Question 1:

Name : **NVIDIA Corporation**

Ticker : **NVDA**

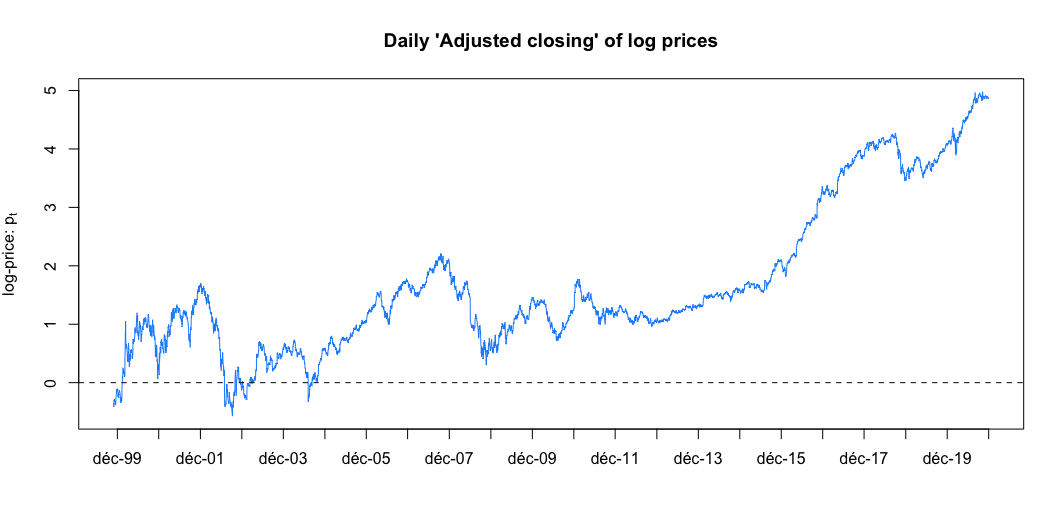
Initial date : **29 November 1999**

Final date : **31 December 2020**

Sample length: **5307 days, 254 months, 22 years**.

Stock Market: **NasdaqGS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Daily** | **Weekly** | **Montly** | **Annual** |
| **Mean** | 0.0985 | 0.47209 | 2.09064 | 23.73851 |
| **St.Deviation** | 3.80607 | 8.43828 | 17.06301 | 75.51615 |
| **Diameter.C.I.Mean** | 0.10241 | 0.49866 | 2.10254 | 32.2982 |
| **Skewness** | -0.26838 | -0.01894 | -0.22985 | -1.17859 |
| **Kurtosis** | 16.81122 | 10.72867 | 4.71784 | 4.36744 |
| **Excess.Kurtosis** | 13.81122 | 7.72867 | 1.71784 | 1.36744 |
| **Min** | -43.43811 | -49.17694 | -66.66041 | -175.99816 |
| **Quant.5%** | -5.51277 | -11.68059 | -25.36747 | -143.87956 |
| **Quant.25%** | -1.58735 | -3.6779 | -5.41137 | -10.53604 |
| **Median.50%** | 0.09628 | 0.4992 | 2.80918 | 33.37097 |
| **Quant.75%** | 1.73384 | 4.71371 | 11.15474 | 70.0935 |
| **Quant.95%** | 5.64291 | 13.26347 | 27.82824 | 118.46684 |
| **Max** | 35.35721 | 69.52672 | 60.2258 | 140.69658 |
| **Jarque.Bera.stat.X-squared** | 42235.2948 | 2737.79495 | 33.33608 | 6.49791 |
| **Jarque.Bera.pvalue.X100** | 0 | 0 | 0.00001 | 3.88148 |
| **Lillie.test.stat.D** | 0.08989 | 0.07541 | 0.08134 | 0.16917 |
| **Lillie.test.pvalue.X100** | 0 | 0 | 0.03497 | 12.05429 |
| **N.obs** | 5306 | 1100 | 253 | 21 |

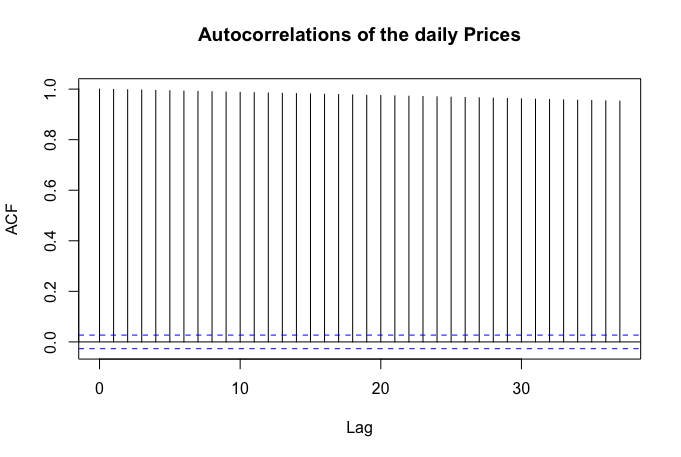
Stylized fact number 1: Prices are non-stationary:

The NVIDIA stock displays a **long-term exponentially increasing trend**. This trend is **not deterministic**, as the sudden and negative swings show. For example, we can look at the evolution of log-prices as the evolution is similar as that of prices, since the logarithm function is increasing. Around the tech bubble in 2001 and 2002, the log-price went from 1.1762863 on the 31st of August 2001 to -0.572381929 on the 9th of October 2002. The former level of log-prices was not observed before January 2006.

We can thus say that the log-prices (and prices) have a **stochastic trend**: the best way to predict the future value is by considering the current value, as every change seems to have a permanent effect on its future log-prices.

The first condition of the weak stationarity definition is therefore not met: the **expected value is time varying**.

Another proof for the non**-stationarity of prices** is the autocorrelation function. The auto-correlation of is defined as follows:



**Almost = 1**

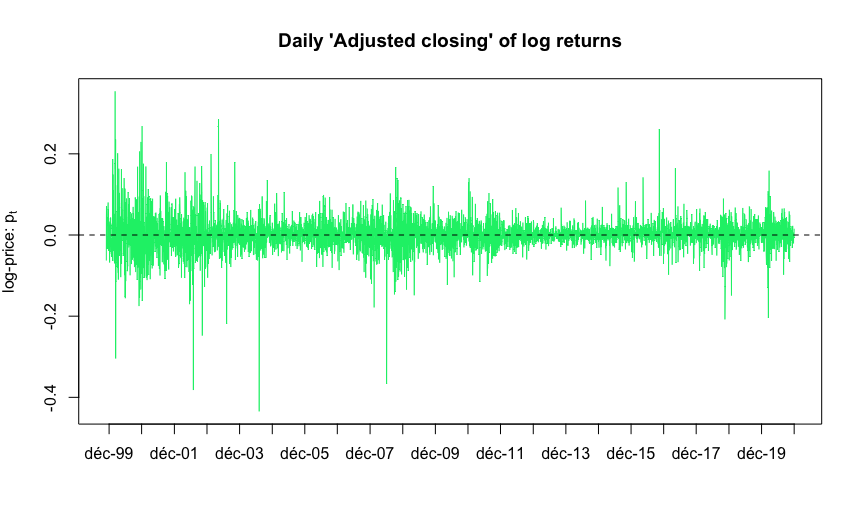
**Slowly decaying as k increases**

We can see on this second figure an illustration of the long memory of this time series: for non-stationary time series, and for a fixed sample size, we expect to see large values of , i.e. near to 1, slowly decaying as increases

Stylized fact number 2: returns are stationary

Contrary to prices, log returns fluctuate around a constant level, suggesting a **constant mean over time**. This is consistent with a second-order stationarity.

Here, as the table shows, the mean is close to 0 for the daily log returns: 0.0985

This mean **increases with time for the daily log-returns**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Daily** | **Weekly** | **Montly** | **Annual** |
| **Mean** | 0.0985 | 0.47209 | 2.09064 | 23.73851 |

Stylized facts 3, 4 and 5: Asymmetry, Heavy tails & High-frequency non-Gaussianity

The distribution is **negatively skewed**, as the introductory table shows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Daily** | **Weekly** | **Montly** | **Annual** |
| **Skewness** | -0.26838 | -0.01894 | -0.22985 | -1.17859 |

We can observe this from both histograms (in the next figure) which have longer left-tails x.

We can also notice the **leptokurtic distributions**, especially on the daily log-return one by looking at the Q-Q plots (in the next figure)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Daily** | **Weekly** | **Montly** | **Annual** |
| **Kurtosis** | 16.81122 | 10.72867 | 4.71784 | 4.36744 |
| **Excess.Kurtosis** | 13.81122 | 7.72867 | 1.71784 | 1.36744 |

The **departure from Gaussianity** can be tested with the Jarque-Bera test. Using the values from the first table:

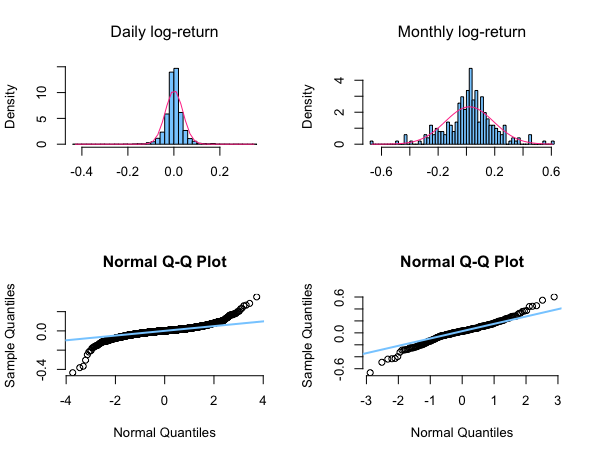
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Daily** | **Weekly** | **Montly** | **Annual** |
| **Jarque.Bera.stat.X-squared** | 42235.2948 | 2737.79495 | 33.33608 | 6.49791 |
| **Jarque.Bera.pvalue.X100** | 0 | 0 | 0.00001 | 3.88148 |

The null and alternative hypothesis of the Jarque-Bera test are:

If the null is rejected, the distribution of the returns cannot be normal.

The **null is rejected** at the level 0.1% for the daily, weekly, and monthly log-returns (by looking at the p-value). This shows that the distribution of these returns is not normal. However, the null is not rejected for the annual log-returns. This illustrate the stylized fact 5: the higher the frequency, the bigger the departure from the Gaussianity distribution.

We can also observe this **aggregational Gaussianity** on the QQ-plots: the departure from the normal distribution is shown by all the dots which are not on the blue line. More dots departs from that blue line on the daily returns (left) compared to the monthly ones (right).



**More values on the left side of the graph**

**The distribution departs from the normal one especially on the extreme values, which results in fatter tails**

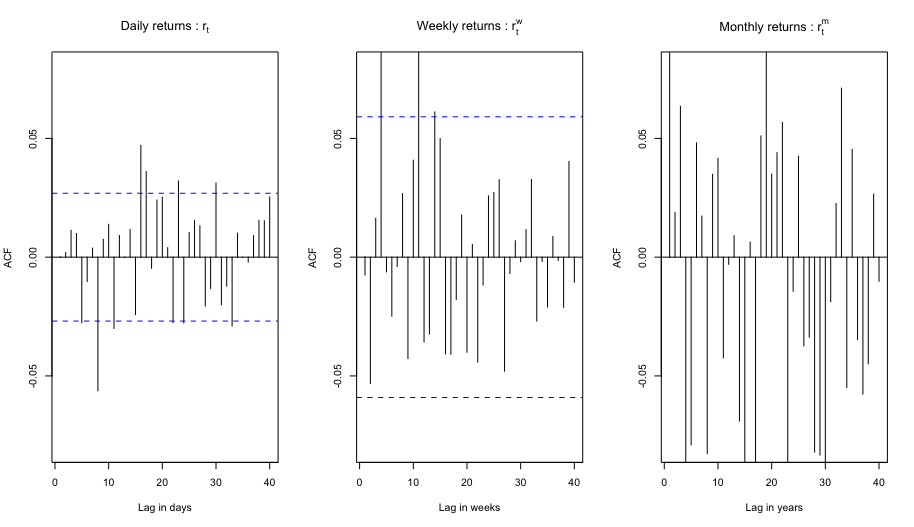
Stylized fact 6: Returns are not autocorrelated

The null and alternative hypothesis of the **significance test for the individual autocorrelation of order j** are:

Under the null hypothesis and assuming that the data rt are iid then, the asymptotic distribution is:

From this, it is possible to build the Bartlett interval, inside which the null is not rejected;

This interval is represented on the next figure by the dashed blue lines. It is thus possible to observe that very few bars cross the blue lines, which means that the null is usually not rejected. Therefore, the **returns are not correlated for most lags for the daily, weekly and monthly returns**.



Let’s do a **Box-Pierce** test to test the autocorrelation. The null and alternative hypothesis are:

Under the null hypothesis and assuming that the data are iid we get:

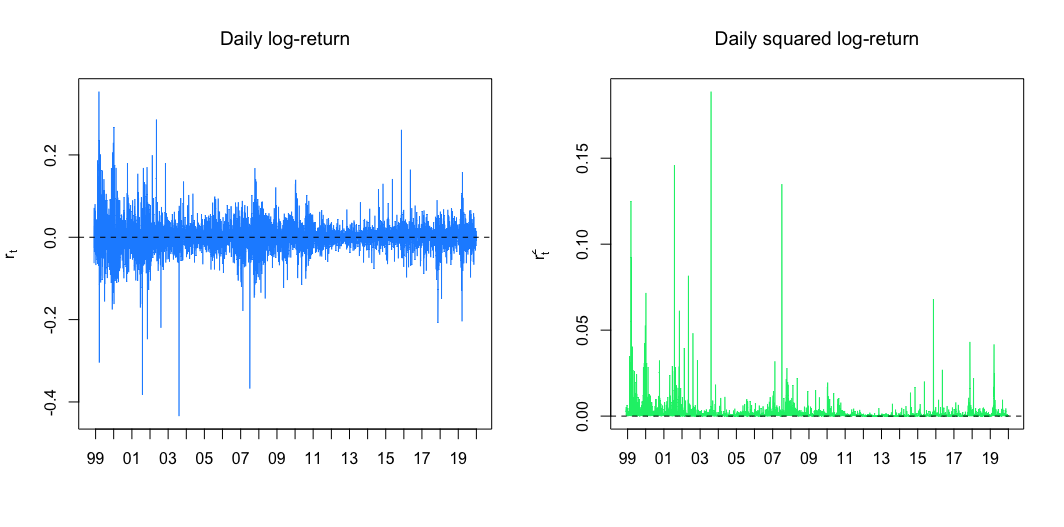
We should reject if the p-value associated to the test statistic is smaller than the significance level .

Une image contenant texte, appareil

Description générée automatiquementLet’s look at the R-generated table of the p-value for this test, with the daily returns.

For the 7 first lags, the p-value is bigger than 5%, so we do not reject the null. Therefore, the **returns are not correlated for the daily returns**.

Stylized fact 7: Volatility clustering and long range dependence of squared returns

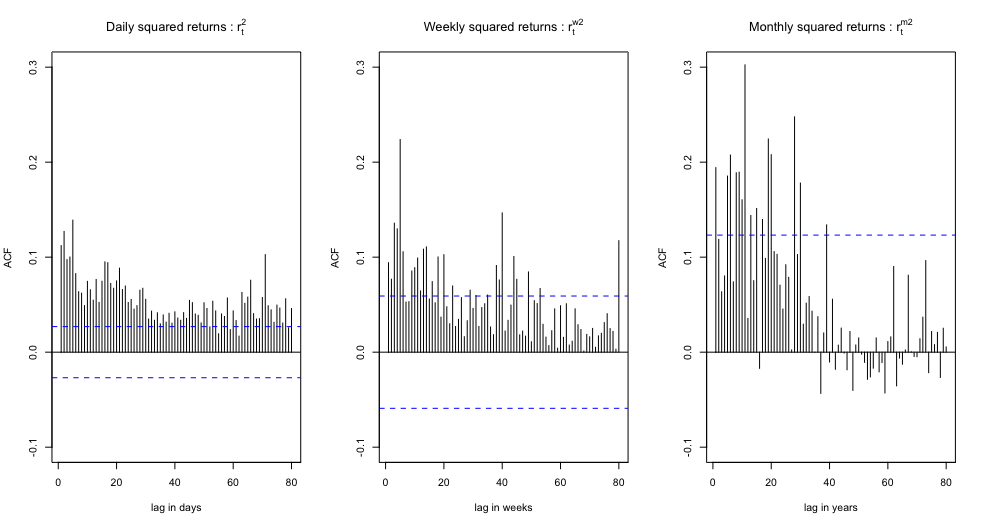
**Volatility clustering** means that large price changes, i.e. returns with large absolute values or large squares, occur in clusters. We can see that the sudden larger bars in the next graph are usually followed by large bars.

This volatility clustering is noticeable by **significant autocorrelation of the squared returns**:

The null and alternative hypothesis of the **significance test for the individual autocorrelation of order k of the squared returns** are:

Under the null hypothesis and assuming that the data rt are iid then, the asymptotic distribution is:

From this, it is possible to build the Bartlett interval, inside which the null is not rejected;

We can observe that on the second figure, the dashed blue lines being the Bartlett intervals inside which, the null is not rejected.

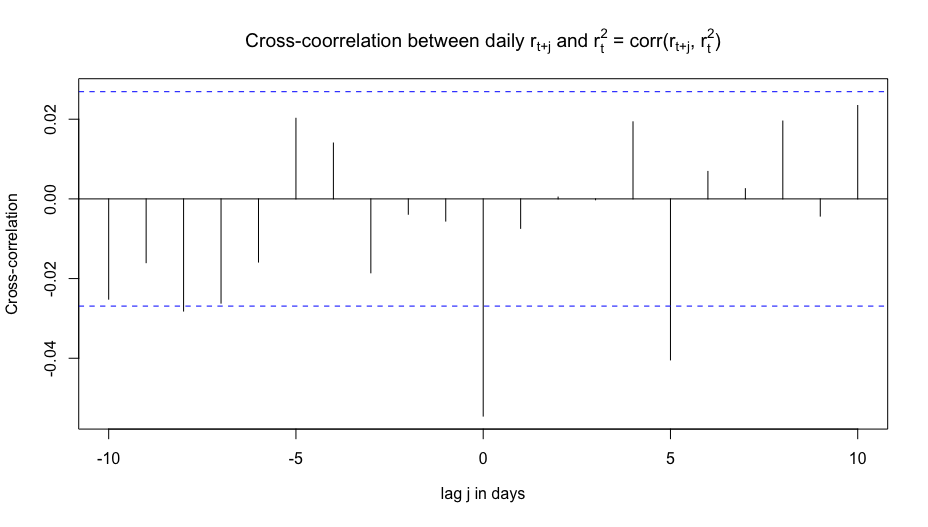
We can see that the **null is almost systematically rejected for the daily squared returns**, and then **sometimes rejected for the weekly and monthly ones**. Therefore, the **squared returns are autocorrelated**, and these autocorrelations becomes become weaker and less persistent when the sampling interval is increased from a day to a week to a month. So, **there is volatility clustering, which is very present for daily returns, and present but to a lesser extent for weekly and monthly returns**.

Stylized fact 8: Leverage effect

Asset returns are negatively correlated with the changes of their volatilities: this negative correlation is called **leverage effect**.

In particular, there is strong empirical evidence that asset-return volatility rises after price declines, with larger declines inducing greater volatility spikes, and that:

Dashed blue lines are the (asymptotic) bounds for the rejection region a significance test of each cross-correlation. A line above or below the blue dashed line represent a significant cross-correlation.

Therefore the stylized fact number 8 is **almost not observed**, **since the cross-correlation of most lags are not significant**.

Question 2.A)

Let’s do a **Bresuch-Godfrey test** to see the presence of autocorrelation, based on the auxiliary regression.

The **null and the alternative** are :

The Breusch-Godfrey-statistic is , and under it is asymptotically distributed as:

Here: ,

Let’s denote the adjusted as

So, since , we get

So:

At the 10% level:

This is a one-sided test, and because , **we do not reject** .

Therefore, we have no evidence of significant autocorrelation in the innovation of the model up to lag 5 at the 10% level.

Question 2.B)

Since there is no evidence of significant autocorrelation in the innovation of the model at the 10% level, it is **possible to keep the OLS estimates for the standard errors for the coefficients of the regression model** in equation (1) .

Under assumptions A1-A5 an unbiased estimator the variance-covariance matrix of the OLS estimator , denoted by :

Where

is the unbiased estimator of defined above. Indeed assumptions of spherical errors (absence of autocorrelation of the erros, proved before, and the assumption of homoskedasticity) are necessary to prove that .

is the estimated variance of : it is in position of matrix .The standard error of can be denoted as

Question 2.C)

Let’s do an **F-test** to see if all 3 “Fama and French factors” and the “Momentum” factor are jointly useful to explain the time- series variability of the excess returns of IBM in this sample.

The **null and the alternative** are :

Under Assumptions A1-A6, the global F test statistics for is:

Here: , .

is the sum of squared residuals of the constrained model estimated under (a model with only the constant term) and denotes the sum of squared residuals of the unconstrained model estimated under .

The R output shows the **residual standard error for both models**: the first table allows us to get the as it is the unconstrained model, and the second table the , as it has no other term than the constant one (and the error term).

With T being the number of observations and k the number of parameters. So:

So: .

At the 5% level:

This is a one-sided test, and because **, we reject** .

Therefore, the variables of the model are jointly significant: at least one of them is different form 0.

Question 2.D)

We can first use the **monthly returns** as an indicator of the monthly risk premium, as the two are positively correlated: the higher the risk premium, the bigger the monthly returns. On the second model provided, no variables explain the IBM monthly returns. The latter therefore only depends on the intercept and the error term. So, by testing if the **intercept is significantly different from 0**, we can test whether the IBM monthly returns are different from 0.

To test the significance of the intercept, let’s do a **T-test**.

The null and the alternative are :

Under , the Student test-statistics is:

Using the R output, we get:

This is a two-sided test at the 1%, so if , we reject the null hypothesis.

However, **, therefore we do not reject the null hypothesis**. The IBM monthly returns are thus not significantly different from 0.

The unbiased estimate of the standard error is

Here: and (the intercept).

The R output provides us with the residual standard error, here being equal to , which is exactly the unbiased estimate of the standard error.

Therefore:

Question 2.E)

In the presence of autocorrelation and heteroskedasticity, we should modify the previous t-test by recomputing a ne**w standard error of the estima**te. For this new standard error, we can use the **Newey-West Standard Errors, which are robust to serial correlation and to heteroskedasticity**.

Under the spheriticity of errors conditions, the expression of the standard error is usually:

Assuming that the **residuals are autocorrelated**: for , that is the autocorrelation between two period far in time is approximately 0, a consistent estimator of can be obtained, by weighting the contributions of the terms to give the estimate:

The weighting function is called the **kernel**. Using this kernel method, we can get the corresponding estimates of the standard errors of the OLS estimator that are called HAC-that is, they are Heteroskedasticity and Autocorrelation Consistent.

The appropriate estimator of is the Newey West estimator

Therefore, the estimate of the standard error is:

From this estimate of the standard error, it is possible to **conduct the t-test of the null hypothesis of the test** done in question 2.d), which is consistent for both autocorrelation and heteroskedasticity of the excess returns

Question 3:

The volat**ility of an asset is a measure of its risk**. So, we need to estimate this volatility now, and forecast its future values, as it is unobservable.

It is usually measured as the conditional standard deviation of an asset return given the available information up to the date/time when it is measured:

where is the log returns on date t.

One way to estimate this volatility is by using the conditional variance from **dynamic models like the ARCH and GARCH type models**.

Introduced by Engle in 1982, the ARCH model stands for : AutoRegressive Conditional Heteroskedasticity. This models assumes that the variance is constant over time, but the conditional variance is time-varying.

A process is said to be an **ARCH process if**:

Where is a sequence of independent and identically distributed (i.i.d.) random variables with and , and is a non-negative process such that

With and .

The process  **corresponds to the conditional variance of** .

Where is the information set available at time .

The conditional variance of is therefore deterministic, since, given , is a constant.

The process is an IID noise, therefore, there is **no memory in This means that past values**

Now, let’s see one example with the **ARCH(1)** model. If has an ARCH(1) representation, with

Then has an **AR(1) representation with**:

Where is an innovation process

The ARCH representation of a process implies the existence of an AR(1) representation of the process . We can then observe an ARCH effect because and are correlated, especially for small values of k.

**This ARCH effect is in keeping with the stylized fact 7 which states that:** The daily squared returns often exhibit significant correlations.

If is an ARCH(1) process, then **it is a martingale difference**

This property implies that for , i.e. the process has no “memory”. Therefore, even though , there is no correlation between and .

**This absence of correlation between and is in keeping with the stylized fact 2: The autocorrelation of asset returns are often insignificant.**

If is an ARCH(1) process with

Then, its two first unconditional moments are equal to

With and .

The consequence of this third property is that an **ARCH(1) process is unconditionally homoscedastic**, since its variance does not change over time (), but it is conditionally heteroscedastic () as its variance is not constant over time.

Combining the unconditional homoscedasticity of the ARCH(1) process, coming from this 3rd property, and the absence of correlation between and , coming from the 2nd property, we get that the **ARCH(1) process is weakly stationary, if . This corresponds to the 1st stylized fact: in general, prices are non-stationary, whereas returns are stationary.**

If is an ARCH(1) process with Gaussian innovations , then, its **conditional and unconditional fourth moments** are equal to:

If is an ARCH(1) process with Gaussian innovations , then, its **conditional and unconditional Kurtosis coefficients** are equal to:

Therefore, if is Gaussian, the **conditional distribution of is Gaussian**, but its unconditional distribution is not.  **has a leptokurtic unconditional distribution (if ) , which is in keeping with the stylized fact 3: The return distribution often exhibits heavier tails than those of a normal distribution**.

However, the ARCH model does not differentiate between positive and negative shocks on the volatility. And this model is quite restrictive, for example forcing for the 4th moment.

The process is said to be an **ARCH(p) process** if:

Where is a sequence of independent and identically distributed (i.i.d.) random variables with and , and is a non-negative process such that

With , , and

GARCH models **are a parsimonious alternative to ARCH models**, which often require large p to fit the data due to the large persistence in volatility.

**GARCH: Generalized AutoRegressive Conditional Heteroskedasticity**

The process is said to be a **GARCH(p,q) process**, if

Where is a sequence of i.i.d. random variables with and , and,

With , , and

The conditional variance of a GARCH(p,q) depends on:

• The first p lag oh the (e.g. the squared error terms)

• The first q lag of the conditional variance .

Let’s see an example now with the GARCH(1,1) process. The process is said to be a **GARCH(1,1)** process, if

Where is a sequence of i.i.d. random variables with and , and,

With , , and

The **conditional variance depends on two effects**:

• An **intrinsic persistence effect** through the first lag of the conditional variance ()

• An **extrinsic persistence effect** (

The main properties of a GARCH process are similar to those of an ARCH process

• has an **ARMA representation**: **this is in keeping with the stylized fact 7** (as an AR model is a particular case of an ARMA model)

• is a **martingale difference**, **in keeping with the stylized fact 2**.

• is a **stationary process** under some conditions on the parameter and , **in keeping with the stylized fact 1**.

•  **is (unconditionally) homoscedastic**

• is conditionally heteroscedastic

• The (marginal) distribution of and are **leptokurtic, in keeping with the stylized fact 3.**

• If has a normal distribution, the conditional distribution of and are normal.

APPENDIX: R code for Question 1

# library(ggplot2) # produce good looking graphs (qplot)

library(quantmod) # allows to easily import data directly from downloading financial data from the internet, directly

# from some open sources, including Yahoo Finance, Google Finance, and the Federal

# Reserve Economic Data (FRED) of Federal Reserve Bank of St. Louis.

library(xts)

library(readr)

library(latex2exp) # to wtie latex formulas in graphs!

#library(gridExtra) # multiple plots in one graph

library(summarytools)

library(qwraps2)

library(normtest)

library(nortest)

library(moments)

library(xtable)

library(sm)

library(astsa)

library(portes)

#library(xlsx)

# library(timeSeries)

library(forecast)

# library(forecast)

library(portes)

NVDA <- getSymbols("NVDA",from="1999-11-29", to="2020-12-31", auto.assign=FALSE)

# daily prices

Pt.d <- NVDA$NVDA.Adjusted ; names(Pt.d) <- "Pt.d" # Prices

pt.d <- log(Pt.d) ; names(Pt.d) <- "pt.d" # log -prices

# find end of month/week/year dates

last\_day\_of\_month <- endpoints(pt.d, on = "months")

last\_day\_of\_week <- endpoints(pt.d, on = "weeks")

last\_day\_of\_year <- endpoints(pt.d, on = "years")

# Compute weekly (w), monthly(m), and annual(y) log prices

pt.w <- pt.d[last\_day\_of\_week] ; names(pt.w) <- "pt.w.all"

pt.m <- pt.d[last\_day\_of\_month]; names(pt.m) <- "pt.m.all"

pt.y <- pt.d[last\_day\_of\_year] ; names(pt.y) <- "pt.y.all"

# compute log returns # for entire history of S&P 500

rt.d <- diff(pt.d) ; names(rt.d) <- "rt.d"

rt.w <- diff(pt.w) ; names(rt.w) <- "rt.w"

rt.m <- diff(pt.m) ; names(rt.m) <- "rt.m"

rt.y <- diff(pt.y) ; names(rt.y) <- "rt.y"

# convert prices int dataframes to produce nice plots

Pt.d.df <- cbind(index(Pt.d), data.frame(Pt.d)); names(Pt.d.df)[1] <- "date";

pt.d.df <- cbind(index(pt.d), data.frame(pt.d)); names(pt.d.df)[1] <- "date";

pt.w.df <- cbind(index(pt.w), data.frame(pt.w)); names(pt.w.df)[1] <- "date";

rt.d.df <- cbind(index(rt.d), data.frame(rt.d)); names(rt.d.df)[1] <- "date";

rt.w.df <- cbind(index(rt.w), data.frame(rt.w)); names(rt.w.df)[1] <- "date";

rt.m.df <- cbind(index(rt.m), data.frame(rt.m)); names(rt.m.df)[1] <- "date";

rt.y.df <- cbind(index(rt.y), data.frame(rt.y)); names(rt.y.df)[1] <- "date";

# personalized table of summary statistics

# creates a 'list' with the data of SP500 sampled at different frequencies

# a list is needed and not a matrix / dataframe as the vectors have different lengths!!!

X <-list("Daily" = rt.d.df[-1,2],

"Weekly" = rt.w.df[-1,2],

"Montly" = rt.m.df[-1,2],

"Annual" = rt.y.df[-1,2]);

# Create function (named 'multi.fun' which computes the statics that we want on the inpit 'x')

###############################################

multi.fun <- function(x) {

c(Mean = mean(x)\*100,

St.Deviation = sd(x)\*100,

Diameter.C.I.Mean = qnorm(0.975)\*sqrt(var(x)/length(x))\*100,

Skewness=moments::skewness(x),

Kurtosis=moments::kurtosis(x),

Excess.Kurtosis=moments::kurtosis(x)-3,

Min = min(x)\*100,

Quant = quantile(x, probs = 0.05)\*100,

Quant = quantile(x, probs = 0.25)\*100,

Median = quantile(x, probs = 0.50)\*100,

Quant = quantile(x, probs = 0.75)\*100,

Quant = quantile(x, probs = 0.95)\*100,

Max = max(x)\*100,

Jarque.Bera.stat = jarque.bera.test(x)$statistic,

Jarque.Bera.pvalue.X100 = jarque.bera.test(x)$p.value\*100,

Lillie.test.stat = lillie.test(x)$statistic,

Lillie.test.pvalue.X100 = lillie.test(x)$p.value\*100,

N.obs = length(x)

)}

# GENRETAES TABLE 1 in slide 91

a <- sapply(X, multi.fun) # apply function to all elements of list X,

# PRINT TABLE 1 ON R console

print(a)

# and return results in a nice and tidy table

round(a, digits = 5) # show nicer-looking table

#Stylized fact 1

#dailylogprice

lwd2plot <- c(1 , 1 , 1 )

lty2plot <- c(1 , 1 , 1 )

plot(x = pt.d.df[,1], y = pt.d.df[,2], type = 'l', col="dodgerblue1" , lty = lty2plot[1], lwd = lwd2plot[1],

xlab="" , ylab=TeX('log-price: $p\_t$'), main="Daily 'Adjusted closing' of log prices", xaxt ="none") # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq\_sel <- endpoints(pt.d.df$date, on = 'years'); date\_seq = pt.d.df$date[seq\_sel]; date\_lab = format(date\_seq,"%b-%y")

axis(1, at = date\_seq, label = date\_lab, las = 1, cex.axis=1.0)

abline(0,0, lty = 2)

#Daily log price

# scatterplot of log prices: p\_t vs. p\_{t-1}

plot(Lag(pt.d.df[,2]), pt.d.df[,2], col = "dodgerblue", lwd = 1, cex = 2, xlab=TeX('LAGGED daily log-price ($p\_{t-1}$)') , ylab=TeX('daily log-price ($p\_{t}$)'))

abline(0, 1, lty = 1, lwd = 2, col="violetred1")

#ACF

acf(pt.d, main="Autocorrelations of the daily Prices")

#Stylized fact 2

plot(x = rt.d.df[,1], y = rt.d.df[,2], type = 'l', col="springgreen2" , lty = lty2plot[1], lwd = lwd2plot[1],

xlab="" , ylab=TeX('log-price: $p\_t$'), main="Daily 'Adjusted closing' of log returns", xaxt ="none") # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq\_sel <- endpoints(rt.d.df$date, on = 'years'); date\_seq = rt.d.df$date[seq\_sel]; date\_lab = format(date\_seq,"%b-%y")

axis(1, at = date\_seq, label = date\_lab, las = 1, cex.axis=1.0)

abline(0,0, lty = 2)

#Histograms and QQ-plots

par(mfrow=c(2,2))

# daily returns

hist\_OUT <- hist(rt.d.df[,2], freq = FALSE, breaks = 50, col="skyblue1", xlab="", main=TeX('Daily log-return'), )

norm\_y <- dnorm(hist\_OUT$mids, mean=mean(rt.d.df[,2], na.rm=TRUE), sd=sd(rt.d.df[,2], na.rm=TRUE));

lines(x=hist\_OUT$mids, y=norm\_y,col="violetred1", lwd=1)

# monthly returns

hist\_OUT <- hist(rt.m.df[,2], freq = FALSE, breaks = 50, col="skyblue1", xlab="", main=TeX('Monthly log-return'), )

norm\_y <- dnorm(hist\_OUT$mids, mean=mean(rt.m.df[,2], na.rm=TRUE), sd=sd(rt.m.df[,2], na.rm=TRUE));

lines(x=hist\_OUT$mids, y=norm\_y,col="violetred1", lwd=1)

# QQ-plot vs quantiles of normal distribution

qqnorm(rt.d.df[,2], pch = 1, frame = FALSE, xlab="Normal Quantiles")

qqline(rt.d.df[,2], col = "skyblue1", lwd = 2)

qqnorm(rt.m.df[,2], pch = 1, frame = FALSE, xlab="Normal Quantiles")

qqline(rt.m.df[,2], col = "skyblue1", lwd = 2)

# ACF of returns: Autocorrelation function

par(mfrow=c(1,3))

lag.max.acf = 40; lim.y.axes = c(-0.08,0.08)

# daily returns

Acf(rt.d.df[-1,2], main=TeX('Daily returns : $r\_t$'), lag.max = lag.max.acf, xlab = "Lag in days", ylim=lim.y.axes, xlim = c(1,40))

# weekly returns

Acf(rt.w.df[-1,2], main=TeX('Weekly returns : $r\_t^w$'), lag.max = lag.max.acf, xlab = "Lag in weeks", ylim=lim.y.axes, xlim = c(1,40))

# monthly returns

Acf(rt.m.df[-1,2], main=TeX('Monthly returns : $r\_t^m$'), lag.max = lag.max.acf, xlab = "Lag in years", ylim=lim.y.axes, xlim = c(1,40))

#dailylogprice&squared MIN

par(mfrow=c(1,2))

lwd2plot <- c(1 , 1 , 1 )

lty2plot <- c(1 , 1 , 1 )

plot(x = rt.d.df[,1], y = rt.d.df[,2], type = 'l', col="dodgerblue1" , lty = lty2plot[1], lwd = lwd2plot[1],

xlab="" , ylab=TeX('log-price: $p\_t$'), main="Daily 'Adjusted closing' of log prices", xaxt ="none") # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq\_sel <- endpoints(rt.d.df$date, on = 'years'); date\_seq = rt.d.df$date[seq\_sel]; date\_lab = format(date\_seq,"%b-%y")

axis(1, at = date\_seq, label = date\_lab, las = 1, cex.axis=1.0)

abline(0,0, lty = 2)

plot(x = rt.d.df[,1], y = (rt.d.df[,2]^2), type = 'l', col="springgreen2" , lty = lty2plot[1], lwd = lwd2plot[1],

xlab="" , ylab=TeX('squared log-price: $p\_t$'), main="Daily 'Adjusted closing' of squared log prices", xaxt ="none") # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq\_sel <- endpoints(rt.d.df$date, on = 'years'); date\_seq = rt.d.df$date[seq\_sel]; date\_lab = format(date\_seq,"%b-%y")

axis(1, at = date\_seq, label = date\_lab, las = 1, cex.axis=1.0)

abline(0,0, lty = 2)

# Daily log-returns and squared returns on a recent subsample PROF

par(mfrow=c(1,2))

plot(x = rt.d.df[,1], y = rt.d.df[,2], type = 'l', col="dodgerblue1" , lty = lty2plot[1], lwd = lwd2plot[1],

xlab="" , ylab=TeX('$r\_t$'), main=TeX('Daily log-return'), xaxt ="none") # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq\_sel <- endpoints(rt.d.df$date, on = 'years'); date\_seq = rt.d.df$date[seq\_sel]; date\_lab = format(date\_seq,"%y")

axis(1, at = date\_seq, label = date\_lab, las = 1, cex.axis=1.0); abline(0,0, lty = 2) # add zero line

plot(x = rt.d.df[,1], y = (rt.d.df[,2])^2, type = 'l', col="springgreen2" , lty = lty2plot[1], lwd = lwd2plot[1],

xlab="" , ylab=TeX('$r\_t^2$'), main=TeX('Daily squared log-return'), xaxt ="none") # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq\_sel <- endpoints(rt.d.df$date, on = 'years'); date\_seq = rt.d.df$date[seq\_sel]; date\_lab = format(date\_seq,"%y")

axis(1, at = date\_seq, label = date\_lab, las = 1, cex.axis=1.0); abline(0,0, lty = 2) # add zero line

#ACF of squared returns

par(mfrow=c(1,3))

lag.max.acf = 80; lim.y.axes = c(-0.10,0.30)

data2plot = (rt.d.df[,2])^2 ; # daily squared returns

Acf(data2plot, main=TeX('Daily squared returns : $r\_t^2$'), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes)

data2plot = (rt.w.df[,2])^2; # weekly squared returns

Acf(data2plot, main=TeX('Weekly squared returns : $r\_t^{w 2}$'), lag.max = lag.max.acf, xlab = "lag in weeks", ylim=lim.y.axes)

data2plot = (rt.m.df[,2])^2; # monthly squared returns

Acf(data2plot, main=TeX('Monthly squared returns : $r\_t^{m 2}$'), lag.max = lag.max.acf, xlab = "lag in years", ylim=lim.y.axes)

# ACF of absolute returns

lag.max.acf = 80; lim.y.axes = c(-0.10,0.30)

data2plot = abs(rt.d.df[,2]) ; # daily abs returns

Acf(data2plot, main=TeX('Daily absolute returns : $|r\_t|$'), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes)

data2plot = abs(rt.w.df[,2]); # weekly abs returns

Acf(data2plot, main=TeX('Weekly absolute returns : $|r\_t^w|$'), lag.max = lag.max.acf, xlab = "lag in weeks", ylim=lim.y.axes)

data2plot = abs(rt.m.df[,2]); # monthly abs returns

Acf(data2plot, main=TeX('Monthly absolute returns : $|r\_t$^m|'), lag.max = lag.max.acf, xlab = "lag in years", ylim=lim.y.axes)

ret = (rt.d.df[-1,2]) ; # daily returns

ret2 = (rt.d.df[-1,2])^2 ; # daily SQUARED returns

# Cross correaltion plot in the slides

par(mfrow=c(1,1))

ss\_dates3 <- "19880101/20181231"

rt.d.subsamp3 <- rt.d.all[ss\_dates3];

ret <- as.numeric(rt.d.subsamp3); ret2 <- ret^2

ccf(ret, ret2, lag.max = 10, type = "correlation", plot = TRUE,

main=TeX('Cross-coorrelation between daily $r\_{t+j}$ and $r\_t^2$ = corr($r\_{t+j}$, $r\_{t}^2$)'),

xlab = TeX('lag $j$ in days'), ylab=TeX('Cross-correlation'))

#Box-Pierce and Ljung Box tests

my.data <- rt.d.df[-1,2];

# my.data <- rt.m.df[-1,2];

# my.data <- rt.d.all.noOct87crash.df[-1,2]

lags.all <- seq(1,25,1);

acf(my.data, lag.max <- max(lags.all), plot = FALSE)

# Barteltt interval

1.96/sqrt(length(my.data))

# a<- acf(my.data, max.lag = 19, plot = FALSE)

# str(a)

##############################################################

##############################################################

# test only first lag

Box.test(my.data, lag = 1, type = c("Box-Pierce"), fitdf = 0)

Box.test(my.data, lag = 1, type = c("Ljung-Box"), fitdf = 0)

# test all first 5 lags

Box.test(my.data, lag = 5, type = c("Ljung-Box"), fitdf = 0)

Box.test(my.data, lag = 5, type = c("Box-Pierce"), fitdf = 0)

##############################################################

#############################################################

# a1<- acf2(my.data, max.lag = 19) # computes ACF and PACF (used in time series) !

# Customized table with values of ACF, Box-pierce and Ljung Box statistics

my.max.lag <- 25

lags.all <- seq(1,my.max.lag,1)

my.acf <- acf(my.data, lag.max = my.max.lag, plot = FALSE)

my.acf.diameter <- qnorm(0.975)/sqrt(length(my.data))

my.acf.tstat.0 <- (my.acf$acf[-1] - 0)/sqrt(1/length(my.data))

my.LjungBox <- LjungBox(my.data, lags=lags.all)

my.BoxPierce <- BoxPierce(my.data, lags=lags.all)

crit.value.5.BP <- qchisq(0.95,lags.all)

my.table <- cbind(my.BoxPierce[,1],

my.acf$acf[-1],

my.acf.diameter,

my.acf.tstat.0,

my.BoxPierce[,2],

my.BoxPierce[,4],

my.LjungBox[,2],

my.LjungBox[,4],

crit.value.5.BP)

# change appearance to export more easily !

my.table.df <-as.data.frame(my.table)

names(my.table.df) <- c("lag","acf","acf diam.","acf test","Box-Pierce stat","BP pval","LB stat","LB pval","crit")

rownames(my.table.df) <-c()

options(scipen = 999)

a <- data.matrix(my.table.df)