Setup Stage

```
# Place your code here
from google.colab import files
import io
import numpy as np


uploaded = files.upload()
```

Browse... No files selected. Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving 2020_1.csv to 2020_1.csv
Saving 2017.csv to 2017.csv
Saving 2018.csv to 2018.csv
Saving 2019.csv to 2019.csv
```

```
import pandas as pd
df17 = pd.read_csv("2017_updated.csv")
df18 = pd.read_csv("2018_updated.csv")
df19 = pd.read_csv("2019_updated.csv")
df20 = pd.read_csv("2020_1_updated.csv")


frames = [ df17, df18, df19, df20]


result = pd.concat(frames)
```

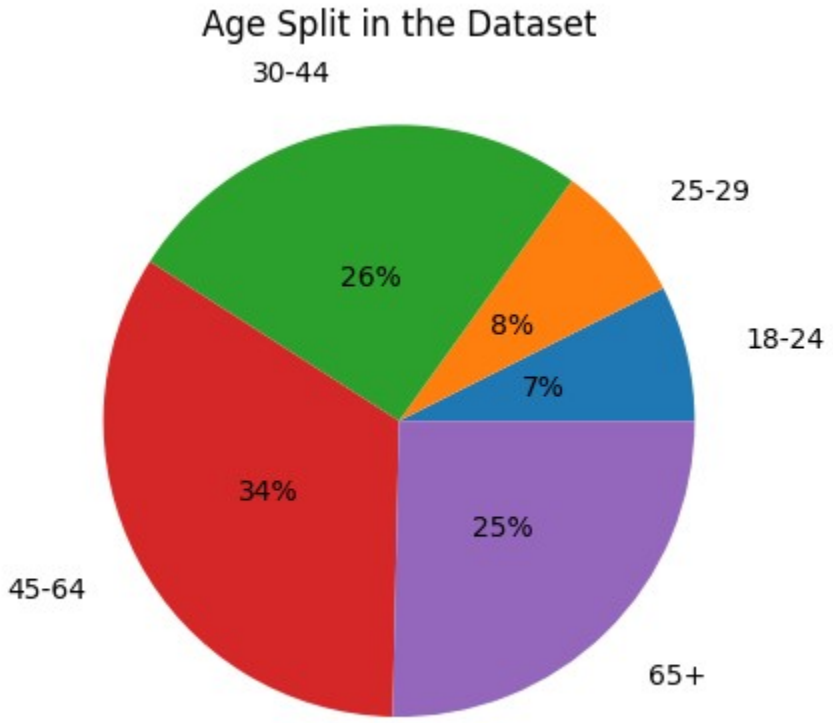
Understanding the Data

In this section, we will be observing how the data is split across various demographic categories and employment categories.

```
import matplotlib.pyplot as plt
age_split = dict()
age_cols = ['agegroup_18_24', 'agegroup_25_29', 'agegroup_30_44', 'agegroup_45_64', 'age
age_columns = ['18-24','25-29','30-44','45-64','65+']
pos = list()
neg = list()

count_by_age = list()
for col in age_cols:
  count_by_age.append(result[col].value_counts()[1])
```

```
plt.title("Age Split in the Dataset")
plt.pie(count_by_age, labels=age_columns, autopct='%1.0f%%', pctdistance=0.5, labeldistar
```

```
([<matplotlib.patches.Wedge at 0x7b9c0c1fb3d0>,
  <matplotlib.patches.Wedge at 0x7b9c0df7add0>,
  <matplotlib.patches.Wedge at 0x7b9c0c24c100>,
  <matplotlib.patches.Wedge at 0x7b9c0c24c790>,
  <matplotlib.patches.Wedge at 0x7b9c0c24ce20>],
 [Text(1.167669789992623, 0.2766717577538113, '18-24'),
  Text(0.9132932418834232, 0.7783928663149907, '25-29'),
  Text(-0.22520803105806614, 1.1786777942877134, '30-44'),
  Text(-1.0585910350693757, -0.5651415932938821, '45-64'),
  Text(0.839441920535312, -0.8575180826361546, '65+')],
 [Text(0.48652907916359295, 0.11527989906408805, '7%'),
  Text(0.3805388507847597, 0.32433036096457946, '8%'),
  Text(-0.09383667960752756, 0.4911157476198806, '26%'),
  Text(-0.4410795979455732, -0.2354756638724509, '34%'),
  Text(0.34976746688971333, -0.35729920109839775, '25%')])
```



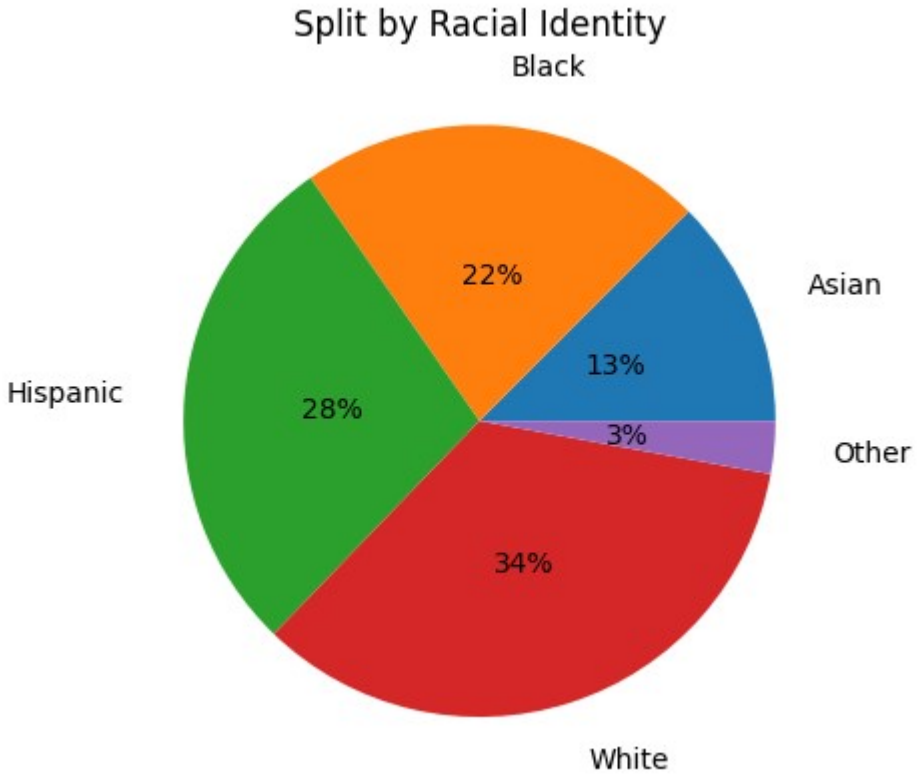
We see that around 60% of the participants of our dataset are 45 and over. This can be attributed to two factors.

1. Increasing Average age in the United States of Americs
2. Older participants are more likely to go through the survey due to the duration of the survey

```
import matplotlib.pyplot as plt
```

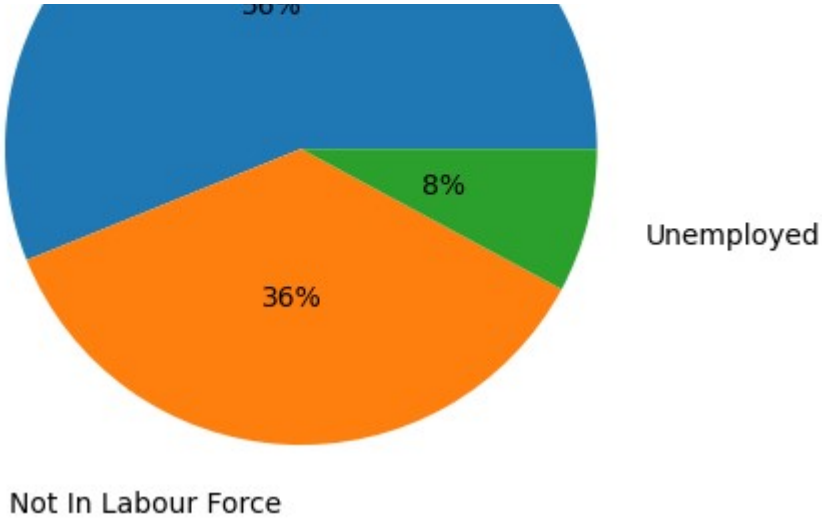
```
race_cols = ['newrace_Asian_Non_hispanic', 'newrace_Black_non_hispanic', 'newrace_Hispan
race_columns = ['Asian','Black','Hispanic','White','Other']

count_by_race = list()
for col in race_cols:
  count_by_race.append(result[col].value_counts()[1])

plt.pie(count_by_race, labels=race_columns, autopct='%1.0f%%', pctdistance=0.5, labeldis
plt.title("Split by Racial Identity")
```

```
Text(0.5, 1.0, 'Split by Racial Identity')
```



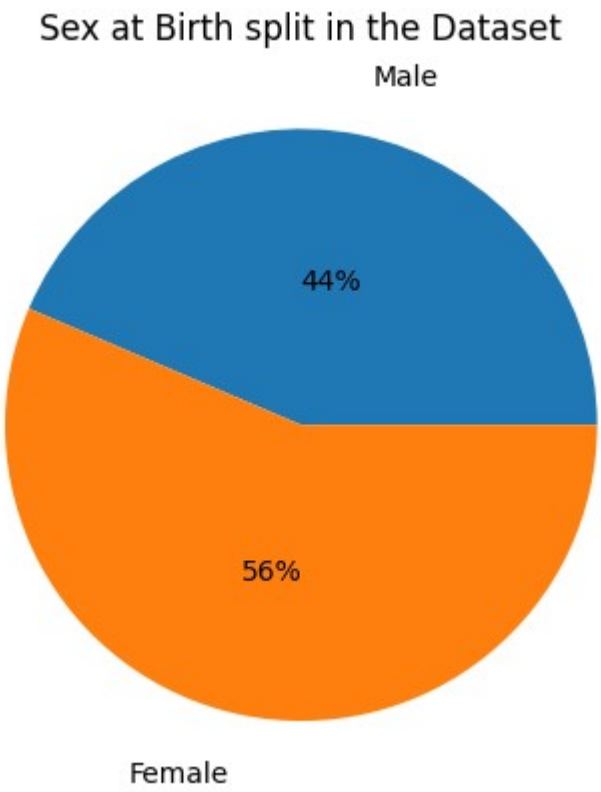

```
work_cols = ['emp_Employed', 'emp_Not_in_labour_force', 'emp_Unemployed']
work_columns = ['Employed','Not In Labour Force','Unemployed']

count_by_work = list()
for col in work_cols:
  count_by_work.append(result[col].value_counts()[1])
plt.pie(count_by_work, labels=work_columns, autopct='%1.0f%%', pctdistance=0.5, labeldis
plt.title("Employment Split in the Dataset")
```

```
Text(0.5, 1.0, 'Employment Split in the Dataset')
```




```
sex_cols = ['sex_Male', 'sex_Female']
sex_columns = ['Male','Female']

count_by_sex = list()
for col in sex_cols:
  count_by_sex.append(result[col].value_counts()[1])

plt.pie(count_by_sex, labels=sex_columns, autopct='%1.0f%%', pctdistance=0.5, labeldistar
plt.title("Sex at Birth split in the Dataset")
```

```
Text(0.5, 1.0, 'Sex at Birth split in the Dataset')
```



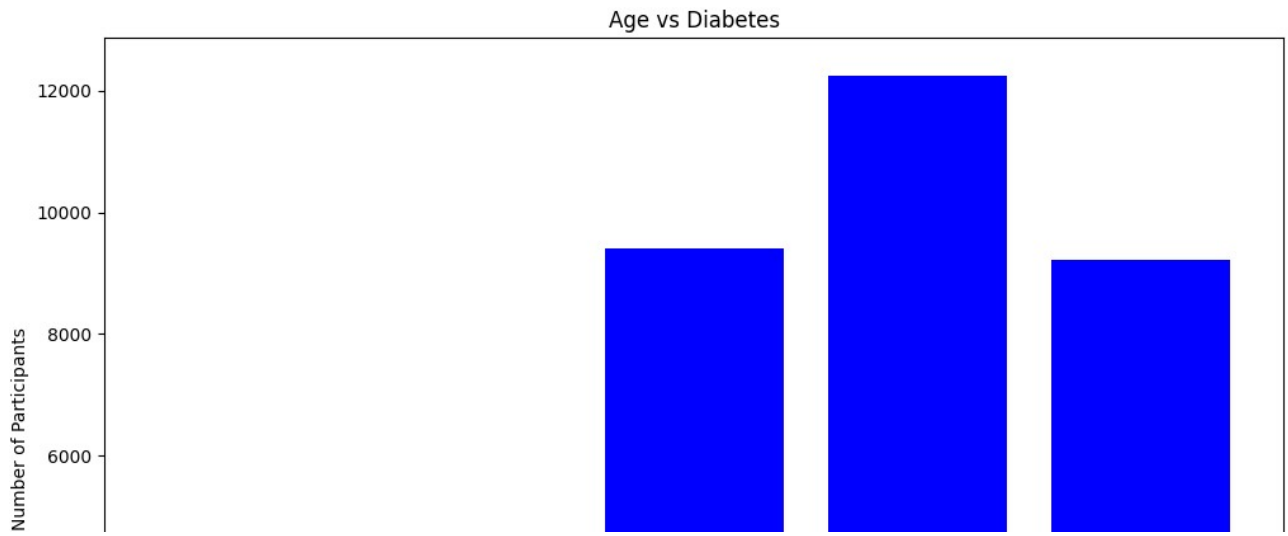
Exploratory Data Analysis

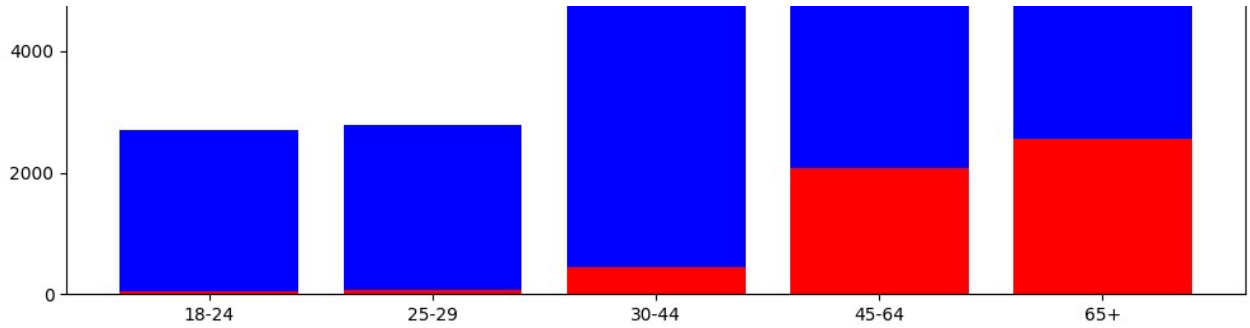
Exploratory Data Analyses

This section will try and explore how various attributes that have been selected after the data preparation stage relate to diabetes.

Demographic Impact

```
have_diabetes = result.loc[result['diabetes'] == 1]
dont_have = result.loc[result['diabetes'] == 2]
have_diabetes = have_diabetes.dropna()
dont_have = dont_have.dropna()

import matplotlib.pyplot as plt
age_split = dict()
age_cols = ['agegroup_18_24', 'agegroup_25_29', 'agegroup_30_44', 'agegroup_45_64', 'age
age_columns = ['18-24','25-29','30-44','45-64','65+']
pos = list()
neg = list()

for col in age_cols:
  age_split[col] = [result.groupby([col,'diabetes'])['pcp'].count()[1][1], result.groupby
  pos.append(result.groupby([col,'diabetes'])['pcp'].count()[1][1])
  neg.append(result.groupby([col,'diabetes'])['pcp'].count()[1][2])

from matplotlib.pyplot import figure

figure(figsize=(12,8))
plt.bar(age_columns, pos, color='r')
plt.bar(age_columns, neg, bottom=pos, color='b')
plt.ylabel("Number of Participants")
plt.title("Age vs Diabetes")
plt.show()
```

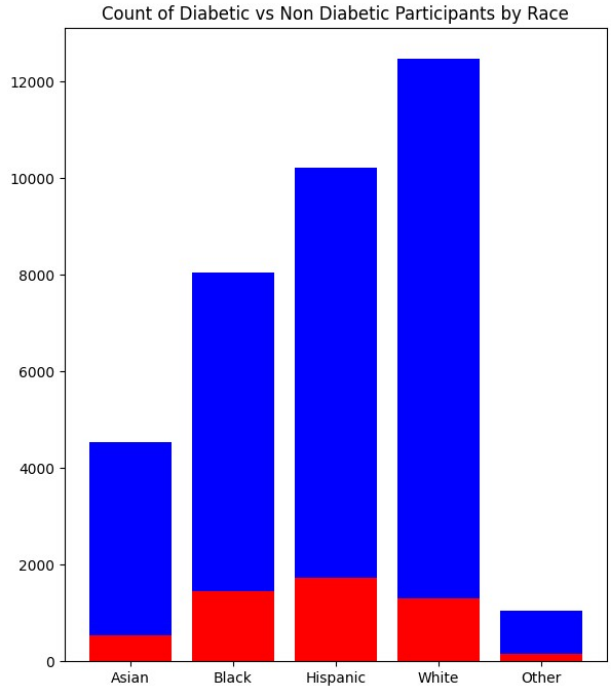


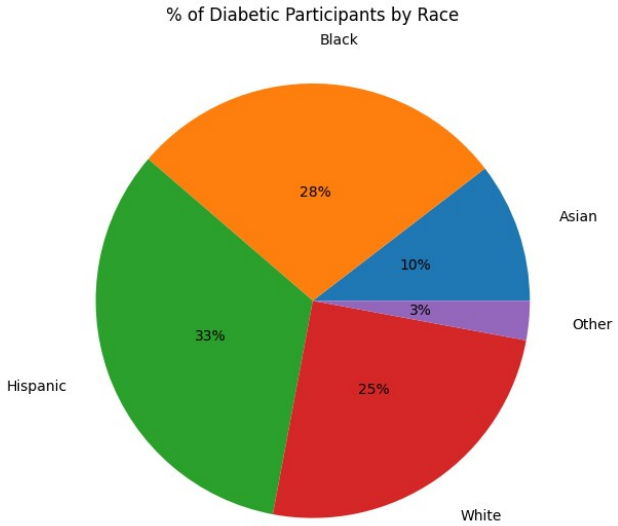


We see that a larger population of those over 45 are diabetic. This is inline with the CDC who mention that people over 45 are more susceptible to Diabetes Type 2. https://www.cdc.gov/diabetes/basics/type2.html

```
import matplotlib.pyplot as plt
race_split = dict()
race_cols = ['newrace_Asian_Non_hispanic', 'newrace_Black_non_hispanic', 'newrace_Hispan
race_columns = ['Asian','Black','Hispanic','White','Other']
pos = list()
neg = list()

for col in race_cols:
  age_split[col] = [result.groupby([col,'diabetes'])['pcp'].count()[1][1], result.groupb
  pos.append(result.groupby([col,'diabetes'])['pcp'].count()[1][1])
  neg.append(result.groupby([col,'diabetes'])['pcp'].count()[1][2])

pp = list()
for val in pos:
  pp.append(val*100/sum(pos))

from matplotlib.pyplot import figure

fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(15, 8)
```

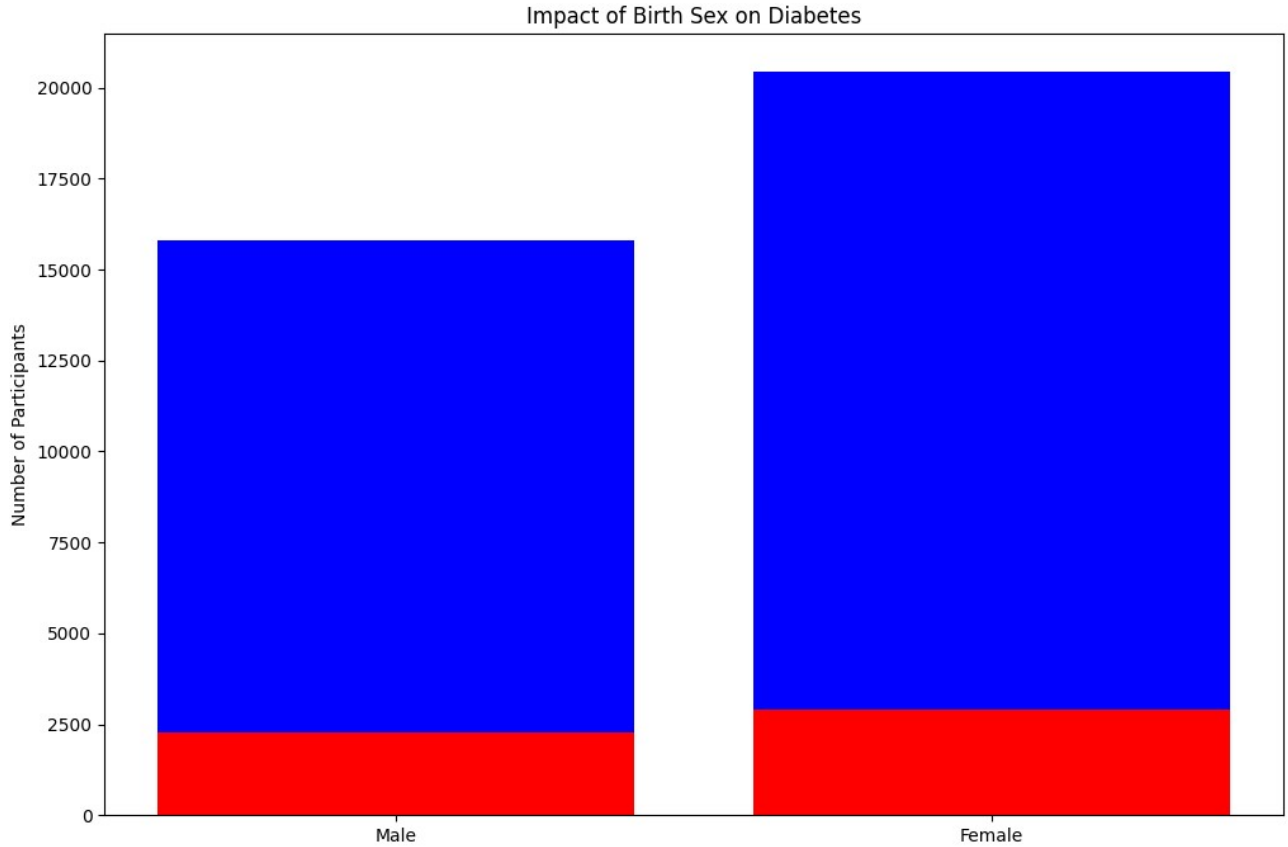
```
ax1.bar(race_columns, pos, color='r')
ax1.bar(race_columns, neg, bottom=pos, color='b')
ax1.set_title("Count of Diabetic vs Non Diabetic Participants by Race")
ax2.pie(pp, labels=race_columns, autopct='%1.0f%%', pctdistance=0.5, labeldistance=1.2)
ax2.set_title("% of Diabetic Participants by Race")
plt.show()
```





Again, here we see that certian racial groups - ie the Black and Hispanic communities are more likely to be diagnosed with Diabetes, as compared to white and Asian communities, where this a lower chance of risk. Related Studies: https://minorityhealth.hhs.gov/diabetes-and-african-americans

```
import matplotlib.pyplot as plt
sex_cols = ['sex_Male', 'sex_Female']
sex_columns = ['Male','Female']
pos = list()
neg = list()

for col in sex_cols:

  pos.append(result.groupby([col,'diabetes'])['pcp'].count()[1][1])
  neg.append(result.groupby([col,'diabetes'])['pcp'].count()[1][2])

from matplotlib.pyplot import figure

figure(figsize=(12,8))
plt.bar(sex_columns, pos, color='r')
plt.bar(sex_columns, neg, bottom =pos , color='b')
plt.title("Impact of Birth Sex on Diabetes")
plt.ylabel("Number of Participants")
plt.show()
```



We see here that there are more female participants and the percentage of diabetic population across the sex at birth does not seem to have much of an impact.

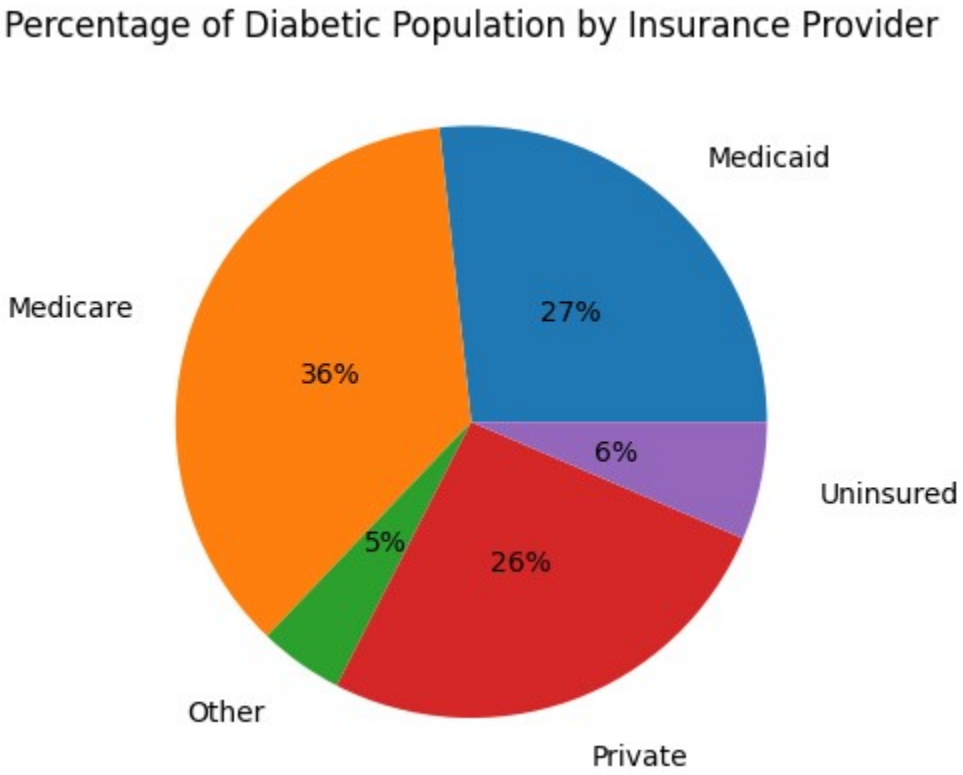
Health and Insurance

```
import matplotlib.pyplot as plt
ins_split = dict()
ins_cols = ['insure_Medicaid', 'insure_Medicare', 'insure_Other', 'insure_Private', 'insu
ins_columns = ['Medicaid','Medicare','Other','Private','Uninsured']
pos = list()
neg = list()

for col in ins_cols:
  ins_split[col] = [result.groupby([col,'diabetes'])['pcp'].count()[1][1], result.groupb
  pos.append(result.groupby([col,'diabetes'])['pcp'].count()[1][1])
  neg.append(result.groupby([col,'diabetes'])['pcp'].count()[1][2])

pp = list()
for val in pos:
  pp.append(val*100/sum(pos))
```

```
fig, ax = plt.subplots()
ax.set_title("Percentage of Diabetic Population by Insurance Provider")
ax.pie(pp, labels=ins_columns, autopct='%1.0f%%', pctdistance=0.5, labeldistance=1.2)
```

```
([<matplotlib.patches.Wedge at 0x7b9c0cb8c2e0>,
  <matplotlib.patches.Wedge at 0x7b9c0cb8c520>,
  <matplotlib.patches.Wedge at 0x7b9c0cba72e0>,
  <matplotlib.patches.Wedge at 0x7b9c0ca5fd60>,
  <matplotlib.patches.Wedge at 0x7b9c0ca5c280>],
 [Text(0.8014659546317726, 0.8931138357265446, 'Medicaid'),
  Text(-1.1364116912595874, 0.38544580419110047, 'Medicare'),
  Text(-0.6946575933755011, -0.9784941634806805, 'Other'),
  Text(0.4077933367026898, -1.1285852181119893, 'Private'),
  Text(1.1755187448994497, -0.24115488879560892, 'Uninsured')],
 [Text(0.3339441477632386, 0.37213076488606023, '27%'),
```

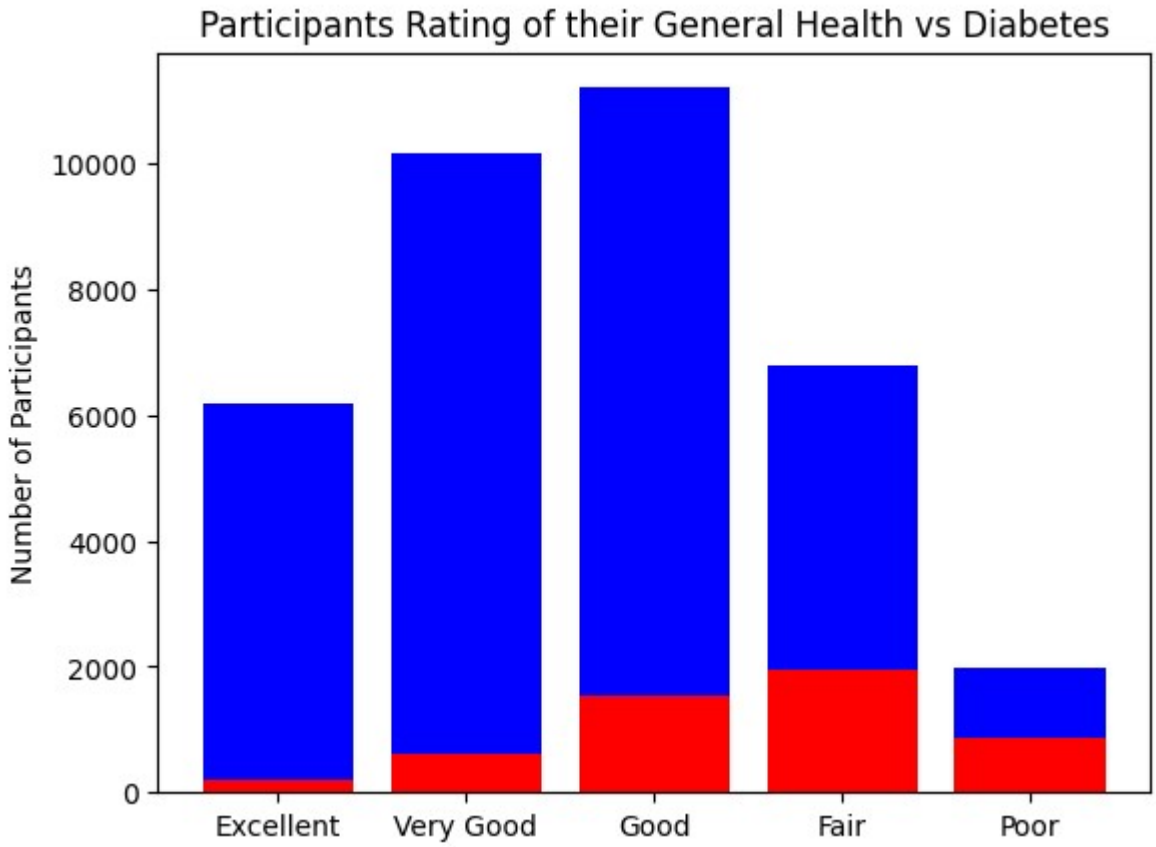
```
[Text(0.3339441477032586, 0.37213670488066023, '27%'),
 Text(-0.47350487135816144, 0.16060241841295853, '36%'),
 Text(-0.2894406390645884, -0.40770590145028357, '5%'),
 Text(0.16991389029278742, -0.4702438408799956, '26%'),
 Text(0.4897994770414374, -0.10048120366483705, '6%')])
```



From the above graph, we can see that over 60% of diabetics are on medicare or medicaid. This strongly correlates to the fact that Medicaid is available to only senior citizens who are already at a higher risk of diabetes and Medicare is available to financially impoverished people, who again seem to be at the risk of diabetes.

```
import matplotlib.pyplot as plt
rating_cols = ['Excellent', 'Very Good', 'Good', 'Fair', 'Poor']
pos = list()
neg = list()


for val in range(1,6):

  pos.append(result.groupby(['generalhealth','diabetes'])['pcp'].count()[val][1])
  neg.append(result.groupby(['generalhealth','diabetes'])['pcp'].count()[val][2])


plt.bar(rating_cols, pos, color='r', label='Diabetic')
plt.bar(rating_cols, neg, bottom =pos , color='b', label='Non Diabetic')

plt.title("Participants Rating of their General Health vs Diabetes")
```

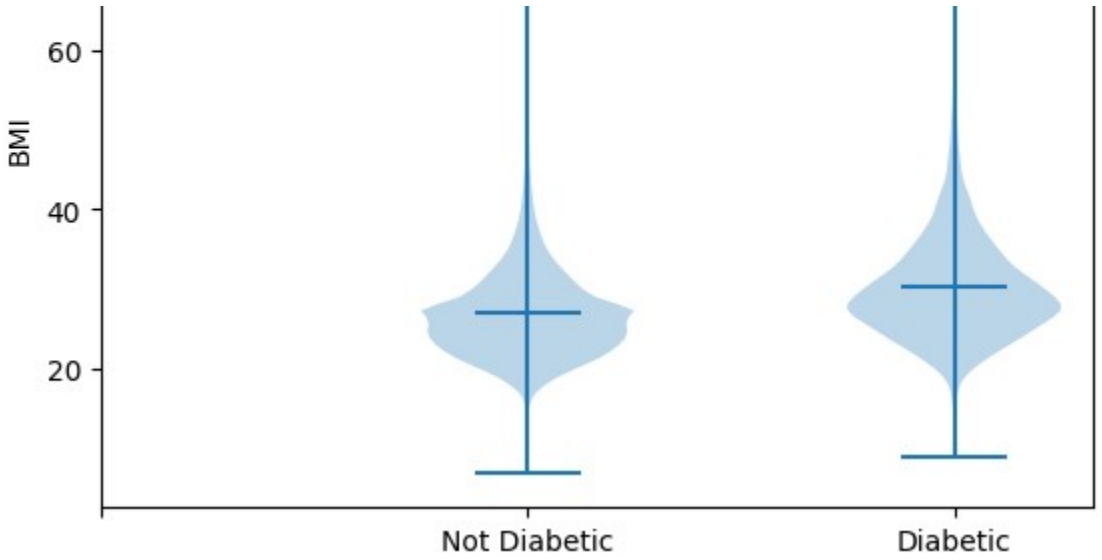
```
plt.ylabel("Number of Participants")
plt.show()
```



The above question asked participants to rate their own health in one of the above 5 categories. We can see that there are more diabetic people in the categories as the rating goes down.

```
import numpy as np
Diabetic = ["", "Not Diabetic", "Diabetic"]
y1 = dont_have['bmi']
y2 = have_diabetes['bmi']
y = [y1,y2]
plt.violinplot(y, showmeans = True)
X_axis = np.arange(len(Diabetic))
plt.xticks(X_axis, Diabetic)
plt.title("BMI Variation in Diabetics vs Non Diabetic")
plt.ylabel("BMI")
```

```
Text(0, 0.5, 'BMI')
```

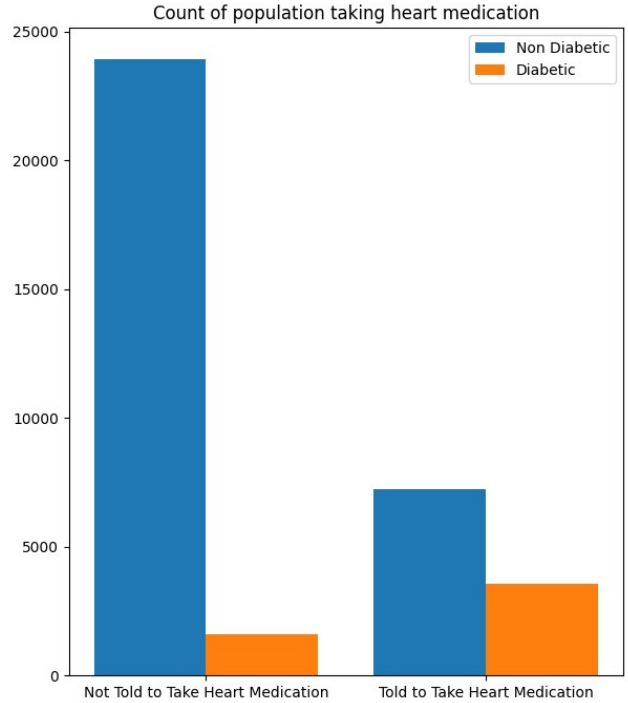


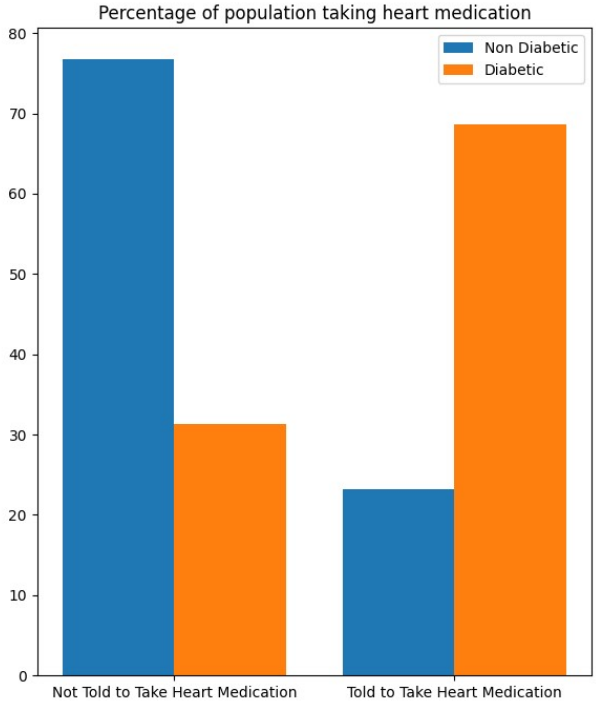
In accordance with studies, we can see that Diabetic folk tend to have a higher average BMI (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4457375/) and this is inline with many studies drawing relations between Diabetes and BMI

```
Attributes = ['Not Told to Take Heart Medication', 'Told to Take Heart Medication']
vals = list()
vals.append(dont_have['toldprescription'].value_counts()[2])
vals.append(dont_have['toldprescription'].value_counts()[1])

vals2 = list()
vals2.append(have_diabetes['toldprescription'].value_counts()[2])
vals2.append(have_diabetes['toldprescription'].value_counts()[1])

pp = list()
for val in vals:
  pp.append(val*100/sum(vals))

pp2 = list()
for val in vals2:
  pp2.append(val*100/sum(vals2))

fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(15, 8)

X_axis = np.arange(len(Attributes))
ax1.bar(X_axis - 0.2, vals, 0.4, label="Non Diabetic")
ax1.bar(X_axis + 0.2, vals2, 0.4 , label="Diabetic")
ax1.set(xticks=X_axis, xticklabels=Attributes)
ax1.set_title("Count of population taking heart medication")
ax1.legend()
```

```
X_axis = np.arange(len(Attributes))
ax2.bar(X_axis - 0.2, pp, 0.4, label = "Non Diabetic")
ax2.bar(X_axis + 0.2, pp2, 0.4,  label = "Diabetic")
ax2.set(xticks=X_axis, xticklabels=Attributes)
ax2.set_title("Percentage of population taking heart medication")
ax2.legend()
```

```
<matplotlib.legend.Legend at 0x7b9c0ca75390>
```

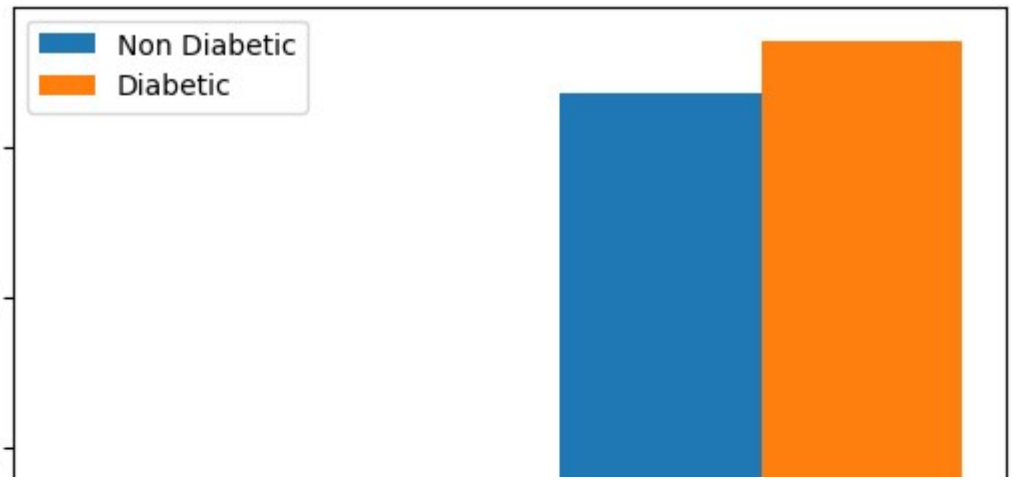


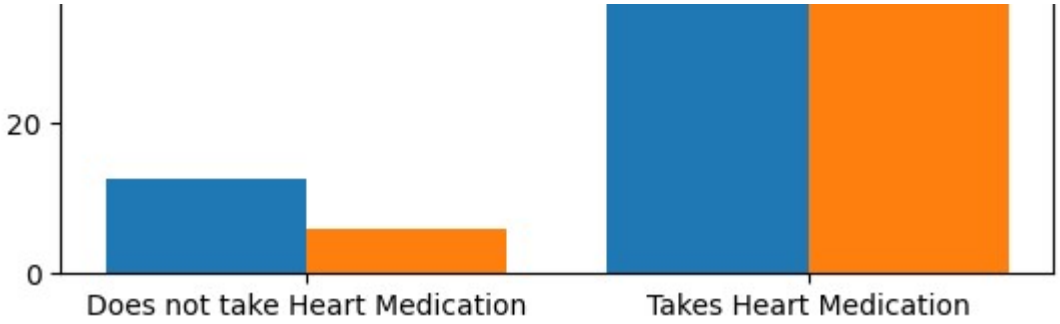


Here we see a larger percentage of diabetics being asked to also take Heart Medication. And we see a high correlation being diabetics and cardiovascular issues. Diabetic Patients are at a risk of being impacted by various Cardiovascular issues.

```
told_to_take_meds_nd = vals
told_to_take_meds_d = vals2

import numpy as np



ActualTakesMed = ['Does not take Heart Medication', 'Takes Heart Medication']
vals = list()
vals.append(dont_have['takingmeds'].value_counts()[2])
vals.append(dont_have['takingmeds'].value_counts()[1])

vals2 = list()
vals2.append(have_diabetes['takingmeds'].value_counts()[2])
vals2.append(have_diabetes['takingmeds'].value_counts()[1])

takes_meds_nd = [ told_to_take_meds_nd[1] - vals[1], vals[1]]
takes_meds_d = [told_to_take_meds_d[1] - vals2[1], vals2[1]]


pp = list()
for val in takes_meds_nd:
  pp.append(val*100/sum(takes_meds_nd))

pp2 = list()
for val in takes_meds_d:
  pp2.append(val*100/sum(takes_meds_d))



X_axis = np.arange(len(ActualTakesMed))
plt.bar(X_axis - 0.2, pp, 0.4, label = "Non Diabetic")
plt.bar(X_axis + 0.2, pp2, 0.4, label = "Diabetic")
plt.xticks(X_axis, ActualTakesMed)
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7b9c0c98bd30>
```



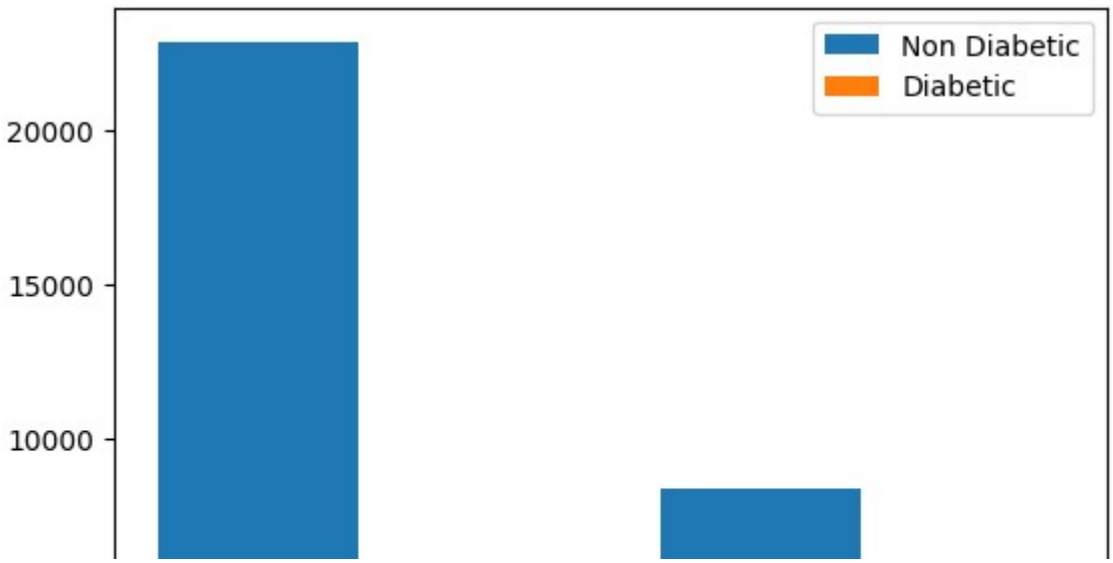


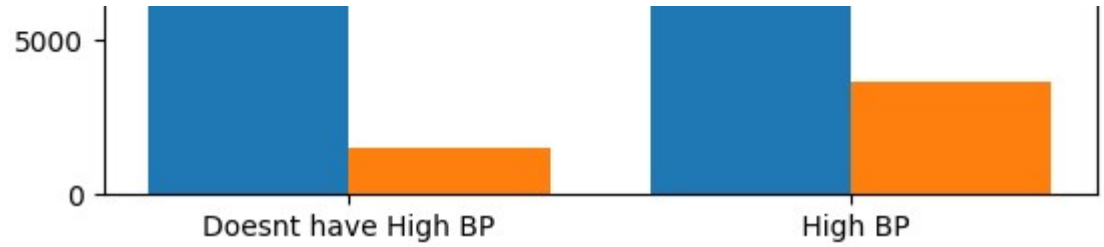
This graph shows us the percentage of Heart Patients who actually take their medication and this might give us some insight as Non diabetics seem to be more lax taking their heart medication compared to diabetics.

```
import numpy as np
```

```
ActualTakesMed = ['Doesnt have High BP', ' High BP']
vals = list()
vals.append(dont_have['toldhighbp'].value_counts()[2])
vals.append(dont_have['toldhighbp'].value_counts()[1])
X_axis = np.arange(len(ActualTakesMed))
plt.bar(X_axis - 0.2, vals, 0.4, label = "Non Diabetic")

vals2 = list()
vals2.append(have_diabetes['toldhighbp'].value_counts()[2])
vals2.append(have_diabetes['toldhighbp'].value_counts()[1])
plt.bar(X_axis + 0.2, vals2, 0.4, label = "Diabetic")
plt.xticks(X_axis, ActualTakesMed)
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7b9c0c7d4160>
```

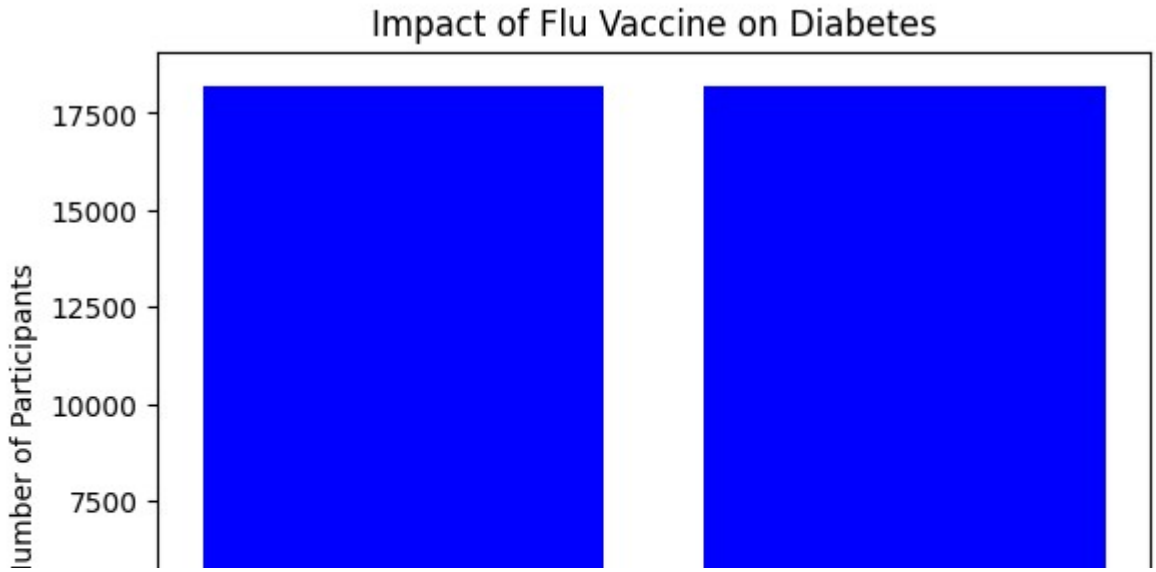


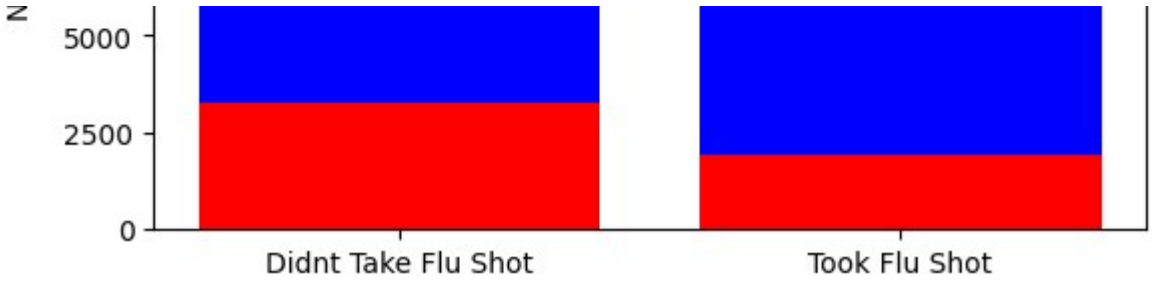


We again see a very high correlation between cardio vascular issues and diabetes, with people with High Blood Presure being more likely to have diabetes as well

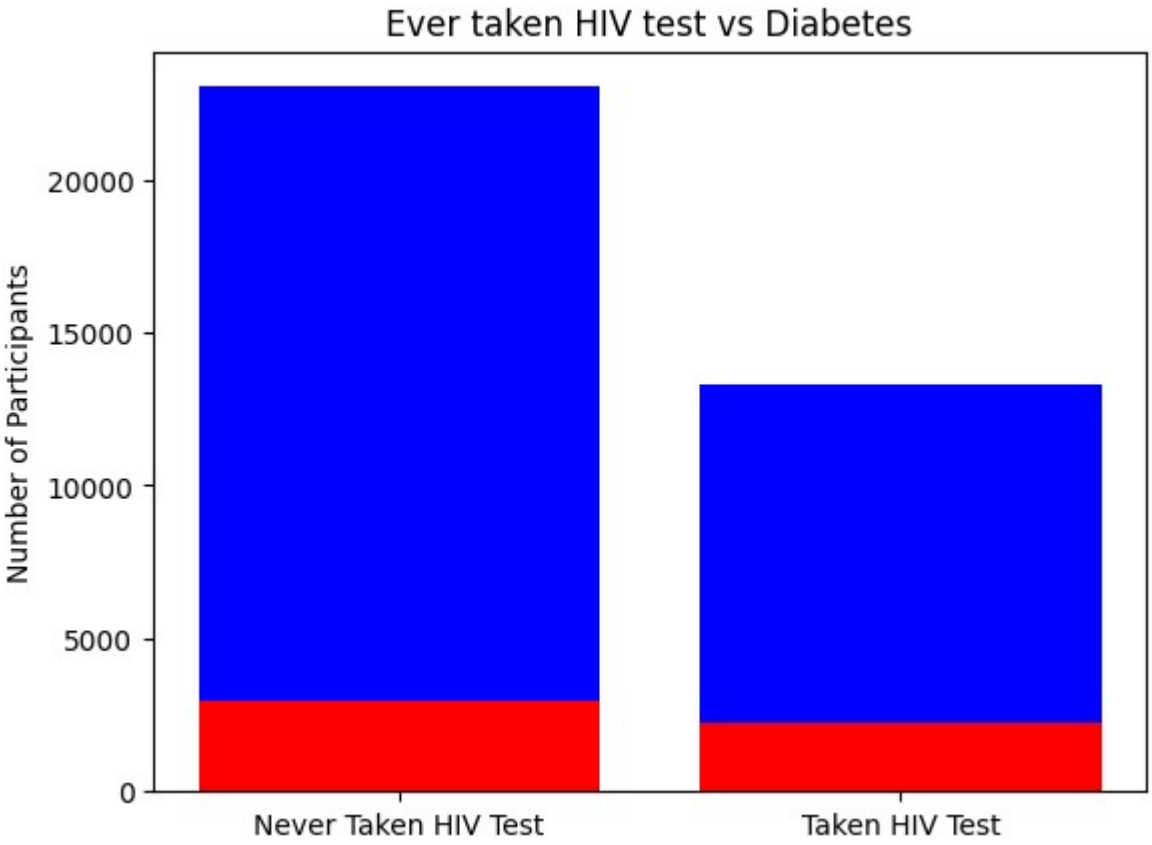
The two graphs related to the relation between HIV tests and Flu Vaccine shots vs Diabetes can give us an idea that people more likely to take precautions to their health in other categories also take care of their health when it comes to diabetes.

```
import matplotlib.pyplot as plt
flu_cols = ['Didnt Take Flu Shot','Took Flu Shot']
pos = list()
neg = list()

for val in range(1,3):

  pos.append(result.groupby(['fluvaccineshot','diabetes'])['pcp'].count()[val][1])
  neg.append(result.groupby(['fluvaccineshot','diabetes'])['pcp'].count()[val][2])


plt.bar(flu_cols, pos, color='r', label='Diabetic')
plt.bar(flu_cols, neg, bottom =pos , color='b', label='Non Diabetic')
plt.title("Impact of Flu Vaccine on Diabetes")
plt.ylabel("Number of Participants")
plt.show()
```





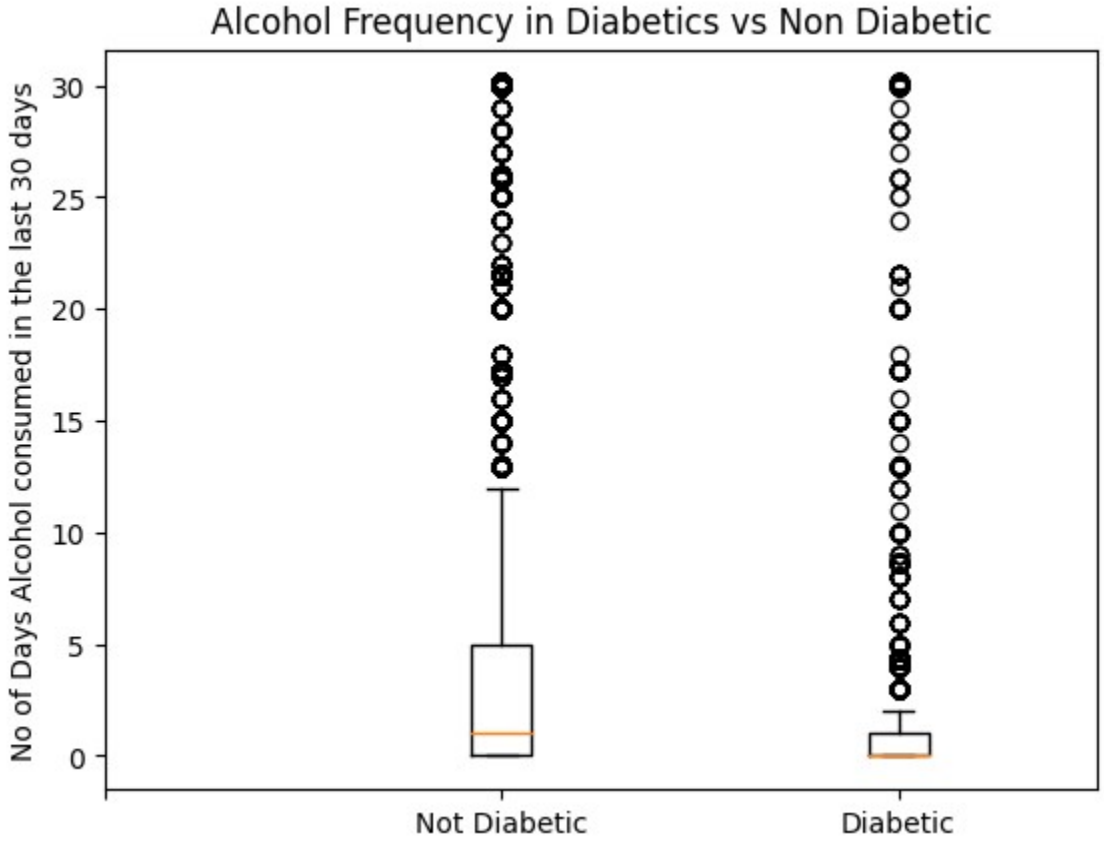

```
import matplotlib.pyplot as plt
flu_cols = ['Never Taken HIV Test','Taken HIV Test']
pos = list()
neg = list()

for val in range(1,3):

  pos.append(result.groupby(['everhivtest','diabetes'])['pcp'].count()[val][1])
  neg.append(result.groupby(['everhivtest','diabetes'])['pcp'].count()[val][2])


plt.bar(flu_cols, pos, color='r', label='Diabetic')
plt.bar(flu_cols, neg, bottom =pos , color='b', label='Non Diabetic')
plt.title("Ever taken HIV test vs Diabetes")
plt.ylabel("Number of Participants")
plt.show()
```



Lifestyle

```
y1 = dont_have['daysalc30']
y2 = have_diabetes['daysalc30']
y = [y1,y2]
plt.boxplot(y )
X_axis = np.arange(len(Diabetic))
plt.xticks(X_axis, Diabetic)
plt.title("Alcohol Frequency in Diabetics vs Non Diabetic")
plt.ylabel("No of Days Alcohol consumed in the last 30 days")
```

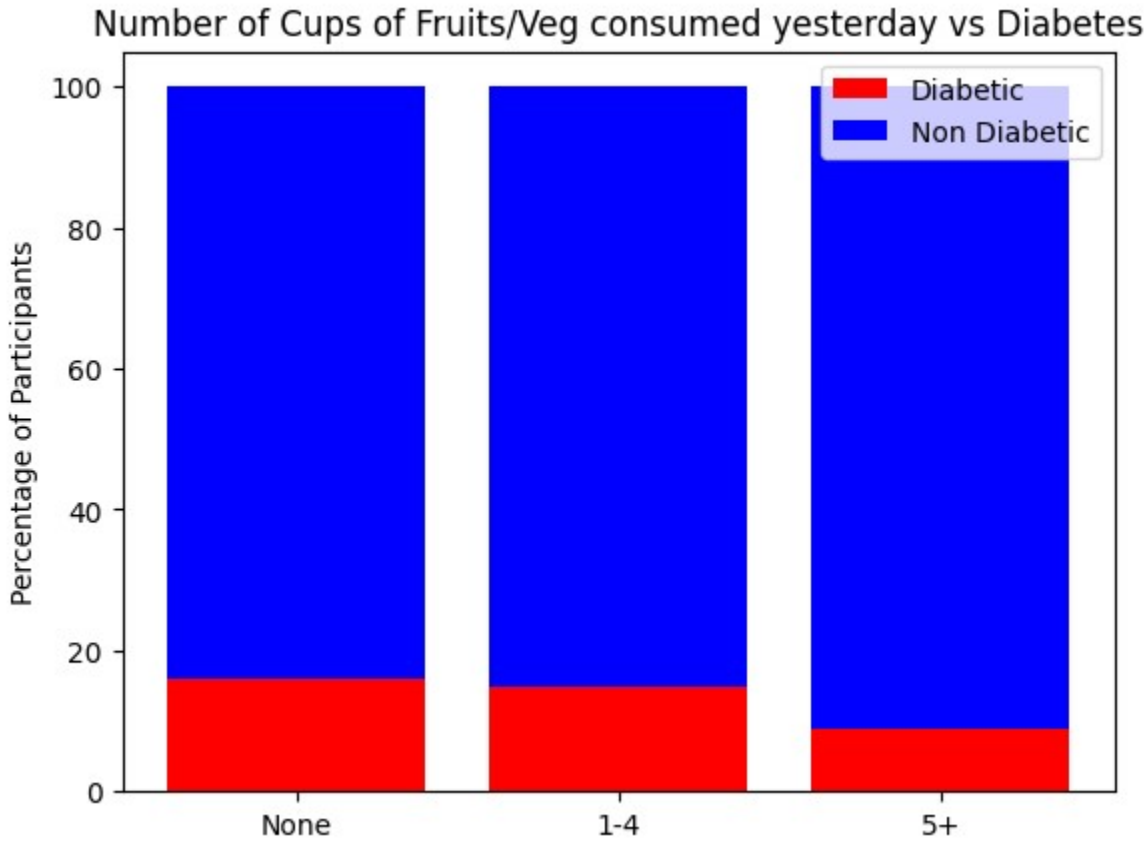
```
Text(0, 0.5, 'No of Days Alcohol consumed in the last 30 days')
```



From the above graph we can see that Non Diabetic people tend to be more relaxed with alcohol consumption with there being a higher mean and mean + 1 standard deviation for non diabetics as compared to Diabetics

```
import matplotlib.pyplot as plt
rating_cols = ['None', '1-4', '5+']
pos = list()
neg = list()


for val in range(1,4):
```

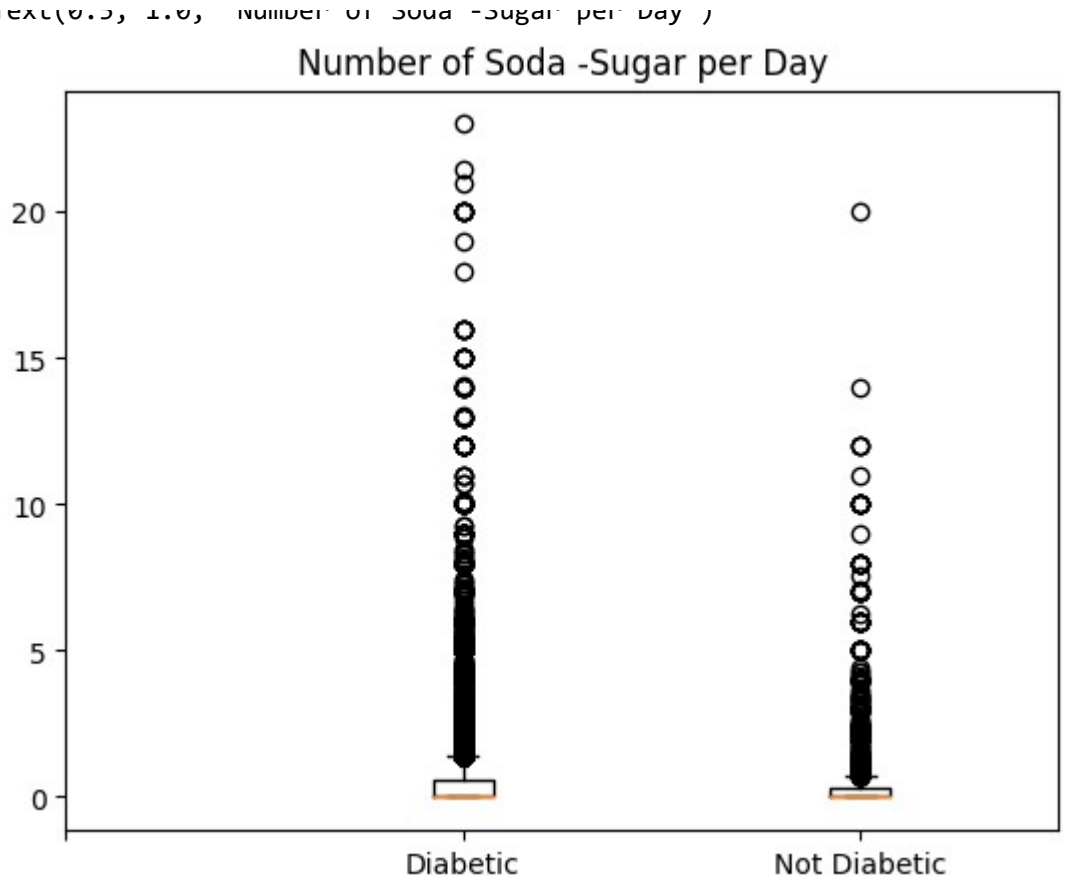
```
for val in range(1,4):

  pos_percent = result.groupby(['fruitveg','diabetes'])['pcp'].count()[val][1]/(result.gr
  pos.append(pos_percent*100)
  neg.append((1-pos_percent)*100)


plt.bar(rating_cols, pos, color='r', label='Diabetic')
plt.bar(rating_cols, neg, bottom =pos , color='b', label='Non Diabetic')
plt.title("Number of Cups of Fruits/Veg consumed yesterday vs Diabetes")
plt.ylabel("Percentage of Participants")
plt.legend()
plt.show()
```



We can see that people who have a healthier more fibrous diet also tend to be less diabetic as compared to those who do not consume any fruits or vegetables in their diet.

```
Diabetic = ['', 'Diabetic', 'Not Diabetic']
y1 = dont_have['nsodasugarperday']
y2 = have_diabetes['nsodasugarperday']
y = [y1,y2]
plt.boxplot(y)
X_axis = np.arange(len(Diabetic))
plt.xticks(X_axis, Diabetic)
plt.title("Number of Soda -Sugar per Day")
```

```
Text(0.5, 1.0, 'Number of Soda -Sugar per Day')
```



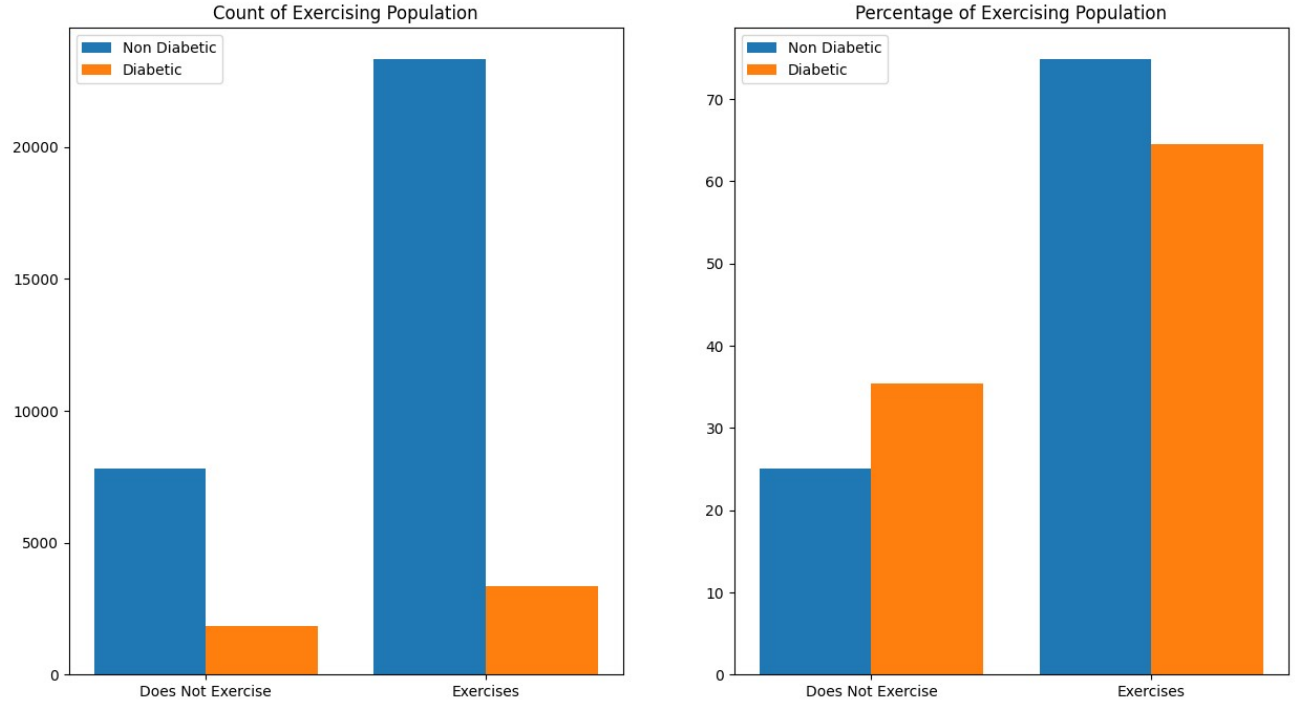
From the above grpah we can see that, while mean and mean + 1 st.deviation for diabetics and non-diabetics seems to be close to each other, there are more diabetics who tend to consume more sugary drinks than Non Diabteic folks.

```
import numpy as np

Exercise = ['Does Not Exercise', 'Exercises']
vals = list()
vals.append(dont_have['exercise'].value_counts()[2])
vals.append(dont_have['exercise'].value_counts()[1])

vals2 = list()
vals2.append(have_diabetes['exercise'].value_counts()[2])
vals2.append(have_diabetes['exercise'].value_counts()[1])

pp = list()
for val in vals:
  pp.append(val*100/sum(vals))

pp2 = list()
for val in vals2:
  pp2.append(val*100/sum(vals2))
```

```
fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(15, 8)
Exercise = ['Does Not Exercise', 'Exercises']

X_axis = np.arange(len(Exercise))
ax1.bar(X_axis - 0.2, vals, 0.4, label="Non Diabetic")
ax1.bar(X_axis + 0.2, vals2, 0.4 , label="Diabetic")
ax1.set(xticks=X_axis, xticklabels=Exercise)
ax1.set_title("Count of Exercising Population")
ax1.legend()
```

```
X_axis = np.arange(len(Exercise))
ax2.bar(X_axis - 0.2, pp, 0.4, label = "Non Diabetic")
ax2.bar(X_axis + 0.2, pp2, 0.4,  label = "Diabetic")
ax2.set(xticks=X_axis, xticklabels=Exercise)
ax2.set_title("Percentage of Exercising Population")
ax2.legend()
```

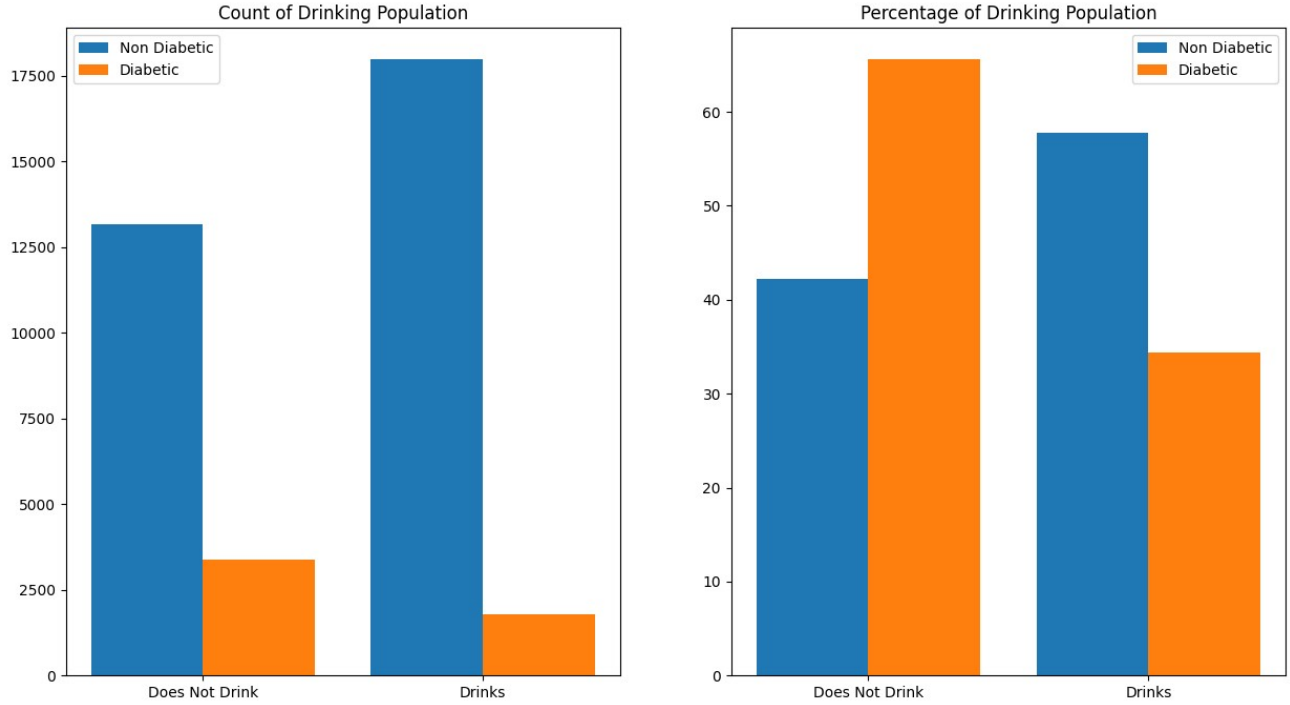
```
<matplotlib.legend.Legend at 0x7bc57c69ee30>
```



The question posed by the survey about exercise asked if people exercised in the last 30 days. But, we still see a large diabetic population not exercising compared to the non diabetic population

```
Attributes = ['Does Not Drink', 'Drinks']
vals = list()
vals.append(dont_have['drinker'].value_counts()[2])
vals.append(dont_have['drinker'].value_counts()[1])

vals2 = list()
vals2.append(have_diabetes['drinker'].value_counts()[2])
vals2.append(have_diabetes['drinker'].value_counts()[1])

pp = list()
for val in vals:
  pp.append(val*100/sum(vals))

pp2 = list()
for val in vals2:
  pp2.append(val*100/sum(vals2))

fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(15, 8)

X_axis = np.arange(len(Exercise))
ax1.bar(X_axis - 0.2, vals, 0.4, label="Non Diabetic")
ax1.bar(X_axis + 0.2, vals2, 0.4 , label="Diabetic")
ax1.set(xticks=X_axis, xticklabels=Attributes)
ax1.set_title("Count of Drinking Population")
ax1.legend()



X_axis = np.arange(len(Exercise))
ax2.bar(X_axis - 0.2, pp, 0.4, label = "Non Diabetic")
ax2.bar(X_axis + 0.2, pp2, 0.4,  label = "Diabetic")
ax2.set(xticks=X_axis, xticklabels=Attributes)
ax2.set_title("Percentage of Drinking Population")
ax2.legend()
```

```
<matplotlib.legend.Legend at 0x7bc577ee19f0>
```

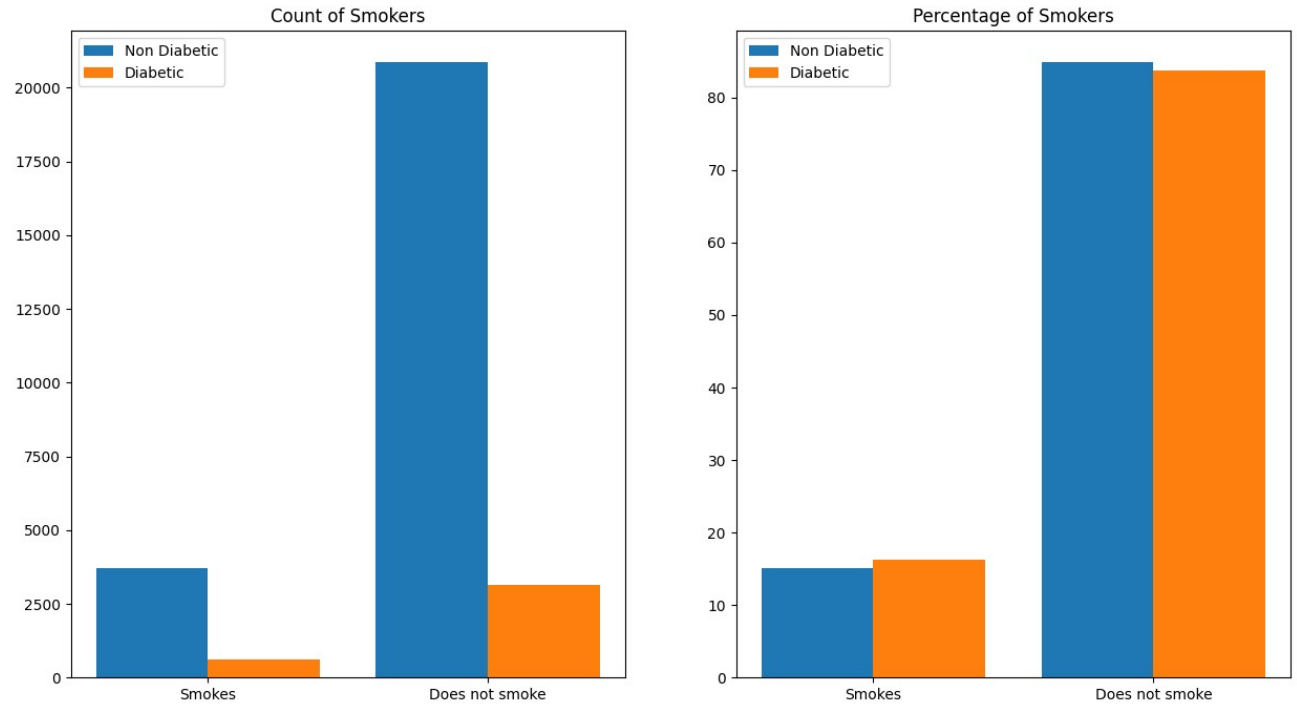


As observed previously we see that Non Diabetic participants are more likely to drink compared to Diabetic participants

```
Attributes = ['Smokes', 'Does not smoke']
vals = list()
vals.append(dont_have['smoker'].value_counts()[2])
vals.append(dont_have['smoker'].value_counts()[1])

vals2 = list()
vals2.append(have_diabetes['smoker'].value_counts()[2])
```

```
vals2.append(have_diabetes['smoker'].value_counts()[1])

pp = list()
for val in vals:
  pp.append(val*100/sum(vals))

pp2 = list()
for val in vals2:
  pp2.append(val*100/sum(vals2))

fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(15, 8)

X_axis = np.arange(len(Exercise))
ax1.bar(X_axis - 0.2, vals, 0.4, label="Non Diabetic")
ax1.bar(X_axis + 0.2, vals2, 0.4 , label="Diabetic")
ax1.set(xticks=X_axis, xticklabels=Attributes)
ax1.set_title("Count of Smokers")
ax1.legend()



X_axis = np.arange(len(Exercise))
ax2.bar(X_axis - 0.2, pp, 0.4, label = "Non Diabetic")
ax2.bar(X_axis + 0.2, pp2, 0.4,  label = "Diabetic")
ax2.set(xticks=X_axis, xticklabels=Attributes)
ax2.set_title("Percentage of Smokers")
ax2.legend()
```

```
<matplotlib.legend.Legend at 0x7bc5711b2440>
```

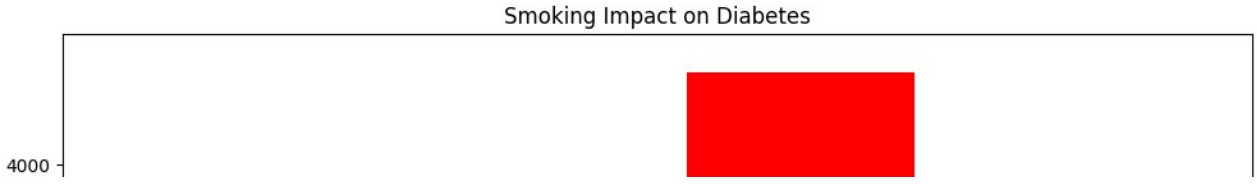


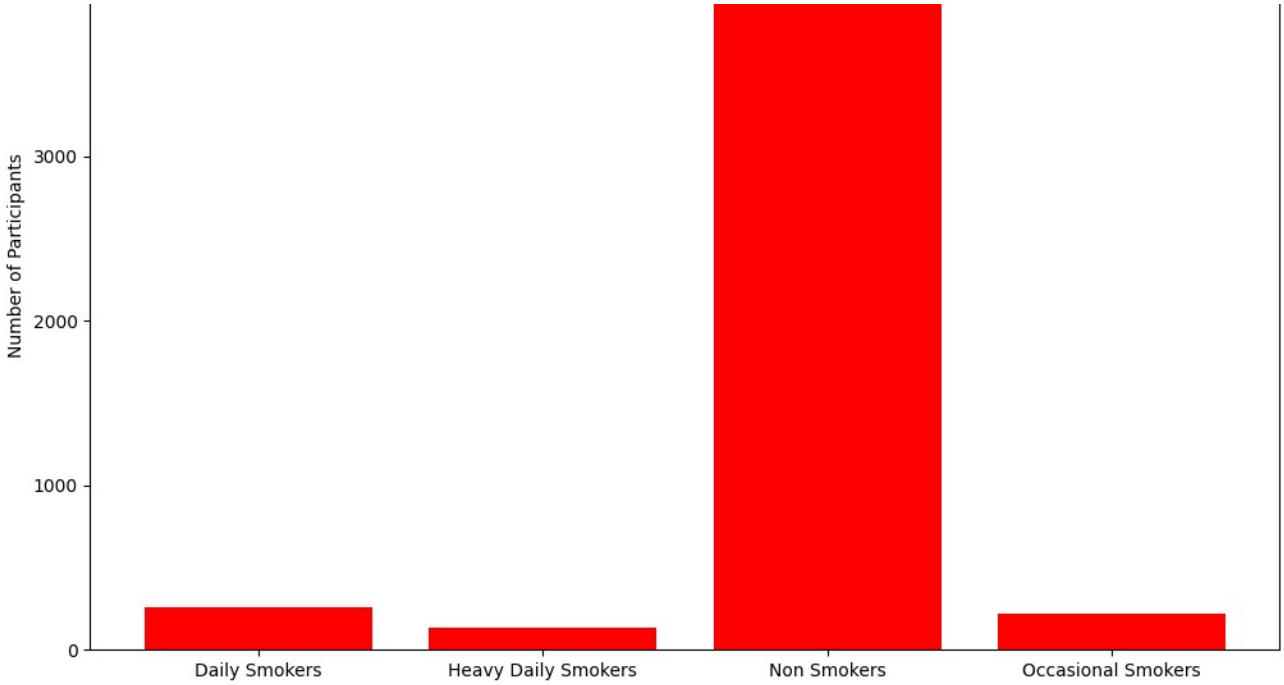
We do not see any direct corerlation here as the percentages are almost the same between smokers and non smokers.

```
import matplotlib.pyplot as plt
smoke_split = dict()
smoke_cols = ['smokecat_Daily_smoker', 'smokecat_Heavy_daily_smoker', 'smokecat_Non-smok
smoke_columns = ['Daily Smokers','Heavy Daily Smokers','Non Smokers','Occasional Smokers
pos = list()
neg = list()

for col in smoke_cols:
  smoke_split[col] = [result.groupby([col,'diabetes17'])['pcp17'].count()[1][1], result.
  pos.append(result.groupby([col,'diabetes17'])['pcp17'].count()[1][1])
  neg.append(result.groupby([col,'diabetes17'])['pcp17'].count()[1][2])
```

```
[261, 1526]
[132, 687]
[4564, 27471]
[217, 1493]
```

```
from matplotlib.pyplot import figure

figure(figsize=(12,8))
plt.bar(smoke_columns, pos, color='r')
plt.title("Smoking Impact on Diabetes")
plt.ylabel("Number of Participants")
plt.show()
```



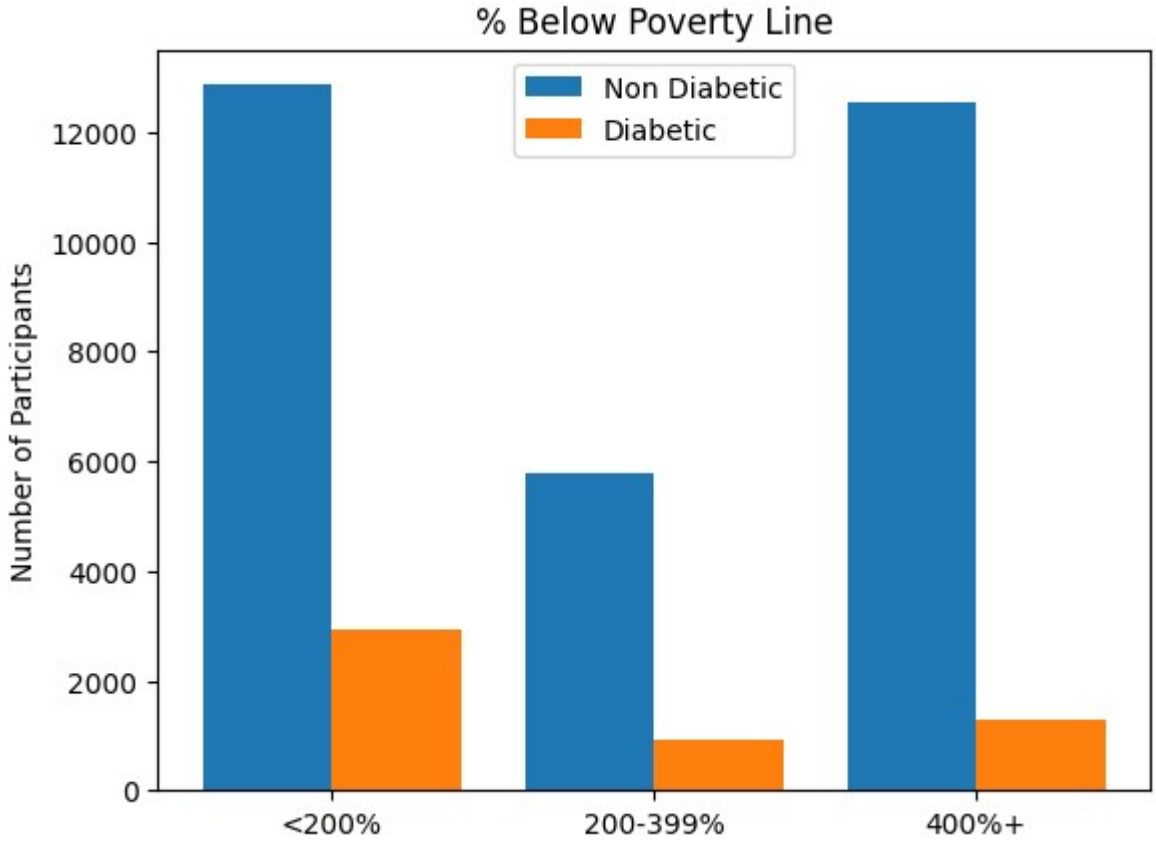


Education, Employment and Income

```
import numpy as np

Poverty = ['<200%', '200-399%', '400%+']
vals = list()
vals.append(dont_have['imputed_povgroup'].value_counts()[1])
vals.append(dont_have['imputed_povgroup'].value_counts()[2])
vals.append(dont_have['imputed_povgroup'].value_counts()[3])
X_axis = np.arange(len(Poverty))
plt.bar(X_axis - 0.2, vals, 0.4, label = "Non Diabetic")

vals2 = list()
vals2.append(have_diabetes['imputed_povgroup'].value_counts()[1])
vals2.append(have_diabetes['imputed_povgroup'].value_counts()[2])
vals2.append(have_diabetes['imputed_povgroup'].value_counts()[3])
```

```
plt.bar(X_axis + 0.2, vals2, 0.4, label = "Diabetic")
plt.xticks(X_axis, Poverty)
plt.legend()
plt.title("% Below Poverty Line")
plt.ylabel("Number of Participants")
```

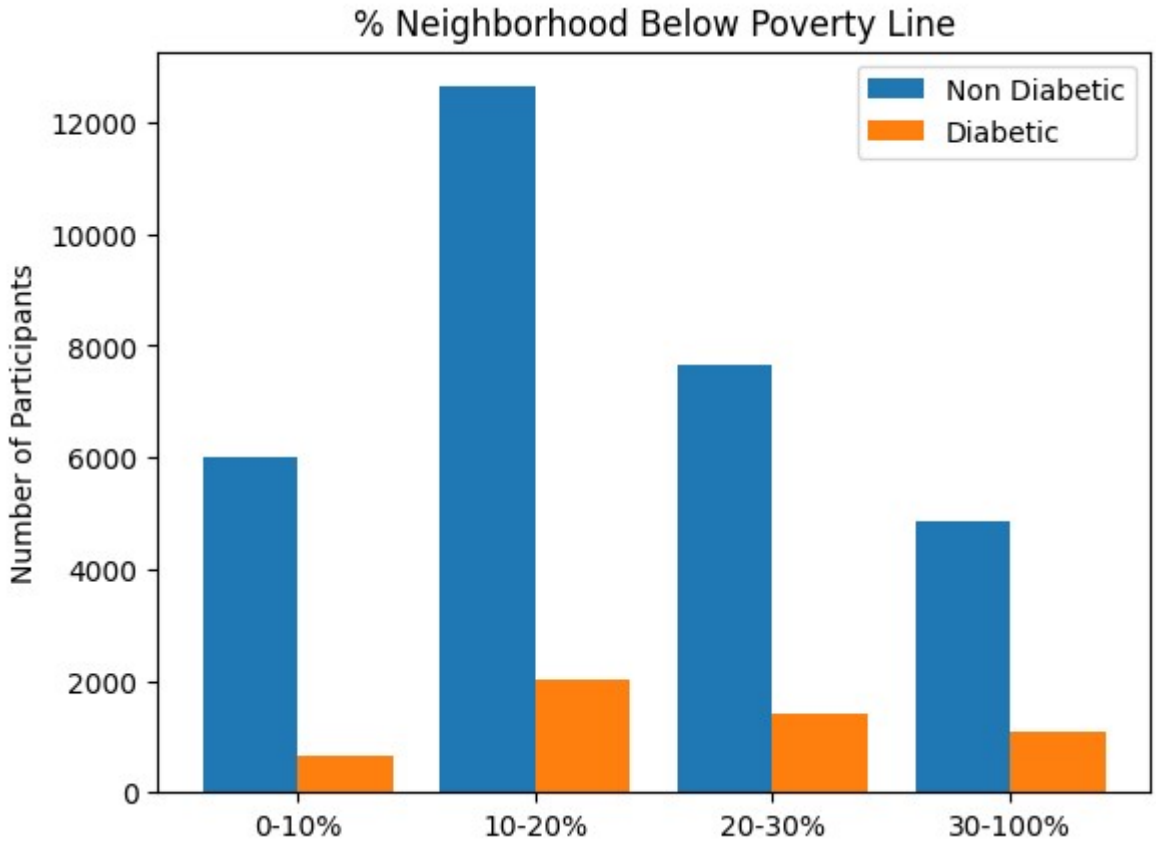
```
Text(0, 0.5, 'Number of Participants')
```




```
import numpy as np

Poverty = ['0-10%', '10-20%', '20-30%', '30-100%']
vals = list()
vals.append(dont_have['imputed_neighpovgroup'].value_counts()[1])
vals.append(dont_have['imputed_neighpovgroup'].value_counts()[2])
vals.append(dont_have['imputed_neighpovgroup'].value_counts()[3])
vals.append(dont_have['imputed_neighpovgroup'].value_counts()[4])
X_axis = np.arange(len(Poverty))
plt.bar(X_axis - 0.2, vals, 0.4, label = "Non Diabetic")

vals2 = list()
vals2.append(have_diabetes['imputed_neighpovgroup'].value_counts()[1])
vals2.append(have_diabetes['imputed_neighpovgroup'].value_counts()[2])
vals2.append(have_diabetes['imputed_neighpovgroup'].value_counts()[3])
vals2.append(have_diabetes['imputed_neighpovgroup'].value_counts()[4])
plt.bar(X_axis + 0.2, vals2, 0.4, label = "Diabetic")
plt.xticks(X_axis, Poverty)
plt.legend()
```

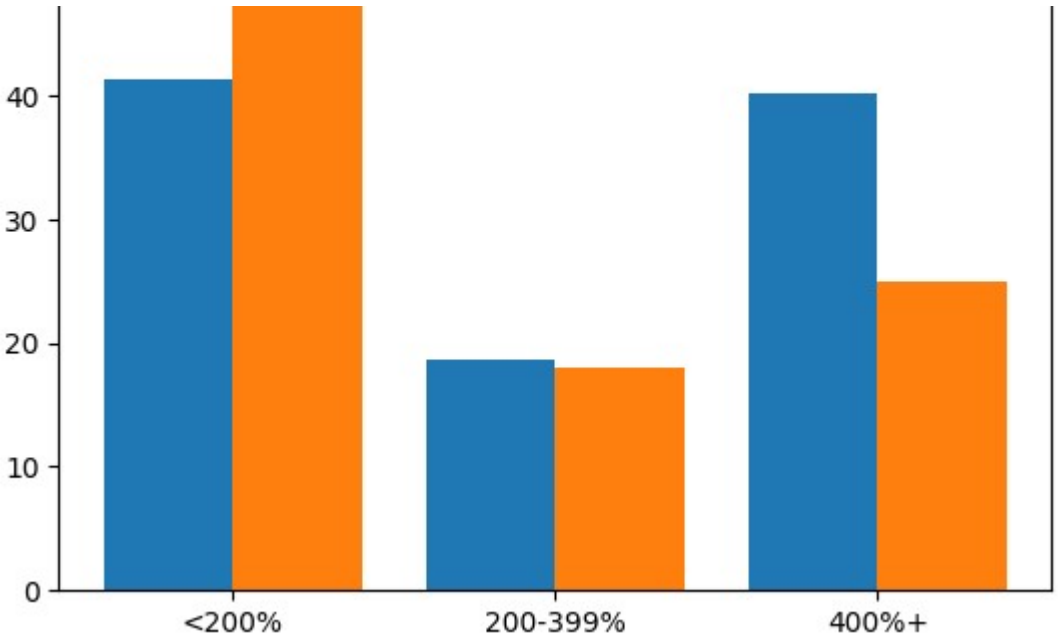
```
plt.title("% Neighborhood Below Poverty Line")
plt.ylabel("Number of Participants")
```

```
Text(0, 0.5, 'Number of Participants')
```




```
import numpy as np

pp = list()
for val in vals:
  pp.append(val*100/sum(vals))

X_axis = np.arange(len(Poverty))
plt.bar(X_axis - 0.2, pp, 0.4, label = "Non Diabetic")

pp2 = list()
for val in vals2:
  pp2.append(val*100/sum(vals2))


plt.bar(X_axis + 0.2, pp2, 0.4, label = "Diabetic")
plt.xticks(X_axis, Poverty)
plt.legend()
```

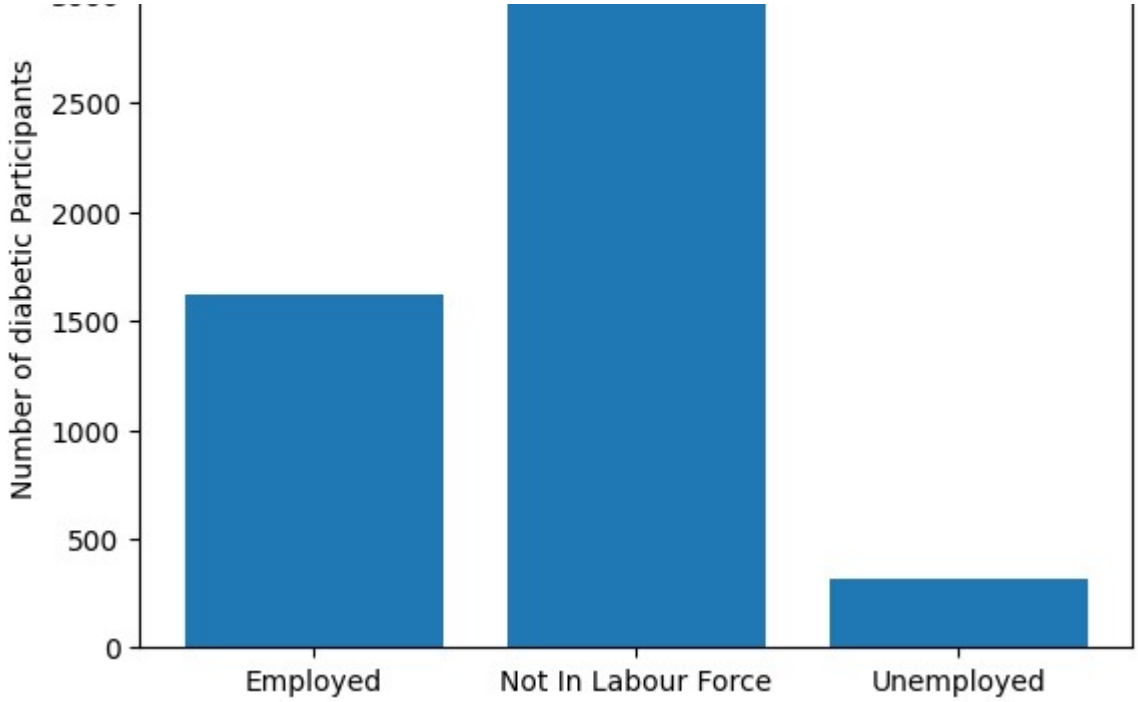
```
<matplotlib.legend.Legend at 0x7bc576d6bf40>
```



We see a higher percentage of diabetics being from the < 200% category. This could be a result of unhealthier diets in this group or could also be due to a lack of ability to test and report in the other 2 groups. However for the sake of the project we have considered all the data to be accurate from these groups

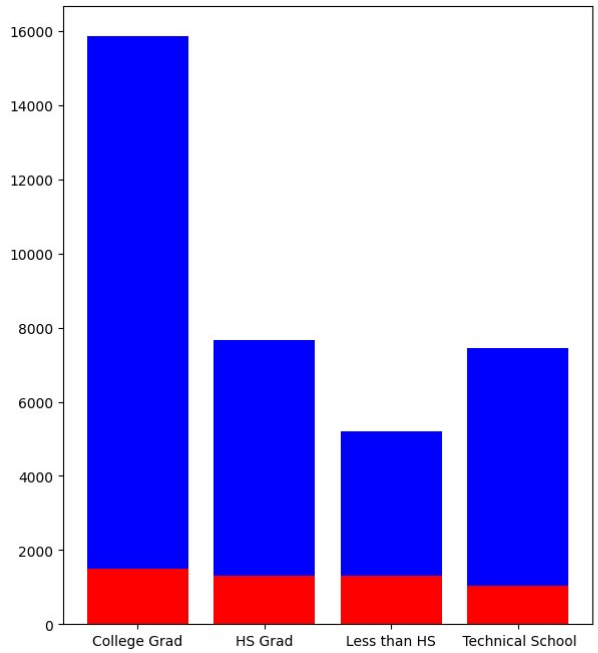
Double-click (or enter) to edit

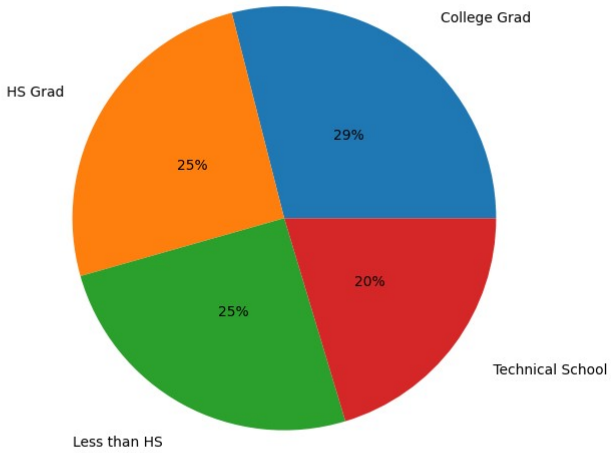
```
import matplotlib.pyplot as plt

work_cols = ['emp_Employed', 'emp_Not_in_labour_force', 'emp_Unemployed']
work_columns = ['Employed','Not In Labour Force','Unemployed']
pos = list()
neg = list()

for col in work_cols:
  pos.append(result.groupby([col,'diabetes'])['pcp'].count()[1][1])
  neg.append(result.groupby([col,'diabetes'])['pcp'].count()[1][2])

from matplotlib.pyplot import figure

plt.bar(work_columns, pos)
plt.title("Employment vs Diabetes")
plt.ylabel("Number of diabetic Participants")
plt.show()
```



This data considers anyone who is not retired as a person not in the labour force and thus we have a high diabetics who are not in the labour force. Additionally, most of the participants in the 45-64 category who are also at the risk of diabetes are likely to be employed as well and would account for the numbers in the employed group

```
import matplotlib.pyplot as plt

ed_cols = ['education_college_grad', 'education_high_school_grad', 'education_less_than_h
ed_columns = ['College Grad', 'HS Grad', 'Less than HS', ' Technical School']
pos = list()
neg = list()

for col in ed_cols:
  pos.append(result.groupby([col,'diabetes'])['pcp'].count()[1][1])
  neg.append(result.groupby([col,'diabetes'])['pcp'].count()[1][2])

pp = list()
for val in pos:
  pp.append(val*100/sum(pos))

from matplotlib.pyplot import figure

fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(15, 8)
ax1.bar(ed_columns, pos, color='r')
ax1.bar(ed_columns, neg, bottom=pos, color='b')
ax2.pie(pp, labels=ed_columns, autopct='%1.0f%%', pctdistance=0.5, labeldistance=1.2)
plt.show()
```





The increased likeliness of High School Dropouts of being diabetic may correlate to other demographic factors and poverty levels as we have already seen people in these groups are more likely to be diabetic

```
from matplotlib.pyplot import figure

plt.bar(work_columns, pos)
plt.title("Employment vs Diabetes")
plt.ylabel("Number of diabetic Participants")
plt.show()
```

Double-click (or enter) to edit

Double click (or enter) to edit

Exploratory Data Analyses on 2020 Data

This section will focus on some of the features exclusively available in 2020 dataset to try and understand how these new questions correlate to diabetes. However these columns were not used in the Data Modelling.

```
uploaded = files.upload()
```

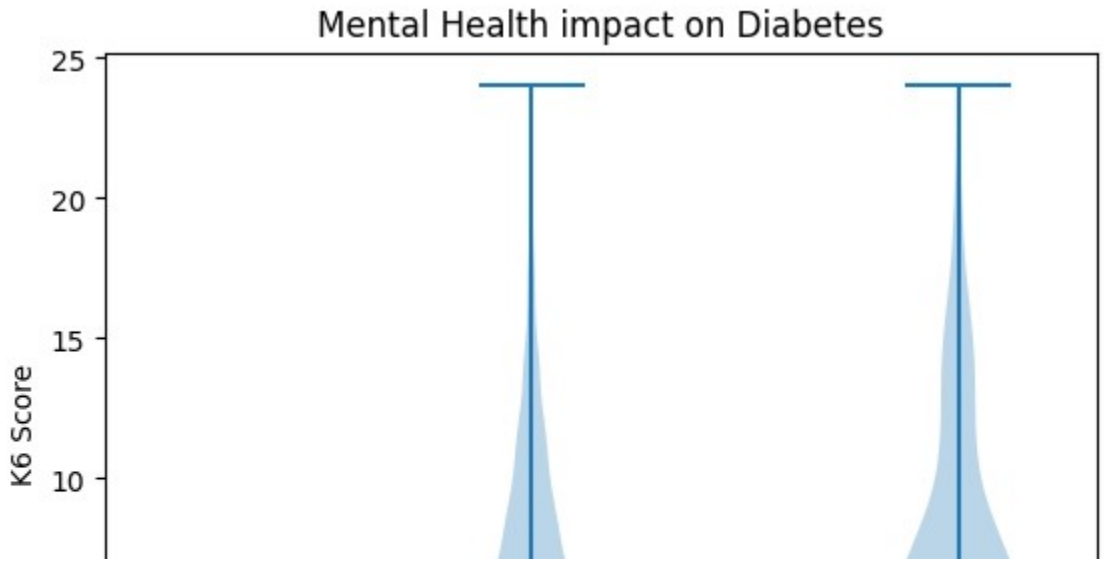
Browse... No files selected. Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

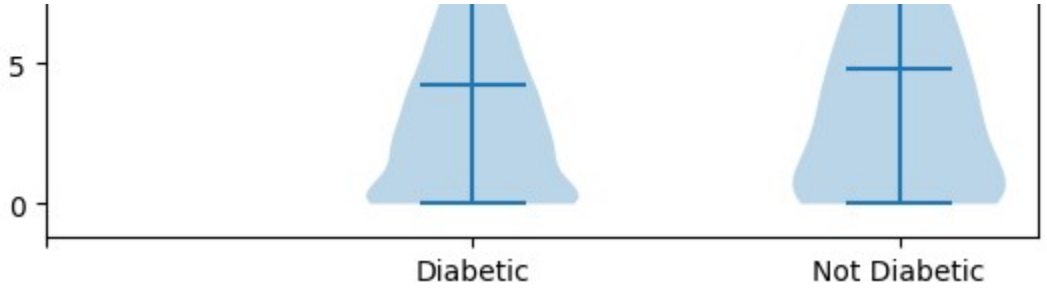
Saving 2020_2.csv to 2020_2.csv

```
df20_eda = pd.read_csv("2020_2.csv")
```

```
have_diabetes_20 = df20_eda.loc[df20_eda['diabetes20'] == 1]
dont_have_20 = df20_eda.loc[df20_eda['diabetes20'] == 2]
have_diabetes_20 = have_diabetes_20.dropna()
dont_have_20 = dont_have_20.dropna()

Diabetic = ['', 'Diabetic', 'Not Diabetic']
y1 = dont_have_20['k6']
y2 = have_diabetes_20['k6']
y = [y1,y2]
plt.violinplot(y, showmeans = True)
X_axis = np.arange(len(Diabetic))
plt.xticks(X_axis, Diabetic)
plt.title("Mental Health impact on Diabetes")
plt.ylabel("K6 Score")
```

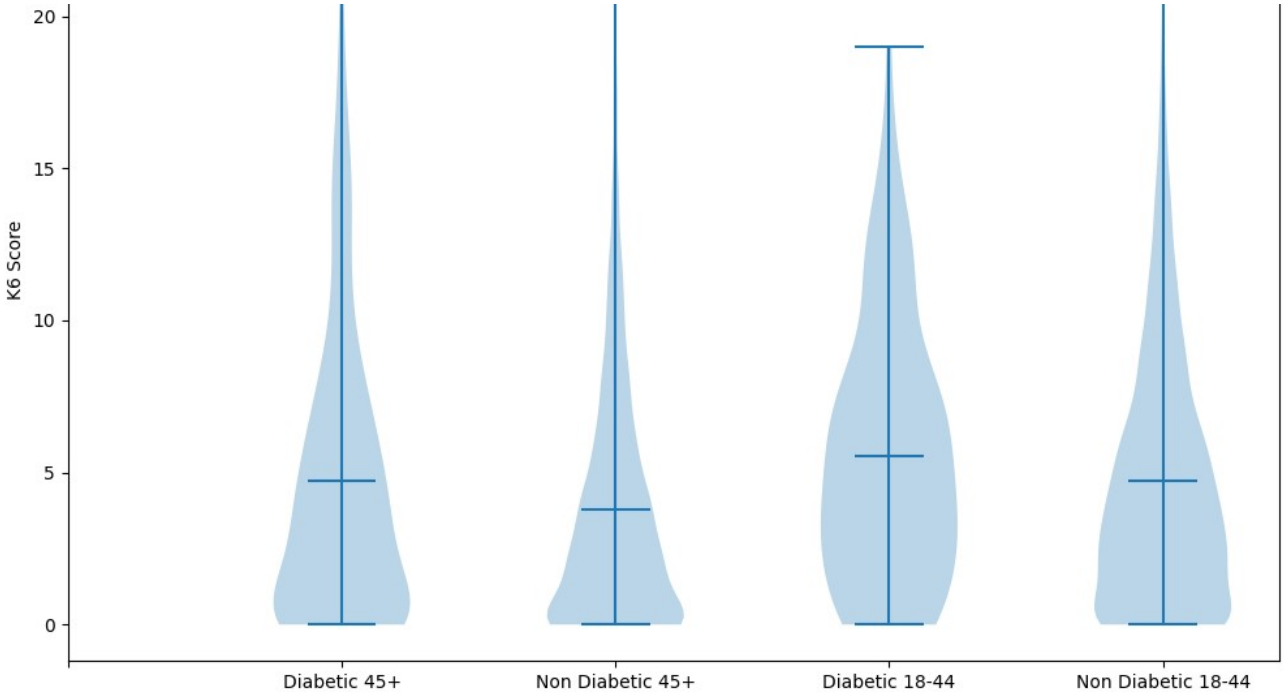
```
Text(0, 0.5, 'K6 Score')
```



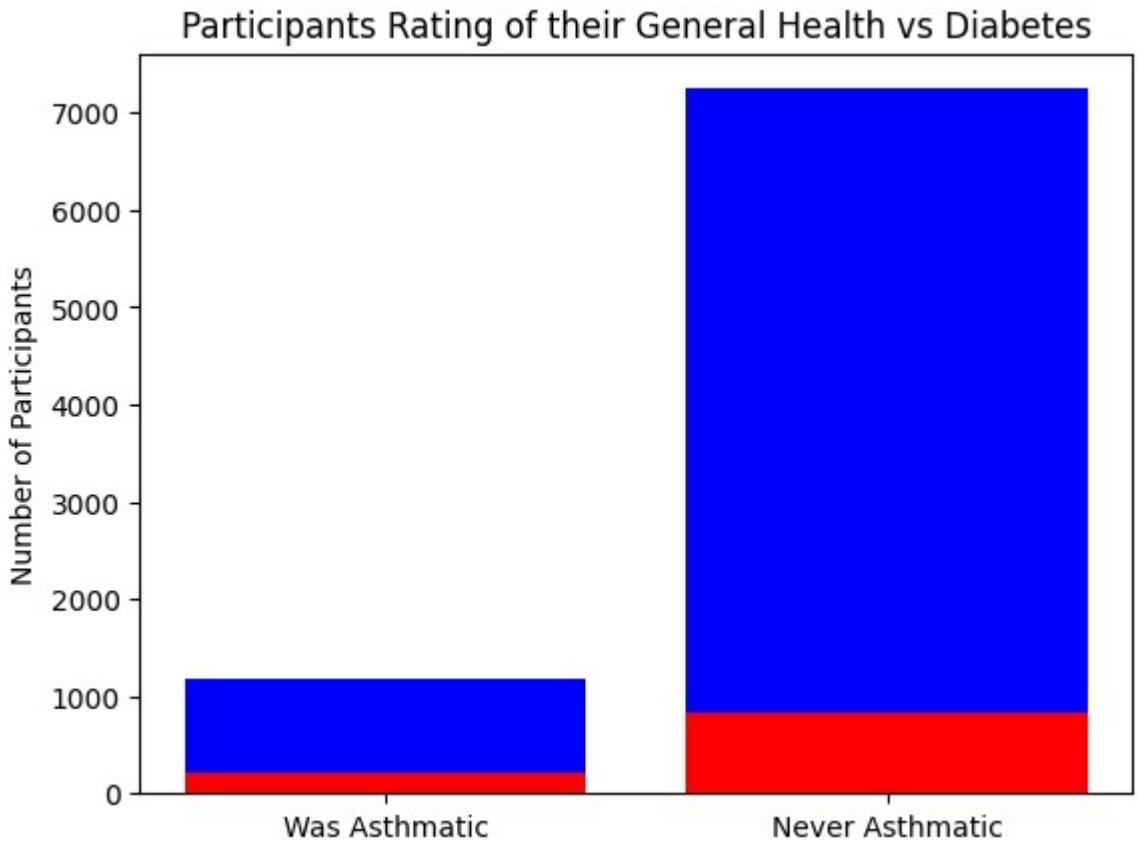



```
hd2045a = have_diabetes_20.loc[df20_eda['agegroup5_45_64'] == 1]
hd2065a = have_diabetes_20.loc[df20_eda['agegroup5_above_65'] == 1]
frames= [hd2045a, hd2065a]
res1 = pd.concat(frames)

hd2045a = dont_have_20.loc[df20_eda['agegroup5_45_64'] == 1]
hd2065a = dont_have_20.loc[df20_eda['agegroup5_above_65'] == 1]
frames= [hd2045a, hd2065a]
res3 = pd.concat(frames)

hd2024a = have_diabetes_20.loc[df20_eda['agegroup5_18_24'] == 1]
hd2029a = have_diabetes_20.loc[df20_eda['agegroup5_25_29'] == 1]
hd2044a = have_diabetes_20.loc[df20_eda['agegroup5_30_44'] == 1]
frames= [hd2024a, hd2029a, hd2044a]
res2 = pd.concat(frames)

hd2024a = dont_have_20.loc[df20_eda['agegroup5_18_24'] == 1]
hd2029a = dont_have_20.loc[df20_eda['agegroup5_25_29'] == 1]
hd2044a = dont_have_20.loc[df20_eda['agegroup5_30_44'] == 1]
frames= [hd2024a, hd2029a, hd2044a]
res4 = pd.concat(frames)

figure(figsize=(12,8))
Diabetic = ['', 'Diabetic 45+', 'Non Diabetic 45+', 'Diabetic 18-44', 'Non Diabetic 18-4
y1 = res1['k6']
y2 = res2['k6']
y3 = res3['k6']
y4 = res4['k6']
y = [y1, y3 , y2, y4]
plt.violinplot(y, showmeans = True)
X_axis = np.arange(len(Diabetic))
plt.xticks(X_axis, Diabetic)
plt.title("Mental Health impact on Diabetes - Age Split")
plt.ylabel("K6 Score")
```

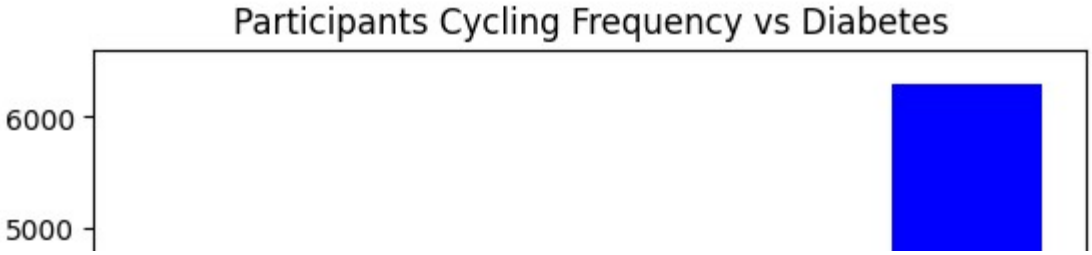
```
Text(0, 0.5, 'K6 Score')
```

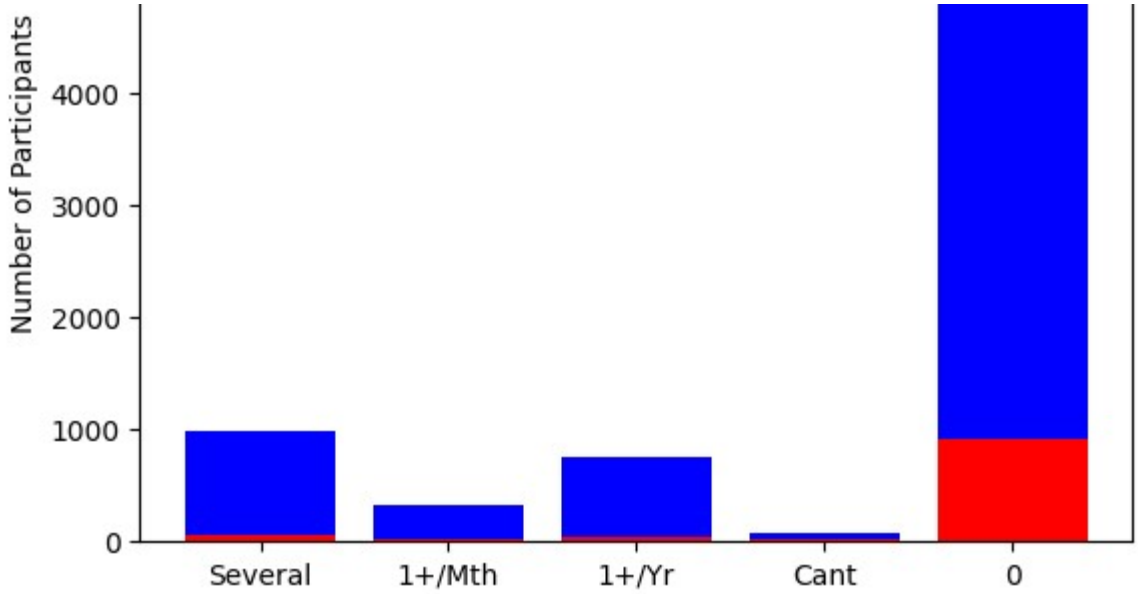



```
import matplotlib.pyplot as plt
everasthma = ['Was Asthmatic', 'Never Asthmatic']
pos = list()
neg = list()


for val in range(1,3):

  pos.append(df20_eda.groupby(['everasthma','diabetes20'])['pcp20'].count()[val][1])
  neg.append(df20_eda.groupby(['everasthma','diabetes20'])['pcp20'].count()[val][2])


plt.bar(everasthma, pos, color='r', label='Diabetic')
plt.bar(everasthma, neg, bottom =pos , color='b', label='Non Diabetic')

plt.title("Participants Rating of their General Health vs Diabetes")
plt.ylabel("Number of Participants")
plt.show()
```

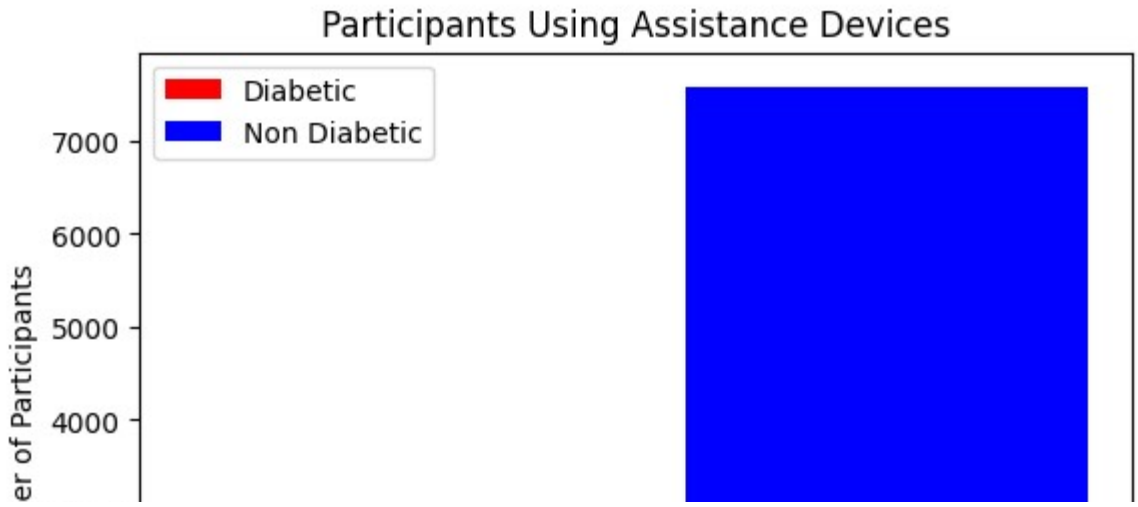


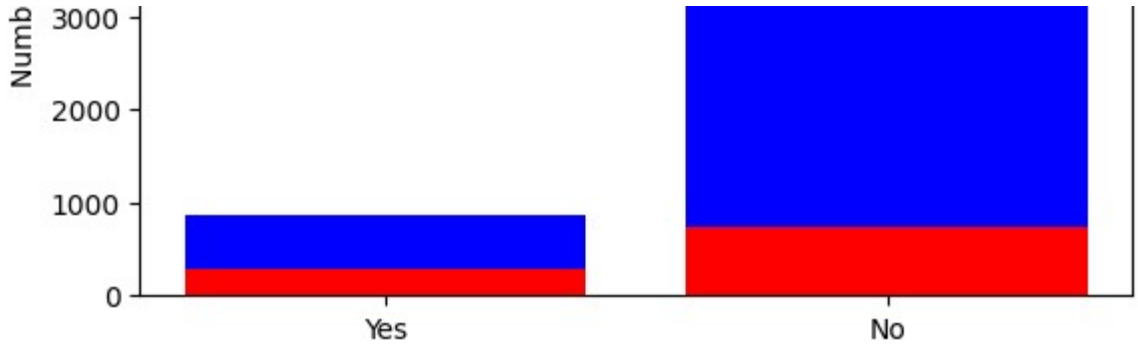

```
import matplotlib.pyplot as plt
everasthma = ['Several', '1+/Mth', '1+/Yr', 'Cant', '0']
pos = list()
neg = list()


for val in range(1,6):

  pos.append(df20_eda.groupby(['cycling20','diabetes20'])['pcp20'].count()[val][1])
  neg.append(df20_eda.groupby(['cycling20','diabetes20'])['pcp20'].count()[val][2])


plt.bar(everasthma, pos, color='r', label='Diabetic')
plt.bar(everasthma, neg, bottom =pos , color='b', label='Non Diabetic')

plt.title("Participants Cycling Frequency vs Diabetes")
plt.ylabel("Number of Participants")
plt.show()
```






```
import matplotlib.pyplot as plt
everasthma = ['Yes', 'No']
pos = list()
neg = list()


for val in range(1,3):

  pos.append(df20_eda.groupby(['assistdevice','diabetes20'])['pcp20'].count()[val][1])
  neg.append(df20_eda.groupby(['assistdevice','diabetes20'])['pcp20'].count()[val][2])


plt.bar(everasthma, pos, color='r', label='Diabetic')
plt.bar(everasthma, neg, bottom =pos , color='b', label='Non Diabetic')

plt.title("Participants Using Assistance Devices")
plt.ylabel("Number of Participants")
plt.legend()
plt.show()
```





Trends in Diabetes vs Age over the Years

```
import matplotlib.pyplot as plt
age_split = dict()
age_cols = ['agegroup_18_24', 'agegroup_25_29', 'agegroup_30_44', 'agegroup_45_64', 'age
age_columns = ['18-24','25-29','30-44','45-64','65+']
years_cols = ['2017', '2018', '2019', '2020']

dataframes = [df17, df18, df19, df20]
years = list()
for df in dataframes:
  pos = list()
  neg = list()
  for col in age_cols:
    pos.append(df.groupby([col,'diabetes'])['pcp'].count()[1][1])
    neg.append(df.groupby([col,'diabetes'])['pcp'].count()[1][2])
  years.append([pos,neg])


ys = list()
for year in years:
  pp = list()
  for val in range(len(year[0])):
    pp.append(year[0][val]*100/sum(year[0]))
  ys.append(pp)


percents = list()
for i in range(5):
  percent = list()
  for j in range(4):
    percent.append(ys[j][i])
  percents.append(percent)
```

```
print((percents))
```

```
[[0.63875088715401, 0.6068779501011463, 1.2126111560226354, 0.7655502392344498], [1.
```

```
from matplotlib.pyplot import figure

figure(figsize=(12,8))
plt.bar(years_cols, percents[0])
plt.bar(years_cols, percents[1], bottom=percents[0])
plt.bar(years_cols, percents[2], bottom=percents[1])
plt.bar(years_cols, percents[3], bottom=percents[2])
plt.bar(years_cols, percents[4], bottom=percents[3])
plt.ylabel("Percentage of the population")
plt.title("Age vs Diabetes")
plt.legend(labels=age_columns)
plt.show()
```