Faggin Alberto

September 13, 2023

1 Introduction to the Problem

The purpose of this homework is to test and highlight the knowledge and skills acquired during the course, specifically related to high-frequency data. To this end, a file containing five years of trades for the CRM stock with a frequency of 1 second (for the seconds without transactions, no market activity was detected) has been made available. Specifically, this homework pertains to the *Salesforce* stock, an American cloud computing company based in San Francisco and belonging to the SP 500.
The paper will be divided into four sections:

- Observation of price and return series
- Detection of jumps and application of tests
- Estimation of models for forecasting realized volatility
- Implementation of a quantitative indicator to build a trading strategy

This division has been designed in relation to the topics covered during the course.

2 Observation of Price and Return Series

The section presented here aims to observe the evolution of prices at one-second intervals and at 5-minute intervals, as well as the 5-minute returns. For the purpose of their representation, the management of missing data will be crucial. Additionally, it will be of interest to observe the signature plot (frequencies from 1 to 600 seconds) that will compare realized volatility with a robust estimator of microstructural noise.

2.1 Evolution of Prices and Returns

The first fundamental aspect is the data reading, which has the following structure:

V1	V2	V3	V4	V5	V6	V7
01/02/2018	09:30:00	102.8800	103.0000	102.880	102.990	70798
01/02/2018	09:30:01	102.7900	102.7900	102.640	102.790	3095
01/02/2018	09:30:02	102.7700	102.7700	102.770	102.770	200
01/02/2018	09:30:15	102.6477	102.6477	102.620	102.620	300
01/02/2018	09:30:22	102.6300	102.6300	102.630	102.630	100

Table 1: Available Data for CRM Stock

As you can see, the data provide information regarding the date and time of the trade, opening price, highest price, lowest price, closing price, and traded volume. An extremely important aspect to note is that there are not observations for every second. For the analysis to be developed, it is of interest to handle these missing data in order to obtain equispaced data with an observation for every second. To achieve this, a new dataset was created from the available data, which includes information about the date and time of the trade, as well as the opening and closing prices. Additionally, the created dataset was cleaned from values outside the trading hours (09:30:00 - 15:59:59), and in case of missing data, the lack of trades was considered as a price invariant. Specifically, regarding the handling of missing data, a for loop was used where:

- I created a vector of length 23,400 observations (one for each second) with missing values (NA).
- I replaced the NA values with the observed values.
- I replaced the NA values with the price values from the previous observation using the *na.locf* function from the *zoo* package.

- I repeated the loop for all the considered days.

After this initial data cleaning, it is possible to observe the price evolution for one-second and 5-minute interval data:

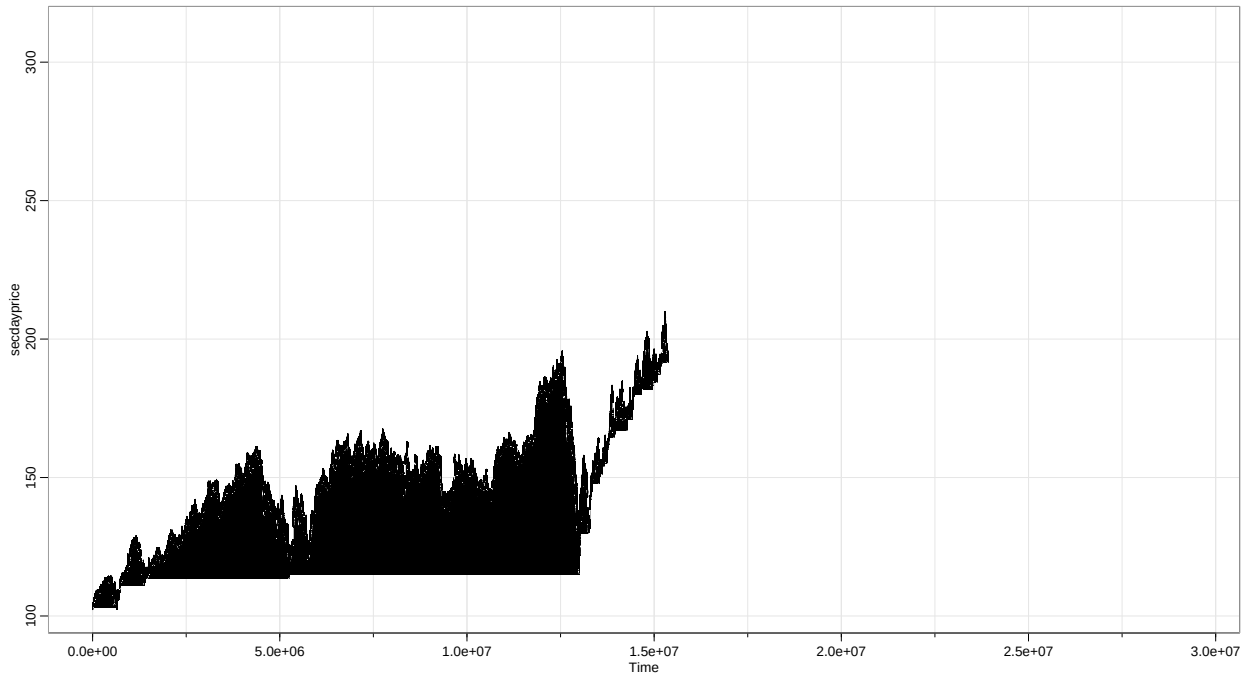

Figure 1: Price Series for 1-second Data

Figure 2: Price Series for 5-minute Data

From the price representations, it can be observed that the stock experienced a shock at the beginning of 2020, coinciding with the outbreak of the Covid-19 pandemic. Subsequently, prices show an uptrend until the end of December 2021, when news of vaccines was released, followed by a significant decline. This trend is likely due to the widespread adoption of remote work and, consequently, the increased use of cloud computing in people's daily lives. Despite the observed price variations, the primary focus when modeling the price series of a financial instrument, due to its non-stationarity, is on considering returns. For this reason, the series of 5-minute returns for the CRM stock is represented:

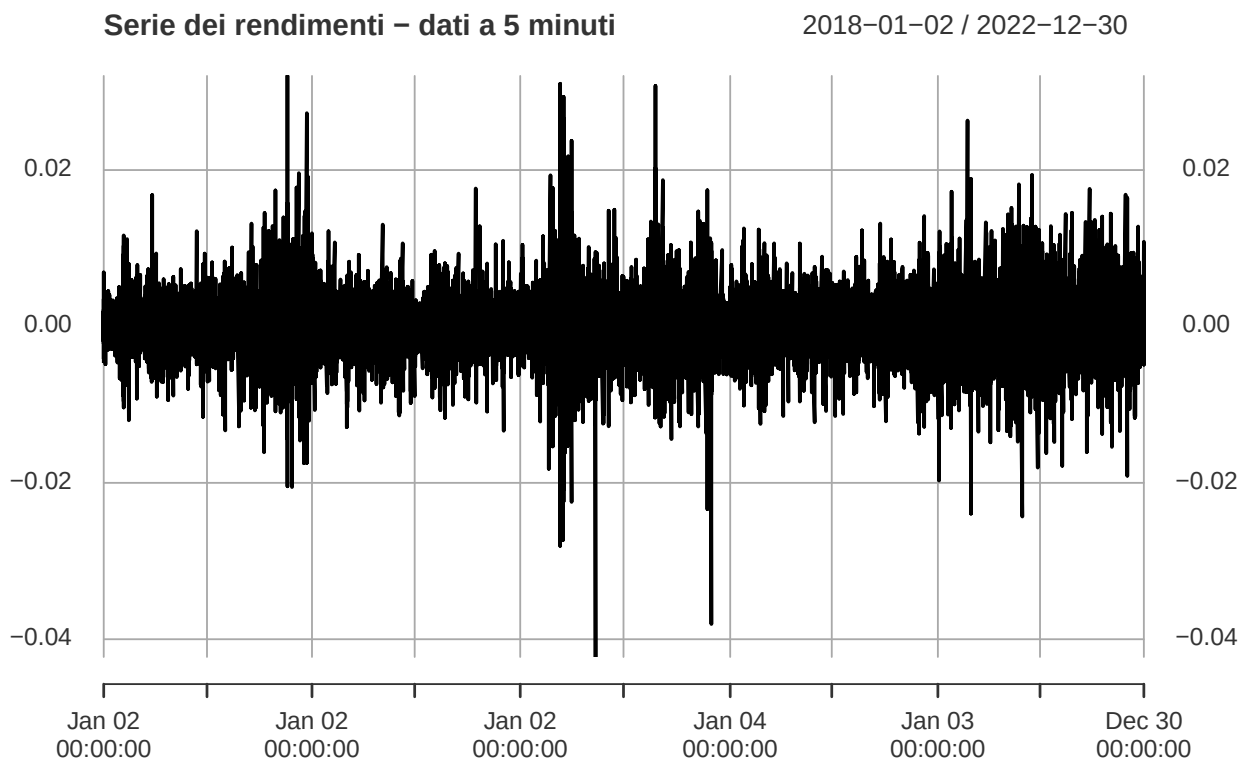

Figure 3: 5-Minute Returns

2.1.1 Codice R:

```
z = read.table("CRM.txt",sep = ",")
POPC = cbind(z$V3,z$V6)
dt = paste(z$V1,z$V2,sep = " ")
dt = as.POSIXct(dt,tz = "","%m/%d/%Y %H:%M:%OS")
db = xts(x = POPC,order.by=dt)
dbtday = db["T09:30:00/T15:59:59"]

datenow = as.Date("2017-12-31")
secdata = NULL
secprice = NULL
secday = NULL
for(i in 1:1826) {
    datenow=datenow+1
    st=paste(as.character(datenow),"9:30:00",sep=" ")
    ed=paste(as.character(datenow),"15:59:59",sep=" ")
```

```
    test=seq(as.POSIXct(st),as.POSIXct(ed),by="sec")
    testx.NA=xts(matrix(NA,23400,1),test)
    selday=dbtday[as.character(datenow)]
    N=nrow(selday)
    if(N>0) {
        p = c( as.numeric(selday[1,1]),as.numeric(selday[2:N,2]))
        testx.NA[index(selday)]=p
        price = na.locf(testx.NA, xout=test )
        dtx.NA=as.numeric(price)
        secprice=paste(secprice, dtx.NA)
        secdayprice = vec(secprice
    }
}

tsplot(secdayprice)
```

2.2 Estimated realized volatility

To obtain an estimate of realized volatility, i.e. the integrated variance in a given day or time interval, it is possible to start from the following Browian process which describes the price movement:

$$dp_t = \mu p_t d_t + \sigma_t dW_t \tag{1}$$

The process proposed here can be rewritten in the form:

$$p_t = p_0 + \int_0^t \mu_\tau d\tau + \int_0^t \sigma_t dW_t \tag{2}$$

Where $\int_0^t \sigma_t dW_t$ represents the estimate of the integrated volatility ($IV(0,t)$) in the interval [0,t]. If we assume to divide the interval [0,t] into n sub-intervals and assuming that t measures days, it is possible to obtain the following relationship:

$$QV(0,t) = IV(0,t) = \int_0^t \sigma_t dW_t \tag{3}$$

In particular

$$QV_n(t-1,t) = \sum_{i=1}^{n} (p_{t-1+i\Delta} - p_{t-1+(i-1)\Delta}) \tag{4}$$

which under the assumptions just made is also called realized volatility. For the series in question it is possible to estimate the volatility realized for one-minute frequency data, which can be observed in the following graph.

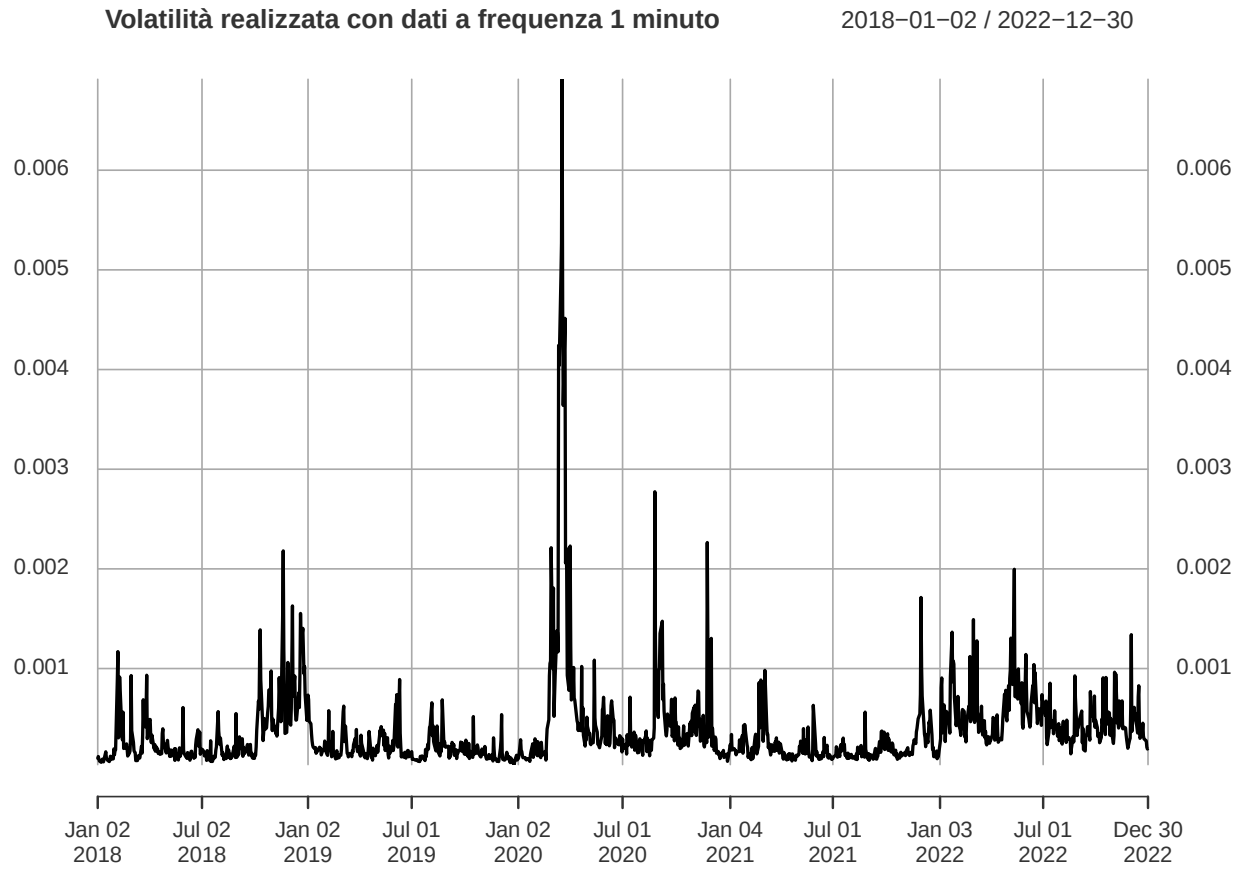

Figure 4: Realized volatility - one minute frequency data

It is interesting to observe how the highest peak corresponds to the beginning of the Covid-19 pandemic, and subsequently the stock stabilizes even if a higher variability is maintained compared to the previous period. In reality, however, the true price value is not observed due to a microstructure error. The price that is observed is not the true price, but is "tainted":

$$\tilde{p}_{t,i} = p_{t,i} + \epsilon_{t,i} \tag{5}$$

where the microstructure error $\epsilon_{t,i}$ $i.i.d.(0, \omega^2)$ is independent of $p_{t,i}$. In order to be able to handle this error, you can use two methods:

is lower frequencies in order to reduce microstructural noise

are robust estimators

Using lower frequencies would lead to a loss of information, due to fewer data. For this reason, the second method is taken into consideration: robust estimators, where, specifically,

the two-time scale estimator will be used. You can compare the different estimates using the signature diagram:

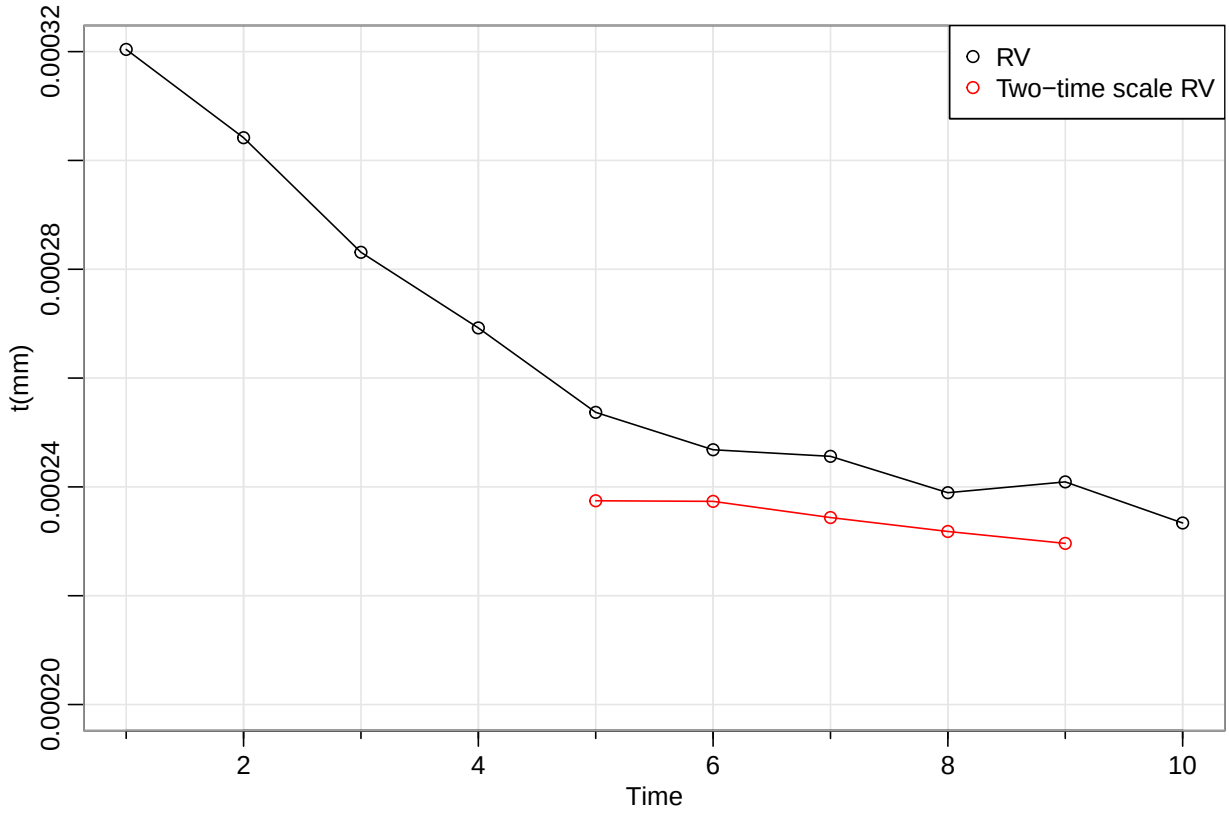

Figure 5: Characteristic plot

Looking at the graph it is possible to notice the presentation of microstructural noise collected by the estimator at two time scales.

2.2.1 R code:

```
RV=colSums(secdata^2)
m=c(300)
N=dim(secdata)[1]
T=dim(secdata)[2]
mm=median(RV)
d=diag(1,N/m,N/m) %x% t(rep(1,m))
r=d %*% secdata
N=dim(r)[1]
```

```
RV=colSums(r^2)

tsplot(RV, main = " Volatilità realizzata con dati a frequenza 1 minuto")

RV=colSums(secdata^2)
m=c(2,5,10,30,60,120,180,300,600)
N=dim(secdata)[1]
T=dim(secdata)[2]
mm=median(RV)

for (i in m){
d=diag(1,N/i,N/i) %x% t(rep(1,i))
r=d %*% secdata
RVl=colSums(r^2)
RV=cbind(RV,RVl)
mm=cbind(mm,median(RVl))
}

kall=c(30,60,120,180,300)
RV2=NULL
tw = NULL
for (j in 1:5){
    k=kall[j]
    m=N/k
    mav=apply(secdata,2,rollsum,k)
    RV1=NULL
    for (i in 1:k){
    id=seq((i+1),(N-k+1),by=k)
    locr=mav[id,]
    RV1=cbind(RV1,colSums(locr^2))
    }
    RV1=rowMeans(RV1)
    RV2=cbind(RV2, RV1 - (1/k)*RV[,1])
    tw=cbind(tw,median(RV2))
}

tw1 = c(0,0,0,0,tw)
```

```
tw_sub <- subset(tw1,  seq_along(tw1)>=5)

tsplot(t(mm), type = "o", col  = "black", ylim = c(0.00020,0.00032))
lines(5:length(tw1), tw_sub, type = "o", col = "red")
legend("topright", legend = c("RV", "Two-time scale RV"),
        col = c("black", "red"), pch = c(1, 1))
```

3 Jump detection and testing application

When looking at the time series of prices, it is sometimes possible to notice the presence of discontinuities in the data. This, however, may not be due to a lack of observations, but due to jumps, understood as a sharp movement in prices. When describing price movement using the concept of Browian process (1), this aspect was not considered. For this reason it is necessary to add a discontinuous component that captures this effect

$$r_t = \int_0^t \mu_\tau d\tau + \int_0^t \sigma_t dW_t + \sum_{i=1}^{N_t} J_i \tag{6}$$

where $\sum_{i=1}^{N_t} J_i$ represents the discontinuous part. Given the importance of this component, it is useful to identify, through some tests, the presence of jumps in the available historical series. Furthermore, in this section, we will focus on the implementation of the Lee-Mykland jump test (not covered in class), which will be compared with the BNS test.

3.1 BNS Test

To identify the jumps you can use the BNS test which is based on the BPV[1], which exploits the intuition that the probability of observing two consecutive jumps is very small. It is interesting to note how in the presence of jumps the BPV estimator is consistent; while the RV estimator is biased. For this reason, the consistency of the two statistics will be exploited to identify the jumps. The proposed BNS test is

$$BNS_t = \frac{\sqrt{n}(RV_t - BPV_t)}{\sqrt{\theta QPQ_t}} \to N(0,1) \tag{8}$$

Where the test statistic only observes the upper tail of the $N(0,1)$ since if there are jumps $RV > BPV$. Furthermore, $\alpha = 0.001$ is considered as the critical value associated with the test statistic, since observing jumps is a rare event. In this context, the test detects 68 hops. It is important to underline that in the presence of jumps the discontinuous component can be estimated based on the difference between RV_t and BPV_t; while the component continues for the difference: $RV_t - J_t^2$. For this reason, it may be useful to look at the graph of the differentials between the realized variance and the BPV estimator.

[1]

$$estimatorBPV_t^n = \mu_1^{-2} \frac{n}{n-1} \sum_{i=2}^{n} |r_{t-1+i\Delta}| |r_{t-1+(i-1)\Delta}| \tag{7}$$

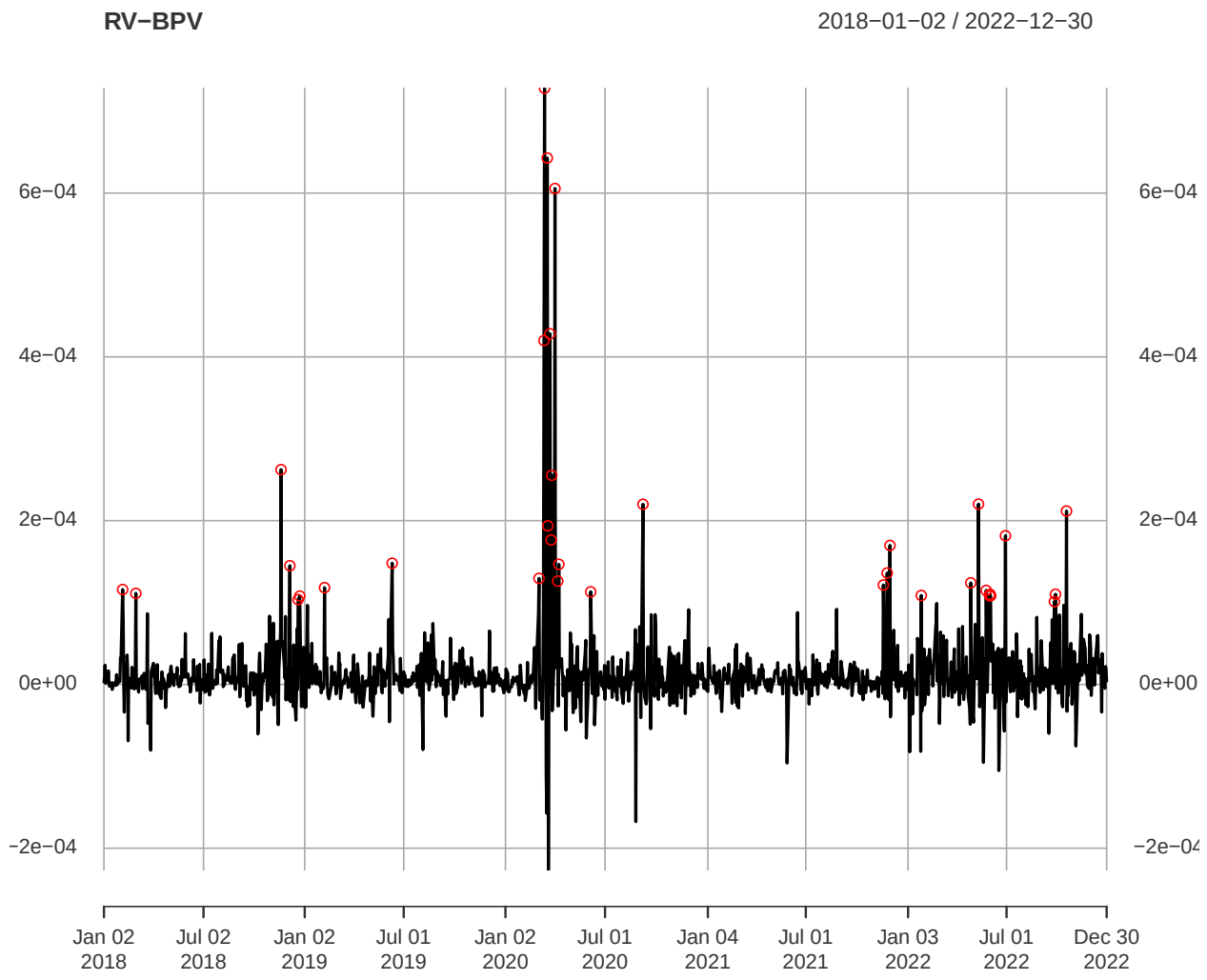

Figure 6: RV vs BPV

The differences are generally positive in the presence of jumps, but negative differences can also be observed in the absence of jumps. This is because the BPV misses an observation since it takes the absolute values of the consecutive returns. Furthermore, there are red dots in the graph, indicating the presence of jumps.

3.1.1 R code:

```
ar=abs(r)
arr=ar[2:N,] * ar[1:(N-1),]
mu1=sqrt(2/pi) # calcolo μ
BPV=colSums(arr)*(N/(N-1))*(mu1)^(-2) # applico la formula
```

```
tsplot(RV-BPV)

theta=pi*pi/4+pi-5
ar4=ar[4:N,]*ar[3:(N-1),]*ar[2:(N-2),]*ar[1:(N-3),]
QPQ=colSums(ar4)*N*(mu1^(-4))

T=NROW(RV)
V2=QPQ/(BPV^2)
V1=(V2>1)*V2+(V2<=1)*1
BNS=(((RV-BPV)/RV)*sqrt(N))/sqrt(theta*V1)
tsplot(BNS)
a=qnorm(0.999)
sum(BNS>a)
```

3.2 Jumps at intra-daily level

A useful approach to identify jumps at a daily level, considering the dynamics due to the presence of overnight information, is that proposed by Boudt, Croux and Laurent in 2011, who, considering a simplified intraday return generating model, filtering the returns:

$$r_{i,t} = f_{i,t} s_{i,t} z_{i,t} + a_{i,t} \tag{9}$$

Where

- ${f_{i,t}}^2$ is a deterministic component that influences volatility
- ${s_{i,t}}^3$ is a constant volatility component within a window
- $z_{i,t}$ is an i.i.d. innovation. with zero mean and unit variance (noise)
- $a_{i,t}$ is a jump component

Thanks to this approach, it was possible to re-estimate the BNS test with the filtered residuals, obtaining 18 jumps. Given the high difference compared to the BNS test previously estimated with the unfiltered residuals, it may be interesting to observe the comparison of the filtered and unfiltered (one-minute) returns:

[2]The authors of this method, among all the estimation methods, succeed the following: $\hat{f_{i,t}} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \tilde{r_{i,t}^2}}$, where the estimated value must subsequently be standardized in order to calculate the returns filtered by the periodic component.

[3]For $s_{i,t}$ it is necessary to consider the presence of jumps, therefore the rescaled BPV_t estimator for n is used. Consequently $s_{i,t}$ is constant within the day.

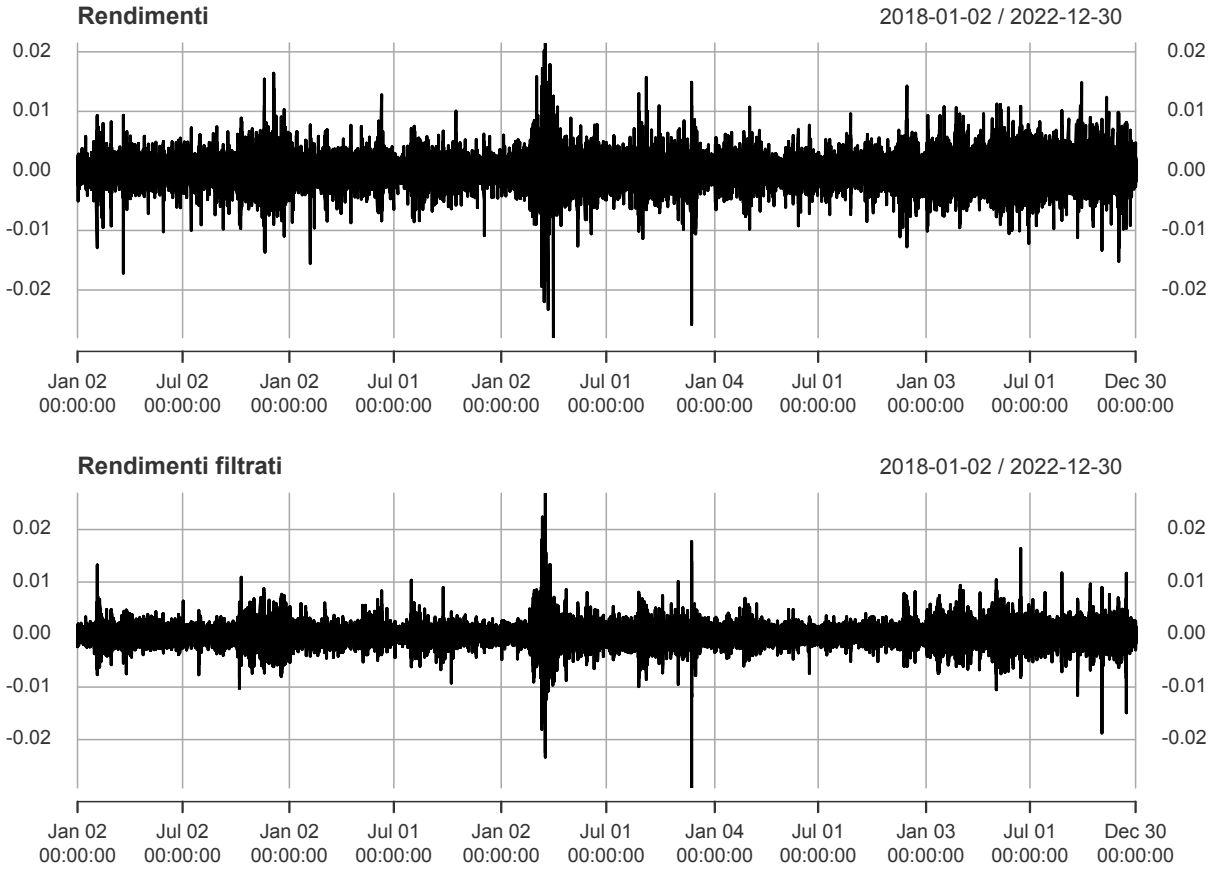

Figure 7: Number of jumps when opening

From the graph it is possible to see how the variability around the filtered yields is lower than that of the unfiltered ones. This indicates how it is possible to grasp intraday volatility due to the presence of overnight information. In this sense, it may be useful to compare the decomposition between the continuous and discontinuous components obtained with the filtered and unfiltered returns.

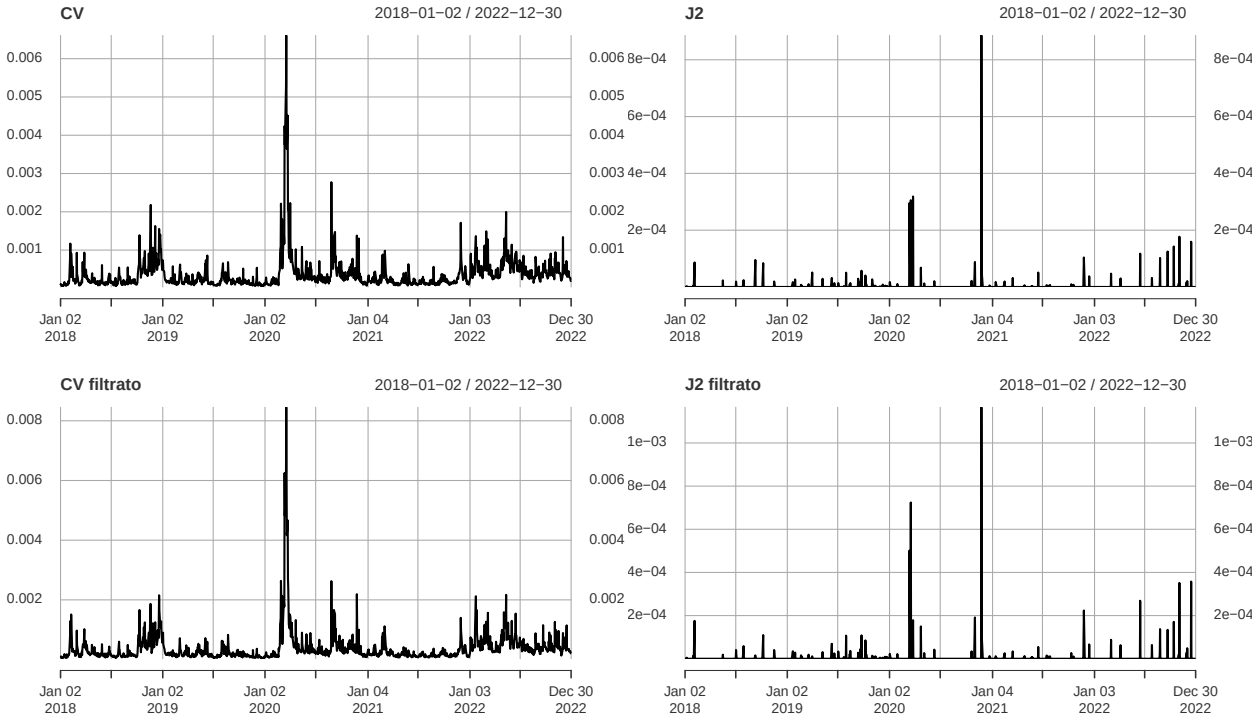

Figure 8: Comparison of continuous and discontinuous components

From this representation it can be seen how the values of the discontinuous component of the filtered returns are slightly higher than that of the unfiltered returns. This indicates that it is possible to capture the overnight effect in the discontinuous component.
Furthermore, to identify jumps at the intraday level, the test proposed by Andersen, Bollerslev and Dobrev in 2007 was used, who roar to identify jumps on the basis of

$$|r_{t,i}| > \Phi(1 - \frac{\beta}{2})\sqrt{\frac{1}{n}BPV_t} \tag{10}$$

identifying the returns in the extreme tails as jumps. The following graphs are particularly explanatory:

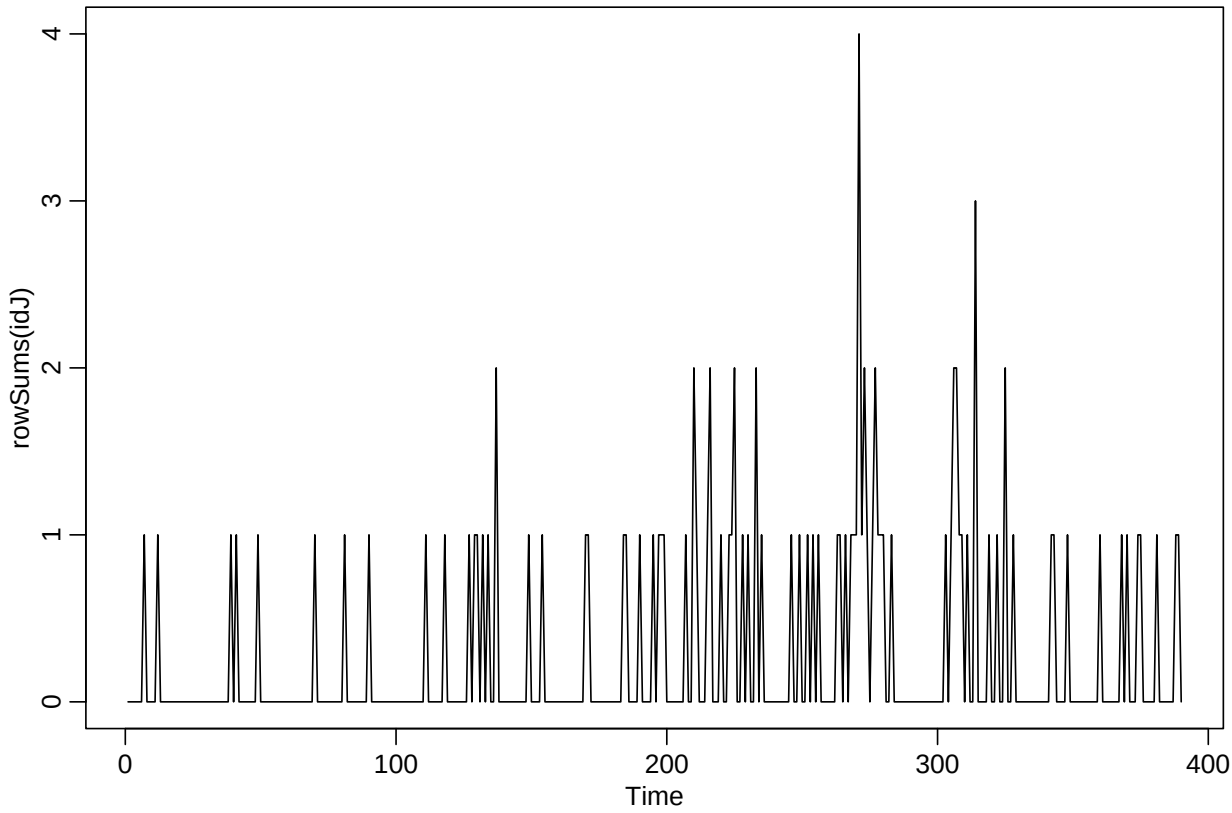

Figure 9: Number of daily hops

The graph shown shows that there are 99 jumps at intraday level. This is a very common phenomenon, often caused by the disclosure of information that occurred during the hours in which the stock exchange remains closed. Although there may be price variations, in reality these variations are quickly reabsorbed when the market opens. The phenomenon described here is defined as "*opening jumps*", and it is therefore important to underline how important it is to filter the returns as proposed by Boudt, Croux and Laurent.

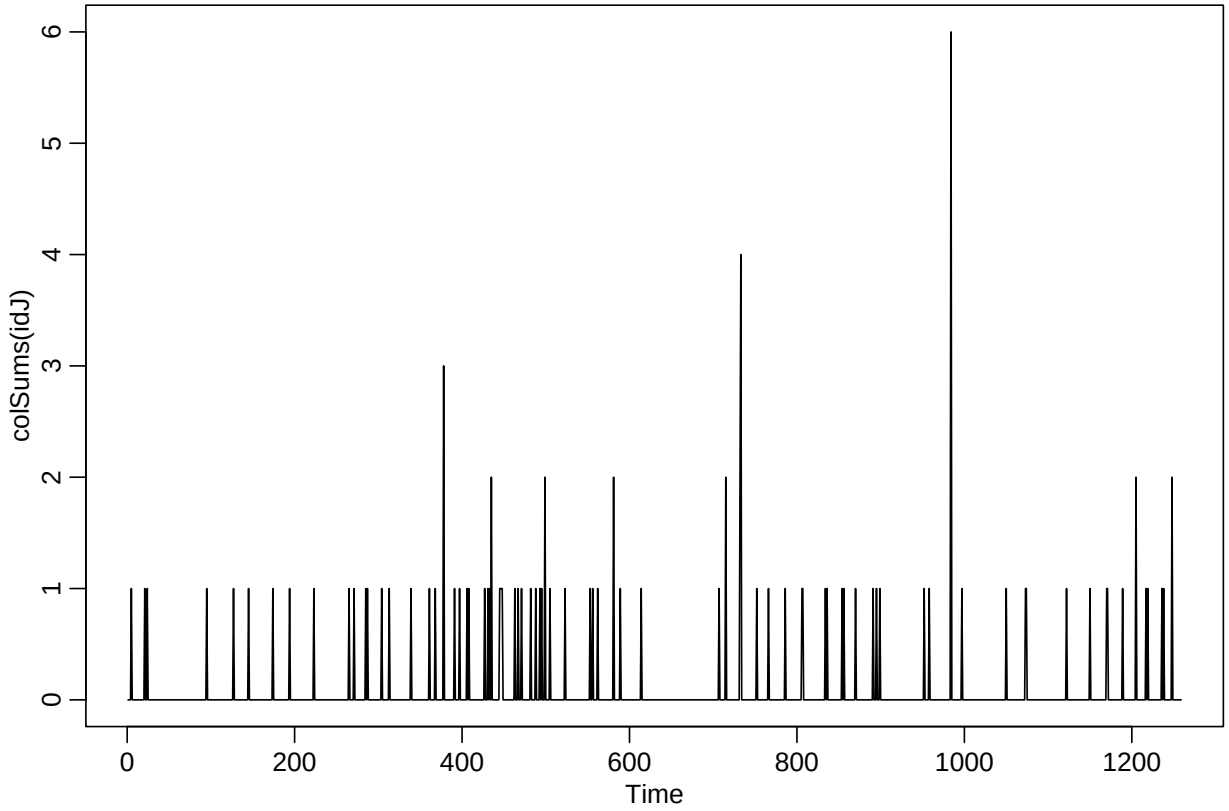

Figure 10: Number of intra-daily level hops

Finally, this representation shows the days on which there are jumps. It is interesting to observe, given the rarity of this phenomenon, how it was possible to observe more than two jumps in some days.

3.2.1 R code:

```
ar=abs(rs2)
arr=ar[2:N,] * ar[1:(N-1),]
mu1=sqrt(2/pi)
BPV=colSums(arr)*(N/(N-1))*(mu1)^(-2)
BPVs=t(as.matrix(sqrt(BPV/N)))
m1=matrix(rep(BPVs,N),ncol=ncol(BPVs),byrow=TRUE)
rs3=rs2/m1
beta=1-(1-(10^(-5)))^(1/N)
idJ=abs(rs3)>qnorm(1-beta/2)
```

```
plot.ts(rowSums(idJ))
plot.ts(colSums(idJ))
```

3.3 Test for Lee-Mykland jumps

The test introduced by Lee & Mykland aims to capture the presence of jumps more accurately than the BNS test. In particular their research shows how, unlike the BNS test, it is possible to determine: how many jumps have occurred, whether the jumps are negative or positive, what time of day the jumps occurred and how large each jump is; finally, the two researchers compared the performance of the two tests in terms of overall probability of success in detecting actual jumps within a given interval, observing how the test they proposed surpasses the BNS test.
The statistic *L(i)*, which tests at time t_i whether there is a state jump in the interval from t_{i-1} to t_i , is defined as:

$$L(i) = \frac{logS(t_i)/S(t_{i-1})}{\hat{\sigma(t_i)}} \tag{11}$$

Where

$$\hat{\sigma(t_i)}^2 = \frac{1}{K-2} \sum_{j=i-K+2}^{i-1} |logS(t_j)/S(t_{j-1})||logS(t_{j-1})/S(t_{j-2})| \tag{12}$$

Specifically, the statistic is formulated by taking the ratio between the last return in a window and the instantaneous volatility estimated by BPV using window returns[4] same.
The research observes that in the absence of jumps the statistics approximately follows a normal distribution; while in the presence of jumps it presents the following distribution:

$$\frac{max_{i \in A_n}|L(i) - C_n|}{S_n} \rightarrow \xi \tag{13}$$

where ξ is defined as a Gumbel distribution: $P(\xi \leq x) = exp(-e^{-x})$. Where:

- $$C_n = \frac{(2\log n)^{\frac{1}{2}}}{c} - \frac{\log \pi + \log(\log n)}{2c(2\log n)^{\frac{1}{2}}} \tag{14}$$
- $$\frac{1}{c(2\log n)^{\frac{1}{2}}} \tag{15}$$

[4]The study shows different optimal K window values for a week, a day, one hour, 30 minutes, 15 minutes and 5 minutes, providing values of 7, 16, 78, 110, 156 and 270 respectively. An optimal window for 1 minute yields has not been observed in the literature, for this reason it will be considered $K = 270$.

Therefore, in the presence of jumps, those values $P(\xi \leq \beta) = exp(-e^{-\beta}) \rightarrow \frac{max_{i \in A_n}|L(i)-C_n|}{S_n} > 6.907255$, where $\beta = -\log(-\log(0.999))$.
By applying the LM test, 48 jumps were obtained, unlike the BNS test which had 68. Instead, by standardizing the returns and observing the intraday jumps, the test measures 163 jumps. This value is higher than what was observed for the intraday vase of 99 jumps. This aspect may be dictated by an estimation error of the LM test.

3.3.1 R Code:

```
    mu1 = sqrt(2/pi)
    k = 270
    r_BPV = abs(r)
    n = T*k
    beta = -log(-log(0.999))
    S = 1/(mu1*(sqrt(2*log(n))))
    C = (2*log(n))^(1/2)/mu1 - (log(pi)+log(log(n)))/
        (2*mu1*((2*log(n))^(1/2)))
    a = NULL
    rend = NULL
    sigma_hat = NULL
    dist_i = NULL
    L_matrix = NULL
    test_LM_i = NULL
    test_LM = NULL
    for (i in k:N) {
      rend = (r_BPV[(i-k+2):(i-1),]*r_BPV[(i-k+1):(i-2),])
      a = colSums(rend)
      sigma_hat = sqrt((1/(k-2))*a)
      L = r[i,]/sigma_hat
      L_matrix = cbind(L_matrix, L)
      dist_i = (abs(L)-C)/S
      test_LM_i = sum(dist_i>beta)
      test_LM = c(test_LM,test_LM_i)
    }
sum(test_LM)

    m2=matrix(rep(nM,T),ncol=T,byrow=FALSE)
```

```
rs2=r/m2

for (i in k:N) {
rend = (r_BPV[(i-k+2):(i-1),]*r_BPV[(i-k+1):(i-2),])
a = colSums(rend)
sigma_hat = sqrt((1/(k-2))*a)
L = rs2[i,]/sigma_hat
L_matrix = cbind(L_matrix, L)
dist_i = (abs(L)-C)/S
test_LM_i = sum(dist_i>beta)
test_LM = c(test_LM,test_LM_i)
}
sum(test_LM)
```

4 Estimating models for forecasting

4.1 HAR - RV model

The idea behind this model lies in the different investment time horizons of financial operators. For this reason, the model explains three different time horizons, expressing the heterogeneity of the operators: daily, weekly and monthly.

$$RV_t = \alpha + \beta_D RV_{t-1} + \beta_W RV_{t-1:t-5} + \beta_M RV_{t-1:t-22} + \epsilon_t \tag{16}$$

Where:

$$RV_{t-1:t-5} = \frac{1}{5}\sum_{j=1}^{5} RV_{t-j} RV_{t-1:t-22} = \frac{1}{22}\sum_{j=1}^{22} RV_{t-j} \tag{17}$$

The model parameters are simply estimated via OLS and the output of the model estimate for the available data is:

```
     Call:
lm(formula = RV ~ RV1 + RV5 + RV22)

Residuals:
        Min 1Q Median 3Q Max
-0.0020349 -0.0000855 -0.0000299 0.0000504 0.0031875

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.147e-05 1.156e-05 4.451 9.32e-06 ***
RV1 6.475e-01 3.237e-02 20.001 < 2e-16 ***
RV5 2.314e-01 4.296e-02 5.386 8.60e-08 ***
RV22 -1.475e-02 3.288e-02 -0.449 0.654
---
Significant codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0002691 on 1233 degrees of freedom
   (22 observations deleted due to missingness)
Multiple R-squared: 0.6837, Adjusted R-squared: 0.683
F-statistic: 888.6 on 3 and 1233 DF, p-value: < 2.2e-16
```

The goal of these models is to make predictions (they are predictions of variances), so you expect to find positive values. As can be seen, the estimates of β_D and β_W have positive

values; while the value of β_M, relating to the monthly delay, is not significant and negative, presenting an anomalous value. In this context, the literature suggests replacing this value with the latest forecast, or, a forecast equal to the historical average[5]. In this regard, it may be interesting to observe the QQ - Plot. We can notice a behavior that leads to the rejection of the hypothesis of normality, since the points are not all distributed on the bisector of the first and third quadrants (red line).

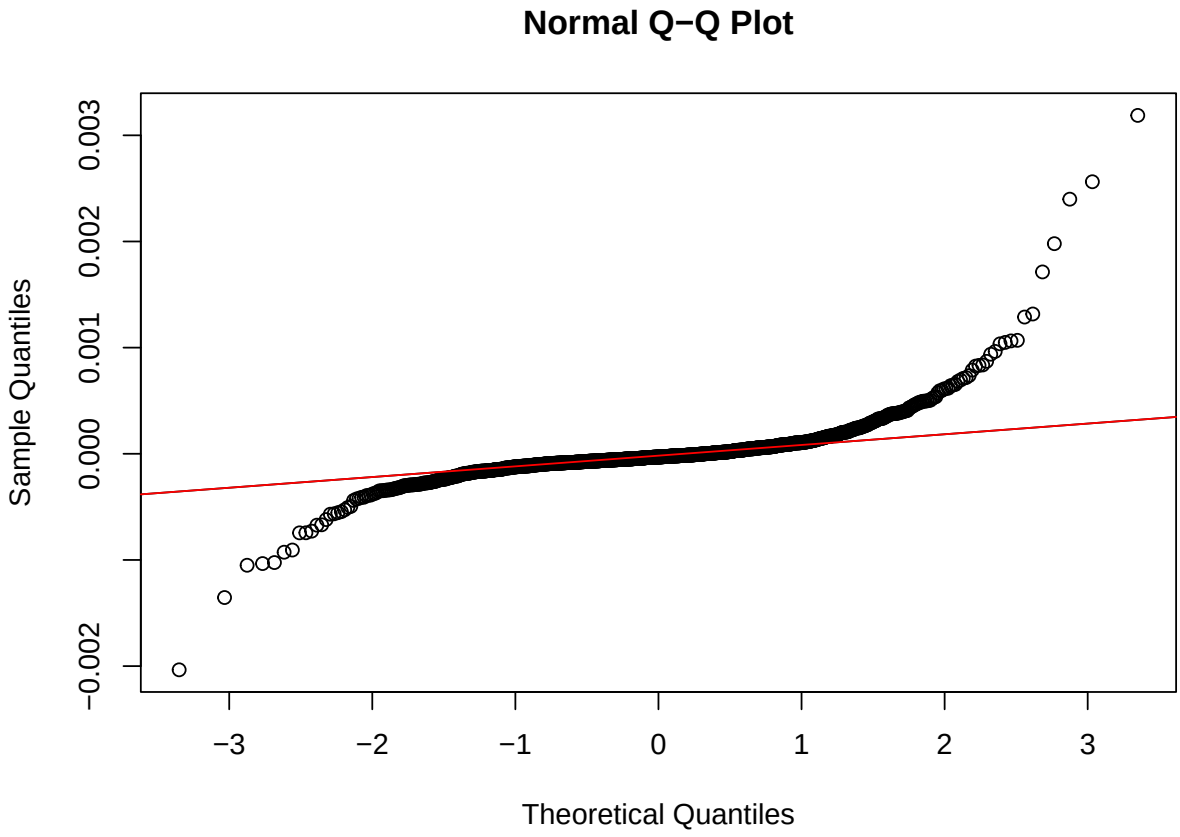

Figure 11: QQ - Plot

The distribution of RV_t is clearly non-Gaussian. One approach to deal with this situation is to perform a transformation of RV_t, where the logarithmic one is among the most common. The output of R's estimation of the model via logarithmic transformation is presented below.

```
    Call:
lm(formula = log(RV) ~ RV1l + RV5l + RV22l)
```

[5]These substitutions, in practice, are also carried out in the case of off-scale values, in the sense that the forecast value of the variance is excessively high

```
Residuals:
      Min 1Q Median 3Q Max
-1.40439 -0.30207 -0.03449 0.26467 2.12823

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.76662 0.17448 -4.394 1.21e-05 ***
RV1l 0.50788 0.03323 15.285 < 2e-16 ***
RV5l 0.33537 0.04735 7.084 2.36e-12 ***
RV22l 0.06336 0.03669 1.727 0.0844 .
---
Significant codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4348 on 1233 degrees of freedom
   (22 observations deleted due to missingness)
Multiple R-squared: 0.6732, Adjusted R-squared: 0.6725
F-statistic: 846.8 on 3 and 1233 DF, p-value: < 2.2e-16
```

The model thus estimated shows how the coefficient of β_M is now positive. By also observing the QQ-Plot it is possible to notice how the residuals are closer to the normal distribution.

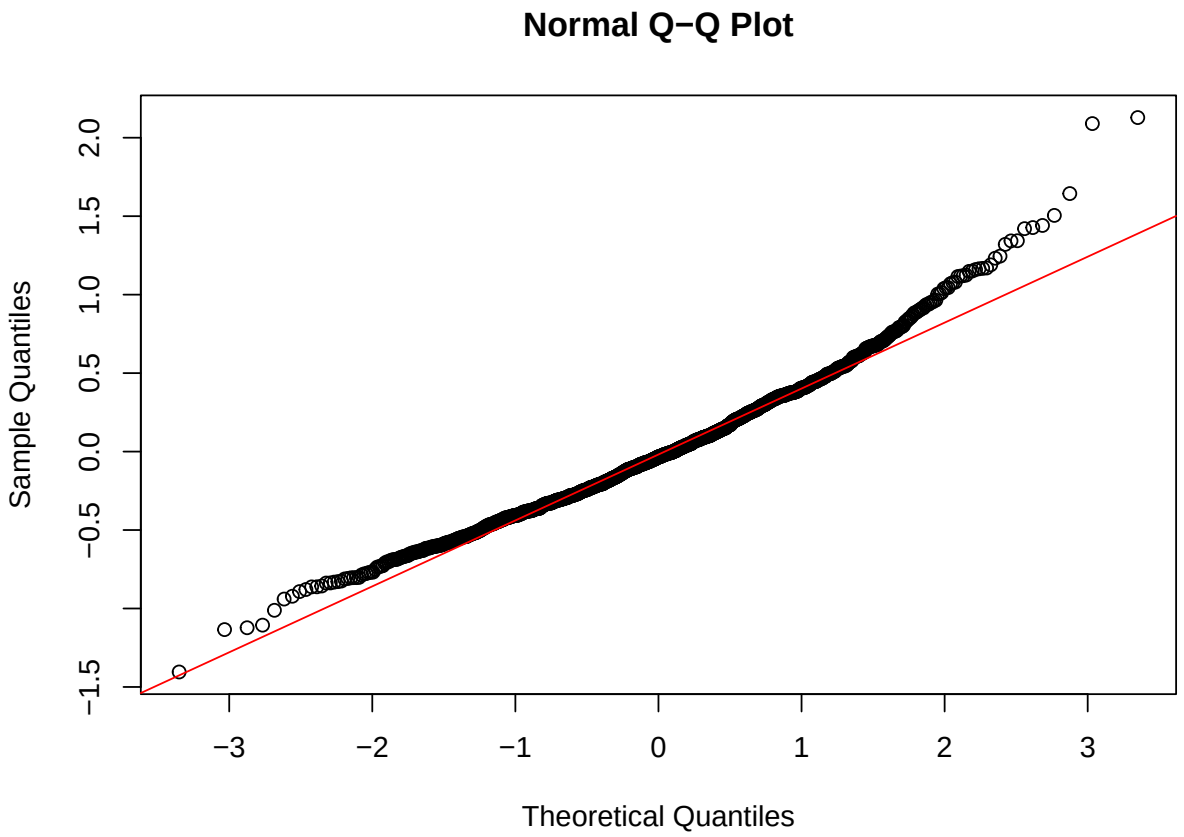

Figure 12: QQPlot, transformation RV_t

In addition to the representation of the QQ-Plots, it may be interesting to observe the correlogram of the squared residuals. From the figure we can see how the model, in which the logarithmic transformation is applied to RV_t, manages to remove a large part of the serial correlation; in fact, in the second graph we can see how the residuals do not show dependence on their delays, since the values are within the Bartlett bands.

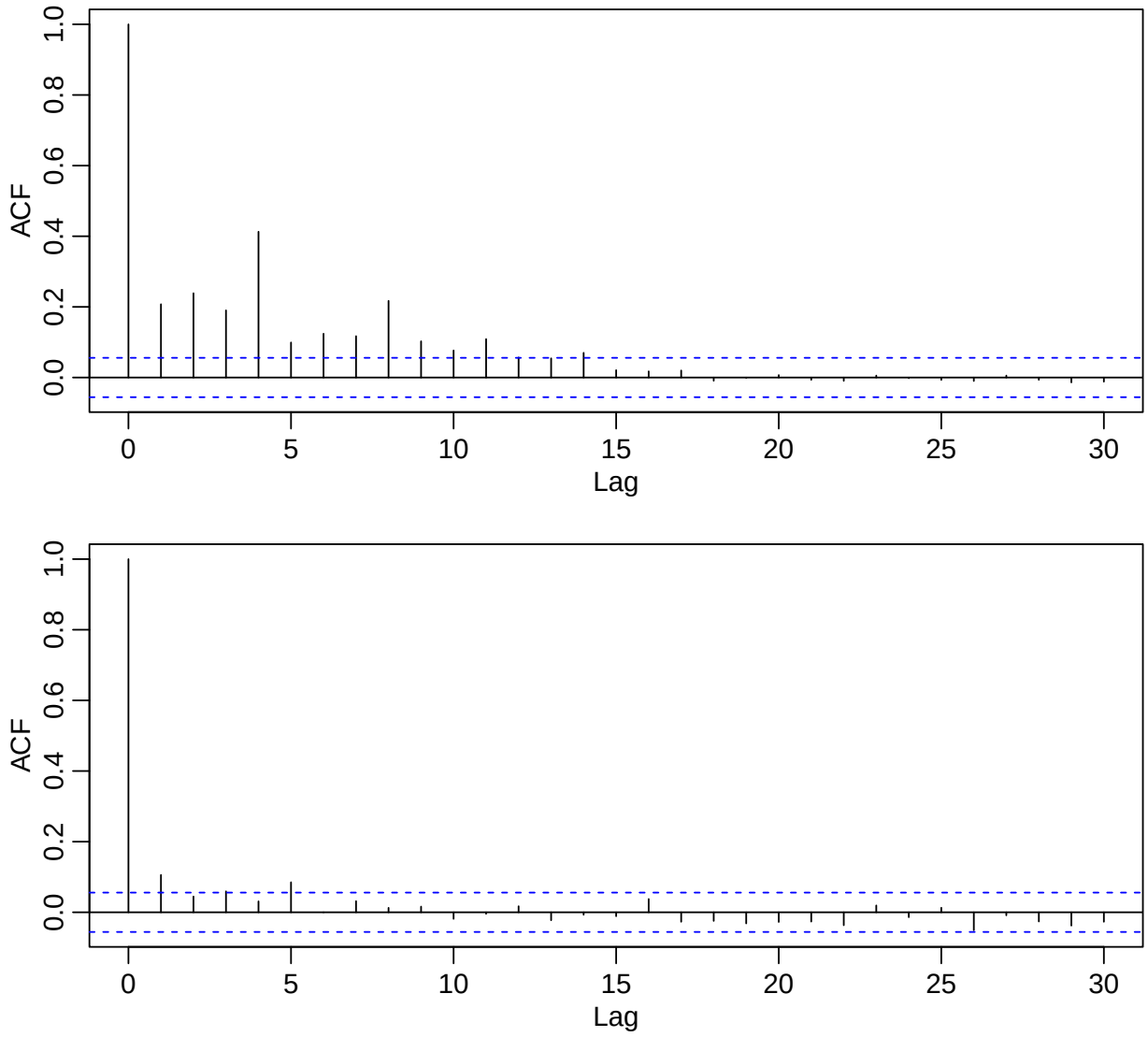

Figure 13: Correlogram comparison of residuals

4.1.1 R code:

```
RV=as.timeSeries(RV)
RVlag=lag(RV,k=1:22)
RV1=RVlag[,1]
RV5=rowSums(RVlag[,1:5])/5
RV22=rowSums(RVlag)/22

outHAR=lm(RV ~ RV1+RV5+RV22)
summary(outHAR)
```

```
qqnorm(outHAR$residuals)
qqline(outHAR$residuals, col = "red")

acf((outHAR$residuals)^2, main = "Squared residuals correlogram")

RVlag=lag(log(RV),k=1:22)
RV1l=RVlag[,1]
RV5l=rowSums(RVlag[,1:5])/5
RV22l=rowSums(RVlag)/22

outHARlog=lm(log(RV) ~ RV1l+RV5l+RV22l)
summary(outHARlog)

qqnorm(outHARlog$residuals)
qqline(outHARlog$residuals, col = "red")

par(mfrow = c(2,1))
acf(outHAR$residuals^2, main = "ACF HAR model residues")
acf(outHARlog$residuals^2, main = "ACF residuals HARlog model")
par(mfrow = c(1,1))
```

4.2 HAR - CJ model

A possible generalization of the HAR - RV model proposed previously is to decompose the realized variance into the continuous component and the discontinuous component ($RV_t = CV_t + J_t^2$). In this way, the model will have the following representation:

$$RV_t = \alpha + \beta_D CV_{t-1} + \beta_W CV_{t-1:t-5} + \beta_M CV_{t-1:t-22} + \gamma_D J_{t-1}^2 + \gamma_W J_{t-1:t-5}^2 + \gamma_M J_{t-1:t-22}^2 + \epsilon_t \quad (18)$$

The estimated model presents the following output:

```
    Call:
lm(formula = RV ~ CV1 + CV5l + CV22l + J2.1 + J2.5l + J2.22l)

Residuals:
       Min 1Q Median 3Q Max
-2.083e-03 -8.653e-05 -2.912e-05 5.397e-05 3.158e-03
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.598e-05 1.154e-05 3.983 7.19e-05 ***
CV1 6.940e-01 3.360e-02 20.653 < 2e-16 ***
CV5l 1.822e-01 4.454e-02 4.091 4.57e-05 ***
CV22l 3.357e-02 3.934e-02 0.853 0.3936
J2.1 -6.533e-01 2.717e-01 -2.404 0.0163 *
J2.5l 1.021e+00 7.180e-01 1.422 0.1552
J2.22l -2.866e+00 1.427e+00 -2.009 0.0448 *
---
Significant codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.000266 on 1230 degrees of freedom
   (22 observations deleted due to missingness)
Multiple R-squared: 0.6917, Adjusted R-squared: 0.6902
F-statistic: 460 on 6 and 1230 DF, p-value: < 2.2e-16
```

In this case it is possible to notice how the values of the discontinuous component, relating to the jumps, have a negative value.

4.2.1 R code:

```
J2POS=colSums(((r^2)*idJ)*(r>0))
J2NEG=colSums(((r^2)*idJ)*(r<0))
J2=J2POS+J2NEG
CVPOS=colSums(((r^2)*(1-idJ))*(r>0))
CVNEG=colSums(((r^2)*(1-idJ))*(r<0))
CV=CVPOS+CVNEG

CV = as.timeSeries(CV)
CVlag = lag(CV, k=1:22)
CV1 = CVlag[,1]
CV5l=rowSums(CVlag[,1:5])/5
CV22l=rowSums(CVlag)/22

J2 = as.timeSeries(J2)
J2lag = lag(J2, k=1:22)
J2.1 = J2lag[,1]
```

```
J2.5l=rowSums(J2lag[,1:5])/5
J2.22l=rowSums(J2lag)/22

outHAR_CJ=lm(RV ~ CV1 + CV5l + CV22l + J2.1 + J2.5l + J2.22l)
summary(outHAR_CJ)
```

4.3 HAR - PS model

A different approach, compared to those proposed so far, is to decompose the realized volatility using good and bad volatility. In this way, the model can be represented in the following way:

$$RV_t = +\beta_{D,+} GV_{t-1} + \beta_{D,-} BV_{t-1} + \beta_W RV_{t-1:t-5} + \beta_M RV_{t-1:t-22} + \epsilon_t \tag{19}$$

The estimated model presents the following output:

```
    Call:
lm(formula = RV ~ GVOL1 + BVOL1 + RV5 + RV22)

Residuals:
        Min 1Q Median 3Q Max
-0.0020102 -0.0000870 -0.0000254 0.0000491 0.0032011

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0000470 0.0000116 4.050 5.43e-05 ***
GVOL1 0.3091963 0.1100881 2.809 0.00505 **
BVOL1 0.9829387 0.1092307 8.999 < 2e-16 ***
RV5 0.2562832 0.0434947 5.892 4.91e-09 ***
RV22 -0.0259821 0.0329372 -0.789 0.43036
---
Significant codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0002681 on 1232 degrees of freedom
   (22 observations deleted due to missingness)
Multiple R-squared: 0.6864, Adjusted R-squared: 0.6854
F-statistic: 674.1 on 4 and 1232 DF, p-value: < 2.2e-16
```

The model presents significance for all parameters except for the realized volatility relating to the last month, which is also negative. The non-significance of this parameter should not

be surprising, since it may be normal that this value, relating to the last month, does not influence the realized volatility. However, since the value is negative, I apply the logarithmic transformation, as done for the HAR - RV model, and estimate the model:

```
    Call:
lm(formula = log(RV) ~ GVOL1_log + BVOL1_log + RV5l + RV22l)

Residuals:
     Min       1Q   Median       3Q      Max
-1.41230 -0.30143 -0.03371  0.26235  2.12062

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.41045    0.17825  -2.303   0.0215 *
GVOL1_log    0.22198    0.05006   4.435 1.00e-05 ***
BVOL1_log    0.28618    0.04597   6.226 6.57e-10 ***
RV5l         0.33536    0.04801   6.986 4.63e-12 ***
RV22l        0.06302    0.03672   1.716   0.0863 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4348 on 1232 degrees of freedom
  (22 observations deleted due to missingness)
Multiple R-squared:  0.6734,Adjusted R-squared:  0.6724
F-statistic: 635.1 on 4 and 1232 DF,  p-value: < 2.2e-16
```

Unlike the previous model, I now have all positive values, even if the value of RV_{22} is still not significant for a 5% level. Despite this, it is interesting to observe the QQ-Plot of the residuals, which, following the logarithm transformation, are closer to a normal distribution:

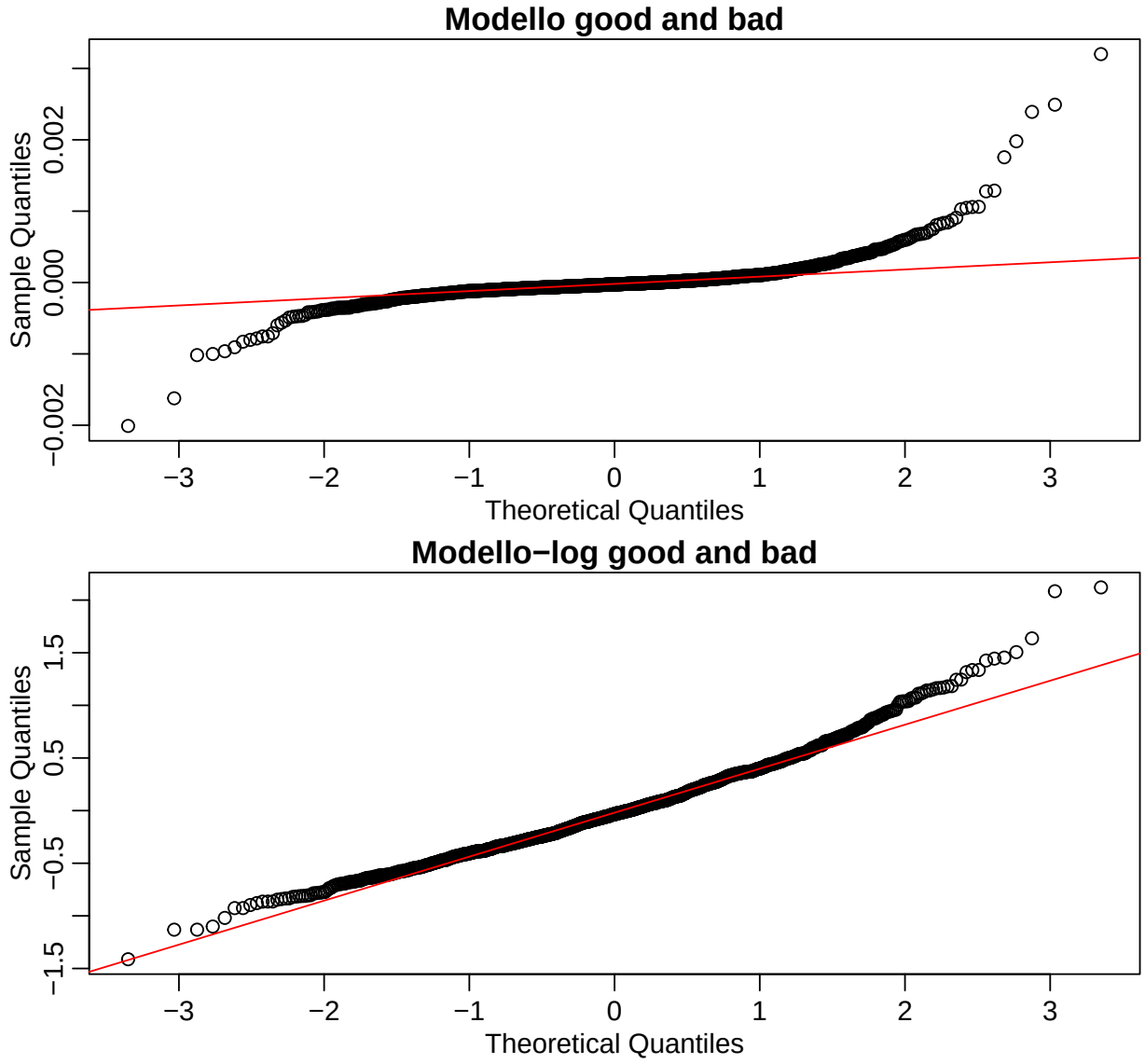

Figure 14: QQ-Plot residual HAR - PS model

4.3.1 R code:

```
GVOL = CVPOS + J2POS
BVOL = CVNEG + J2NEG

GVOL = as.timeSeries(GVOL)
GVOLlag = lag(GVOL, k=1:22)
GVOL1 = GVOLlag[,1]

BVOL = as.timeSeries(BVOL)
```

```
BVOLlag = lag(BVOL, k=1:22)
BVOL1 = BVOLlag[,1]

outHAR.GB=lm(RV ~ GVOL1+BVOL1+RV5+RV22)
summary(outHAR.GB)

GVOL_log = as.timeSeries(GVOL)
GVOLlag_log = lag(log(GVOL_log), k=1:22)
GVOL1_log = GVOLlag_log[,1]

BVOL_log = as.timeSeries(BVOL)
BVOLlag_log = lag(log(BVOL_log), k=1:22)
BVOL1_log = BVOLlag_log[,1]

outHAR_GB_log=lm(log(RV) ~ GVOL1_log+BVOL1_log+RV5l+RV22l)
summary(outHAR_GB_log)

qqnorm(outHAR.GB$residuals, main = "Good and bad model")
qqline(outHAR.GB$residuals, col = "red")
qqnorm(outHAR_GB_log$residuals, main = "Model-log good and bad")
qqline(outHAR_GB_log$residuals, col = "red")
```

4.4 Comparison between models

Following the estimation of two or more models, it is natural to ask which of the estimated ones has a greater predictive capacity, and therefore it is useful to make a comparison between them. One of the most used methods in this context is the Model Confidence Set, which defines as a null hypothesis that all the models included in a set all provide the same expected loss function:

$$H_0 : E[l_t^m] = E[l_t^i] \tag{20}$$

with an alternative hypothesis that one of the models is inferior to the others because it is characterized by a higher loss. In this case, to apply this method, a package available on R[6] (*MCS*) was directly applied, and the function that applies the Model Confidence Set (*MCSprocedure*). You can see the table of p-values provided:

[6]For this reason I will not dwell on this method

HAR-RV	HAR-CJ	HAR-GB	HAR-RV-log	HAR-GB-log
eliminated	eliminated	eliminated	1	1

Table 2: P-value

The Model Confidence Set for the HAR-RV, HAR-CJ and HAR-GB models provides "eliminated". I assume that this result is due to the models not being included in the confidence set for a $\alpha = 0.25$

4.4.1 R code:

```
pred.HAR = predict(outHAR)
pred.HAR.log = predict(outHARlog)
pred.HAR_CJ = predict(outHAR_CJ)
pred.HAR.GB = predict(outHAR.GB)
pred.HAR.GB.log = predict(outHAR_GB_log)

predictions = cbind(pred.HAR, pred.HAR.log, pred.HAR_CJ,
                    pred.HAR.GB, pred.HAR.GB.log)
predictions = as.matrix(predictions)
mcs_result = MCSprocedure(predictions, alpha = 0.25, B = 5000,
                          statistic='Tmax',cl=NULL)

print(mcs_result)
```

5 Quantitative technical analysis

Technical analysis means the use of graphical or quantitative approaches to identify pattern and trend signals in financial instruments. Due to its nature of easy and quick application it is widely used, but obviously presents critical issues such as: strong elements of subjectivity and in contrast with the theory of market efficiency. For the analysis of this last paragraph it is important to underline the difference between graphic analysis and quantitative analysis: the first is based on the identification of trendlines present in different graphic representations of prices; while the second uses a set of indicators used to build graphical representations and identify buy/sell signals.
Specifically, we will implement, on 1 second data, a quantitative technical analysis indicator, resulting less subjective, in order to build a trading strategy. For a better analysis and trading strategy, the period before the outbreak of the Covid-19 pandemic will be taken into consideration, this is because the event led to an extremely significant market shock also leading to a change in investor habits.

5.1 Rate of Change - ROC

The indicator used to carry out the analysis is the Rate Of Change (ROC) which measures the percentage change in prices over h days:

$$ROC_t = 100 \times \frac{P_t - P_{t-h}}{P_{t-h}} \tag{21}$$

In this context, the buy/sell signals are linked to the crossing of a threshold value: downward crossing for sale, upward crossing for purchase. The identification of threshold values and the definition of the range used to evaluate the ROC represent elements of subjectivity of this indicator. For this reason, analysts often combine moving averages alongside indicators to confirm their intuitions.
The graph of the ROC curve is compared below with that of the prices, where the period from 23-10-2018 to 17-03-2020 is considered. It can be seen that at the end of the series considered there is a collapse in prices and the indicator. This aspect, as previously noted, is due to the onset of the Covid-19 pandemic.

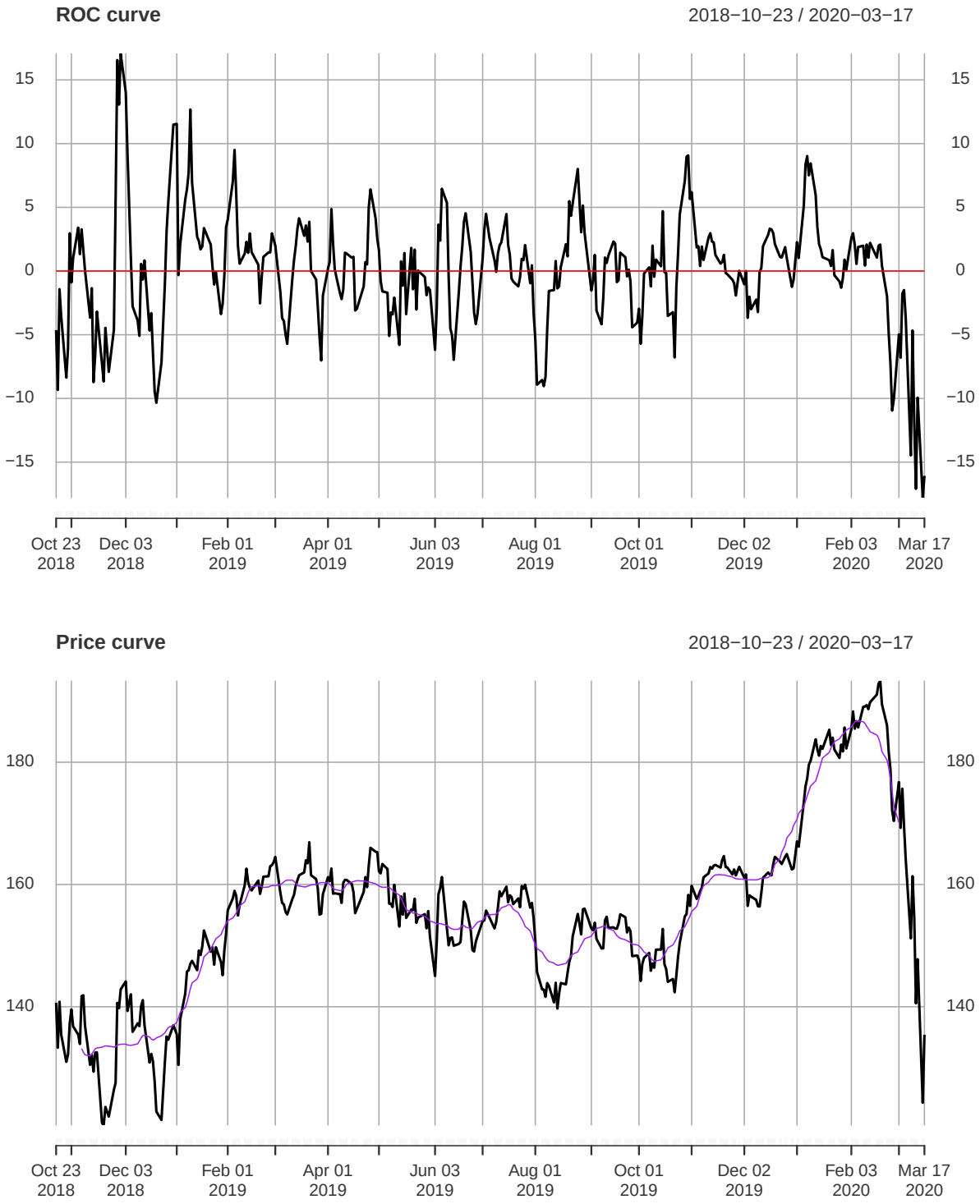

Figure 15: Comparison of ROC curve and price curve

In the graph, a single threshold has been defined, with a value of 0, for the shape that

the ROC indicator presents (it is possible to have distinct ones for purchase and sale, with $s_a > a_v$); in fact, if supports and resistances were constructed[7] with the maximum price and the minimum price, the indicator curve would remain within them. To define a better trading strategy I also report the price graph with the 5-day moving average. The moving average, in purple, should show an upward trend when the price curve is above that of the moving average; while it should show a bearish trend when the price curve is below that of the moving average. In this way we observe that, out of a total of 350 days in which the market is open, 183 trades with bullish expectations and 146 with bearish expectations were observed.

Based on the indicator and the graphical representations proposed, the best period to have an open position on the CRM stock is from the beginning of 2019 until the beginning of February 2020. The graphs in 15 allow us to provide this, with particular relevance on the end of the series of the ROC curve, which highlights, a few days ahead of the price curve and the moving average, closing the position on the security analyzed here.

[7]Typical aspects of graphic technical analysis: they are widely used elements and define price levels that support the (support) or oppose the (resistance) price movement