**EC 9560 – DATA MINING**

**LAB 02**

**DARMILA.T**

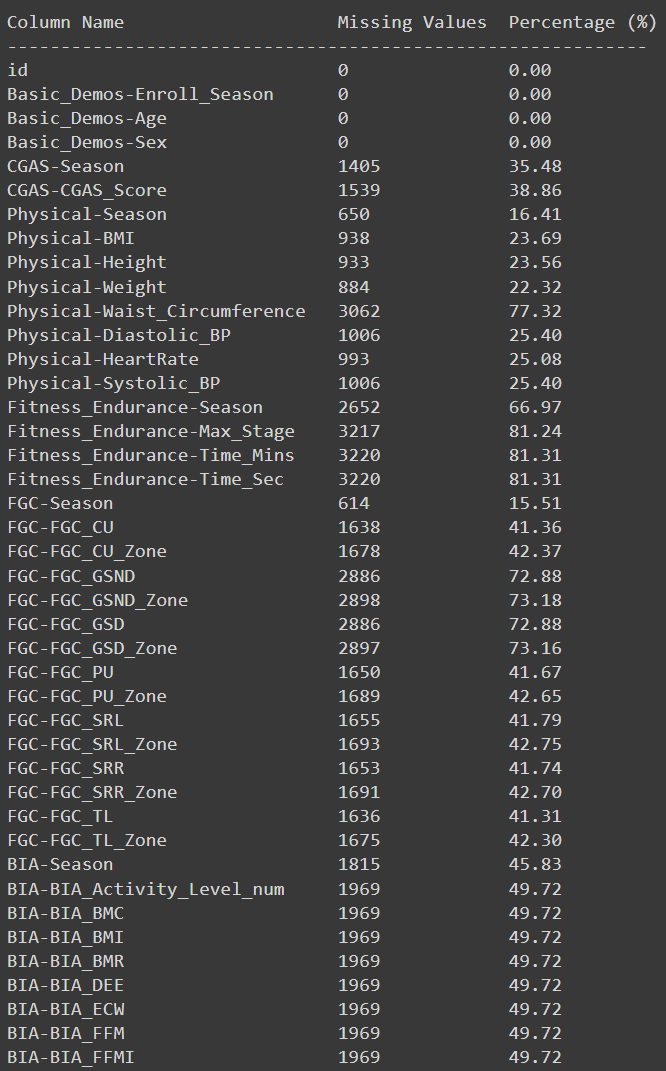
**2020/E/027**

**SEMESTER 07**

**10th OCTOBER 2024**

**A comprehensive study on your data including data visualization, distribution analysis, correlation analysis.**

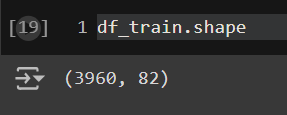
A screenshot of a computer

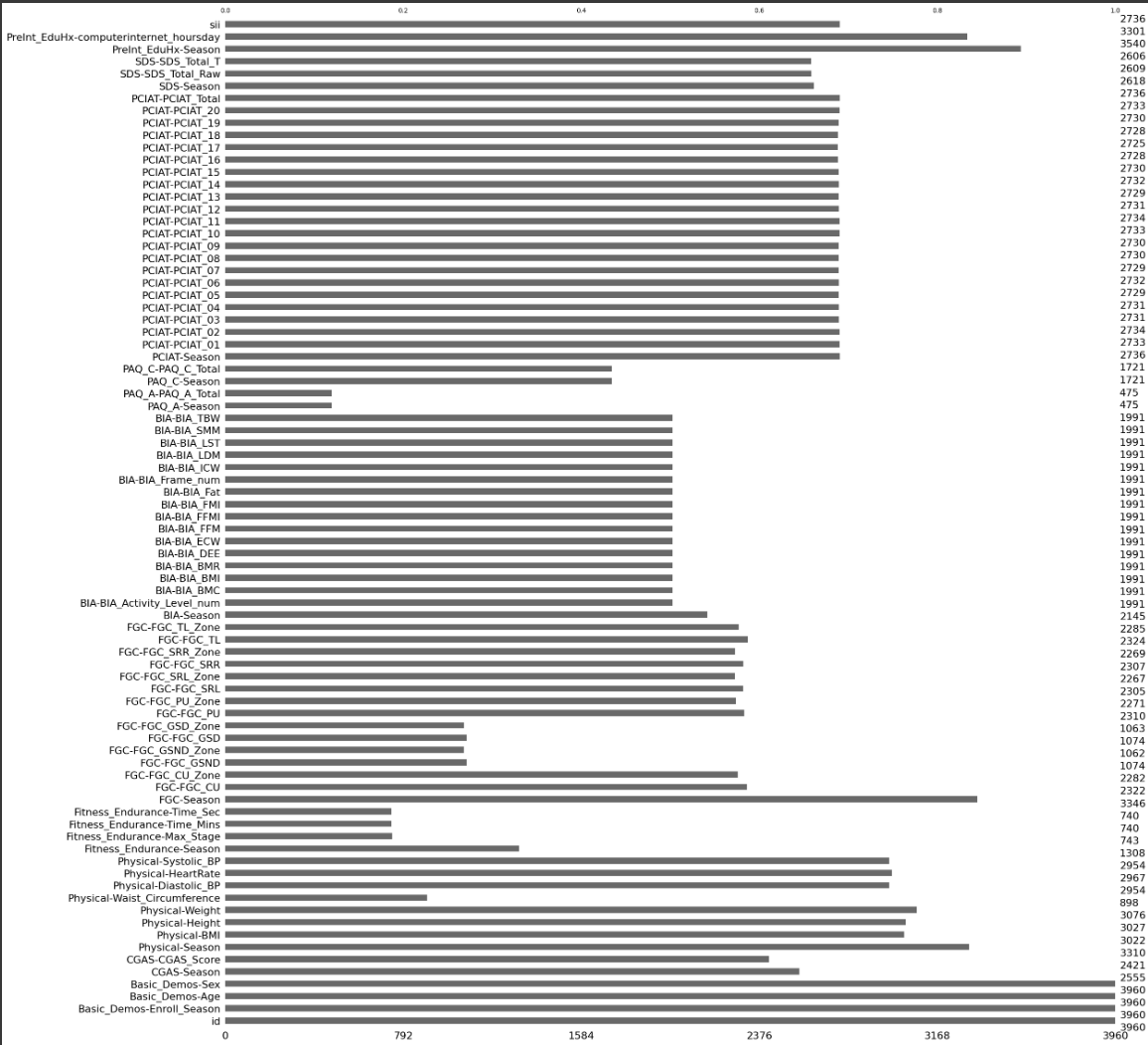
Description automatically generatedThe dataset have total 81 features. So first of all, I find the missing values count in the dataset.

A computer screen with colorful text

Description automatically generated

Here, we can see most of the columns have much more null values because total there only 3960 samples.



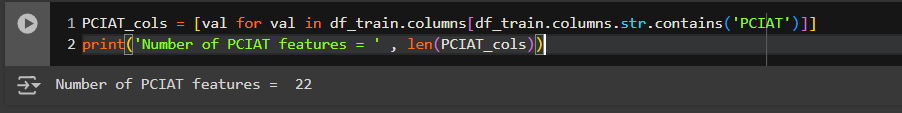


Total values of every features

A screen shot of a computer

Description automatically generatedNumber of duplicates in the dataset

PCIAT Features:



* As mentioned, there are 22 PCIAT features. These comprise answers to 20 questions (each marked out of 5), the total score and 'season' when the test was carried out.
* The sii target is derived from the total PCIAT score:
  + 0-30 gives sii = 0
  + 31-49 gives sii = 1
  + 50-79 gives sii = 2
  + 80-100 gives sii = 3.

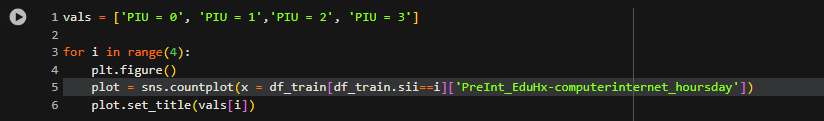
A screenshot of a computer

Description automatically generated

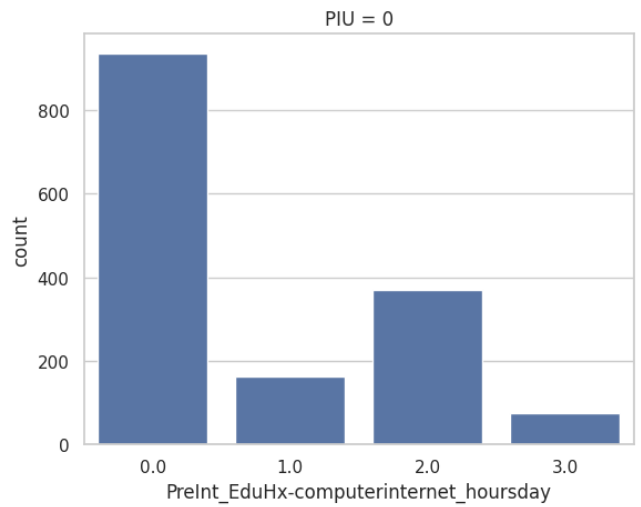
This graph shows the outliers and category the target column sii.

Severity Impairment Index:

Here I plot when the sii = 0,1,2,3 for the PreInt\_EduHx-computerinternet\_hoursday



A graph of a bar chart

Description automatically generated with medium confidence

A graph of a bar graph

Description automatically generated with medium confidence

A graph of a number of blue rectangular bars

Description automatically generated with medium confidence

Correlation matrix



Based on this correlation find the maximum number of correlation pairs:

A screen shot of a computer code

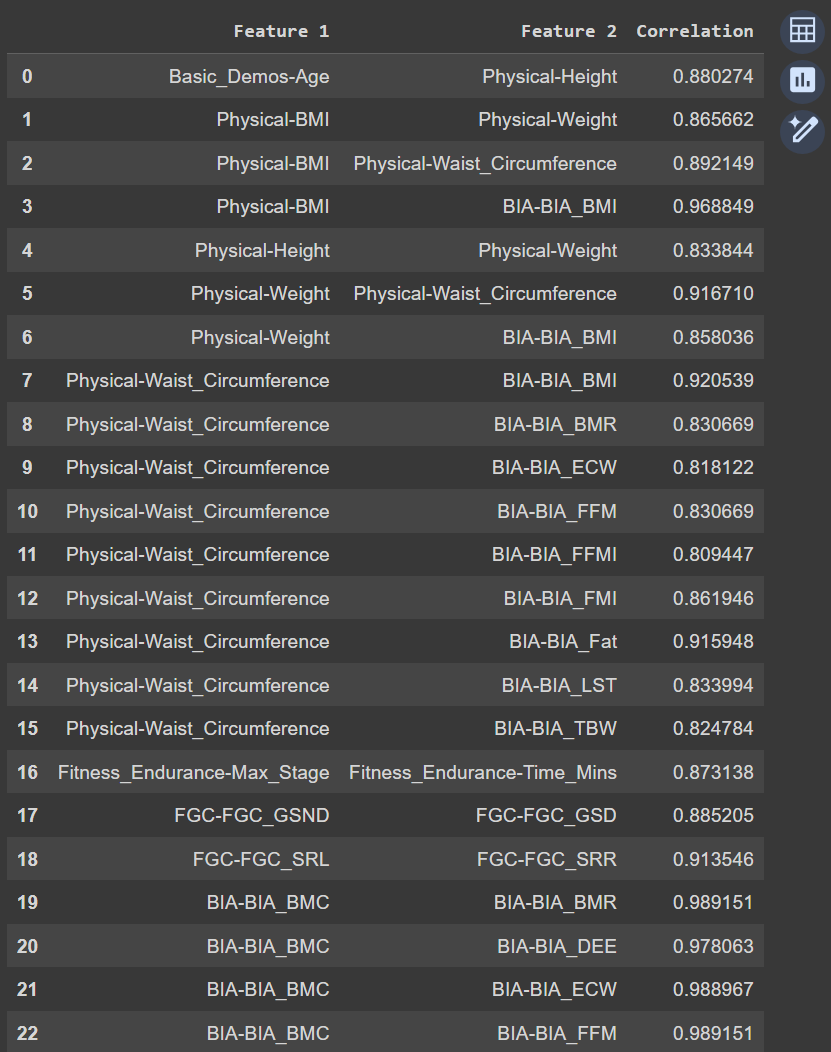
Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Based on this correlation,

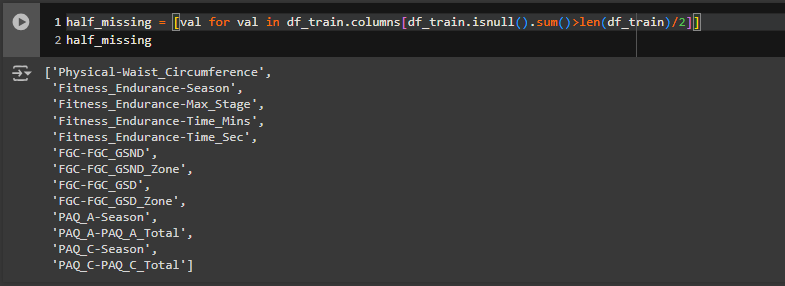
A screen shot of a computer program

Description automatically generated

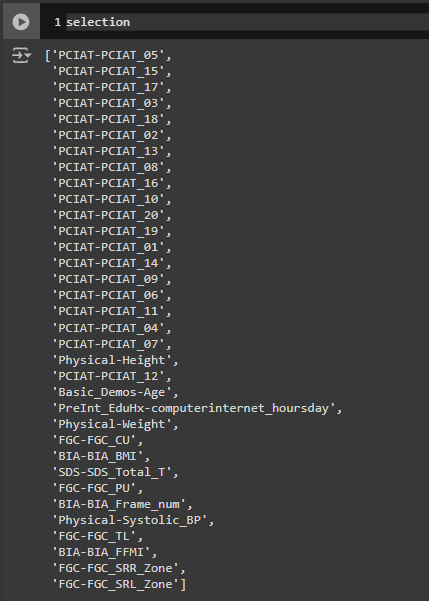
A screenshot of a computer program

Description automatically generated

I select these features based on the correlation with target more than 0.1 or less than 0.1

Find the features that are have the missing values more than the half of the data samples.  


Then I checked is this features are selected above in selection matrix  

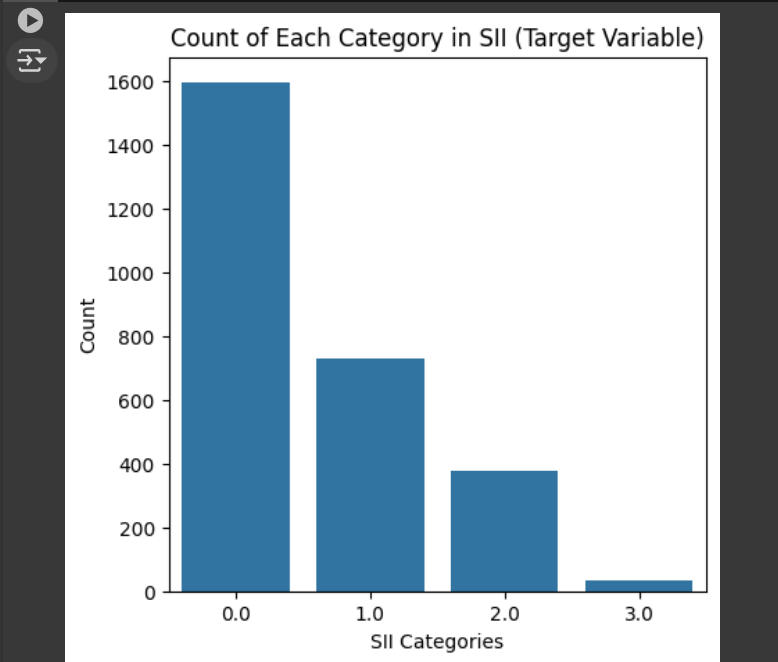



I now have 16 selected features based on a) correlation with the target and b) relatively few missing values.  

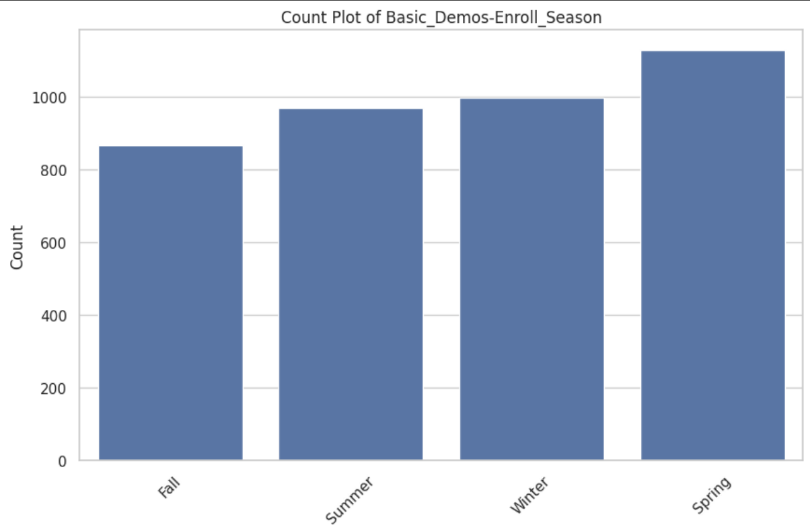

histograms for a selection of numerical features in a DataFrame named df\_train

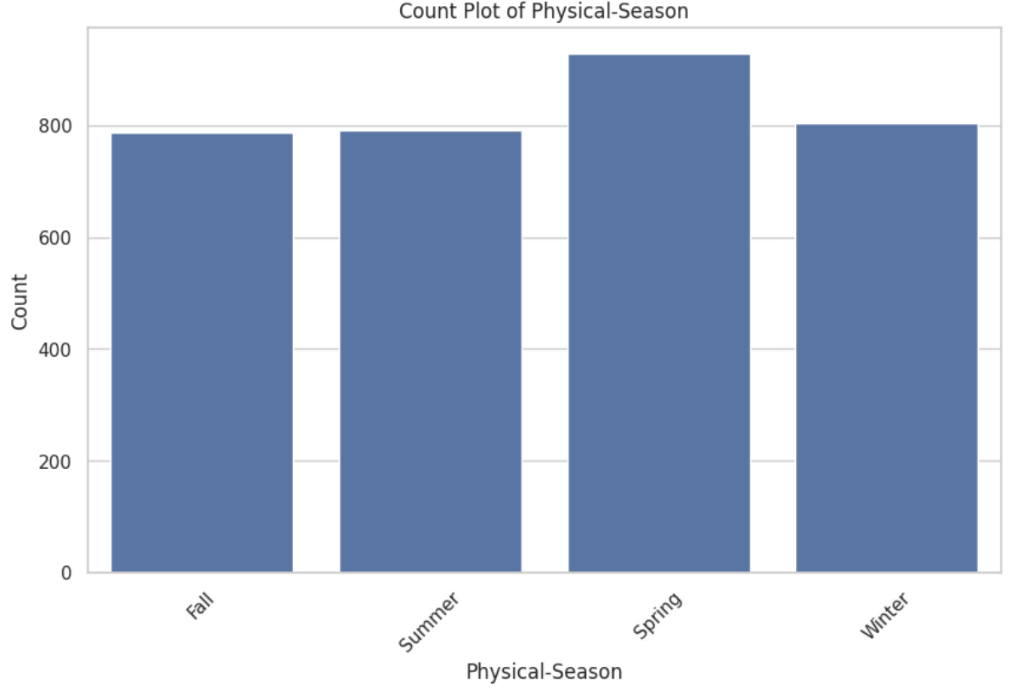
A screenshot of a graph

Description automatically generated

Visualization  
SII – Target  


A graph showing the number of seasons

Description automatically generated with medium confidence



A graph of a box

Description automatically generated

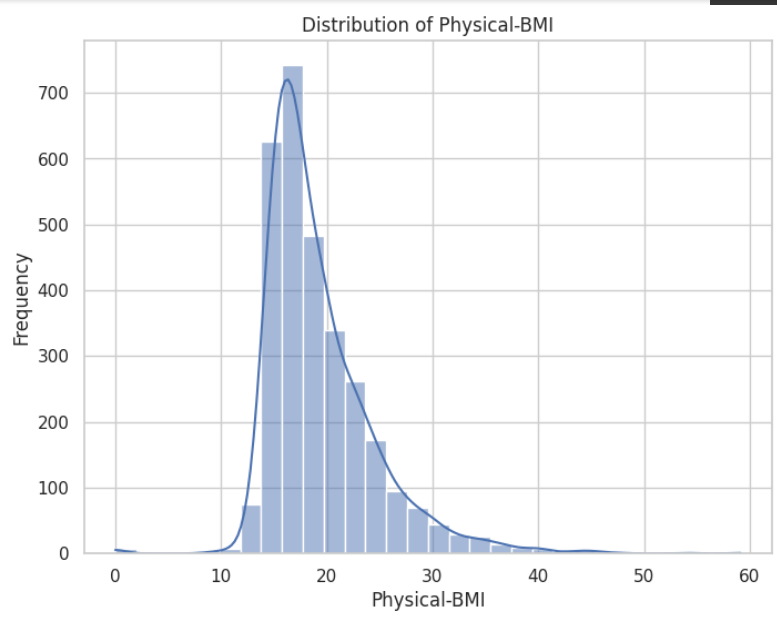
A graph of a distribution of basic events

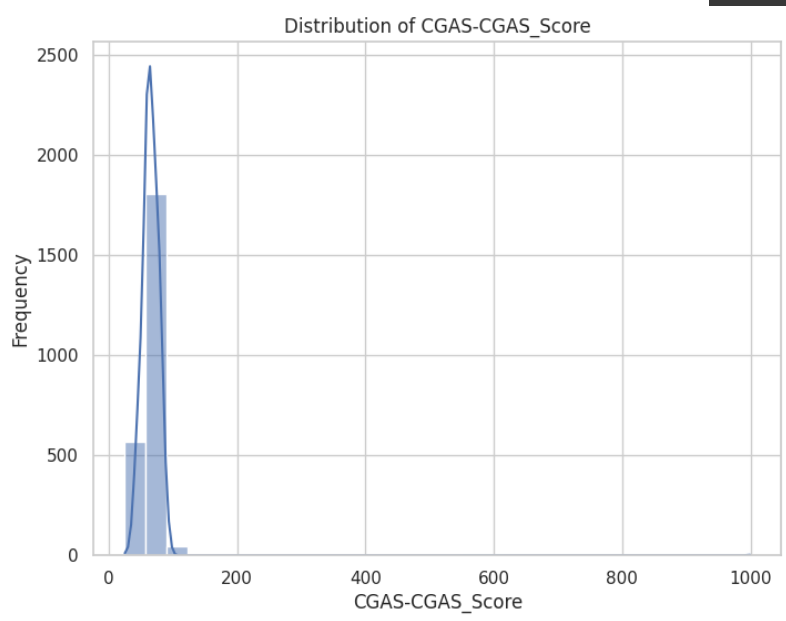
Description automatically generated

A blue rectangular object with white text

Description automatically generatedA graph of a distribution of basic demos sex

Description automatically generated

A graph with a bar graph

Description automatically generated with medium confidence  
A graph of a box plot

Description automatically generated

*# Set the style for seaborn*

sns.set(style="whitegrid")

*# List of categorical features*

categorical\_features = [

    "Basic\_Demos-Enroll\_Season",

    "CGAS-Season",

    "Physical-Season"

]

*# List of numerical features*

numerical\_features = [

    "Basic\_Demos-Age",

    "Basic\_Demos-Sex",

    "CGAS-CGAS\_Score",

    "Physical-BMI"

]

*# Plotting categorical features*

for feature in categorical\_features:

    plt.figure(figsize=(10, 6))

    sns.countplot(data=df\_train, x=feature)

    plt.title(f'Count Plot of {feature}')

    plt.xlabel(feature)

    plt.ylabel('Count')

    plt.xticks(rotation=45)

    plt.show()

*# Plotting numerical features*

for feature in numerical\_features:

    plt.figure(figsize=(8, 6))

*# Histogram*

    sns.histplot(df\_train[feature], bins=30, kde=True)  *# Add kde=True for a kernel density estimate*

    plt.title(f'Distribution of {feature}')

    plt.xlabel(feature)

    plt.ylabel('Frequency')

    plt.show()

*# Box Plot*

    plt.figure(figsize=(8, 6))

    sns.boxplot(x=df\_train[feature])

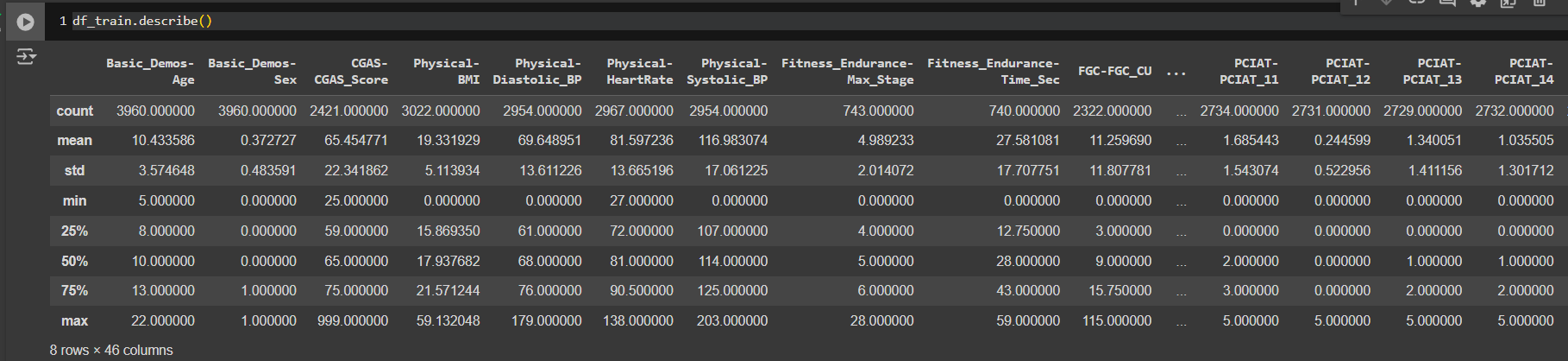
    plt.title(f'Box Plot of {feature}')

    plt.xlabel(feature)

    plt.show()

Here, I used bar chart and box plot for visualize the categorical and numerical values.

Summary Statistics



A screenshot of a computer

Description automatically generated