

# 10.6/10 Nothing but PERFECT!

Ziming Huang

## Project 3

Econ 672  
September 20, 2019

### Exercise 1

(A) To compute the time-of-day-factor, we need to do the following steps:

- (i) Use log-price  $X_{t,i}$  to calculate log-return  $r_{t,i}$ ;
- (ii) Use this formula to calculate variable  $b_i$ :  $b_i = \frac{1}{T} \sum_{t=1}^T |r_{t,i}| \times |r_{t,i-1}|$ , for  $i = 2, 3, \dots, n$ ;
- (iii) Set  $b_1 = b_2$  and use  $b_i$  to calculate time-of-day-factor  $\tau_i$ :  $\tau_i = \frac{b_i}{\frac{1}{n} \sum_{j=1}^n b_j}$ , for  $i = 1, 2, 3, \dots, n$ .

Here is the plot of time-of-day-factor:

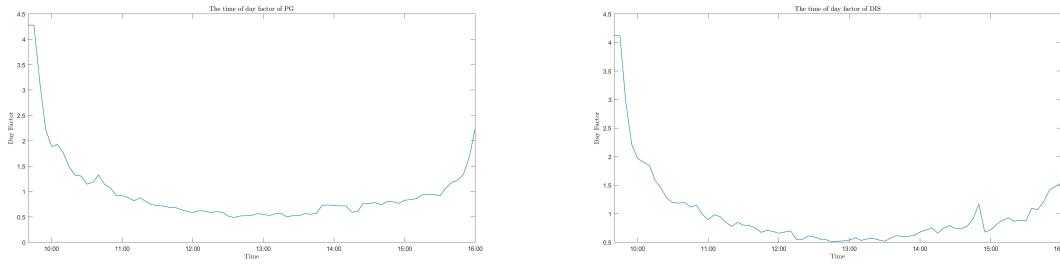


Figure 1: Time-of-day-factor for PG and DIS

From these figures, we can see that the range of time-of-day-factor of PG and DIS are  $(0.5, 4.3)$  and  $(0.5, 4.1)$  respectively. The curve of PG's time-of-day-factor is smoother than the curve of DIS'. We can also find at DIS' figure, the time-of-day-factor has a small peak at around 15:00, which does not show in PG' figure.

Both PG and DIS have a U-shape time-of-day-factor: at the early time of trade hours, the time-of-day-factor is very large and then shows a trend of decreasing; at the middle of trading hours, the time-of-day-factor reaches its lowest value; then the time-of-day-factor begins to increase till the end of the trading hours. Even though the time-of-day-factor shows a upside trend after middle trading hours, the highest value it can reach is still smaller than the time-of-day-factor at the very beginning of the trading day.

The **MATLAB** code:

```
1 function [factor] = time_of_day_factor(log_returns)
2 B=sum(transpose(abs(log_returns(2:end,:).*log_returns(1:end-1,:))));
3 B=[B(1) B];
4 factor=transpose(B/mean(B));
5
6 end
```

**Excellent job! (+0.6)**

- (B) Lee and Mykland (2006)<sup>1</sup> have found that the jumps are more frequency happens within the 9:30 to 10:30. Tassel(2007)<sup>2</sup> shows that: “The number of flagged jumps also begins to increase later in the trading day, a potential response to the Federal Open Market Committee statements that are released at approximately 2:15pm”.

The frequency jumps may be the reason to cause the volatility much higher at the begin of trading hours. McInish and Ord in their paper(1985)<sup>3</sup> have discovered that the price of individual stock price volatility intraday shows U-shape pattern. The U-shape pattern of price volatility means that the volatility of stock price will follow the distribution with different variance.

Since we want to use the bi-power variance as a estimator of price's really volatility, in order to make this estimator more efficient, we need to multiple a parameter to adjust the value of bi-power variance and to let it match the small intervals' variance. The parameter we use is the time-of-day-factor. Since the function of the time-of-day-factor is to help us to estimate the variance with day, the shape of time-of-day-factor with share the same pattern of the volatility of price intraday. If the volatility of intraday price does not have the pattern of U-shape of it is just show a reasonable flat pattern, then our day-of-time-factor will have a constant pattern or similar.

- (C) To separate the diffusive returns from jump returns, we have the following steps;

- (i) Calculate  $BV_t$  to estimate the returns volatility;
- (ii) Choose the number of  $\sigma$  that we want to bound our diffusive returns(i.e.  $\alpha$ );
- (iii) Use this formula to calculate our cut\_off matrix:  $\alpha\Delta_n^0.49\sqrt{\tau_iBV_t}$ ;
- (iv) Use cut\_off matrix as indicator to separate diffusive returns and jump returns:

We can verify that  $r_{t,i} = r_{t,i}^c + r_{t,i}^d$  by using the code: `k=sum((diffusive return(:)+jump return(:)==regular(:)))`. We have k=0, which means for every element in regular log returns, it will equal to diffusive return plus jump return.

Here are the plots of diffusive returns and jump returns for PG and DIS:

---

<sup>1</sup>Lee, S. S., and P. Mykland (2006). Jumps in Financial Markets: A New Nonparametric Test and Jump Dynamics. Working Paper. Georgia Institute of Technology and University of Chicago.

<sup>2</sup>P. Van Tassel. Patterns Within the Trading Day: Volatility and Jump Discontinuities in High-Frequency Equity Price Series. Duke University Senior Honors Thesis, 2009.

<sup>3</sup>Wood, R. A., T. H. McInish, and J. K. Ord (1985). An investigation of transaction data for NYSE stocks. *Journal of Finance*, 25, 723-739.

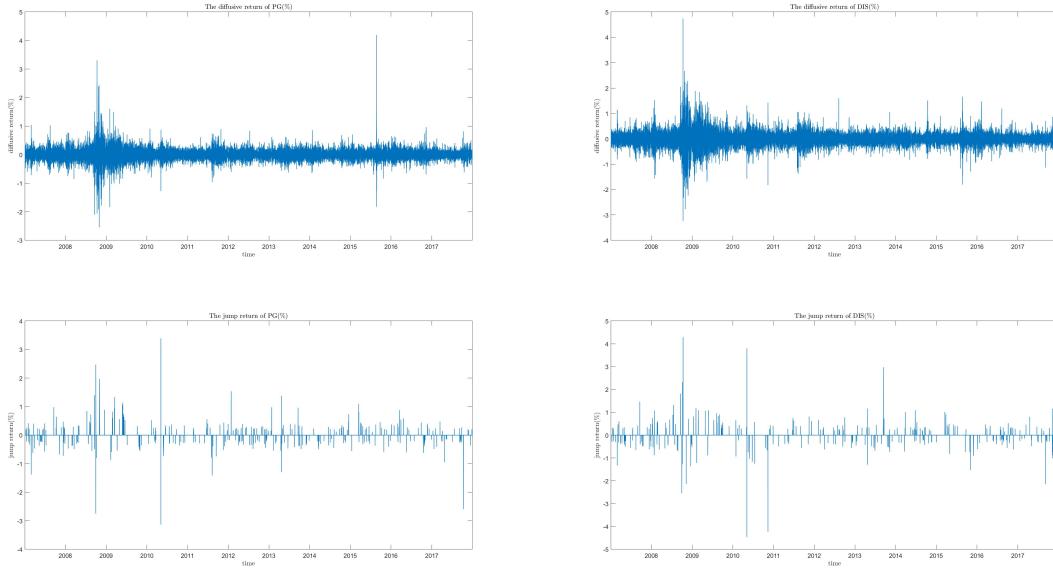


Figure 2: Diffusive returns and jump returns of PG and DIS

We can find the diffusive returns for both PG and DIS are moving around value 0, however, there are some abnormal values in these diffusive returns. For PG, it has three obvious abnormal diffusive returns: around the end of 2008, median 2010 and median 2015; for DIS, there are 5 obvious abnormal diffusive returns: around the begin of 2006, end of 2008, the end of 2011, median 2012 and median 2015. These abnormal returns show that our data may probably exist outliers.

As for the jump returns, we find that the jumps return has smaller sample number than diffusive returns. Same as diffusive returns, we can observe that the jump returns move around 0 value and we can not find significant trend of jump returns deviate from zero. The frequency of large returns are higher in jump returns than diffusive returns. Without doing some statistic works, We can not tell which jump type (positive jump or negative jump) happens more frequently just through these figures.

The MATLAB code:

```

1 function [bv] = bipower_var_day(log_returns)
2 %calculate the annual bipower variance percentage
3
4 bv=(pi/2)*sum(abs(log_returns(2:end,:).*log_returns(1:end-1,:)));
5 %bv=100*sqrt(252*bv);
6 %bv=bv(:)
7 end

```

```

1 function [outputArg1,outputArg2] = cut_off(log_retun,a,N,T)
2 %UNTITLED7 Summary of this function goes here
3 % Detailed explanation goes here
4 F=time_of_day_factor(log_retun);
5 BV=bipower_var_day(log_retun);
6 cut_off=[];
7 for T=1:T
8     for i=1:N
9         cut_off(i,T)=a*(1/N)^(0.49)*sqrt(F(i)*BV(T));
10    end
11 end
12
13 end

```

```

1 function [c_log_returns,d_log_returns] = c_d_log_returns(log_returns,N,a)
2 %log_returns is a N*T matrix,N is the number of observation per day and T is
3 %the number of days for data
4 %a is the chosen number of sd.
5 addpath('functions');
6 bv=bipower_var_day(log_returns);
7 F=time_of_day_factor(log_returns);
8 cutoff=a*(1/N)^(0.49)*sqrt(F.*bv);
9 c_log_returns=log_returns.*(abs(log_returns)<=cutoff);
10 d_log_returns=log_returns.*(abs(log_returns)>=cutoff);
11 end

```

better to use a table next time

- (D) The number of jumps from 2007 to 2017 for PG is [38, 24, 21, 29, 22, 44, 28, 28, 25, 24, 27] and total 310. The number of jumps each for PG is [37, 33, 20, 21, 14, 24, 26, 32, 15, 32, 27] and total 281 times.

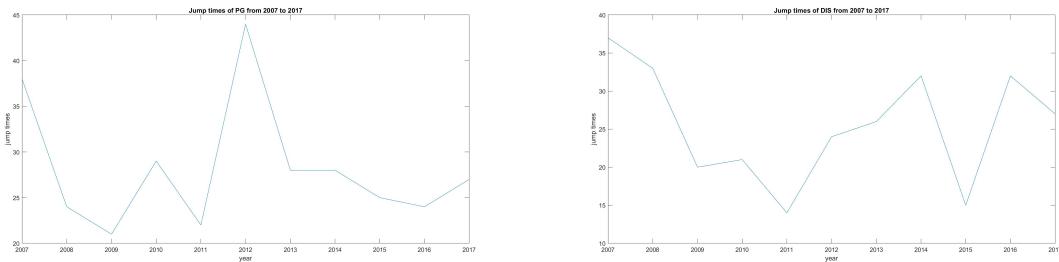


Figure 3: Jump times each year(from 2007 to 2017)

As we can notice from this figure, there are more jumps in the year when there are

crisis, say 2007, 2012 and 2015. From the jump returns figures in part C, we can find that there usually more jumps around the end of year.

- (E) The follows are the histogram figures of PG and DIS' diffusive returns and non-zero jump returns.  
Great insight here!

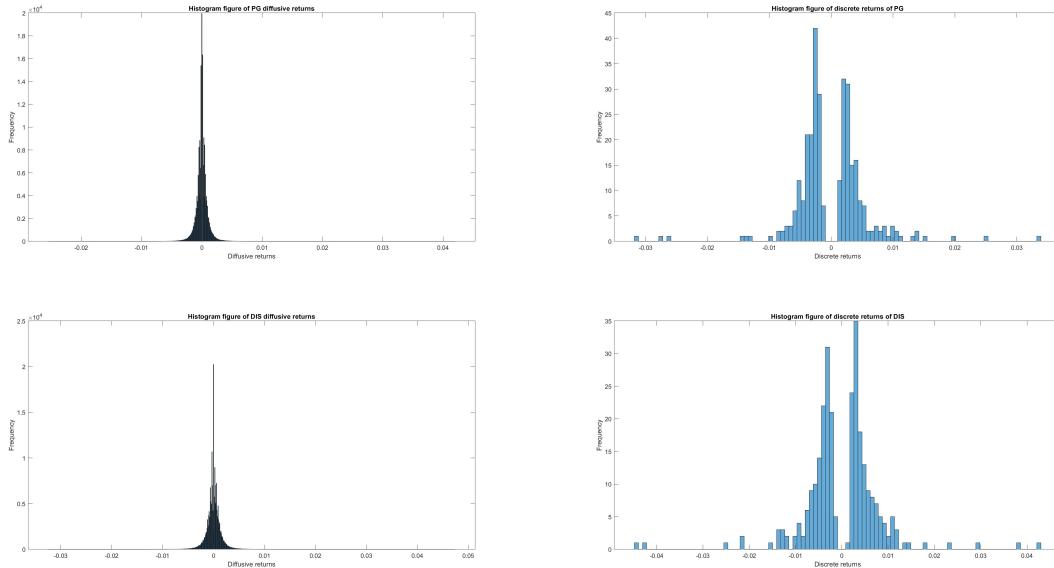


Figure 4: Histograms of diffusive returns and jump returns for PG and DIS

We can see from the figures that the distribution of diffusive returns for PG and DIS are very similar to normal distribution with mean equal to zero. In order to catch the characteristics of its distribution, we delete the zero value when plot the figures. As we can see from the figure, the jump returns distribution may have two peaks and skew to the left.

It is hard to tell the distribution of jump returns without deleting zero values in our sample. If there must be a distribution of total jump returns(include bunch of zeros), the distribution may have mean =0 and volatility very close to 0. Since the non zeros jumps returns just take up 0.15% in total sample, the distribution type will mainly depend on large zero part.

- (F) Since it makes no sense to discuss about the distribution of jump returns with bunch of zero samples, so here just do distribution estimation for non-zero jump returns. The follows are the figures of estimated p.d.f. of diffusive returns and non-zero jump returns of PG and DIS.

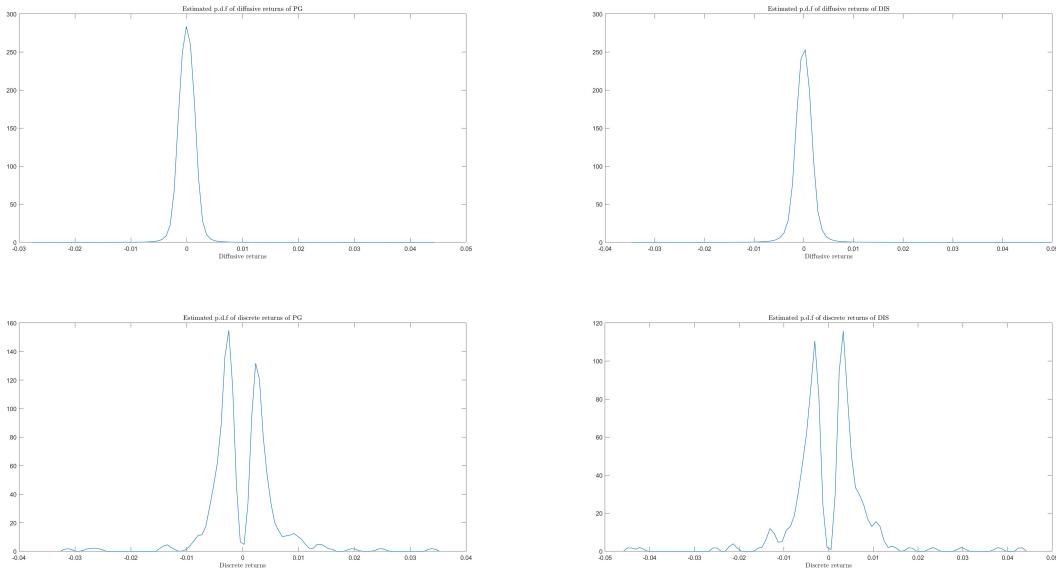


Figure 5: Estimated p.d.f. of diffusive returns and non-zero jump returns for PG and DIS

We can see from the figures that, both PG and DIS' diffusive returns' distribution are very similar to normal distribution with mean equals to zero. As for the jump returns, we know that the distributions of jump returns have two peaks. PG's jump returns skews to the left but the skewness isn't very obvious in DIS' jump returns distributions.

- (G) In the figures of PG's jump returns distribution, it does show that stock crashes are larger and more often than rallies, but this conclusion inverses for the case of DIS.

The MATLAB code:

```

1 addpath('D:\Documents\MATLAB\data','functions','scripts');
2 [dates_PG,lp_PG,Y,M,D,H,MN,S]=load_stock('PG.csv','m');
3 N_PG=sum(floor(dates_PG(1,1))==floor(dates_PG(:,1)));% number of observations
   per day
4 T_PG=size(dates_PG,1)/N_PG;
5 [rdates_PG,lr_PG]=log_return([dates_PG lp_PG],N_PG,1);

6
7
8 %---ex1
9 %part a
10 F_PG=time_of_day_factor(lr_PG);
11 figure;
12 plot(rdates_PG(:,1),F_PG);
13 xlabel('Time');
14 ylabel('Day Factor');

```

```
15 xlim([min(rdates_PG(:,1)),max(rdates_PG(:,1))]);
16 title('The time of day factor of PG');
17 datetick('x',15,'keeplimits');
18
19 %part c
20 a=4;
21 bv_PG=bipower_var_day(lr_PG);
22 cutoff_PG=a*(1/N_PG)^(0.49)*sqrt(F_PG.*bv_PG);
23 lr_c_PG=lr_PG.*(abs(lr_PG)<=cutoff_PG);
24 lr_d_PG=lr_PG.*(abs(lr_PG)>=cutoff_PG);
25 k=sum((lr_c_PG(:)+lr_d_PG(:)==lr_PG(:))); %verify whether the continuous
     returns + jump returns=total returns
26
27 %plot diffusion return
28 figure;
29 plot(rdates_PG(:,lr_c_PG(:)*100);
30 xlabel('time');
31 ylabel('diffusive return(\%)');
32 xlim([min(rdates_PG(:)),max(rdates_PG(:))]);
33 title('The diffusive return of PG(\%)');
34 datetick('x','keeplimits');
35 %plot jump return
36 figure;
37 plot(rdates_PG(:,lr_d_PG(:)*100);
38 xlabel('time');
39 ylabel('jump return(\%)');
40 xlim([min(rdates_PG(:)),max(rdates_PG(:))]);
41 title('The jump return of PG(\%)');
42 datetick('x','keeplimits');
43
44
45 %part d
46 Y1={'20070101','20080101','20090101','20100101','20110101','20120101','
      20130101','20140101','20150101','20160101','20170101','20180101'};
47 numY_PG=datenum(Y1,'yyyymmdd');
48 M=[rdates_PG(:,lr_d_PG:)];
49 numJ_PG=[];
50 for i=1:11
51     numJ_PG(i)=sum(M(:,1)<=numY_PG(i+1) & M(:,2)~=0);
52 end
53 numJ_PG=diff([0 numJ_PG]);
54 figure;
55 plot(numY_PG(1:11),numJ_PG);
56 xlabel('year');
```

```
57 ylabel('jump times');
58 xlim([min(numY_PG(1:11)),max(numY_PG(1:11))]);
59 title('Jump times of PG from 2007 to 2017');
60 datetick('x','keeplimits');
61
62 %part e
63 figure;
64 histogram(lr_c_PG(:,1000);
65 xlabel('Diffusive returns');
66 ylabel('Frequency');
67 title('Histogram figure of PG diffusive returns');
68 figure;
69 histogram(nonzeros(lr_d_PG),100);
70 xlabel('Discrete returns');
71 ylabel('Frequency');
72 title('Histogram figure of discrete returns of PG');
73
74 %part g
75 %diffusion returns
76 [f_rc_PG,x_rc_PG]=ksdensity(lr_c_PG(:, 'Kernel','epanechnikov','Bandwidth'
    ,0.001);
77 figure;
78 plot(x_rc_PG,f_rc_PG);
79 xlabel('Diffusive returns');
80 title('Estimated p.d.f of diffusive returns of PG');
81 %jump returns(delete zero returns)
82 [f_rd_PG,x_rd_PG]=ksdensity(nonzeros(lr_d_PG(:)), 'Kernel','epanechnikov','
    Bandwidth',0.0006);
83 figure;
84 plot(x_rd_PG,f_rd_PG);
85 xlabel('Discrete returns');
86 title('Estimated p.d.f of discrete returns of PG');
87 %part h
88 pJ_PG=sum(lr_d_PG(:)>0);
89 nJ_PG=sum(lr_d_PG(:)<0);
```

**Exercise 2**

- (A) To calculate the truncated variance  $TV_t$ , we use the diffusive returns we get from Exercise 1 and apply the formula:  $TV_t = \sum_{i=1}^n (r_{i,t}^c)^2$ .

The follows are the figures.

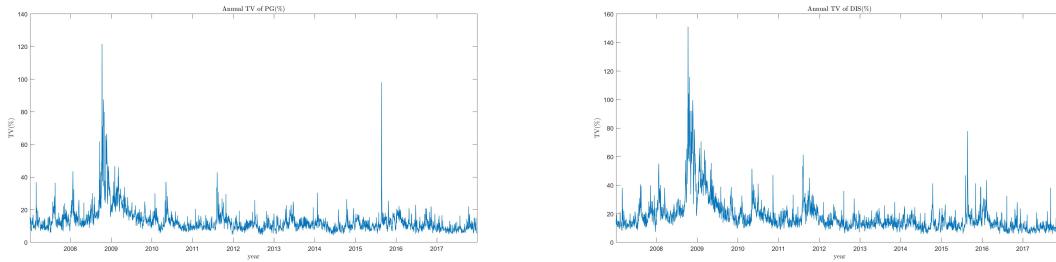


Figure 6: Annual TV for PG and DIS

From the figures, the range of TV for PG and DIS are (13%, 120%) and (18%, 155%) respectively. We can see that the shape of TV for both stocks are really similar: the value of TV move around 18% and there are some fluctuations. The TV value is abnormally high at some period, so there may exist some outliers or big price jump in our data for PG and DIS. Besides, we can also observe that DIS' TV is bigger than PG's TV for most of time, which mean that the diffusive returns of DIS have more volatility.

The MATLAB code:

```

1 function [tv] = truncated_var_day(diffusive_log_returns)
2 addpath('functions')
3 tv=sum(diffusive_log_returns.^2);
4 end

```

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

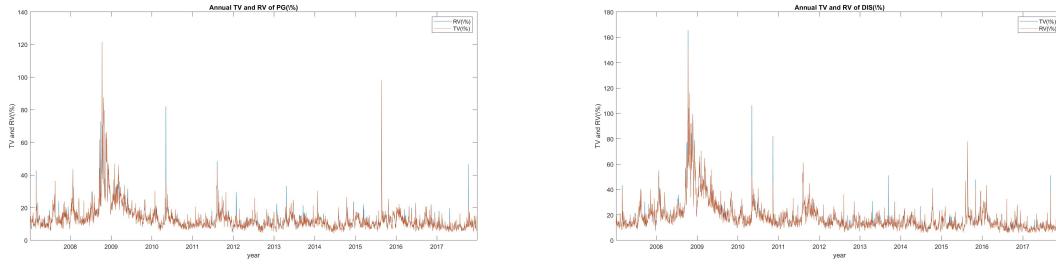


Figure 7: Annual RV and TV for PG and DIS

From the figures, the range of RV for PG and DIS are (15%, 120%) and (18%, 170%) respectively. We can see that the shape of RV and TV for both stocks are really similar, however, the value of RV is a little larger than TV. These figures show us that by using the diffusive returns instead of the whole regular returns to estimate the volatility of stock's log-returns can give us a smooth(smaller) estimator when the prices of stock exist jumps.

(C) We can estimate the quartic integrated variance(QIV) by applying the formula:

$$Q\hat{IV}_t = \frac{1}{3\Delta_n} \sum_{i=1}^n (r_{t,i}^c)^4$$

In order to see the volatility of QIV clearly, we annualized daily QIV. Here are the figures of PG and DIS' QIV estimator.

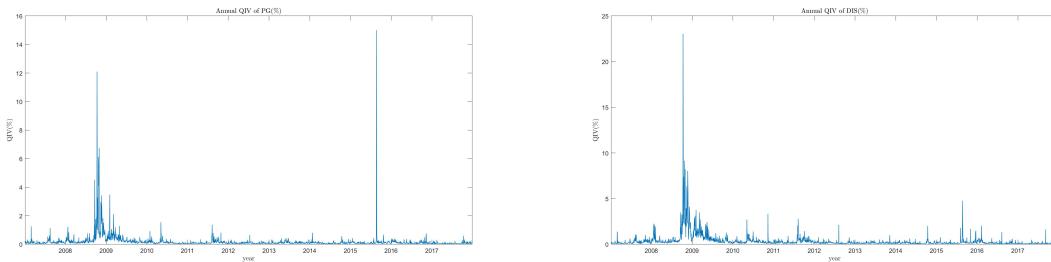


Figure 8: Annual estimated QIV for PG and DIS

From the figures, we can see that the most value of QIV are very closed to zero. However, we can observe during some period of financial crisis, the QIV can have a large value. For example, around the end of 2008 and the middle of 2015, both PG and DIS' QIV show large volatility and the value of QIV have even reached 12% and 24% of end of 2008 and 15% and 5% of middle 2015 for PG and DIS respectively. We can say that

the QIV of PG and DIS are very sensitive to the whole market situation, which may mean that the correlation between market and PG or DIS may be high. Besides, the figures also verifies that the diffusive returns of DIS have large volatility since the QIV value of DIS is larger.

The MATLAB code:

```

1 function [QIV] = QIV(diffusive_log_returns,N)
2 %diffusive_log_returns is a N*T matrix,N is the number of observation per day
3 %and T is the number of days for data
3 QIV=sum((N/3)*(diffusive_log_returns.^4));
4 end

```

(D) We use this formula to calculate the confidence intervals for IV:

$$CI(IV_t, 1 - \alpha) = [TV_t \pm q_z\left(\frac{\alpha}{2}\right) \times \sqrt{\frac{2}{3} \sum_{i=1}^n (r_{t,i}^c)^4}]$$

where  $\alpha$  is the significant level.

Here are the figures of TV and estimated confidence interval of IV for PG and DIS.

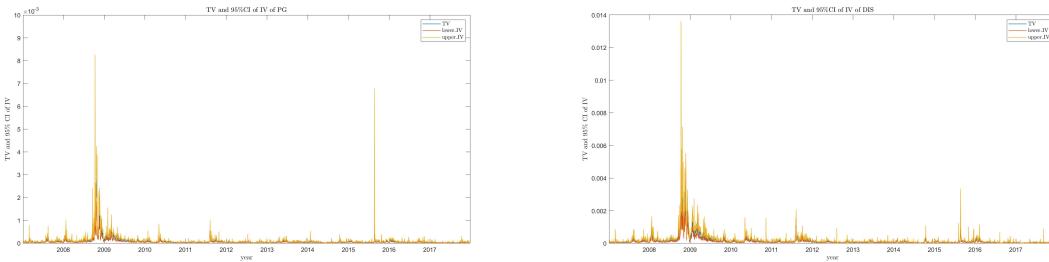


Figure 9: TV and Estimated Confidence Interval of IV for PG and DIS

(E) Now we focus the TV and IV's estimated confidence interval at a specific time interval, say October 2008. Here are the figures.

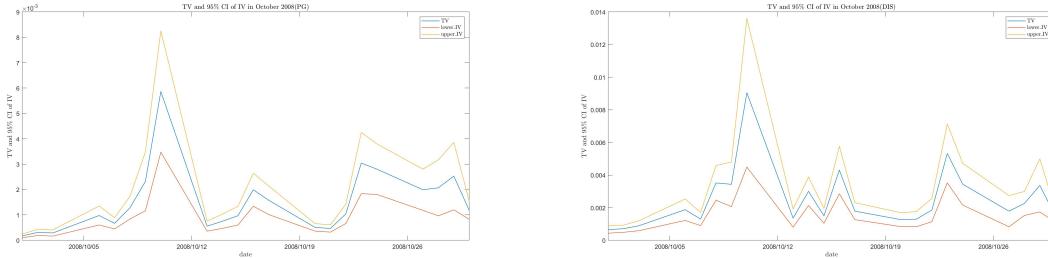


Figure 10: TV and IV's estimated confidence interval in October 2008

From these figures we can find that, the range of PG and DIS' TV are  $(0, 0.006)$  and  $(0.0005, 0.009)$  respectively. The volatility of TV is larger for DIS' TV than PG's. If the TV is a unbiased estimator of IV, then IV's values have 95% probability fall into the intervals between upper TV and lower TV, which means that we will 95% IV's value are bounded by the upper value of TV and the lower value of TV. However, if TV is not a unbiased estimator of IV, the percentage that IV fall into the confidence interval estimated by using RV will not equal(or close) to 95%.

- (F)**
  - (i) If we know the asymptotic distribution of an estimator, then we can get the asymptotic distribution of a function of this estimator by using Delta Method.
  - (ii) By using Delta method, we can easily calculate the asymptotic distribution of the function of estimator, given the asymptotic distribution of this estimator. Specifically, if we know the asymptotic distribution of the returns or variance, by using Delta Method, we can get the annual returns or variance's estimator's asymptotic distribution then calculate its confidence interval or other statistic estimators .
  - (iii) Function  $g$  in this case is  $g(\theta) = 100\sqrt{252}\theta$  ( $\theta$  in our case is  $IV_t$ ).
  - (iv) By taking derivative of function  $g$ , we get  $g'(IV_t) = \frac{300\sqrt{7}}{\sqrt{IV_t}}$ .
  - (v) Since the asymptotic distribution of  $\Delta_n^{-1/2}(TV_t - IV_t)$  is  $\mathcal{N}(0, 2QIV_t)$ , by using the Delta Method we can get the asymptotic distribution of  $\Delta_n^{-1/2}(100\sqrt{252}TV_t - 252IV_t)$  is  $\mathcal{N}(0, 2(g'(IV_t))^2QIV_t)$ , that is  $\mathcal{N}(0, 2 * (\frac{300\sqrt{7}}{\sqrt{x}})^2QIV_t)$ .
- (G)** The confidence interval for  $\Delta_n^{-1/2}(100\sqrt{252}TV_t - 252IV_t)$  is:

$$CI(100\sqrt{252 \times IV_t}, 1 - \alpha) = [100\sqrt{252TV_t} \pm q_z(\frac{\alpha}{2}) \times 100 \frac{3\sqrt{7}}{\sqrt{TV_t}} \sqrt{2\Delta_t Q \hat{IV}_t}]$$

- (H)** Here are the plots of annual TV and 95% confidence intervals.

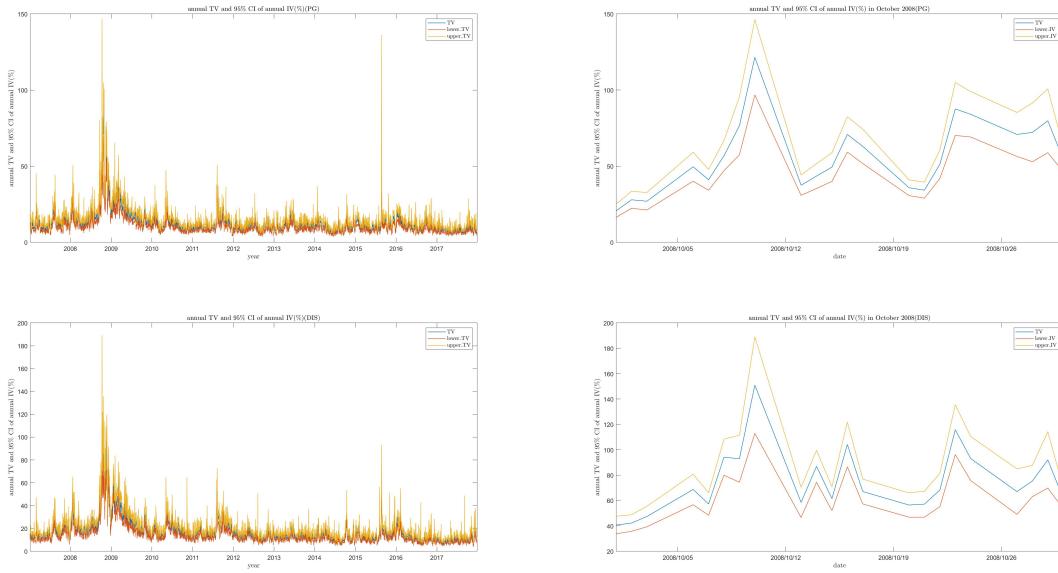


Figure 11: Estimated confidence interval for PG and DIS

As we can see from the figures (focus on October 2008) that, PG and DIS' annual TV have similar shape, they both have small fluctuations on 10 Oct. 2008 and 16 Oct. 2008 and 23 Oct. 2008 and 29 Oct. 2008. According to the figure, we can find that the upper bound and the lower bound of confidence interval have the similar shape of TVt, which may means that we have calculate the correct confidence interval(at lease the shape of confidence interval is right). If the TVt is a unbiased estimator of IVt, then IVt's values have 95% probability fall into the intervals between upper TVt and lower TVt, that is, the percentage that IVt fall into the confidence interval estimated by using RVt will not equal(or close) to 95%.

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

```

1 addpath('D:\Documents\MATLAB\data','functions','scripts');
2 [dates_PG,lp_PG,Y,M,D,H,MN,S]=load_stock('PG.csv','m');
3 N_PG=sum(floor(dates_PG(1,1))==floor(dates_PG(:,1)));% number of observations
   per day
4 T_PG=size(dates_PG,1)/N_PG;
5 [rdates_PG,lr_PG]=log_return([dates_PG lp_PG],N_PG,1);
6 [lr_c_PG,lr_d_PG]=c_d_log_returns(lr_PG,N_PG,4);
7
8 %ex2
9 days_PG=unique(floor(rdates_PG));
10 %part a

```

```
11 tv_PG=truncated_var_day(lr_c_PG);
12 figure;
13 plot(days_PG,100*sqrt(tv_PG(:)*252));
14 xlabel('year');
15 ylabel('TV(\%)');
16 xlim([min(days_PG),max(days_PG)]);
17 title('Annual TV of PG(\%)');
18 datetick('x','keeplimits');
19 %part b
20 rv_PG=realized_var_day(lr_PG);
21 figure;
22 plot1=plot(days_PG,100*sqrt(rv_PG(:)*252));
23 hold on;
24 plot2=plot(days_PG,100*sqrt(tv_PG(:)*252));
25 plot2.Color(4) = 0.8;
26 xlabel('year');
27 ylabel('TV and RV(\%)');
28 xlim([min(days_PG),max(days_PG)]);
29 legend('RV(\%)','TV(\%)');
30 title('Annual TV and RV of PG(\%)');
31 datetick('x','keeplimits');
32 hold off;
33 %part c
34 % estimator of QIV
35 QIV_PG=QIV(lr_c_PG,N_PG)
36 figure;
37 plot(days_PG,100*sqrt(252*QIV_PG));
38 xlabel('year');
39 ylabel('QIV(\%)');
40 title('Annual QIV of PG(\%)');
41 xlim([min(days_PG),max(days_PG)]);
42 datetick('x','keeplimits');
43 %part d
44 %CI
45 c=norminv(0.025);
46 CI_l_PG=tv_PG-c*sqrt(2*QIV_PG/N_PG);
47 CI_r_PG=tv_PG+c*sqrt(2*QIV_PG/N_PG);
48 figure;
49 plot(days_PG,tv_PG);
50 hold on;
51 plot(days_PG,CI_l_PG);
52 hold on;
53 plot(days_PG,CI_r_PG);
54 legend('TV','lower\_\_IV','upper\_\_IV');
```

```
55 xlim([min(days_PG),max(days_PG)]);
56 xlabel('year');
57 ylabel('TV and 95\% CI of IV');
58 title('TV and 95\%CI of IV of PG');
59 datetick('x','keeplimits');
60 hold off;
61 %part e
62 num1=datenum('20081001','yyyyymmdd');
63 num2=datenum('20081031','yyyyymmdd');
64 a=sum(rdates_PG(:)<num1)/77;
65 b=sum(rdates_PG(:)<=num2)/77;
66 CI_l_PG_1=CI_l_PG(a+1:b);
67 CI_r_PG_1=CI_r_PG(a+1:b);
68 tv_PG_1=tv_PG(a+1:b);
69
70 figure;
71 plot(days_PG(a+1:b),tv_PG_1);
72 hold on;
73 plot(days_PG(a+1:b),CI_l_PG_1);
74 hold on;
75 plot(days_PG(a+1:b),CI_r_PG_1);
76 legend('TV','lower\_IV','upper\_IV');
77 xlabel('date');
78 ylabel('TV and 95\% CI of IV');
79 title('TV and 95\% CI of IV in October 2008(PG)');
80 datetick('x',26,'keeplimits');
81 xlim([min(days_PG(a+1:b)),max(days_PG(a+1:b))]);
82 hold off;
83
84 %part h
85 annualtv_PG=100*sqrt(252*tv_PG);
86 annualCI_l_PG=annualtv_PG-c*sqrt(2*QIV_PG/N_PG)*100*sqrt(63).*tv_PG.^(-1/2);
87 annualCI_r_PG=annualtv_PG+c*sqrt(2*QIV_PG/N_PG)*100*sqrt(63).*tv_PG.^(-1/2);
88 figure;
89 plot(days_PG,annualtv_PG);
90 hold on;
91 plot(days_PG,annualCI_l_PG);
92 hold on;
93 plot(days_PG,annualCI_r_PG);
94 legend('TV','lower\_TV','upper\_TV');
95 xlim([min(days_PG),max(days_PG)]);
96 xlabel('year');
97 ylabel('annual TV and 95\% CI of annual IV(\%)');
98 title('annual TV and 95\% CI of annual IV(\%)(PG)');
```

```
99 datetick('x','keeplimits');
100 hold off;
101 %Focus on October 2008
102 CI_l_PG_2=annualCI_l_PG(a+1:b);
103 CI_r_PG_2=annualCI_r_PG(a+1:b);
104 tv_PG_2=annualtv_PG(a+1:b);
105 figure;
106 plot(days_PG(a+1:b),tv_PG_2);
107 hold on;
108 plot(days_PG(a+1:b),CI_l_PG_2);
109 hold on;
110 plot(days_PG(a+1:b),CI_r_PG_2);
111 legend('TV','lower\_\_IV','upper\_\_IV');
112 xlabel('date');
113 ylabel('annual TV and 95\% CI of annual IV(\%)');
114 title('annual TV and 95\% CI of annual IV(\%) in October 2008(PG)');
115 datetick('x',26,'keeplimits');
116 xlim([min(days_PG(a+1:b)),max(days_PG(a+1:b))]);
117 hold off;
```

**Exercise 3**

(A) To construct stochastic variance process  $C_{t,i}$ :

- (i) Construct a series number from the standard normal distribution;
- (ii) Compute each  $C_{t,i}$  by iteration;

Here is the plot of time series prices:

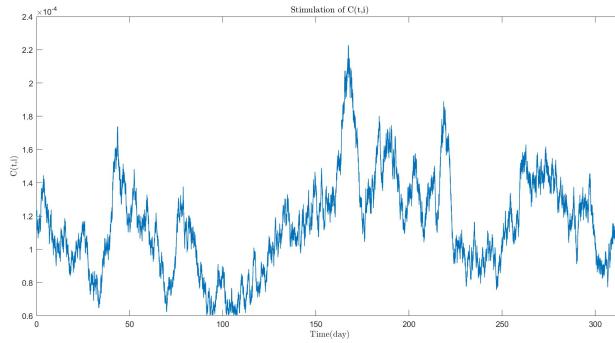


Figure 12: Stimulation of  $C_{t,i}$

The MATLAB code:

```

1 function [c] = JD_sv_ct(ne,T,l,u_c,sigma_c,c0)
2 % simulate stochastic variance process {ct} in a Jump-diffusion model
3 % c is the array of ct between t vary from 0~T
4 % T is total observe time
5 % l is the correlation parameter
6 % ne is the number of observations in every unit of time
7 % sigma is the standard variance of {ct}
8 % u is the mean of {ct}
9 c=[];
10 c(1)=c0; % initial value of ct(at t=0)
11 delta_e=1/ne; % delta_e is the time gap between every observation
12 for j=1:ne*T %observation number from 1 to ne*T
13     r=normrnd(0,1); %construct random variable from standard normal
14         distribution
15     c(j+1)=max(c(j)+l*(u_c-c(j))*delta_e+sigma_c*sqrt(c(j)*delta_e)*r,u_c/2)
16         ; % calculate c(i+1) by iteration
17 end
18 end

```

- (B)** (i)  $IV_t$  is the volatility of log returns' variance  $C_t$  at day t.
- (ii) Since we do not have a continuous  $C_t$  to calculate  $IV_t$ , we need to use an Euler discretization scheme, that is we will simulate the model at a very high frequency and sum all the small interval areas(  $\Delta_n \times C_{t,i}$  ) to estimate the integral of  $C_t$ .

**(C)** The follow are the volatility plots for  $C_t$  ( $IV_t$ ):

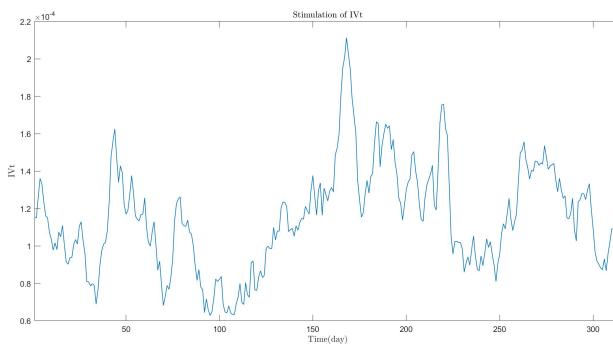


Figure 13: Estimated  $IV_t$

From the figure, we can find the range of  $IV_t$  is  $(0.6 \times 10^{-6}, 0.2 \times 10^{-6})$ . At time around 20, 200 and 250, there are large jump of  $IV_t$ . Comparing with the plot of  $C_{t,i}$  we can find that  $IV_t$  have a similar shape of  $C_{t,i}$  and the curve of  $IV_t$  is smoother than the curve of  $C_{t,i}$ . The method we use to estimate  $IV_t$  is efficient.

**(D)** Here are the plots of  $IV_t$  and  $RV_t$ .

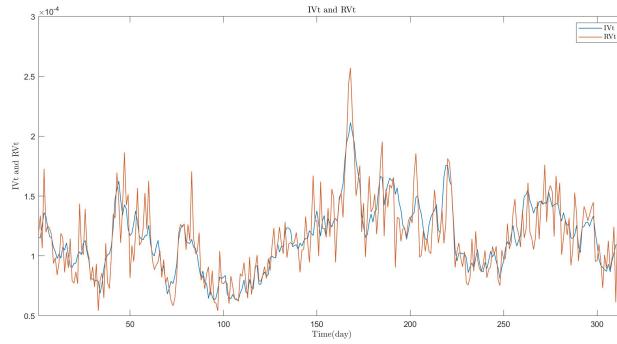


Figure 14:  $IV_t$  and  $RV_t$

From this figure we can find that, RV and IV have the similar shape but RV shows more volatility than IV. The range of RV and IV are  $(0.5 \times 10^{-4}, 2.6 \times 10^{-4})$  and  $(0.6 \times$

$10^{-4}, 2.1 \times 10^{-4}$ ) respectively. RV has show all the movements of IV, which means that RV here is a good estimator of IV.

- (E) We can get the estimated 95% confidence interval by using QIV's estimator  $\hat{QIV}$  and IV's estimator RV. Here is the plot.

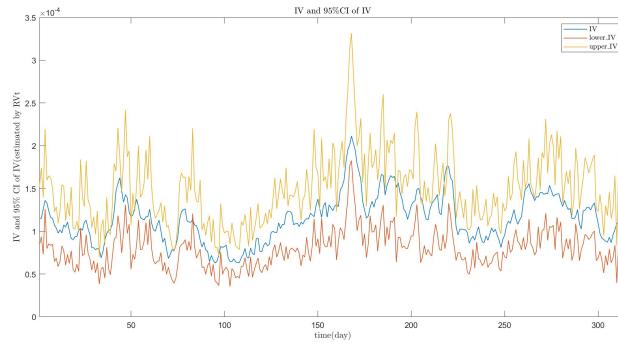


Figure 15:  $IV_t$  and Estimated Confidence Interval

From the figure, we can find that both the lower bound and upper bound of IV have the similar shape of Iv. For most of the sample of IV, IV's values are bounded by estimated confidence interval. According to the figure, we can say that using RV as an estimator of IV to calculate the confidence interval of IV works well.

- (F) (i) The number of days that  $IV_t$ 's values within the confidence interval is 298;  
(ii) The average cover rate is 94.6%;  
(iii) the average cover rate is close to the expected 95%.
- (G) Here is the figure of stock price with jump.

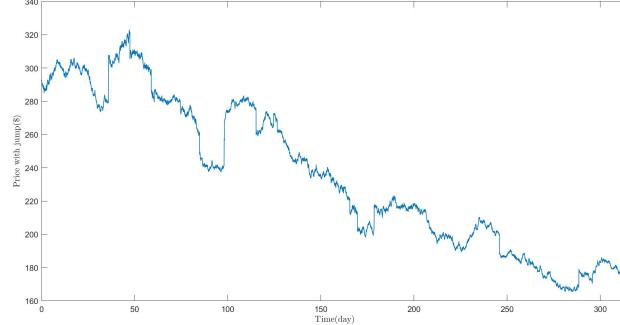


Figure 16: Stimulated stock price with jump

The MATLAB code:

```

1 function [j] =jump1(lamda,T,n,sigma_jump)
2 % simulate a Compound Poisson Process
3 % T is total observe time
4 % j is the array of accumulated jump at a specifit time from 0 to T
5 % n is the number of observations in every unit of time
6 % sigma_jump is the standard variance of jump
7 j=[];
8 j(1)=0; % the initial value of jump (at t=0)
9 N=random('poisson',lamda*T); % the number of total jumps follow a Poisson
    distribution with density lamda*T
10 t=random('uniform',0,T,N,1); % the time of jumps follow a uniform
    distribution from (0,T)
11 y=random('normal',0,sigma_jump,N,1);% the size of jumps follow normal
    distribution with mean=0, sd=(18*sigma)/sqrt(n)
12 for i=1:T*n
13     k=0;
14     for h=1:N % this loop aims at calculate the jumps which happens during
        time period ((i-1)/n,i/n)
15         p=y(h)*indic((i-1)/n,i/n,t(h));
16         k=k+p;
17     end
18     j(i+1)=j(i)+k;% accumulate jumps from time 0 to (i+1)/n
19 end
20 end

```

- (H) Here is the figure of estimated confidence interval of annual IV based on the asymptotic theory.

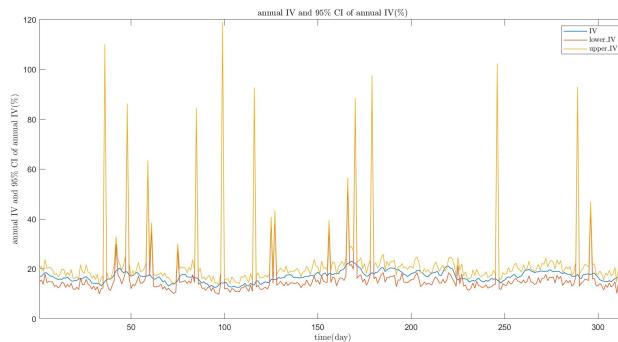


Figure 17: Estimated confidence interval of annual IV(%)

- (i) The coverage rate of the confidence interval is 88.25%;

- (ii) The cover rate when price with jump is smaller than the case when price does not include jumps.
- (iii) The cover rate is what I expected, since when there is jump in stock price, RV will never be a unbiased estimator of IV. So, if we use the biased estimator to calculate the confidence interval of IV, then the confidence interval will not be a good estimated confidence interval and the cover rate will decrease.
- (iv) When we fail to account the jump returns, we may regard RV as a unbiased estimator of IV. However, RV is no longer a unbiased estimator for IV now. This incorrect assumption will affect the confidence interval we calculated: the estimated confidence interval will also bias from the “true” confidence interval, which mean the probability that Iv falls within the estimated confidence interval will not equal(or close) to confidence level anymore.

- (I)** Here is the figure of estimated confidence interval of annual IV based on the asymptotic theory.

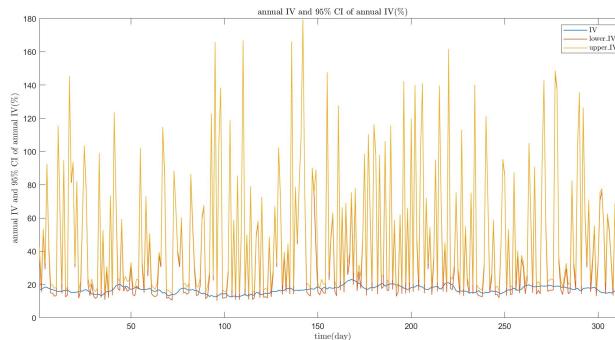


Figure 18: Estimated confidence interval of annual IV(%)

- (i) The coverage rate of the confidence interval is 41.27%;
- (ii) The cover rate when price with jump is smaller than the case when price does not include jumps.
- (iii) The cover rate is what I expected, since when we set  $\lambda = 1$  which means that the probability of day's price includes jump will be 1. In that case, the jump return will constructs major part of daily return. If we still use RV to calculate the confidence interval of IV, the confidence interval will bias more than the case in part H where the jumps are less frequency.
- (iv) When we fail to account the jump returns, we may regard RV as a unbiased estimator of IV. However, RV is no longer a unbiased estimator for IV now. This incorrect assumption will affect the confidence interval we calculated, and the affect will go up as the frequency of jumps goes up(i.e.  $\lambda$  goes up). The estimated

confidence interval will also bias from the “true” confidence interval so the probability that IV falls within the estimated confidence interval will decrease as the frequency of jumps goes up.

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

```

1 % -----EXCERSIZE #3(Monte Carlo Simulation)-----
2 % SET PARAMETERS
3 n=80;
4 T=1.25*252;
5 ne=n*20;
6 l=0.03;
7 u_c=0.011^2;
8 sigma_c=0.001;
9 c0=u_c;
10 x0=log(292.58);
11
12 %Part B(Produce Ct)
13 Ct=JD_sv_ct(ne,T,l,u_c,sigma_c,c0);%calculate Ct
14 te=0:1/ne:T; % construct observation time series
15 figure;
16 plot(te,Ct);% plot of time series Ct(annual percentage)
17 xlabel('Time(day)');
18 ylabel('C(t,i)');
19 title('Stimulation of C(t,i)');
20 xlim([0,T]);
21
22 %Part C(Compute Real IV)
23 Ct1=reshape(Ct(2:end),ne,[]);
24 IVt=sum(Ct1*1/ne);
25 figure;
26 plot(1:T,IVt);% plot of time series Ct(annual percentage)
27 xlabel('Time(day)');
28 ylabel('IVt');
29 title('Stimulation of IVt');
30 xlim([1,T]);
31
32 %Part D(Stimulate Xt and Compute RV)
33 %produce high frequency Xt
34 Xt=JD_sv_xt(ne,T,Ct,x0);
35 %compute coarser frequency log-return
36 Xt1=Xt(1:20:end);
37 lr=reshape(diff(Xt1),80,[]);
```

```
38 %compute RVt
39 RVt=realized_var_day(lr);
40 %plot IVt and RVt
41 figure;
42 plot(1:T,IVt);% plot of time series Ct(annual percentage)
43 hold on;
44 plot(1:T,RVt);
45 xlabel('Time(day)');
46 ylabel('IVt and RVt');
47 legend('IVt','RVt');
48 title('IVt and RVt');
49 xlim([1,T]);
50
51
52
53 %Part E(Estimate the Ci of IVt by RVt)
54 QIVt=QIV(lr,n);
55 c=norminv(0.025);
56 CI_l=RVt-c*sqrt(2*QIVt/n);
57 CI_r=RVt+c*sqrt(2*QIVt/n);
58 figure;
59 plot(1:T,IVt);
60 hold on;
61 plot(1:T,CI_l);
62 hold on;
63 plot(1:T,CI_r);
64 legend('IV','lower\_IV','upper\_IV');
65 xlim([1,T]);
66 xlabel('time(day)');
67 ylabel('IV and 95\% CI of IV(estimated by RVt)');
68 title('IV and 95\%CI of IV');
69 hold off;
70
71 % Part F(Evaluate RVt)
72 num_in=sum(CI_l<=IVt & IVt<=CI_r);
73 cover_rate=num_in/T;
74
75
76 %Part G(Stimulate Xt(with jump) and Compute RV)
77 % SET PARAMETERS
78 sigma_jump=30*sqrt(u_c/n);
79 lamda=20/252;
80
81 Jt=jump1(lamda,T,n,sigma_jump); % calculate value of Jt
```

```
82 XJt=Xt1+Jt;
83 PJt=exp(XJt);
84 t=0:1/n:T;
85 figure;
86 plot(t,PJt);% plot of time series price
87 xlabel('Time(day)');
88 ylabel('Price with jump(\$)');
89 title('Stimulation of Price Xt with Jump(\$)');
90 xlim([0,T]);
91
92
93 %Part H(Compute RV and CI of IV)
94 lr_j=reshape(diff(XJt),80,[]);
95 %compute Rvt and
96 RVt_j=realized_var_day(lr_j);
97 %compute QIV and TV
98 [lr_c,lr_d]=c_d_log_returns(lr_j,n,4);
99 QIVt_j=QIV(lr_c,n);
100 TVt_j=truncated_var_day(lr_c);
101
102 %compute CI
103 annualIVt=100*sqrt(IVt*252);
104 annualRVt_j=100*sqrt(RVt_j*252);
105 annualCI_l_j=annualRVt_j-c*sqrt(2*QIVt_j/n)*100.*sqrt(63./TVt_j);
106 annualCI_r_j=annualRVt_j+c*sqrt(2*QIVt_j/n)*100.*sqrt(63./TVt_j);
107 figure;
108 plot(1:T,annualIVt);
109 hold on;
110 plot(1:T,annualCI_l_j);
111 hold on;
112 plot(1:T,annualCI_r_j);
113 legend('IV','lower\_IV','upper\_IV');
114 xlim([1,T]);
115 xlabel('time(day)');
116 ylabel('annual IV and 95\% CI of annual IV(\%)');
117 title('annual IV and 95\% CI of annual IV(\%)');
118 hold off;
119 % Part F(Evaluate RVt)
120 num_in_j=sum(annualCI_l_j<=annualIVt & annualIVt<=annualCI_r_j);
121 cover_rate_j=num_in_j/T;
122
123 %part I
124 % SET PARAMETERS
125 lamda2=1;
```

```
126
127 Jt2=jump1(lamda2,T,n,sigma_jump); % calculate value of Jt
128 XJt2=Xt1+Jt2;
129 PJt2=exp(XJt2);
130
131
132
133 %Compute RV and CI of IV
134 lr_j2=reshape(diff(XJt2),80,[]);
135 %compute Rvt
136 Rvt_j2=realized_var_day(lr_j2);
137 %compute QIV and TV
138 [lr_c2,lr_d2]=c_d_log_returns(lr_j2,n,4);
139 QIVt_j2=QIV(lr_c2,n);
140 TVt_j2=truncated_var_day(lr_c2);
141
142
143 %compute CI
144 annualIVt2=100*sqrt(IVt*252);
145 annualRvt_j2=100*sqrt(Rvt_j2*252);
146 annualCI_l_j2=annualRvt_j2-c*sqrt(2*QIVt_j2/n)*100*sqrt(63).*TVt_j2.^(-1/2);
147 annualCI_r_j2=annualRvt_j2+c*sqrt(2*QIVt_j2/n)*100*sqrt(63).*TVt_j2.^(-1/2);
148 figure;
149 plot(1:T,annualIVt);
150 hold on;
151 plot(1:T,annualCI_l_j2);
152 hold on;
153 plot(1:T,annualCI_r_j2);
154 legend('IV','lower\_IV','upper\_IV');
155 xlim([1,T]);
156 xlabel('time(day)');
157 ylabel('annual IV and 95\% CI of annual IV(\%)');
158 title('annual IV and 95\% CI of annual IV(\%)');
159 hold off;
160 % Part F(Evaluate Rvt)
161 num_in_j2=sum(annualCI_l_j2<=annualIVt & annualIVt<=annualCI_r_j2);
162 cover_rate_j2=num_in_j2/T;
```