ECG Dataset EDA: Single Combined Table

LOADING DATASET

```
% Loads the data Cleaned Data
data = readtable('Cleaned_Combined_Dataset.xlsx');
```

SETTING UP DATA FOR EDA

```
% Step 1: Remove extreme outliers from a key feature (e.g., '0_rPeak')
if any(strcmp('0_rPeak', data.Properties.VariableNames))
    data= data(data.('0_rPeak') > -3000 & data.('0_rPeak') < 3000, :);
end

% Step 2: Normalize numeric features using z-score normalization
numericVars = varfun(@isnumeric, data, 'OutputFormat', 'uniform');
T_norm = data;
T_norm(:, numericVars) = varfun(@(x) (x - mean(x)) ./ std(x), data(:, numericVars));

% Step 3: Filter to desired beat types
desiredTypes = ["N", "SVEB", "VEB","F","Q"];
T_filtered = data(ismember(data.type, desiredTypes), :);

%% Dataset Overview
fprintf('Cleaned Dataset: %d rows, %d columns\n\n', size(T_filtered));
```

```
Cleaned Dataset: 963654 rows, 35 columns
```

```
disp('Preview of Data:');
```

```
Preview of Data:
```

```
disp(head(T_filtered, 5));
```

```
    record       type      x0_pre_RR    x0_post_RR    x0_pPeak     x0_tPeak     x0_rPeak     x0_sPeak     x0_qPeak
    _______    _______    _________    __________    ________    _________    _________    _________    ________

    {'I01'}    {'N'  }       163           165        0.06961    -0.083281      0.61413     -0.39276    0.047159
    {'I01'}    {'N'  }       165           166       -0.09703      0.59725    -0.078704    -0.078704    -0.13778
    {'I01'}    {'N'  }       166           102         0.1094      0.68053    -0.010649    -0.010649    -0.72062
    {'I01'}    {'VEB'}       102           231        0.17638      0.25643      -0.1011     -0.70752     -0.1011
    {'I01'}    {'N'  }       231           165        0.58558      0.60746    -0.083499    -0.083499    -0.16786
```

```
disp('Unique Beat Types Present:');
```

```
Unique Beat Types Present:
```

```
disp(unique(T_filtered.type));
```

```
    {'F'   }
    {'N'   }
    {'Q'   }
    {'SVEB'}
    {'VEB' }
```

```
disp('Unique Records Present:');
```

```
Unique Records Present:
```

```
if ismember('record', T_filtered.Properties.VariableNames)
    disp(unique(T_filtered.record));
end
```

```
    {'100'}
    {'101'}
    {'103'}
    {'105'}
    {'106'}
    {'108'}
    {'109'}
    {'111'}
    {'112'}
    {'113'}
    {'114'}
    {'115'}
    {'116'}
    {'117'}
    {'118'}
    {'119'}
    {'121'}
    {'122'}
    {'123'}
    {'124'}
    {'200'}
    {'201'}
    {'202'}
    {'203'}
    {'205'}
    {'207'}
    {'208'}
    {'209'}
    {'210'}
    {'212'}
    {'213'}
    {'214'}
    {'215'}
    {'219'}
    {'220'}
    {'221'}
    {'222'}
    {'223'}
    {'228'}
    {'230'}
    {'231'}
    {'232'}
    {'233'}
    {'234'}
    {'30' }
    {'31' }
    {'32' }
    {'34' }
    {'35' }
    {'36' }
```

```
{'41' }
{'45' }
{'46' }
{'49' }
{'51' }
{'52' }
{'800'}
{'801'}
{'802'}
{'803'}
{'804'}
{'805'}
{'806'}
{'807'}
{'808'}
{'809'}
{'810'}
{'811'}
{'812'}
{'820'}
{'821'}
{'822'}
{'823'}
{'824'}
{'825'}
{'826'}
{'827'}
{'828'}
{'829'}
{'840'}
{'841'}
{'842'}
{'843'}
{'844'}
{'845'}
{'846'}
{'847'}
{'848'}
{'849'}
{'850'}
{'851'}
{'852'}
{'853'}
{'854'}
{'855'}
{'856'}
{'857'}
{'858'}
{'859'}
{'860'}
{'861'}
{'862'}
{'863'}
{'864'}
{'865'}
{'866'}
{'867'}
{'868'}
{'869'}
{'870'}
{'871'}
{'872'}
{'873'}
{'874'}
```

```
{'875'}
{'876'}
{'877'}
{'878'}
{'879'}
{'880'}
{'881'}
{'882'}
{'883'}
{'884'}
{'885'}
{'886'}
{'887'}
{'888'}
{'889'}
{'890'}
{'891'}
{'892'}
{'893'}
{'894'}
{'I01'}
{'I02'}
{'I03'}
{'I04'}
{'I05'}
{'I06'}
{'I07'}
{'I08'}
{'I09'}
{'I10'}
{'I11'}
{'I12'}
{'I13'}
{'I14'}
{'I15'}
{'I16'}
{'I17'}
{'I18'}
{'I19'}
{'I20'}
{'I21'}
{'I22'}
{'I23'}
{'I24'}
{'I25'}
{'I26'}
{'I27'}
{'I28'}
{'I29'}
{'I30'}
{'I31'}
{'I32'}
{'I33'}
{'I34'}
{'I35'}
{'I36'}
{'I37'}
{'I38'}
{'I39'}
{'I40'}
{'I41'}
{'I42'}
{'I43'}
{'I44'}
```

```
    {'I45'}
    {'I46'}
    {'I47'}
    {'I48'}
    {'I49'}
    {'I50'}
    {'I51'}
    {'I52'}
    {'I53'}
    {'I54'}
    {'I55'}
    {'I56'}
    {'I57'}
    {'I58'}
    {'I59'}
    {'I60'}
    {'I61'}
    {'I62'}
    {'I63'}
    {'I64'}
    {'I65'}
    {'I66'}
    {'I67'}
    {'I68'}
    {'I69'}
    {'I70'}
    {'I71'}
    {'I72'}
    {'I73'}
    {'I74'}
    {'I75'}
```

```
%% Summary Statistics
numericData = T_filtered(:, varfun(@isnumeric, T_filtered, 'OutputFormat',
'uniform'));
disp('Mean Values of Numeric Features:');
```

```
Mean Values of Numeric Features:
```

```
disp(varfun(@mean, numericData));
```

```
    mean_x0_pre_RR    mean_x0_post_RR    mean_x0_pPeak    mean_x0_tPeak    mean_x0_rPeak    mean_x0_sPeak    mean_x
    ______________    _______________    _____________    _____________    _____________    _____________    ______

        199.29            203.19           0.051941          0.43303          0.48327         -0.35278        -0.1
```

```
disp('Standard Deviation of Numeric Features:');
```

```
Standard Deviation of Numeric Features:
```

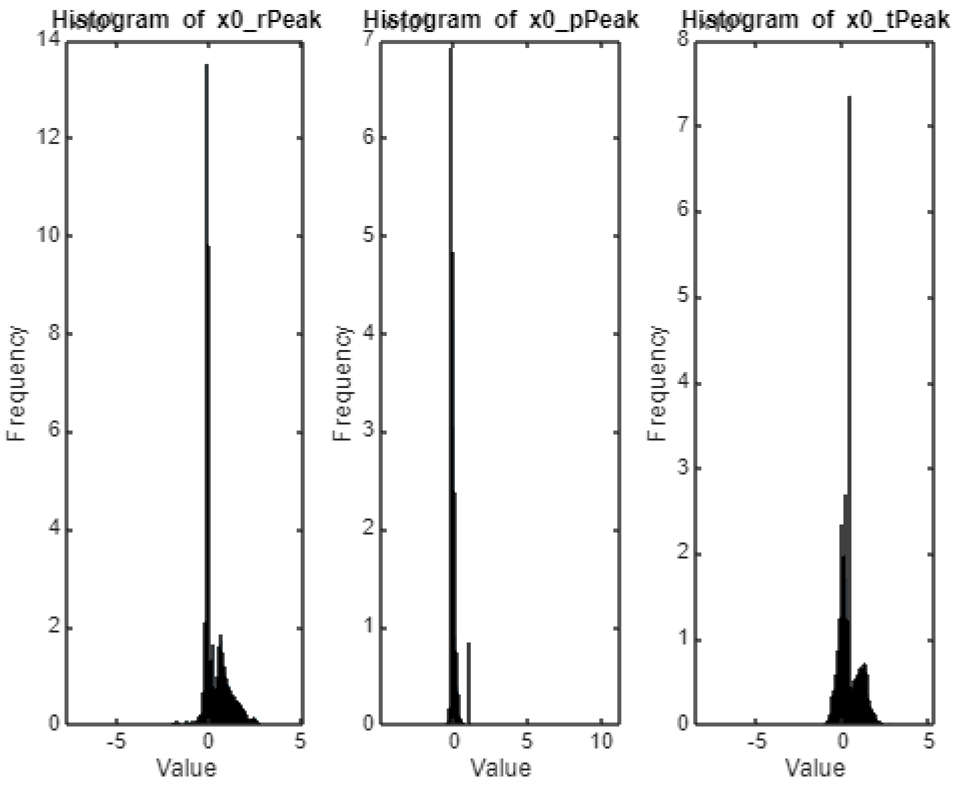
```
disp(varfun(@std, numericData));
```

```
    std_x0_pre_RR    std_x0_post_RR    std_x0_pPeak    std_x0_tPeak    std_x0_rPeak    std_x0_sPeak    std_x0_qPeak
    _____________    ______________    ____________    ____________    ____________    ____________    ____________

       78.755            146.76          0.21428         0.63055         0.74088         0.67201         0.26891
```

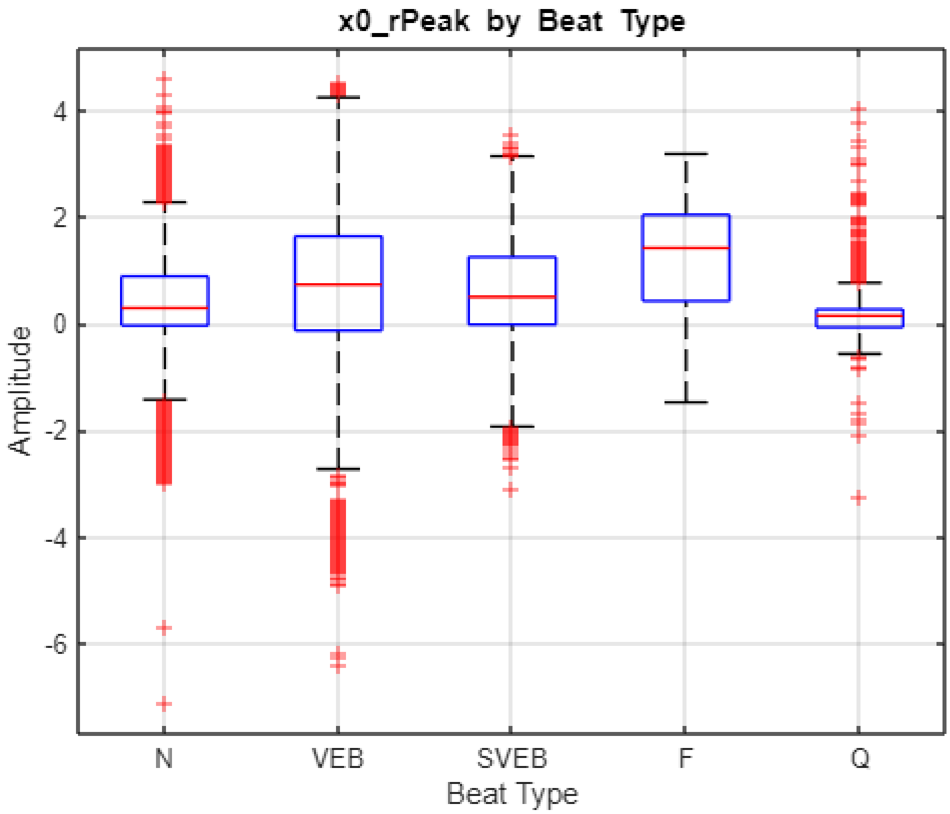
HISTOGRAM

```
%% Histograms of Selected Features
selectedFeatures = {'x0_rPeak', 'x0_pPeak', 'x0_tPeak'};
figure;
for i = 1:length(selectedFeatures)
    if ismember(selectedFeatures{i}, T_filtered.Properties.VariableNames)
        subplot(1, length(selectedFeatures), i);
        histogram(T_filtered.(selectedFeatures{i}));
        title(['Histogram of ', strrep(selectedFeatures{i}, '_', '\_')]);
        xlabel('Value'); ylabel('Frequency');
    end
end
```



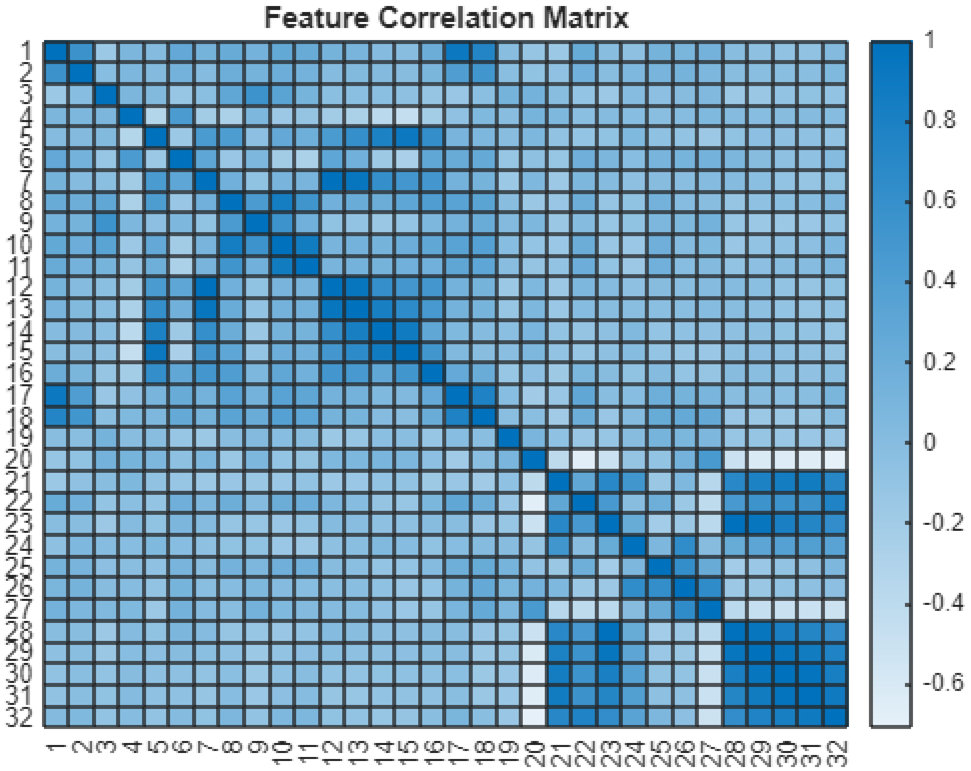
BOXPLOT

```
%% Boxplot: 0_rPeak by Beat Type
if ismember('x0_rPeak', T_filtered.Properties.VariableNames)
    figure;
    boxplot(T_filtered.('x0_rPeak'), T_filtered.type);
    title('x0\_rPeak by Beat Type');
    xlabel('Beat Type'); ylabel('Amplitude');
    grid on;
end
```



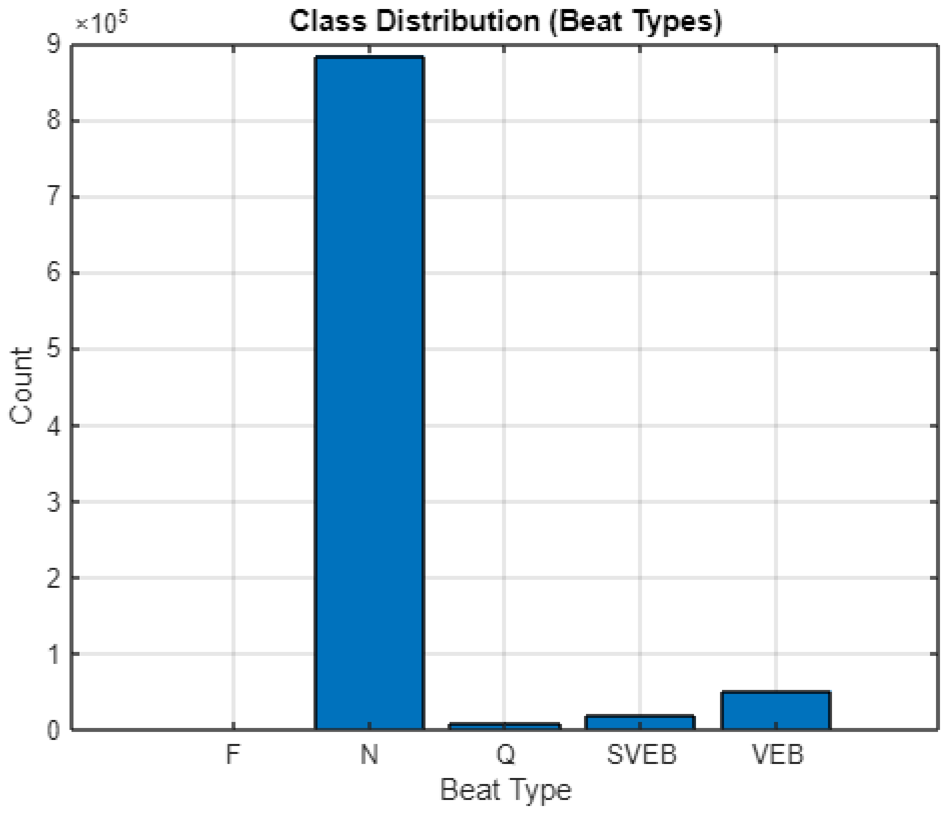
CORRELATION MATRIX

```
%% Correlation Matrix
if ~isempty(numericData)
    numericMatrix = table2array(numericData);
    corrMatrix = corr(numericMatrix, 'Rows', 'complete');
    figure;
    heatmap(corrMatrix);
    title('Feature Correlation Matrix');
end
```



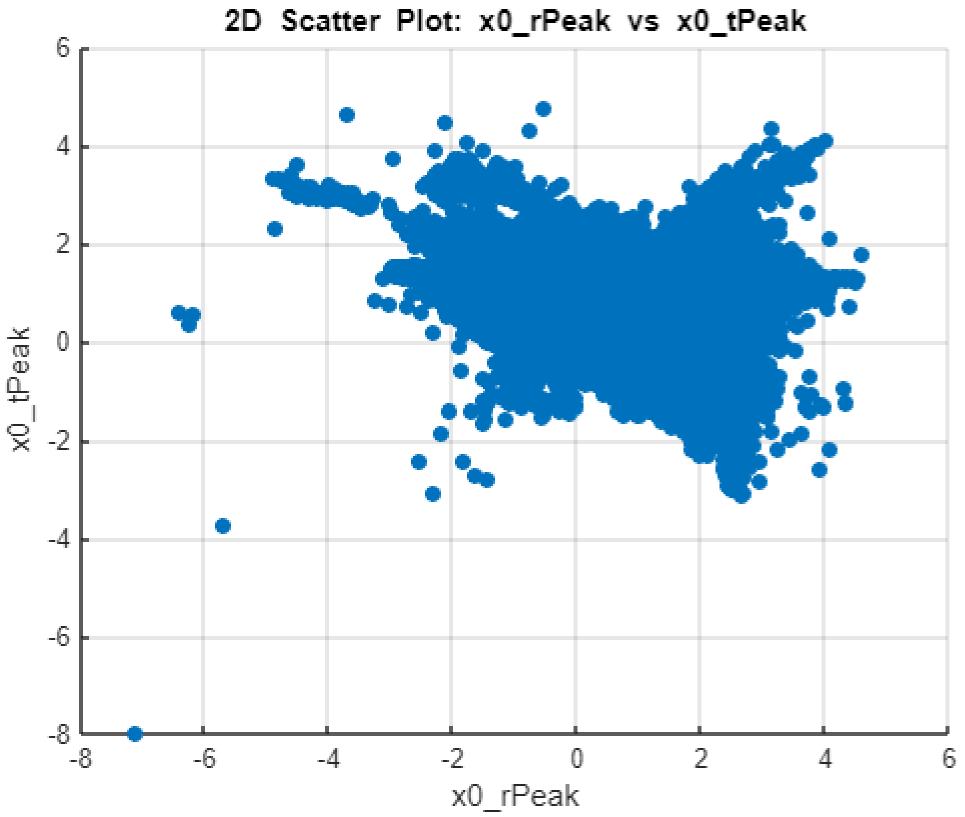
BAR PLOT

```
%% Class Distribution Bar Plot
typeCounts = groupcounts(T_filtered.type);
figure;
bar(typeCounts);
xticks(1:length(typeCounts));
xticklabels(unique(T_filtered.type));
title('Class Distribution (Beat Types)');
xlabel('Beat Type'); ylabel('Count');
grid on;
```



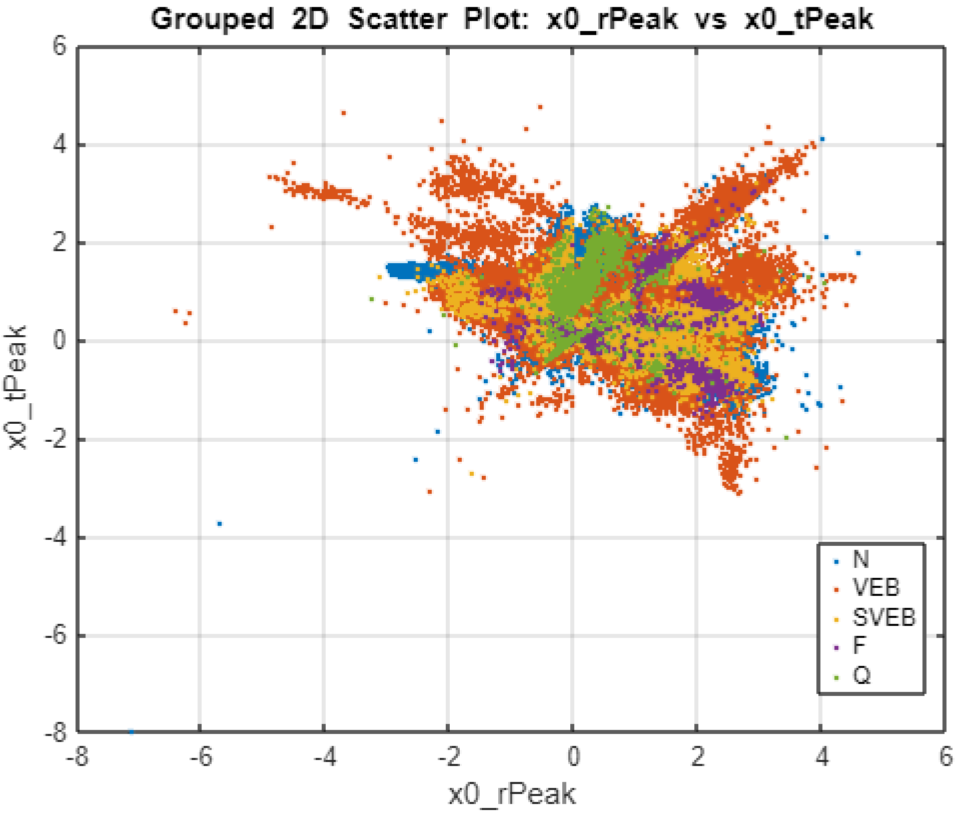
2D SCATTER PLOT SINGLE COLOR

```
% Ensure the selected features exist
if all(ismember({'x0_rPeak', 'x0_tPeak'}, data.Properties.VariableNames))
    figure;
    scatter(data.x0_rPeak, data.x0_tPeak, 36, 'filled');
    title('2D Scatter Plot: x0\_rPeak vs x0\_tPeak');
    xlabel('x0\_rPeak');
    ylabel('x0\_tPeak');
    grid on;
end
```



2D SCATTER PLOT MULTICOLOR

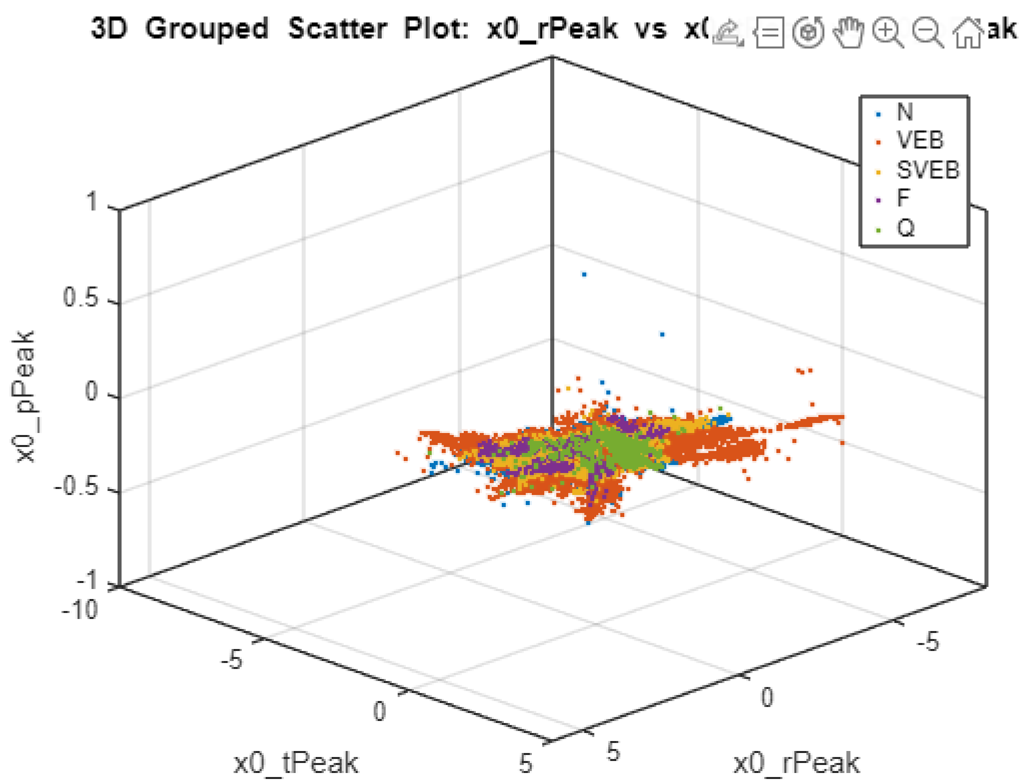
```
% Check feature availability
if all(ismember({'x0_rPeak', 'x0_tPeak'}, data.Properties.VariableNames))
    figure;
    gscatter(data.x0_rPeak, data.x0_tPeak, data.type);
    title('Grouped 2D Scatter Plot: x0\_rPeak vs x0\_tPeak');
    xlabel('x0\_rPeak');
    ylabel('x0\_tPeak');
    grid on;
    legend('Location', 'best');
end
```



3D GSCATTER PLOT

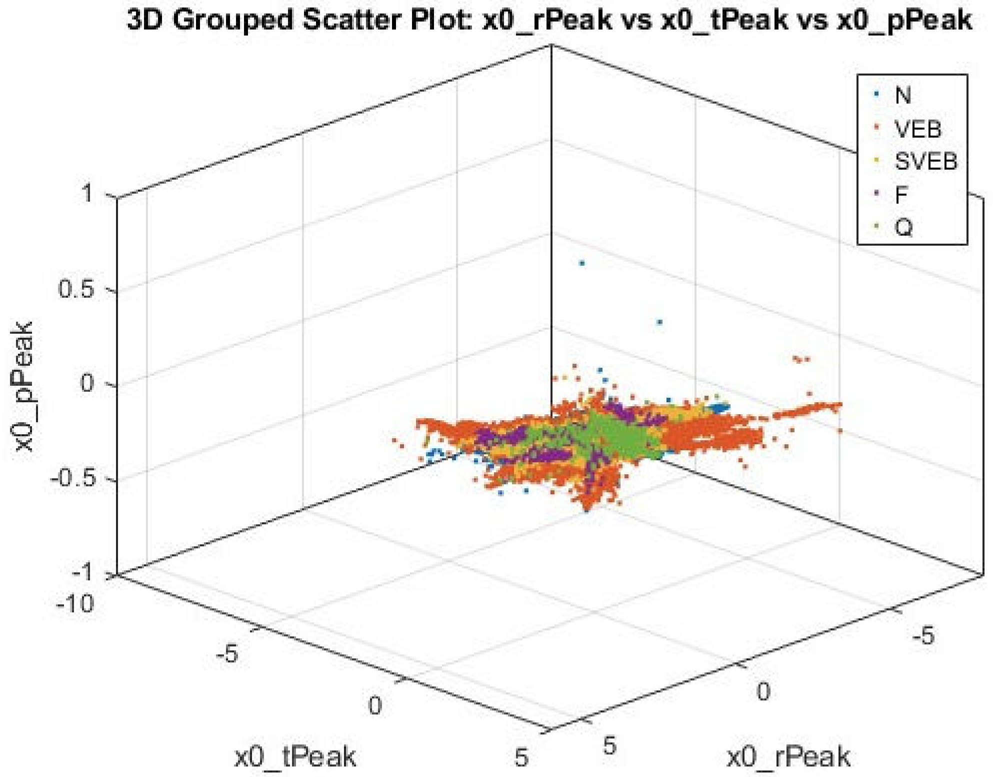
```
% Assuming beatTypes and data.type are defined
% Also assumes data.x0_rPeak, data.x0_tPeak, data.x0_pPeak are numeric vectors of the same size

% Grouped 3D scatter using custom function
gscatter3(data.x0_rPeak, data.x0_tPeak, data.x0_pPeak, ...
          data.type, ...               % Grouping variable
          'lines', ...                 % Color map
          'o', ...                     % Marker style
          5, ...                       % Marker size
          'auto', ...                  % Marker face color
          1, ...                       % Show legend
          'NorthEastOutside');    % Legend location
```



```
If mrksle is either a cell array of strings or a character array,
the length of mrksle must be either of length of g or of length equal to no of groups.
```

```
% Add labels and title
title('3D Grouped Scatter Plot: x0\_rPeak vs x0\_tPeak vs x0\_pPeak');
xlabel('x0\_rPeak');
ylabel('x0\_tPeak');
zlabel('x0\_pPeak');
view(135, 30);  % Adjust viewing angle
grid on;
```



SUMMARY OF THE DATA SET

```
%% Final Class Summary
fprintf('\nClass Counts:\n');
```

```
Class Counts:
```

```
disp(['Total Records: ' num2str(height(T_filtered))]);
```

```
Total Records: 963654
```

```
disp(['Normal Beats (N): ' num2str(sum(strcmp(T_filtered.type, 'N')))]);
```

```
Normal Beats (N): 884214
```

```
disp(['VEB Beats: ' num2str(sum(strcmp(T_filtered.type, 'VEB')))]);
```

```
VEB Beats: 51669
```

```
disp(['SVEB Beats: ' num2str(sum(strcmp(T_filtered.type, 'SVEB')))]);
```

```
SVEB Beats: 18540
```

```
disp(['Fusion Beats (F): ' num2str(sum(strcmp(T_filtered.type, 'F')))]);
```

```
Fusion Beats (F): 1256
```

```
disp(['Unknown Beats (Q): ' num2str(sum(strcmp(T_filtered.type, 'Q')))]);
```

```
Unknown Beats (Q): 7975
```

```
% Prepare for further modeling
types = T_filtered.type;
data = T_filtered;
```