```matlab
% Lorenzo Guglielmetti
% EURO vs Colombian Peso (EURCOP)

load('EURCOP.mat')
converted_dates = datestr(mx_dates); %to convert dates from matlab format to
string.

%% 1)
figure
plot(mx_dates, mx_quote)
title('Lorenzo Guglielmetti EURCOP')
grid on
xlabel('Dates')
ylabel('Quotes')
datetick('x', 'dd mmm YYYY')

%% 2)
figure
histfit(mx_quote)
xlabel('Quotes')
ylabel('Frequency of quotes')
title('Distribution of the historical FX vs Normal')

figure
histfit(mx_quote, [],'lognormal')
xlabel('Quotes')
ylabel('Frequency of quotes')
title('Distribution of the historical FX vs LogNormal')

% Here we can compare our distribution of the historical series with the
% normal (1st graph) and lognormal distribution (2nd graph), through the
% command "histfit" that allow us to see on the x-axis the values that the
% exchange rate has assumed over the period and on y-axis the frequencies
% with which the exchange rate has assumed that values.

% CONCLUSION: Comparing the 2 plots we can see that the Normal distribution
% fits better the data than the LogNormal. Indeed it seems that the real
% distribution is a little bit positively skewed, which means that there is
% a higher probability of obtaining positive values than negative ones.

%% 3)
Mytitle= 'Distribution of the yearly FX vs Normal';
Mytitle2= 'Distribution of the yearly FX vs LogNormal';
for ii=2013:2021
    myBoolean = year(mx_dates) == ii;
    yearly_quotes = mx_quote(myBoolean);
    figure
    histfit(yearly_quotes)
    xlabel('Quotes')
    ylabel('Frequency of quotes')
    title([Mytitle ' ' num2str(ii)])
    figure
    histfit(yearly_quotes, [],'lognormal')
    xlabel('Quotes')
    ylabel('Frequency of quotes')
```

```
    title([Mytitle2 ' ' num2str(ii)])
end

% In this part I replicated point 2 for each year of the historical series,
% through a for-cycle. I created the variable myBoolean (that gives 1 for
% the years selected, and 0 for the other ones) and using the command year(),
% given ii=2013:2021,I was able to select each year from all the data.
% Then I selected the quotes for the year indicated by myBoolean and I
% plotted these data using histfit(), where I have on the x-axis all the
% quotes for the exchange rate in the year that I selected, and on the
% y-axis the frequency of quotes (of the year selected). I've done it both
% for the Normal distribution and for the LogNormal one.
% For the title I created the two variables Mytitle (for the Normal) and
% Mytitle2 (for the LogNormal), then I insert in the for-cycle, title, the
% function num2str that allow me to transform the numbers of the years into
% strings and to add it in the title.

%% 4)
aa = log(mx_quote);
LogReturns = diff(aa);
mu = (mean(LogReturns))*260;
sigma = (std(LogReturns))*sqrt(260);

% After calculating the logarithm of the exchange rates, I calculated the
% first difference and I took the mean (mu) and the standard deviation
% (sigma) from the mean.
% To calibrate the GBM we have to remember that mean and standard deviation
% are calculated on a daily basis, so are both daily mean and daily
% standard deviation; if we want to find yearly mean and  yearly standard
% deviation, we have to multiply the mean term by 260 (that are the
% working/trading days in a year) and the standard deviation by the square
% root of 260, so we have to switch from a daily basis to a yealry basis
% because the GBM function is expecting yearly parameters.

%% 5)
for ii=2013:2021
    myBoolean_years = year(mx_dates) == ii;
    yearly_quotes_years = mx_quote(myBoolean_years);
    aa_years = log(yearly_quotes_years);
    LogReturns_years = diff(aa_years);
    MeanArray(ii) = (mean(LogReturns_years))*260;
end
Mean_Array_years = nonzeros(MeanArray)';
%or
Mean9=MeanArray(1,2013:2021);

for ii=2013:2021
    myBoolean_years = year(mx_dates) == ii;
    yearly_quotes_years = mx_quote(myBoolean_years);
    aa_years = log(yearly_quotes_years);
    LogReturns_years = diff(aa_years);
    sigmaArray(ii) = std(LogReturns_years);
end
sigma_Array_years = nonzeros(sigmaArray)';
%or
```

```
Sigma9=sigmaArray(1,2013:2021);

% In this part I used a for cycle to compute the mean and the standard
% deviation of the LogReturns for each year considered.
% I have used a myBollean to make all the years that are not in out interval
% equal to 0. After I have calibrated the GBM parameters and finallly with
% nonzeros I was able to eliminate all the zeros from the mean and sigma
% arrays.
% An alternative way to eliminate the zeros is creating an array in which we
% take the first row and a number of columns equals to the number of the
% years selected and this is what I do in Mean9 and Sigma9.

%% 6.a)

%GBM simulation with parameters calculated from the whole period (point 4):
n_simulation    = 10000;
n_T             = 260;
f_dt            = 1/260;
f_mu            = mu;
f_sigma         = sigma;
f_quote         = mx_quote(end);
S_wholeperiod  = GBM_simulation(n_simulation ,n_T, f_dt ,f_mu ,f_sigma ,f_quote);

% In this part I simulated the Geometric Brownian with 10000 simulations,
% with a time period equal to 1 year (time horizont), with a daily
% frequency (f_dt=1/260) and using yearly mean and standard deviation
% computed for the whole period. Finally I used the last available price
% from the data as starting point.
% f_dt is the time step we want to use and it is always reported in terms
% of years; so, if we want an interval that is less than an year, for
% example 6 months, we will write 0.5 or, like in this case in which we
% want a daily frequency I have written 1/260 where 260 is the number of
% days in one commercial year.
% Instead n_T, that is the time horizon of the analysis, it is reportend in
% terms of f_dt, so to take in consideration 1 year we need to write 260.

%% 6.b)

%GBM simulation with parameters calculated from each year (point 5):
for ii=2013:2021
    myBoolean = year(mx_dates) == ii;
    yearly_quotes = mx_quote(myBoolean);
    lastprice_array(ii)= yearly_quotes(end);
end
lastprice_array_a=lastprice_array(2013:2021);

% I created this for cycle to select the last available price for each year,
% to use it into the GBM as different starting point, and I put it into the
% lastprice_array_a.

for k=1:9
    n_simulation1     = 10000;
    n_T1              = 260;
    f_dt1             = 1/260;
    f_mu1             = Mean_Array_years(k);
```

```
    f_sigma1         = sigma_Array_years(k);
    f_quote1         = lastprice_array_a(k);
    S_annual  = GBM_simulation(n_simulation1 ,n_T1, f_dt1 ,f_mu1 ,f_sigma1
,f_quote1);
end

% Using a breakpoint in the S_annual line, we can see the GBM with 10000
% number of simulations, with a time period equal to 1, with daily frequency
% and using: mean and standard deviation computed for each year and as
% starting point the last available price for each year (through the array
% that I've previously created).

%% 7.a)

% Whole period
figure
histfit(S_wholeperiod(:,end))
title('Last simulated day GBM whole period')
ylabel('Last simulated day')
xlabel('Quotes')
grid on

% annual
figure
histfit(S_annual(:,end))
title('Last simulated day GBM annual')
ylabel('Last simulated day')
xlabel('Quotes')
grid on

%% 7.b)

% Whole period
figure
plot(S_wholeperiod')
title('Paths GBM whole period')
xlabel('Days')
ylabel('Quotes')
grid on

% annual
figure
plot(S_annual')
title('Paths GBM annual')
xlabel('Days')
ylabel('Quotes')
grid on

%% 7.c)

% From point 6.a
mean_GBMwholeperiod= mean(S_wholeperiod);
mean_GBMwhitoutfirst1= mean_GBMwholeperiod(2:261);
Ywhole=prctile(S_wholeperiod, [5 95]);
```

```matlab
% From point 6.b
mean_GBMannual= mean(S_annual);
mean_GBMwhitoutfirst2= mean_GBMannual(2:261);
Yannual=prctile(S_annual, [5 95]);

% I have calculated the average, 5∞ percentile and 95∞ percentile of the
% GBM both of the whole period and for each year.
% I have also calculate the averages without considering the first value so
% we can have an array with effectively 260 values.

%% 8)

for ii=2020
    myBoolean4 = year(mx_dates) == ii;
end
quotes2020 = mx_quote(myBoolean4);

ll = log(quotes2020);
LogReturns4 = diff(ll);
mu_2020 = (mean(LogReturns4))*260;
sigma_2020 = (std(LogReturns4))*sqrt(260);

n_simulation4    = 10000;
n_T4             = 260;
f_dt4            = 1/260;
f_mu4            = mu_2020;
f_sigma4         = sigma_2020;
f_quote4         = mx_quote(end);
S_2020  = GBM_simulation(n_simulation4 ,n_T4, f_dt4 ,f_mu4 ,f_sigma4 ,f_quote4);

mean_GBM2020= mean(S_2020);
Y2020=prctile(S_2020, [5 95]);

subplot(1,2,1);
plot(mean_GBMwholeperiod, Ywhole);
title('Mean, 5∞ and 95∞ percentiles of Whole Period')

subplot(1,2,2);
plot(mean_GBM2020, Y2020);
title('Mean, 5∞ and 95∞ percentiles of Year 2020')

% In the first part I have calibrated and simulated the GBM parameters on
% year 2020 because I needed them for the right plot. After I have studied
% the 'subplot' and I put in the left graph the mean, the 5∞ and the 95∞ of
% the Whole period and in the right graph I put the same information
% regarding the year 2020.
% The blue lines represent the 5∞ percentile and the orange lines represent
% the 95∞ percentile.

%% 9.c)
subplot(1,2,1);
mean_differences=(mu-Mean_Array_years);
plot(mean_differences)
bar(mean_differences)
title('Whole Period mean-Yearly mean')
```

```
xlabel('Years')
ylabel('Differences')
grid on

subplot(1,2,2);
mean_diff_percentage=(mu - Mean_Array_years)./Mean_Array_years;
plot(mean_diff_percentage)
bar(mean_diff_percentage)
title('Whole Period mean-Yearly mean')
xlabel('Years')
ylabel('Percentage differences')
grid on

subplot(1,2,1);
sigma_differences=(sigma-sigma_Array_years);
plot(sigma_differences)
bar(sigma_differences)
title('Whole Period volatility-Yearly volatility')
xlabel('Years')
ylabel('Differences')
grid on

subplot(1,2,2);
sigma_diff_percentage=(sigma - sigma_Array_years)./sigma_Array_years;
plot(sigma_diff_percentage)
bar(sigma_diff_percentage)
title('Whole Period volatility-Yearly volatility')
xlabel('Years')
ylabel('Percentage differences')
grid on

%% 9.dii)

aaaa = log(mx_quote);
LogReturns = diff(aaaa);
mu5 = (mean(LogReturns))*260;
sigma5 = (std(LogReturns))*sqrt(260);

n_simulation    = 10000;
n_Tm             = 12;
f_dtm            = 1/12;
f_mu            = mu5;
f_sigma         = sigma5;
f_quote         = mx_quote(end);
S_wholeperiodm  = GBM_simulation(n_simulation ,n_Tm, f_dtm ,f_mu ,f_sigma
,f_quote);


mean_GBMmonth= mean(S_wholeperiodm);
Y=prctile(S_wholeperiodm, [5 95]);
VAR_5p= (mean_GBMmonth-(Y(1,:)))./(mean_GBMmonth);
VAR_95p= (mean_GBMmonth-(Y(2,:)))./(mean_GBMmonth);
VAP_5p= ((Y(1,:))-mean_GBMmonth)./(mean_GBMmonth);
VAP_95p= ((Y(2,:))-mean_GBMmonth)./(mean_GBMmonth);
```

```
% I have calibrated and simulated a new GBM with a monthly frequency and a
% time period of one year.
% After I calulated the Value at Risk and the Value at Profit of the 5∞ and
% 95∞ percentiles.
```