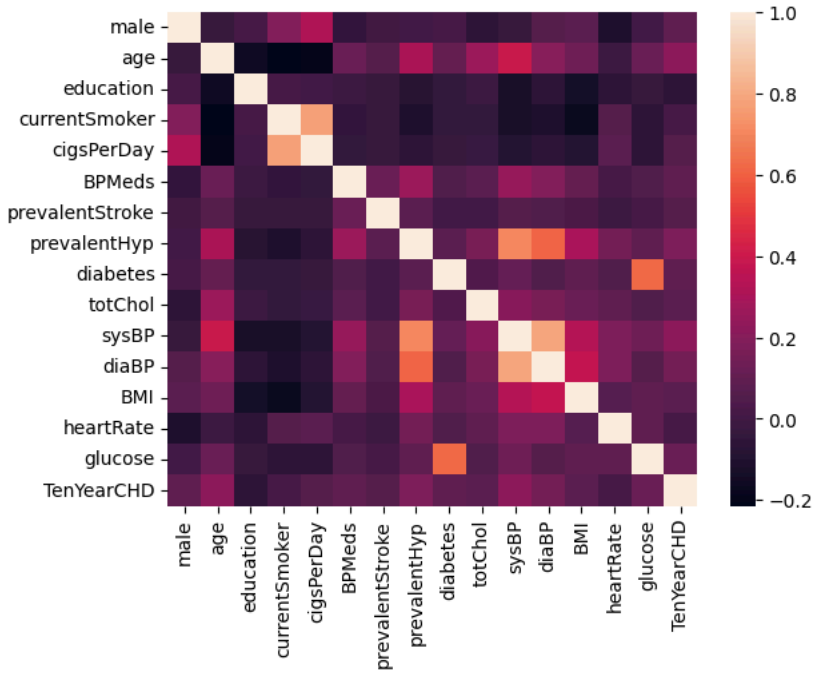
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

```
health_df = pd.read_csv('framingham.csv')
```

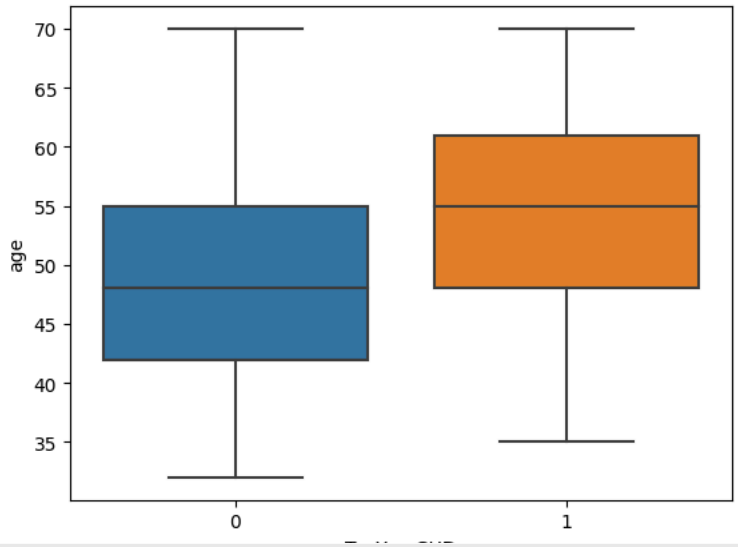
```
sns.heatmap(health_df.corr())
```

```
<Axes: >
```



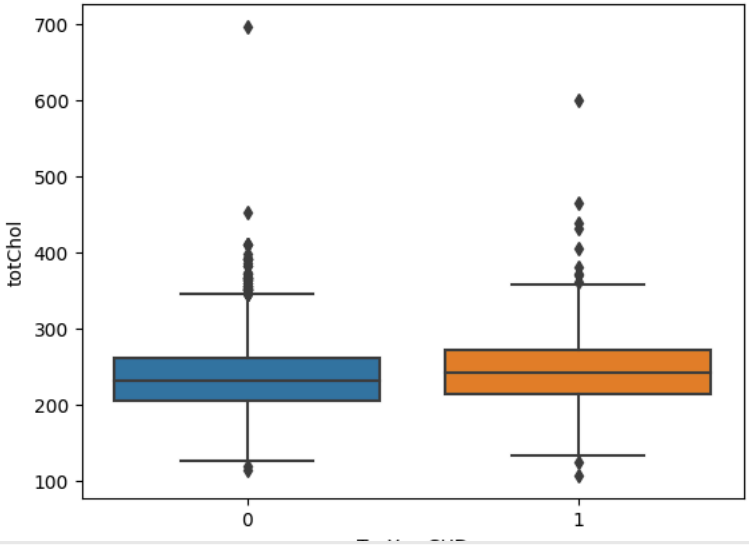

```
yes_CHD_df = health_df[health_df['TenYearCHD'] == 1]
no_CHD_df = health_df[health_df['TenYearCHD'] == 0]

#people with heart disease tend to be older
sns.boxplot(health_df, x='TenYearCHD', y='age')
t1, p1 = stats.ttest_ind(yes_CHD_df['age'], no_CHD_df['age'], equal_var=False, nan_policy='omit')
print(p1)
```

```
1.0406170043995591e-48
```



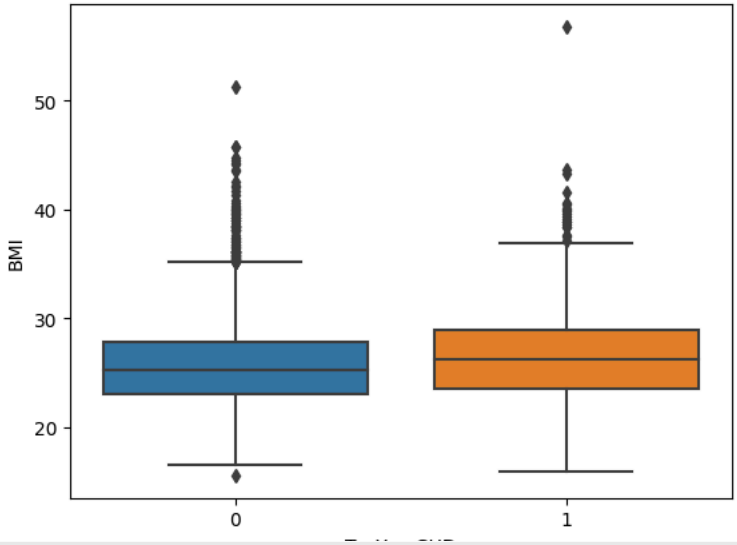

```
sns.boxplot(health_df, x='TenYearCHD', y='totChol')
stats.ttest_ind(yes_CHD_df['totChol'], no_CHD_df['totChol'], equal_var=False, nan_policy='omit')
```

```
Ttest_indResult(statistic=4.997692247478818, pvalue=7.074190796880142e-07)
```



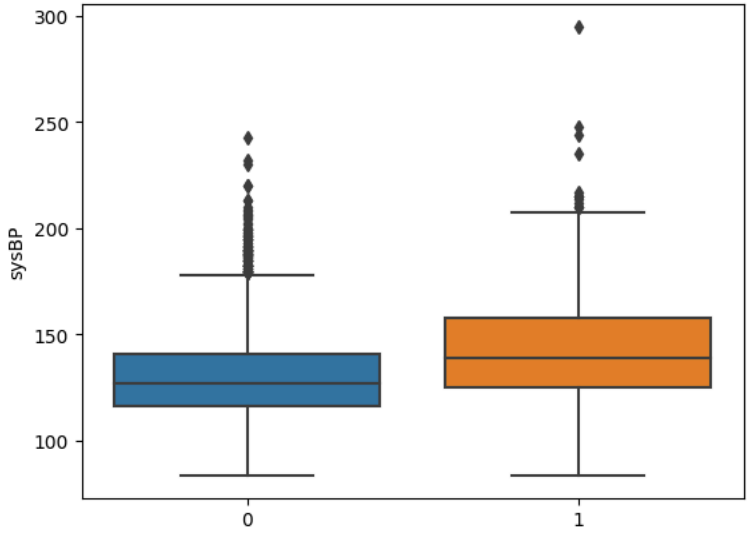

```
sns.boxplot(health_df, x='TenYearCHD', y='BMI')
stats.ttest_ind(yes_CHD_df['BMI'], no_CHD_df['BMI'], equal_var=False, nan_policy='omit')
```

```
Ttest_indResult(statistic=4.482156194116062, pvalue=8.445526345053698e-06)
```

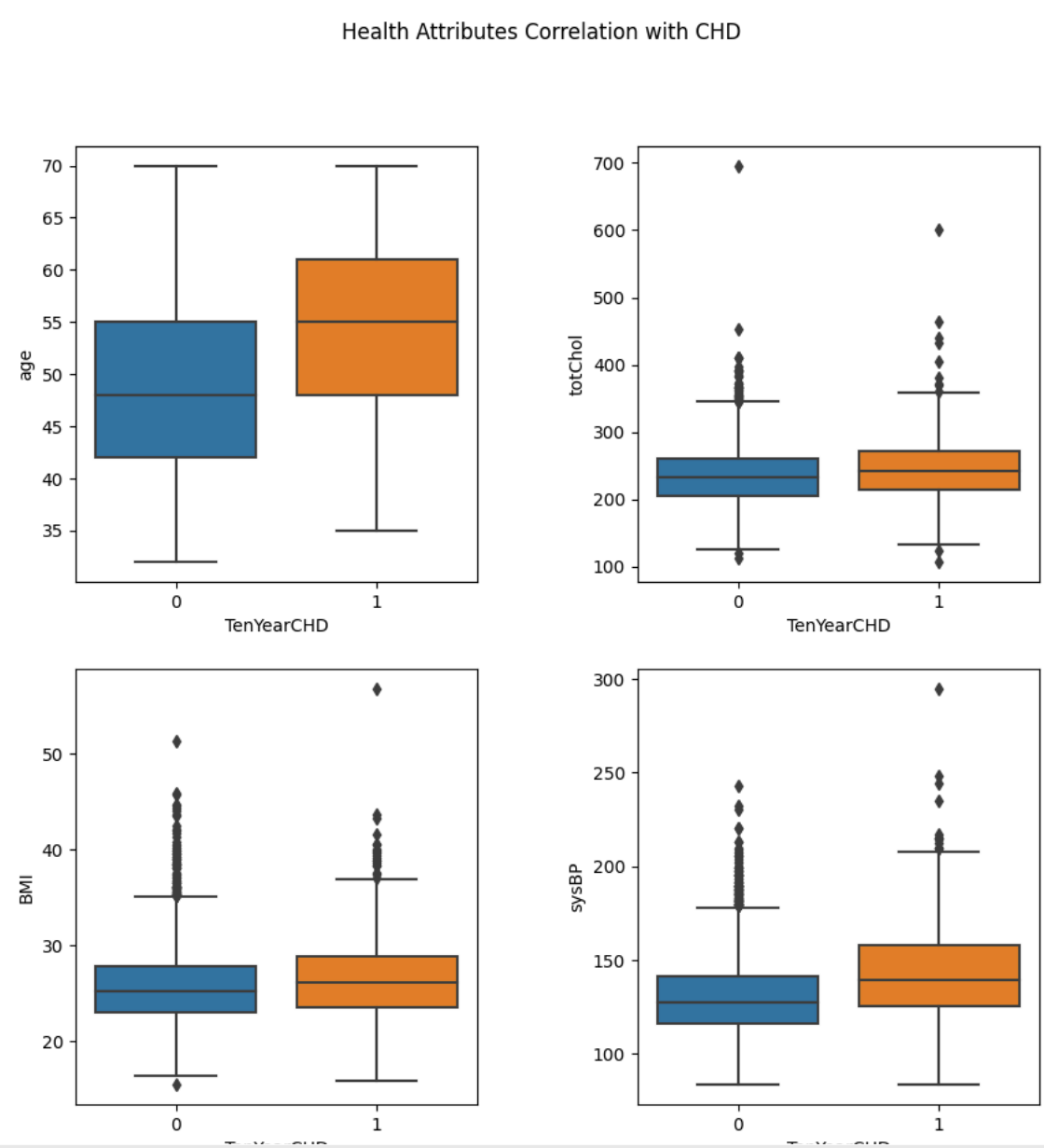



```
sns.boxplot(health_df, x='TenYearCHD', y='sysBP')
stats.ttest_ind(yes_CHD_df['sysBP'], no_CHD_df['sysBP'], equal_var=False, nan_policy='omit')
```

```
Ttest_indResult(statistic=12.015100860431446, pvalue=1.2096163396951795e-30)
```




```
fig, axs = plt.subplots(2, 2, figsize=(10, 10))
fig.suptitle('Health Attributes Correlation with CHD')

sns.boxplot(ax=axs[0, 0], data=health_df, x='TenYearCHD', y='age')
sns.boxplot(ax=axs[0, 1], data=health_df, x='TenYearCHD', y='totChol')
sns.boxplot(ax=axs[1, 0], data=health_df, x='TenYearCHD', y='BMI')
sns.boxplot(ax=axs[1, 1], data=health_df, x='TenYearCHD', y='sysBP')

plt.subplots_adjust(hspace=0.2, wspace=0.4)
```




```
from scipy import stats
```

```
#null hypothesis: the means are the same
#small p value means we reject the null hypothesis
#bc we have a small p-value, the mean variable for people with and w/o CHD are different

print(yes_CHD_df['totChol'].mean())
print(no_CHD_df['totChol'].mean())
print(yes_CHD_df['totChol'].mean()-no_CHD_df['totChol'].mean())
stats.ttest_ind(yes_CHD_df['totChol'], no_CHD_df['totChol'], equal_var=False, nan_policy='omit')
```

```
245.38897637795276
235.17253025612158
10.216446121831183
Ttest_indResult(statistic=4.997692247478818, pvalue=7.074190796880142e-07)
```

```
#comparing proportion of smokers with CHS and no CHD
#two sample proportion test (one sample with ChD and one without)
#but we are using t test because it is the same mathematically since the data is binary

#people that have had a stroke are more likely to have CHD
print(yes_CHD_df['prevalentStroke'].mean())
```

```
#people that have diabetes are more likely to have CHD
print(yes_CHD_df['prevalentHyp'].mean())
print(no_CHD_df['prevalentHyp'].mean())
print(yes_CHD_df['prevalentHyp'].mean()-no_CHD_df['prevalentHyp'].mean())
stats.ttest_ind(yes_CHD_df['prevalentHyp'], no_CHD_df['prevalentHyp'], equal_var=False, nan_policy='omit')
```

```
0.5046583850931677
0.27573734001112965
0.2289210450820381
Ttest_indResult(statistic=10.859823031458566, pvalue=8.561360042008926e-26)
```

```
#people that consume BPMeds are more likely to have CHD
print(yes_CHD_df['BPMeds'].mean())
print(no_CHD_df['BPMeds'].mean())
print(yes_CHD_df['BPMeds'].mean()-no_CHD_df['BPMeds'].mean())
stats.ttest_ind(yes_CHD_df['BPMeds'], no_CHD_df['BPMeds'], equal_var=False, nan_policy='omit')
```

```
0.06477093206951026
0.023367117117117118
0.04140381495239315
Ttest_indResult(statistic=4.094088761149889, pvalue=4.719245921147637e-05)
```

```
p_vals = []
t1, p1 = stats.ttest_ind(yes_CHD_df['currentSmoker'], no_CHD_df['currentSmoker'], equal_var=False, nan_policy='omit')
p_vals.append(p1)
t2, p2 = stats.ttest_ind(yes_CHD_df['BPMeds'], no_CHD_df['BPMeds'], equal_var=False, nan_policy='omit')
p_vals.append(p2)
t3, p3 = stats.ttest_ind(yes_CHD_df['prevalentStroke'], no_CHD_df['prevalentStroke'], equal_var=False, nan_policy='omit')
p_vals.append(p3)
t4, p4 = stats.ttest_ind(yes_CHD_df['diabetes'], no_CHD_df['diabetes'], equal_var=False, nan_policy='omit')
p_vals.append(p4)
print(p_vals)
```

```
[0.20573113267977536, 4.719245921147637e-05, 0.01167052350702821, 1.3420705485055781e-05]
```

```
fig, ax = plt.subplots()
bars = ax.bar(range(len(p_vals)), p_vals)

# Red dashed horizontal line
ax.axhline(y=0.05, color='red', linestyle='--')
```