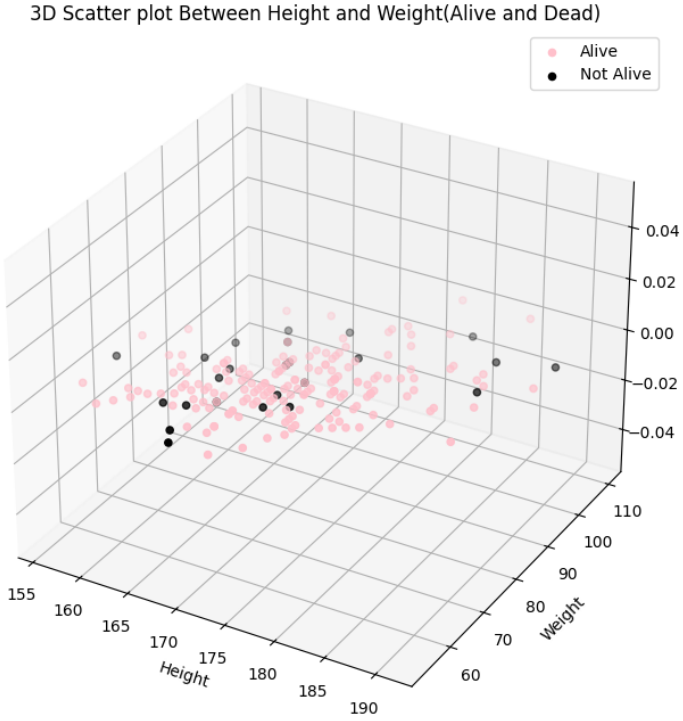
In [ ]:
```
import numpy as np
from numpy import linspace
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
from mpl_toolkits.mplot3d import Axes3D
from scipy import signal
from mpl_toolkits.basemap import Basemap
%matplotlib widget

mortality=pd.read_csv('Q8_MortalityDataset.csv')
print(mortality.head(10))
```

```
   AGE  HEIGHT  WEIGHT  CHOL   SMOKE BLOOD   MORT
0   20     176      77   195  nonsmo     b  alive
1   53     167      56   250  sigare     o   dead
2   44     170      80   304  sigare     a   dead
3   37     173      89   178  nonsmo     o  alive
4   26     170      71   206  sigare     o  alive
5   41     165      62   284  sigare     o  alive
6   39     174      75   232  sigare     o  alive
7   28     171      68   152    pipe     a  alive
8   33     180     100   209  sigare     a  alive
9   39     166      74   150  sigare     a  alive
```

In [ ]:
```
alive=mortality[mortality['MORT']=='alive']
dead=mortality[mortality['MORT']=='dead']
#For Alive
a=alive['HEIGHT']
b=alive['WEIGHT']
#For Dead
x=dead['HEIGHT']
y=dead['WEIGHT']
```

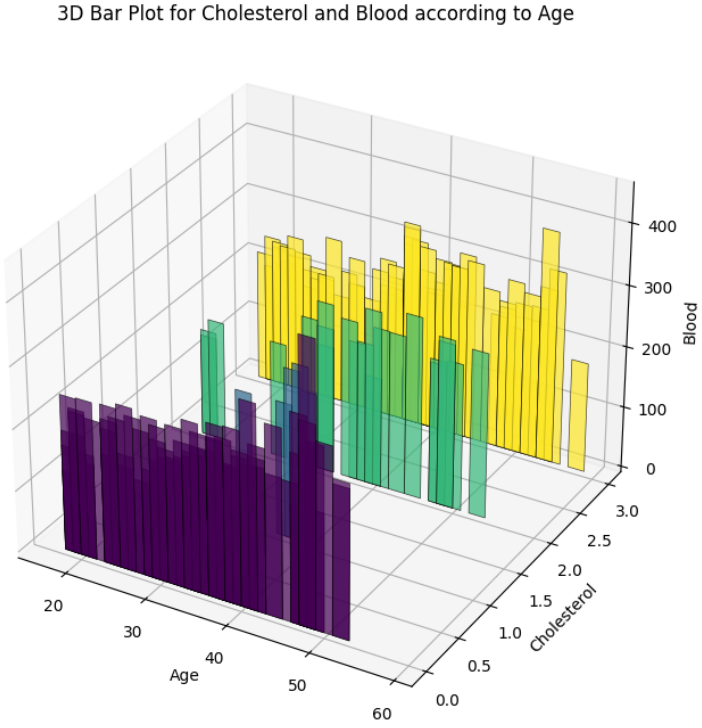
In [ ]:
```
#1. Draw a 3-D scatter plot for variables HEIGHT, WEIGHT, &MORT
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot for alive individuals
ax.scatter(alive['HEIGHT'], alive['WEIGHT'], c='pink', label='Alive')

# Scatter plot for dead individuals
ax.scatter(dead['HEIGHT'], dead['WEIGHT'], c='black', label='Not Alive')

# Set axis labels
ax.set_title('3D Scatter plot Between Height and Weight(Alive and Dead)')
ax.set_xlabel('Height')
ax.set_ylabel('Weight')

# Add legend
ax.legend()

# Show the plot
plt.show()
```

Figure



In [ ]:
```
#2. Draw a 3-D Bar plot for variables CHOL and BLOOD according to AGE
from matplotlib.colors import Normalize

# Extract relevant columns
age = mortality['AGE']
chol = pd.to_numeric(mortality['CHOL'], errors='coerce')
blood = mortality['BLOOD'].astype('category').cat.codes

# Create a color map for 'BLOOD' categories
normalize = Normalize(vmin=blood.min(), vmax=blood.max())
colormap = cm.viridis

# Create a 3D bar plot
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

# Plotting with borders and adjusted bar size
bars = ax.bar(age, chol, blood, zdir='y', width=2, color=colormap(normalize(bloc
              alpha=0.7, edgecolor='black', linewidth=0.5)

# Set labels and title
ax.set_xlabel('Age')
ax.set_ylabel('Cholesterol')
ax.set_zlabel('Blood')
ax.set_title('3D Bar Plot for Cholesterol and Blood according to Age')

# Show the plot
plt.show()
```

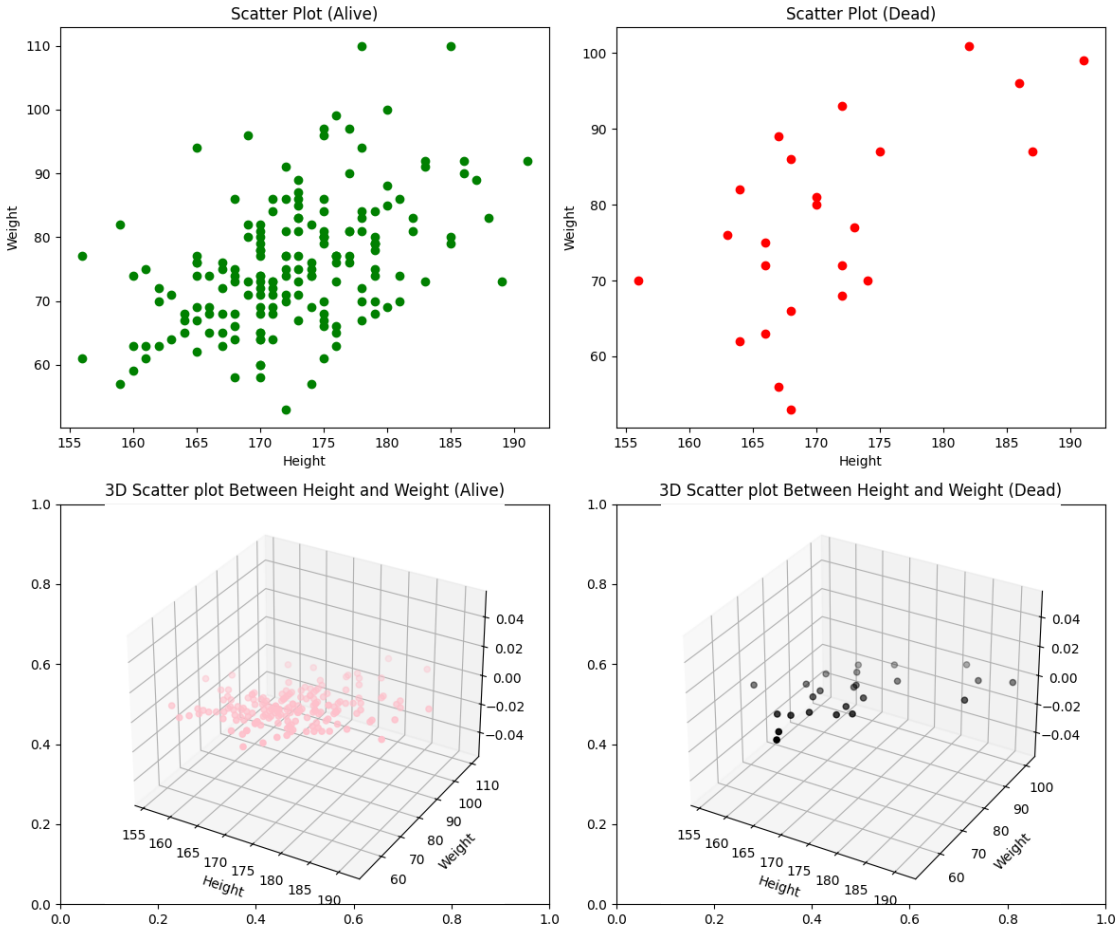
Figure



In [ ]:
```
#3. Generate an image of four scatter plots(2D plot for alive, 2D plot for notal
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))

# Scatter plot for alive data
axes[0, 0].scatter(a,b, color='green')
axes[0, 0].set_title('Scatter Plot (Alive)')
axes[0, 0].set_xlabel('Height')
axes[0, 0].set_ylabel('Weight')

# Scatter plot for dead data
axes[0, 1].scatter(x,y, color='red')
axes[0, 1].set_title('Scatter Plot (Dead)')
axes[0, 1].set_xlabel('Height')
axes[0, 1].set_ylabel('Weight')

# 3D Scatter plot for alive data
ax1 = fig.add_subplot(223, projection='3d')
ax1.scatter(a,b, c='pink')
ax1.set_title('3D Scatter plot Between Height and Weight (Alive)')
ax1.set_xlabel('Height')
ax1.set_ylabel('Weight')

# 3D Scatter plot for dead data
ax2 = fig.add_subplot(224, projection='3d')
ax2.scatter(x,y, c='black')
ax2.set_title('3D Scatter plot Between Height and Weight (Dead)')
ax2.set_xlabel('Height')
ax2.set_ylabel('Weight')
```

```
plt.tight_layout()
plt.show()
```

Figure



In [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from matplotlib.colors import Normalize

# Convert CHOL and BLOOD to numeric values for better plotting
mortality['CHOL'] = pd.to_numeric(mortality['CHOL'], errors='coerce')
mortality['BLOOD'] = mortality['BLOOD'].astype('category').cat.codes

# Create a color map for 'BLOOD' categories
normalize = Normalize(vmin=mortality['BLOOD'].min(), vmax=mortality['BLOOD'].max
colormap = cm.viridis

# Get unique 'BLOOD' categories
blood_categories = mortality['BLOOD'].unique()

# Create subplots
fig, axs = plt.subplots(2, 2, figsize=(12, 10), subplot_kw={'projection': '3d'})
axs = axs.flatten()

# Plotting for each 'BLOOD' category
for i, blood_category in enumerate(blood_categories):
    subset_df = mortality[mortality['BLOOD'] == blood_category]

    bars = axs[i].bar(subset_df['AGE'], subset_df['CHOL'], subset_df['BLOOD'],
```

```
                        zdir='y', width=2, color=colormap(normalize(subset_df['BLO
                        alpha=0.7, edgecolor='black', linewidth=0.5)

    axs[i].set_xlabel('AGE')
    axs[i].set_ylabel('CHOL')
    axs[i].set_zlabel('BLOOD')
    axs[i].set_title(f'3D Bar Plot for BLOOD Category {blood_category}')

# Adjust layout
plt.tight_layout()
# Show the plot
plt.show()
```

Figure