Homework 11 Quantitative Risk Management

Group G03
Jakob Amaya Scott, Martin Hammerues, Albin Henriksson, Markus Trætli

December 22, 2022

Question 1

We have

$$F_v(x) = \mathbf{P}(X - v < x | X > v) = 1 - \mathbf{P}(X - v > x | X > v) = 1 - \frac{\mathbf{P}(X - v > x)}{\mathbf{P}(X > v)} =$$

$$= 1 - \frac{\mathbf{P}(X - u > x + v - u)}{\mathbf{P}(X > u)} \frac{\mathbf{P}(X > u)}{\mathbf{P}(X - u > v - u)} = 1 - \frac{\mathbf{P}(X - u > x + v - u | X > u)}{\mathbf{P}(X - u > v - u | X > u)} =$$

$$= 1 - \frac{1 - F_u(x + v - u)}{1 - F_u(v - u)}.$$

To proceed, we look at page 22 of lecture 9 to find the expression for the generalized pareto distribution. We first consider the case $\xi \neq 0$:

$$F_v(x) = 1 - \frac{1 - F_u(x + v - u)}{1 - F_u(v - u)} = 1 - \frac{\left(1 + \frac{\xi(x+v-u)}{\beta}\right)^{-\frac{1}{\xi}}}{\left(1 + \frac{\xi(v-u)}{\beta}\right)^{-\frac{1}{\xi}}} = 1 - \left(\frac{\frac{\beta+\xi(x+v-u)}{\beta}}{\frac{\beta+\xi(v-u)}{\beta}}\right)^{-\frac{1}{\xi}} =$$

$$= 1 - \left(\frac{\beta + \xi(v - u) + \xi x}{\beta + \xi(v - u)}\right)^{-\frac{1}{\xi}} = 1 - \left(1 + \frac{\xi x}{\beta + \xi(v - u)}\right)^{-\frac{1}{\xi}} = G_{\xi, \beta + (v-u)\xi}.$$

We now consider the case $\xi = 0$:

$$F_v(x) = 1 - \frac{1 - F_u(x + v - u)}{1 - F_u(v - u)} = 1 - \frac{e^{-\frac{x+v-u}{\beta}}}{e^{-\frac{v-u}{\beta}}} = 1 - e^{-\frac{x}{\beta}} = G_{\xi,\beta} = G_{\xi,\beta+(v-u)\cdot 0} = G_{\xi,\beta+(v-u)\xi}.$$

We see that the formula is valid in both cases.

Question 2

Mean excess

Figure 1 is the Sample Mean Excess Plot. It is calculated using the formula

$$e_n(v) = \frac{\sum_{i=1}^{n}(X_i - v)\mathbb{1}_{X_i > v}}{\sum_{i=1}^{n}\mathbb{1}_{X_i > v}}$$

where X is the negative log returns of Microsoft in the period November 26th 2011 to November 26th 2016. The points are $(X_i, e_n(X_i))$ for $0 < X_i < 0.04$.

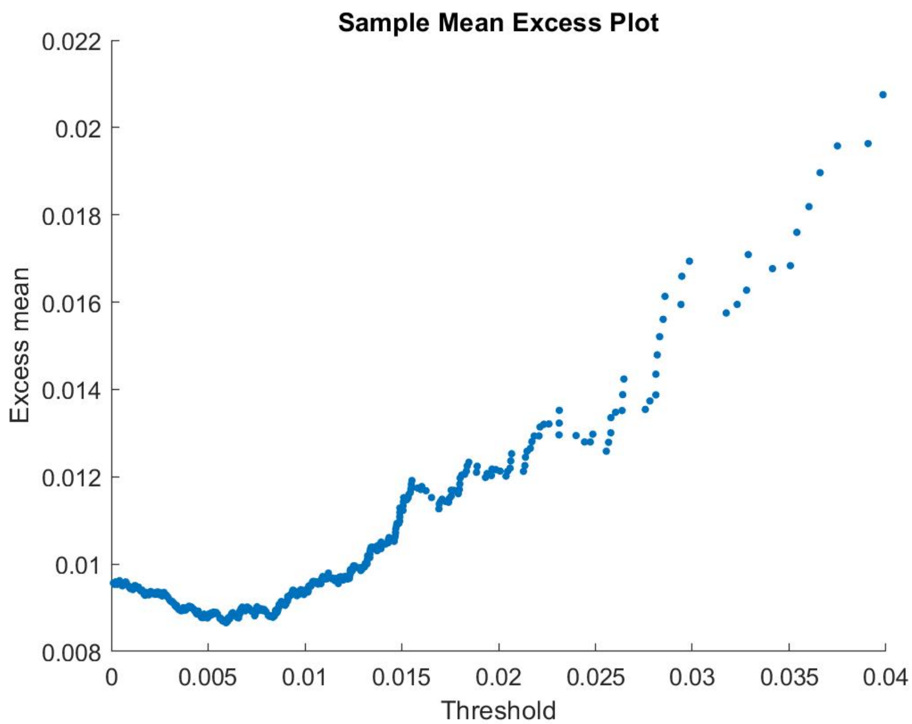

Figure 1: Sample Mean Excess Plot

CDF

A reproduction of the plot seen on page 33 of lecture 9 can be seen in figure 2. The optimal parameters were found to be

$$(\xi, \beta) = (0.0072, 0.2251)$$

which resulted in a log-likelihood of 785.9. The optimization problem was solved by using the Matlab function fmincon with $-\log(L(\xi, \beta; Y_1, ..., Y_{n_u}))$ as the target function where

$$L(\xi, \beta; Y_1, ..., Y_{n_u}) = \Pi_{i=1}^{n_u} g_{\xi,\beta}(Y_i)$$

and $g_{\xi,\beta}$ is the GPD pdf.

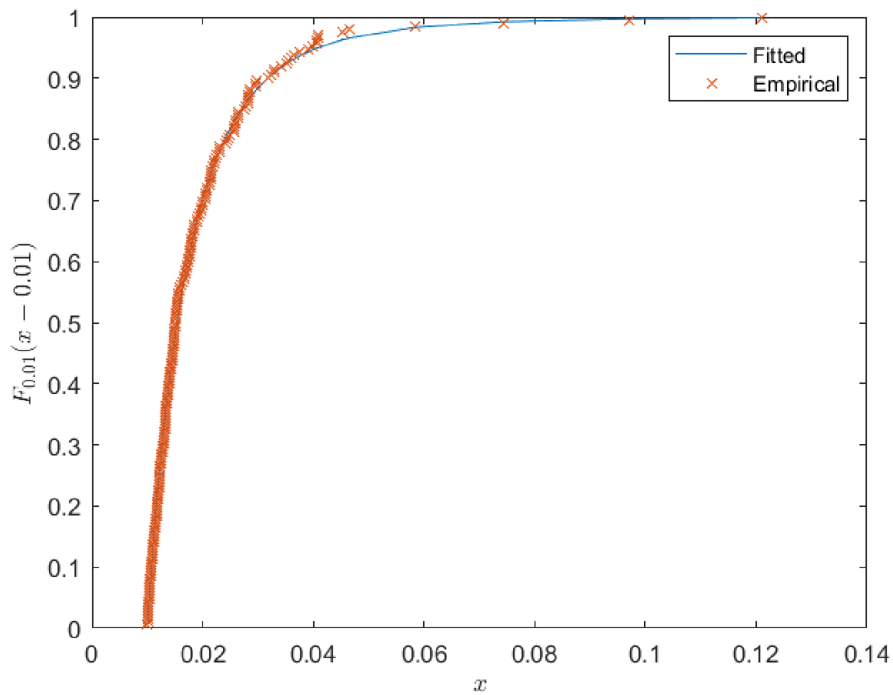

Figure 2: Fitted and non-parametric CDF

Question 3

One thing we can observe in Figure 1 in the Culp, Gandhi, Nozawa and Veronesi paper is that the Implied Spread goes to infinity for options that are in the money when the time to maturity approaches 0. This behavior is not captured in Figure 3 since T is fixed at 0.25. The normalized implied spread is increasing as time to maturity increases, and does not go towards infinity which is an advantage to the IS. This behavior is not captured in Figure 4 for the same reason as for the IS, that is, T is fxied and only the behavior against the moneyness plane is captured in the figure.

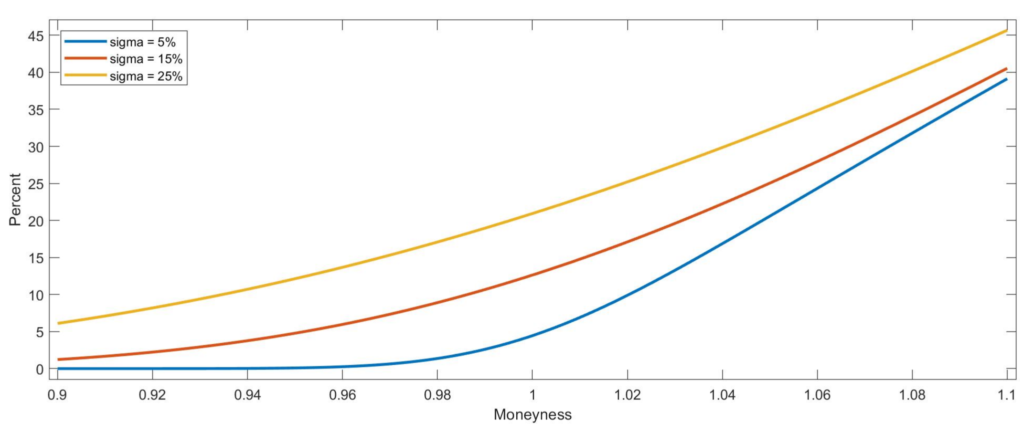

Figure 3: Implied Spread

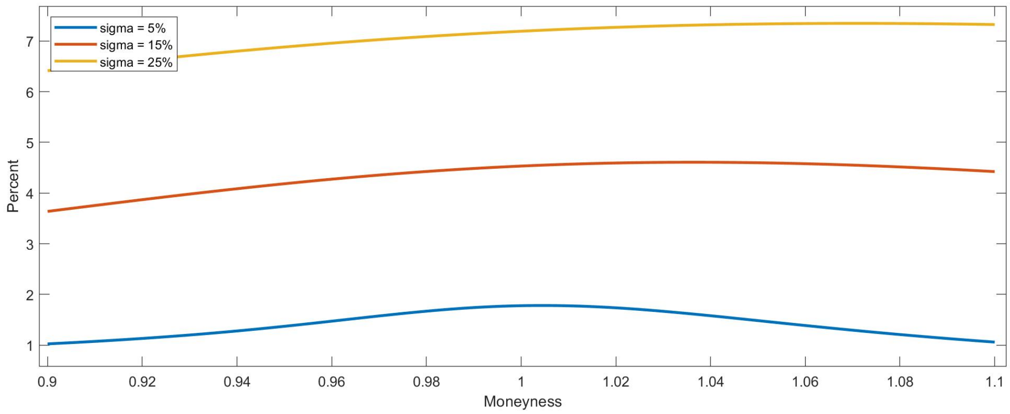

Figure 4: Normalized Implied Spread

Code

Q2

Mean excess part

```
%Assignment 11 Q2 a)
table = readmatrix('MSFT_stock_data.csv');
stock_prices = table(:,3);
neg_log_ret = -log(stock_prices(2:size(stock_prices,1))./stock_prices(1:size(
    stock_prices,1)-1));

thresh=[];
for i=1:size(neg_log_ret,1);
    if neg_log_ret(i) > 0 & neg_log_ret(i) < 0.04
        thresh(end+1) = neg_log_ret(i);
    end
end

mean_excess = zeros(1,size(thresh,2));
for i=1:size(thresh,2)
    %Takes out the values greater than threshold
    log_ret_threshold = neg_log_ret(neg_log_ret(:,1) > thresh(i), :);

    mean_excess(i) = sum(log_ret_threshold-thresh(i))/size(log_ret_threshold
        ,1);
end

scatter(thresh,mean_excess,10,'filled')
title('Sample Mean Excess Plot')
xlabel('Threshold')
ylabel('Excess mean')
```

CDF part

```
clear all
close all

```

```
table = readtable('MSFT.csv','PreserveVariableNames',true);

% Let us consider the closing prices
closing_prices=table2array(table(:,5));

% Compute loss (negative log returns)
loss=-diff(log(closing_prices));

% Filter the loss such that the resulting vector only contains losses of at least 0.01
threshold=.01;
loss_over_threshold=nonzeros((loss).*(loss>=threshold));
loss_over_threshold=sort(loss_over_threshold);
excess=loss_over_threshold-threshold;

% Define functions for the GDP PDF, CDF and the negative log likelihood, which will be minimized
fun_GPD_pdf=@(v) 1/v(2)*(1+v(1)*excess/v(2)).^(-1/v(1)-1);
fun_GPD=@(v) 1-(1+excess*v(1)/v(2)).^(-1/v(1));
log_likelihood=@(v)-sum(log(fun_GPD_pdf(v)));

% Find optimal parameters and compute the CDF
[params,y]=fmincon(log_likelihood,[0.1,1]);
GPD_CDF=fun_GPD(params);

% Create non-parametric CDF
empiric_CDF=(1:length(excess))/length(excess);

% Reproduce lecture plot
figure
plot(loss_over_threshold,GPD_CDF)
hold on
scatter(loss_over_threshold,empiric_CDF,'x')
legend('Fitted','Empirical')
```

Q3

```
%Assignment 11 Q3
sigma1 = 0.05;
sigma2 = 0.15;
sigma3 = 0.25;
M = 0.9:0.001:1.1;

%Implied Spread
IS1 = ImpliedSpread(sigma1,M);
IS2 = ImpliedSpread(sigma2,M);
IS3 = ImpliedSpread(sigma3,M);

%Normalized Implied Spread
NIS1 = NormalizedImpliedSpread(sigma1,M);
NIS2 = NormalizedImpliedSpread(sigma2,M);
NIS3 = NormalizedImpliedSpread(sigma3,M);


```

```
%Plot Implied Spread
plot(M,IS1.*100,'LineWidth',2.0)
hold on
plot(M,IS2.*100,'LineWidth',2.0)
plot(M,IS3.*100,'LineWidth',2.0)
xlabel("Moneyness")
ylabel("Percent")
legend({'sigma = 5%', 'sigma = 15%', 'sigma = 25%'},'Location','northwest')
hold off

%Plot Normalized Implied Spread
plot(M,NIS1.*100,'LineWidth',2.0)
hold on
plot(M,NIS2.*100,'LineWidth',2.0)
plot(M,NIS3.*100,'LineWidth',2.0)
xlabel("Moneyness")
ylabel("Percent")
legend({'sigma = 5%', 'sigma = 15%', 'sigma = 25%'},'Location','northwest')
hold off
```

```
function [IS] = ImpliedSpread(sigma,M)
%Returns the implied spread

%Constants
r = 0.03;
delta = 0.02;
T = 0.25;

%a = S/K
a = 1./(M.*exp((r-delta)*T));

d1 = (log(1./M) + (r - delta + 0.5*sigma^2)*T)./(sigma*sqrt(T));
d2 = d1 - sigma*sqrt(T);

IS = -(1/T).*log(1-normcdf(-d2)+(normcdf(-d1).*a));
end
```

```
function [NIS] = NormalizedImpliedSpread(sigma,M)
%Returns the normalized implied spread

%Constants
r = 0.03;
delta = 0.02;
T = 0.25;

%a = S/K
a = 1./(M.*exp((r-delta)*T));

d1 = (log(1./M) + (r - delta + 0.5*sigma^2)*T)./(sigma*sqrt(T));
d2 = d1 - sigma*sqrt(T);

NIS = log(1-normcdf(-d2)+(normcdf(-d1).*a))./log(1-normcdf(-d2));
end
```