Table of Contents


```
% sum_file = 'Stanley.sum';
% c = read_sum_file('stanley_catsum.sum');
clear all close all;
c = read_sum_file('import any sum file here')
c( all( cell2mat( arrayfun( @(x) structfun( @isempty, x ), c, 'UniformOutput',
 false ) ), 1 ) ) = [];
Cat = struct2table(c);
dates = datetime(Cat.otime, 'convertfrom', 'datenum'); % create a Datetime
 column
Cat.Dates = dates;
for i = 1:height(Cat)
       event_id(i) = (i);
       Cat.eventID(i)=  event_id(i) ;
end
```

```
Error using fgets
Invalid file identifier. Use fopen to generate a valid file identifier.

Error in fgetl (line 32)
[tline,lt] = fgets(fid);

Error in read_sum_file>linecount (line 92)
tline = fgetl(fid);

Error in read_sum_file (line 3)
n = linecount(filename); % determine the number of lines
```

```
Error in Stanley_EQT_Events (line 4)
c = read_sum_file('import any sum file here')
```

depth constraints

thresholdmin = 4.99; thresholdmax = 5.09; Cat(Cat.depth > thresholdmin & Cat.depth < thresholdmax,:) = [] ; % Cat(Cat_badDepths,:)= [];

create t

```
T = readtable('1970_present_EQ_westernUS_M2.5.xlsx');%this is the full catalog
 noe filter using time and space
T2 = length(T.horizontalError);

formatIn = 'yyyy-mm-ddTHH:MM:SS' %tell matlab the format for time in the
 catalog T
t1 = T.time(:,1);
t1 = datenum(t1, formatIn); %Get the correct format to compare the times
 between arrays
T.time = t1;
```

STANLEY create table

```
clear  i  idx St
St = array2table(zeros(length(T2),6));
St.Properties.VariableNames = {'otime','lon', ...
           'lat','depth', 'horErr', 'rms'}

%for loop for horizontal error
for i = 1:T2
    if i>49630 && i< 52643 && 47.46 >= T.latitude(i)...
        && T.latitude(i) >= 41.46 && T.longitude(i) >= -117.636 &&
 T.longitude(i) <= -112.636

        St.otime(i) = T.time(i);
        St.lat(i) = T.latitude(i);
        St.lon(i) = T.longitude(i);
        St.depth(i) = T.depth(i);
        St.horErr(i) = T.horizontalError(i);
        St.rms(i) = T.rms(i);
        %quality

    end

end
St(~St.otime,:) = [] % deletes the rows where otime is 0 and save the table as
 C
```

For A events

```
clear A_evs A_Events B_evs B C C_evs;
```

```
A_Events = strcmp(Cat.quality, 'A') ;% create logical index of all rows that
 are A events =1 and if not =0
A_evs = Cat(A_Events, :) ; % index into table and grab rows with 'A'
A_MR = mean(A_evs.rmse);
AMD = mean(A_evs.depth);
A_MST= std(A_evs.rmse);
```

For B events

```
B = strcmp(Cat.quality, 'B') ;
B_evs = Cat(B, :) ;
B_MR = mean(B_evs.rmse);
BMD = mean(B_evs.depth);
B_MST= std(B_evs.rmse);
```

For C events

```
C = strcmp(Cat.quality, 'C') ;
C_evs = Cat(C, :);
C_MR = mean(C_evs.rmse);
CMD = mean(C_evs.depth);
C_MST= std(C_evs.rmse);
```

Ac_Events = strcmp(c.quality, 'A') % create logical index of all rows that are A events =1 and if not =0 Ac_evs = c(Ac_Events, :) % index into table and grab rows with 'A'

For D events

```
D = strcmp(Cat.quality, 'D') ;
D_evs = Cat(D, :);
D_MR = mean(D_evs.rmse);
DMD = mean(D_evs.depth);
D_MST= std(D_evs.rmse);
```

Percentage of Events That are A, B, C, D

```
 A_st_perc = (height(A_evs)/ height(Cat))*100
 B_st_perc = (height(B_evs)/ height(Cat))*100
 C_st_perc = (height(C_evs)/ height(Cat))*100
 D_st_perc = (height(D_evs)/ height(Cat))*100
```

find Percentage of quality events in each month for Stanley events

```
 [C_chron, Idx] = sort(C_evs.otime);
 [D_chron, idd] = sort(D_evs.otime);
 [A_chron, ida] = sort(A_evs.otime);
 [B_chron, idb] = sort(B_evs.otime);

figure(3); clf
```

```
plot(C_evs.Dates(Idx), 'o', 'color', [0.5 0 0.8]); hold on %these are the
 purple
plot( D_evs.Dates(idd),'o', 'color', [0 0.52 0.10]);  %these are green
plot(B_evs.Dates(idb), 'o', 'color', 'r');
plot(A_evs.Dates(ida), 'o', 'color', 'b');
xlabel('count')
ylabel('DateTime')
view(0,90);
legend('C events', 'D events' , 'B events', 'A events')
grid on
title('Stanley Event Perentage in EQT catlog ')
```

MAke Table of Percentage of Quality Events

```
 QT = table([ A_st_perc; B_st_perc;  C_st_perc;  D_st_perc ],[height(A_evs);
 height(B_evs); height(C_evs); height(D_evs)], 'VariableNames',
 {'Percent', 'Number of Events'}, 'RowNames', {'Stanley As', 'Stanley
 Bs', 'Stanley Cs', 'Stanley Ds'})
```

Make Table figure

```
 f = figure;
uit = uitable(f, 'Data', table2cell(QT));
uit.ColumnName={QT.Properties.VariableNames{:}};
uit.RowName={QT.Properties.RowNames{:}}; %removing default row numbering as in
 your uitable
saveas(f, 'figname.png')
```

Histogram of USGS Events by month Stanley

```
dates = datetime(St.otime, 'convertfrom', 'datenum');
St.Dates = dates;
figure(14);
subplot(2,1,1);
St_edges = dateshift(min(Cat.Dates), "start", "month"):
 calmonths(1):dateshift(max(St.Dates), "start","month", "next");
histogram(St.Dates, St_edges)
title('Stanley USGS Events n=1023')
xlabel('Months')
% xticks([ 2 3 4 5 6 7 8 9 10 11 12 13])
% xticklabels({'January', 'Feburuary', 'March', 'April', 'May', 'June',
 'July', 'August', 'September', 'October', 'November', 'December'})
ylabel('Number of Events')


Cat_edges = dateshift(min(Cat.Dates), "start", "month"):
 calmonths(1):dateshift(max(Cat.Dates), "start","month", "next");
subplot(2,1,2);
histogram(Cat.Dates, Cat_edges);
title(sprintf('Stanley EQT Events n=%d', height(Cat)));
xlabel('Months');
```

```
% xticks([ 2 3 4 5 6 7 8 9 10 11 12 13])
% xticklabels({'January', 'Feburuary', 'March', 'April', 'May', 'June',
 'July', 'August', 'September', 'October', 'November', 'December'})
ylabel('Number of Events');
```

create standard deviation by month

```
clear std_month
std_month = zeros(1,length(Cat_edges));
depth_month = zeros(height(Cat), length(Cat_edges));

for ii =
 min(month(Cat.Dates,"monthofyear")):max(month(Cat.Dates,"monthofyear"))
        for i = 1 : height(Cat)
             if Cat.Months(i) == ii
                 depth_month(i, ii) = Cat.depth(i);
                depth_month(depth_month==0) = NaN; %this removes all the zeros
 from the rows
             else
             end
        end
    std_month(:,ii) = nanstd(depth_month(:,ii));
end
%remove zeros from std_month
std_month(std_month == 0)=[];
```

Plot Depths now

```
figure(2); clf
subplot(3,1,1);
sz = 15;
scatter(Cat.Dates(find(Cat.depth ~= 5)), Cat.depth(find(Cat.depth ~=
 5)),sz, 'o', 'filled'); hold on
title('All Events')
%
 plot(min(month(Cat.Dates,"monthofyear")):max(month(Cat.Dates,"monthofyear")),
 std_month,'r')
ylim([-3 20])
xlabel('Months')
ylabel('Depth (km)')
set(gca, 'YDir','reverse')
subplot(3,1,2);
Cat_no_d = Cat(~D, :); %removes the D's from the catalog using ~
scatter(Cat_no_d.Dates(find(Cat_no_d.depth ~= 5)),
 Cat_no_d.depth(find(Cat_no_d.depth ~=
 5)),sz, 'o', 'filled', 'MarkerFaceColor','r'); hold on
title('No D Events')
ylim([-3 20])
xlabel('Months')
ylabel('Depth (km)')
set(gca, 'YDir','reverse')
subplot(3,1,3);
Cat_no_dc = strcmp(Cat_no_d.quality, 'C') ;
```

```
Cat_NDC = Cat_no_d(~Cat_no_dc, :);

scatter(Cat_NDC.Dates(find(Cat_NDC.depth ~= 5)),
 Cat_NDC.depth(find(Cat_NDC.depth ~=
 5)),sz, 'o', 'filled', 'MarkerFaceColor','g'); hold on
find(Cat_NDC.depth == 5)
title('No D or C Events')
xlabel('Months')
ylabel('Depth (km)')
ylim([-3 20])

set(gca, 'YDir','reverse')
```

Geo Statistics for Stanley

```
figure(3)

Cat_md = mean(Cat_NDC.depth(find(Cat_NDC.depth ~= 5)));
Cat_st = std(Cat_NDC.depth(find(Cat_NDC.depth ~= 5)));
% histogram(Cat_st, 'Normalization','pdf')

x0 = 0:Cat_st/10:(Cat_md+2*Cat_st); %sort the values from least to greatest by
 10 off the average

figure(3); clf

[distr, xbins]= hist(Cat_NDC.depth, x0);
dx = Cat_st/100; %bin width
% dist_out = max(distr);
distr = distr / sum(distr)*10; %the probablitly

bar(xbins, distr)
ylabel('Probability')
xlabel('Depth (km)')
ylim([0,1]);
xlim([0, 15]);
view([90,90])
% histogram(Cat_st,xbins)
```

Geo Statistics for Stanley Horizontal Error

```
clear Cat_md Cat_st;

Cat_md = mean(Cat_NDC.Herr(find(Cat_NDC.depth ~= 5)));
Cat_st = std(Cat_NDC.Herr(find(Cat_NDC.depth ~= 5)));
% histogram(Cat_st, 'Normalization','pdf')

x0 = 0:Cat_st/6:(Cat_md+2*Cat_st); %sort the values from least to greatest by
 10 off the average

figure(5); clf

[distr, xbins]= hist(Cat_NDC.Herr(find(Cat_NDC.depth ~= 5)), x0);
```

```
dx = Cat_st/100; %bin width
% dist_out = max(distr);
distr = distr / sum(distr);


bar(xbins, distr)
ylabel('Probability')
xlabel('Horizontal Error (km)')
ylim([0,0.2]);
xlim([0, 0.31]);
% histogram(Cat_st,xbins)
```

Geo Statistics for Stanley Vertical Error

```
clear Cat_md Cat_st;

Cat_md = mean(Cat_NDC.Verr(find(Cat_NDC.depth ~= 5)));
Cat_st = std(Cat_NDC.Verr(find(Cat_NDC.depth ~= 5)));
% histogram(Cat_st, 'Normalization','pdf')

x0 = 0:Cat_st/2:(Cat_md+2*Cat_st); %sort the values from least to greatest by
 10 off the average

figure(5); clf

[distr, xbins]= hist(Cat_NDC.Verr(find(Cat_NDC.depth ~= 5)), x0);
dx = Cat_st/100; %bin width
% dist_out = max(distr);
distr = distr / sum(distr); %mean of standrad deviations


bar(xbins, distr)
ylabel('Probabilty')
xlabel('Vertical Error')
ylim([0,.4]);
xlim([0, 1]);
```

Geo Statistics for Stanley RMSE

```
figure(3)
Cat_md = mean(Cat_no_d.rmse(find(Cat_NDC.depth ~= 5)));
Cat_st = std(Cat_no_d.rmse(find(Cat_NDC.depth ~= 5)));
% histogram(Cat_st, 'Normalization','pdf')

x0 = 0:Cat_st/20:(Cat_md+2*Cat_st); %sort the values from least to greatest by
 10 off the average

figure(3); clf

[distr, xbins]= hist(Cat_NDC.rmse(find(Cat_NDC.depth ~= 5)), x0);
dx = Cat_st/100; %bin width
% dist_out = max(distr);
distr = distr / sum(distr);
```

```
bar(xbins, distr)
ylabel('probabilty')
xlabel('Travel Time Residual Error (seconds)')
ylim([0,.25]);
xlim([-1, 15]);
% histogram(Cat_st,xbins)
```

Set Up Time arrays for USGS AND EQT COM-PARISON

```
clear t1 t2_n t2;
t1 = St.otime(:,1); % USGS origin times
t2_st = zeros(height(Cat),1);
for y = 1:height(Cat)
    t2_st(y) = Cat.otime(y) ;                  %t2 = inl(k).otime; for
 structure
end
```

%% MATCH THE USGS EVENTS TO EQT for Stanley

```
clear dt dist val ids counter ;
 %Change the format to math the sum files and compare the times
E = referenceEllipsoid('Earth'); % reference ellipse [m] for distance
 calculation
counter = 0;

for ii = 1 : numel(t1) % loop through eqt events CHANGE TO T1

    % find the USGS event that is closest in time to eqt_event(i)
    t_diff  = t2_st - t1(ii);
    [val, ids(ii)] = min( abs(t_diff) ); % this lets them be negative

    % Display some useful information to the user



    fprintf('EQT: %s, Seisan: %s\n', datestr(t2_st(ids(ii))),
 datestr(t1(ii)));

    dt(ii) = val * 24*3600; % [s] convert time difference to seconds
%      dt_nm(i) = dt(i);
    if dt(ii) > 10
        counter = counter +1 ;
```

```
    else

                % compute distance in meters
        dist(ii) = distance( Cat.lat(ids(ii)), Cat.lon(ids(ii)),...  % right
 now this is set to run for A events
        St.lat(ii), St.lon(ii), E) ; %CHANGE CATALOGS HERE
        dist_nm(ii) = dist(ii); %create new vectors that keep track of these
 values
%         plot(q_out(idx(i)), dist_nm);  % create cross plot that shows
 quality type versus delay time
%
% writetable(Cat(ids(ii)),'eqt_matched_sawvel.csv','Delimiter',',')
%         type 'eqt_matched_sawvel.csv'

    end
end
```

Plots for Stanley Sawtooth Vel (EDIT)

```
clear h
title_text = sprintf('n=%d', length(t1)-counter );

h = figure;

% time differences: use 0.1s bins from 0 to 2 seconds
subplot(1,3,1);
histogram(dt, 0:0.1:10); grid on;
xlabel('Origin time difference [s]');
ylabel('No. events'); title(title_text);

% epicenter distance difference: use 1km bins from 0 to 15 kilometers
subplot(1,3,2);
histogram(dist./1000, 0:1:15); grid on;
xlabel('Epicenter location difference [km]');
ylabel('No. events'); title(title_text);

% 1:1 depth plot: plot the USGS depth vs. the USGS depths
subplot(1,3,3);
plot([0, 30],[0, 30],'k'); hold on; title(title_text);
plot(Cat.depth(ids), St.depth,'ko'); grid on;
xlabel('Revised Depth [km]');
ylabel('USGS Catalog Depth [km]');
axis([0 30 0 30]);

set(h,'color','w');
set( findall( h, '-property', 'FontSize' ), 'FontSize', 18 );
set( findall( h, '-property', 'FontName' ), 'FontName', 'Helvetica' );
set( findall( h, '-property', 'FontWeight' ), 'FontWeight', 'Bold' );
set( h, 'Position', [100 100 1500 400] );
set( h, 'PaperPositionMode', 'auto' );
```

Published with MATLAB® R2021b