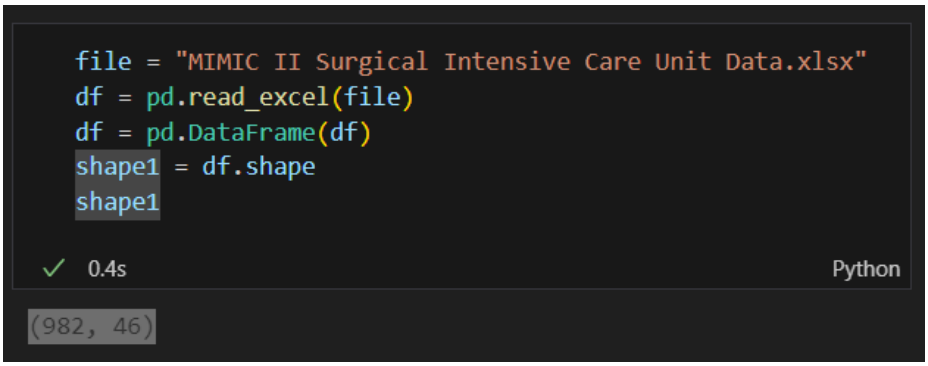
Prediction of death in ICU on MIMIC II Surgical Intensive Care Unit Dataset

Question 1. Load your data into Jupyter Notebook.

To load dataset into Jupyter Notebook, used pandas library. The dataset size is (982, 46) 44 columns are numeric, 2 columns are object.



Question 2. Create pairwise scatterplots of variables of interest. Describe your discoveries and the relationships, if any

The goal of this analysis is to explore relationships between selected health-related variables in our dataset. The columns of interest include 'weight', 'bmi', 'age', 'heart_rate', 'platelet', and 'white_blood_cells'.

1. Column Selection: We started by selecting specific columns from our dataset that we found interesting and relevant for our exploration.
2. Correlation Analysis: To understand how these health-related variables are related, we calculated the correlation matrix using the corr() function. This matrix provides us with pairwise correlation coefficients between all selected columns.
3. Pairwise Scatterplots: Based on the correlation matrix, we identified pairs of columns with notable correlation coefficients. We then created scatter plots for these pairs to visually inspect their relationships.

Findings:

1. 'Weight' and 'BMI':

Correlation Coefficient: 0.33

Interpretation: We observed a moderate positive correlation. When 'weight' increases, 'bmi' tends to increase as well.

2. 'Age' and 'Heart Rate':

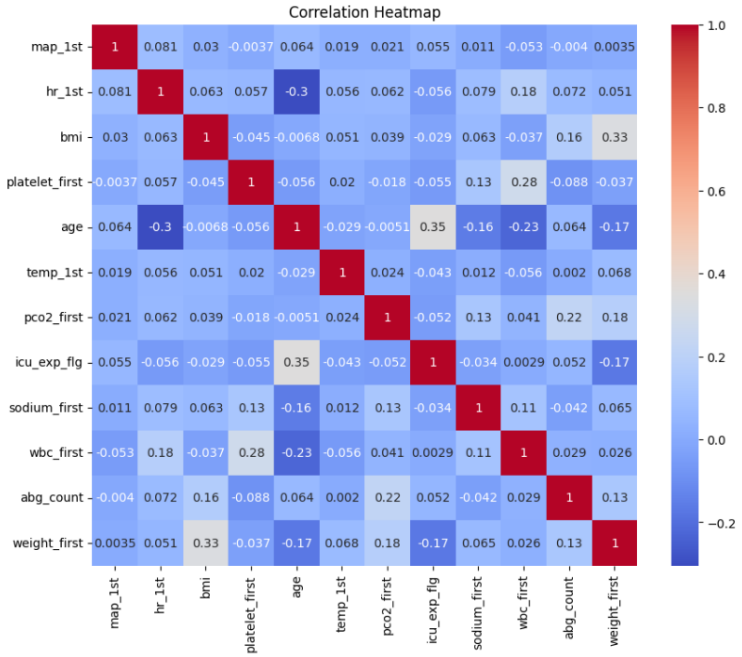
Correlation Coefficient: -0.3

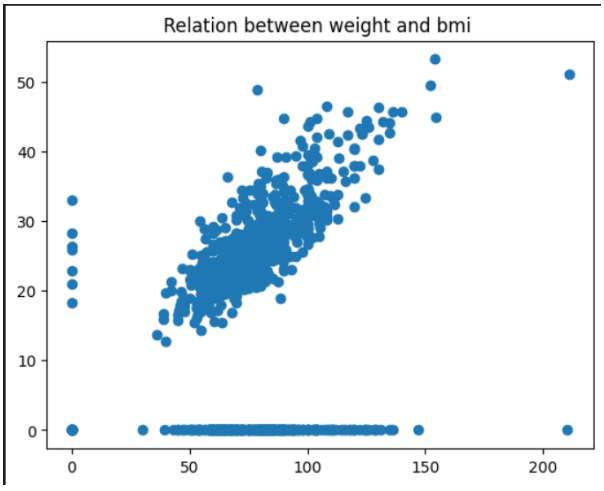
Interpretation: We found a moderate negative correlation. On average, as 'age' increases, 'heart_rate' tends to decrease.

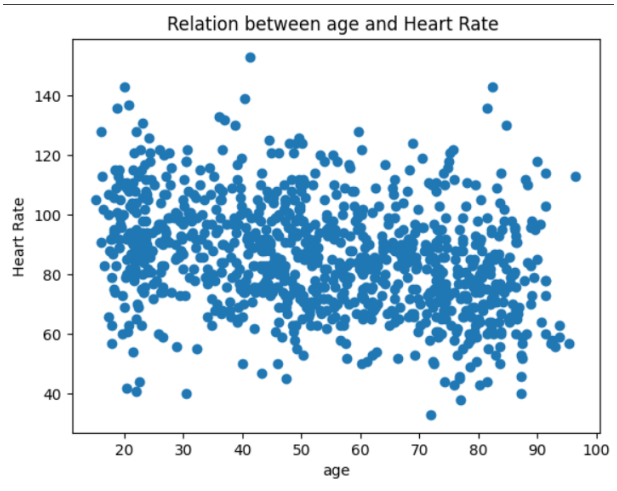
3. 'Platelet' and 'White Blood Cells':

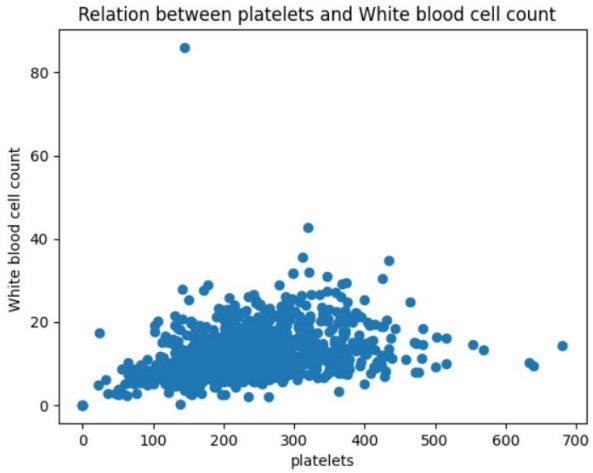
Correlation Coefficient: 0.28

Interpretation: We identified a weak positive correlation. There is a positive relationship, but it's not very strong.









Question 3. Create box plots to identify outliers.

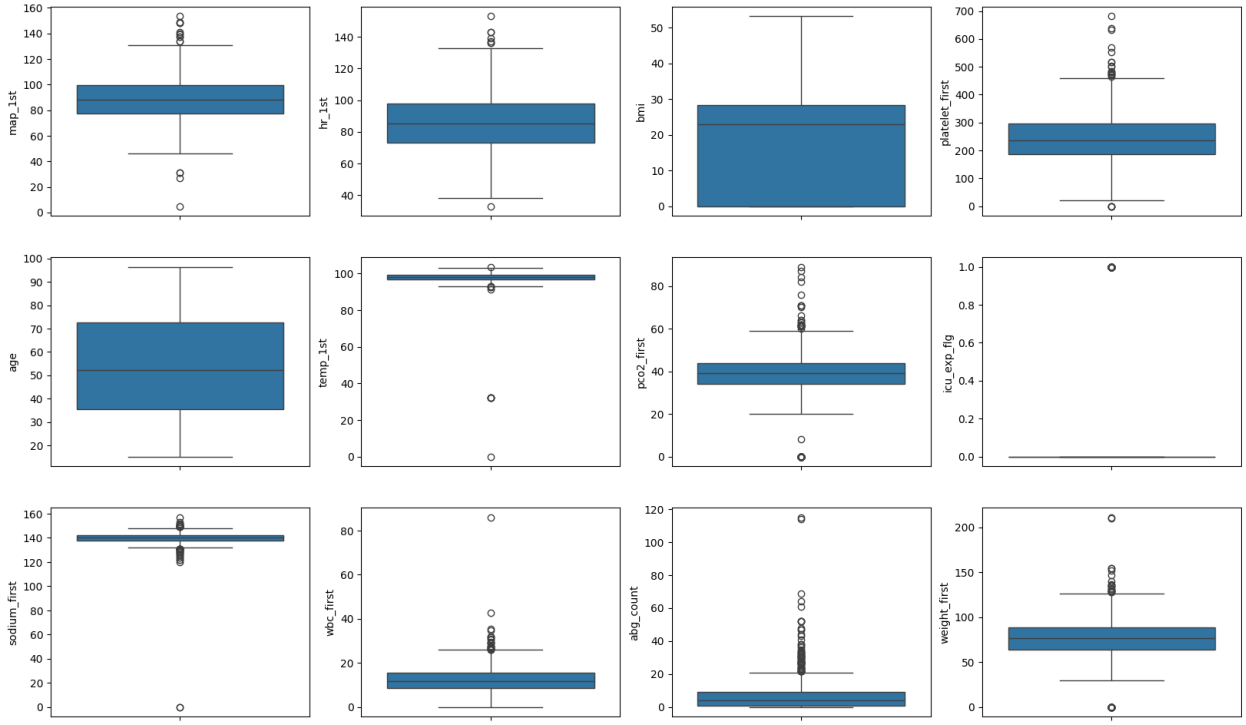
To explore and identify potential outliers in key health-related variables from our dataset. The variables of interest include 'map_1st', 'hr_1st', 'bmi', 'platelet_first', 'age', 'temp_1st', 'pco2_first', 'icu_exp_flg', 'sodium_first', 'wbc_first', 'abg_count', and 'weight_first'.

1. Boxplot Creation:

We began by creating boxplots for the selected health-related variables to visualize their distributions and identify any potential outliers.

2. Outlier Identification:

Our focus was on the variables 'temp_1st', 'sodium_first', 'wbc_first', and 'abg_count'. Through careful examination of the boxplots, we looked for data points that fall outside the typical range, which could indicate outliers.



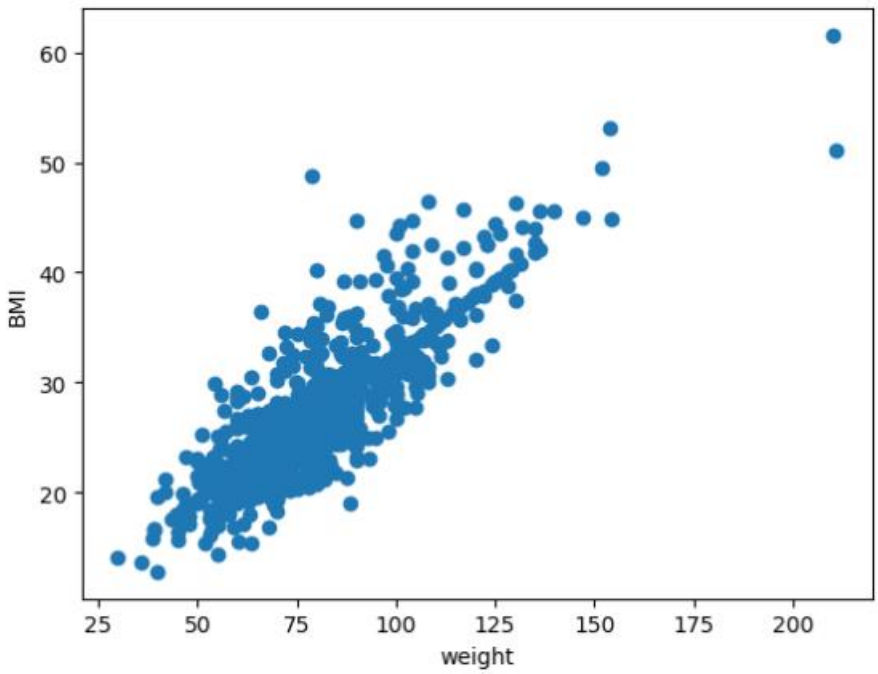
Question 4. Cleaning data and handling extreme values.

1. No null values were found, indicating no missing data in the dataset.
2. Rows with 'weight_first' values equal to zero were removed from the dataset, which were 50 values.
3. Rows with 'bmi' values equal (305 values) to zero were separated into a new DataFrame, 'dfbmi0'.
4. A linear regression model was created and trained using the weight data.
5. The original dataset was updated with the predicted BMI values for rows initially having 'bmi' equal to zero.

```
(df['bmi'] == 0).sum(), (df['weight_first'] == 0).sum()
```

✓ 0.0s Python

```
(305, 50)
```



After cleaned BMI and weight

- Removed rows with BMI values exceeding 56 and weight values over 150.
- This helps ensure realistic and accurate information related to body mass.
- Filtered out rows with initial temperature below 80 and sodium levels under 100.
- Focused on addressing unusually low temperatures and potential errors in sodium readings.
- Eliminated rows with extreme white blood cell counts (above 50) and arterial blood gas counts (above 80).

```
df = df[df['bmi'] < 56]
df = df[df['weight_first'] < 150]
df = df[df['temp_1st'] > 80]
df = df[df['sodium_first'] > 100]
df = df[df['wbc_first'] < 50]
df = df[df['abg_count'] < 80]
```

✓ 0.0s

```
shape2 = df.shape
removedValue = shape1[0]-shape2[0]
removedValue
```

✓ 0.0s

64

After cleaning dataset, I dropped 64 rows.

Question 5. Calculate the median, mean and standard deviation of variables of interest.

- I used the describe() function to calculate the mean, standard deviation, maximum, and minimum values for the dataset.

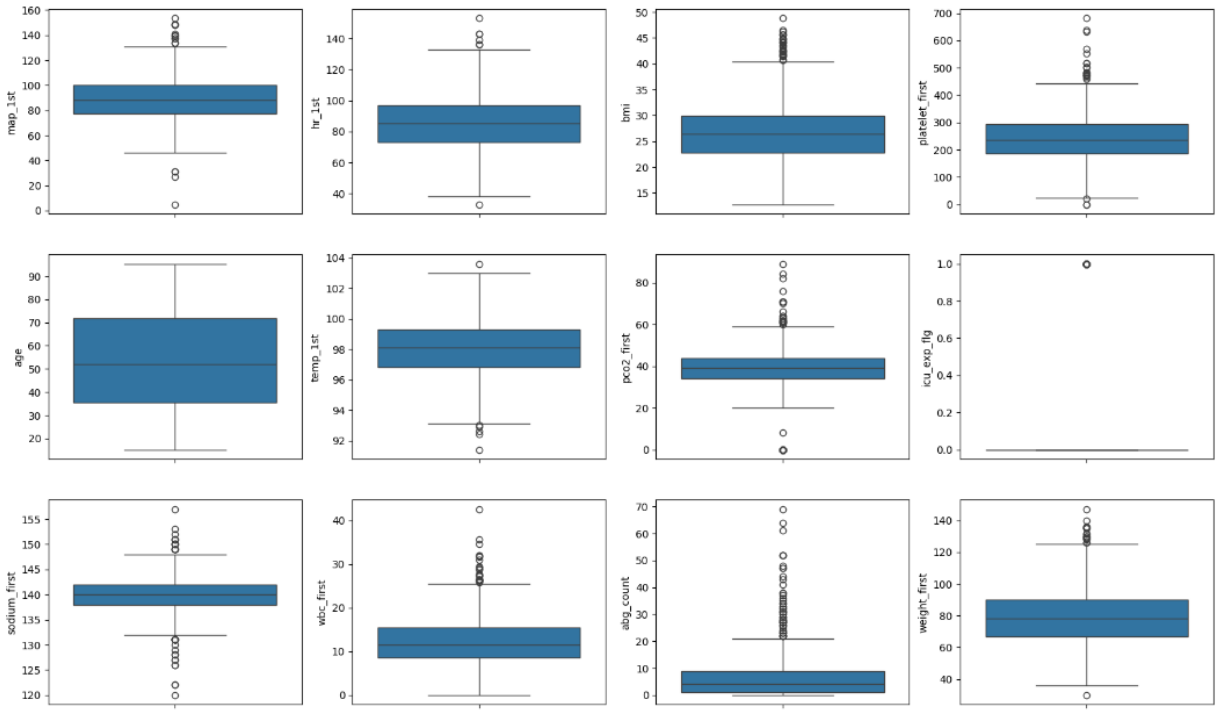
	aline_flg	icu_los_day	hospital_los_day	age	gender_num	weight_first	bmi	sapsi_first	sofa_first	service_num	...
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.0	...
mean	0.714597	3.535806	9.033769	52.745667	0.626362	79.183802	27.045541	13.759259	5.864924	1.0	...
std	0.451852	3.569124	8.759725	21.253245	0.484033	18.432930	5.875512	4.792738	2.153147	0.0	...
min	0.000000	0.500000	1.000000	15.190460	0.000000	30.000000	12.784877	0.000000	0.000000	1.0	...
25%	0.000000	1.460000	4.000000	35.429160	0.000000	66.650000	22.836542	11.000000	4.000000	1.0	...
50%	1.000000	2.350000	7.000000	51.916005	1.000000	78.000000	26.381773	14.000000	6.000000	1.0	...
75%	1.000000	4.115000	11.000000	71.691775	1.000000	90.000000	29.921373	17.000000	7.000000	1.0	...
max	1.000000	28.240000	112.000000	95.311070	1.000000	147.000000	48.785651	32.000000	16.000000	1.0	...

- I used median() function to calculate median.

```
aline_flg              1.000000
icu_los_day            2.350000
hospital_los_day       7.000000
age                   51.916005
gender_num             1.000000
weight_first          78.000000
bmi                   26.381773
sapsi_first           14.000000
sofa_first             6.000000
service_num            1.000000
day_icu_intime_num     4.000000
hour_icu_intime        8.000000
hosp_exp_flg           0.000000
icu_exp_flg            0.000000
day_28_flg             0.000000
mort_day_censored    731.000000
censor_flg             1.000000
sepsis_flg             0.000000
chf_flg                0.000000
afib_flg               0.000000
renal_flg              0.000000
liver_flg              0.000000
copd_flg               0.000000
cad_flg                0.000000
stroke_flg             0.000000
...
```

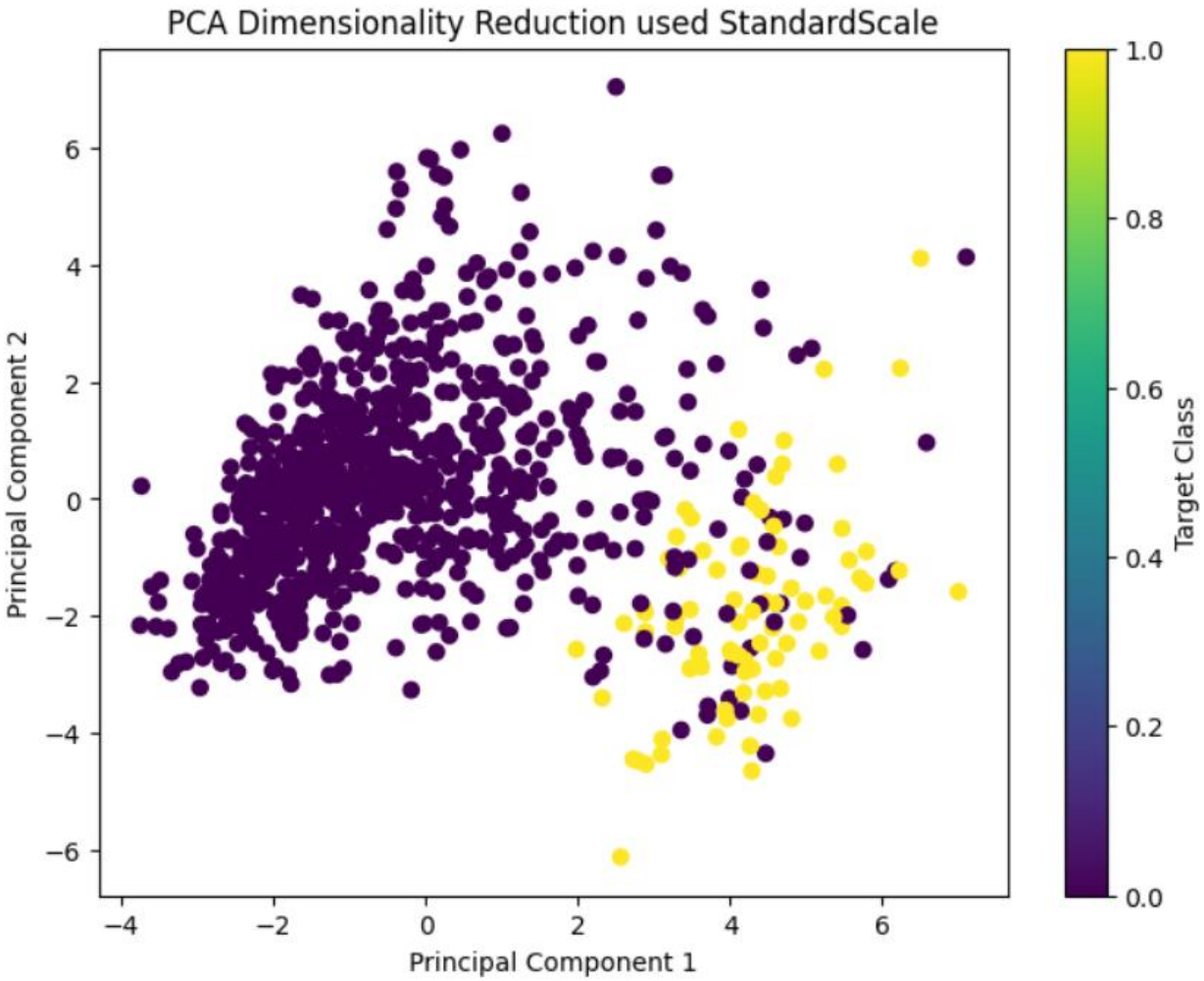
Question 6. Create new box plots after data imputation and analyse the results.

Created boxplots on interested columns after cleaning dataset. And showed in below.



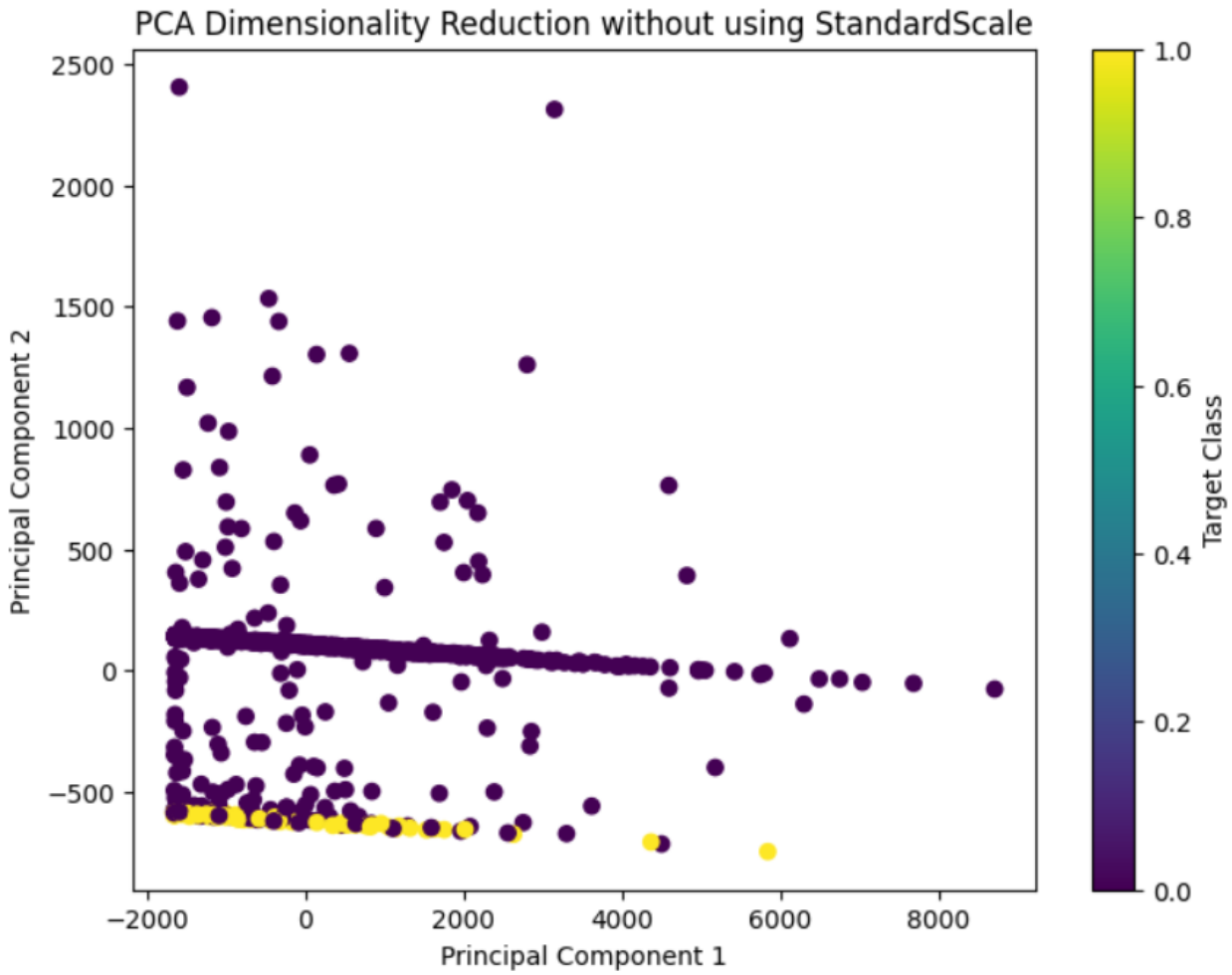
Question 7. Apply PCA and visualize Principal Component 1 and Principal Component 2.

Visualized the dataset's reduced-dimensional representation using PCA, both with and without standard scaling, and to observe how the target class ('icu_exp_flg') influences the distribution.



PCA with Standard Scaling

- Standard scaling was applied to the feature variables using StandardScaler to ensure all features have the same scale.
- PCA with 2 components was then applied to the scaled data.
- A scatter plot was generated to visualize the reduced-dimensional data, with colors representing the target class.
- The use of standard scaling ensures that all features contribute equally to the PCA, preventing biases due to different scales.



PCA without Standard Scaling:

- PCA with 2 components was applied directly to the original unscaled data.
- Another scatter plot was created to visualize the reduced-dimensional data without standard scaling, with colors indicating the target class.

Question 8. Create Logistic Regression model.

- The dataset was split into feature variables (X) and the target variable ('icu_exp_flg').
- The data was divided into training and testing sets using a test size of 20%, ensuring the model's evaluation on unseen data.
- Standard scaling was applied to the feature variables to ensure consistent scales.
- A logistic regression model was created and trained on the scaled training data.
- The trained model was used to make predictions on the scaled test set.
- Evaluation metrics, including accuracy, confusion matrix, and classification report, were calculated to assess the model's performance.

Results: Accuracy: 96%

The accuracy score indicates that the model correctly predicted the target variable in 96% of cases.

Confusion Matrix:

True Positive (TP): 161

True Negative (TN): 15

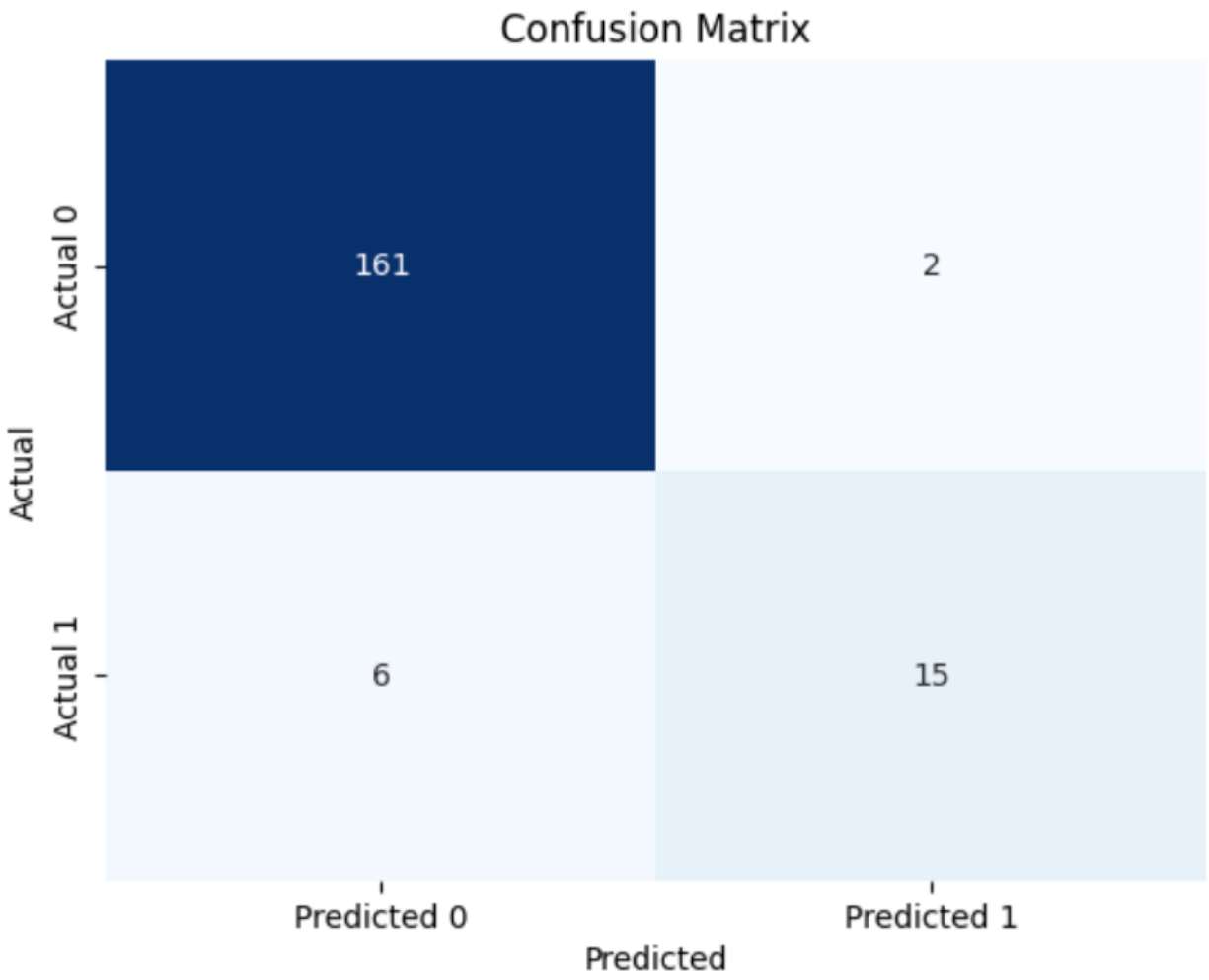
False Positive (FP): 2

False Negative (FN): 6

```
Accuracy: 0.96
Confusion Matrix:
[[161   2]
 [  6  15]]
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       163
           1       0.88      0.71      0.79        21

    accuracy                           0.96       184
   macro avg       0.92      0.85      0.88       184
weighted avg       0.95      0.96      0.95       184
```



The logistic regression model, trained and evaluated on the scaled dataset, demonstrates strong predictive capabilities for the 'icu_exp_flg' target variable.