique categories for column 'State\_New Mexico':  
[False True]

Unique categories for column 'State\_New York':  
[False True]

Unique categories for column 'State\_North Carolina':  
[False True]

Unique categories for column 'State\_North Dakota':  
[False True]

Unique categories for column 'State\_Ohio':  
[False True]

Unique categories for column 'State\_Oklahoma':  
[False True]

Unique categories for column 'State\_Oregon':  
[False True]

Unique categories for column 'State\_Pennsylvania':  
[False True]

Unique categories for column 'State\_Puerto Rico':  
[False True]

Unique categories for column 'State\_Rhode Island':  
[False True]

Unique categories for column 'State\_South Carolina':  
[False True]

Unique categories for column 'State\_South Dakota':  
[False True]

Unique categories for column 'State\_Tennessee':  
[False True]

Unique categories for column 'State\_Texas':  
[False True]

Unique categories for column 'State\_Utah':  
[False True]

Unique categories for column 'State\_Vermont':  
[False True]

Unique categories for column 'State\_Virgin Islands':  
[False True]

Unique categories for column 'State\_Virginia':  
[False True]

Unique categories for column 'State\_Washington':  
[False True]

Unique categories for column 'State\_West Virginia':  
[False True]

Unique categories for column 'State\_Wisconsin':  
[False True]

Unique categories for column 'State\_Wyoming':  
[False True]

Unique categories for column 'Sex\_Female':  
[ True False]

Unique categories for column 'Sex\_Male':  
[False True]

Unique categories for column 'GeneralHealth\_Excellent':  
[False True]

Unique categories for column 'GeneralHealth\_Fair':  
[False True]

Unique categories for column 'GeneralHealth\_Good':  
[False True]

Unique categories for column 'GeneralHealth\_Poor':  
[False True]

Unique categories for column 'GeneralHealth\_Very good':  
[ True False]

Unique categories for column 'LastCheckupTime\_5 or more years ago':  
[False True]

Unique categories for column 'LastCheckupTime\_Within past 2 years (1 year but less than 2 years ago)':  
[False True]

Unique categories for column 'LastCheckupTime\_Within past 5 years (2 years but less than 5 years ago)':  
[False True]

Unique categories for column 'LastCheckupTime\_Within past year (anytime less than 12 months ago)':  
[ True False]

Unique categories for column 'RemovedTeeth\_1 to 5':  
[False True]

Unique categories for column 'RemovedTeeth\_6 or more, but not all':  
[False True]

Unique categories for column 'RemovedTeeth\_All':  
[False True]

Unique categories for column 'RemovedTeeth\_None of them':  
[ True False]

Unique categories for column 'HadDiabetes\_No':  
[ True False]

Unique categories for column 'HadDiabetes\_No, pre-diabetes or borderline diabetes':  
[False True]

Unique categories for column 'HadDiabetes\_Yes':  
[False True]

Unique categories for column 'HadDiabetes\_Yes, but only during pregnancy (female)':  
[False True]

Unique categories for column 'SmokerStatus\_Current smoker - now smokes every day':  
[False True]

Unique categories for column 'SmokerStatus\_Current smoker - now smokes some days':  
[False True]

Unique categories for column 'SmokerStatus\_Former smoker':  
[ True False]

Unique categories for column 'SmokerStatus\_Never smoked':  
[False True]

Unique categories for column 'ECigaretteUsage\_Never used e-cigarettes in my entire life':  
[ True False]

Unique categories for column 'ECigaretteUsage\_Not at all (right now)':  
[False True]

Unique categories for column 'ECigaretteUsage\_Use them every day':  
[False True]

Unique categories for column 'ECigaretteUsage\_Use them some days':  
[False True]

Unique categories for column 'RaceEthnicityCategory\_Black only, Non-Hispanic':  
[False True]

Unique categories for column 'RaceEthnicityCategory\_Hispanic':  
[False True]

Unique categories for column 'RaceEthnicityCategory\_Multiracial, Non-Hispanic':

[False True]

Unique categories for column 'RaceEthnicityCategory\_Other race only, Non-Hispanic':

[False True]

Unique categories for column 'RaceEthnicityCategory\_White only, Non-Hispanic':

[ True False]

Unique categories for column 'AgeCategory\_Age 18 to 24':

[False True]

Unique categories for column 'AgeCategory\_Age 25 to 29':

[False True]

Unique categories for column 'AgeCategory\_Age 30 to 34':

[False True]

Unique categories for column 'AgeCategory\_Age 35 to 39':

[False True]

Unique categories for column 'AgeCategory\_Age 40 to 44':

[False True]

Unique categories for column 'AgeCategory\_Age 45 to 49':

[False True]

Unique categories for column 'AgeCategory\_Age 50 to 54':

[False True]

Unique categories for column 'AgeCategory\_Age 55 to 59':

[False True]

Unique categories for column 'AgeCategory\_Age 60 to 64':

[False True]

Unique categories for column 'AgeCategory\_Age 65 to 69':

[ True False]

Unique categories for column 'AgeCategory\_Age 70 to 74':

[False True]

Unique categories for column 'AgeCategory\_Age 75 to 79':

[False True]

Unique categories for column 'AgeCategory\_Age 80 or older':

[False True]

Unique categories for column 'TetanusLast10Tdap\_No, did not receive any tetanus shot in the past 10 years':  
[False True]

Unique categories for column 'TetanusLast10Tdap\_Yes, received Tdap':  
[ True False]

Unique categories for column 'TetanusLast10Tdap\_Yes, received tetanus shot but not sure what type':  
[False True]

Unique categories for column 'TetanusLast10Tdap\_Yes, received tetanus shot, but not Tdap':  
[False True]

Unique categories for column 'CovidPos\_No':  
[ True False]

Unique categories for column 'CovidPos\_Test positive using home test without a health professional':  
[False True]

Unique categories for column 'CovidPos\_Yes':  
[False True]

```
[34]: # Display the DataFrame and print dimensions
      # Note increase in total column/variable count due to one-hot encoding
      # Note that all column/variables are normalized, binary, or boolean
      print("Number of rows:", encoded_df.shape[0])
      print("Number of columns:", encoded_df.shape[1])
      encoded_df.head()
      encoded_df.tail()
```

Number of rows: 246022

Number of columns: 134

```
[34]:
```

	PhysicalHealthDays	MentalHealthDays	PhysicalActivities	SleepHours	\
342	0.133333	0.0	1	0.347826	
343	0.000000	0.0	1	0.217391	
345	0.000000	0.0	0	0.304348	
346	0.166667	0.0	1	0.347826	
347	0.100000	0.5	1	0.173913	

	HadHeartAttack	HadAngina	HadStroke	HadAsthma	HadSkinCancer	HadCOPD	\
342	0	0	0	0	0	0	
343	0	0	0	0	0	0	
345	0	0	0	0	0	0	

346	0	0	0	0	1	0
347	0	0	0	0	0	0

	...	AgeCategory_Age 70 to 74	AgeCategory_Age 75 to 79	\
342	...	False	False	
343	...	True	False	
345	...	False	True	
346	...	False	False	
347	...	False	False	

		AgeCategory_Age 80 or older	\
342		False	
343		False	
345		False	
346		True	
347		True	

		TetanusLast10Tdap_No, did not receive any tetanus shot in the past 10 years
	\	
342		False
343		False
345		True
346		True
347		True

		TetanusLast10Tdap_Yes, received Tdap	\
342		True	
343		False	
345		False	
346		False	
347		False	

		TetanusLast10Tdap_Yes, received tetanus shot but not sure what type	\
342		False	
343		True	
345		False	
346		False	
347		False	

		TetanusLast10Tdap_Yes, received tetanus shot, but not Tdap	CovidPos_No	\
342		False	True	
343		False	True	
345		False	False	
346		False	False	
347		False	True	

		CovidPos_Test	positive using home test without a health professional	\
--	--	---------------	--	---



342	False
343	False
345	False
346	False
347	False

	CovidPos_Yes
342	False
343	False
345	True
346	True
347	False

[5 rows x 134 columns]

[34]:

	PhysicalHealthDays	MentalHealthDays	PhysicalActivities	SleepHours	\
445117	0.000000	0.000000	1	0.217391	
445123	0.000000	0.233333	1	0.260870	
445124	0.000000	0.500000	1	0.260870	
445128	0.066667	0.066667	1	0.260870	
445130	0.000000	0.000000	0	0.173913	

	HadHeartAttack	HadAngina	HadStroke	HadAsthma	HadSkinCancer	\
445117	0	0	0	0	0	
445123	0	0	0	0	0	
445124	0	0	1	0	0	
445128	0	0	0	0	0	
445130	1	0	0	1	0	

	HadCOPD ...	AgeCategory_Age 70 to 74	AgeCategory_Age 75 to 79	\
445117	0 ...	False	False	
445123	0 ...	False	False	
445124	0 ...	False	False	
445128	0 ...	False	False	
445130	0 ...	True	False	

	AgeCategory_Age 80 or older	\
445117	False	
445123	False	
445124	False	
445128	False	
445130	False	

	TetanusLast10Tdap_No, did not receive any tetanus shot in the past 10 years	\
445117	False	
445123	True	

445124	False
445128	False
445130	True

	TetanusLast10Tdap_Yes, received Tdap \
445117	False
445123	False
445124	False
445128	False
445130	False

	TetanusLast10Tdap_Yes, received tetanus shot but not sure what type \
445117	True
445123	False
445124	True
445128	True
445130	False

	TetanusLast10Tdap_Yes, received tetanus shot, but not Tdap \
445117	False
445123	False
445124	False
445128	False
445130	False

	CovidPos_No \
445117	True
445123	False
445124	False
445128	True
445130	False

	CovidPos_Test positive using home test without a health professional \
445117	False
445123	False
445124	False
445128	False
445130	False

	CovidPos_Yes
445117	False
445123	True
445124	True
445128	False
445130	True

[5 rows x 134 columns]

```
[35]: # Begin analysis of correlation values to select best predictor variables
```

```
[36]: # Determine correlation matrix
# Set pandas display options to show all columns
pd.set_option('display.max_rows', None)

# Calculate correlation between the selected variable and all other variables
correlation_with_HadHeartAttack_variable = encoded_df.corr()['HadHeartAttack'].
    ↪sort_values(ascending=False)

# Print all correlation values
print("Correlation with selected variable:")
print(correlation_with_HadHeartAttack_variable)
```

Correlation with selected variable:

HadHeartAttack

1.000000

HadAngina

0.445903

HadStroke

0.177137

ChestScan

0.167760

DifficultyWalking

0.159878

HadDiabetes\_Yes

0.145868

GeneralHealth\_Poor

0.140607

PhysicalHealthDays

0.133420

HadCOPD

0.133223

RemovedTeeth\_All

0.120564

PneumoVaxEver

0.119955

HadArthritis

0.117773

GeneralHealth\_Fair

0.112319

HadKidneyDisease

0.109355

AgeCategory\_Age 80 or older

0.100296

DeafOrHardOfHearing

0.097662  
 RemovedTeeth\_6 or more, but not all  
 0.092477  
 DifficultyErrands  
 0.089495  
 DifficultyDressingBathing  
 0.083090  
 SmokerStatus\_Former smoker  
 0.074537  
 AgeCategory\_Age 75 to 79  
 0.073567  
 Sex\_Male  
 0.073316  
 BlindOrVisionDifficulty  
 0.072964  
 LastCheckupTime\_Within past year (anytime less than 12 months ago)  
 0.070725  
 AgeCategory\_Age 70 to 74  
 0.058590  
 DifficultyConcentrating  
 0.051663  
 HadSkinCancer  
 0.049408  
 FluVaxLast12  
 0.045235  
 SmokerStatus\_Current smoker - now smokes every day  
 0.039031  
 WeightInKilograms  
 0.038436  
 AgeCategory\_Age 65 to 69  
 0.033260  
 BMI  
 0.030413  
 MentalHealthDays  
 0.025892  
 CovidPos\_No  
 0.024529  
 RaceEthnicityCategory\_White only, Non-Hispanic  
 0.024221  
 HadAsthma  
 0.023756  
 HadDepressiveDisorder  
 0.023706  
 HeightInMeters  
 0.023059  
 TetanusLast10Tdap\_Yes, received tetanus shot but not sure what type  
 0.021735  
 State\_Florida

0.016592  
 GeneralHealth\_Good  
 0.014322  
 State\_Arkansas  
 0.013738  
 State\_West Virginia  
 0.013684  
 HadDiabetes\_No, pre-diabetes or borderline diabetes  
 0.011919  
 TetanusLast10Tdap\_No, did not receive any tetanus shot in the past 10 years  
 0.011883  
 State\_Maine  
 0.011196  
 SmokerStatus\_Current smoker - now smokes some days  
 0.011101  
 RemovedTeeth\_1 to 5  
 0.010878  
 TetanusLast10Tdap\_Yes, received tetanus shot, but not Tdap  
 0.009777  
 State\_Ohio  
 0.009321  
 State\_Nebraska  
 0.008170  
 ECigaretteUsage\_Never used e-cigarettes in my entire life  
 0.008082  
 State\_Arizona  
 0.007373  
 State\_South Dakota  
 0.007210  
 AgeCategory\_Age 60 to 64  
 0.006661  
 State\_New Hampshire  
 0.006257  
 State\_Tennessee  
 0.004819  
 State\_Kentucky  
 0.004761  
 State\_Indiana  
 0.004663  
 State\_Oklahoma  
 0.004509  
 State\_New Mexico  
 0.004247  
 RaceEthnicityCategory\_Multiracial, Non-Hispanic  
 0.004232  
 State\_Alabama  
 0.004112  
 SleepHours

0.003631  
ECigaretteUsage\_Not at all (right now)  
0.003358  
State\_Nevada  
0.003048  
State\_Louisiana  
0.002380  
State\_Texas  
0.002246  
State\_Missouri  
0.001923  
State\_Michigan  
0.000949  
State\_Montana  
0.000843  
State\_Virginia  
0.000614  
State\_Georgia  
0.000528  
State\_North Dakota  
0.000283  
State\_Mississippi  
-0.000025  
State\_South Carolina  
-0.000093  
State\_Maryland  
-0.000424  
State\_Kansas  
-0.000524  
State\_Rhode Island  
-0.000631  
State\_Wisconsin  
-0.000865  
State\_Guam  
-0.001264  
State\_Alaska  
-0.001581  
State\_Vermont  
-0.001748  
State\_Delaware  
-0.001859  
State\_Wyoming  
-0.001927  
State\_Pennsylvania  
-0.002054  
State\_North Carolina  
-0.002173  
State\_Idaho

-0.003548  
State\_Oregon  
-0.003800  
State\_Puerto Rico  
-0.003878  
State\_Iowa  
-0.004618  
State\_Virgin Islands  
-0.005078  
State\_Connecticut  
-0.005794  
State\_Illinois  
-0.006004  
RaceEthnicityCategory\_Other race only, Non-Hispanic  
-0.006220  
State\_Massachusetts  
-0.006244  
AgeCategory\_Age 55 to 59  
-0.006342  
State\_Hawaii  
-0.006790  
State\_New York  
-0.006820  
State\_District of Columbia  
-0.007547  
State\_New Jersey  
-0.007760  
State\_California  
-0.008075  
State\_Utah  
-0.008129  
State\_Washington  
-0.008832  
State\_Colorado  
-0.008955  
State\_Minnesota  
-0.009852  
HadDiabetes\_Yes, but only during pregnancy (female)  
-0.010461  
RaceEthnicityCategory\_Black only, Non-Hispanic  
-0.011076  
ECigaretteUsage\_Use them some days  
-0.012412  
HIVTesting  
-0.014563  
CovidPos\_Yes  
-0.016444  
ECigaretteUsage\_Use them every day

-0.017250  
HighRiskLastYear  
-0.021127  
CovidPos\_Tested positive using home test without a health professional  
-0.022104  
RaceEthnicityCategory\_Hispanic  
-0.023148  
AgeCategory\_Age 50 to 54  
-0.025214  
LastCheckupTime\_5 or more years ago  
-0.035137  
AgeCategory\_Age 45 to 49  
-0.035142  
LastCheckupTime\_Within past 5 years (2 years but less than 5 years ago)  
-0.037198  
TetanusLast10Tdap\_Yes, received Tdap  
-0.040362  
LastCheckupTime\_Within past 2 years (1 year but less than 2 years ago)  
-0.041811  
AgeCategory\_Age 25 to 29  
-0.048216  
AgeCategory\_Age 40 to 44  
-0.049331  
AgeCategory\_Age 30 to 34  
-0.050453  
AgeCategory\_Age 35 to 39  
-0.051119  
AgeCategory\_Age 18 to 24  
-0.053068  
Sex\_Female  
-0.073316  
AlcoholDrinkers  
-0.074181  
GeneralHealth\_Excellent  
-0.079933  
PhysicalActivities  
-0.083187  
GeneralHealth\_Very good  
-0.085347  
SmokerStatus\_Never smoked  
-0.094843  
RemovedTeeth\_None of them  
-0.122556  
HadDiabetes\_No  
-0.136692  
Name: HadHeartAttack, dtype: float64



```
[37]: # Find high correlation values based on the threshold of > +0.07 or < -0.07
correlations_above_threshold = correlation_with_HadHeartAttack_variable > +0.07
correlations_below_threshold = correlation_with_HadHeartAttack_variable < -0.07
high_correlation_variables = ␣
    ↪correlation_with_HadHeartAttack_variable[(correlations_above_threshold) | ␣
    ↪(correlations_below_threshold)]

# Sort the high correlation values
high_correlation_variables = high_correlation_variables.
    ↪sort_values(ascending=False)

# Print high correlation values
print("Variables with correlation greater than +0.7 or less than -0.7:")
print(high_correlation_variables)
```

```
Variables with correlation greater than +0.7 or less than -0.7:
HadHeartAttack                                1.000000
HadAngina                                      0.445903
HadStroke                                      0.177137
ChestScan                                     0.167760
DifficultyWalking                             0.159878
HadDiabetes_Yes                               0.145868
GeneralHealth_Poor                            0.140607
PhysicalHealthDays                           0.133420
HadCOPD                                       0.133223
RemovedTeeth_All                             0.120564
PneumoVaxEver                                0.119955
HadArthritis                                  0.117773
GeneralHealth_Fair                           0.112319
HadKidneyDisease                             0.109355
AgeCategory_Age 80 or older                   0.100296
DeafOrHardOfHearing                          0.097662
RemovedTeeth_6 or more, but not all           0.092477
DifficultyErrands                             0.089495
DifficultyDressingBathing                     0.083090
SmokerStatus_Former smoker                   0.074537
AgeCategory_Age 75 to 79                     0.073567
Sex_Male                                      0.073316
BlindOrVisionDifficulty                       0.072964
LastCheckupTime_Within past year (anytime less than 12 months ago) 0.070725
Sex_Female                                    -0.073316
AlcoholDrinkers                              -0.074181
GeneralHealth_Excellent                      -0.079933
PhysicalActivities                           -0.083187
GeneralHealth_Very good                      -0.085347
SmokerStatus_Never smoked                    -0.094843
RemovedTeeth_None of them                    -0.122556
HadDiabetes_No                               -0.136692
```

Name: HadHeartAttack, dtype: float64

```
[38]: # Create empty list
high_correlation_variable_list = []
# Convert first column from 'high_correlation_variables' series into a list
for i in range(0, len(high_correlation_variables)):
    high_correlation_variable_list.append(high_correlation_variables.index[i])

print(high_correlation_variable_list)
```

```
['HadHeartAttack', 'HadAngina', 'HadStroke', 'ChestScan', 'DifficultyWalking',
'HadDiabetes_Yes', 'GeneralHealth_Poor', 'PhysicalHealthDays', 'HadCOPD',
'RemovedTeeth_All', 'PneumoVaxEver', 'HadArthritis', 'GeneralHealth_Fair',
'HadKidneyDisease', 'AgeCategory_Age 80 or older', 'DeafOrHardOfHearing',
'RemovedTeeth_6 or more, but not all', 'DifficultyErrands',
'DifficultyDressingBathing', 'SmokerStatus_Former smoker', 'AgeCategory_Age 75
to 79', 'Sex_Male', 'BlindOrVisionDifficulty', 'LastCheckupTime_Within past year
(anytime less than 12 months ago)', 'Sex_Female', 'AlcoholDrinkers',
'GeneralHealth_Excellent', 'PhysicalActivities', 'GeneralHealth_Very good',
'SmokerStatus_Never smoked', 'RemovedTeeth_None of them', 'HadDiabetes_No']
```

```
[39]: # Create smaller data frame consisting only of high correlation variables from
#      original larger data frame
high_corr_encoded_df = encoded_df[high_correlation_variable_list]

# Display original larger data frame with its dimensions
print("LARGER DATA FRAME CONSISTING OF ALL VARIABLES\n")
print("Number of rows:", encoded_df.shape[0])
print("Number of columns:", encoded_df.shape[1])
encoded_df.head()
encoded_df.tail()

# Display new data smaller frame with its dimensions
print("\n\nSMALLER DATA FRAME CONSISTING OF ONLY HIGH CORRELATION VARIABLES\n")
print("Number of rows:", high_corr_encoded_df.shape[0])
print("Number of columns:", high_corr_encoded_df.shape[1])
high_corr_encoded_df.head()
high_corr_encoded_df.tail()
```

LARGER DATA FRAME CONSISTING OF ALL VARIABLES

Number of rows: 246022

Number of columns: 134

```
[39]:      PhysicalHealthDays  MentalHealthDays  PhysicalActivities  SleepHours  \
342          0.133333          0.0          1          0.347826
343          0.000000          0.0          1          0.217391
345          0.000000          0.0          0          0.304348
```

346	0.166667	0.0	1	0.347826
347	0.100000	0.5	1	0.173913

	HadHeartAttack	HadAngina	HadStroke	HadAsthma	HadSkinCancer	HadCOPD	\
342	0	0	0	0	0	0	
343	0	0	0	0	0	0	
345	0	0	0	0	0	0	
346	0	0	0	0	1	0	
347	0	0	0	0	0	0	

...	AgeCategory_Age 70 to 74	AgeCategory_Age 75 to 79	\
342	...	False	False
343	...	True	False
345	...	False	True
346	...	False	False
347	...	False	False

	AgeCategory_Age 80 or older	\
342	False	
343	False	
345	False	
346	True	
347	True	

	TetanusLast10Tdap_No, did not receive any tetanus shot in the past 10 years	\
342	False	
343	False	
345	True	
346	True	
347	True	

	TetanusLast10Tdap_Yes, received Tdap	\
342	True	
343	False	
345	False	
346	False	
347	False	

	TetanusLast10Tdap_Yes, received tetanus shot but not sure what type	\
342	False	
343	True	
345	False	
346	False	
347	False	

	TetanusLast10Tdap_Yes, received tetanus shot, but not Tdap	CovidPos_No	\
--	--	-------------	---

342	False	True
343	False	True
345	False	False
346	False	False
347	False	True

	CovidPos_Tested positive using home test without a health professional \
342	False
343	False
345	False
346	False
347	False

	CovidPos_Yes
342	False
343	False
345	True
346	True
347	False

[5 rows x 134 columns]

[39]:

	PhysicalHealthDays	MentalHealthDays	PhysicalActivities	SleepHours	\
445117	0.000000	0.000000	1	0.217391	
445123	0.000000	0.233333	1	0.260870	
445124	0.000000	0.500000	1	0.260870	
445128	0.066667	0.066667	1	0.260870	
445130	0.000000	0.000000	0	0.173913	

	HadHeartAttack	HadAngina	HadStroke	HadAsthma	HadSkinCancer	\
445117	0	0	0	0	0	
445123	0	0	0	0	0	
445124	0	0	1	0	0	
445128	0	0	0	0	0	
445130	1	0	0	1	0	

	HadCOPD ...	AgeCategory_Age 70 to 74	AgeCategory_Age 75 to 79	\
445117	0 ...	False	False	
445123	0 ...	False	False	
445124	0 ...	False	False	
445128	0 ...	False	False	
445130	0 ...	True	False	

	AgeCategory_Age 80 or older \
445117	False
445123	False
445124	False

445128	False
445130	False

TetanusLast10Tdap\_No, did not receive any tetanus shot in the past 10 years \

445117	False
445123	True
445124	False
445128	False
445130	True

TetanusLast10Tdap\_Yes, received Tdap \

445117	False
445123	False
445124	False
445128	False
445130	False

TetanusLast10Tdap\_Yes, received tetanus shot but not sure what type \

445117	True
445123	False
445124	True
445128	True
445130	False

TetanusLast10Tdap\_Yes, received tetanus shot, but not Tdap \

445117	False
445123	False
445124	False
445128	False
445130	False

CovidPos\_No \

445117	True
445123	False
445124	False
445128	True
445130	False

CovidPos\_Testeds positive using home test without a health professional \

445117	False
445123	False
445124	False
445128	False
445130	False

	CovidPos_Yes
445117	False
445123	True
445124	True
445128	False
445130	True

[5 rows x 134 columns]

SMALLER DATA FRAME CONSISTING OF ONLY HIGH CORRELATION VARIABLES

Number of rows: 246022

Number of columns: 32

```
[39]: HadHeartAttack HadAngina HadStroke ChestScan DifficultyWalking \
342          0          0          0          0          0
343          0          0          0          0          0
345          0          0          0          1          1
346          0          0          0          0          1
347          0          0          0          0          0
```

	HadDiabetes_Yes	GeneralHealth_Poor	PhysicalHealthDays	HadCOPD	\
342	False	False	0.133333	0	
343	True	False	0.000000	0	
345	False	False	0.000000	0	
346	False	False	0.166667	0	
347	False	False	0.100000	0	

	RemovedTeeth_All	...	BlindOrVisionDifficulty	\
342	False	...	0	
343	False	...	0	
345	False	...	1	
346	False	...	0	
347	False	...	0	

	LastCheckupTime_Within past year (anytime less than 12 months ago)	\
342	True	
343	True	
345	True	
346	True	
347	True	

	Sex_Female	AlcoholDrinkers	GeneralHealth_Excellent	PhysicalActivities	\
342	True	0	False	1	
343	False	0	False	1	
345	False	1	False	0	

346	True	0	False	1
347	True	0	False	1

	GeneralHealth_Very good	SmokerStatus_Never smoked \
342	True	False
343	True	False
345	True	False
346	False	True
347	False	True

	RemovedTeeth_None of them	HadDiabetes_No
342	True	True
343	True	False
345	False	True
346	True	True
347	False	True

[5 rows x 32 columns]

[39]:

	HadHeartAttack	HadAngina	HadStroke	ChestScan	DifficultyWalking \
445117	0	0	0	0	0
445123	0	0	0	0	0
445124	0	0	1	0	0
445128	0	0	0	0	0
445130	1	0	0	1	0

	HadDiabetes_Yes	GeneralHealth_Poor	PhysicalHealthDays	HadCOPD \
445117	False	False	0.000000	0
445123	False	False	0.000000	0
445124	True	False	0.000000	0
445128	False	False	0.066667	0
445130	False	False	0.000000	0

	RemovedTeeth_All ...	BlindOrVisionDifficulty \
445117	False ...	0
445123	False ...	0
445124	False ...	0
445128	False ...	0
445130	False ...	0

	LastCheckupTime_Within past year (anytime less than 12 months ago) \
445117	False
445123	True
445124	True
445128	True
445130	True

	Sex_Female	AlcoholDrinkers	GeneralHealth_Excellent	\
445117	False	1	False	
445123	True	0	False	
445124	False	1	False	
445128	True	0	True	
445130	False	0	False	

	PhysicalActivities	GeneralHealth_Very good	\
445117	1	True	
445123	1	False	
445124	1	False	
445128	1	False	
445130	0	True	

	SmokerStatus_Never smoked	RemovedTeeth_None of them	HadDiabetes_No
445117	True	True	True
445123	True	True	True
445124	True	False	False
445128	True	True	True
445130	True	True	True

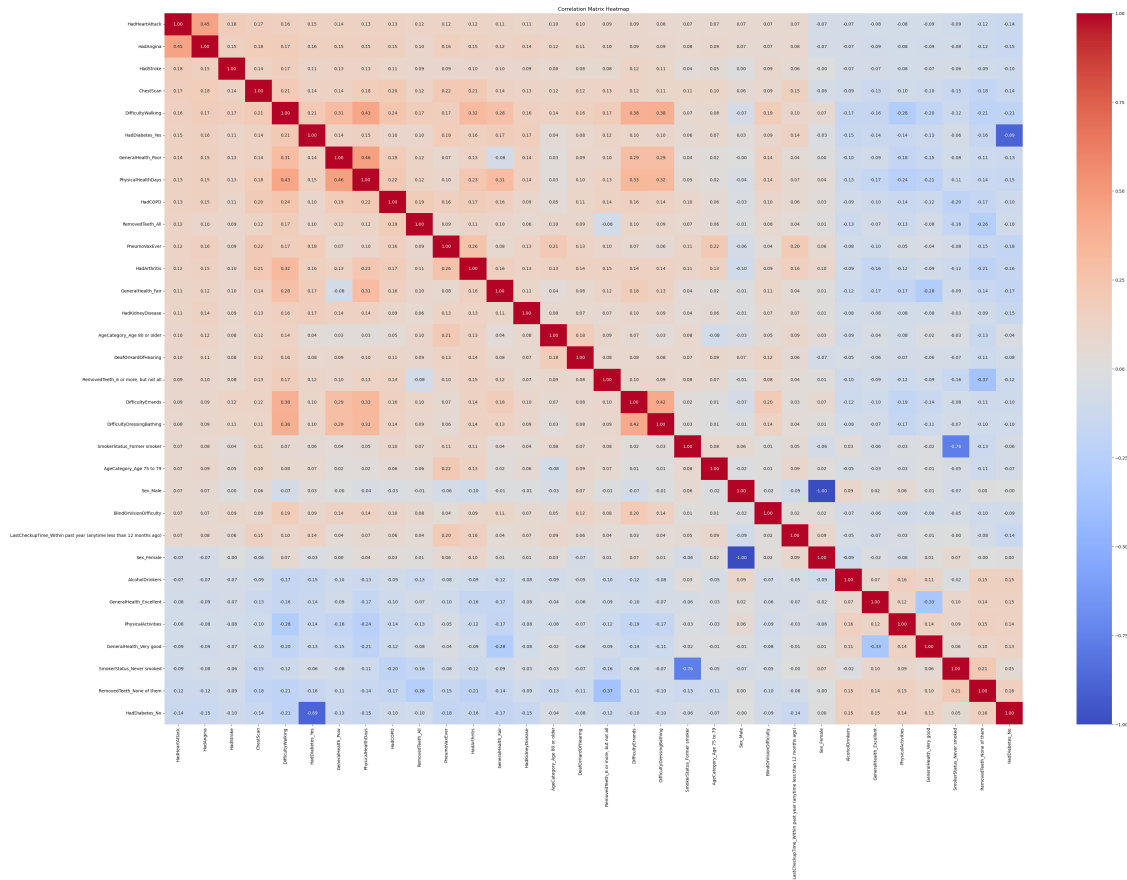
[5 rows x 32 columns]

```
[40]: # COMMENT:
#   Focusing only on high correlaton variables, drops the total
#   variable count from 134 to 32
```

```
[41]: # Correlation matrix of only high correlaton variables and
#   dependent variables 'HadHeartAttack' using smaller data frame,
#   ↪ 'high_corr_encoded_df'
correlation_matrix = high_corr_encoded_df[high_correlation_variable_list].corr()
# Create heatmap
plt.figure(figsize=(45, 30))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f");
plt.title('Correlation Matrix Heatmap');
plt.show();

# NOTE: Double click on image to zoom or right click to open in new tab (better)
```





[42]: #####  
 ## III. MODEL BUILDING AND EVALUATION:  
 #####

[43]: # LOGISTIC REGRESSION

```
# Isolate independent variables
independent_variables_list = [x for x in high_correlation_variable_list if x != 'HadHeartAttack']

independent_variables = high_corr_encoded_df[independent_variables_list]
# Isolate dependent variables
dependent_variable = high_corr_encoded_df['HadHeartAttack']

# Split data into train and test sets
X_train1, X_test1, y_train1, y_test1 = train_test_split(independent_variables,
  dependent_variable, test_size=0.2, random_state=42)

# Initialize logistic regression model
log_reg = LogisticRegression()
```

```

# Fit logistic regression model
log_reg.fit(X_train1, y_train1)

# Create a list of tuples containing coefficients and variables
coefficients_with_variables = [(coefficient, variable) for coefficient, variable in zip(log_reg.coef_[0], independent_variables)]
# Sort the list based on the absolute value of coefficients in descending order
coefficients_with_variables.sort(key=lambda x: x[0], reverse=True)

# Evaluate and print model accuracy
accuracy_lr1 = log_reg.score(X_test1, y_test1)
print(f"\nACCURACY:\t{accuracy_lr1:.5f}")

# Print model coefficients for each independent variable in descending order
print("\nMODEL INTERCEPT AND COEFFICIENTS IN DESCENDING ORDER:\n")

# Print model coefficients
print(f"INTERCEPT:\t{log_reg.intercept_[0]}")
print(f"COEFFICIENT:\tVARIABLE:\n")
for coefficient, variable in coefficients_with_variables:
    print(f"{coefficient:.5f}:\t{variable}")

```

[43]: LogisticRegression()

ACCURACY: 0.94899

MODEL INTERCEPT AND COEFFICIENTS IN DESCENDING ORDER:

INTERCEPT: -3.850075594149293

COEFFICIENT: VARIABLE:

2.50148:	HadAngina
0.90330:	HadStroke
0.61369:	ChestScan
0.47206:	AgeCategory_Age 80 or older
0.42342:	RemovedTeeth_All
0.35085:	LastCheckupTime_Within past year (anytime less than 12 months ago)
0.33718:	GeneralHealth_Poor
0.32893:	Sex_Male
0.32789:	AgeCategory_Age 75 to 79
0.21815:	GeneralHealth_Fair
0.18364:	HadDiabetes_Yes
0.17069:	RemovedTeeth_6 or more, but not all

```

0.16947:      PneumoVaxEver
0.16527:      BlindOrVisionDifficulty
0.13044:      HadArthritis
0.09380:      DifficultyWalking
0.07365:      DeafOrHardOfHearing
0.06801:      HadCOPD
0.04120:      HadKidneyDisease
0.03421:      DifficultyErrands
-0.02655:      PhysicalHealthDays
-0.07039:      DifficultyDressingBathing
-0.07605:      PhysicalActivities
-0.13161:      SmokerStatus_Former smoker
-0.15674:      HadDiabetes_No
-0.19865:      AlcoholDrinkers
-0.21184:      RemovedTeeth_None of them
-0.29520:      GeneralHealth_Very good
-0.30845:      Sex_Female
-0.39795:      SmokerStatus_Never smoked
-0.63267:      GeneralHealth_Excellent

```

```

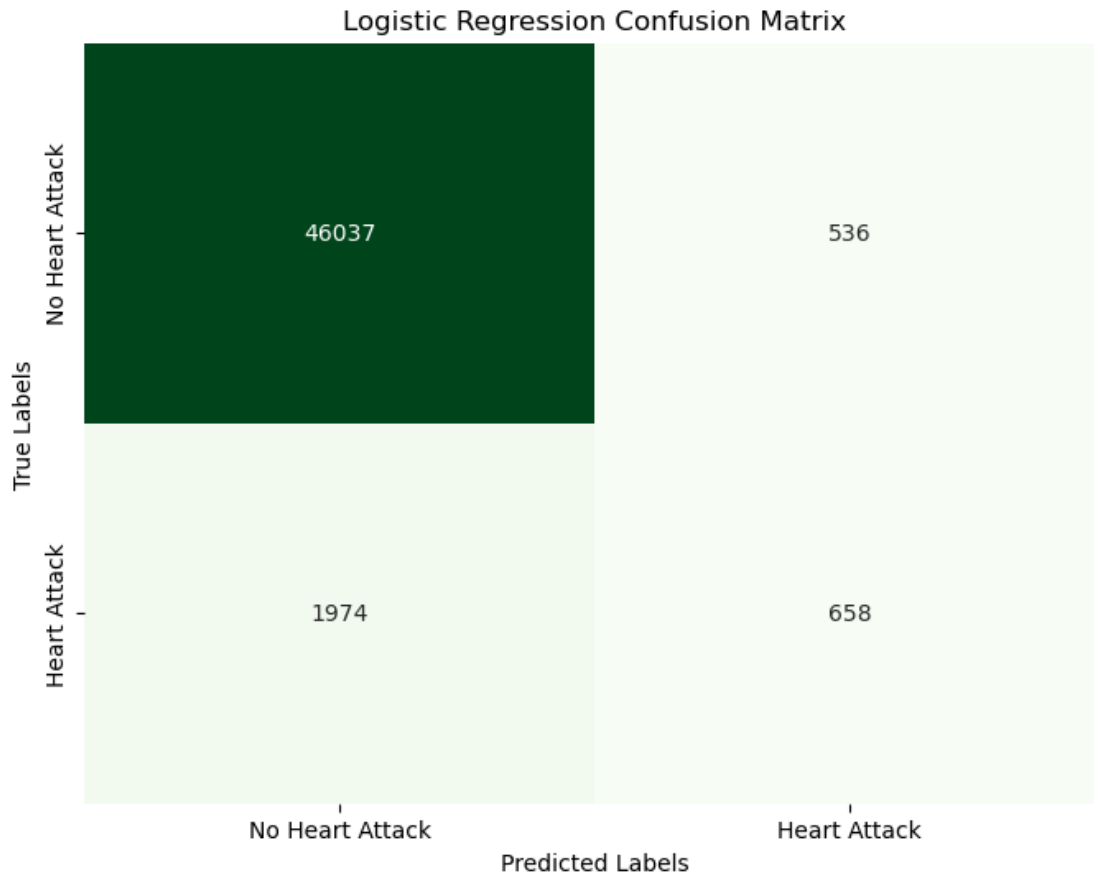
[44]: # Define custom labels
label_names = ['No Heart Attack', 'Heart Attack'];

# Calculate the confusion matrix
# Predict the labels for the test set
y_pred1 = log_reg.predict(X_test1)
conf_matrix1 = confusion_matrix(y_test1, y_pred1);

# Create a heatmap of the confusion matrix
plt.figure(figsize=(8, 6));
sns.heatmap(conf_matrix1, annot=True, fmt='d', cmap='Greens', cbar=False);
plt.title('Logistic Regression Confusion Matrix');
plt.xlabel('Predicted Labels');
plt.ylabel('True Labels');

# Set custom labels for ticks
plt.xticks(ticks=[0.5, 1.5], labels=label_names);
plt.yticks(ticks=[0.5, 1.5], labels=label_names);
plt.show();

```



```
[45]: # SUPPORT VECTOR MACHINE
# Using the same dependent_variable and independent_variables defined above

# Split data into train and test sets
X_train2, X_test2, y_train2, y_test2 = train_test_split(independent_variables,
    ↪ dependent_variable, test_size=0.2, random_state=42)

# Initialize SVM classifier
svm_classifier = SVC(kernel='linear') # Linear kernel for binary classification

# Fit SVM classifier
svm_classifier.fit(X_train2, y_train2)

# Evaluate model performance (optional)
accuracy_svm = svm_classifier.score(X_test2, y_test2)
print(f"\nACCURACY:\t{accuracy_svm:.5f}")
```

```
[45]: SVC(kernel='linear')
```

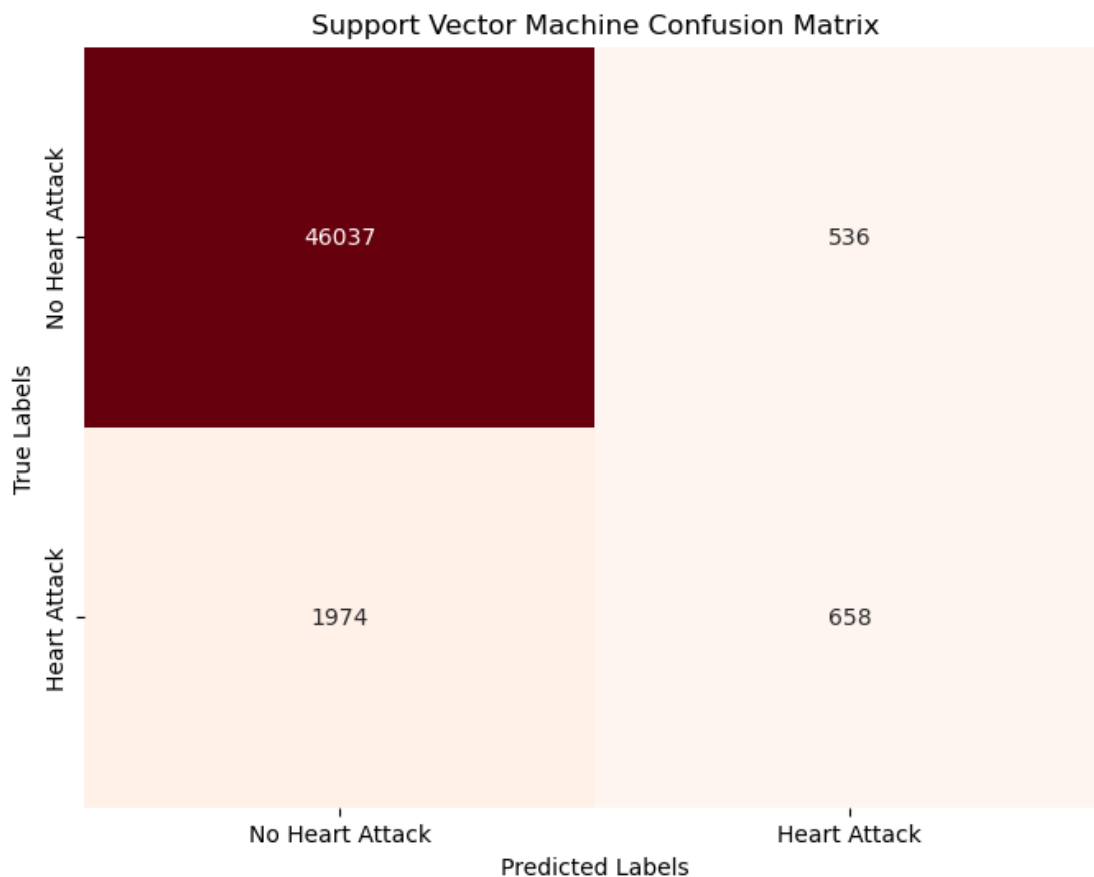
ACCURACY: 0.94651

```
[46]: # Define custom labels
label_names = ['No Heart Attack', 'Heart Attack'];

# Calculate the confusion matrix
# Predict the labels for the test set
y_pred2 = log_reg.predict(X_test2)
conf_matrix2 = confusion_matrix(y_test2, y_pred2);

# Create a heatmap of the confusion matrix
plt.figure(figsize=(8, 6));
sns.heatmap(conf_matrix2, annot=True, fmt='d', cmap='Reds', cbar=False);
plt.title('Support Vector Machine Confusion Matrix');
plt.xlabel('Predicted Labels');
plt.ylabel('True Labels');

# Set custom labels for ticks
plt.xticks(ticks=[0.5, 1.5], labels=label_names);
plt.yticks(ticks=[0.5, 1.5], labels=label_names);
plt.show();
```



```
[47]: # COMMENT:
# As shown above, the accuracy of the SVM model is slightly less than the
# accuracy of the logistic regression (LR) model. The LR model took about 10
# seconds
# to run. The SVM model took about 4 minutes to run. These ratios between
# runtime
# and accuracy suggest that the LR model is the better choice. As a result,
# for
# this simple classification problem, only the LR model will be used going
# forward.
```

```
[48]: #####
## IV. ITERATIVE PROCESS:
#####
# A possible interaction term will be considered to capture the interaction
# between
# diabetes (DM) and kidney disease (CKD),
# ('HadDiabetes_Yes' * 'HadKidneyDisease') = 'Had_DM+_CKD'.
# This is meant to capture diabetic nephropathy or diabetic kidney disease
# which is associated
# with increased risk of cardiovascular disease:
# https://en.wikipedia.org/wiki/Diabetic\_nephropathy
# https://www.sciencedirect.com/science/article/pii/S1548559514000512
# https://www.sciencedirect.com/science/article/abs/pii/S027092951830024X
#
```

```
[49]: # Display new data smaller frame with its dimensions
print("\n\nSMALLER DATA FRAME CONSISTING OF ONLY HIGH CORRELATION VARIABLES\n")
print("Number of rows:", high_corr_encoded_df.shape[0])
print("Number of columns:", high_corr_encoded_df.shape[1])
high_corr_encoded_df.head()
high_corr_encoded_df.tail()
```

SMALLER DATA FRAME CONSISTING OF ONLY HIGH CORRELATION VARIABLES

Number of rows: 246022

Number of columns: 32

```
[49]: HadHeartAttack HadAngina HadStroke ChestScan DifficultyWalking \
342 0 0 0 0 0
343 0 0 0 0 0
345 0 0 0 1 1
346 0 0 0 0 1
```

347	0	0	0	0	0
-----	---	---	---	---	---

	HadDiabetes_Yes	GeneralHealth_Poor	PhysicalHealthDays	HadCOPD	\
342	False	False	0.133333	0	
343	True	False	0.000000	0	
345	False	False	0.000000	0	
346	False	False	0.166667	0	
347	False	False	0.100000	0	

	RemovedTeeth_All	...	BlindOrVisionDifficulty	\
342	False	...	0	
343	False	...	0	
345	False	...	1	
346	False	...	0	
347	False	...	0	

	LastCheckupTime_Within past year (anytime less than 12 months ago)	\
342	True	
343	True	
345	True	
346	True	
347	True	

	Sex_Female	AlcoholDrinkers	GeneralHealth_Excellent	PhysicalActivities	\
342	True	0	False	1	
343	False	0	False	1	
345	False	1	False	0	
346	True	0	False	1	
347	True	0	False	1	

	GeneralHealth_Very good	SmokerStatus_Never smoked	\
342	True	False	
343	True	False	
345	True	False	
346	False	True	
347	False	True	

	RemovedTeeth_None of them	HadDiabetes_No
342	True	True
343	True	False
345	False	True
346	True	True
347	False	True

[5 rows x 32 columns]

```

[49]:
      HadHeartAttack  HadAngina  HadStroke  ChestScan  DifficultyWalking  \
445117              0          0          0          0              0
445123              0          0          0          0              0
445124              0          0          1          0              0
445128              0          0          0          0              0
445130              1          0          0          1              0

      HadDiabetes_Yes  GeneralHealth_Poor  PhysicalHealthDays  HadCOPD  \
445117             False                 False             0.000000      0
445123             False                 False             0.000000      0
445124              True                 False             0.000000      0
445128             False                 False             0.066667      0
445130             False                 False             0.000000      0

      RemovedTeeth_All  ...  BlindOrVisionDifficulty  \
445117             False  ...                      0
445123             False  ...                      0
445124             False  ...                      0
445128             False  ...                      0
445130             False  ...                      0

      LastCheckupTime_Within past year (anytime less than 12 months ago)  \
445117                                                                False
445123                                                                True
445124                                                                True
445128                                                                True
445130                                                                True

      Sex_Female  AlcoholDrinkers  GeneralHealth_Excellent  \
445117         False              1                     False
445123          True              0                     False
445124         False              1                     False
445128          True              0                      True
445130         False              0                     False

      PhysicalActivities  GeneralHealth_Very good  \
445117                 1                      True
445123                 1                     False
445124                 1                     False
445128                 1                     False
445130                 0                      True

      SmokerStatus_Never smoked  RemovedTeeth_None of them  HadDiabetes_No
445117                        True                      True      True
445123                        True                      True      True
445124                        True                     False     False
445128                        True                      True      True

```



445130                      True                      True                      True

[5 rows x 32 columns]

```
[50]: # Construction of the interaction term, 'Had_DM_CKD'.
# Insert column 'Had_DM_CKD' at head of high_corr_encoded_df dataframe
high_corr_encoded_df.insert(0, 'Had_DM+_CKD', value=np.nan)
# Define the new column, Had_DM_CKD, as the product of columns 'HadDiabetes' and
↳ 'HadKidneyDisease'
high_corr_encoded_df['Had_DM+_CKD'] = high_corr_encoded_df['HadDiabetes_Yes']_
↳ * high_corr_encoded_df['HadKidneyDisease']

[51]: # Reorder the columns to allow for easier viewing of relevant column/variables
reordered_columns = ['HadHeartAttack'] + ['Had_DM+_CKD'] + ['HadDiabetes_Yes']_
↳ + ['HadKidneyDisease'] + \
[col for col in high_corr_encoded_df.columns if col != 'HadHeartAttack' and col_
↳ != 'Had_DM+_CKD' \
and col != 'HadDiabetes_Yes' and col != 'HadKidneyDisease']
high_corr_encoded_df = high_corr_encoded_df[reordered_columns]

# Display new data smaller frame with its dimensions
print("\n\nSMALLER DATA FRAME CONSISTING OF ONLY HIGH CORRELATION VARIABLES\n")
print("Number of rows:", high_corr_encoded_df.shape[0])
print("Number of columns:", high_corr_encoded_df.shape[1])
high_corr_encoded_df.head()
high_corr_encoded_df.tail()
```

SMALLER DATA FRAME CONSISTING OF ONLY HIGH CORRELATION VARIABLES

Number of rows: 246022

Number of columns: 33

```
[51]:      HadHeartAttack  Had_DM+_CKD  HadDiabetes_Yes  HadKidneyDisease  \
342                0                0             False                0
343                0                0              True                0
345                0                0             False                0
346                0                0             False                0
347                0                0             False                0

      HadAngina  HadStroke  ChestScan  DifficultyWalking  GeneralHealth_Poor  \
342           0           0           0                  0                  False
343           0           0           0                  0                  False
345           0           0           1                  1                  False
```

346	0	0	0	1	False
347	0	0	0	0	False

	PhysicalHealthDays	...	BlindOrVisionDifficulty	\
342	0.133333	...	0	
343	0.000000	...	0	
345	0.000000	...	1	
346	0.166667	...	0	
347	0.100000	...	0	

	LastCheckupTime_Within past year (anytime less than 12 months ago)	\
342	True	
343	True	
345	True	
346	True	
347	True	

	Sex_Female	AlcoholDrinkers	GeneralHealth_Excellent	PhysicalActivities	\
342	True	0	False	1	
343	False	0	False	1	
345	False	1	False	0	
346	True	0	False	1	
347	True	0	False	1	

	GeneralHealth_Very good	SmokerStatus_Never smoked	\
342	True	False	
343	True	False	
345	True	False	
346	False	True	
347	False	True	

	RemovedTeeth_None of them	HadDiabetes_No
342	True	True
343	True	False
345	False	True
346	True	True
347	False	True

[5 rows x 33 columns]

[51]:

	HadHeartAttack	Had_DM+_CKD	HadDiabetes_Yes	HadKidneyDisease	\
445117	0	0	False	0	
445123	0	0	False	0	
445124	0	0	True	0	
445128	0	0	False	0	
445130	1	0	False	0	

	HadAngina	HadStroke	ChestScan	DifficultyWalking	\
445117	0	0	0	0	
445123	0	0	0	0	
445124	0	1	0	0	
445128	0	0	0	0	
445130	0	0	1	0	

	GeneralHealth_Poor	PhysicalHealthDays	...	BlindOrVisionDifficulty	\
445117	False	0.000000	...	0	
445123	False	0.000000	...	0	
445124	False	0.000000	...	0	
445128	False	0.066667	...	0	
445130	False	0.000000	...	0	

	LastCheckupTime_Within past year (anytime less than 12 months ago)	\
445117	False	
445123	True	
445124	True	
445128	True	
445130	True	

	Sex_Female	AlcoholDrinkers	GeneralHealth_Excellent	\
445117	False	1	False	
445123	True	0	False	
445124	False	1	False	
445128	True	0	True	
445130	False	0	False	

	PhysicalActivities	GeneralHealth_Very good	\
445117	1	True	
445123	1	False	
445124	1	False	
445128	1	False	
445130	0	True	

	SmokerStatus_Never smoked	RemovedTeeth_None of them	HadDiabetes_No
445117	True	True	True
445123	True	True	True
445124	True	False	False
445128	True	True	True
445130	True	True	True

[5 rows x 33 columns]

```
[52]: # RE-Verify that all variables (including 'Had_DM+_CKD') are now some form of
      ↪ numeric:
      # -- integer, binary 0 or 1
```

```

# -- float (normalized/scaled between 0 and 1)
# -- boolean, True "1"/False "0" (after one-hot encoding)
# Print unique categories for each column/variable
for column in high_corr_encoded_df.columns:
    unique_categories = high_corr_encoded_df[column].unique()
    print(f"Unique categories for column '{column}':")
    print(unique_categories)
    print()

```

Unique categories for column 'HadHeartAttack':  
[0 1]

Unique categories for column 'Had\_DM+\_CKD':  
[0 1]

Unique categories for column 'HadDiabetes\_Yes':  
[False True]

Unique categories for column 'HadKidneyDisease':  
[0 1]

Unique categories for column 'HadAngina':  
[0 1]

Unique categories for column 'HadStroke':  
[0 1]

Unique categories for column 'ChestScan':  
[0 1]

Unique categories for column 'DifficultyWalking':  
[0 1]

Unique categories for column 'GeneralHealth\_Poor':  
[False True]

Unique categories for column 'PhysicalHealthDays':  
[0.13333333 0.16666667 0.1 0.06666667 0.83333333  
1. 0.5 0.96666667 0.26666667 0.53333333 0.66666667  
0.33333333 0.3 0.23333333 0.03333333 0.7 0.2  
0.9 0.46666667 0.4 0.36666667 0.43333333 0.93333333  
0.56666667 0.76666667 0.8 0.86666667 0.6 0.73333333  
0.63333333]

Unique categories for column 'HadCOPD':  
[0 1]

Unique categories for column 'RemovedTeeth\_All':

[False True]

Unique categories for column 'PneumoVaxEver':

[1 0]

Unique categories for column 'HadArthritis':

[1 0]

Unique categories for column 'GeneralHealth\_Fair':

[False True]

Unique categories for column 'AgeCategory\_Age 80 or older':

[False True]

Unique categories for column 'DeafOrHardOfHearing':

[0 1]

Unique categories for column 'RemovedTeeth\_6 or more, but not all':

[False True]

Unique categories for column 'DifficultyErrands':

[0 1]

Unique categories for column 'DifficultyDressingBathing':

[0 1]

Unique categories for column 'SmokerStatus\_Former smoker':

[ True False]

Unique categories for column 'AgeCategory\_Age 75 to 79':

[False True]

Unique categories for column 'Sex\_Male':

[False True]

Unique categories for column 'BlindOrVisionDifficulty':

[0 1]

Unique categories for column 'LastCheckupTime\_Within past year (anytime less than 12 months ago)':

[ True False]

Unique categories for column 'Sex\_Female':

[ True False]

Unique categories for column 'AlcoholDrinkers':

[0 1]

Unique categories for column 'GeneralHealth\_Excellent':  
[False True]

Unique categories for column 'PhysicalActivities':  
[1 0]

Unique categories for column 'GeneralHealth\_Very good':  
[ True False]

Unique categories for column 'SmokerStatus\_Never smoked':  
[False True]

Unique categories for column 'RemovedTeeth\_None of them':  
[ True False]

Unique categories for column 'HadDiabetes\_No':  
[ True False]

```
[53]: # Determine correlation matrix
# Set pandas display options to show all columns
pd.set_option('display.max_rows', None)

# Calculate correlation between the selected variable and all other variables
correlation_with_HadHeartAttack_variable = high_corr_encoded_df.
    ↪corr()['HadHeartAttack'].sort_values(ascending=False)

# Print all correlation values
print("Correlation with selected variable:")
print(correlation_with_HadHeartAttack_variable)
```

Correlation with selected variable:

HadHeartAttack	1.000000
HadAngina	0.445903
HadStroke	0.177137
ChestScan	0.167760
DifficultyWalking	0.159878
HadDiabetes_Yes	0.145868
GeneralHealth_Poor	0.140607
PhysicalHealthDays	0.133420
HadCOPD	0.133223
RemovedTeeth_All	0.120564
PneumoVaxEver	0.119955
HadArthritis	0.117773
GeneralHealth_Fair	0.112319
HadKidneyDisease	0.109355
Had_DM+_CKD	0.106030
AgeCategory_Age 80 or older	0.100296

DeafOrHardOfHearing	0.097662
RemovedTeeth_6 or more, but not all	0.092477
DifficultyErrands	0.089495
DifficultyDressingBathing	0.083090
SmokerStatus_Former smoker	0.074537
AgeCategory_Age 75 to 79	0.073567
Sex_Male	0.073316
BlindOrVisionDifficulty	0.072964
LastCheckupTime_Within past year (anytime less than 12 months ago)	0.070725
Sex_Female	-0.073316
AlcoholDrinkers	-0.074181
GeneralHealth_Excellent	-0.079933
PhysicalActivities	-0.083187
GeneralHealth_Very good	-0.085347
SmokerStatus_Never smoked	-0.094843
RemovedTeeth_None of them	-0.122556
HadDiabetes_No	-0.136692

Name: HadHeartAttack, dtype: float64

```
[54]: # LOGISTIC REGRESSION AGAIN

# Isolate independent variables
column_names_with_interaction_list = high_corr_encoded_df.columns.tolist()
independent_variables_list = [x for x in column_names_with_interaction_list if
    x != 'HadHeartAttack']
independent_variables = high_corr_encoded_df[independent_variables_list]
# Isolate dependent variables
dependent_variable = high_corr_encoded_df['HadHeartAttack']

# Split data into train and test sets
X_train3, X_test3, y_train3, y_test3 = train_test_split(independent_variables,
    dependent_variable, test_size=0.2, random_state=42)

# Initialize logistic regression model
log_reg = LogisticRegression()

# Fit logistic regression model
log_reg.fit(X_train3, y_train3)

# Create a list of tuples containing coefficients and variables
coefficients_with_variables = [(coefficient, variable) for coefficient,
    variable in zip(log_reg.coef_[0], independent_variables)]
# Sort the list based on the absolute value of coefficients in descending order
coefficients_with_variables.sort(key=lambda x: x[0], reverse=True)

# Evaluate and print model accuracy
accuracy_lr_3 = log_reg.score(X_test3, y_test3)
```

```

print(f"\nACCURACY:\t{accuracy_lr_3:.5f}")

# Print model coefficients for each independent variable in descending order
print("\nMODEL INTERCEPT AND COEFFICIENTS IN DESCENDING ORDER:\n")

# Print model coefficients
print(f"INTERCEPT:\t{log_reg.intercept_[0]}")
print("\nCOEFFICIENT:\tVARIABLE:\n")
for coefficient, variable in coefficients_with_variables:
    print(f"{coefficient:.5f}:\t{variable}")

```

[54]: LogisticRegression()

ACCURACY: 0.94899

MODEL INTERCEPT AND COEFFICIENTS IN DESCENDING ORDER:

INTERCEPT: -3.8379177682053203

COEFFICIENT: VARIABLE:

2.50163:	HadAngina
0.90367:	HadStroke
0.61366:	ChestScan
0.47199:	AgeCategory_Age 80 or older
0.42350:	RemovedTeeth_All
0.35063:	LastCheckupTime_Within past year (anytime less than 12 months ago)
0.33744:	GeneralHealth_Poor
0.32790:	AgeCategory_Age 75 to 79
0.31744:	Sex_Male
0.21822:	GeneralHealth_Fair
0.18307:	HadDiabetes_Yes
0.17065:	RemovedTeeth_6 or more, but not all
0.16948:	PneumoVaxEver
0.16498:	BlindOrVisionDifficulty
0.13041:	HadArthritis
0.09381:	DifficultyWalking
0.07362:	DeafOrHardOfHearing
0.06793:	HadCOPD
0.04023:	HadKidneyDisease
0.03432:	DifficultyErrands
0.00121:	Had_DM+_CKD
-0.02692:	PhysicalHealthDays
-0.07036:	DifficultyDressingBathing
-0.07598:	PhysicalActivities
-0.13149:	SmokerStatus_Former smoker



```

-0.15720:      HadDiabetes_No
-0.19864:      AlcoholDrinkers
-0.21189:      RemovedTeeth_None of them
-0.29525:      GeneralHealth_Very good
-0.31989:      Sex_Female
-0.39783:      SmokerStatus_Never smoked
-0.63300:      GeneralHealth_Excellent

```

```

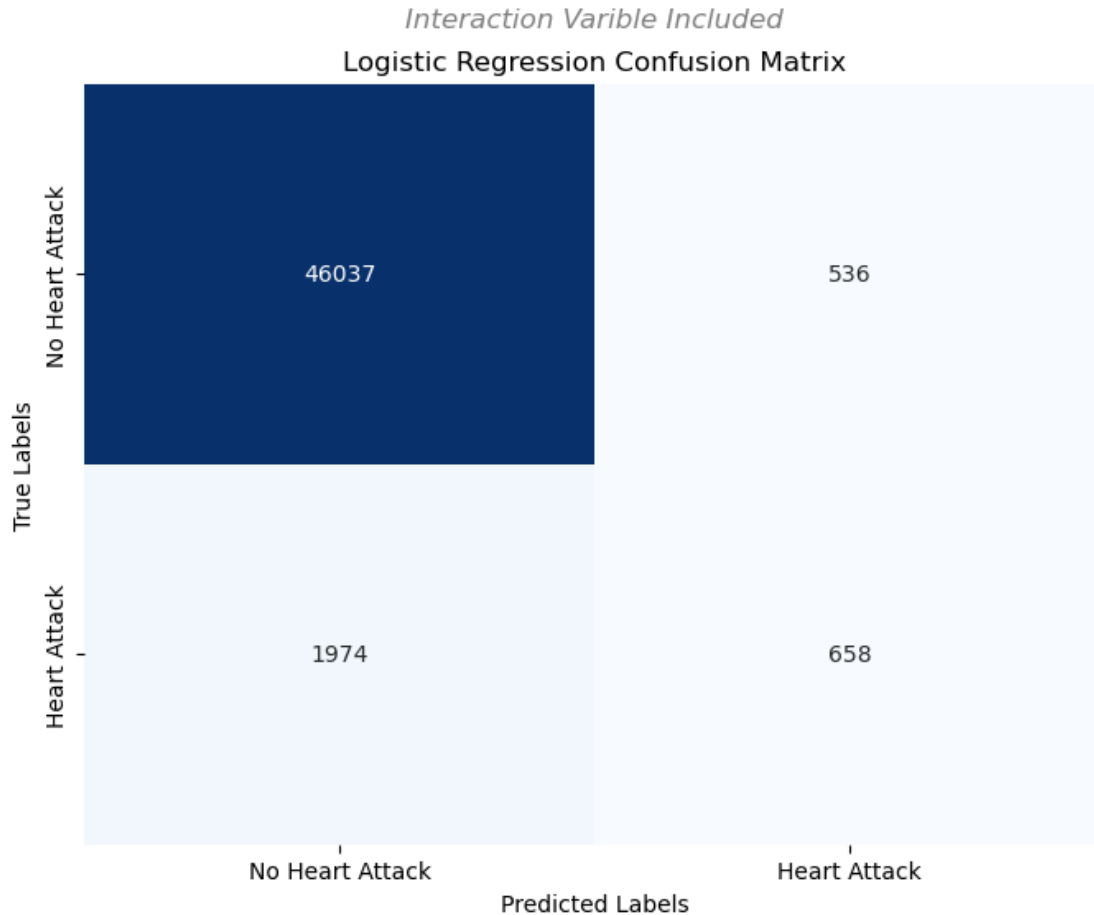
[55]: # Define custom labels
label_names = ['No Heart Attack', 'Heart Attack'];

# Calculate the confusion matrix
# Predict the labels for the test set
y_pred3 = log_reg.predict(X_test3)
conf_matrix3 = confusion_matrix(y_test3, y_pred3);

# Create a heatmap of the confusion matrix
plt.figure(figsize=(8, 6));
sns.heatmap(conf_matrix3, annot=True, fmt='d', cmap='Blues', cbar=False);
plt.title('Logistic Regression Confusion Matrix');
plt.text(1, -0.15, 'Interaction Variable Included',
        ↪horizontalalignment='center', \
            fontsize=12, fontstyle='italic', color='gray');
plt.xlabel('Predicted Labels');
plt.ylabel('True Labels');

# Set custom labels for ticks
plt.xticks(ticks=[0.5, 1.5], labels=label_names);
plt.yticks(ticks=[0.5, 1.5], labels=label_names);
plt.show();

```



```
[56]: #####
##      V.      CONCLUSION:
#####
# Based on the the results of the above logisic regression,
# the top 7 variables or factors most associated with having a heart attack are:
#   1.) HadAngina
#   2.) HadStroke
#   3.) ChestScan
#   4.) AgeCategory_Age 80 or older
#   5.) RemovedTeeth_All
#   6.) LastCheckupTime_Within past year (anytime less than 12 months ago)
#   7.) GeneralHealth_Poor

# Surprisingly, the variables HadDiabetes_Yes, HadCOPD, HadKidneyDisease, and
# the interaction term, Had_DM+_CKD, hoping to capture diabetic kidney disease,
# were not present in the top 7 risk factors.

# Based on the the results of the above logisic regression,
```

```

# the top 7 variables or factors most protective against with having a heart_
↳attack are:
# 1.) GeneralHealth_Excellent
# 2.) SmokerStatus_Never smoked
# 3.) Sex_Female
# 4.) GeneralHealth_Very good
# 5.) RemovedTeeth_None of them
# 6.) AlcoholDrinkers
# 7.) HadDiabetes_No

# Surprising among top 7 protective factors is AlcoholDrinkers.
# This most likely indicates light, moderate or social drinking, rather than_
↳heavy drinking**.
# However, this distinction is not clear in the data available**.
# Also surprising is the factor of diabetes. When absent it is protective_
↳factor,
# HadDiabetes_No, but when present it is a risk factor, but not a top 7 risk_
↳factor.

# Public Health Policy Recommendations:
# 1.) Maintain excellent health through diet and exercise.
# 2.) Don't smoke - ever.
# 3.) Practice good daily oral hygiene and see your dentist regularly.
# 4.) Drink alcohol sparingly (?**).
# 5.) Screen routinely for diabetes and prevent it, if possible, through_
↳recommendation 1.).

```