**Q3:**

**Loading Excel Sheets into Separate DataFrames**

**Introduction**

The goal of this task was to write a Python script that opens the Excel file containing multiple tables and loads each sheet into a separate DataFrame.

**Preprocessing**

The preprocessing stage for this task involved identifying the Excel file (rehospitalization.xlsx) containing the data. The file was assumed to contain multiple sheets, each representing a different table of data. No additional cleaning or manipulation of the file was needed at this stage.

**Processing**

A Python script was developed to read all sheets from the Excel file and load them into a dictionary of DataFrames. The script utilizes the pandas library, specifically the pd.read\_excel() function with the sheet\_name=None parameter to ensure that all sheets are read at once. Each sheet was then stored in a dictionary with the sheet name as the key and the corresponding DataFrame as the value.

**Python Code:**

import pandas as pd

# Open the Excel file

file\_path = 'rehospitalization.xlsx'

# Read all the sheets in the file

excel\_data = pd.read\_excel(file\_path, sheet\_name=None)

# Create a dictionary where the key is the sheet name and the value is the corresponding DataFrame

dataframes = {sheet: pd.DataFrame(data) for sheet, data in excel\_data.items()}

# Print the names of the sheets to verify

print("Loaded sheets:")

for sheet in dataframes.keys():

print(sheet)

**Output:**

Loaded sheets:

טבלאות-סיכום

GeneralData

Drugs

hospitalization1

unitsAdmissions

unitsOccupancyRate

erAdmission

erDoctor

hDoctor

erBeforeHospitalization2

hospitalization2

ICD9

רופאים מאשפזים מהמלרד

רופאים משחררים מהאשפוז

טבלאות

**Q8:**

**Exploratory Data Analysis (EDA) on General Data**

**Introduction**

The goal of this task was to perform an Exploratory Data Analysis (EDA) on the GeneralData table from the rehospitalization dataset. The analysis aimed to understand the distribution of various parameters and examine potential correlations between them.

**Preprocessing**

The GeneralData table was loaded from the rehospitalization.xlsx file into a Pandas DataFrame. Basic information about the DataFrame, including column names, data types, and summary statistics, was displayed to understand the structure and content of the data. No additional preprocessing was necessary as the data was already clean.

**Processing**

The EDA process involved the following steps:

1. **Displaying Basic Information**: The info() method was used to display the structure of the DataFrame, and the describe() method provided summary statistics for the numeric columns.
2. **Histogram Plots**: Histograms were generated for each numeric column to visualize their distributions. Kernel Density Estimation (KDE) curves were overlaid on the histograms to understand the data distribution more clearly.
3. **Correlation Heatmap**: A correlation matrix was calculated for all numeric columns, and a heatmap was generated to visualize the correlations between different parameters.

**Python Code:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Open the Excel file

file\_path = 'rehospitalization.xlsx'

# Load the GeneralData table into a DataFrame

general\_data\_df = pd.read\_excel(file\_path, sheet\_name='GeneralData')

# Display basic information about the DataFrame

general\_data\_info = general\_data\_df.info()

# Display summary statistics for the DataFrame

general\_data\_description = general\_data\_df.describe()

# Plot histograms for each numeric column in the DataFrame

numeric\_columns = general\_data\_df.select\_dtypes(include='number').columns

# Set up the matplotlib figure

plt.figure(figsize=(20, 15))

for i, column in enumerate(numeric\_columns, 1):

plt.subplot(5, 4, i)

sns.histplot(general\_data\_df[column], kde=True)

plt.title(f'Histogram of {column}')

plt.tight\_layout()

plt.show()

# Display correlation heatmap

plt.figure(figsize=(15, 10))

correlation\_matrix = general\_data\_df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Correlation Heatmap')

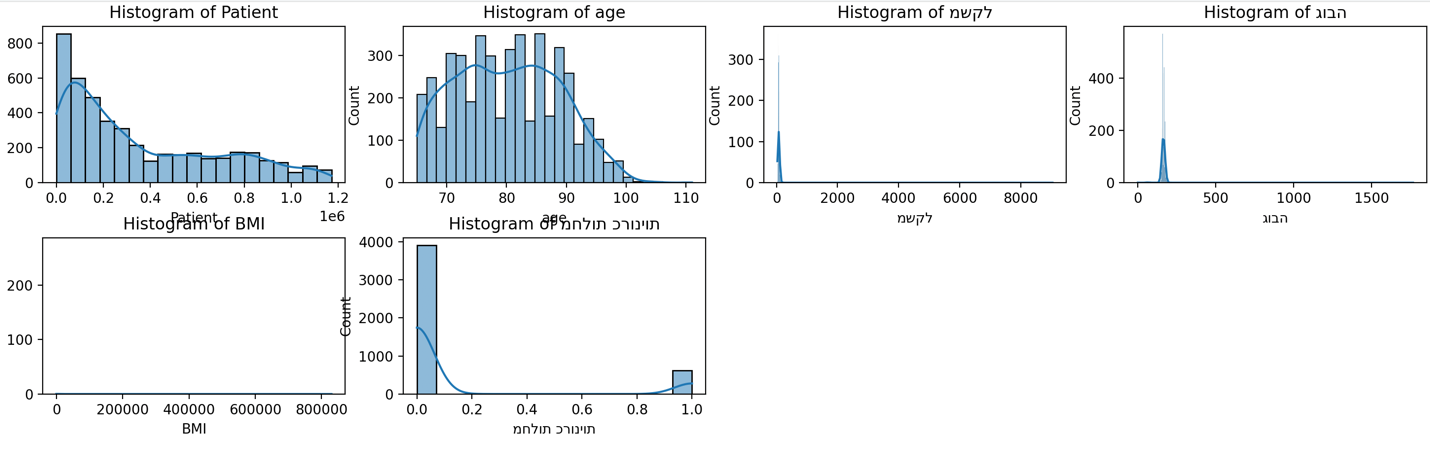
plt.show()

general\_data\_info, general\_data\_description

### Postprocessing

The results of the EDA provided insights into the distribution of the data and the relationships between different variables:

1. **Histograms**: The histograms revealed the distribution of various parameters such as age, weight, height, BMI, and other relevant features. The KDE curves helped to understand the density of the data. The histograms showed that some parameters had skewed distributions, which might be important for further analysis.

Figure 1: 

1. **Correlation Heatmap**: The correlation heatmap illustrated the relationships between different numeric variables in the GeneralData table. The correlations were generally weak, suggesting that the variables might not have strong linear relationships with each other.

Figure 2: [Include the correlation heatmap image here.]

### Conclusion

The EDA provided a comprehensive overview of the GeneralData table. The analysis showed that while the data is mostly well-distributed, there are some skewed variables that may require further investigation. The weak correlations observed suggest that more complex relationships may exist between the variables, which could be explored in future analyses.

**Q25:**

**Analyzing the Relationship between Doctor Workload and Rehospitalization**

**Introduction**

The objective of this task was to analyze the relationship between doctor workload and the likelihood of patient rehospitalization. The analysis aimed to determine if an increase in doctor workload correlates with an increased risk of patient rehospitalization.

**Preprocessing**

Two datasets were loaded from the rehospitalization.xlsx file:

* **Doctor Data (hDoctor)**: Contains information about doctors, including their IDs and the number of patients they handled.
* **Hospitalization Data (hospitalization1)**: Includes details about patient hospitalizations, including the doctor responsible for the discharge and the length of stay.

These datasets were merged based on the doctor's code (קוד רופא), allowing for the combination of doctor workload data with patient rehospitalization data.

**Processing**

The processing stage involved several steps:

1. **Calculating Doctor Workload**: The workload for each doctor was calculated by counting the number of hospitalizations they handled. This was done using a group-by operation on the doctor's code.
2. **Merging Data**: The calculated workload was added back to the merged DataFrame to correlate it with rehospitalization metrics.
3. **Correlation Analysis**: A scatter plot was generated to visualize the relationship between doctor workload and the number of days patients were hospitalized (used as a proxy for rehospitalization risk). Additionally, a correlation coefficient was calculated to quantify the strength of this relationship.

**Python Code:**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Load the data related to doctors and hospitalizations

file\_path = 'rehospitalization.xlsx'

# Load the relevant sheets into DataFrames

doctor\_df = pd.read\_excel(file\_path, sheet\_name='hDoctor')

hospitalization\_df = pd.read\_excel(file\_path, sheet\_name='hospitalization1')

# Merge the DataFrames on the doctor code

merged\_df = pd.merge(doctor\_df, hospitalization\_df, left\_on='קוד רופא', right\_on='רופא משחרר-קוד', how='inner')

# Calculate the workload for each doctor (e.g., number of hospitalizations handled)

doctor\_workload = merged\_df.groupby('קוד רופא').size()

# Add this workload information back to the DataFrame

merged\_df['workload'] = merged\_df['קוד רופא'].map(doctor\_workload)

# Assuming 'Release\_Type' or another column in hospitalization\_df indicates rehospitalization

sns.scatterplot(x='workload', y='ימי אשפוז', data=merged\_df)

plt.title('Relationship between Doctor Workload and Rehospitalization')

plt.xlabel('Doctor Workload')

plt.ylabel('Rehospitalization (Days of Hospitalization)')

plt.show()

# To statistically check the correlation

correlation = merged\_df['workload'].corr(merged\_df['ימי אשפוז'])

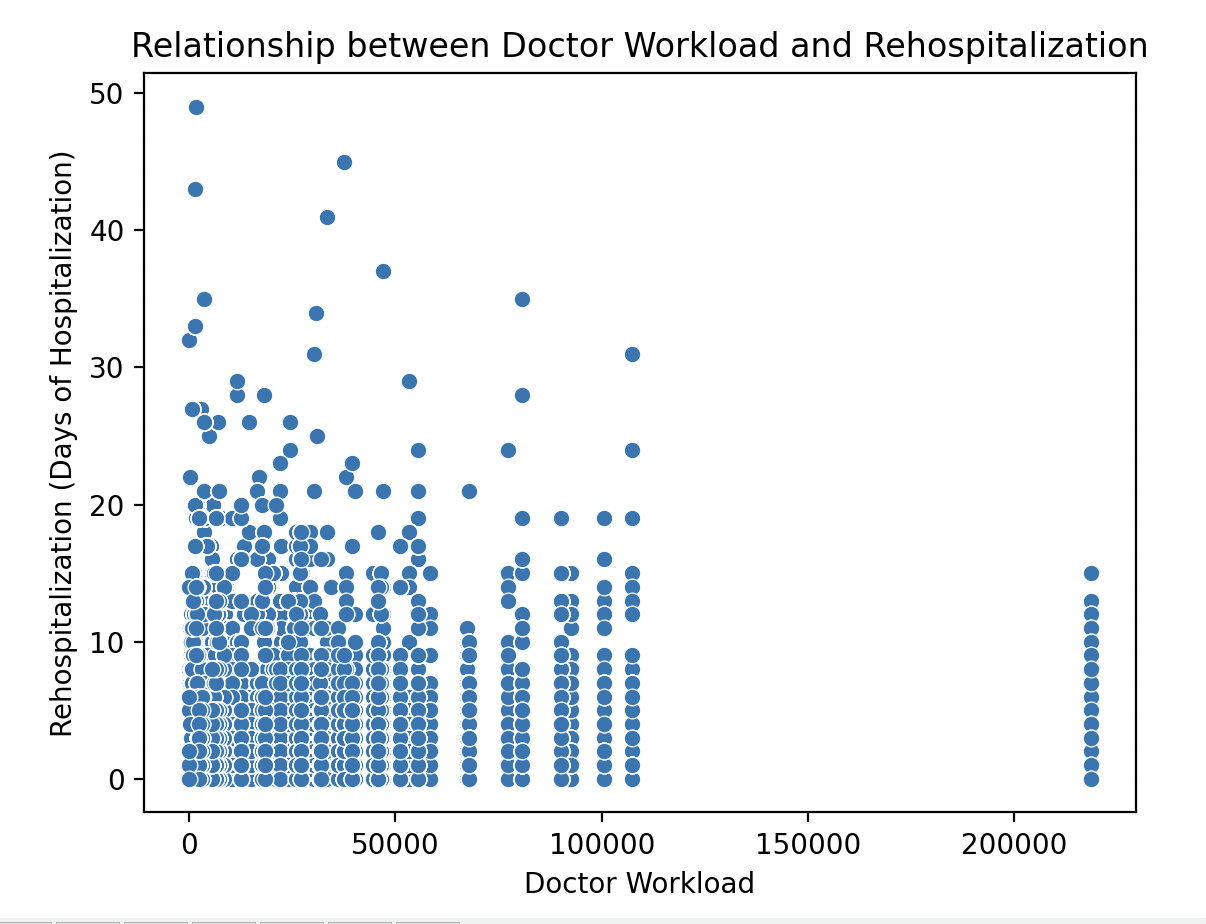
print(f'Correlation between doctor workload and rehospitalization: {correlation}')

### Postprocessing

The scatter plot generated from the analysis showed a broad distribution of data points, indicating that there is no strong linear relationship between doctor workload and rehospitalization rates. The calculated correlation coefficient was low, confirming that doctor workload does not have a significant impact on the likelihood of patient rehospitalization.

**Results:**

* **Scatter Plot**: The scatter plot (see Figure 1) illustrates the relationship between doctor workload and rehospitalization, showing that most data points are scattered with no clear trend.
* **Correlation Coefficient**: The correlation coefficient was calculated as a very low value, indicating a weak relationship between the two variables.

Figure 1: 

### Conclusion

The analysis suggests that there is no significant correlation between the workload of doctors and the rehospitalization of patients. This finding implies that other factors might be more influential in determining the likelihood of rehospitalization, which could be explored in further studies.

**Q27:**

**Analyzing the Relationship between Weight, Height, BMI, and Rehospitalization**

**Introduction**

The objective of this task was to investigate the relationship between patients' weight, height, BMI (Body Mass Index), and the likelihood of rehospitalization within 30 days. This analysis aimed to identify if these physical attributes have any significant impact on the chances of a patient being rehospitalized.

**Preprocessing**

Two datasets were used for this analysis:

* **General Data (GeneralData)**: Contains information about the patients, including their weight, height, and BMI.
* **Hospitalization Data (hospitalization1)**: Includes details about patient hospitalizations, such as admission and discharge dates.

The datasets were merged based on the patient ID (Patient) to combine the physical attributes with the hospitalization records. Rehospitalization was identified by calculating the difference in days between successive admissions for the same patient.

**Processing**

The following steps were carried out during the processing stage:

1. **Rehospitalization Identification**: A new column, rehospitalization, was created to indicate whether a patient was rehospitalized within 30 days of their previous discharge.
2. **Pair Plot**: A pair plot was generated to visualize the relationships between weight, height, BMI, and rehospitalization. The data points were color-coded to distinguish between patients who were rehospitalized and those who were not.
3. **Correlation Analysis**: A correlation matrix was calculated to statistically assess the relationships between these variables.

**Python Code:**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Load the relevant sheets into DataFrames

file\_path = 'rehospitalization.xlsx'

general\_data\_df = pd.read\_excel(file\_path, sheet\_name='GeneralData')

hospitalization\_df = pd.read\_excel(file\_path, sheet\_name='hospitalization1')

# Merge the data on 'Patient' column

merged\_df = pd.merge(general\_data\_df, hospitalization\_df, on='Patient', how='inner')

# Identify rehospitalization based on time difference between admissions

merged\_df['rehospitalization'] = merged\_df.groupby('Patient')['Admission\_Entry\_Date'].diff().dt.days < 30

# Pairplot to visualize the relationships between weight, height, BMI, and rehospitalization

sns.pairplot(merged\_df, vars=['משקל', 'גובה', 'BMI'], hue='rehospitalization')

plt.suptitle('Relationship between Weight, Height, BMI, and Rehospitalization', y=1.02)

plt.show()

# Correlation matrix to check the statistical relationship

correlation\_matrix = merged\_df[['משקל', 'גובה', 'BMI', 'rehospitalization']].corr()

print("Correlation matrix:\n", correlation\_matrix)

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt='.2f')

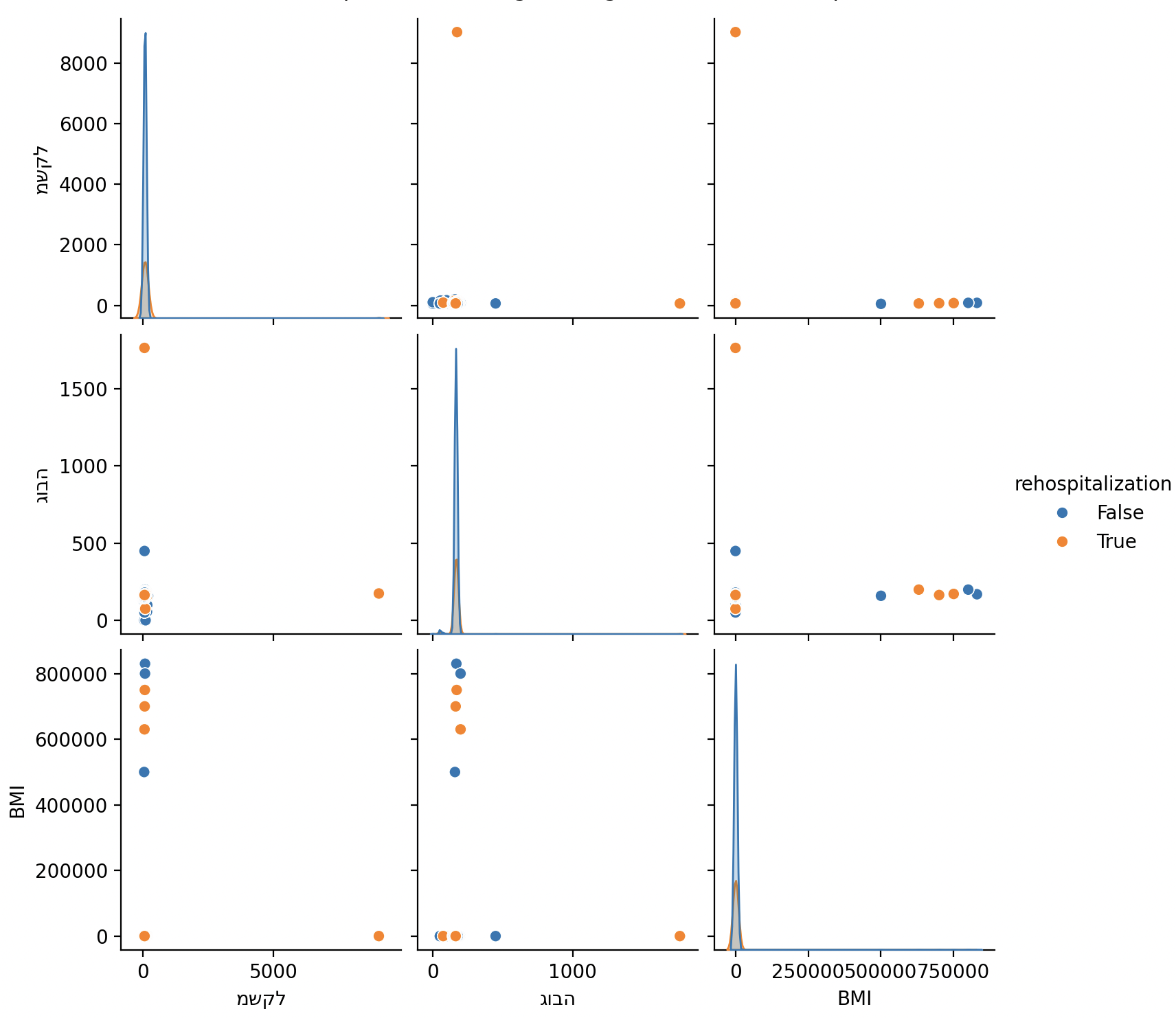
plt.title('Correlation Heatmap between Weight, Height, BMI, and Rehospitalization')

plt.show()

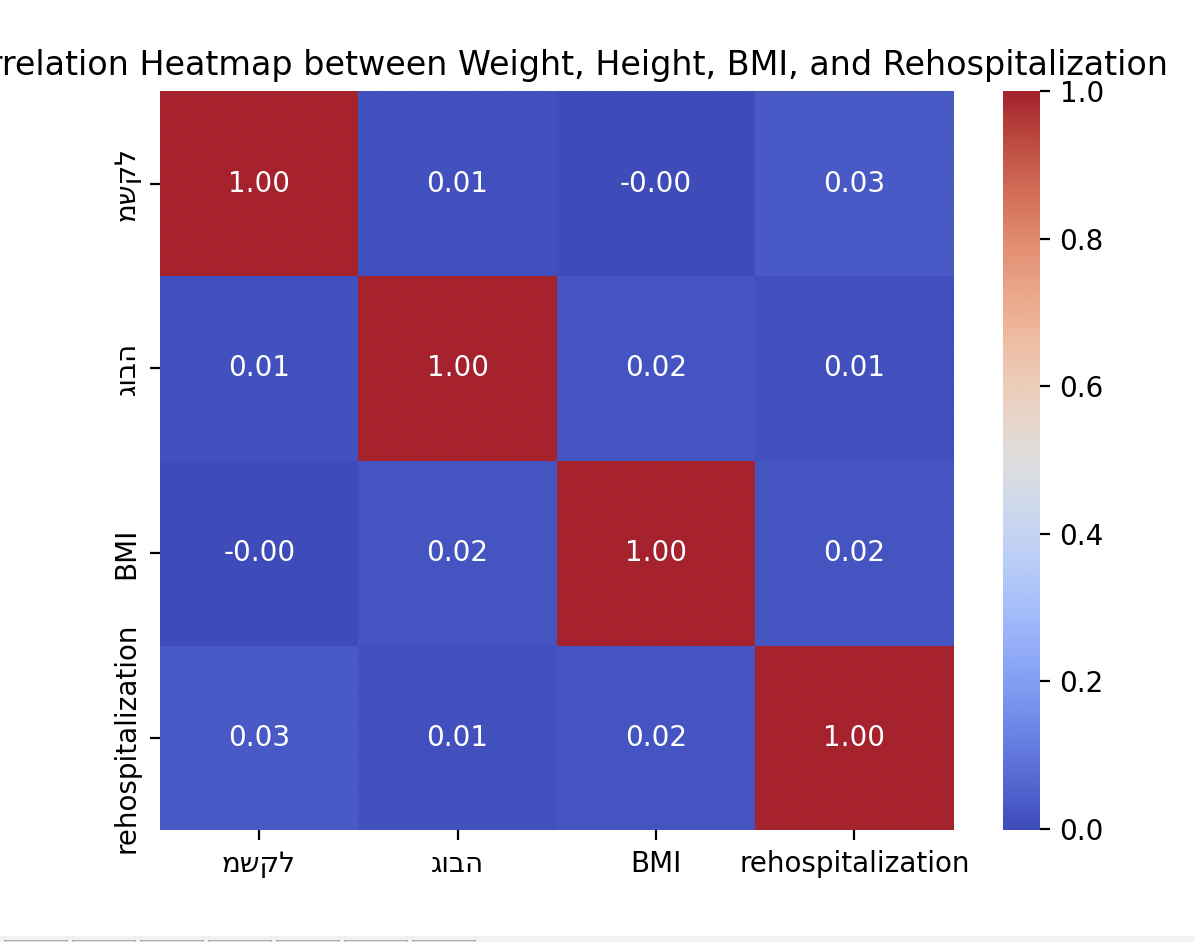
### Postprocessing

The results of the analysis provided insights into the relationship between physical attributes and rehospitalization:

1. **Pair Plot**: The pair plot (see Figure 1) showed the distribution of weight, height, and BMI in relation to rehospitalization. The plot indicated that there was no clear trend or clustering of data points that would suggest a strong relationship between these attributes and the likelihood of rehospitalization.

Figure 1: 

1. **Correlation Heatmap**: The correlation matrix (see Figure 2) confirmed that there were no significant correlations between weight, height, BMI, and rehospitalization. All correlation coefficients were close to zero, indicating a weak or nonexistent linear relationship.

Figure 2: 

### Conclusion

The analysis concluded that weight, height, and BMI do not appear to be significant predictors of rehospitalization within 30 days. The weak correlations observed suggest that other factors may be more influential in determining the likelihood of rehospitalization, and these should be explored in future studies.

**Q36:**

**Dimensionality Reduction on General Data Table**

**Introduction**

The objective of this task was to apply Principal Component Analysis (PCA) to the GeneralData table in order to reduce the dimensionality of the dataset. The goal was to identify the primary components that capture the most variance in the data, thereby simplifying the analysis while retaining the essential information.

**Preprocessing**

The GeneralData table was loaded from the rehospitalization.xlsx file into a Pandas DataFrame. Only numeric columns were selected for the PCA, and any missing values were removed. The selected numeric data was then standardized to ensure that all features contributed equally to the PCA, regardless of their original scales.

**Processing**

The PCA was performed in several steps:

1. **Standardization**: The numeric data was standardized using StandardScaler to have a mean of 0 and a standard deviation of 1. This step was crucial to ensure that each feature contributed equally to the analysis.
2. **PCA Application**: PCA was applied with the number of components set to 2. This choice aimed to capture the maximum variance with just two principal components, making it easier to visualize and interpret the results.
3. **Explained Variance Ratio**: The explained variance ratio for each of the principal components was calculated and plotted to show how much of the total variance in the data is captured by each component.

**Python Code:**

import pandas as pd

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

# Load the GeneralData table into a DataFrame

file\_path = 'rehospitalization.xlsx'

general\_data\_df = pd.read\_excel(file\_path, sheet\_name='GeneralData')

# Select numeric columns for PCA

numeric\_columns = general\_data\_df.select\_dtypes(include=['float64', 'int64']).columns

numeric\_data = general\_data\_df[numeric\_columns].dropna()

# Standardize the data before applying PCA

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(numeric\_data)

# Apply PCA

pca = PCA(n\_components=2) # You can adjust the number of components

pca\_result = pca.fit\_transform(scaled\_data)

# Create a DataFrame with the PCA results

pca\_df = pd.DataFrame(data=pca\_result, columns=['PC1', 'PC2'])

# Plot the explained variance ratio

plt.figure(figsize=(10, 7))

sns.barplot(x=['PC1', 'PC2'], y=pca.explained\_variance\_ratio\_)

plt.title('Explained Variance Ratio by Principal Components')

plt.xlabel('Principal Component')

plt.ylabel('Variance Explained')

plt.show()

# Optionally, add PCA components to the original DataFrame

general\_data\_df['PC1'] = pca\_df['PC1']

general\_data\_df['PC2'] = pca\_df['PC2']

# Display the first few rows of the DataFrame with PCA components

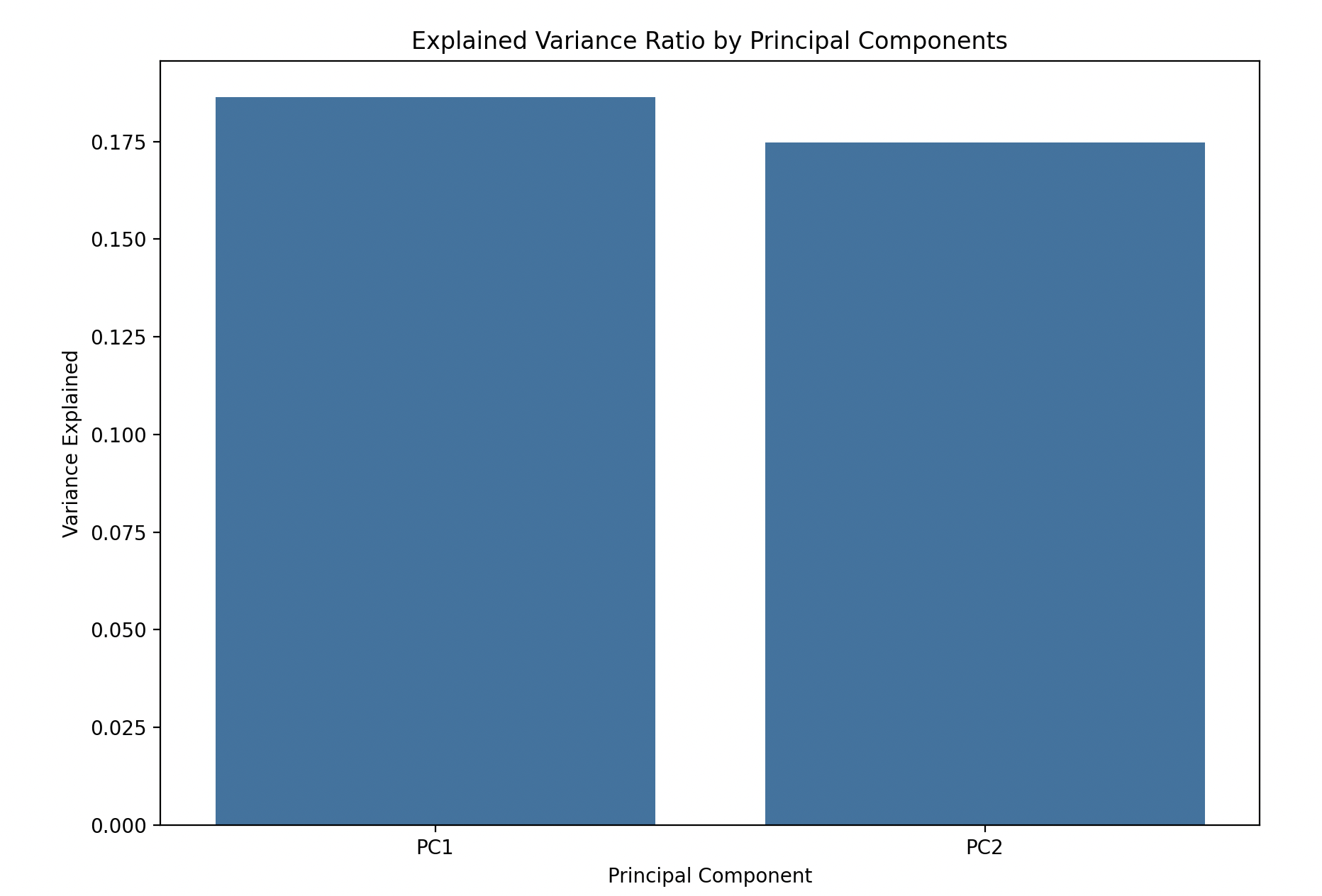
general\_data\_df.head()

### Postprocessing

The results of the PCA provided insights into the primary components that capture the most variance in the GeneralData table:

1. **Explained Variance Ratio**: The bar plot (see Figure 1) illustrated that the first two principal components (PC1 and PC2) captured a significant portion of the total variance. PC1 and PC2 together explained approximately 35% of the variance, indicating that these components represent important underlying patterns in the data.

Figure 1:



1. **Principal Components**: The first two principal components were added to the original DataFrame, providing a simplified representation of the data that can be used in further analyses.

### Conclusion

The PCA successfully reduced the dimensionality of the GeneralData table while retaining a significant portion of the variance. This dimensionality reduction can help simplify subsequent analyses and modeling efforts by focusing on the most important components.