[Afib query code] by [MATLAB]

%This is a detailed example of processing primitive data for ‘hidden’ exposures. Since atrial fibrillation with rapid ventricular response(AfibRVR) is not a defined variable/event in the database, recognition of these event requires a separate algorithm to combine multiple information and conditions. In this case, we chose Matlab to develop a sophisticated algorithm that recognize an event that meets multiple criteria, including heart rate(HR) over 110 beats/min, received rate control medication when the HR is over 110, HR is over 110 for a substantial time, HR subsequently under control over a substantial time.

%% make dataset matrix, adjust column number

dataset = zeros( length(med\_s), 21 );

% error tolerance variable control

% this controls the time window of looking for HR >=110 around the 1st\_drug time

find\_rvr\_window = datenum([0 0 0 2 0 0]);

% this controls the time window of looking for drugs administered around the RVR episode

find\_drug\_window = datenum([0 0 0 2 0 0]);

%This part cleans up the HR data. Although HR data is a structured data by definition, several reasons make this data look like an unstructured data, including the fact that one patient could have thousands of HR data, HR being highly variable as well as missing or incomplete recording of HR.

[~,index]=unique(HR\_data\_ts\_sorted(:,12),'stable');

HR\_data\_ts\_sorted = HR\_data\_ts\_sorted(sort(index),:);

% HR data(value and tn) interpolation

% added for revised\_new\_strategy

HR\_dn = HR\_data\_ts\_sorted(:,12);

HR\_val = HR\_data\_ts\_sorted(:,4);

min\_step = datenum([0 0 0 0 5 0]); % ASSUMING MINS!!!

HR\_dn\_new = HR\_dn(1):min\_step:HR\_dn(end);

HR\_val\_new = interp1(HR\_dn,HR\_val,HR\_dn\_new,'linear');

interpolated\_HR\_m =[HR\_val\_new', HR\_dn\_new'];

% \*\*\*\*\*now you have all the data of this\_subject needed for analysis\*\*\*\*\*

else

meds\_data\_ts\_sorted = [];

vaso\_data\_ts\_sorted = [];

HR\_data\_ts\_sorted = [nan,nan,this\_s];

first\_drug = nan;

first\_drug\_dn = nan;

end

% end of get data within ICUSTAY

% % % % % % % % % % % % % % % % % % % % % % % %

%\*\*\*\*\*\*\*\*\_Core\_Algorithm

—Recognize a substantial RVR event\_\*\*\*\*\*\*\*\*\*

check\_hr\_start\_dn = first\_drug\_dn - find\_rvr\_window;

check\_hr\_stop\_dn = first\_drug\_dn + find\_rvr\_window;

within\_check\_index = find(check\_hr\_start\_dn <= interpolated\_HR\_m(:,2) & interpolated\_HR\_m(:,2) <= check\_hr\_stop\_dn );

interpolated\_HR\_m\_within\_check = interpolated\_HR\_m(within\_check\_index,:);

first\_rvr\_within\_check\_i = find( (interpolated\_HR\_m\_within\_check(:,1)>=110)==1, 1 ); % find the first hr\_tn where HR>=110 within the checking period

% syntax: find(A,1) A: condition

if ~isempty(first\_rvr\_within\_check\_i)

rvr\_start\_dn = interpolated\_HR\_m\_within\_check( first\_rvr\_within\_check\_i ,2);

rvr\_start\_i = find( interpolated\_HR\_m(:,2)==rvr\_start\_dn );

rvr\_start\_HR = interpolated\_HR\_m(rvr\_start\_i,1);

else

rvr\_start\_dn =nan;

rvr\_start\_i = nan;

rvr\_start\_HR = nan;

end

% find the end of RVR per criteria 3b

% asign value to controlled\_for\_4hour label

if ~isnan(rvr\_start\_i)

check\_end = find( interpolated\_HR\_m(rvr\_start\_i:end,1)<110 );

check\_rvr\_end\_points\_i = check\_end + (rvr\_start\_i-1);

if ~isempty(check\_end)

for j = 1:length(check\_rvr\_end\_points\_i)

if check\_rvr\_end\_points\_i(j)+47<=length(interpolated\_HR\_m)

if nnz( interpolated\_HR\_m( check\_rvr\_end\_points\_i(j):check\_rvr\_end\_points\_i(j)+47 ,1) >=110 )<=5

rvr\_end\_i = check\_rvr\_end\_points\_i(j);

rvr\_end\_dn = interpolated\_HR\_m(rvr\_end\_i,2);

controlled\_for\_4hour = 1; %dataset column 5 output

break

end

elseif check\_rvr\_end\_points\_i(j)+47 > length(interpolated\_HR\_m)

if nnz( interpolated\_HR\_m( check\_rvr\_end\_points\_i(j):end,1) >=110 )<= (length(interpolated\_HR\_m) - length(interpolated\_HR\_m(1:check\_rvr\_end\_points\_i(j))))/10

rvr\_end\_i = check\_rvr\_end\_points\_i(j);

rvr\_end\_dn = interpolated\_HR\_m(rvr\_end\_i,2);

controlled\_for\_4hour=0.5; %dataset column 5 output

break

end

end

rvr\_end\_i=nan;

rvr\_end\_dn=nan;

controlled\_for\_4hour=0; %dataset column 5 output

end

elseif isempty(check\_end)

rvr\_end\_i=nan;

rvr\_end\_dn=nan;

controlled\_for\_4hour=0; %dataset column 5 output

end

elseif isnan(rvr\_start\_i)

rvr\_end\_i=nan;

rvr\_end\_dn=nan;

controlled\_for\_4hour=nan; %dataset column 5 output

end

% get rvr\_duration, if no true end of RVR identified previously, take

% the last HR record tn for calculating the duration

if ~isnan(rvr\_start\_dn) && ~isnan(rvr\_end\_dn)

rvr\_duration = etime( datevec(rvr\_end\_dn) , datevec(rvr\_start\_dn) )/60.0; %dataset column 4 output

elseif ~isnan(rvr\_start\_dn) && isnan(rvr\_end\_dn)

rvr\_duration = etime( datevec(interpolated\_HR\_m(end,2)) , datevec(rvr\_start\_dn) )/60.0; %dataset column 4 output

else

rvr\_duration = nan; %dataset column 4 output

end

% find data only within the rvr episode( error tolerance time window considered )

find\_drug\_start\_dn = rvr\_start\_dn - find\_drug\_window;

find\_drug\_end\_dn = rvr\_end\_dn + find\_drug\_window;

if ~isnan(rvr\_start\_dn) && ~isnan(rvr\_end\_dn)

meds\_withinrvr\_index = find(find\_drug\_start\_dn <=meds\_data\_ts\_sorted(:,11) & meds\_data\_ts\_sorted(:,11) <= find\_drug\_end\_dn );

meds\_data\_ts\_withinrvr = meds\_data\_ts\_sorted(meds\_withinrvr\_index,:);

vaso\_withinrvr\_index = find( find\_drug\_start\_dn <=vaso\_data\_ts\_sorted(:,11) & vaso\_data\_ts\_sorted(:,11) <= find\_drug\_end\_dn );

vaso\_data\_ts\_withinrvr = vaso\_data\_ts\_sorted(vaso\_withinrvr\_index,:);

elseif ~isnan(rvr\_start\_dn) && isnan(rvr\_end\_dn)

meds\_withinrvr\_index = find(find\_drug\_start\_dn <=meds\_data\_ts\_sorted(:,11) );

meds\_data\_ts\_withinrvr = meds\_data\_ts\_sorted(meds\_withinrvr\_index,:);

vaso\_withinrvr\_index = find( find\_drug\_start\_dn <=vaso\_data\_ts\_sorted(:,11) );

vaso\_data\_ts\_withinrvr = vaso\_data\_ts\_sorted(vaso\_withinrvr\_index,:);

else

meds\_data\_ts\_withinrvr =[];

vaso\_data\_ts\_withinrvr =[];

end

% % % % % % % % % % % % % % % % % % % % % % % %

% end of find data within RVR episode