Econ 672
Ziming Huang **Project 2** September 14, 2019

Exercise 1

(A) Interprets of parameters:

(i) We can use the function datevec to get [Y,M,D,H,MN,S] of our data by inputting our dates and the date form (the input dates need to be strings) :
datevec('dates','yyyymmdd hhmmss').

We can also get [Y,M,D,H,MN,S] by doing some calculations:
Y=floor(dates(:,1)/10000);
M=floor((dates(:,1)-Y*10000)/100);
D=dates(:,1)-10000/*Y-100/*M;
H=floor(data(:,2)/10000);
MN=floor(data(:,2)-H/*10000)/100;
S=data(:,2)-10000/*H-100/*MN.
Given the data we have, the second method will cost less time since we do not need to change our data form(double) to strings which we will need to do if we want to use the first method.

(ii) We can store prices in either a matrix or a vector. Comparing to vector, storing prices in matrix makes it possible for us to group this data by day, which will be helpful for avoiding calculate overnight returns. What's more, store changing from matrix to vector is very simple by using the code: matrix(:).

The **MATLAB** code:

```matlab
function [dates,log_prices] = load_stock(file_name,type)
%file_name: the name of data, is a string
%type: the minimal unit of observation frequency
%m: minutes
%s: seconds
data=csvread(file_name);

Y=floor(data(:,1)/10000); % year
M=floor((data(:,1)-Y*10000)/100); %month
D=data(:,1)-10000*Y-100*M; % day

if type=='m'
    H=floor(data(:,2)/100); %hour
    MN=data(:,2)-H*100; %minute
    S=0; %second
elseif type=='s'
```

```
    H=floor(data(:,2)/10000); %hour
    MN=floor(data(:,2)-H*10000)/100; %minute
    S=data(:,2)-10000*H-100*MN; %second
end

dates=datenum(Y,M,D,H,MN,S);
log_prices=log(data(:,3));

end
```

(B) From our data, we have:

$$
\begin{aligned}
T_{PG} &= 2769 \qquad N_{PG} = 78 \\
T_{DIS} &= 2769 \qquad N_{DIS} = 78
\end{aligned}
$$

We can find the value of N by count the number of values that equal to the first value in first column; We can find the value of T by dividing the row size of our dates into N:

N=sum(dates(1,1)==dates(:,1));

n=N-1;

T=size(dates,1)/N;

(C) (i) The regular market time is from 09:30 a.m. to 4:00 p.m., so the market hour is 6.5 hours.

(ii) If the data is sampled every 5 minutes, that is $\Delta_n = 5minutes$, then the observation numbers every day is $N = \frac{1}{\Delta_n} + 1 = 79$.

(iii) The total number of observation is differ from the value in part(B);

(iv) The reason of these differencse can be that since the open price may contain large noises, this will have a big influence on our calculation (especially for high frequency data). In order to cancel this effects, we always let out the open price from our data.

(D) We can avoid calculating overnight returns by reshape our prices into group (by day); we can delete the date time of the first observation each day to construct a new date time matrix which is right for our log-return data.

To extent the application of the function log_return, I have added two other parameters:

N: is the number of observations our data provided every data. If N=78, then we will have 78 data every day;

J: is the number of steps(intervals) between every used observation. If J=1, means we will use all our data to do analysis; if J=10, means we will just pick 1th,11th,21th... observation data to do analysis.

The outputs of this function are return_dates and log_return(not in percentage), they are grouped by day and shown in matrix.

The **MATLAB** code:

```
function [return_dates,log_returns] = log_return(data,N,J)

%data is a 2 column matrix where the first column is date,
%the second column is the log-price
%data=[dates, log-price]
%N is the number of observations every day
%J is the number of steps between observations
%output are return_dates and log_return: they are group by day and shown in matrix
return_dates=reshape(data(:,1),N,[]); %reshape by day group
return_dates=return_dates(1:J:end,:); %pick up useful observations
return_dates(1,:)=[];

log_prices=reshape(data(:,2),N,[]); %reshape by day group
log_prices=log_prices(1:J:end,:); %pick up useful observations
log_returns=diff(log_prices);

end
```

(E) We first need to transfer the outputs of function log_return into vectors by using the code: matrix(:), and then convert the log-return to percentage. The follows are the figures of price and log-return for PG and DIS:

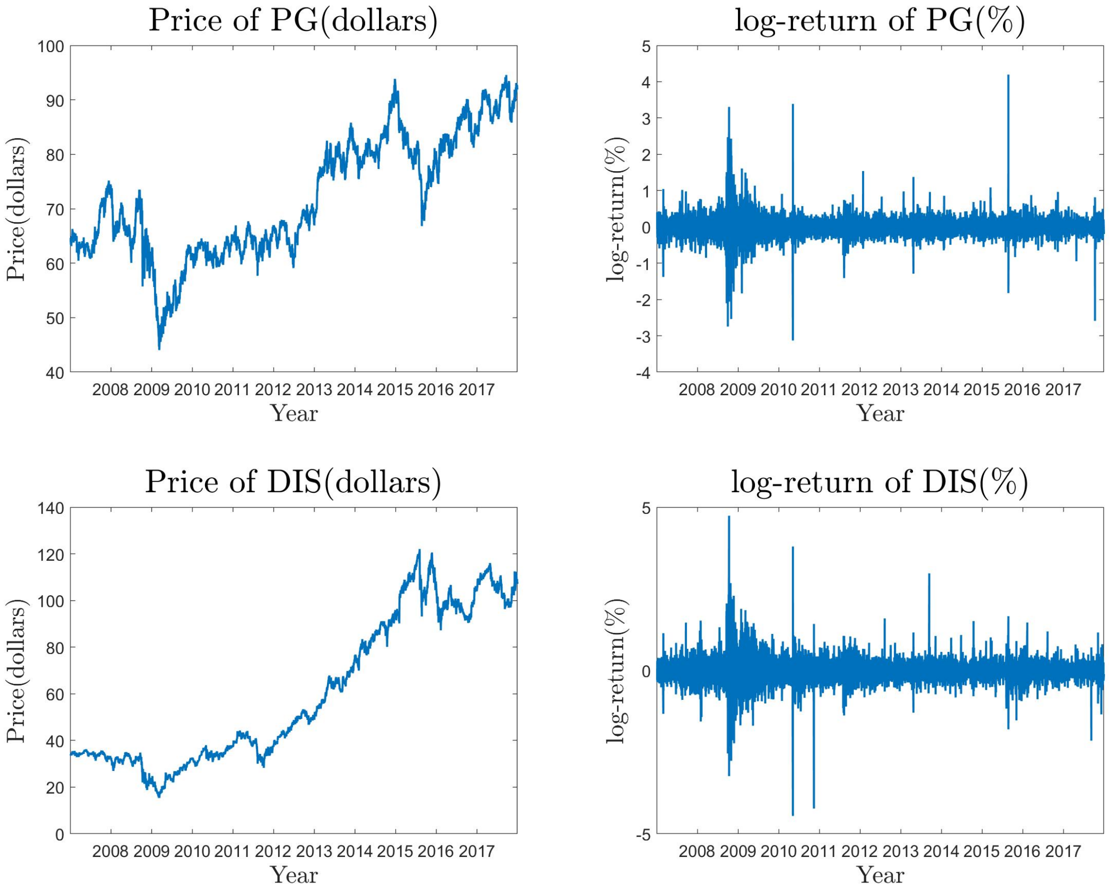

Figure 1: Price and log-return

From the price figure, we can know the range of price for PG and DIS are (44\$,95\$) and (18\$,120\$) respectively. Even thought the price range for DIS' price is larger, PG's price shows more volatility. As we can see, PG's price went through a big fluctuation around 2009, median 2014 and median 2015.

From the log-return figures, we can find the both the log-returns of PG and DIS move around with 0. But we can also find that there are some abnormal values in both the PG and DIS' log-returns. For PG, it has three abnormal return: around 2009, median 2010 and median 2015; for DIS, there are 4 abnormal return: around 2009, 2010, 2011 and 2013. These abnormal returns show that our data may probably exist outliers.

(F) (i) Stock splits is the decision made by the board of company to increasing company outstanding share by issuing more stocks to current investors.

(ii) By stock splits, the company's stock price decreases, so the stock price will become more attractive to potential investors. As a result, the liquidity of stock will increase. This is good for company to raise fund when they are needed.

(iii) A company will choose stock splits when the stock price is too high to attractive potential investors to invest. A company can choose any time to split stocks through the whole year.

(iv) Usually stock splits will lead to a decrements of stock price, so the log-return will show a downside jump.

(v) From the figure, we can find some downside jump of log-return, the time of these downside jumps are around the middle of the year, so there is a big chance that these jumps are caused by stock splits.

(vi) Yes, they provide the adjusted price which has been adjusted for stock dividends or stock splits.

(vii) No, stock price will affect the price begin the stock market hours, so it will not affect the return calculated within the day.

The **MATLAB** code for Exercise 1(for PG):

```
addpath('C:\Users\zmhua\Documents\MATLAB\data','functions');
PG=csvread('PG.csv');
name='PG';
%--------EXCERCISE 1(PG) ----------------

%Part A
% get Y,M,D,H,MN,S
Y_PG=floor(PG(:,1)/10000); % year
M_PG=floor((PG(:,1)-Y_PG*10000)/100); %month
D_PG=PG(:,1)-10000*Y_PG-100*M_PG; %day
% create a dates_prices matrix
[PG_dates,PG_lprices]=load_stock('PG.csv','m');

%Part B
% calculate N and T
N_PG=sum(PG(1,1)==PG(:,1));% number of observations per day
n_PG=N_PG-1; % number of intervals per day
T_PG=size(PG,1)/N_PG; %number of days

%Part D
% calculate stock log-return and return dates
[PG_rdates,PG_lr]=log_return([PG_dates,PG_lprices],N_PG,1);

%Part E
```

```
% plot price
figure;
plot(PG_dates,exp(PG_lprices));
xlim([min(PG_dates),max(PG_dates)]);
xlabel('Year');
ylabel('Price(dollars)');
title(['Price of ' name '(dollars)']);
datetick('x','keeplimits');
% plot log−return
figure;
plot(PG_rdates(:),100∗PG_lr(:));
xlabel('Year');
ylabel('log−return(\%)');
title(['log−return of ' name '(\%)']);
xlim([min(PG_rdates(:)),max(PG_rdates(:))]);
datetick('x','keeplimits');%transfer the x−axis format
```

Exercise 2

(A) Here are the figures of RV for PG and DIS:

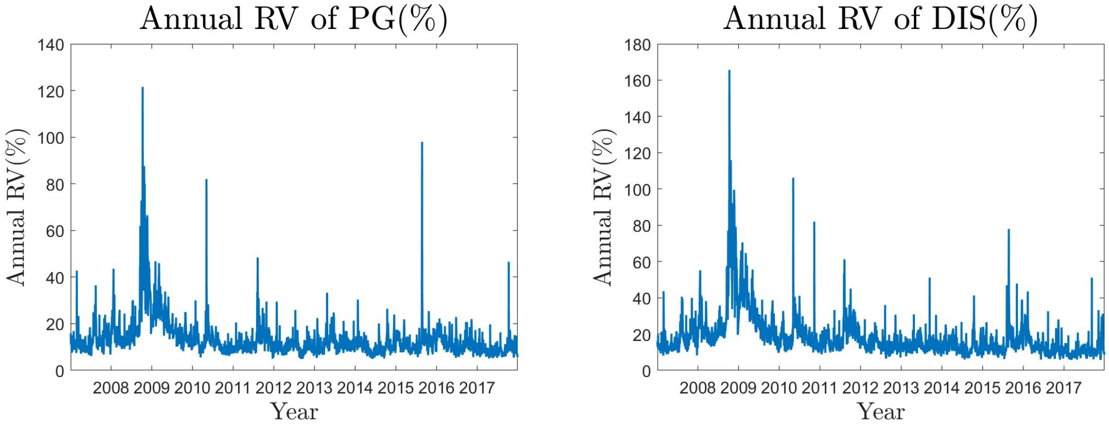

Figure 2: RV for PG and DIS

From the figures, the range of RV for PG and DIS are (15%, 120%) and (15%,170%) respectively. We can see that the shape of RV for both stocks are really similar: the value of RV move around 18% and there are some fluctuations. The same as what we find in Exercise 1, the RV is abnormally high at some period. There may exist some outliers or big price jump in our data for PG and DIS.

The **MATLAB** code:

```
function [rv] = realized_var(log_returns)
%calculate the annual realized variance percentage
rv=sum(log_returns.^2);
rv=100*sqrt(252*rv);
rv=rv(:);
end
```

(B) Here are the figures of RV for PG and DIS:

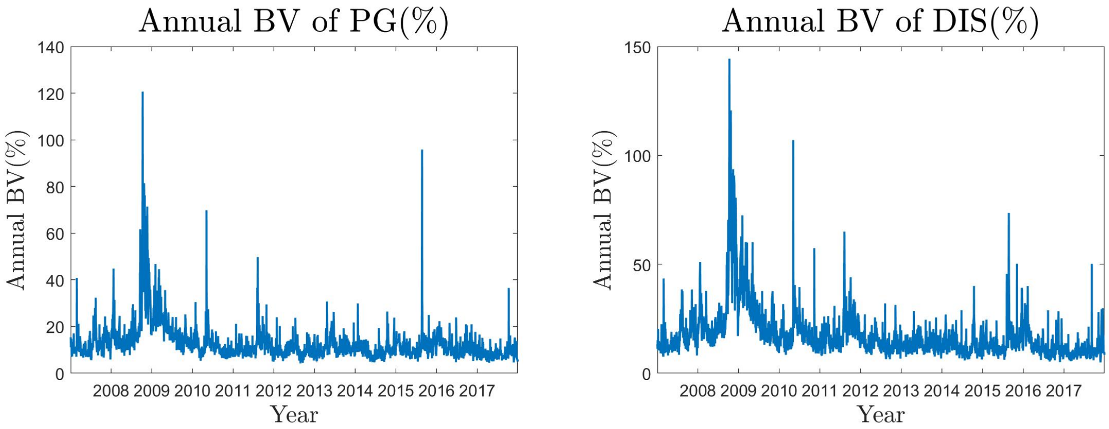

Figure 3: BV for PG and DIS

From the figures, the range of BV for PG and DIS are (13%, 120%) and (14%,150%) respectively. We can see that the shape of BV for both stocks are really similar: the value of RV move around 15% and there are some fluctuations. The same as what we find in Exercise 1, the BV is abnormally high at some period. There may exist some outliers or big price jump in our data for PG and DIS.

The **MATLAB** code:

```matlab
function [bv] = bipower_var(log_returns)
%calculate the annual bipower variance percentage

bv=(pi/2)*sum(abs(log_returns(2:end,:).*log_returns(1:end-1,:)));
bv=100*sqrt(252*bv);
bv=bv(:)
end
```

(C) Here are the figures of RV and BV for PG and DIS:

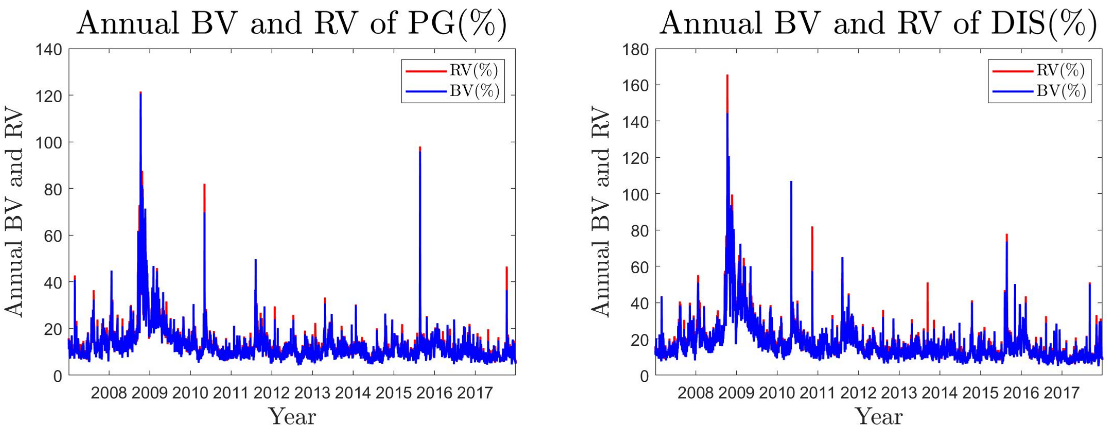

Figure 4: RV and BV for PG and DIS

From the figures, we can see that the shape of BV are very similar to RV for both stocks: they show the same movements. However, there are some differences between RV and BV: the value and volatility of BV is smaller than RV, this is obvious for DIS. By using BV, the max variance of DIS's log-return decrease from 170% to 150%. These figure show us that BV can give a smoother variance curve if our price data exists some jumps or outliers.

(D) Here are the figures of contribution of jumps for PG and DIS:

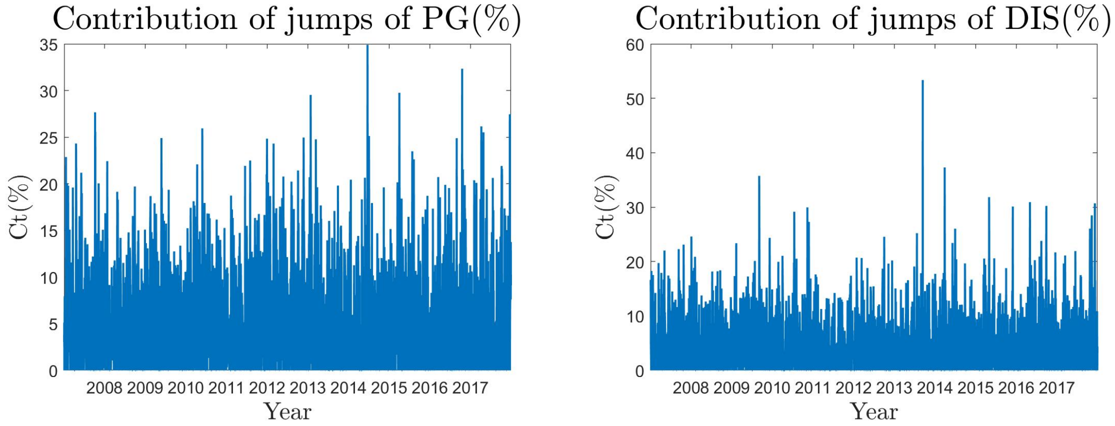

Figure 5: Ct for PG and DIS

From the figures, we can see that the relative contribution of jump for PG is bigger than DIS', which mean the price of PG may contains more jumps than DIS'. The mean of relative contribution of jump for PG and DIS are 4.69% and 4.49% respectively. These results are consistent to the finding of Huang and Tauchen in 2005: “The empirical work indicates strong evidence for jumps, where jumps account for about 4.5% to 7% of the total daily variance of S&P index, cash or futures”.

The **MATLAB** code for Exercise 2(here use PG as data sample):

```
addpath('C:\Users\zmhua\Documents\MATLAB\data','functions');
PG=csvread('PG.csv');
name='PG';
%-------EXCERCISE 2 ---------------

% create a dates_prices matrix
[PG_dates,PG_lprices]=load_stock('PG.csv','m');
N_PG=sum(PG(1,1)==PG(:,1));% number of observations per day
% calculate stock log-return and return dates
[PG_rdates,PG_lr]=log_return([PG_dates,PG_lprices],N_PG,1);

%Part A
% calculate RV
rv_PG=realized_var(PG_lr);
PG_days=unique(floor(PG_rdates));
% plot RV
figure;
plot(PG_days,rv_PG);%plot annual rv
xlim([min(PG_days),max(PG_days)]);
xlabel('Year');
ylabel('Annual RV(\%)');
title(['Annual RV of ' name '(\%)']);
datetick('x','keeplimits');%transfer the x-axis format

%Part B
% calculate BV
bv_PG=bipower_var(PG_lr);
% plot BV
figure;
plot(PG_days,bv_PG);
xlim([min(PG_days),max(PG_days)]);
xlabel('Year');
ylabel('Annual BV(\%)');
title(['Annual BV of ' name '(\%)']);
datetick('x','keeplimits');

%Part C
% plot RV & BV
```

```
figure;
plot(PG_days,rv_PG,'-r');
hold on;
plot(PG_days,bv_PG,'b');
xlim([min(PG_days),max(PG_days)]);
xlabel('Year');
ylabel('Annual BV and RV');
title(['Annual BV and RV of ' name '(\%)']);
legend('RV(\%)','BV(\%)');
datetick('x','keeplimits');

%Part D
% calculate contribution of jumps
Ct_PG=max(rv_PG-bv_PG,0)./rv_PG;
C_PG=mean(Ct_PG);
% plot Ct
figure;
plot(PG_days,100*Ct_PG);
xlim([min(PG_days),max(PG_days)]);
xlabel('Year');
ylabel('Ct(\%)');
title(['Contribution of jumps of ' name '(\%)']);
datetick('x','keeplimits');
```

Exercise 3

(A) (i) The realized variance is a unbiased estimator for variance if the prices contain no jumps. Through using realized variance, we can treat volatility as observable by using the square of log-returns from high frequency sample to approach it.

(ii) The volatility signature plot is a figure to describe the relationship between average realized variance and the frequency of data sample.

(iii) From the volatility signature plot, we can learn that when the data frequency is extremely high, the market's microstructure noises will cause huge bias to variance. So, the sample frequency is not as highest as better, some frequency with intermediate value can let us avoid the effect of microstructure noises but provide enough data for us to estimate the variance.

(B) In 5s frequency('BAC(2015).csv') data, we have: $N = 4621$ and $T = 252$.

(C) The follow are the volatility plot for the 1st trade day:

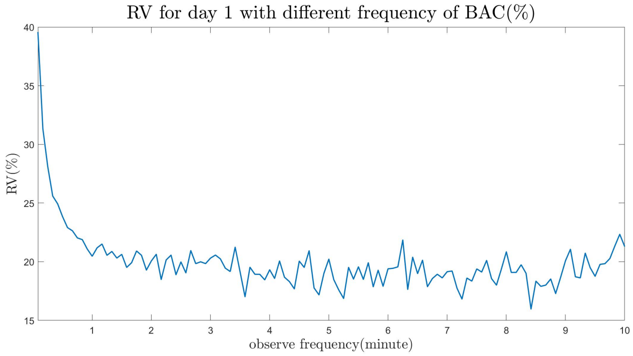

Figure 6: RV for day1 with different frequency sample

From the figure, when sample frequency is extremely high, the value of RV is extremely high, too. We can find that the value of RV first shows a down-trend and then increase when the sample frequency start decreases from 8 minute to 10 minute, which means that as the frequency of observation decreases from extremely high frequency, the volatility decreases. However, if the sample frequency is too low, say lower than 8 minute, the volatility will increase. The value of RV comes relatively stable When the frequency falls into 4 to 8 minute.

(D) The follow are the year average volatility plot:

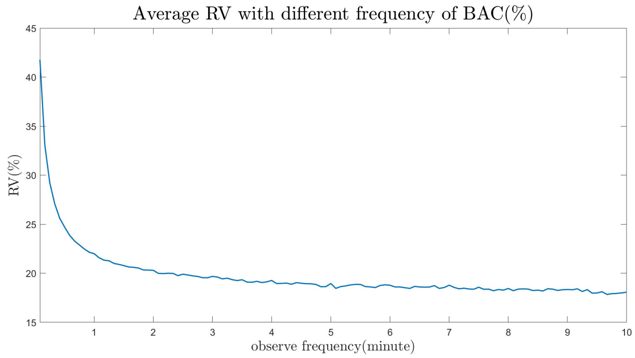

Figure 7: Annual RV with different frequency sample

From the figure, we can find that the RV curve becomes smooth as we take average for day's RV. Same as what we find in day's RV, the value of average RV shows a downtrend, which means that as the frequency of observation decreases from extremely high frequency, the volatility decreases. However, if the sample frequency continues to decrease, the volatility will still decrease(but just a little bit). The RV value becomes stable when frequency lower than 4 minute.

(E) We have construct a matrix to record the mean-difference of RV_{J1} and RV_{J2}. According to the matrix, we find that when sample frequency is extremely high (both $J1$ and $J2$ are very small), the mean difference is large than the mean-difference of other high frequency samples. The reason may be that, since the sample is of extremely high frequency, it is of higher possibility that we will observe the noise and take it into our price data.

(F) Yes. as we can see from the figure in part D, the RV curve becomes very flat. The value of RV for frequency=3 minute is 20%, and the value of RV for frequency=8 is around 18%. Even the fequency of observation has decreased almost two times, the value of RV just decreases by 10%.

The **MATLAB** code for Exercise 3(A) to 3(F):

```
addpath('C:\Users\zmhua\Documents\MATLAB\data','functions');
BAC=csvread('BAC—2015.csv');
name='BAC';
%————EXCERCISE 3 ————————

%Part B
% produce dates_prices matrix
[BAC_dates,BAC_prices]=load_stock('BAC—2015.csv','s');
```

```
% calculate N,T,n
N_BAC=sum(BAC(1,1)==BAC(:,1));% number of observations per day
n_BAC=N_BAC-1; % number of intervals per day
T_BAC=size(BAC,1)/N_BAC;

%Part C
% calculate RV for day1 with different frequency data
rv_J_BAC=[];
for J=1:120
    [BAC_rdates,BAC_lr]=log_return([BAC_dates,BAC_prices],N_BAC,J);
    rv_J_BAC(:,J)=realized_var(BAC_lr);
end
% plot RV for day 1 with different observe frequency
J=1:120;
figure;
plot(J/12,100*sqrt(rv_J_BAC(1,:)*252));
xlabel('observe frequency(minute)');
ylabel('RV(\%)');
title(['RV for day 1 with different frequency of ' name '(\%)']);
xlim([min(J/12),max(J/12)]);

%Part D
% calculate annual average RV with different frequency data
mean_rv_J_BAC=mean(rv_J_BAC);
% plot annual RV with different observe frequency
figure;
plot(J/12,mean_rv_J_BAC);
xlabel('observe frequency(minute)');
ylabel('RV(\%)');
title(['Average RV with different frequency of ' name '(\%)']);
xlim([min(J/12),max(J/12)]);

%Part E
% construct RV pair-difference matrix w.r.t different J
Diff_J_BAC=[]
for i=1:120
    for k=1:120
      Diff_J_BAC(i,k)=mean(abs(rv_J_BAC(:,i)-rv_J_BAC(:,k)));
    end
end
```

(G) Here is the figure of adjusted average volatility signature plot for IBM from 2007 to 2017:

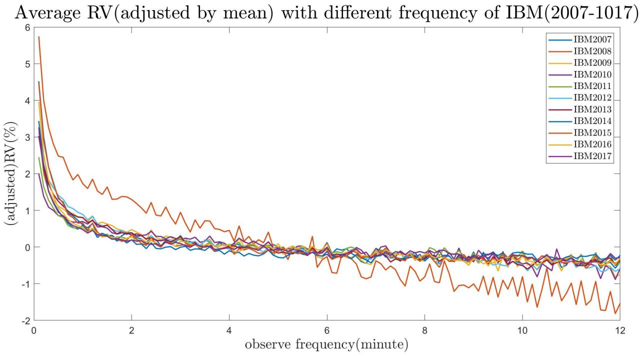

Figure 8: Annual RV with different frequency sample

From the figure, for most of our sample, the adjusted average volatility curves are smooth and show a reasonable flat pattern when the frequency decreases from 3 to 8 minutes. However, for year 2008, even the year average volatility curve still has lots of fluctuations (the most volatile orange line). when the frequency decreases from 5s to 10 minutes, the volatility shows a down-trend, but it does not show a reasonable flat for sampling intervals corresponding to about 3 to 8 minutes.

We can see that the volatility signature plot will show a similar shape for different years. However, when there is a huge shock from the market, such as the financial crisis, the volatility signature plot will still show high volatility even we choose a "great" frequency for our sample.

The **MATLAB** code for function to calculate adjusted volatility with different frequency:

```
function [mRV_J] = mRV_J(file_name)
addpath('functions');
data=csvread(file_name);
log_prices=log(data(:,3));
log_prices=reshape(log_prices,4621,[]); %reshape by day group

rv_J_STOCK=[];
for J=1:120
    rv_J_STOCK(:,J)=realized_var(diff(log_prices(1:J:end,:)));
end
mean_rv_J_STOCK=mean(rv_J_STOCK);
mRV_J=mean_rv_J_STOCK-mean(mean_rv_J_STOCK);
```

```
end
```

```
addpath('C:\Users\zmhua\Documents\MATLAB\data\Stocks5Sec','functions');
plotDefaults
%————————EXCERCISE 3 ————————————————
%Part B


IBM7=mRV_J('IBM–2007.csv');
IBM8=mRV_J('IBM–2008.csv');
IBM9=mRV_J('IBM–2009.csv');
IBM10=mRV_J('IBM–2010.csv');
IBM11=mRV_J('IBM–2011.csv');
IBM12=mRV_J('IBM–2012.csv');
IBM13=mRV_J('IBM–2013.csv');
IBM14=mRV_J('IBM–2014.csv');
IBM15=mRV_J('IBM–2015.csv');
IBM16=mRV_J('IBM–2016.csv');
IBM17=mRV_J('IBM–2017.csv');

J=1:120;
plot(J/10,IBM7);
hold on;
plot(J/10,IBM8);
hold on;
plot(J/10,IBM9);
hold on;
plot(J/10,IBM10);
hold on;
plot(J/10,IBM11);
hold on;
plot(J/10,IBM12);
hold on;
plot(J/10,IBM13);
hold on;
plot(J/10,IBM14);
hold on;
plot(J/10,IBM15);
hold on;
plot(J/10,IBM16);
hold on;
plot(J/10,IBM17);

xlabel('observe frequency(minute)');
ylabel('(adjusted)RV(\%)');
```

```
legend('IBM2007','IBM2008','IBM2009','IBM2010','IBM2011','IBM2012','IBM2013','IBM2014','IBM2015','IBM2016','IBM2017');
title('Average RV(adjusted by mean) with different frequency of IBM (2007—1017)');
```