Diabetes Indicators - EDA

Group Members: Catalina Barboza Solis, Grace Heemeryck, Deysy Londono, Guangwen Jia, Connor Horemans

Introduction

Chronic disease is a classification of health conditions that last one or more years with ongoing medical care and/or limits daily activities [1]. One of the most common types of chronic diseases worldwide is diabetes, characterized by a high blood sugar level [2]. 90% of diabetes cases in Canada are type 2 diabetes, which is heavily influenced by lifestyle factors and acquired later in life, as opposed to type 1, which is genetic and often presents in childhood [3]. In Canada, 8.9% of the population have diagnosed diabetes with a 3.3% average rate increase every year. As well, 6.1% of Canadian adults are diagnosed with prediabetes, indicating that they are at a high risk of developing type 2 diabetes. The out-of-pocket cost for people living with diabetes in Canada is between 10,014 and 18,306 dollars per year depending on the type of diabetes [4].

Research has shown that year-long lifestyle modification programs help reduce the risk associated with type 2 diabetes. These programs include education on healthier food choices, stress management and physical activity [5]. These can lead to a promising plan to decrease important risk factors for diabetes such as weight, BMI and blood pressure [6].

Some surveys, such as the Behavioral Risk Factor Surveillance System (BRFSS) [7] conducted by the CDC in the United States, aim to gather health information from the general population through telephone surveys. This information can then be used to analyze behavioral trends among those with certain illnesses or at risk of developing certain illnesses, including diabetes. Better understanding of these populations helps further education and support where most needed.

The main objective of our project is to characterise the population in hand to determine where more resources and efforts are needed towards stopping the rise in diabetes. This information can later be used to make data driven decisions to build and improve health promotion activities. To help us achive this we have four guiding questions:

1- How do lifestyle choices relate to diabetes status? For this we will be looking into physical activity, smoking, heavy alcohol consumption and food choices.

2- What type of relationship exists between diabetes status and other health complications? Variables to consider in the question are high blood pressure, high cholesterol, previous strokes, difficulty walking, and heart disease.

3- How do diabetes status and demographic variables relate to each other? For this we will be looking into sex and age.

4- How does socioeconomic status relate to diabetes status? For this we will be looking into education, income, and ability to afford a doctor within the last 12 months.

Dataset

The Behavioural Risk Factor Surveillance System (BRFSS) managed by the CDC, is a state-based system of telephone health surveys that collects information on health risk behaviors, preventive health practices, and health care access primarily related to chronic disease and injury [7].

For this project, the data to be used is a subset of the 2015 survey that focused on health indicators related to diabetes. A large number of the variables are tabulated as yes or no responses, a few variables are categorical (e.g. Income and Education level) and other variables are numerical.

Although the original was released by the CDC and can be downloaded in SAS data format from their wesite, our data collection was carried out from the publicly available data source Kaggle [8]. The file found in Kaggle is already converted into a csv file which contains 253680 records reflecting the actual responses to the survey conducted by the CDC's BRFSS2015.

The dataset comprises a total of 21 columns. Each column represents a variable such as Sex, Education, Income, Diabetes, BMI, etc. and each variable is classified in different levels or categories using numbers. For instance, the variable sex, has 2 categories (1- Male and 2 – Female), while the variable Education Level has 6 categories represented by numbers 1 to 6 (No School, Elementary, Some High School, HS Graduate, Some college, College Graduate). For a better understanding of these levels the project refers to the codebook report published by the BRFSS [9]

A sample of the dataset can be seen below:

In [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
```

In [ ]:

```
diabetes_original = pd.read_csv('DATASET/diabetes_012_health_indicators_BRFSS2015.csv')
diabetes_original.head()
```

Out[ ]:

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	Ph
0	0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0	
2	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	
3	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	
4	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0	

5 rows × 22 columns

Guiding question 1: How do lifestyle choices relate to diabetes status?

1.1. Wrangling the data:

In [ ]:

```
#Made the data frame smaller to just focused on the variables I will be working with
diabetes = diabetes_original.drop(columns=['HighBP', 'HighChol','CholCheck', 'BMI', 'Str
'GenHlth','MentHlth','PhysHlth','Sex','Age','Education','Income','DiffWalk'])
```

After getting a view of the dataset, we decided that to better compare each type of person (Diabetic, Pre-diabetic, Non-diabetic), it was best to work with percentages. For this we first separated the data by diabetic status and calculated the percentage of people within that status that replied yes or no to the different lifestyle choices.

In [ ]:

```
#Make a mask for diabetes
diabetes_mask = diabetes['Diabetes_012']==2
#Create a Dataframe with only diabetics
diab_df = diabetes[diabetes_mask].copy()

#Make a mask for pre-diabetes
pre_diabetes_mask = diabetes['Diabetes_012']==1
#Create a Dataframe with only pre-diabetics
pdiab_df = diabetes[pre_diabetes_mask].copy()

#Make a mask for no-diabetes
no_diabetes_mask = diabetes['Diabetes_012']==0
#Create a Dataframe with only non-diabetics
ndiab_df = diabetes[no_diabetes_mask].copy()
```

In [ ]:

```
#Get % for each variable for each data frame
#Diabetics
diab_df.loc[:, 'Smokers'] = (diab_df['Smoker']/diab_df['Smoker'].count())*100
diab_df.loc[:, 'Heavy Drinker'] = (diab_df['HvyAlcoholConsump']/diab_df['HvyAlcoholConsu
diab_df.loc[:, 'Physical Activity'] = (diab_df['PhysActivity']/diab_df['PhysActivity'].c
diab_df.loc[:, 'Eats Fruits'] = (diab_df['Fruits']/diab_df['Fruits'].count())*100
```

```python
diab_df.loc[:, 'Eats Veggies'] = (diab_df['Veggies']/diab_df['Veggies'].count())*100

def group_and_melt (dataframe):
    #Group them then pivot the % columns to be able to plot them
    dataframe = dataframe.groupby('Diabetes_012').sum().reset_index()
    dataframe  = dataframe .melt(id_vars=['Diabetes_012','Smoker','PhysActivity','Fruits

    #Round numbers in value column
    dataframe['value']=dataframe['value'].round(1)

    #Separate negative variables from positive
    neg = dataframe.iloc[:2].copy()
    pos = dataframe.iloc[2:].copy()

    #Make labels for the graphs
    label_p = [str(num)+'%' for num in pos['value'].tolist()]
    label_n = [str(num)+'%' for num in neg['value'].tolist()]

    return neg, pos , label_p, label_n

neg_diab, pos_diab, label_dp, label_dn  = group_and_melt(diab_df)

display(neg_diab)
display(pos_diab)
```

	Diabetes_012	Smoker	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	variable	value
0	2.0	18317.0	22287.0	20693.0	26736.0	832.0	Smokers	51.8
1	2.0	18317.0	22287.0	20693.0	26736.0	832.0	Heavy Drinker	2.4

	Diabetes_012	Smoker	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	variable	value
2	2.0	18317.0	22287.0	20693.0	26736.0	832.0	Physical Activity	63.1
3	2.0	18317.0	22287.0	20693.0	26736.0	832.0	Eats Fruits	58.5
4	2.0	18317.0	22287.0	20693.0	26736.0	832.0	Eats Veggies	75.6

In [ ]:

```python
#Pre-diabetics
pdiab_df.loc[:, 'Smokers'] = (pdiab_df['Smoker']/pdiab_df['Smoker'].count())*100
pdiab_df.loc[:, 'Heavy Drinker'] = (pdiab_df['HvyAlcoholConsump']/pdiab_df['HvyAlcoholCon
pdiab_df.loc[:, 'Physical Activity'] = (pdiab_df['PhysActivity']/pdiab_df['PhysActivity']
pdiab_df.loc[:, 'Eats Fruits'] = (pdiab_df['Fruits']/pdiab_df['Fruits'].count())*100
pdiab_df.loc[:, 'Eats Veggies'] = (pdiab_df['Veggies']/pdiab_df['Veggies'].count())*100

neg_pdiab, pos_pdiab, label_pp, label_pn  = group_and_melt(pdiab_df)

display(neg_pdiab)
display(pos_pdiab)
```

	Diabetes_012	Smoker	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	variable	value
0	1.0	2282.0	3142.0	2789.0	3561.0	208.0	Smokers	49.3
1	1.0	2282.0	3142.0	2789.0	3561.0	208.0	Heavy Drinker	4.5

	Diabetes_012	Smoker	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	variable	value
2	1.0	2282.0	3142.0	2789.0	3561.0	208.0	Physical Activity	67.8
3	1.0	2282.0	3142.0	2789.0	3561.0	208.0	Eats Fruits	60.2
4	1.0	2282.0	3142.0	2789.0	3561.0	208.0	Eats Veggies	76.9

In [ ]:

```
#Non-Diabetics
ndiab_df.loc[:, 'Smokers'] = (ndiab_df['Smoker']/ndiab_df['Smoker'].count())*100
ndiab_df.loc[:, 'Heavy Drinker'] = (ndiab_df['HvyAlcoholConsump']/ndiab_df['HvyAlcoholCo
ndiab_df.loc[:, 'Physical Activity'] = (ndiab_df['PhysActivity']/ndiab_df['PhysActivity'
ndiab_df.loc[:, 'Eats Fruits'] = (ndiab_df['Fruits']/ndiab_df['Fruits'].count())*100
ndiab_df.loc[:, 'Eats Veggies'] = (ndiab_df['Veggies']/ndiab_df['Veggies'].count())*100

neg_ndiab, pos_ndiab, label_np, label_nn  = group_and_melt(ndiab_df)

display(neg_ndiab)
display(pos_ndiab)
```

	Diabetes_012	Smoker	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	variable	value
0	0.0	91824.0	166491.0	137416.0	175544.0	13216.0	Smokers	43.0
1	0.0	91824.0	166491.0	137416.0	175544.0	13216.0	Heavy Drinker	6.2

	Diabetes_012	Smoker	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	variable	value
2	0.0	91824.0	166491.0	137416.0	175544.0	13216.0	Physical Activity	77.9
3	0.0	91824.0	166491.0	137416.0	175544.0	13216.0	Eats Fruits	64.3
4	0.0	91824.0	166491.0	137416.0	175544.0	13216.0	Eats Veggies	82.1

1.2. Visualizing the data

In [ ]:

```
#Plotting the Positive Lifestyle Choices
fig = go.Figure()

fig.add_trace(go.Bar(
    y=pos_diab['variable'],
    x=pos_diab['value'],
    name='Diabetics',
    orientation='h',
    marker=dict( #Set color to bar
        color='red',
        line=dict(color='red', width=3)
    )
))

fig.add_trace(go.Bar(
    y=pos_pdiab['variable'],
    x=pos_pdiab['value'],
    name='Pre-diabetics',
    orientation='h',
    marker=dict( #Set color to bar
        color='purple',
        line=dict(color='purple', width=3)
    )
))
fig.add_trace(go.Bar(
    y=pos_ndiab['variable'],
    x=pos_ndiab['value'],
    name='Non-diabetics',
    orientation='h',
    marker=dict( #Set color to bar
        color='blue',
        line=dict(color='blue', width=3)
    )
))

fig.update_layout(title_text='Positive Lifestyle Choices', title_x = 0.5,
                  yaxis=dict(tickfont=dict(size=18)), plot_bgcolor='#bdbdbd',legend_trac
fig.show()
```

On the positive lifestyle choices which include dietary habits and physical activity we can observe that a higher percentage of the non-diabetic population partake in these lifestyle choices vs the prediabetic and diabetics. Research shows that positive lifestyle interventions towards diet and physical activity can reduce in 67% the risk of developing type 2 diabetes [10].

In [ ]:

```
fig = go.Figure()

fig.add_trace(go.Bar(
    y=neg_diab['variable'],
    x=neg_diab['value'],
    name='Diabetics',
    orientation='h',
    marker=dict(#Set color to bar
        color='red',
        line=dict(color='red', width=3)
    )
))

fig.add_trace(go.Bar(
    y=neg_pdiab['variable'],
    x=neg_pdiab['value'],
    name='Pre-diabetics',
    orientation='h',
    marker=dict(#Set color to bar
        color='purple',
        line=dict(color='purple', width=3)
    )
))
fig.add_trace(go.Bar(
    y=neg_ndiab['variable'],
    x=neg_ndiab['value'],
    name='Non-diabetics',
    orientation='h',
    marker=dict(#Set color to bar
        color='blue',
        line=dict(color='blue', width=3)
    )
))

fig.update_layout(title_text='Negative Lifestyle Choices', title_x = 0.5,
                  yaxis=dict(tickfont=dict(size=18)), plot_bgcolor='#bdbdbd',legend_trac
fig.show()
```

On the negative lifestyle choices the non-diabetic population has a higher percentage of people who are heavy drinkers.

Additionally, we can observe that more than half of the population that is diabetic is a smoker followed by 49% of the prediabetic population and 43% of non-diabetics. This might be an area of importance to improve efforts on reducing smoking habits in the overall population, as smoking can increase the probability of developing type 2 diabetes by up to 40% [11].

Guiding question 2: How do Health complcation relate to diabetes status?

2.1 Wrangling the data:

Define a fuction to calculate the value of binrary values in each diabetes group

In [ ]:

```
non_diabetes_df = diabetes_original[diabetes_original['Diabetes_012'] == 0]
pre_diabetes_df = diabetes_original[diabetes_original['Diabetes_012'] == 1]
diabetes_df = diabetes_original[diabetes_original['Diabetes_012'] == 2]

# We define a function to calculate the number for each groups.
# input :  dataset, variable
# output :  sum of result, total number

def count_binary_values(df, column):

    total_zeros = df[df[column] == 0].shape[0]
    total_ones = df[df[column] == 1].shape[0]
    total_overall = total_zeros + total_ones

    return total_zeros, total_ones, total_overall

#high blood pressure,  previous strokes, difficulty walking

non_diabetes_df.head()
pre_diabetes_df.head()
diabetes_df.head()
```

Out[ ]:

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	P
8	2.0	1.0	1.0	1.0	30.0	1.0	0.0	1.0	
10	2.0	0.0	0.0	1.0	25.0	1.0	0.0	0.0	
13	2.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	
17	2.0	0.0	0.0	1.0	23.0	1.0	0.0	0.0	
23	2.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	

5 rows × 22 columns

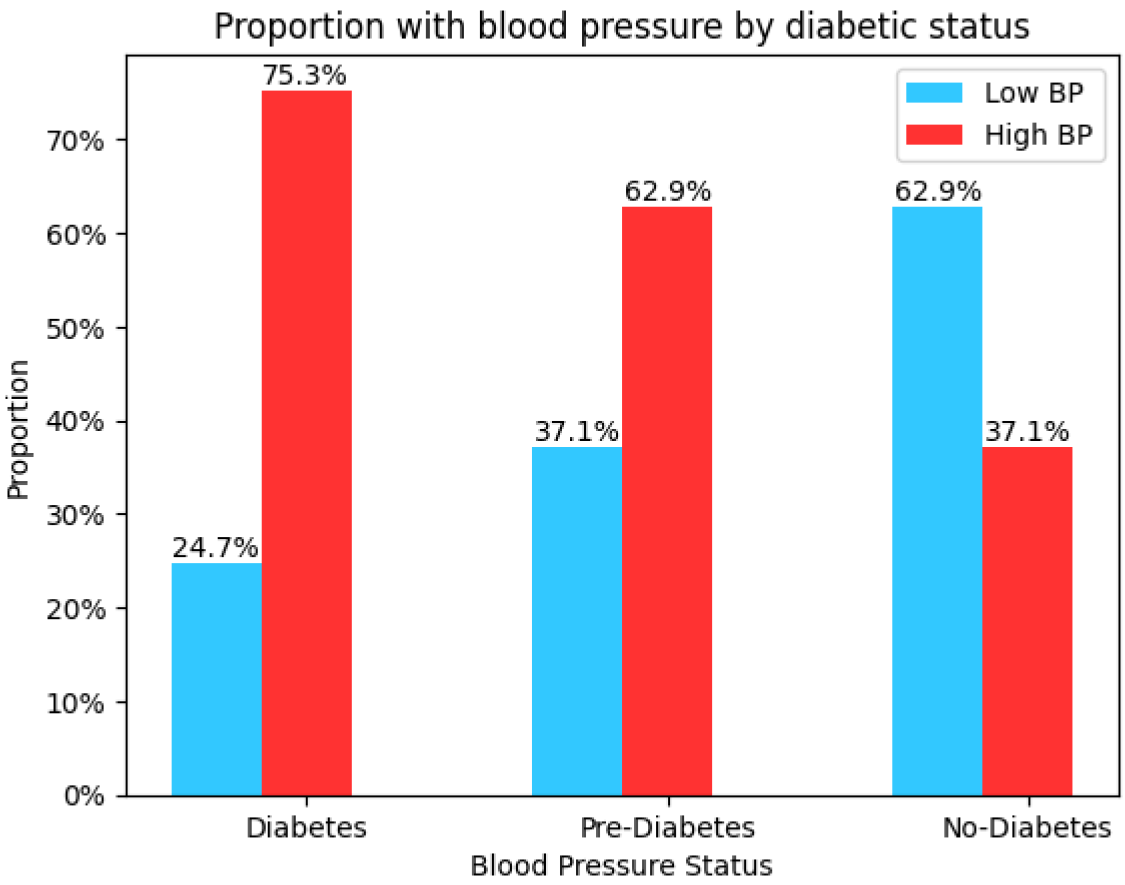
2.2 Visualizing the data

Blood Pressure

In [ ]:

```
#high blood pressure
from matplotlib.ticker import PercentFormatter
HB_ND= count_binary_values ( non_diabetes_df,"HighBP")
HB_PD= count_binary_values ( pre_diabetes_df,"HighBP")
HB_D= count_binary_values ( diabetes_df,"HighBP")

Data_HB_No = [(HB_D[0]/HB_D[2]),(HB_PD[0]/HB_PD[2]),(HB_ND[0]/HB_ND[2])]
Data_HB_Yes = [(HB_D[1]/HB_D[2]),(HB_PD[1]/HB_PD[2]),(HB_ND[1]/HB_ND[2])]

# Create labels for the bars
labels = ['Diabetes', 'Pre-Diabetes', 'No-Diabetes']

# Set the width of the bars
bar_width = 0.25
# Set the opacity
opacity = 0.8
# Set the bar positions
index = np.arange(len(labels))

# Create bars for each tuple
bars1 = plt.bar(index, Data_HB_No, bar_width, alpha=opacity, color='deepskyblue', label=
bars2 = plt.bar(index + bar_width, Data_HB_Yes, bar_width, alpha=opacity, color='red', la

max_value = 1
for bars1, Data_HB_No in zip(bars1, Data_HB_No):
    percentage = Data_HB_No / max_value * 100  # Calculating the percentage relative to
    plt.text(bars1.get_x() + bars1.get_width()/2.0, bars1.get_height(),
             f'{percentage:.1f}%',  # Formatting the percentage
             va='bottom',
             ha='center',
             color='black')

for bars2, Data_HB_Yes in zip(bars2, Data_HB_Yes):
    percentage = Data_HB_Yes / max_value * 100  # Calculating the percentage relative to
    plt.text(bars2.get_x() + bars2.get_width()/2.0, bars2.get_height(),
             f'{percentage:.1f}%',  # Formatting the percentage
             va='bottom',
             ha='center',
             color='black')



plt.gca().yaxis.set_major_formatter(PercentFormatter(1))
# Add some text for labels, title and axes ticks
plt.xlabel('Blood Pressure Status')
plt.ylabel('Proportion')
plt.title('Proportion with blood pressure by diabetic status')
plt.xticks(index + bar_width, labels)
plt.legend()
```

```
Out[ ]: <matplotlib.legend.Legend at 0x1ba58f91ba0>
```



In the analysis, we split the whole dataset to three sub data set based on the diabetic status, diabetes, pre-diabetes, and no – diabetes. In this form, this is easier to say the relationship between the diabetic status to the variable.

With the variable of blood pressure. From the graph, it is easy to say there is a strong relationship between blood pressure and diabetic status. In the diabetes group, there is a higher proportion of people who have high blood pressure compared with other two groups with a figure of 75.3%. Compared to the other two groups, pre-diabetes owns 62.9% of people who have high blood pressure while 37.1 % of people who have high blood pressure, which is a considerably high figure. It draws a clean straight line from the diabetes group to no-diabetes. The proportion between diabetes group and no-diabetes group is around 40%, which shows the strong relationship of high blood pressure and diabetic status.

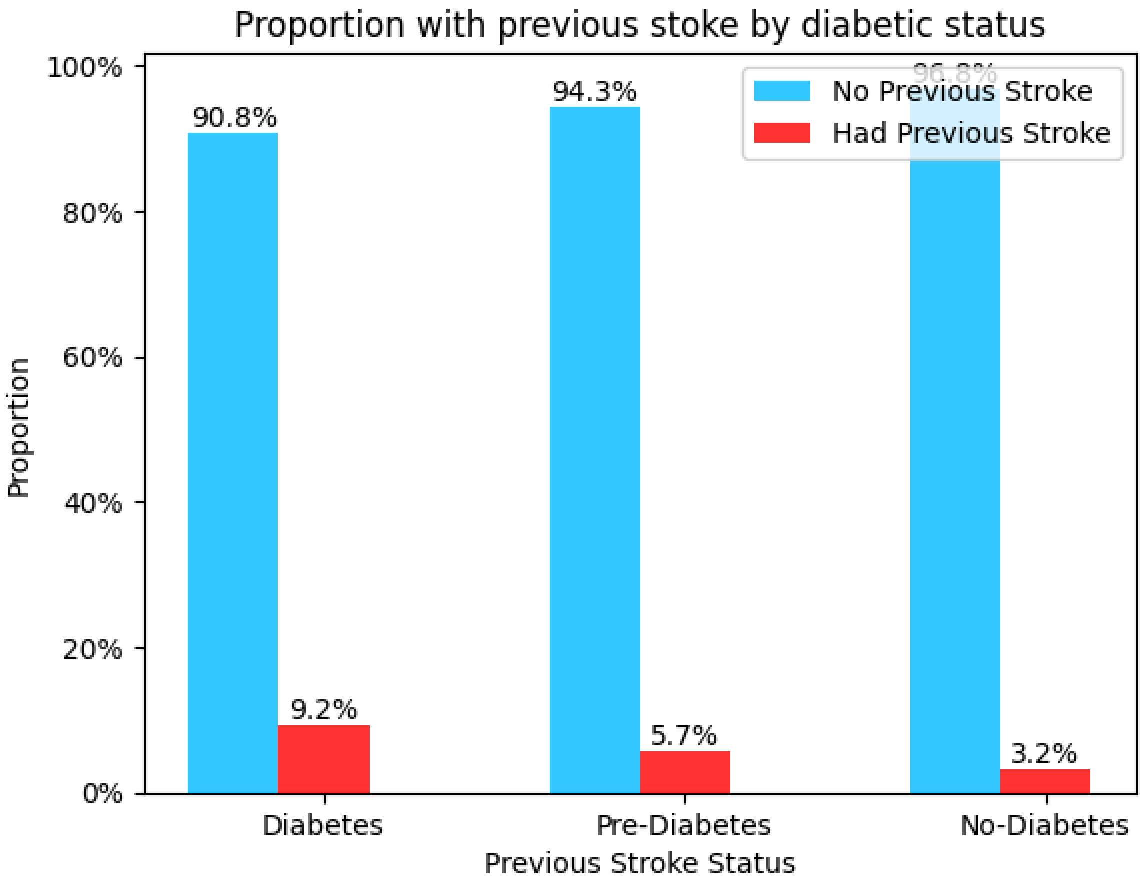
This shows a same result as public research.[12]

Previous Stroke

In [ ]:

```
#previous strokes
PS_ND= count_binary_values ( non_diabetes_df,"Stroke")
PS_PD= count_binary_values ( pre_diabetes_df,"Stroke")
PS_D= count_binary_values ( diabetes_df,"Stroke")

Data_PS_No = [(PS_D[0]/PS_D[2]),(PS_PD[0]/PS_PD[2]),(PS_ND[0]/PS_ND[2])]
Data_PS_Yes = [(PS_D[1]/PS_D[2]),(PS_PD[1]/PS_PD[2]),(PS_ND[1]/PS_ND[2])]

# Create labels for the bars
labels = ['Diabetes', 'Pre-Diabetes', 'No-Diabetes']

# Set the width of the bars
bar_width = 0.25
# Set the opacity
opacity = 0.8
# Set the bar positions
index = np.arange(len(labels))

# Create bars for each tuple
bars1 = plt.bar(index, Data_PS_No, bar_width, alpha=opacity, color='deepskyblue', label=
bars2 = plt.bar(index + bar_width, Data_PS_Yes, bar_width, alpha=opacity, color='red', la
plt.gca().yaxis.set_major_formatter(PercentFormatter(1))

#Percentage
max_value = 1
for bars1, Data_PS_No in zip(bars1, Data_PS_No):
    percentage = Data_PS_No / max_value * 100  # Calculating the percentage relative to t
    plt.text(bars1.get_x() + bars1.get_width()/2.0, bars1.get_height(),
             f'{percentage:.1f}%',  # Formatting the percentage
             va='bottom',
             ha='center',
             color='black')

for bars2, Data_PS_Yes in zip(bars2, Data_PS_Yes):
    percentage = Data_PS_Yes / max_value * 100  # Calculating the percentage relative to
    plt.text(bars2.get_x() + bars2.get_width()/2.0, bars2.get_height(),
             f'{percentage:.1f}%',  # Formatting the percentage
             va='bottom',
             ha='center',
             color='black')


# Add some text for labels, title and axes ticks
plt.xlabel('Previous Stroke Status')
plt.ylabel('Proportion')
plt.title('Proportion with previous stoke by diabetic status')
plt.xticks(index + bar_width, labels)
plt.legend()
```

```
Out[ ]: <matplotlib.legend.Legend at 0x1ba59032530>
```



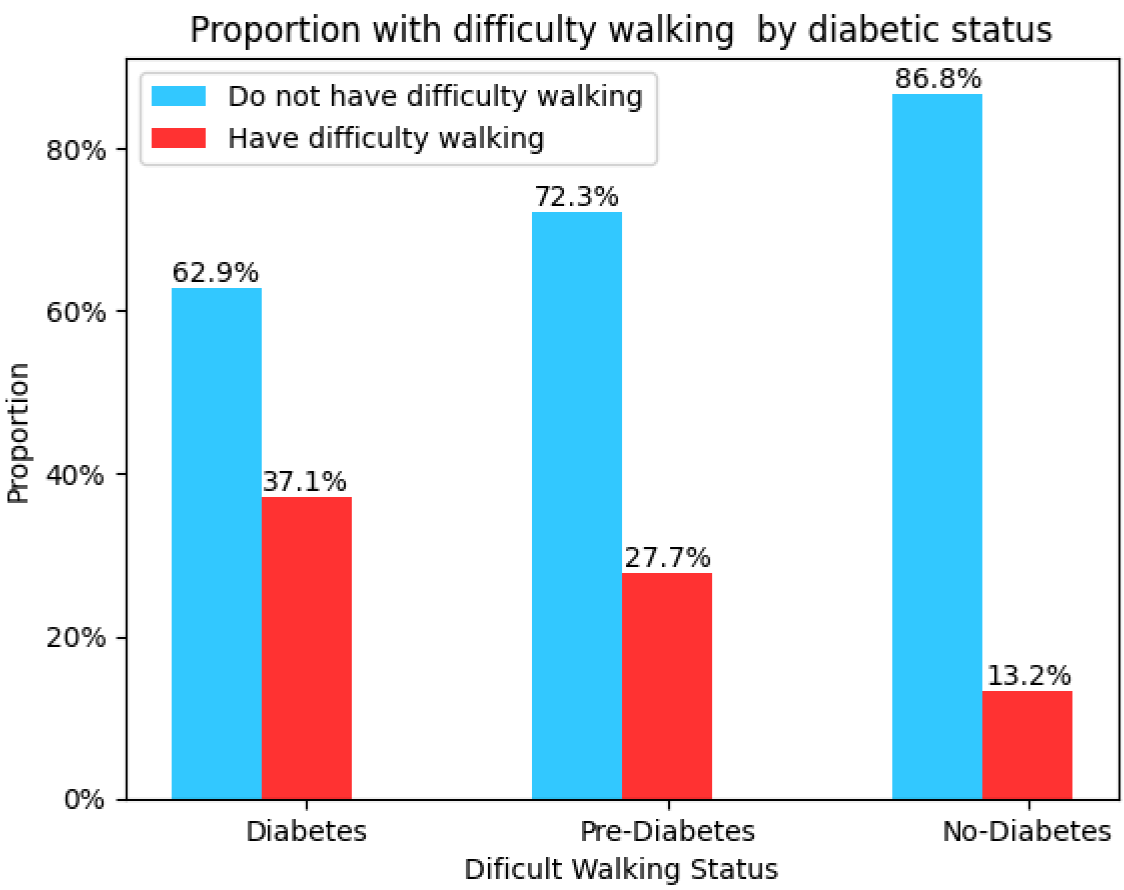
The next variable is if had previous stoke. The proportion had previous stroke in all three groups is relatively low with a maximum figure of 9.2% and a minimum figure of 3.2%. Which shows the relationship between the previous stoke status and diabetic status is low. We cannot conclude that if a person had previous stroke, this person may have a higher probability of getting diabetes. However, there is a still a pattern in there. Based on the graph, we could still say the proportion of previous stoke in diabetes is higher than proportion in no-diabetes group. The proportion or previous stroke is still increases from no-diabetes group to diabetes group.

This shows a similar result as the public research states there is a relationship between previous stroke and diabetic status[13]

Difficulty Walking

```
In [ ]: #difficulty walking
DW_ND= count_binary_values ( non_diabetes_df,"DiffWalk")
DW_PD= count_binary_values ( pre_diabetes_df,"DiffWalk")
DW_D= count_binary_values ( diabetes_df,"DiffWalk")

Data_DW_No = [(DW_D[0]/DW_D[2]),(DW_PD[0]/DW_PD[2]),(DW_ND[0]/DW_ND[2])]
Data_DW_Yes = [(DW_D[1]/DW_D[2]),(DW_PD[1]/DW_PD[2]),(DW_ND[1]/DW_ND[2])]

# Create labels for the bars
labels = ['Diabetes', 'Pre-Diabetes', 'No-Diabetes']

# Set the width of the bars
bar_width = 0.25
# Set the opacity
opacity = 0.8
# Set the bar positions
index = np.arange(len(labels))

# Create bars for each tuple
bars1 = plt.bar(index, Data_DW_No, bar_width, alpha=opacity, color='deepskyblue', label=
bars2 = plt.bar(index + bar_width, Data_DW_Yes, bar_width, alpha=opacity, color='red', la
plt.gca().yaxis.set_major_formatter(PercentFormatter(1))

#Percentage
max_value = 1
for bars1, Data_DW_No in zip(bars1, Data_DW_No):
    percentage = Data_DW_No / max_value * 100  # Calculating the percentage relative to t
    plt.text(bars1.get_x() + bars1.get_width()/2.0, bars1.get_height(),
             f'{percentage:.1f}%',  # Formatting the percentage
             va='bottom',
             ha='center',
             color='black')

for bars2, Data_DW_Yes in zip(bars2, Data_DW_Yes):
    percentage = Data_DW_Yes / max_value * 100  # Calculating the percentage relative to
    plt.text(bars2.get_x() + bars2.get_width()/2.0, bars2.get_height(),
             f'{percentage:.1f}%',  # Formatting the percentage
             va='bottom',
             ha='center',
             color='black')


# Add some text for labels, title and axes ticks
plt.xlabel('Dificult Walking Status')
plt.ylabel('Proportion')
plt.title('Proportion with difficulty walking  by diabetic status')
plt.xticks(index + bar_width, labels)
plt.legend()
```

```
Out[ ]: <matplotlib.legend.Legend at 0x1ba59256e30>
```



Last variable in health complication, the difficult walking status. Compared with the blood pressure and previous stroke variables, the difficult walking status appear to have a moderate relationship between diabetic status. In the graph, the maximum proportion of have difficulty walking is the group diabetes with a figure of 37.1%, while the minimum proportion is from group no-diabetes with a figure of 13.2%. From the graph, we can still see the trends from group diabetes to no-diabetes, the proportion decreases as we move to no-diabetes, which conclude a similar result of previous publish that diabetic status will influence the mobility of walking[14].

2.3 Wrangling the data

In [ ]:

```
converteddata = diabetes_original.astype(int)
converteddata.head()
```

Out[ ]:

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	Ph
0	0	1	1	1	40	1	0	0	
1	0	0	0	0	25	1	0	0	
2	0	1	1	1	28	0	0	0	
3	0	1	0	1	27	0	0	0	
4	0	1	1	1	24	0	0	0	

5 rows × 22 columns

2.4 Visualizing the data

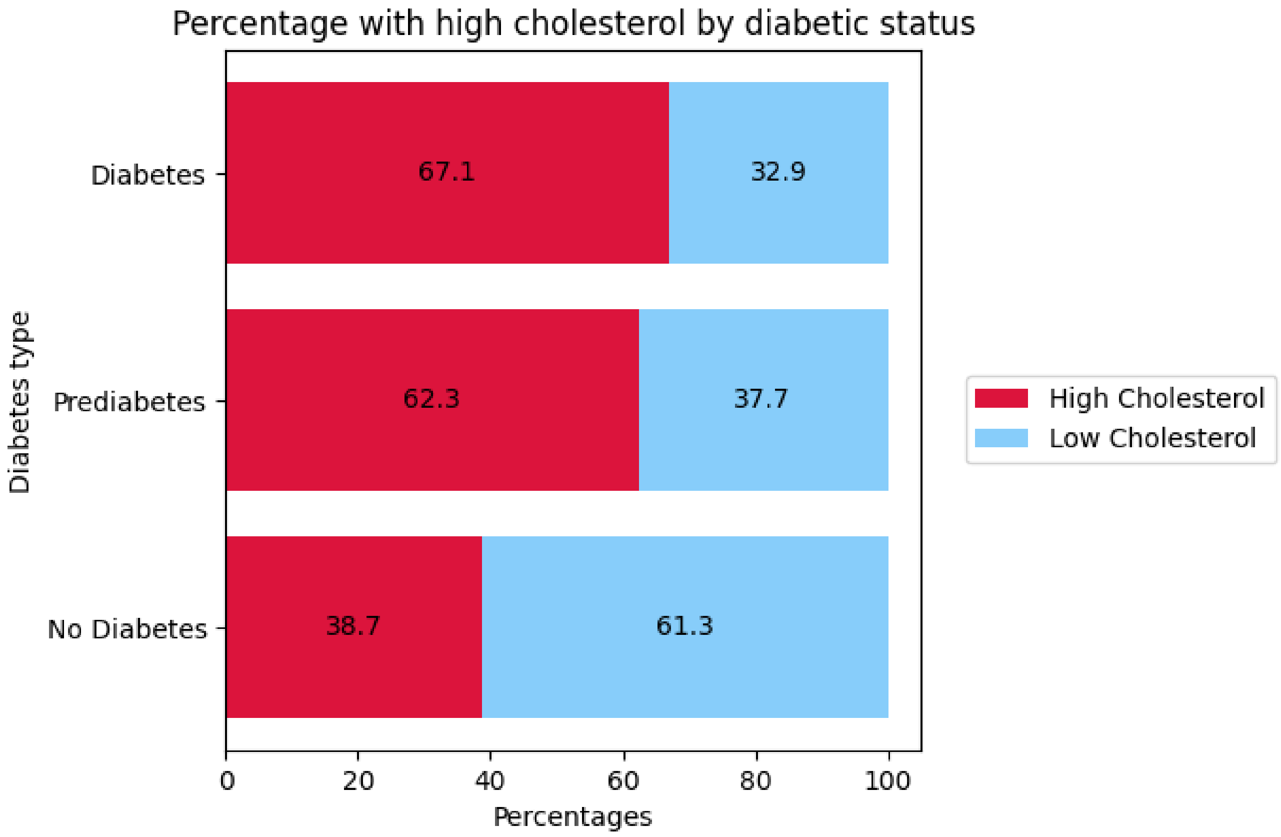
Next, we will examine the value of high cholesterol. For this particular analysis, to try and get the most accurate data possible, we filtered this data by those who had a cholesterol check in the last five years (defined by variable CholCheck).

Cholesterol

In [ ]:

```
#Examine high cholesterol

#Filter based on those who had a cholesterol check in the last five years
checkedChol = converteddata[converteddata["CholCheck"] == 1].filter(["Diabetes_012", "Hi
checkedChol.head()

#Group by diabetes status
groupedCholesterol = checkedChol.groupby("Diabetes_012").sum()

#Count the no. with each diabetes status
nodiabetesCount = len(checkedChol[checkedChol["Diabetes_012"] == 0])
prediabetesCount = len(checkedChol[checkedChol["Diabetes_012"] ==1])
diabetesCount = len(checkedChol[checkedChol["Diabetes_012"] ==2])

#All the total counts
totals = np.array([nodiabetesCount, prediabetesCount, diabetesCount])

#All the choletserol values
cholcounts = np.array(list(groupedCholesterol["HighChol"]))

#Calculate the proportions
ratios = cholcounts/totals

reverseRatios = 1 - ratios

#Convert to percentages
ratios*=100
reverseRatios *=100

#Get labels
```

```
titles = ["No Diabetes", "Prediabetes", "Diabetes"]
levels = {"High Cholesterol":ratios, "Low Cholesterol":reverseRatios}

#Plot figures
fig, ax = plt.subplots()

high = ax.barh(titles, ratios, color='crimson')
low = ax.barh(titles, reverseRatios, left=ratios, color='lightskyblue')
fig.legend(["High Cholesterol", "Low Cholesterol"], loc="center right")
fig.subplots_adjust(right=0.70)
ax.set_ylabel("Diabetes type")
ax.set_xlabel("Percentages")
ax.bar_label(high,label_type='center',fmt='%.1f')
ax.bar_label(low,label_type='center',fmt='%.1f')
ax.set_title("Percentage with high cholesterol by diabetic status")
plt.show()
```



Here, we can see that a larger proportion of people with diabetes and prediabetes have high cholesterol compared to those with no diabetes. This is consistent with the literature which states that people with diabetes are more likely to have high cholesterol levels. [15]

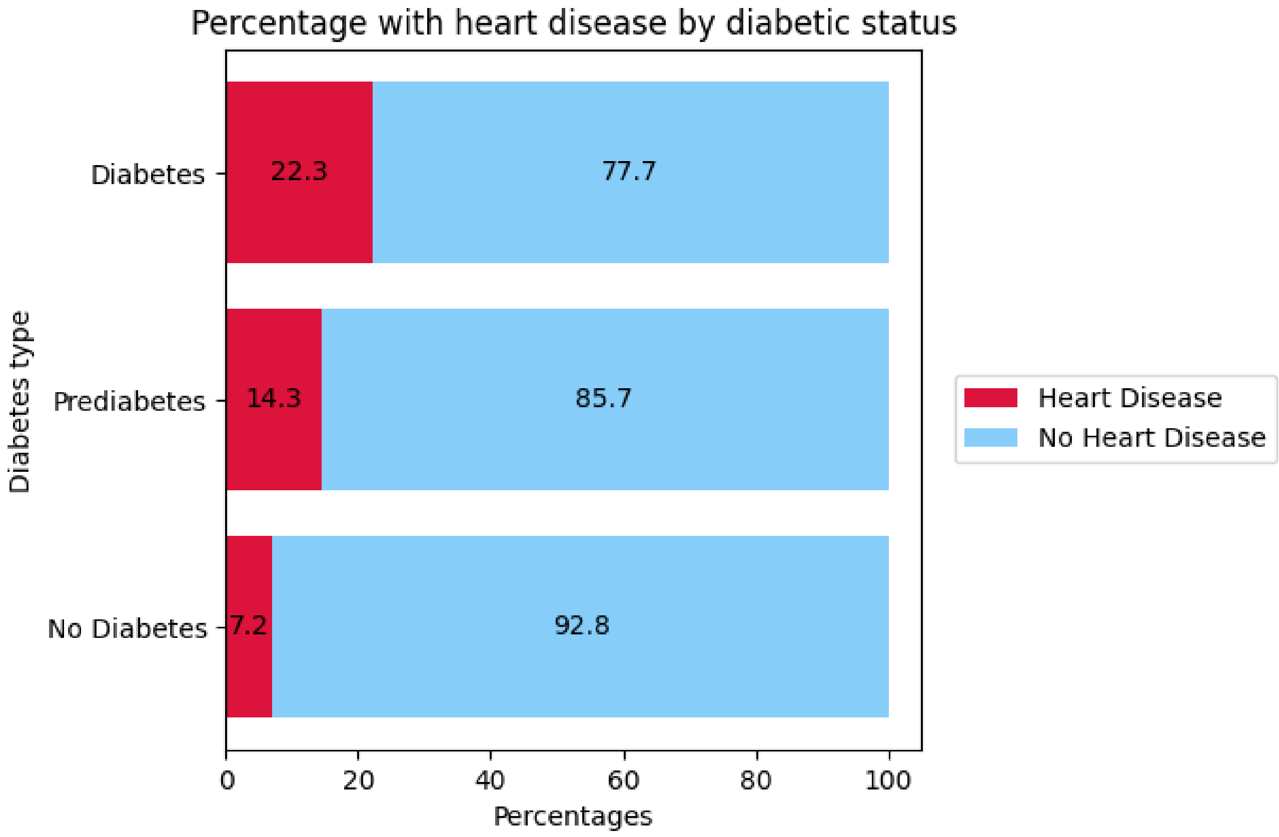
Following this, our last variable of interest is heart disease, defined as being diagnosed with cardiovascular disease or having had a heart attack in the past.

Heart Disease

In [ ]:

```
#Get heart disease data
heartDisease = converteddata.filter(["Diabetes_012", "HeartDiseaseorAttack"])
```

```
heartDisease.head()

#Group by diabetes status
groupedHeart = heartDisease.groupby("Diabetes_012",as_index=False).sum()
groupedHeart

#Get our new total count by diabetes status
nodiabetesCount = len(heartDisease[heartDisease["Diabetes_012"] == 0])
prediabetesCount = len(heartDisease[heartDisease["Diabetes_012"] ==1])
diabetesCount = len(heartDisease[heartDisease["Diabetes_012"] ==2])

#Get our totals
totals = np.array([nodiabetesCount, prediabetesCount, diabetesCount])

heartcounts = np.array(list(groupedHeart["HeartDiseaseorAttack"]))

#Calculate proportions
ratios = heartcounts/totals

reverseRatios = 1 - ratios

#Convert to percentages for the graph
ratios*=100
reverseRatios *=100

#Get labels
titles = ["No Diabetes", "Prediabetes", "Diabetes"]

#Plot figures
fig, ax = plt.subplots()

high = ax.barh(titles, ratios, color='crimson')
low = ax.barh(titles, reverseRatios, left=ratios, color='lightskyblue')
fig.legend(["Heart Disease", "No Heart Disease"], loc="center right")
fig.subplots_adjust(right=0.70)
ax.set_ylabel("Diabetes type")
ax.set_xlabel("Percentages")
ax.bar_label(high,label_type='center',fmt='%.1f')
ax.bar_label(low,label_type='center',fmt='%.1f')
ax.set_title("Percentage with heart disease by diabetic status")
plt.show()
```



Again, we can see that people with diabetes have a greater proportion of heart disease, followed by those with prediabetes, and finally those with no diabetes having the lowest incidence of heart disease. This is consistent with the literature which states that those with diabetes are at a greater risk for heart disease morbidity and mortality. [16]

Overall, we see that for all of the health conditions we examined there is a higher proportion with these diseases in those with diabetes and prediabetes compared to non-diabetics.

Guiding question 3: How does diabetes status relate to Sex and Age?

1.1 Wrangling the data:

In [ ]:

```
diabetes_original.head()
# Sliced the columns needed to answer this question
df1 =  diabetes_original.loc[: , ['Diabetes_012','Sex','Age']]

df1.loc[df1['Diabetes_012'] == 0, 'Diabetes_012'] = 'No_Diabetes'
df1.loc[df1['Diabetes_012'] == 1, 'Diabetes_012'] = 'Pre_diabetes'
df1.loc[df1['Diabetes_012'] == 2, 'Diabetes_012'] = 'Diabetes'
df1.loc[df1['Sex'] == 0, 'Sex'] = 'Female'
df1.loc[df1['Sex'] == 1, 'Sex'] = 'Male'

#Now, we regroup the ages, to have less bars in the visualization an make the data easier
```

```
df1.loc[(df1['Age'] == 1)|(df1['Age'] == 2)|(df1['Age'] == 3), 'Age'] = '18 to 34'
df1.loc[(df1['Age'] == 4)|(df1['Age'] == 5)|(df1['Age'] == 6), 'Age'] = '35 to 49'
df1.loc[(df1['Age'] == 7)|(df1['Age'] == 8)|(df1['Age'] == 9), 'Age'] = '50 to 64'
df1.loc[(df1['Age'] == 10)|(df1['Age'] == 11)|(df1['Age'] == 12)|(df1['Age'] == 13), 'Ag

## Analysis Sex and Diabetes
gr_by_sx = df1.groupby([df1['Sex']])                                  #df1 grouped by se
df_m = gr_by_sx.get_group('Male').groupby('Diabetes_012').count()     # Number for ND, P
df_f = gr_by_sx.get_group('Female').groupby('Diabetes_012').count()   # Number for ND, P

gr_age = df1.groupby([df1['Age']])
df_1 = gr_age.get_group('18 to 34').groupby('Diabetes_012').count()   # Number for ND, P
df_2 = gr_age.get_group('35 to 49').groupby('Diabetes_012').count()   # Number for ND, P
df_3 = gr_age.get_group('50 to 64').groupby('Diabetes_012').count()   # Number for ND, P
df_4 = gr_age.get_group('65 or older').groupby('Diabetes_012').count()# Number for ND, P
#Proportions by sex
pmd = df_m.Sex['Diabetes']/df1['Sex'].value_counts()['Male']        # Proportion of male
pfd = df_f.Sex['Diabetes']/df1['Sex'].value_counts()['Female']      # Proportion of fema
pmpd = df_m.Sex['Pre_diabetes']/df1['Sex'].value_counts()['Male']   # Proportion of male
pfpd = df_f.Sex['Pre_diabetes']/df1['Sex'].value_counts()['Female'] # Proportion of fema
pmnd = df_m.Sex['No_Diabetes']/df1['Sex'].value_counts()['Male']    # Proportion of male
pfnd = df_f.Sex['No_Diabetes']/df1['Sex'].value_counts()['Female']  # Proportion of fema

lstm_prop = [pmd,pmpd,pmnd]                                         #list of male porport
lstf_prop = [pfd,pfpd,pfnd]                                         #list of female propo

#Proportions by age
pd_1834 = df_1.Age['Diabetes']/df1['Age'].value_counts()['18 to 34']   # Proportion of p
pd_3549 = df_2.Age['Diabetes']/df1['Age'].value_counts()['35 to 49']   # Proportion of p
pd_5065 = df_3.Age['Diabetes']/df1['Age'].value_counts()['50 to 64']   # Proportion of p
pd_65   = df_4.Age['Diabetes']/df1['Age'].value_counts()['65 or older']# Proportion of p
ppd_1834 = df_1.Age['Pre_diabetes']/df1['Age'].value_counts()['18 to 34']   # Proportion
ppd_3549 = df_2.Age['Pre_diabetes']/df1['Age'].value_counts()['35 to 49']   # Proportion
ppd_5065 = df_3.Age['Pre_diabetes']/df1['Age'].value_counts()['50 to 64']   # Proportion
ppd_65   = df_4.Age['Pre_diabetes']/df1['Age'].value_counts()['65 or older']# Proportion
pnd_1834 = df_1.Age['No_Diabetes']/df1['Age'].value_counts()['18 to 34']   # Proportion o
pnd_3549 = df_2.Age['No_Diabetes']/df1['Age'].value_counts()['35 to 49']   # Proportion o
pnd_5065 = df_3.Age['No_Diabetes']/df1['Age'].value_counts()['50 to 64']   # Proportion o
pnd_65   = df_4.Age['No_Diabetes']/df1['Age'].value_counts()['65 or older']# Proportion o

lst34_prop = [pd_1834,ppd_1834,pnd_1834]          #list of male porportions
lst49_prop = [pd_3549,ppd_3549,pnd_3549]          #list of female proportions
lst64_prop = [pd_5065,ppd_5065,pnd_5065]          #list of female proportions
lst65_prop = [pd_65,ppd_65,pnd_65]                #list of female proportions
```

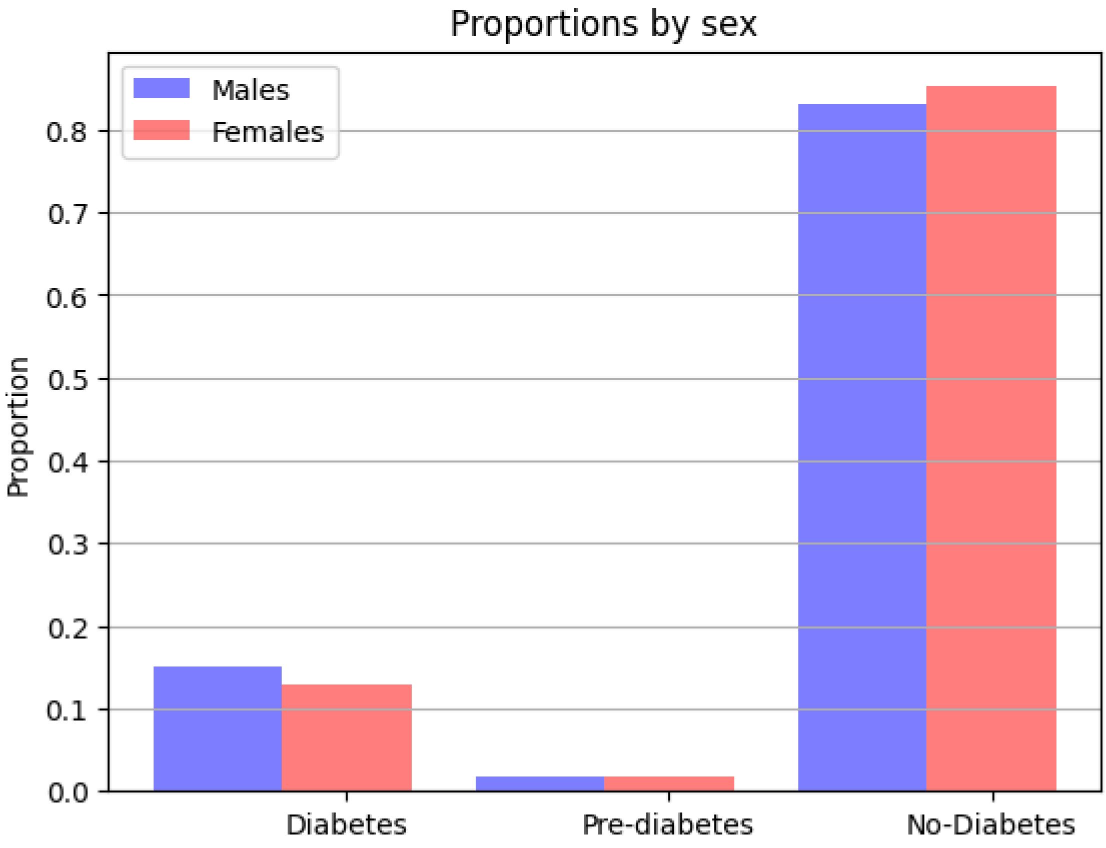
2.2 Visualizing the data

Sex related to Diabetic status

In [ ]:

```
fig = plt.figure()
ax = fig.add_subplot(111)                # By sex
x = np.arange(3)                         # x locations for the groups
wdt = 0.4                                 # the width of the bars
```

```
m1 = ax.bar(x ,lstm_prop,color ='Blue', width = wdt, label = 'Male',alpha=0.5)
m2 = ax.bar(x+wdt,lstf_prop,color ='Red', width = wdt, label = 'Female',alpha=0.5)
ax.set_ylabel('Proportion')
ax.set_xticks(x+wdt)
ax.set_xticklabels(('Diabetes','Pre-diabetes','No-Diabetes'))
ax.legend((m1[0], m2[0]),('Males', 'Females'))
ax.set_title('Proportions by sex')
ax.grid(axis='y')
```



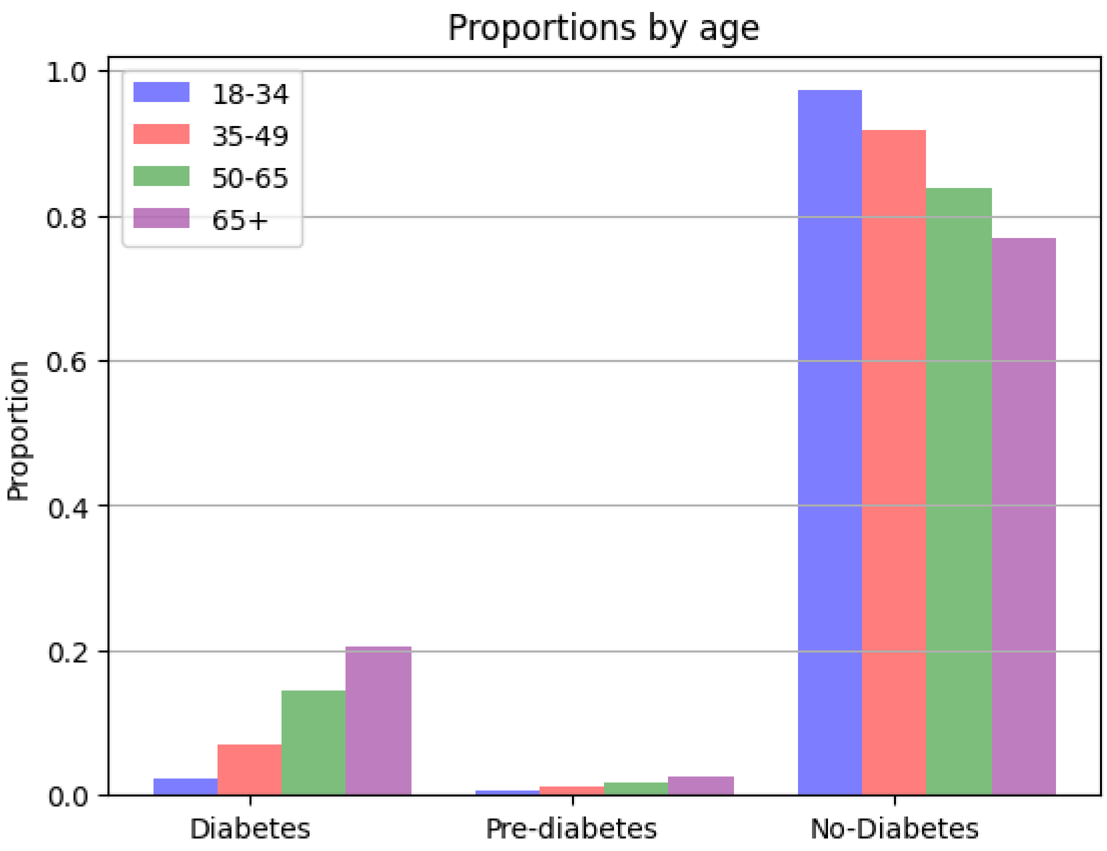
We can see from this graph that males have a slightly larger proportion with diabetes. According to previous studies [17], the prevalence of type 2 diabetes is increasing in both sexes; however, it is estimated that worldwide there are 17.7 million more men than women living with this illness. This is consistent with our findings.

Age related to Diabetic status

In [ ]:

```
fig = plt.figure()
ax1 = fig.add_subplot(111)                # By age
wdt2 = 0.2                                 # the width of the bars
m1 = ax1.bar(x ,lst34_prop,color ='Blue', width = wdt2, label = '18-34',alpha=0.5)
m2 = ax1.bar(x+wdt2,lst49_prop,color ='Red', width = wdt2, label = '35-49',alpha=0.5)
m3 = ax1.bar(x+wdt2*2,lst64_prop,color ='Green', width = wdt2, label = '50-64',alpha=0.5
m4 = ax1.bar(x+wdt2*3,lst65_prop,color ='Purple', width = wdt2, label = '65+',alpha=0.5)
ax1.set_ylabel('Proportion')
ax1.set_xticks(x+wdt2)
ax1.set_xticklabels(('Diabetes','Pre-diabetes','No-Diabetes'))
ax1.legend((m1[0], m2[0], m3[0], m4[0]),('18-34', '35-49','50-65','65+'))
```

```
ax1.set_title('Proportions by age')
ax1.grid(axis='y')
```



We can see from this graph that the proportion of those with diabetes increases with age, with the 65+ age group having the highest proportion with diabetes. This is consistent with previous research stating that as a person's age increases they greatly increase the risk of developing diabetes. [18]

Guiding Question 4: How does socioeconomic factors relate to diabetes status

Wrangling and cleaning the data for socioeconomic factors

To help guide us towards answering this guiding question, we dropped all other columns that were not income status, education level and no doctor because of cost.

We then split up the original dataframe into 3 separate dataframes based on diabetes status. This way we were able to look at each of the three diabetes status populations and compare them to one another.

Next, we calculated proportions for each variable based on the diabetes status dataframes. So for instance, for income, we took the total amount of nondiabetic people that said they were in a certain income bracket and divided it by the total amount of nondiabetic people. We did this for each of the three diabetes status groups. This was done for all three variables of interest that were looked at for the guiding question. This decision to get proportions equal to 1 for each diabetes status dataframe was decided upon since it is easier to compare **within** the diabetes status groups when we later visualize our data all together in one big plot (all three groups and all its categories being present). If we instead decided to get proportions based on a single category in one of the variables divided by the total people who are in that category (eg. total number of people with at most a high school level education who are diabetic divided by total number of people with at most a high school education) then we would get proportions that are massively skewed towards the non diabetic group and it would be difficult to see the smaller sized diabetes status groups (prediabetes in this case) on the main bar plot. This is because the three diabetes status groups did not have equal amount of responses in the survey, for instance the amount of people who responded saying they were non-diabetic was around 213,000 while the amount of people who said they were prediabetic was around 4600.

However, for our subplot graphs of the categories for each variable, we made it so the proportions add up to 1 for each category separately. This way allows us to have a consistent y axis range throughout our subplots so that we can compare **between** the different categories.

We then converted the proportions into percentages by multiplying each proportion by 100. Converting them to percentages was a decision made by our group. This was done since we believe it was better to visualize the data as percentages which are easier to follow. We also ensured that the data for every guiding question we did was shown as a percentage to keep it consistent throughout the project.

In [ ]:

```
#drop columns that I am not using for the guided question

#make a new df for socioeconomic factors from original dataframe
diabetes_socioeconomic_df = pd.DataFrame()

columns_to_keep = ["Diabetes_012","NoDocbcCost","Education","Income"]
#loop to add columns I want to keep to the socioeconomic dataframe
for column in columns_to_keep:
    diabetes_socioeconomic_df[column] = diabetes_original[column]
```

Wrangling the data below into 3 separate dataframes based on diabetes status

In [ ]:

```
#DATAFRAMES MADE FOR DIABETIC,PREDIABETIC AND NONDIABETIC RESPONSES TO SPLIT UP POPULATIO
#we are doing this for the main bar charts

#grab all non-diabetic rows (make df with just non-diabetic survey responses, which are 0
non_dia_value = 0.0 #preset non diabetic response value to a variable to use below
non_diabetic = diabetes_socioeconomic_df[diabetes_socioeconomic_df["Diabetes_012"] == nor

#grab all pre-diabetic rows (make df with just pre-diabetic survey responses, which are 1
pre_dia_value = 1.0 #preset pre diabetic response value to a variable to use below
pre_diabetic = diabetes_socioeconomic_df[diabetes_socioeconomic_df["Diabetes_012"] == pre

#grab all diabetic rows (make df with just diabetic survey responses, which are 2.0 in th
dia_value = 2.0 #preset diabetic response value to a variable to use below
diabetic = diabetes_socioeconomic_df[diabetes_socioeconomic_df["Diabetes_012"] == dia_va
```

Converting the 3 separate dataframes made from above into proportions and then to percentages to visualize the main bar plots down below

In [ ]:

```
#*******************Making Dataframes for education, income and nodoc percentages*******

#make empty dataframes for education, income and nodoc for the percentages
edu_props_df = pd.DataFrame()
income_props_df = pd.DataFrame()
nodoc_props_df = pd.DataFrame()

#set the value to multiply by to get percentage from the proportion gathered. This will b
percentage_mult = 100

#add in dia,pre and non nodoc percentages to nodocbccost dataframe
nodoc_props_df["Diabetic"] = diabetic["NoDocbcCost"].value_counts(normalize=True)*percent
nodoc_props_df["Pre-diabetic"] = pre_diabetic["NoDocbcCost"].value_counts(normalize=True)
nodoc_props_df["Non-diabetic"] = non_diabetic["NoDocbcCost"].value_counts(normalize=True)

#reset the index and transpose it and rename survey response to doccost with more appropr
nodoc_props_df = nodoc_props_df.T.reset_index()
nodoc_props_df = nodoc_props_df.rename(columns={"index": "Diabetes Status", 0.0:"No", 1.0

print("NoDocbcCost Proportion Dataframe")
display(nodoc_props_df)

#add in dia,pre-dia and non-dia income percentages to income dataframe
income_props_df["Diabetic"] = diabetic["Income"].value_counts(normalize=True)*percentage_
income_props_df["Pre-diabetic"] = pre_diabetic["Income"].value_counts(normalize=True)*pe
income_props_df["Non-diabetic"] = non_diabetic["Income"].value_counts(normalize=True)*pe

#reset the index and transpose it and rename each income category to more appropriate nam
income_props_df = income_props_df.T.reset_index()
income_props_df = income_props_df.rename(columns={"index": "Diabetes Status", 1.0:"Less 

print("\n \n Income Proportion Dataframe")
display(income_props_df)

#add in dia,pre-dia and non-dia education percentages to education dataframe
edu_props_df["Diabetic"] = diabetic["Education"].value_counts(normalize=True)*percentage_
edu_props_df["Pre-diabetic"] = pre_diabetic["Education"].value_counts(normalize=True)*pe
edu_props_df["Non-diabetic"] = non_diabetic["Education"].value_counts(normalize=True)*pe
edu_props_df = edu_props_df.sort_index()

#reset the index and transpose it and rename each education category to more appropriate 
edu_props_df = edu_props_df.T.reset_index()
edu_props_df = edu_props_df.rename(columns={"index" : "Diabetes Status", 1.0:"No School/I

print("\n \n Education Proportion Dataframe")
display(edu_props_df)
```

```
NoDocbcCost Proportion Dataframe
```

	Diabetes Status	No	Yes
0	Diabetic	89.413229	10.586771
1	Pre-diabetic	87.065429	12.934571
2	Non-diabetic	92.038951	7.961049

```
Income Proportion Dataframe
```

	Diabetes Status	Greater than 75,000	35,000 - 50,000	50,000 - 75,000	25,000 - 35,000	20,000 - 25,000	15,000 - 20,000	10,000 - 15,000	Less than 10,000
0	Diabetic	20.355910	14.969162	14.895603	12.742602	11.469473	10.094494	8.730832	6.741923
1	Pre-diabetic	21.831138	16.152019	15.871302	12.675448	9.911466	9.090909	7.687325	6.780393
2	Non-diabetic	38.454771	14.239856	17.416227	9.729391	7.310145	5.617609	3.903080	3.328919

```
Education Proportion Dataframe
```

	Diabetes Status	No School/Kindergarten Only	Grades 1-8	Grades 9-11	High School Graduate	1-3 Years College Education	4+ Years College Education
0	Diabetic	0.132971	3.346913	6.495785	31.307644	29.293272	29.423414
1	Pre-diabetic	0.043187	3.476571	6.780393	29.151371	28.784280	31.764198
2	Non-diabetic	0.058492	1.262968	3.213806	23.553249	27.244821	44.666664

Wrangling the socioeconomic dataframe into education categories and also income categories which are then converted into proportions/percentages to plot the subplots down below

In [ ]:

```
#want total amounts for each category so you can get proportions/percentages equal to 10

#get the total education and income counts for the entire survey
totalpop_education = diabetes_socioeconomic_df.groupby("Education").count()
totalpop_income = diabetes_socioeconomic_df.groupby("Income").count()


#get the total education and income counts for the non diabetic group
nondiapop_education = non_diabetic.groupby("Education").count()
nondiapop_income = non_diabetic.groupby("Income").count()


#get the total education and income counts for the diabetic group
diapop_education = diabetic.groupby("Education").count()
diapop_income = diabetic.groupby("Income").count()
```

```
#get the total education and income counts for the prediabetic group
prepop_education = pre_diabetic.groupby("Education").count()
prepop_income = pre_diabetic.groupby("Income").count()


##
## MAKING THE NEW DATAFRAME FOR PERCENTAGES IN EACH EDUCATION CATEGORY FOR PLOTTING IN SU
##

#drop the unnecessary columns for total education count dataframe
#list of columns to drop for total education count dataframe
column_drop1 = ["NoDocbcCost", "Income"]
totalpop_education = totalpop_education.drop(column_drop1, axis = 1)

#make a new df for the percentages of each education category for each diabetes status gr
education_subplot_perc = pd.DataFrame()

#set the value to multiply by to get percentage from the proportion gathered. This will b
percentage_mult = 100

#add new percentage columns into new df and drop unnecesarry columns using the columndrop
education_subplot_perc["Diabetic"] = diapop_education.drop(column_drop1, axis = 1) / tota
education_subplot_perc["Pre-diabetic"] = prepop_education.drop(column_drop1, axis = 1) /
education_subplot_perc["Non-diabetic"] = nondiapop_education.drop(column_drop1, axis = 1)

#reset the index and transpose and rename each education category to more appropriate nam
#puts it in proper orientation for plotting
education_subplot_perc = education_subplot_perc.T.reset_index()
education_subplot_perc = education_subplot_perc.rename(columns={"index" : "Diabetes Statu


##
## MAKING THE NEW DATAFRAME FOR PERCENTAGES IN EACH INCOME CATEGORY FOR PLOTTING IN SUBPL
##

#drop the unnecessary columns for total income count dataframe
#list of columns to drop for total income count dataframe
column_drop2 = ["NoDocbcCost", "Education"]
totalpop_income = totalpop_income.drop(column_drop2, axis = 1)

#make a new df for the percentages of each income category for each diabetes status group
income_subplot_perc = pd.DataFrame()
#add new percentage columns into new df and drop unnecesarry colummns using the columndro
income_subplot_perc["Diabetic"] = diapop_income.drop(column_drop2, axis = 1) / totalpop_i
income_subplot_perc["Pre-diabetic"] = prepop_income.drop(column_drop2, axis = 1) / totalp
income_subplot_perc["Non-diabetic"] = nondiapop_income.drop(column_drop2, axis = 1) / tot

#reset the index and transpose and rename each education category to more appropriate nam
#puts it in proper orientation for plotting
income_subplot_perc = income_subplot_perc.T.reset_index()
income_subplot_perc = income_subplot_perc.rename(columns={"index": "Diabetes Status", 1.0



#Show the end subplot DF results for the education and income percentages
print("Percentages of Each Education Category totaling to 100% from each Diabetes Status
display(education_subplot_perc)

print("Percentages of Each Income Category totaling to 100% from each Diabetes Status Gro
display(income_subplot_perc)
```

```
Percentages of Each Education Category totaling to 100% from each Diabetes Status Groups
```

Education	Diabetes Status	No School/Kindergarten Only	Grades 1-8	Grades 9-11	High School Graduate	1-3 Years College Education	4+ Years College Education
0	Diabetic	27.011494	29.260450	24.224520	17.635060	14.810471	9.690193
1	Pre-diabetic	1.149425	3.982191	3.312935	2.151394	1.906737	1.370603
2	Non-diabetic	71.839080	66.757358	72.462545	80.213546	83.282792	88.939203

```
Percentages of Each Income Category totaling to 100% from each Diabetes Status Groups
```

Income	Diabetes Status	Less than 10,000	10,000 - 15,000	15,000 - 20,000	20,000 - 25,000	25,000 - 35,000	35,000 - 50,000	50,000 - 75,000	(
0	Diabetic	24.289063	26.190274	22.308366	20.134095	17.401383	14.507815	12.182142	7.
1	Pre-diabetic	3.200489	3.021302	2.632237	2.279613	2.267898	2.051001	1.700641	1.
2	Non-diabetic	72.510447	70.788424	75.059397	77.586293	80.330719	83.441185	86.117217	90.

Visualizing the wrangled/cleaned data for socioeconomic factors

After wrangling and cleaning the data, we were then able to visualize all of the data for this specific guiding question.

To look within the three diabetes status groups, the percentages of the three diabetic groups (so each group totalled up to 100% separately) were graphed on a bar plot together. This was done for the three variables of interest: education, income and inability to see a doctor due to cost.

To look between the three diabetes status groups for education and income, we made subplots for each of the categories in the two variables. Each category totalled up to 100% which allowed us to make comparisons between the three diabetes status groups for each category.

Below each plot there will be a markdown cell which mentions what we see for each visualization/plot.

Visualize Percentages of Inability to see a Doctor due to Cost for diabetic, pre-diabetic and non-diabetic responses using a bar chart

In [ ]:

```
#making a figure looking at the ability to see doctor in past 12 months based on differen
fig = go.Figure()

#preset the number of decimals you want to round the values that are visible on the barpl
round_value = 2

#add specific columns using fig.add_trace to make the entire barplot. x sets x value base

fig.add_trace(go.Bar(
    x=nodoc_props_df["Diabetes Status"],
    y=nodoc_props_df["Yes"],
    name="Unable to see Doctor",
    marker_color="#fdbb84",
    text=nodoc_props_df["Yes"].round(round_value) #text shows the percentage values on th

))
fig.add_trace(go.Bar(
    x=nodoc_props_df["Diabetes Status"],
    y=nodoc_props_df["No"],
    name="Able to see doctor",
    marker_color="#e34a33",
    text=nodoc_props_df["No"].round(round_value)

))
#add plot titles
fig.update_layout(
    title = "Ability to see Doctor in Past 12 Months due to Cost versus Diabetes Status"
    xaxis_title = "Diabetes Status",
    yaxis_title = "Proportion"
)

fig.show()
```

For above plot:

From the above plot we see that the percentage of people that are able to see a doctor is very high within each of the three diabetes status groups. This also shows that there is a large discrepancy between inability to see a doctor due to cost and the ability to see a doctor because of those huge percentage differences between the two categories in each of the three diabetes status groups. The unable to see a doctor category were both slightly higher for the diabetic and pre-diabetic group compared to the non-diabetic group. The highest percentage difference was 4.97% between the diabetic group and nondiabetic group for Unable to see a doctor due to cost.

Overall, while the majority of each diabetes status groups are able to pay to see a doctor, the diabetes and pre-diabetes groups do have slightly higher percentages for people unable to see a doctor compared to the non diabetic group.

Visualize the Education Percentages of the three diabetes status groups (diabetic, pre-diabetic and non-diabetic) using a Bar Plot

In [ ]:

```
#GRAPH OF PERCENTAGE OF HIGHEST LEVEL OF EDUCATION BASED ON DIABETES STATUS
```

```
fig = go.Figure()

#add specific columns using fig.add_trace to make the entire barplot. x sets x value base

fig.add_trace(go.Bar(
    x=edu_props_df["Diabetes Status"],
    y=edu_props_df["No School/Kindergarten Only"],
    name="No School/Kindergarten Only",
    marker_color="#ffffcc"
))
fig.add_trace(go.Bar(
    x=edu_props_df["Diabetes Status"],
    y=edu_props_df["Grades 1-8"],
    name="Grades 1-8",
    marker_color="#d9f0a3"
))
fig.add_trace(go.Bar(
    x=edu_props_df["Diabetes Status"],
    y=edu_props_df["Grades 9-11"],
    name="Grades 9-11",
    marker_color="#addd8e"
))
fig.add_trace(go.Bar(
    x=edu_props_df["Diabetes Status"],
    y=edu_props_df["High School Graduate"],
    name="High School Graduate",
    marker_color="#78c679"
))
fig.add_trace(go.Bar(
    x=edu_props_df["Diabetes Status"],
    y=edu_props_df["1-3 Years College Education"],
    name="1-3 Years College Education",
    marker_color="#31a354"
))
fig.add_trace(go.Bar(
    x=edu_props_df["Diabetes Status"],
    y=edu_props_df["4+ Years College Education"],
    name="4+ Years College Education",
    marker_color="#006837"
))
#set titles for the plot
fig.update_layout(
    title="Highest Level of Education versus Diabetes Status",
    xaxis_title="Diabetes Status",
    yaxis_title="Percentage(%)",
    legend_title = "Education Levels")
fig.show()
```

For the above plot:

From the above plot, we see within the diabetic group that the percentage increases as you move up education levels where it eventually reaches a peak at the high school graduate level. After reaching this level, the percentage remains relatively level for the higher education levels afterwards. The pre-diabetic group shows a similar case where it increases and the percentage starts to remain relatively level at the three highest education levels. The only difference being that the highest percentage education level for the pre-diabetic group is 4+ years of college education.

For the non-diabetic group, we see the percentage of individuals increase all the way until reaching the highest education level, 4+ years of college education.

From this plot we can say that within each diabetes status group, there are a higher percentage of individuals with a higher education level in comparison to individuals with a lower education level. We cannot make any comparisons with research until we look at all three groups together, which is seen in the subplot below.

Visualize the subplot of the three diabetes status groups for Education Categories

In [ ]:

```
#SUBPLOTS OF ALL EDUCATION LEVEL CATEGORIES FOR DIABETES STATUS

#preset the number of decimals you want to round the values to that are visible on the bo
round_value = 2

#set row/column orientation for how the subplots are shown
fig = make_subplots(rows=2, cols=3, start_cell="bottom-left",shared_xaxes=True)

#add specific columns using fig.add_trace to make the entire barplot. x sets x value base

fig.add_trace(go.Bar(
    x=education_subplot_perc["Diabetes Status"],
    y=education_subplot_perc["No School/Kindergarten Only"],
    name="No School/Kindergarten Only",
    marker_color="#ffffcc",
    text=education_subplot_perc["No School/Kindergarten Only"].round(round_value)), row=

fig.add_trace(go.Bar(
    x=education_subplot_perc["Diabetes Status"],
    y=education_subplot_perc["Grades 1-8"],
    name="Grades 1-8",
    marker_color="#d9f0a3",
    text=education_subplot_perc["Grades 1-8"].round(round_value)), row=1, col=2)

fig.add_trace(go.Bar(
    x=education_subplot_perc["Diabetes Status"],
    y=education_subplot_perc["Grades 9-11"],
    name="Grades 9-11",
    marker_color="#addd8e",
    text=education_subplot_perc["Grades 9-11"].round(round_value)), row=1, col=3)

fig.add_trace(go.Bar(
    x=education_subplot_perc["Diabetes Status"],
    y=education_subplot_perc["High School Graduate"],
    name="High School Graduate",
    marker_color="#78c679",
    text=education_subplot_perc["High School Graduate"].round(round_value)), row=2, col=

fig.add_trace(go.Bar(
    x=education_subplot_perc["Diabetes Status"],
    y=education_subplot_perc["1-3 Years College Education"],
    name="1-3 Years College Education",
    marker_color="#31a354",
    text=education_subplot_perc["1-3 Years College Education"].round(round_value)), row=

fig.add_trace(go.Bar(
    x=education_subplot_perc["Diabetes Status"],
    y=education_subplot_perc["4+ Years College Education"],
    name="4+ Years College Education",
    marker_color="#006837",
    text=education_subplot_perc["4+ Years College Education"].round(round_value)), row=2

#add titles for plot
fig.update_layout(
    title="Highest Level of Education versus Diabetes Status",
    xaxis_title="Diabetes Status",
    yaxis_title="Percentage(%)",
    legend_title = "Education Levels")

#set figure height and y axis ticks
fig.update_layout(height = 700)
fig.update_yaxes(tick0=0, dtick=10)
fig.show()
```

For the plot above:

To take a closer look between the diabetes status groups we took total proportions/percentages for each education level (so each category adds up to 100%). We can see for each level that the majority of individuals are from the non-diabetic group since they have a very high percentage compared to the other two groups. We can also note however that as we move up education levels, that the percentage of diabetic individuals slowly decreases, starting at 27.01% at the No school/kindergarten only education level and going all the way down to 9.69% at the 4+ years college education level. As this occurs, the percentage of non-diabetic individuals also increases as you go up education levels. Their percentage starts at 71.84% for No school/Kindergarten only education level and goes all the way up to 88.94% at the 4+ years college education level. The percentage of pre-diabetics shows no real trend as you move between the education levels and instead sticks to around 1 to 4%.

For this subplot which looks at each education category for all three diabetes status groups together, we can see that the decrease in the percentage of diabetics as you move up education levels correlates with the trend seen in research which states that the percentage of diabetes is higher in lower education levels compared to higher education levels [19].

Visualize the Income Percentages of the three diabetes status groups (diabetic, pre-diabetic and non-diabetic) using a Bar Plot

In [ ]:

```
#GRAPH OF PERCENTAGE OF INCOME LEVELS BASED ON DIABETES STATUS


fig = go.Figure()
```

```
#add specific columns using fig.add_trace to make the entire barplot. x sets x value base

fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["Less than 10,000"],
    name="Less than 10,000",
    marker_color="#ffffd9"
))
fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["10,000 - 15,000"],
    name="10,000 - 15,000",
    marker_color="#edf8b1"
))
fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["15,000 - 20,000"],
    name="15,000 - 20,000",
    marker_color="#c7e9b4"
))
fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["20,000 - 25,000"],
    name="20,000 - 25,000",
    marker_color="#7fcdbb"
))
fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["25,000 - 35,000"],
    name="25,000 - 35,000",
    marker_color="#41b6c4"
))
fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["35,000 - 50,000"],
    name="35,000 - 50,000",
    marker_color="#1d91c0"
))
fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["50,000 - 75,000"],
    name="50,000 - 75,000",
    marker_color="#225ea8"
))
fig.add_trace(go.Bar(
    x=income_props_df["Diabetes Status"],
    y=income_props_df["Greater than 75,000"],
    name="Greater than 75,000",
    marker_color="#0c2c84"
))

#add plot titles
fig.update_layout(
    title="Income Brackets versus Diabetes Status",
    xaxis_title="Diabetes Status",
    yaxis_title="Percentage(%)",
    legend_title = "Income Brackets")

fig.show()
```

For the plot above:

From the plot above, we can see within each of the diabetes status group a similar trend where the percentages of individuals increases as you move up income brackets. The highest percentage income bracket for all three of the groups is the greater than 75,000 income bracket.

From this we can say that there is a higher percentage of individuals at the higher income brackets in each of the three diabetes status groups (diabetic, pre-diabetic and non-diabetic). We cannot make any comparisons with research until we look at all three groups together, which is seen in the subplot below.

Visualize the subplot of the three diabetes status groups for Income Categories

In [ ]:

```
#SUBPLOT OF ALL INCOME BRACKETS BASED ON DIABETES STATUS

#preset the number of decimals you want to round the values to that are visible on the bo
round_value = 2

#set figure orientation for the subplots
fig = make_subplots(rows=2, cols=4, start_cell="bottom-left",shared_xaxes=True)

#add specific columns using fig.add_trace to make the entire barplot. x sets x value base

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["Less than 10,000"],
    name="Less than 10,000",
    marker_color="#ffffd9",
    text=income_subplot_perc["Less than 10,000"].round(round_value)), row=1, col=1) #tex

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["10,000 - 15,000"],
    name="10,000 - 15,000",
    marker_color="#edf8b1",
    text=income_subplot_perc["10,000 - 15,000"].round(round_value)),row=1, col=2)

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["15,000 - 20,000"],
    name="15,000 - 20,000",
    marker_color="#c7e9b4",
    text=income_subplot_perc["15,000 - 20,000"].round(round_value)), row=1, col=3)

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["20,000 - 25,000"],
    name="20,000 - 25,000",
    marker_color="#7fcdbb",
    text=income_subplot_perc["20,000 - 25,000"].round(round_value)),  row=1, col=4)

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["25,000 - 35,000"],
    name="25,000 - 35,000",
    marker_color="#41b6c4",
    text=income_subplot_perc["25,000 - 35,000"].round(round_value)), row=2, col=1)

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["35,000 - 50,000"],
    name="35,000 - 50,000",
    marker_color="#1d91c0",
    text=income_subplot_perc["35,000 - 50,000"].round(round_value)),  row=2, col=2)

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["50,000 - 75,000"],
    name="50,000 - 75,000",
    marker_color="#225ea8",
    text=income_subplot_perc["50,000 - 75,000"].round(round_value)), row=2, col=3)

fig.add_trace(go.Bar(
    x=income_subplot_perc["Diabetes Status"],
    y=income_subplot_perc["Greater than 75,000"],
    name="Greater than 75,000",
    marker_color="#0c2c84",
    text=income_subplot_perc["Greater than 75,000"].round(round_value)),  row=2, col=4)

#add plot titles
fig.update_layout(
    title="Income Brackets versus Diabetes Status",
    xaxis_title="Diabetes Status",
    yaxis_title="Percentage(%)",
    legend_title = "Income Brackets")

#set graph size and set the ticks for the y axis
fig.update_layout(height = 700)
fig.update_yaxes(tick0=0, dtick=10)
fig.show()
```

For the plot above:

To take a closer look between the diabetes status groups we took total proportions/percentages for each income level/bracket (so each bracket adds up to 100%). We can see for each level that the majority of individuals are from the non-diabetic group since they have a very high percentage compared to the other two groups. We can also note however that as we move up income brackets, that the percentage of diabetic individuals slowly decreases, starting at 24.29% at the less than 10,000 income bracket and all the way down to 7.96% at the greater than 75,000 income bracket. The percentage of pre-diabetics also slightly decreases as you move up income brackets. The percentage of pre-diabetics started at 3.2% at the lowest income bracket, less than 10,000, and goes all the way down to 1.12% at the highest income bracket, greater than 75,000. As this occurs, the percentage of non-daibetic individuals also increases as you go up income brackets. Their percentage starts at 72.51% for the less than 10,000 income bracket and goes all the way up to 90.92% at the greater than 75,000 income bracket.

From this subplot which looks at each income bracket for all three diabetes status groups together, we can see that the decrease in the percentage of diabetics as you move up income brackets correlates with the trend seen in research which states that diabetes percentages decrease the higher the income level [20].

Conclusion to the Socioeconomic guiding question

From the visualizations for the Inability to see a doctor due to cost variable, we see that the majority of individuals in all three diabetes status groups are able to pay to see a doctor however there are slightly higher percentages of individuals unable to see a doctor due to cost in the diabetic and pre-diabetic groups compared to the non-diabtic group.

From the visualizations for the Education variable, within each diabetes status group, we saw higher percentages of individuals in the higher education levels compared to lower levels. We also observed from the subplot that the data follows research claims which states that the percentage of diabetics when grouped with pre-diabetics and non-diabetics is higher in lower education levels [19].

From the visualization for the Income variable, within each diabetes status group, we saw higher percentages of individuals in the higher income brackets compared to the lower income brackets. We also observed from the subplot that the data follows research claims which states that the percentage of diabetics when grouped with pre-diabetic and non-diabetics is higher in lower income brackets [20].

Overall, we can see that Socioeconomic status variables on their own do play a role on diabetes status. Further analysis would involve combining more than one variable at a time to see if this still holds true when other variables such as age, sex are included with socioeconomic factors.

Extra Analysis of the Dataset

From the discussion board, and to further the analysis, we decided to dive in deeper into an additional variable that we had not yet explored: BMI.

BMI or the body mass index is obtained by dividing the person’s weight by the square of height in meters. Research shows that it is correlated with various metabolic diseases like type 2 diabetes[21].

The ranges provided by the CDC are as follows[21]:

Below 18.5: Underweight

18.5-24.9: Healthy weight

25.0 - 29.9: Overweight

30 and above: Obesity

1.1 Data wrangling

To inspect the data, we created a separate dataframe for the original to include the variable BMI plus lifestyle choices. Additionally, we changed some of the variable names and created new columns to better suit the analysis.

In [ ]:

```
#Make the data frame smaller to just focused on the variables I will be working with
diabetes2 = diabetes_original.drop(columns=['HighBP', 'HighChol','CholCheck', 'Stroke','
'GenHlth','MentHlth','PhysHlth','Sex','Age','Education','Income','DiffWalk'])
```

In [ ]:

```
#Add column that sums dietary habits related columns
diabetes2['Dietary Habits']=diabetes2['Fruits']+diabetes2['Veggies']

#Changing variable names
diabetes2.loc[diabetes2['Diabetes_012'] == 0.0, 'Diabetes_012'] = 'Non-diabetic'
diabetes2.loc[diabetes2['Diabetes_012'] == 1.0, 'Diabetes_012'] = 'Pre-diabetic'
diabetes2.loc[diabetes2['Diabetes_012'] == 2.0, 'Diabetes_012'] = 'Diabetic'
diabetes2.loc[diabetes2['Smoker']==0.0, 'Smoker'] = 'No'
diabetes2.loc[diabetes2['Smoker']==1, 'Smoker'] = 'Yes'
diabetes2.loc[diabetes2['PhysActivity']==0.0, 'PhysActivity'] = 'No'
diabetes2.loc[diabetes2['PhysActivity']==1, 'PhysActivity'] = 'Yes'
diabetes2.loc[diabetes2['Fruits']==0.0, 'Fruits'] = 'No'
diabetes2.loc[diabetes2['Veggies']==0.0, 'Veggies'] = 'No'
diabetes2.loc[diabetes2['Veggies']==1, 'Veggies'] = 'Yes'
diabetes2.loc[diabetes2['Fruits']==1, 'Fruits'] = 'Yes'
diabetes2.loc[diabetes2['Dietary Habits']==0, 'Dietary Habits'] = 'None'
diabetes2.loc[diabetes2['Dietary Habits']==1, 'Dietary Habits'] = 'Fruit Or Veggies'
diabetes2.loc[diabetes2['Dietary Habits']==2, 'Dietary Habits'] = 'Fruit AND Veggies'

#Rename columns for graphs
diabetes2 = diabetes2.rename(columns={'Diabetes_012':'Diabetes Status', 'PhysActivity':'

display(diabetes2.head())
```

	Diabetes Status	BMI	Smoker	Physical Activity	Fruits	Veggies	HvyAlcoholConsump	Dietary Habits
0	Non-diabetic	40.0	Yes	No	No	Yes	0.0	Fruit Or Veggies
1	Non-diabetic	25.0	Yes	Yes	No	No	0.0	None
2	Non-diabetic	28.0	No	No	Yes	No	0.0	Fruit Or Veggies
3	Non-diabetic	27.0	No	Yes	Yes	Yes	0.0	Fruit AND Veggies
4	Non-diabetic	24.0	No	Yes	Yes	Yes	0.0	Fruit AND Veggies

1.2 Visualizing the data

In [ ]:

```
#Create violin plot to compare BMI between populations
fig = px.violin(diabetes2,
                y="BMI",
                x='Diabetes Status',
                box=True,
                color ='Diabetes Status',
                color_discrete_sequence = ['blue','red','purple'],
                title='BMI Distribution per Diabetic Status'
               )
fig.show()
```

We can observe from the violin plot that non-diabetics have the lowest median BMI (27) followed by pre-diabetic (30) and finally diabetics (31). However, according to the CDC's guidelines, the median BMI of non-diabetics fall in the overweight category. The median BMI of prediabetics and diabetics falls in the obesity category.

Following this plot, we were curious to see if reporting being physically active had an impact in the BMI:

In [ ]:

```
#Plot for BMI and Physical Activity
fig = px.violin(diabetes2,
                y="BMI",
                x='Physical Activity' ,
                box=True,
                color ='Physical Activity',
                color_discrete_sequence = ['red','blue'],
                title='BMI Distribution For People Who Are Physically Active And People W

               )
fig.show()
```

From the graph above we can see that people who do not engage in physical activity have a median BMI of 29 and people who do engage in physical activity have a median BMI of 27. So there is a difference between ebing active or not, but still the median in both categories falls in the overweight category.

Lastly, we wanted to investigate dietary habits and the impact on BMI.

In [ ]:
```
fig = px.violin(diabetes2,
                y="BMI",
                x='Dietary Habits' ,
                box=True,
                color ='Dietary Habits',
                title = 'BMI Distribution by Fruit and Veggie Consumption'
               )
fig.show()
```

As seen in the graph above, the median BMI for people who do not consume either veggies or fruits, and people who only consume one, is the same (28). On the other hand, people who consume both fruits and veggies have a lower median BMI (27).

For a future project, it would be interesting to see if socioeconomic variables have an effect on people choosing to engage in healthy lifestyle habits or unhealthy lifestyle habits.

2.1 Counting Number of Comorbidities

For this analysis, we will look into how the various health conditions correlate with each other and diabetes. The conditions in the dataset are high blood pressure (HighBP), high cholesterol (HighChol), stroke (Stroke), and heart disease (HeartDiseaseorAttack). Here we will examine the proportions of people with each diabetes status who have different counts of conditions, along with which conditions seem most strongly correlated with each other. First, we will examine the distribution of the number of health-related conditions by diabetes status. Someone with two of these other comorbidities would have a value of 2; someone with all 4 would have a value of 4.

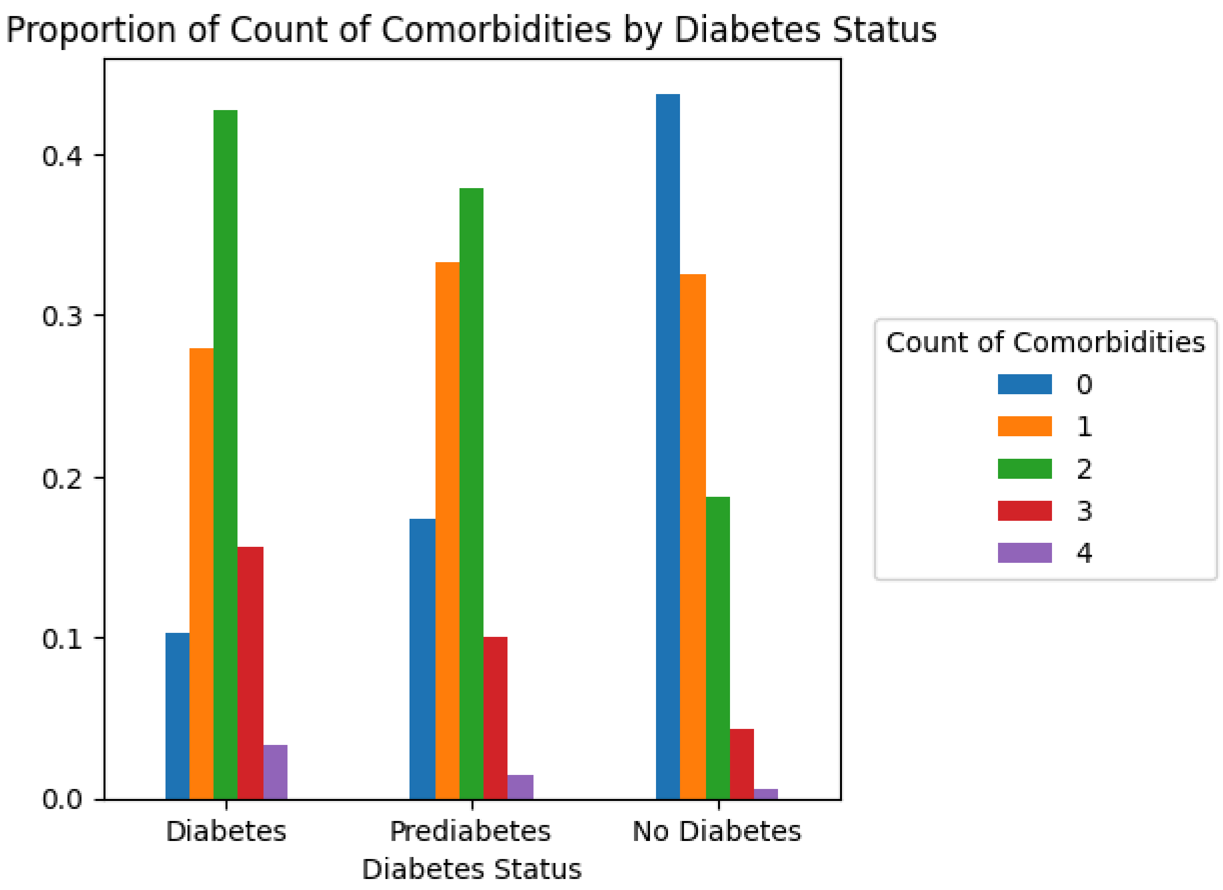
In [ ]:
```
#Get data of interest
healthData = converteddata.filter(["Diabetes_012","HighBP", "HighChol", "Stroke", "HeartI

#make a new column that is sum of rows aka count of illnesses
totalIllnesses = []
for idx,row in healthData.iterrows():
    totalIllnesses.append(sum([row.HighBP, row.HighChol, row.Stroke, row.HeartDiseaseorA

#append to the total illness count
healthData["Total Illness Count"] = totalIllnesses
healthData.head()
```

Out[ ]:

	Diabetes_012	HighBP	HighChol	Stroke	HeartDiseaseorAttack	Total Illness Count
0	0	1	1	0	0	2
1	0	0	0	0	0	0
2	0	1	1	0	0	2
3	0	1	0	0	0	1
4	0	1	1	0	0	2

In [ ]:

```
#Make figures:
illnessCounts = range(0,5)

nodiabetesCount = len(healthData[healthData["Diabetes_012"] == 0])
prediabetesCount = len(healthData[healthData["Diabetes_012"] ==1])
diabetesCount = len(healthData[healthData["Diabetes_012"] ==2])

diabetesData = healthData[healthData["Diabetes_012"] == 2].filter(["Diabetes_012", "Total
prediabetesData = healthData[healthData["Diabetes_012"] == 1].filter(["Diabetes_012", "To
nodiabetesData = healthData[healthData["Diabetes_012"] == 0].filter(["Diabetes_012", "To

#for each of these: groupby total illness count, then take the number and divide them by
diabetesProps = diabetesData.groupby("Total Illness Count").count()["Diabetes_012"] / dia
prediabetesProps = prediabetesData.groupby("Total Illness Count").count()["Diabetes_012"]
nodiabetesProps = nodiabetesData.groupby("Total Illness Count").count()["Diabetes_012"] /

#Turn this into a dataframe
healthPlotFrame = pd.DataFrame([diabetesProps, prediabetesProps, nodiabetesProps])
healthPlotFrame["Diabetes Status"] = ["Diabetes", "Prediabetes", "No Diabetes"]
healthPlotFrame = healthPlotFrame.filter(["Diabetes Status", 0, 1, 2, 3, 4])
healthPlotFrame.head()

#Make plots
fig, ax = plt.subplots()

healthPlotFrame.plot(
    ax = ax,
    x = "Diabetes Status",
    kind = "bar",
    title = "Proportion of Count of Comorbidities by Diabetes Status",
    rot=0,
    legend = False
)

fig.legend(title = "Count of Comorbidities", loc = "center right")
fig.subplots_adjust(right=0.70)
plt.show()
```



We can see from this plot that people with no diabetes have the highest proportion of people with none of the other health conditions. For people with diabetes, over 40% of them seem to have two of these comorbidities. You can see a decrease in proportion of those with no diabetes as the count of comorbidities increases, but we do not see this same trend for those with diabetes and prediabetes, both of which have the highest proportion of people have two or more comorbidities.

This is consistent with our previous analysis stating that people with diabetes are at a greater risk for all analyzed health conditions, along with the previously addressed literature [12,13,15,16]

This plot does not tell us which conditions make up these numbers, so we have done more analysis to explore this further.

2.2 Visualizing Correlation of Health-Related Variables

Next, we will make a heatmap to show how the different variables (diabetes status and all the addressed health conditions) correlate with each other.

In [ ]:

```
#Check heatmap to see how the variables correlate
corrmatrix = healthData[["Diabetes_012", "HighBP", "HighChol", "Stroke", "HeartDiseaseor

fig = px.imshow(corrmatrix, text_auto=True, aspect="auto")
fig.show()
```

Here we can see that the strongest correlation with diabetes status is high blood pressure, and the lowest correlation is with stroke. However, all of these variables have correlations with each other, consistent with our previous findings. The strongest overall correlation is between high cholesterol and high blood pressure.

2.3 Visualizing by Diabetes Status and Health Condition(s)

To further examine this grouped by diabetes status, we made a few plots showing the proportion of people with each diabetes status that have each combination of health conditions.

In [ ]:

```
#Get the list of conditions (not including diabetes status, which is first, or total illr
conditionsList = healthData.columns.values[1:len(healthData.columns.values)-1] #["HighBP

#Try to do this in a "generalizable" way.

#Split dataframes based on the total illness count
splitdfs = dict(iter(healthData.groupby("Total Illness Count")))

noIllnesses = splitdfs.get(0)
oneIllness = splitdfs.get(1)
twoIllnesses = splitdfs.get(2)
threeIllnesses = splitdfs.get(3)
fourIllnesses = splitdfs.get(4)

#Get dataframe indicies as a list to make it a bit easier
indices = ["No Diabetes", "Prediabetes", "Diabetes"]

#Start with the no illnesses dataframe: divide list of each diabetes status by the respec
comorbiditiesdf = pd.DataFrame({"None":[len(noIllnesses[noIllnesses["Diabetes_012"] == 0
              len(noIllnesses[noIllnesses["Diabetes_012"] == 1])/prediabetesCount,
              len(noIllnesses[noIllnesses["Diabetes_012"] == 2])/diabetesCount]}, index=


#Do the same and append to dataframe for all "single" conditions:
for condition in conditionsList:
    #subset the data to only consider those with this one condition
    conditionSubset = oneIllness[oneIllness[condition] == 1]
    #make a new dataframe with proportions and concatenate it with the original comorbidi
    newdf = pd.DataFrame({condition:[len(conditionSubset[conditionSubset["Diabetes_012"]
              len(conditionSubset[conditionSubset["Diabetes_012"] == 1])/prediabetesCoun
              len(conditionSubset[conditionSubset["Diabetes_012"] == 2])/diabetesCount]}
    comorbiditiesdf = pd.concat([comorbiditiesdf, newdf], axis=1)



#Do the same for all "paired conditions"

#Keep a list of the condition pairs for later analysis
doubleConditions = []
for first in range(0,len(conditionsList)-1):
    #index of the "first" comparison variable
    for second in range(first+1, len(conditionsList)):
        #index of the second comparison variable
        doubleConditions.append(conditionsList[first]+ "+" + conditionsList[second])
        #subset the data to get those with these two conditions
        conditionSubset = twoIllnesses[(twoIllnesses[conditionsList[first]] == 1) & (two
        #make a new dataframe with proportions and concatenate it with the original comor
        newdf = pd.DataFrame({(conditionsList[first]+ "+" + conditionsList[second]):[len
              len(conditionSubset[conditionSubset["Diabetes_012"] == 1])/prediabetesCoun
              len(conditionSubset[conditionSubset["Diabetes_012"] == 2])/diabetesCount]}
        comorbiditiesdf = pd.concat([comorbiditiesdf, newdf], axis=1)



# Doing the same for all triple conditions

threePlusConditions = []
for first in range(0,len(conditionsList)-2):
    #index of the "first" comparison variable
    for second in range(first+1, len(conditionsList)-1):
        #index of the second comparison variable
        for third in range(second+1, len(conditionsList)):
            #index of the third comparison variable
            threePlusConditions.append(conditionsList[first]+ "+" + conditionsList[secon
            #subset the data to get those with these three conditions
            conditionSubset = threeIllnesses[(threeIllnesses[conditionsList[first]] == 1
            #make a new dataframe with proportions and concatenate it with the original
            newdf = pd.DataFrame({(conditionsList[first]+ "+" + conditionsList[second] +
              len(conditionSubset[conditionSubset["Diabetes_012"] == 1])/prediabetesCoun
              len(conditionSubset[conditionSubset["Diabetes_012"] == 2])/diabetesCount]}
            comorbiditiesdf = pd.concat([comorbiditiesdf, newdf], axis=1)


#All four conditions:
fourDf = pd.DataFrame({"HighBP+HighChol+Stroke+HeartDiseaseorAttack":[len(fourIllnesses[
              len(fourIllnesses[fourIllnesses["Diabetes_012"] == 1])/prediabetesCount,
              len(fourIllnesses[fourIllnesses["Diabetes_012"] == 2])/diabetesCount]}, in

#Add the four conditions to the dataframe with the list of threePlusConditions, for later
threePlusConditions.append("HighBP+HighChol+Stroke+HeartDiseaseorAttack")
#concatenate it with the original comorbiditiesdf
comorbiditiesdf = pd.concat([comorbiditiesdf, fourDf], axis=1)

#view to ensure output is reliable
comorbiditiesdf
```

Out[ ]:

	None	HighBP	HighChol	Stroke	HeartDiseaseorAttack	HighBP+HighChol	H
No Diabetes	0.437678	0.149638	0.162024	0.004371	0.009509	0.157494	
Prediabetes	0.173397	0.162600	0.158497	0.003023	0.008422	0.326711	
Diabetes	0.102756	0.161744	0.102699	0.003621	0.011826	0.363747	

In [ ]:

```
#initial plot: grouped by diabetes status
fig = px.bar(comorbiditiesdf.reset_index(), barmode="group", x = "index", y = comorbidit
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=Fals
fig.show()
```

This plot is a little difficult to understand and compare variables directly, so we instead grouped this with the condition as the x-axis and the diabetes status within each group.

In [ ]:

```
#above is unclear, group instead by condition types
fig = px.bar(comorbiditiesdf.T.reset_index(), barmode="group", x = "index", y = comorbid
fig.update_traces(textfont_size=8, textangle=0, textposition="outside", cliponaxis=False
fig.show()
```

From this, we can see that one of the most prominent combinations across all diabetes status is high blood pressure and high cholesterol. Surprisingly, we found that blood pressure on its own seemed consistent between all three diabetes statuses, but once combined, prediabetes and diabetes saw an increase in proportions compared to non-diabetes.

The lowest individual variables are stroke and heart disease or attack. There are a few reasons that this could be, but one thing to consider is that this is self-reported survey data. As a result, it may be underestimating the proportions for events like strokes and heart attacks, as someone who dies because of these events would be unable to report it to the survey. This would most strongly impact the proportion of people who had strokes, who would only be considered as people who survived a stroke.

Since some of these variables are comparatively very small, we broke this up further into groups for the combinations of two conditions, and another plot for those with three or more of these conditions.

In [ ]:

```
#since many values are small, make smaller plots focusing on those with two conditions:
fig = px.bar(comorbiditiesdf[doubleConditions].T.reset_index(), barmode="group", x = "in
fig.update_traces(textfont_size=8, textangle=270, textposition="outside", cliponaxis=Fal
fig.show()
```

Within this plot, we again see that high blood pressure and high cholesterol is the most prominent across all diabetes types, though prediabetes and diabetes have a higher proportion than non-diabetes. The smallest variables are those that consider stroke, possibly for the reasons mentioned previously.

In [ ]:

```
#those with three or more conditions
fig = px.bar(comorbiditiesdf[threePlusConditions].T.reset_index(), barmode="group", x = 
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=Fals
fig.show()
```

Within this plot, we can see that the highest proportions are across the high blood pressure, high cholesterol, and heart disease group. For all of these, we can see that it follows the same pattern with diabetes having the largest proportion, followed by prediabetes, and no diabetes as the smallest. This is consistent with our expectations from our previous analysis

Further analysis of this topic should probably consider alternative data such as hospital data to account for mortality, as these self-reported values may be inaccurate and does not account for people who have died as a result of their condition, potentially skewing the numbers.

Some of the discussion feedback we received asked us how we would further our analysis for socioeconomic variables and make it more in depth. Our answer was to make it more similar to how research analyzes factors and their relation to diabetes, which is to look at more than one variable at a time. With the limited time we had, we decided to do a bit moree in depth analysis.

In this case the variables I used were education and age. I took only the group of survey responses that were in the highest level of education (4+ years of college). I then separated these responses based on their diabetes status and their respective age groups. I also did the same for the education level "High school graduate". I only chose to do two education levels due to time constraints. The reason I chose highschool graduate and not the lowest education level (kindergarten only) was because the age brackets for this survey start at 18 and go up to 80+. It would not make sense to use a kindergarten only level education bracket. I did initially attempt to use the kindergarten level but quickly realized that it was such a small sample size that I had no responses when I split them into their age brackets resulting in NAN values appearing. That is why I chose to do high school education instead.

Down below shows how I wrangled the data for this and then the cells after that show the visualizations.

In [ ]:

```
diabetes_age_edu = diabetes_original[["Diabetes_012","Age","Education"]]
dia_value = 2.0 #preset diabetic response value to a variable to use below
diabetic_age_edu = diabetes_age_edu[diabetes_age_edu["Diabetes_012"] == dia_value]
pre_dia_value = 1.0 #preset pre diabetic response value to a variable to use below
prediabetic_age_edu = diabetes_age_edu[diabetes_age_edu["Diabetes_012"] == pre_dia_value
non_dia_value = 0.0 #preset non diabetic response value to a variable to use below
nondiabetic_age_edu = diabetes_age_edu[diabetes_age_edu["Diabetes_012"] == non_dia_value

high_edu_level = 6.0 #preset highest education level to a variable to use below
diabetic_age_eduhigh = diabetic_age_edu[diabetic_age_edu["Education"] == high_edu_level]
prediabetic_age_eduhigh = prediabetic_age_edu[prediabetic_age_edu["Education"] == high_ed
nondiabetic_age_eduhigh = nondiabetic_age_edu[nondiabetic_age_edu["Education"] == high_ed

hs_edu_level = 4.0 #preset high school level to a variable to use below
diabetic_age_eduhs = diabetic_age_edu[diabetic_age_edu["Education"] == hs_edu_level]
prediabetic_age_eduhs = prediabetic_age_edu[prediabetic_age_edu["Education"] == hs_edu_le
nondiabetic_age_eduhs = nondiabetic_age_edu[nondiabetic_age_edu["Education"] == hs_edu_le
age_highedu_percentage = pd.DataFrame()
age_hsedu_percentage = pd.DataFrame()
percentage_mult = 100

#add the highest level of education percentages of diabetes, pre-dia and non-dia columns
age_highedu_percentage["Diabetic"] = diabetic_age_eduhigh["Age"].value_counts(normalize=
age_highedu_percentage["Pre-diabetic"] = prediabetic_age_eduhigh["Age"].value_counts(nor
age_highedu_percentage["Non-diabetic"] = nondiabetic_age_eduhigh["Age"].value_counts(norm

#sort df to put age brackets in proper/neat order
age_highedu_percentage = age_highedu_percentage.sort_index()
age_highedu_percentage = age_highedu_percentage.T.reset_index()

age_highedu_percentage = age_highedu_percentage.rename(columns={"index" : "Diabetes Statu

print("Percentages of Age brackets for Highest Education Level")
display(age_highedu_percentage)

#add the high school education level percentages of diabetes, pre-dia and non-dia columns
age_hsedu_percentage["Diabetic"] = diabetic_age_eduhs["Age"].value_counts(normalize=True
age_hsedu_percentage["Pre-diabetic"] = prediabetic_age_eduhs["Age"].value_counts(normali
age_hsedu_percentage["Non-diabetic"] = nondiabetic_age_eduhs["Age"].value_counts(normali

age_hsedu_percentage = age_hsedu_percentage.sort_index()
age_hsedu_percentage = age_hsedu_percentage.T.reset_index()

age_hsedu_percentage = age_hsedu_percentage.rename(columns={"index" : "Diabetes Status",

print("Percentages of Age brackets for Highest Level of Education being High School Gradu
display(age_hsedu_percentage)
```

```
Percentages of Age brackets for Highest Education Level
```

	Diabetes Status	18-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59	
0	Diabetic	0.144231	0.384615	0.990385	1.721154	3.240385	4.673077	8.480769	10.769231	16.7
1	Pre-diabetic	0.203943	1.087695	1.155676	3.195105	3.535010	6.934058	8.089735	9.245411	16.9
2	Non-diabetic	1.433151	3.928594	5.748319	7.140612	8.226999	9.352149	10.456345	11.560542	12.7

```
Percentages of Age brackets for Highest Level of Education being High School Graduate
```

	Diabetes Status	18-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59	
0	Diabetic	0.298211	0.325321	0.777155	1.581421	2.512200	5.078619	8.901139	12.714621	15.3
1	Pre-diabetic	0.370370	0.814815	1.259259	2.296296	3.481481	6.222222	9.925926	12.888889	14.1
2	Non-diabetic	3.280089	2.576787	3.804585	4.474113	5.149601	7.259506	10.809791	13.291215	12.3

In [ ]:

```
#visualize just the 4+ years college education level of education for all age brackets gr
fig = go.Figure()

fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["18-24"],
    name = "18-24",
    marker_color = "#a6cee3"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["25-29"],
    name = "25-29",
    marker_color = "#1f78b4"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["30-34"],
    name = "30-34",
    marker_color = "#b2df8a"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["35-39"],
    name = "35-39",
    marker_color = "#33a02c"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["40-44"],
    name = "40-44",
    marker_color = "#fb9a99"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["45-49"],
    name = "45-49",
    marker_color = "#e31a1c"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["50-54"],
    name = "50-54",
    marker_color = "#fdbf6f"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["55-59"],
    name = "55-59",
    marker_color = "#ff7f00"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["60-64"],
    name = "60-64",
    marker_color = "#cab2d6"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["65-69"],
    name = "65-69",
    marker_color = "#6a3d9a"
))
```

```
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["70-74"],
    name = "70-74",
    marker_color = "#ffff99"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["75-79"],
    name = "75-79",
    marker_color = "#b15928"
))
fig.add_trace(go.Bar(
    x = age_highedu_percentage["Diabetes Status"],
    y = age_highedu_percentage["80+"],
    name = "80+",
    marker_color = "black"
))

#add titles for plot
fig.update_layout(
    title="Age Brackets for Individuals with 4+ Years College Level of Education per Diab
    xaxis_title="Diabetes Status",
    yaxis_title="Percentage(%)",
    legend_title = "Age Brackets")

fig.show()
```

For the plot above:

From the above plot for the 4+ year college level of education responses, you can see similar trends between the three diabetes status groups. The trend is that the percentage of individuals increases, eventually reaching a peak at one of the age brackets and then decreasing in percentage afterwards. The diabetic peak is at the 65 to 69 age bracket, the pre-diabetic peak is at the 60-64 age bracket (just barely since the 65-69 age bracket is super close percentage wise) and the non-diabetic peak is at the 60 to 64 age bracket.

This data shows us that the highest percentage of diabetics with 4+ years of college education are in the age bracket 65 to 69.

In [ ]:

```
fig = go.Figure()

#add all individuals columns using fig.add_trace to make the entire barplot. x sets x val
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["18-24"],
    name = "18-24",
    marker_color = "#a6cee3"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["25-29"],
    name = "25-29",
    marker_color = "#1f78b4"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["30-34"],
    name = "30-34",
    marker_color = "#b2df8a"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["35-39"],
    name = "35-39",
    marker_color = "#33a02c"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["40-44"],
    name = "40-44",
    marker_color = "#fb9a99"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["45-49"],
    name = "45-49",
    marker_color = "#e31a1c"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["50-54"],
    name = "50-54",
    marker_color = "#fdbf6f"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["55-59"],
    name = "55-59",
    marker_color = "#ff7f00"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["60-64"],
    name = "60-64",
    marker_color = "#cab2d6"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["65-69"],
    name = "65-69",
    marker_color = "#6a3d9a"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["70-74"],
    name = "70-74",
    marker_color = "#ffff99"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["75-79"],
    name = "75-79",
    marker_color = "#b15928"
))
fig.add_trace(go.Bar(
    x = age_hsedu_percentage["Diabetes Status"],
    y = age_hsedu_percentage["80+"],
    name = "80+",
    marker_color = "black"
))

fig.update_layout(
    title="Age Brackets for Individuals with Highschool Level of Education per Diabetes S
    xaxis_title="Diabetes Status",
    yaxis_title="Percentage(%)",
    legend_title = "Age Brackets")

fig.show()
```

For the plot above:

From the above plot for the Highschool Graduate level of education responses, you can see similar trends between the three diabetes status groups. The trend is that the percentage of individuals increases, eventually reaching a peak at one of the age brackets and then decreasing in percentage afterwards. The diabetic peak is at the 65 to 69 age bracket, the pre-diabetic peak is at the 65-69 age bracket and the non-diabetic peak is the only one at the 55 to 59 age bracket. You can also see a slight spike in the 80+ age bracket for the pre-diabetic and non-diabetic group, possibly showing that it may be leveling out after decreasing however since its the last age bracket it is hard to tell.

This data shows us that the highest percentage of diabetics with at most a highschool level of education are in the age bracket 65 to 69.

To make any more conclusions for the extra socioeconomic visualizations made after discussion feedback, we would need to do more analysis of the data (such as visualizing the other eduation levels and their age brackets to compare between all of them and even possibly make subplots for all the education levels grouped by age as well). Unfortunately this further analysis cannot be done within the current timeframe of the course. This could be something we do as a future project outside of this course!

Key Takeaways

1. A large percentage of the population are smokers. Further characterising of smokers based on other demographic features such as gender and age, can help build specific programs targeted to them and work on decreasing smoking habits.
2. The data shows that people with diabetes tend to have a higher risk for high cholesterol, heart disease, high blood pressure, stroke, and mobility difficulties. Therefore, additional screenings

and support for these conditions are necessary for those with diabetes. Upon additional analysis, we also saw that the combinations of high blood pressure and high cholesterol was most prominent within all groups, particularly with those with diabetes or prediabetes. All health conditions are positively correlated with diabetes status.

3. Although the data shows that older ages and males are more at risk of developing Type 2 diabetes, the development of the condition depends on too many other factors that can be addressed by providing health guidance according to age and gender.

4. If using BMI as a tool to measure physical health of an individual, the overall population seems to display high BMI, this being worst for prediabetics and diabetics. People who partake in physical activities and have a diverse diet have lower BMIs than people who do not. Efforts in improving healthy lifestyle choices are necessary to reduce the risk of developing type 2 diabetes in healthy population and to reduce risks associated with obesity in diabetics and prediabetics.

5. Our data shows that socioeconomic status variables do play a role in diabetes status, specifically education and income. Upon extra analysis we used the variable age and two specific education levels, 4+ years college education and high school graduate level. We briefly saw within the diabetic group that the highest percentage was at the 65-69 age bracket for these two education levels.

Future Directions

Some future directions for this project include diving deeper into data, particularly for the multi-category variables of income and education, in order to more thoroughly compare between categories to determine how they influence diabetes status. As well, as mentioned in extra analysis part 2, diving deeper into health conditions with hospital-related data rather than survey data could give a more accurate picture of how different health conditions are related to diabetes status.

As well, this dataset could be used to build a machine learning model to predict diabetes risk. Throughout our exploration, we identified relationships for diabetes status with lifestyle choices, health conditions, demographic variables, and socioeconomic status. Considering all of these factors together, one could make an assessment of diabetes risk with a machine learning model that could classify patients based on the risk factors. This would give individuals a clearer understanding of their own risk of developing diabetes.

References

1. Centers for Disease Control and Prevention. (2022). About chronic diseases. Centers for Disease Control and Prevention. https://www.cdc.gov/chronicdisease/about/index.htm

2. Non communicable diseases. (n.d.). www.who.int. https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases#:~:text=Of%20all%20NCD%20deaths%2C%2077

3. Canada, P. H. A. of. (2022, October 5). Framework for diabetes in Canada. www.canada.ca. https://www.canada.ca/en/public-health/services/publications/diseases-conditions/framework-diabetes-canada.html

4. Diabetes and Diabetes-Related Out-of-Pocket Costs. (n.d.). DiabetesCanadaWebsite. https://www.diabetes.ca/advocacy---policies/advocacy-reports/diabetes-and-diabetes-related-out-of-pocket-costs

5. Albright AL, Gregg EW. Preventing type 2 diabetes in communities across the U.S.: the National Diabetes Prevention Program. Am J Prev Med 2013;44(4 Suppl 4):S346–51.

6. Hulbert, L. R. (2022). Effectiveness of Incentives for Improving Diabetes-Related Health Indicators in Chronic Disease Lifestyle Modification Programs: a Systematic Review and Meta-Analysis. Preventing Chronic Disease, 19. https://doi.org/10.5888/pcd19.220151

7. Centers for Disease Control and Prevention (CDC). (2015). Behavioral Risk Factor Surveillance System Survey Data. Atlanta, Georgia: U.S. Department of Health and Human Services, Centers for Disease Control and Prevention.

8. Teboul, A. (2021, November 8). Diabetes health indicators dataset. Kaggle. https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset

9. Behavioral Risk Factor Surveillance System. 2015 Codebook Report. https://www.cdc.gov/brfss/annual_data/2015/pdf/codebook15_llcp.pdf.

10. Patel R, Keyes D. Lifestyle Modification for Diabetes and Heart Disease Prevention. StatPearls, Treasure Island (FL): StatPearls Publishing; 2023

11. U.S. Department of Health and Human Services. The Health Consequences of Smoking—50 Years of Progress: A Report of the Surgeon General [PDF – 36 MB]. Atlanta: U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Chronic Disease Prevention and Health Promotion, Office on Smoking and Health, 2014 [accessed 2021 June 15].

12. MediLexicon International. (n.d.). Diabetes and hypertension: Connection, Complications, risks. Medical News Today. https://www.medicalnewstoday.com/articles/317220.

13. Chen, R., Ovbiagele, B., & Feng, W. (2016, April). Diabetes and stroke: Epidemiology, pathophysiology, pharmaceuticals and outcomes. The American journal of the medical sciences. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5298897/.

14. D'Silva, L. J., Lin, J., Staecker, H., Whitney, S. L., & Kluding, P. M. (2016, March). Impact of diabetic complications on balance and falls: Contribution of the vestibular system. Physical therapy. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4774386/

15. Veeramalla V, Madas S. Comparison of lipid levels in the diabetic and non diabetic patients: a study in a tertiary care hospital. International Journal of Advances in Medicine. 2017 Nov 22;4(6):1573–7.

16. Leon BM, Maddox TM. Diabetes and cardiovascular disease: Epidemiology, biological mechanisms, treatment recommendations and future research. World Journal of Diabetes. 2015 Oct 10;6(13):1246–58.

17. Kautzky-Willer A, Leutner M, Harreiter J. Sex differences in type 2 diabetes. Diabetologia. 2023. https://pubmed.ncbi.nlm.nih.gov/36897358/.

18. Huizen, J. The average age of onset for type 2 diabetes. Medical news Today, 2023. https://www.medicalnewstoday.com/articles/317375

19. Borrell LN, Dallo FJ, White K. Education and diabetes in a racially and ethnically diverse population. Am J Public Health. 2006 Sep;96(9):1637–42.

20. Dinca-Panaitescu S, Dinca-Panaitescu M, Bryant T, Daiski I, Pilkington B, Raphael D. Diabetes prevalence and income: results of the canadian community health survey. Health Policy. 2011 Feb;99(2):116–23.

21. CDC. All About Adult BMI. Centers for Disease Control and Prevention 2022. https://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/index.html (accessed October 17, 2023).