Contents


```
close all; clc;
if exist('Station_name','var')
inputs.Station_name= Station_name;
else
  inputs.Station_name=input('Insert the station name :','s');
end
```

User Inputs

```
inputs.Minor_Th= 1.84; % Insert minor flood threshold [ft]
inputs.Major_Th= 4.03; % Insert minor flood threshold [ft]

inputs.Return_Period=[1,50,100,200,500,1000]; % Insert the desired return period
```

Reading the mean and maximum daily values from Inputs folder

```
if ispc
    cd([pwd,'\Inputs'])
else
    cd([pwd,'/Inputs'])
end

inputs.DayMax_byYear=xlsread(strcat(inputs.Station_name,'_DailyWaterLevel.xlsx'),'Max');
inputs.DayMean_byYear=xlsread(strcat(inputs.Station_name,'_DailyWaterLevel.xlsx'),'Mean');
cd ..
Years_number=size(inputs.DayMax_byYear,2); % Number of Years with water level data

% Mean and Maxium daily water level data
Day_max_Observed=inputs.DayMax_byYear(:);
Day_max_Observed(isnan(Day_max_Observed))=[]; % Removing NaN data

day_mean_Observed=inputs.DayMean_byYear(:);
day_mean_Observed(isnan(day_mean_Observed))=[]; % Removing NaN data
```

Fitting maximum daily data to Normal distribution

```
pd=fit((1:length(Day_max_Observed))',Day_max_Observed,'poly1');
Y=pd.p1.*(1:length(Day_max_Observed)); % Daily Mean Sea Level
day_mean_Observed_Detrend=Day_max_Observed-Y';
[mu,sigma]=normfit(Day_max_Observed);
[mu2,sigma2]=normfit(day_mean_Observed_Detrend);  %#ok<*ASGLU>
```

Quantile Regresion

```
dayss=(1:size(Day_max_Observed(:),1));
% linear regression of mean daily sea level
Lin_fit=fit(dayss',day_mean_Observed(:),'poly1');
Lin_fit_fun=@(x) Lin_fit.p1*x+Lin_fit.p2;

Daily_mean_Annual_no_noise=Lin_fit_fun(dayss');
Daily_max_Observed=Day_max_Observed(:);
% calculating the GPD variable threshold as the 97% quantile regression
[p_QR,~]=quantreg(Daily_mean_Annual_no_noise,Daily_max_Observed,.97,1);clc;
Y_Hat_equ= @(x) p_QR(1) .* x + p_QR(2);
Y_Hat=[];
for i = 1 : size(Daily_max_Observed,1)
    Y_Hat(i)=Y_Hat_equ(Daily_mean_Annual_no_noise(i));
end
```

```
index=find(Daily_max_Observed>Y_Hat');
```

Declustring

Finding the independent extremes

```
for kk=1 : 10
    for w= 1 : size (index,1)-1
        if index(w+1)<index(w)+3
            aa=Daily_max_Observed(index(w));
            bb=Daily_max_Observed(index(w+1));
            if aa> bb
                index(w+1)=index(w);
            else
                index(w)=index(w+1);
            end
        end
    end
end
index=unique(index);
excess=Daily_max_Observed(index);
```

Estimating parameters of GPD

Estimating the probability of independent exceedances over threshold

```
Rate=size(excess,1)./Years_number/365.25;
% Estimating GPD parameters
POT=excess-Y_Hat_equ(Daily_mean_Annual_no_noise(index));
[pd_Var,~]=gpfit(POT);

% Plotting empirical and simulated CDF
y1=linspace(0,1,1000);
MSL1=nanmean(Daily_mean_Annual_no_noise(end-2013:end-1647));
Threshold=p_QR(1) .* MSL1 + p_QR(2);
hh1=figure(1);
set(hh1,'visible','off');

functionNomrGPD = @(x) cdfNormGPD(Rate,Threshold,pd_Var(1),pd_Var(2),mu,sigma,x);
w = warning ('on','all');
id = w.identifier;
warning('off',id)
GPD_CDF=fplot(functionNomrGPD,[min(Daily_max_Observed)-.1 max(Daily_max_Observed)+.1]);
set(GPD_CDF,'linewidth',2,'color','r','LineStyle','-')
hold on
ecdf1=cdfplot(Daily_max_Observed);
set(ecdf1,'linewidth',2,'color','b','LineStyle','--')
thr=plot([Threshold-0.2 Threshold-.2],[0 1],'-.k','linewidth',1.5);
xlim([min(Daily_max_Observed)+.3 max(Daily_max_Observed)+.1]);
grid on
xlabel('Level of Water [ft,MHHW]')
h_Constant=[ecdf1,GPD_CDF,thr];
leg1=legend(h_Constant,'Emperical CDF','Mixture-Model CDF','GPD Threshold');
% Saving pirical and simulated CDF figure
x0=6;
y0=4;
width=6;
height=3;
set(gcf,'units','inches','position',[x0,y0,width,height])
title(inputs.Station_name)
set(leg1,'fontsize',9,'EdgeColor',[0.8 0.8 0.8]);

if ~exist('Outputs','dir')
    mkdir('Outputs')
end

if ispc
    pwd1= fullfile(pwd,'\Outputs');
else
    pwd1= fullfile(pwd,'/Outputs');
end

print(hh1,fullfile(pwd1 , [inputs.Station_name '_CDF.jpeg']),'-djpeg','-r600'); clc;
```

Return Level-Retun Period Plot with 95% confidence intervals for GPD component (delta method)

```
Var_Rate=Rate.*(1-Rate)./length(Daily_max_Observed);
[~,acov] = gplike(pd_Var,POT);

nn=0;
for N1 = 0.1: 0.01 :1000 % Daily Exceedance Probability
    nn=nn+1;
    N=N1.*365.25; % Annual Exceedance PRobability

    Zn1=((Threshold)+pd_Var(2)/pd_Var(1).*((N.*Rate).^pd_Var(1)-1)); % Return Level
    if Zn1>Threshold
        Zn(1,nn)=Zn1;
    else
        Zn(1,nn)=NaN;
    end
    % 95% confidence intervals for GPD component (Delta Method)
    Delta_T=[pd_Var(2).*(N.^pd_Var(1)),(1./pd_Var(1)).*((N.*Rate).^pd_Var(1)-1),...
        -pd_Var(2).*(pd_Var(1).^-2).*((N.*Rate).^pd_Var(1)-1)+pd_Var(2).*...
        (pd_Var(1).^-1).*(N.*Rate).^pd_Var(1).*log(N.*Rate)];
    Delta=transpose(Delta_T);
    V=[Var_Rate,0,0;0,acov(2,2),acov(1,2);0,acov(1,2),acov(1,1)];
    Var_xm(nn)=Delta_T*V*Delta;
    CI_U(nn)=Zn(1,nn)+1.96.*sqrt(Var_xm(nn));
    CI_L(nn)=Zn(1,nn)-1.96.*sqrt(Var_xm(nn));
    RP(nn)=N1;
end

% Estimatin Return level return period under different sea level rise values
ctr=0;
hh2=figure(2);
set(hh2,'visible','off');
for MSL=(MSL1:0.1:MSL1+2)
    clear RP2 WL
    ctr=ctr+1;
    Threshold(ctr)=p_QR(1) .* MSL + p_QR(2); % GPD variable threshold

    % when GPD shape is negative the distribution is thin?tailed and has upper limit
    if pd_Var(1)<0
        Up_Limit=Threshold(ctr)-(pd_Var(2)/pd_Var(1));
    else
        Up_Limit=20;
    end

    nn=0;
    % Return period estimation
    for wl= -.5 :0.01:Up_Limit
        nn=nn+1;
        if wl>Threshold(ctr)
            RP2(nn)=1/((Rate.*365.25.* (1+pd_Var(1).*((wl-Threshold(ctr)))./pd_Var(2)).^(-1/pd_Var(1))));
        else
            SLR=MSL-MSL1;
            Normal_Coef2=(1-Rate)/normcdf(Threshold(ctr),mu+SLR,sigma);
            RP2(nn)= 1/((1- Normal_Coef2*normcdf(wl,mu+SLR,sigma))*365.25);
        end
        WL(nn)=wl ;
    end

    % Emperical CDF
    [F,X]=ecdf(Daily_max_Observed);
    FF=1./((1-F).*(length(Daily_max_Observed)/Years_number));


    xlim([0.003 1000])
    ylim([0 nanmax(Zn(1,:))+3])
    set(gca, 'XScale', 'log')
    if SLR==0
        semilogx( RP, Zn(1,:),'color',[1 1 1],'linewidth',2);
        hold on

        h1=semilogx(RP2,WL,'m','linewidth',2);
        hold on
        CI_Up=semilogx(RP,CI_U,'--r');

        CI_Low=semilogx(RP,CI_L,'--r'); %#ok<NASGU>

        index=find(WL==0);
        % Plotting return level interval using empirical CDF
        Picks=semilogx(FF(index:end),X(index:end),'o','markerfacecolor','k','markeredgecolor','k','markersize',1.5);
    elseif SLR==1
        h2= semilogx(RP2,WL,'b','linewidth',2);
    end
    grid on
```

```
    hold on
    % Return period Return Level under SLR 0 to 2 ft
    Return_Period{1,ctr}(:,1)=RP2'; %#ok<AGROW>
    Return_Period{1,ctr}(:,2)=WL'; %#ok<AGROW>

    % Selected Return period
    kk1=(0.1:0.1:1);
    kk2=(0:10:1000);kk2(1)=[];
    kk=[kk1,kk2];
    % Selected Return period Return Level under SLR 0 to 2 ft
    RPRL_Final{1,ctr}(:,2)=interp1(Return_Period{1,ctr}(:,1),Return_Period{1,ctr}(:,2),kk');
    RPRL_Final{1,ctr}(:,1)=kk';

    % Minor and major flood frequency
    if inputs.Minor_Th>Threshold(ctr)
        AP_Minor(ctr)=((Rate.*365.25.* (1+pd_Var(1).*((inputs.Minor_Th-Threshold(ctr)))./pd_Var(2)).^(-1/pd_Var(1))));
    else

        Normal_Coef2=(1-Rate)/normcdf(Threshold(ctr),mu+SLR,sigma);
        AP_Minor(ctr)= ((1- Normal_Coef2*normcdf(inputs.Minor_Th,mu+SLR,sigma))*365.25);
    end

    if inputs.Major_Th>Threshold(ctr)
        RP_Major(ctr)=1./((Rate.*365.25.* (1+pd_Var(1).*((inputs.Major_Th-Threshold(ctr)))./pd_Var(2)).^(-1/pd_Var(1))));
    else

        Normal_Coef2=(1-Rate)/normcdf(Threshold(ctr),mu+SLR,sigma);
        RP_Major(ctr)= 1./((1- Normal_Coef2*normcdf(inputs.Major_Th,mu+SLR,sigma))*365.25);
    end
end
AP_Minor=round(AP_Minor,2);
RP_Major=round(RP_Major,2);
xlim([0.003 1000])
ylim([0 nanmax(Zn(1,:))+3])
h_Minor=plot([0.003 1000],[inputs.Minor_Th inputs.Minor_Th],'g');
hh=[h1,h2,Picks,CI_Up,h_Minor];
leg=legend(hh,'Current Annual MSL','1 ft SLR','Observed Data','95% CI','Minor Flood Threshold','location','northwest');
xlabel('Return Period [Year]')
ylabel('Ruturn Level [ft,MHHW]')
set(gca,'xtick',[0.01 1 100 1000])
% Set the dimension of Return period return level figure
x0=6;
y0=4;
width=6;
height=3;
set(gcf,'units','inches','position',[x0,y0,width,height])
title(inputs.Station_name)
set(leg,'fontsize',9,'EdgeColor',[0.8 0.8 0.8]);

if ~exist('Outputs','dir')
    mkdir('Outputs')
end

if ispc
    pwd2= fullfile(pwd,'\Outputs');
else
    pwd2= fullfile(pwd,'/Outputs');
end
print(gcf,fullfile(pwd2 , [inputs.Station_name '_RPRL.jpeg']),'-djpeg','-r600')
```

Displaying results

```
RP_Des=inputs.Return_Period;
ctr3=0;
for kkk=RP_Des
    ctr3=ctr3+1;
    Level_des(ctr3)= RPRL_Final{1,1}((RPRL_Final{1,1}(:,1)==kkk),2);
    % estimatin reteurn period of current ??-year flood under SLR 0 to 2 ft
    SLR=0;
    for jj= 2: length(Threshold)
        SLR=SLR+0.1;
        if Level_des(ctr3)>(Threshold(jj))
            RPP1(ctr3,jj)= 1./(Rate.*365.25.* (1+pd_Var(1).*((Level_des(ctr3)-(Threshold(jj)))./pd_Var(2))).^(-1/pd_Var(1)));
        else
            Normal_Coef2=(1-Rate)/normcdf(Threshold(jj),mu+SLR,sigma);
            RPP1(ctr3,jj)= 1./((1- Normal_Coef2*normcdf(Level_des(ctr3),mu+SLR,sigma))*365.25);

        end
        RPP1(ctr3,1)=kkk;
    end
```

```
end
T_Variable_Name={'SLR0', 'SLR0_1', 'SLR0_2','SLR0_3','SLR0_4','SLR0_5','SLR0_6',...
    'SLR0_7','SLR0_8','SLR0_9','SLR1','SLR1_1','SLR1_2','SLR1_3','SLR1_4','SLR1_5','SLR1_6','SLR1_7',...
    'SLR1_8','SLR1_9','SLR2'};



T_Future_RP=table(RPP1(:,2),RPP1(:,3),RPP1(:,4),RPP1(:,5),RPP1(:,6),RPP1(:,7),RPP1(:,8),...
    RPP1(:,9),RPP1(:,10),RPP1(:,11),RPP1(:,12),RPP1(:,13),RPP1(:,14),RPP1(:,15),RPP1(:,16),...
    RPP1(:,17),RPP1(:,18),RPP1(:,19),RPP1(:,20),RPP1(:,21),'RowNames',cellstr(strcat(string(RP_Des'),string(repmat('_year',length(RP_Des),1)))));
T_Future_RP.Properties.VariableNames=T_Variable_Name(2:end);

Sc={'Major Flood Return Period [Yr]';'Minor Flood Annual Frequncy [Day/Yr]'};
Minor_Major=[RP_Major;AP_Minor];
T_Minor_Major=table(Minor_Major(:,1),Minor_Major(:,2),Minor_Major(:,3),Minor_Major(:,4),Minor_Major(:,5),...
    Minor_Major(:,6),Minor_Major(:,7),Minor_Major(:,8),Minor_Major(:,9),Minor_Major(:,10),...
    Minor_Major(:,11),Minor_Major(:,12),Minor_Major(:,13),Minor_Major(:,14),Minor_Major(:,15),...
    Minor_Major(:,16),Minor_Major(:,17),Minor_Major(:,18),Minor_Major(:,19),Minor_Major(:,20),...
    Minor_Major(:,21),'RowNames',Sc);
T_Minor_Major.Properties.VariableNames=T_Variable_Name;

T_RPRL=table(RPRL_Final{1,1}(:,2),RPRL_Final{1,2}(:,2),RPRL_Final{1,3}(:,2),RPRL_Final{1,4}(:,2),...
    RPRL_Final{1,5}(:,2),RPRL_Final{1,6}(:,2),RPRL_Final{1,7}(:,2),RPRL_Final{1,8}(:,2),...
    RPRL_Final{1,9}(:,2),RPRL_Final{1,10}(:,2),RPRL_Final{1,11}(:,2),RPRL_Final{1,12}(:,2),...
    RPRL_Final{1,13}(:,2),RPRL_Final{1,14}(:,2),RPRL_Final{1,15}(:,2),RPRL_Final{1,16}(:,2),...
    RPRL_Final{1,17}(:,2),RPRL_Final{1,18}(:,2),RPRL_Final{1,19}(:,2),RPRL_Final{1,20}(:,2),...
    RPRL_Final{1,21}(:,2),'RowNames',cellstr(strcat(string(kk'),string(repmat('_year',length(kk'),1)))));
T_RPRL.Properties.VariableNames=T_Variable_Name;

T_RPRL.Properties.Description = 'Return Period-Return Level curve';
clear Station_name inputs
```

Published with MATLAB® R2017b