1. Functions **get_feature()**, **get_feature_heart split()** split data into two set in accordance to presence\absence the heart disease and then extract certain feature values and plot histograms of positive/negative cases in accordance to it.
This allows to detect correlation of feature and the disease - if histogram peaks for different groups are dispersed, which means that the disease is characteristic of a certain value of the feature.
We try it to "Age" and "Weight" features and observe no correlation.

   def get_feature(feature_name)
   # feature_name - name of the person feature in accordance to which we want to plot histogram

2. Functions **get_feature_sin_rhytm()** do the same things but in case of maximum of two features corresponding to some observable (feature 'Sin' in or case). In this case a certain correlation of existence the sinus rhythm with the presence of the disease is observed.

   def get_feature_sin_rhytm(data)
   # data - dataset

3. Part "From ECG features to 2D".
This section extracts from initial table features that correspond to one group (dose of drus, for example, group could be made up by records about drug dose or value of ECG features).

   Now for every person we have a feature vector of features from the group. We map this vector to 2D dimension using either PCA (model finding or TSNe(finding k principal component by changing basis) or t-SNE (model finding low-dimential representation by minimizing the Kullback-Leibler divergence between the joint probabilities of coordinates).

   Then we plot persons as a point on plain with labels describing whether the person has a disease or not.

   def plot_points_by_bokeh(x, y, labels)
   # x - array of size (N), x coordinate obtained by PCA/TSNe for every person
   # y - array of size (N), x coordinate for every person
   # labels - label for every person, in our case presence/absence heart disease (True/False).

4. Function **plot_time_dinamic()** illustrates the changes in one of the electrocardiogram parameters for a patient with fixed ID during the time.
It can be useful both in monitoring the patient's conditions during the time and in finding the dependencies of the ECG parameters on the presence / absence of a symptom (provided that at different time steps a symptom may appear or not).

def plot_time_dinamic(data, user_id, feature)
 # data - dataset
 # user_id - inedntificatio of user we want to observe
 # feature - ECG feature we want to change over time that we want to observe