EDHEC BUSINESS SCHOOL

COURSEWORK

EDHEC BUSINESS SCHOOL

MASTER IN MANAGEMENT FINANCIAL ECONOMICS

# Financial Econometrics With R: Final Exam

*Author:*
Matteo Mario Di Venti (ID: 80389)

Date: December 10, 2021

# 1 Question 1

The stock index considered is the **DAX Performance Index** or "Deutscher Aktienindex" (ticker: **GDAXI**), a stock market index (currently) tracking the performance of the Prime Standard's 40 largest blue chip companies in terms of order book volume and market capitalization on **Frankfurt Stock Exchange** in **Germany**.
The index is a total return index, therefore it includes dividends, interest, rights offerings and other distributions. DAX has base date 30/12/1987 and it started from base value of 1000.
The index is operated by company Xetra that recalculates it at a frequency of every second since 2006. In the third quarter of 2021, DAX expanded from 30 to 40 members. In the sample considered, the DAX has constantly 30 members.

The sample considered spans from **30-12-1987**, the first trading day of the Index, to **30-12-2020**, the last trading day of 2020. The sample comprises **8335** trading days, **1723** weeks, **397** months or **33** years of financial data. The data touches 34 years of data as it starts in Dec 1987.

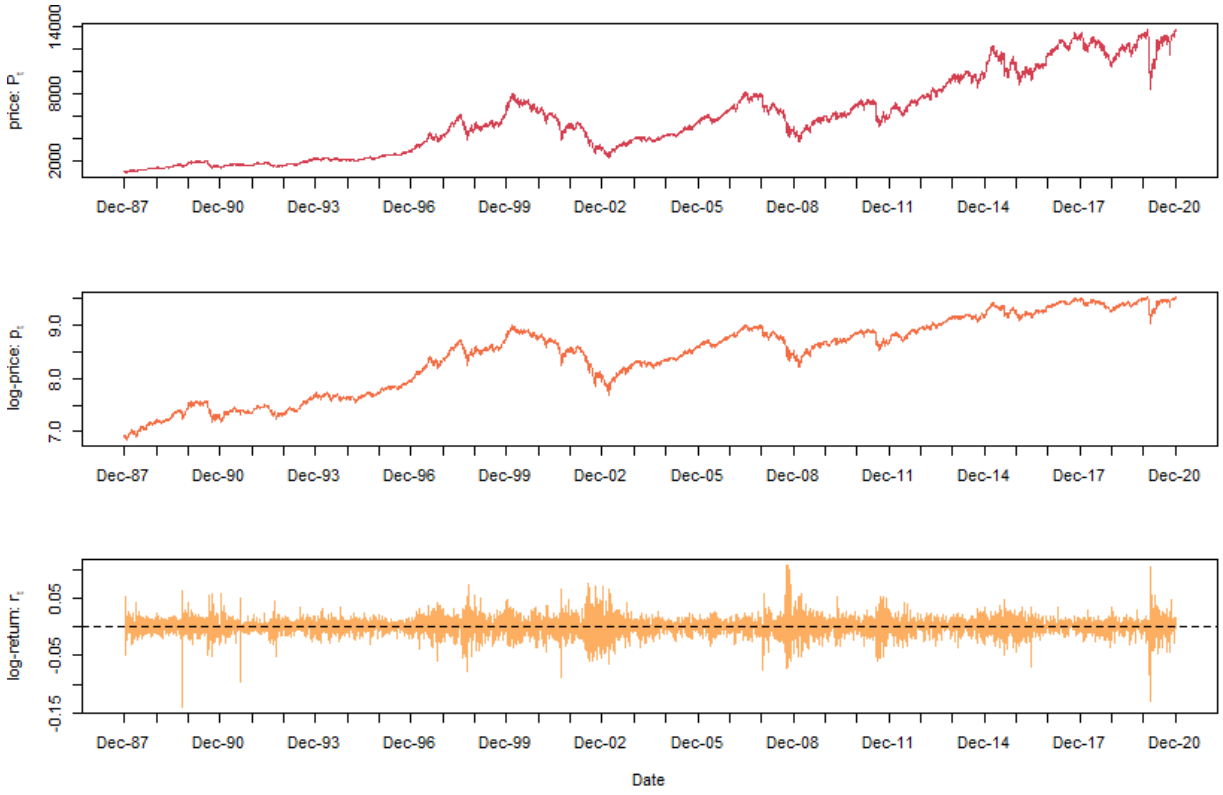


**Figure 1:** DAX Prices, log-prices and log-returns at daily frequency from "adjusted closing" prices taken from yahoo finance for sample: 30-12-1987 to 30-12-2020

### 1.0.1 Summary Statistics

After downloading the adjusted-closing sample data from Yahoo Finance and following the computation of log-returns for daily, weekly, monthly and annual frequencies, the summary statistics table below is derived.

| | daily | weekly | montly | annual |
|---|---|---|---|---|
| Mean | 0.0314 | 0.1520 | 0.6608 | 7.9294 |
| St.Deviation | 1.4127 | 3.0522 | 6.0406 | 23.4674 |
| Diameter.C.I.Mean | 0.0303 | 0.1442 | 0.5950 | 8.0067 |
| Skewness | -0.3110 | -0.7843 | -0.8557 | -1.1627 |
| Kurtosis | 9.7006 | 8.8354 | 5.6422 | 3.9937 |
| Excess.Kurtosis | 6.7006 | 5.8354 | 2.6422 | 0.9937 |
| Min | -14.0912 | -24.3470 | -29.3327 | -57.8790 |
| Quant.5% | -2.2010 | -4.8527 | -9.7383 | -35.4865 |
| Quant.25% | -0.6384 | -1.4077 | -2.3438 | -2.4927 |
| Median.50% | 0.0798 | 0.3734 | 1.3071 | 12.6853 |
| Quant.75% | 0.7491 | 1.9288 | 4.4300 | 23.9579 |
| Quant.95% | 2.1222 | 4.6248 | 8.7345 | 35.0503 |
| Max | 10.7975 | 14.9422 | 19.3738 | 38.3039 |
| Jarque.Bera.stat | 15723.2443 | 2619.8009 | 163.5216 | 8.7933 |
| Jarque.Bera.pvalue.X100 | 0.0000 | 0.0000 | 0.0000 | 1.2319 |
| Lillie.test.stat | 0.0696 | 0.0581 | 0.0706 | 0.1448 |
| Lillie.test.pvalue.X100 | 0.0000 | 0.0000 | 0.0066 | 7.6587 |
| KS.test.stat | 0.4766 | 0.4636 | 0.4336 | 0.3508 |
| KS.test.pvalue.X100 | 0.0000 | 0.0000 | 0.0000 | 0.0389 |
| AD.test.stat | 93.9694 | 12.9727 | 4.0721 | 1.0001 |
| AD.test.pvalue.X100 | 0.0000 | 0.0000 | 0.0000 | 1.0699 |
| N.obs | 8333 | 1722 | 396 | 33 |

**Table 1:** Summary statistics and distribution tests for DAX log-returns from ”adjusted closing” prices from 31-12-1987 to 30-12-2020.
Mean, st. deviation, min,max, quantiles are multiplied by 100, so their order of magnitude is that of percentage returns. P-values for the Jarque-Bera, Lilliefors, Kolgomorov-Smirnoff and Anderson-darling tests are also multiplied by 100, so to reject the the null hypothesis $H_0$ at a confidence level of= 5% they must be smaller than 5.

## 1.1 Stylized fact 1: DAX prices are non-stationary

To test for (weak) stationarity of prices it is first useful to look at the plot of the time series. As can be noted by inspection of **figure 1** both daily prices and log-prices exhibit long run increasing trends of fairly non-linear characteristics. The trend seems also non-deterministic as the index experiences prolonged swings during crises and economic downturns like the 2007-2008 financial crisis, the early-2000s german stagnation period or the recent Covid-19 market crash.

Another evidence towards the process being non stationary is given by the plot of auto-correlation of prices, which shows a slow decaying ACF, hinting towards a long memory process.

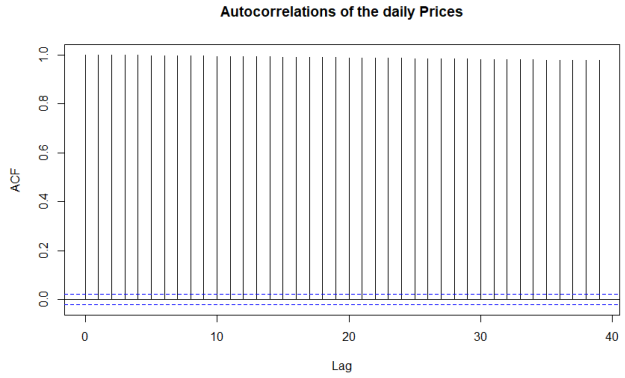


**Figure 2:** DAX daily prices autocorrelation for sample from 30-12-1987 to 30-12-2020

This non-stationary property can be tested through a Augmented Dickey-Fuller test with stationarity as alternative hypothesis.
The Augmented Dickey-Fuller (ADF) test is a unit root test checking whether modelling the time series as an $AR(p)$ process it would result in a unit root to the characteristic equation.
From time series theory it is known that $AR(p)$ containing an unit root is not stationary.
The Augmented Dickey-Fuller test is a generalization of the original Dickey-Fuller which makes use of an $AR(1)$ process. By including lags of order $p$, it allows for higher order auto-regressive processes.

Performing hypothesis testing with significance value at 5% we fail to reject the null hypothesis by p-value comparison for daily Prices and daily, weekly and monthly log-prices.
This confirms the stylised fact that stock prices are non-stationary.

| Time series | statistic | p.value |
|---|---|---|
| Daily Prices | 'Dickey-Fuller' = -2.83628443079205 | 0.224094206 |
| Daily log-prices | 'Dickey-Fuller' = -2.82790332535581 | 0.227644722 |
| Weekly log-prices | 'Dickey-Fuller' = -2.61560144300804 | 0.317711148 |
| Monthly log-prices | 'Dickey-Fuller' = -2.73228143736849 | 0.267976755 |
| Annual log-prices | 'Dickey-Fuller' = -2.75150795366797 | 0.281825667 |

**Table 2:** Dick-Fuller stationarity test for DAX Prices and log-prices, sample from 30-12-1987 to 30-12-2020

## 1.2 Stylized fact 2: DAX Returns are stationary

Similarly to the first stylised fact, Dick-Fuller test can be performed on daily, weekly and monthly log-returns. Hypothesis testing at 5% level, the null hypothesis that log-returns are non stationary is rejected for all frequencies excep for annual returns which show less evidence of stationarity. We accept stationarity for Annual log returns at 10% level.
The stationarity of the log-returns time series can also be appreciated by looking at the plot of daily log-return in **figure 1**.

| Time series | statistic | p.value |
|---|---|---|
| Daily log-returns | 'Dickey-Fuller' = -20.2046963595434 | 0.01 |
| Weekly log-returns | 'Dickey-Fuller' = -12.7353364426589 | 0.01 |
| Monthly log-returns | 'Dickey-Fuller' = -6.61414333427625 | 0.01 |
| Annual log-returns | 'Dickey-Fuller' = -3.31884057349812 | 0.0863 |

**Table 3:** Dickey-Fuller stationarity test for DAX returns and log-returns, same sample as above

## 1.3 Stylized fact 3: DAX Returns are asymmetric

To test for asymmetry of returns the 3rd central moment (Skewness) of the Data Generating Process has to be estimated. If the estimate of the skewness of the data is different from zero then the data is said to be asymmetric.
To do so the sample estimator is calculated as:

$$\hat{S} = \hat{S}(r_T) := \frac{\frac{1}{T}\sum_{t=1}^{T}(r_t - \overline{r})^3}{\left[\frac{1}{T}\sum_{t=1}^{T}(r_t - \overline{r})^2\right]^{3/2}}$$

while we assume that the data is $\overset{iid}{\sim} DGP$ and $\overline{r}$ is the sample mean.
A standard results in probability theory show $\hat{S}$ is a (weakly) consistent estimator of the theoretical moment.
As can be seen from **table 1** the log-returns present negative skewness at all frequencies. In particular, it can be noted that the skewness increases with the decreasing of the sampling frequency. This is consistent with the stylised facts for stocks that highlight that stocks tend to have asymmetric returns and in particular these tend to be negatively skewed as investors react more strongly to downturns.
Visually this can be also appreciated by plotting the histogram of returns and comparing it with normal returns like in **figure 3** and taking into consideration the greater contribution of the fatter left tail.

## 1.4 Stylized fact 4: DAX Returns have heavy tails

To test for the heaviness of tails of the distribution of log-returns the 4th central moment (Kurtosis) of the Data Generating Process has to be estimated. If the estimate of the kurtosis of the data is greater than 3 (which is the the kurtosis value of a normal distribution) then the data is said to have excess kurtosis or "has fat tails".
To estimate the empirical kurtosis, the sample estimator is calculated as:

$$\hat{K} = \hat{K}(r_T) := \frac{\frac{1}{T}\sum_{t=1}^{T}(r_t - \overline{r})^4}{\left[\frac{1}{T}\sum_{t=1}^{T}(r_t - \overline{r})^2\right]^{2}}$$

while we assume that the data is $\overset{iid}{\sim} DGP$ and $\overline{r}$ is the sample mean.
A standard results in probability theory shows $\hat{K}$ is a (weakly) consistent estimator of the theoretical 4th central moment.
As can be seen from **table 1** the log-returns present excessive kurtosis at all frequencies. In particular, it can be noted that the excessive kurtosis decreases with the decreasing of the sampling frequency. This is consistent with the stylised facts for stocks that highlight that stocks tend to have fat tails and in particular the left tail tends to be significantly fatter due to market crashes and strong market downturns.

Visually, this can be also appreciated by the Quantile-Quantile plot or "QQ-plot" ( **Figure 3**) of the empirical quantiles vs the quantiles of a normal distribution with same mean and standard deviation as the empirical distribution of log-returns.
The QQ-plot shows that the distribution of returns has fatter tails than a normal distribution as the points on the left side of the graph are significantly below the $y = x$ line ,while the ones on the right side are above. These results are valid for daily, monthly, weekly and annual returns, although there seems to be less departure from the $y = x$ line as the sampling frequency decreases. This is consistent with stylised fact number 5.

Interestingly, **Figure 4** shows that the quantiles of the distribution seem to be quite compatible with a Student-t distribution with $\nu = 4$ or $\nu = 5$ degrees of freedom at a daily level.

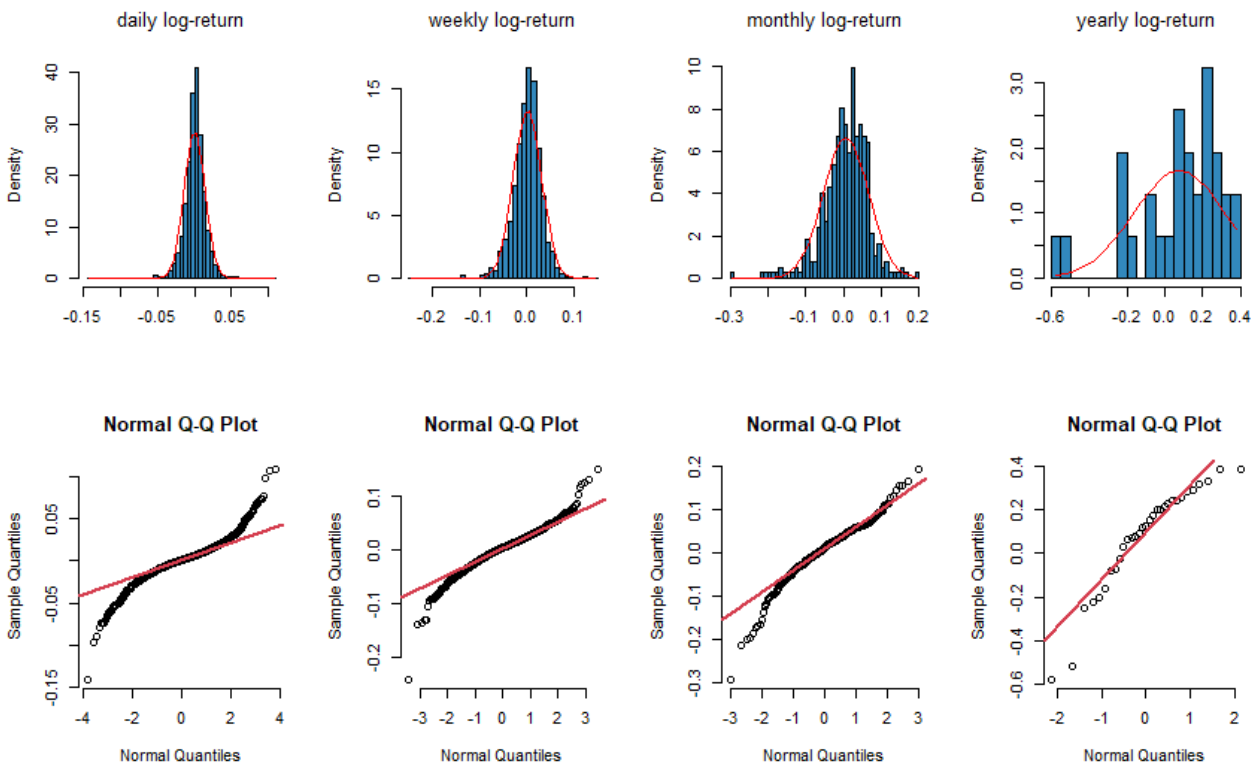


**Figure 3:** Log returns $r_t := p_t - p_{t-1}$ histograms of daily, weekly, monthly and yearly "adjusted closing" of DAX from Yahoo.com. Sample: 30-12-1987 to 30-12-2020. QQ plot against quantiles of normal distribution with same mean and variance as the empirical distribution of returns

## 1.5 Stylized fact 5: DAX High frequency non-Gaussianity

Empirical evidence of non-gaussianity of the returns as already been found by the negative skewness, large kurtosis and fat-tails of the empirical distribution. The departure from gaussianity can be furthermore tested by Jack-Bera (for skewness and kurtosis) and goodness of fit tests like Kolmogorov-Smirnov, Lilliefors and Anderson-Darling.

Performing the tests it can be seen from **Table 1** that Jack-Bera test shows that kurtosis and skewness are incompatible with the ones of a normal distribution at all frequencies. It must also be noted that there is weaker evidence towards the null hypothesis at annual frequency.

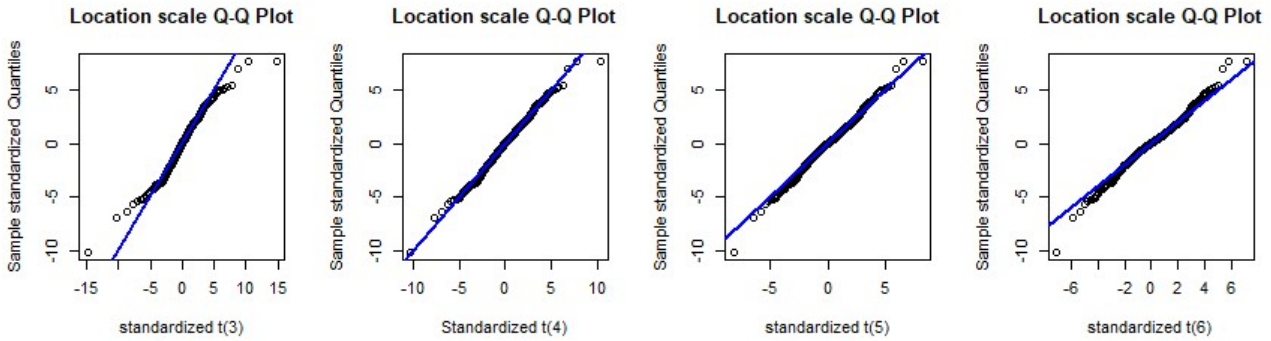


**Figure 4:** Log returns $r_t := p_t - p_{t-1}$: daily "adjusted closing" of DAX. Sample: 30-12-1987 to 30-12-2020. QQ plot of Sample standardized quantiles (0 mean and unit variance) of daily log-returns against quantiles of standardized (0 mean and unit variance) Student-t distributions with $\nu = 3$, 4, 5 and 6 degrees of freedom

Also from **Table 1** and **Figure 5** it can be seen that Lillefors test rejects normality assumption at 5% for all

frequency except at annual level. KS and Anderson-Darling tests reject normality assumption at 5% for all frequencies.

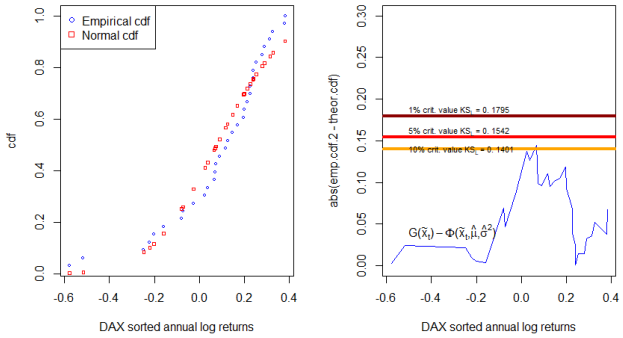


**Figure 5:** Log returns $r_t := p_t - p_{t-1}$: annual "adjusted closing" of DAX. Sample: 30-12-1987 to 30-12-2020. Left panel: empirical and Normal cdf's for the standardized annual returns of the DAX. Right panel: values $|G_T(\tilde{r}_t) - \Phi(\tilde{r}_t, \hat{\mu}, \hat{\sigma}^2)|$ (blue line) and critical values for the Lilliefors test for the three significance levels 10%, 5% and 1%.

## 1.6 Stylized fact 6: DAX Returns are not autocorrelated

To test for auto-correlation of returns we use three different methods.
First, we test the autocorrelation of returns by individual asymptotic tests on the distribution of each correlogram $\hat{\rho}_k$ of lag $k$. The results are here presented in the form of **Figure 6**.
On a daily and weekly scale ACF are significantly different from 0, but very small in absolute value. For monthly returns, ACF are NOT significantly different from 0. The annual returns are not significantly different from zero. Overall, there is little or no evidence of autocorrelation among DAX returns.

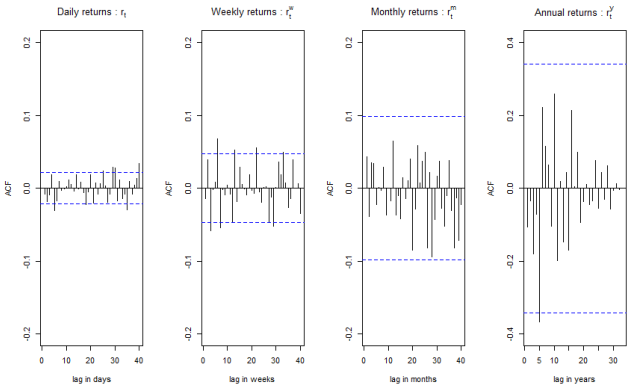


**Figure 6:** Empirical Autocorrelation (ACF) of $r_t$ , the daily, weekly, monthly and annual log-return from the "adjusted closing" of DAX. The autocorrelation of order 0 is not reported to have a better graphical representation of the smaller autocorrelations of order k. Sample: 30-12-1987 to 30-12-2020. Data source: Yahoo.com

The final two methods are the standard Ljung-Box and Box-pierce tests of the distribution of the sum of the autocorrelations $\hat{\rho}_1 + \hat{\rho}_2 + \ldots + \hat{\rho}_p$ . The two tests rely on the same hypothesis: $H_0 : \hat{\rho}_1 = \hat{\rho}_2 = \ldots = \hat{\rho}_p = 0; H_1 : \exists j : \rho_j \neq 0$ for $J = 1, \ldots, p$ and their test statistics are both distributed $\overset{H_0, asy}{\sim} \chi^2_p$ but their values are different as

$$Q_{BP} := T \sum_{j=1}^{p} \hat{\rho}_j{}^2 \qquad Q_{LB} := T(T+2) \sum_{j=1}^{p} \frac{\hat{\rho}_j{}^2}{T+j}$$

As can be seen from **table 4** the daily returns do show weak signs of autocorrelation. While monthly data shows very weak signs of autocorrelation confirming at these frequencies the stylised fact.

| lag | acf | acf diam. | acf test | Box-Pierce stat | BP pval | LB stat | LB pval | crit |
|---|---|---|---|---|---|---|---|---|
| 1 | -0.007 | 0.021 | -0.620 | 0.385 | 0.535 | 0.385 | 0.535 | 3.841 |
| 2 | -0.018 | 0.021 | -1.650 | 3.108 | 0.211 | 3.109 | 0.211 | 5.991 |
| 3 | -0.008 | 0.021 | -0.763 | 3.690 | 0.297 | 3.692 | 0.297 | 7.815 |
| 4 | 0.019 | 0.021 | 1.749 | 6.748 | 0.150 | 6.752 | 0.150 | 9.488 |
| 5 | -0.031 | 0.021 | -2.852 | 14.881 | 0.011 | 14.892 | 0.011 | 11.070 |
| 6 | -0.017 | 0.021 | -1.574 | 17.357 | 0.008 | 17.370 | 0.008 | 12.592 |
| 7 | 0.008 | 0.021 | 0.756 | 17.929 | 0.012 | 17.943 | 0.012 | 14.067 |
| 8 | -0.002 | 0.021 | -0.207 | 17.972 | 0.021 | 17.986 | 0.021 | 15.507 |
| 9 | -0.001 | 0.021 | -0.114 | 17.985 | 0.035 | 17.999 | 0.035 | 16.919 |
| 10 | 0.004 | 0.021 | 0.354 | 18.110 | 0.053 | 18.124 | 0.053 | 18.307 |

**Table 4:** Empirical Autocorrelation (ACF), ACF "diameter" ($1.96 \times \sqrt{\frac{1}{T}}$ ), Box-Pierce (BP) test and Ljung-Box test (LB): statistics and p-values. Data: rt , the daily log-return from the "adjusted closing" of DAX. Sample: 30-12-1987 to 30-12-2020. Data source: Yahoo.com

| lag | acf | acf diam. | acf test | Box-Pierce stat | BP pval | LB stat | LB pval | crit |
|---|---|---|---|---|---|---|---|---|
| 1 | -0.008 | 0.021 | -0.730 | 0.534 | 0.465 | 0.534 | 0.465 | 3.841 |
| 2 | -0.018 | 0.021 | -1.671 | 3.327 | 0.190 | 3.328 | 0.189 | 5.991 |
| 3 | -0.009 | 0.021 | -0.782 | 3.938 | 0.268 | 3.940 | 0.268 | 7.815 |
| 4 | 0.019 | 0.021 | 1.716 | 6.884 | 0.142 | 6.888 | 0.142 | 9.488 |
| 5 | -0.030 | 0.021 | -2.745 | 14.421 | 0.013 | 14.431 | 0.013 | 11.070 |
| 6 | -0.017 | 0.021 | -1.569 | 16.883 | 0.010 | 16.896 | 0.010 | 12.592 |
| 7 | 0.009 | 0.021 | 0.817 | 17.551 | 0.014 | 17.564 | 0.014 | 14.067 |
| 8 | -0.002 | 0.021 | -0.228 | 17.602 | 0.024 | 17.616 | 0.024 | 15.507 |
| 9 | -0.001 | 0.021 | -0.049 | 17.605 | 0.040 | 17.618 | 0.040 | 16.919 |
| 10 | 0.002 | 0.021 | 0.213 | 17.650 | 0.061 | 17.664 | 0.061 | 18.307 |

**Table 5:** Empirical Autocorrelation (ACF), ACF "diameter" ($1.96 \times \sqrt{\frac{1}{T}}$ ), Box-Pierce (BP) test and Ljung-Box test (LB): statistics and p-values. Data: rt , the monthly log-return from the "adjusted closing" of DAX. Sample: 30-12-1987 to 30-12-2020. Data source: Yahoo.com

## 1.7 Stylized fact 7: DAX Returns feature volatility clustering

Results from the autocorrelation tests on the absolute value of returns presented in **Figure 7** confirm the stylised of volatility clustering. Volatility clustering means that large price changes occur in clusters. This implies significant autocorrelation of squared returns (ARCH effect) or absolute returns.
This is confirmed by evidence in **Figure 7** . The daily squared returns show persistent correlation that decay to zero very slowly as k increases. This indicates long-memory properties in higher moments of returns. Also it confirms the corollary fact that volatility clustering is less relevant at lower frequencies as both monthly and annual data do not show sign of strong autocorrelation.

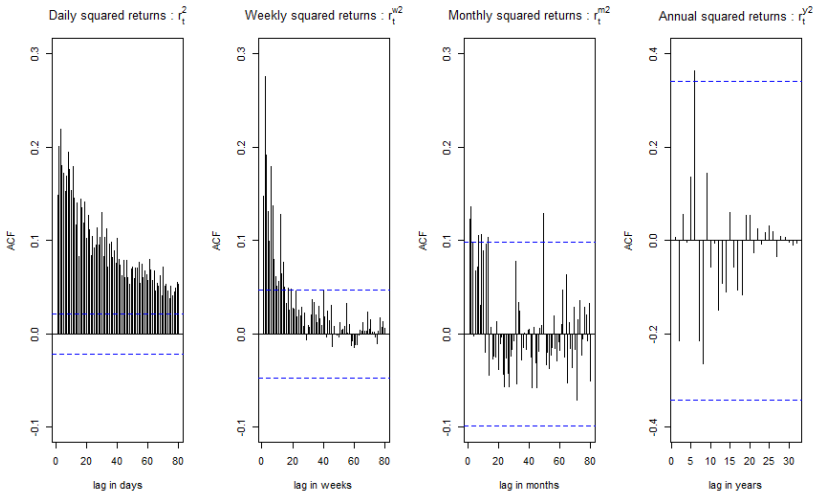


**Figure 7:** Empirical Autocorrelation of the squared value of daily, weekly, monthly and annual log-return from the "adjusted closing" of DAX from Yahoo.com. Sample: 30-12-1987 to 30-12-2020.

## 1.8 Stylized fact 8: DAX Returns show leverage effect

The leverage effect is defined as the negative autocorrelation between asset returns and the change in their volatilities. In particular, it is expected that asset-return volatility rises after the price declines and the magnitude of the volatility spikes is positively correlated with the magnitude of the decline.
In other words we would expect $corr(r_{t-j}, r_t^2) < 0$ for $j > 0$ which is confirmed by result on DAX analysis in **figure 8**. As can be seen from the LHS of the figure, the cross correlation with the past value at a daily level is negative and significant.

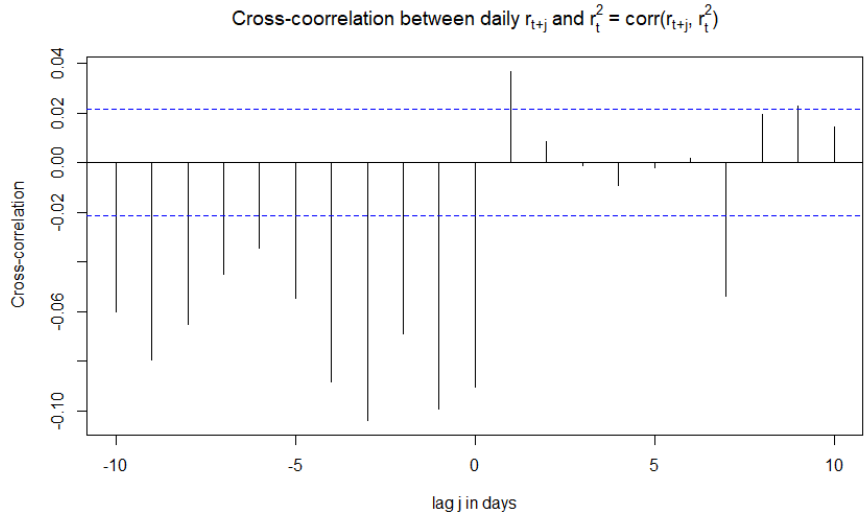


**Figure 8:** Empirical cross-correlation of daily lagged DAX log-returns, with squared returns: $corr(rt + j; r_t^2)$ for $j = -10; \ldots; +10$. Daily log-return from the "adjusted closing" of DAX from Yahoo.com. Sample: 30-12-1987 to 30-12-2020. Dashed blue lines are the (asymptotic) bounds for the rejection region a significance test of each cross-correlation. A line rising above or below the blue dashed lines represent a significant cross-correlation.

Another test to check evidence of leverage effect would be to compare the DAX with the apropriate volatility index the VDAX. In this case, we would check for the negative correlation between the DAX returns and the changes in the VDAX index ($VDAX_t - VDAX_{t-1}$).The VDAX-NEW index expresses the implied volatility of the DAX anticipated on the derivatives market. It indicates the future volatility to be expected in the next 30 days for the DAX. However, this index is not yet available to download on yahoo finance.

# 2 Question 2

## 2.1 Question 2.a

*Using the above information, perform the test for the presence of autocorrelation in the innovations of the model with a significance level of 10% and discuss the results.*

From the information given about the auxiliary regression on the residuals, we can perform a test for the conditional autocorrelation of the residuals like the Breusch-Godfrey test with $p = 5$ lags.

The Breusch-Godfrey test for autoautocorrelation up to order $p = 5$ is based on the auxiliary regression:

$$\hat{\epsilon}_t = \beta_1 + \sum_{k=2}^{K} \beta_k x_{k,t} + \gamma_1 \hat{\epsilon}_{t-1} + \ldots + \gamma_5 \hat{\epsilon}_{t-5} + u_t$$

where $\hat{\epsilon}_t$ denotes the OLS residual at time t. The null and alternative are:

$$H_0 : \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = 0 \implies \textit{No autocorrelation}$$

$$H_1 : \exists \gamma_k \neq 0 \implies \textit{autocorrelation}$$

Bresuch-godfrey proved that $(T-p)R^2$ is asymptotically distributed under $\overset{H_0}{\sim} \chi^2_p$ where $R^2$ is the residual of the auxiliary regression above, $p$ is the number of lags and $T$ is the number of observations. Since the F-test assumes normality, the appropriate test to be used in this case is the
Given the adjusted $\bar{R}^2 = 0.025$, the unadjusted $R^2$ is given by $R^2 = 1 - (1 - \bar{R}^2)\frac{\bar{T}-\bar{K}}{\bar{T}-1} = 1 - (1 - 0.025)\frac{84-5-5-5}{78} = 0.1375$, this since the auxiliary regression runs on $\bar{T} = T - p$ data and is reduced by $\bar{K} = p + k$ degrees of freedom less due to the explanatory variables. Performing the Breusch-Godfrey test at significance level 10%, our test statistic is

$$(84-5)R^2 = 79 \times 0.1375 = 10.86 > \chi^2_{5;0.9} = 9.24$$

We reject the null hypothesis of autocorrelation at confidence level 10%. As $\chi^2_{5;0.95} = 11.07$ we would have failed to reject the null hypothesis at 5% level.

## 2.2 Question 2.b

*Given the results of the test you performed to answer Question 2.a), how would you construct appropriate standard errors for the coefficients of the regression model in equation (1)?*

Since we found (weak) evidence of autocorrelation in the Breusch-Godfrey test the OLS standard errors are not appropriate to be used as they rely on the assumption that residuals are not autocorrelated. The OLS estimator $\hat{\beta}$ is still unbiased consistent and asymptotically normally distributed. However, the OLS estimator $\hat{\beta}$ is generally no longer the Best Unbiased Estimator (BUE) for a parametric model or the Best Linear Unbiased Estimator (BLUE) for a semi parametric model like the one given.

The appropriate errors to be used in this case are the Newey-West which are a form of Heteroskedaticity and Autocorrelation consistent (HAC) errors.

$$\hat{SE}_{NW}(\hat{\beta}_k) = \sqrt{[\hat{V}(\hat{\beta})_{NW}]_{k,k}}$$

where

$$\hat{V}(\hat{\beta})_{NW} = \frac{1}{T}\left(\frac{X'X}{T}\right)^{-1} \frac{1}{T}\hat{V}(X'\epsilon)\left(\frac{X'X}{T}\right)^{-1}$$

and

$$\frac{1}{T}\hat{V}(X'\epsilon) = \frac{1}{T}\sum_{t=1}^{T} \epsilon_t^2 x_t x_t' + \frac{1}{T}\sum_{t=1}^{T-1}\sum_{s=1+t}^{T} w_{s-t}\epsilon_s\epsilon_t(x_t x_s' + x_s x_t')$$

where $w_h = 1$ for the diagonal elements or otherwise $w_h = 1 - \frac{h}{B}$ if $h < B$, else 0.
B is taken in practice to be approximately around $T^{\frac{1}{3}}$ or $T^{\frac{1}{5}}$ for large samples.

## 2.3 Question 2.c

*Are all the 3 "Fama and French factors" and the "Momentum" factor useful to explain the time-series variability of the excess returns of IBM in this sample? Answer to this question by performing only one appropriate test of hypothesis using the information/numbers in Tables 1 and/or 2 and, if necessary, your own calculations based on those numbers.*

According to question 2.a, it would be necessary to modify the following f-tests to be autocorrelation robust. To test for the joint usefulness of the factors a $\chi^2$-test is performed as the innovations in the original model are just assumed to be $iid(0,\sigma^2)$ but not normal.
The F statistic is given by $F^* = \frac{SSR_0 - SSR_1}{SSR_1}\frac{T-K}{K-1}$ where $K$ is the number of factors including the constant.

$$SSR_0 = (T-K_0)\hat{\sigma_0}^2 = 83 \times (5.731)^2 = 2726.082; \quad SSR_1 = (T-K_1)\hat{\sigma_1}^2 = 79 \times (4.504)^2 = 1602.595$$

Under null assumption $H_0 : \beta_M = \beta_{SMB} = \beta_{HML} = \beta_{MOM} = 0$ ( with alternative hypothesis $H_1 : \exists \beta_k \neq 0, k \neq 2$):

$$F^* \sim F_{K-1,T-K}$$

$$F^* = 0.7010\frac{79}{4} = 13.8456 > \chi^2_{4;0.99} = 13.2767$$

We reject the null hypothesis at 99% confidence level, rejecting that the factors are jointly useless.

## 2.4 Question 2.d

*Is the monthly risk premium of IBM different from zero? Explain clearly whether, and eventually how, you can answer to this question by performing an appropriate test of hypothesis using the information/numbers in Tables 1 and 2 and, if necessary, your own calculations based on those numbers. In case you can perform the test for this question, do it by using a significance level of 1%. In case you don't have enough information to perform the test, explain why, and which additional information/quantities you would need to be able to perform it. Finally, in case you can compute it with the information on the table, provide the unbiased estimate of the standard deviation of the returns of IBM.*

Since $z_t$ is already excess returns of IBM over the yield of a T-bill issued by the US Treasury with maturity 1 month, risk premium for IBM is given by $E(z_t) = E(\alpha_0 + u_t) = \alpha_0 + E(u_t) = \alpha_0$ as $u_t \sim iid(0, \sigma_u^2)$.
Performing the normal (asymptotic) two-sided test of significance at confidence level 5% on the t-statistic of $\alpha_0$ (due to lack of normality of errors assumption):

$$|T^*| = |\frac{\hat{\alpha_0}}{SE(\hat{\alpha_0})}| = |\frac{-0.3613}{0.6253}| = |-0.5778| < 2.5758 = \Phi^{-1}(0.995)$$

where $H_0 : \alpha_0 = 0$ and $H_1 : \alpha_0 \neq 0$.
We therefore fail to reject the null hypothesis and find that $\alpha_0$ is not statistically different from zero. The risk premium is not statistically different from zero $E(z_t) = 0$ .

The unbiased estimator of the returns is given by $\hat{\sigma} = \sqrt{\frac{RSS}{T-K}} = RMSE = 5.731$ which can be found calculated in the second table of the question as residual standard error.

## 2.5 Question 2.e

*Propose a way to modify the t-test of the null hypothesis of the test you proposed in question 2.d), which is consistent for both autocorrelation and heteroskedasticity of the excess returns $z_t$.*

In the presence of evidence of autocorrelation and heteroskedaticity of the the OLS standard errors are not appropriate to be used as they rely on the sphericity of errors. The OLS estimator $\hat{\beta}$ is still unbiased consistent and asymptotically normally distributed but not the most effficient estimator.
However, the OLS estimator $\hat{\beta}$ is generally no longer the Best Unbiased Estimator (BUE) for a parametric model or the Best Linear Unbiased Estimator (BLUE) for a semi parametric model like the one given.

A method to conduct the test of hypothesis, without having to use Maximum Likehood Estimator, which requires to impose a precise structure of autocorrelation and heteroskedaticity (and risking misspecification), is to use some robust standard errors.
Using HAC standard errors like Newey-West Standard Errors in the calculation of the t-statistic the estimates are consistent.

The recalculated autocorrelation and heteroskedaticity robust T-statistic is:

$$T^* = \frac{\hat{\alpha_0}}{\hat{SE}_{NW}(\hat{\alpha_0})}$$

Where $\hat{SE}_{NW}(\hat{\alpha_0}) = \sqrt{[\hat{V}(\hat{\alpha_0})_{NW}]_{1,1}}$

$$\hat{V}(\hat{\beta})_{NW} = \frac{1}{T}\left(\frac{X^{'}X}{T}\right)^{-1}\frac{1}{T}\hat{V}(X^{'}\epsilon)\left(\frac{X^{'}X}{T}\right)^{-1}$$

Since in this case $X = \mathbf{1}_{(T,1)}$, then $\hat{V}(\hat{\beta})_{NW} = \frac{\hat{V}(X^{'}\epsilon)}{T^2}$ where $\hat{V}(X'\epsilon)$ is estimated similarly to the answer in 2.b

# 3 Question 3

## 3.1 ARCH (p) model

A process $\{x_t, t \in \mathbf{Z}\}$ is said to be a ARCH(p) process if

$$X_t = Z_t \sigma_t$$

where $Z_t$ is a sequence of iid random variables with $\mathbf{E}(Z_t) = 0$ and $\mathbf{V}(Z_t) = 1$ , and $\sigma_t$ is a non-negative process s.t:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i}^2$$

with $\alpha_0 > 0, \alpha_i \in \mathbf{R}^+, \forall i < p, \alpha_p \in \mathbf{R}^*$ , and $\sum_{i=1}^{p} \alpha_i < 1$ .
Arch(p) process have a number of properties:

### 3.1.1 $X_t$ is a martingale difference

$\mathbb{E}[X_t|\mathcal{F}_{t-1}] = \sigma_t \mathbb{E}[Z_t|\mathcal{F}_{t-1}] = 0$ as $Z_t$ is iid over time.
This in turn implies
$\mathbb{C}ov(X_t, X_{t-k}) = \mathbb{E}[X_t, X_{t-1}] - \mathbb{E}[X_t]\mathbb{E}[X_{t-k}] = \mathbb{E}[Z_t]\mathbb{E}[\sigma_t X_{t-k}] - \mathbb{E}[\mathbb{E}[X_t|\mathcal{F}_{t-1}]]\mathbb{E}[\mathbb{E}[X_{t-k}|\mathcal{F}_{t-k-1}]] = 0$
by the law of iterated expectation and the iid of $Z_t$.
Which means there is no correlation between $X_t$ and $X_{t-k}$with $k \neq 0$.

### 3.1.2 $X_t^2$ is AR(p)

:
$X_t^2$ can be written as $X_t^2 = \sum_{i=1}^{p} \alpha_i X_{t-i}^2 + v_t$
where $v_t := \sigma_t^2[z_t^2 - 1]$ such that
$\mathbb{E}[v_t|\mathcal{F}_{t-1}] = \sigma_t^2 \mathbb{E}[Z_t - 1|\mathcal{F}_{t-1}] = 0$ by iid of $Z_t$.
Therefore, $v_t$ is the innovation term in the AR(p) model.
This makes that $\mathbb{C}ov(X_t^2, X_{t-k}^2) \neq 0$ for $k < p$ by standard AR(p) processes result. Rubin [2021]

### 3.1.3 $X_t$ is stationary

$\mathbb{E}[X_t] = \mathbb{E}[\mathbb{E}[X_t|\mathcal{F}_{t-1}]] = \mathbb{E}[0] = 0$ and

$$V(X_t) = E(\sigma_t^2) = \alpha_0 + \alpha_1 \mathbb{E}[\mathbb{E}[X_{t-1}^2|\mathcal{F}_{t-2}]] + \ldots + \alpha_p \mathbb{E}[\mathbb{E}[X_{t-p}^2|\mathcal{F}_{t-p-1}]] = \alpha_0 + \alpha_1 \mathbb{E}[\sigma_{t-1}^2] + \ldots + \alpha_p \mathbb{E}[\sigma_{t-p}^2]$$

Imposing stationarity of the Ar(p) process (i.e. $\mathbb{E}[\sigma_{t-k}^2] = \mathbb{E}[\sigma_t^2], \forall k, 1 \leq k \leq p$ which can be done by imposing that the absolute value of all the roots of the characteristic equation are bigger than one) derive:

$$V(X_t) = E(\sigma_t^2) = \frac{\alpha_0}{1 - \alpha_1 - \ldots - \alpha_p}$$

where condition $\sum_{i=1}^{p} \alpha_i < 1$ is necessary for the variance to be positive and finite.
Since autocovariance is constantly equal to zero for lags greater than zero, all conditions for 2nd order stationarity are satisfied.

### 3.1.4 Ar(p) is marginally non-gaussian

This, similarly to an Ar(1) process can be seen by calculating the conditional kurtosis which under certain (more complex) parameters condition it remains leptokurtic.

Therefore if we take $X_t$ to model the returns of a stock we can see that the model would agree with many of the stylised facts:

- Martingale difference $\implies$ modeled returns are not autocorrelated
- $X_t$ stationarity $\implies$ modeled returns are stationary

- $X_t^2$ is AR(p) $\implies$ modeled returns feature volatility clustering and autocorrelation
- $X_t$ is leptokurtic $\implies$ modeled returns have heavy tails
- $X_t$ is marginally non-gaussian distributed and conditionally normal distributed if $Z_t$ is normal distributed $\implies$ High-frequency non-gaussianity and aggreagate gaussianity.

Therefore an ARCH(p) process does a good job in modelling financial returns however two facts are missing which are Asymmetry and the Leverage effect.

## 3.2 GARCH(1,1) model

A process $\{\epsilon_t, t \in \mathbf{Z}\}$ is said to be a GARCH(1,1) process if

$$\epsilon_t = Z_t \sigma_t$$

where $Z_t$ is a sequence of iid random variables with $\mathbb{E}(Z_t) = 0$ and $\mathbb{V}(Z_t) = 1$ , and conditional variance:

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

with $\omega > 0$, parameters $\alpha, \beta \in \mathbf{R}_{\geq \mathbf{0}}$ , and $\alpha + \beta < 1$. As can be seen the conditional variance depends on two elements:

- An intrinsic persistence effect through the first lag of the conditional variance $\beta \sigma_{t-1}^2$.
- An extrinsic persistence effect through the first lag of itself $\alpha \epsilon_{t-1}^2$

Garch(1,1) processes have a number of properties similar to ARCH(p) that agree with the stylised facts on returns. Under certain conditions of the parameters:

- $\epsilon_t$ Martingale difference $\implies$ modeled returns are not autocorrelated
- $\epsilon_t$ stationarity $\implies$ modeled returns are stationary
- $\epsilon_t^2$ is ARMA(1,1) $\implies$ modeled returns feature volatility clustering and autocorrelation
- $\epsilon_t$ is leptokurtic $\implies$ modeled returns have heavy tails
- $\epsilon_t$ is marginally non-gaussian distributed and conditionally normal distributed if $Z_t$ is normal distributed $\implies$ High-frequency non-gaussianity and Aggreagate gaussianity.

Another interesting property is that using the student-t distributed $Z_t$ it is possible to increase additionally the fat tails to better model more leptokurtic returns. Furthermore, using a skewed t-distribution allows to reproduce the skewness of financial returns. Rubin [2021]

## 3.3 AGARCH

consider modelling the financial returns time series as $r_t = \mu + \epsilon_t$, where $\mu$ is $\mathbb{E}[r_t]$ and $\epsilon_t$ is zero-mean white noise process. The process $\epsilon_t, t \in \mathbb{Z}$ is said to be AGARCH(1,1) if

$$\epsilon_t = Z_t \sigma_t$$

where $Z_t$ is a sequence of iid random variables $\sim \mathcal{N}(0,1)$ and conditional variance:

$$\sigma_t^2 = \omega + \alpha(\epsilon_{t-1} - \gamma)^2 + \beta \sigma_{t-1}^2$$

with usual restrictions $\omega > 0$, parameters $\alpha, \beta \in \mathbf{R}_{\geq \mathbf{0}}$ , and $\alpha + \beta < 1$ (this last to have mean reverting volatility). As can be seen the conditional variance depends on two elements:

- An intrinsic persistence effect through the first lag of the conditional variance $\beta \sigma_{t-1}^2$.
- An extrinsic persistence effect through the first lag of itself with a parameter $\gamma$ that determines dampening or increasing of this persistence effect $\alpha(\epsilon_{t-1} - \gamma)^2$

Note how no restriction was made on the values that $\gamma$ can assume and if $\gamma = 0$ AGARCH(1,1) is equivalent to GARCH(1,1). Vlab Campos-Martins [2021] finds that the AGARCH models all the stylised facts of financial returns modeled by GARCH(1,1) (see above section for reference) plus the leverage effect. The leverage effect is the empirically observed fact that negative shocks at time t1 have an asymmetrically stronger impact on the variance at time t than positive shocks.
This is modeled in AGARCH by the parameter $\gamma$ when positive.
When the past shock has been negative $\epsilon_{t-1} < 0$, the negative gamma coefficient increases the squared volatility at time $t$ by quantity $\alpha(-2\epsilon_{t-1}\gamma + \gamma^2)$ when compared to the GARCH(1,1) model.
Also similarly it dampens positive shocks by the same quantity, reducing the squared volatility when compared to GARCH(1,1).

V-Lab Campos-Martins [2021] uses MLE to estimate all the parameters $(\omega, \alpha, \gamma, \beta)$ simultaneously.
Like in the GARCH(1,1) model assuming that $z_t$ is Gaussian does not imply the the returns are Gaussian. Even though their conditional distribution is Gaussian, it can be proved that their unconditional distribution presents excess kurtosis (fat tails).

## References

Mirco Rubin. Financial econometrics with r. 2021. pages 13, 14, 15

Engle; Capellini; Reis; DeNard; Campos-Martins. Volatility lab. ://vlab.stern.nyu.edu/docs/volatility/AGARCH, 2021. pages 15

# 4 Appendix: R code for Question 1 Rubin [2021]

```
# 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)

# install.packages("RColorBrewer")

library(RColorBrewer)

#library for root test

library(tseries)


# WARNING:installation of package "portes" does not work as it should, as the file has been taken out form CRAN due to compativbility issues, still we need it!

# install.packages('forecast')

#install.packages("portes_3.0.tar.gz", repos=NULL, type="source")

# library(forecast)

library(portes)
```

```
rm(list=ls()) # clear all variables from environment/workspace
# set working directory (wd)
mydirectory <- "C:/Users/mmd2218/OneDrive - EDHEC/Desktop/Financial Econometrics with R"
setwd(mydirectory)
getwd() # set and show wd
# alternatively click on "Session" --> "Set Working Directory" --> "Choose Directory"
# WARNING: be sure that all the csv files are included in the same working directory!
################################################################################
# LOAD data
# DAX index
DAX<- getSymbols("^GDAXI",from="1987-12-30", to="2020-12-30", auto.assign=FALSE)
DAX<-na.omit(DAX)
write.zoo(DAX, file="Data_DAX.csv", col.names = TRUE, row.names = FALSE)
DAXcsv <-  as.xts(read.zoo("Data_DAX.csv", header = TRUE, format = "%Y-%m-%d"))
DAX <-  as.xts(read.zoo("Data_DAX.csv", header = TRUE, format = "%Y-%m-%d"))
Pt.d.all  <- DAX$GDAXI.Adjusted ; names(Pt.d.all)  <- "Pt.d" # Prices
pt.d.all  <- log(Pt.d.all)      ; names(Pt.d.all)  <- "pt.d" # log -prices

# find end of month/week/year dates
last_day_of_month <- endpoints(pt.d.all, on = "months") # works on xts objects
last_day_of_week  <- endpoints(pt.d.all, on = "weeks")
last_day_of_year  <- endpoints(pt.d.all, on = "years")

# Compute weekly (w), monthly(m), and annual(y) log prices
pt.w.all <- pt.d.all[last_day_of_week] ; names(pt.w.all)   <- "pt.w.all"
pt.m.all <- pt.d.all[last_day_of_month]; names(pt.m.all)   <- "pt.m.all"
pt.y.all <- pt.d.all[last_day_of_year] ; names(pt.y.all)   <- "pt.y.all"

# compute log returns # for entire history of DAX
rt.d.all  <- diff(pt.d.all)     ; names(rt.d.all)  <- "rt.d"
```

```
rt.w.all  <- diff(pt.w.all)     ; names(rt.w.all)  <- "rt.w"
rt.m.all  <- diff(pt.m.all)     ; names(rt.m.all)  <- "rt.m"
rt.y.all  <- diff(pt.y.all)     ; names(rt.y.all)  <- "rt.y"

ss_dates <- "19871230/20201230"



Pt.d <- Pt.d.all[ss_dates];
pt.d <- pt.d.all[ss_dates];
pt.w <- pt.w.all[ss_dates];
rt.d <- rt.d.all[ss_dates];
rt.w <- rt.w.all[ss_dates];
rt.m <- rt.m.all[ss_dates];
rt.y <- rt.y.all[ss_dates];

# 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";
#set up graph palette
brewer.pal(n=11, name="Spectral")
palette(brewer.pal(n=11, name="Spectral"))
#col2plot  <- c("red","blue","black","green")
lwd2plot  <- c(1    , 1     , 1    )
lty2plot  <- c(1    , 1     , 1    )
##############################################################################
# PLOT: Prices, log-prices and log-returns of SP500
```

```
################################################################################

#postscript(file="../figures/DAX_end1987_2020_Ppr.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

#pdf(returnsplot.pdf)

#puts figures into 3 rows in 1 col shape

par(mfrow=c(3,1));

#defines graph margins

par(mar = c(5, 4, 1, 4));

# par(mgp = c(1.5, 0.5, 0))



# Daily Price

data2plot <- Pt.d.df;

plot(x = data2plot[,1], y = data2plot[,2], type = 'l', col=2   , lty = lty2plot[1], lwd = lwd2plot[1],

    xlab=""  , ylab=TeX('price: $P_t$'), main="",   xaxt ="none")     # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq_sel <- endpoints(data2plot$date, on = 'years'); date_seq = data2plot$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

data2plot <- pt.d.df;

plot(x = data2plot[,1], y = data2plot[,2], type = 'l', col=3   , lty = lty2plot[1], lwd = lwd2plot[1],

    xlab=""  , ylab=TeX('log-price: $p_t$'), main="",   xaxt ="none")     # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq_sel <- endpoints(data2plot$date, on = 'years'); date_seq = data2plot$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

data2plot <- rt.d.df;

plot(x = data2plot[,1], y = data2plot[,2], type = 'l', col=4   , lty = lty2plot[1], lwd = lwd2plot[1],

    xlab=TeX("Date")  , ylab=TeX('log-return: $r_t$'), main="",   xaxt ="none")     # do not diaply x and y-axes labels/ticks)

# X-Axis display

seq_sel <- endpoints(data2plot$date, on = 'years'); date_seq = data2plot$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) # add zero line

dev.off()

# PLOT: log-returns of SP500

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

# Daily log-returns

data2plot <- rt.d.df;

plot(x = data2plot[,1], y = data2plot[,2], type = 'l', col=8  , 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(data2plot$date, on = 'years'); date_seq = data2plot$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

dev.off()

# scatterplot of log prices: p_t vs. p_{t-1}

# postscript(file="../figures/SP500_scatter_ptm1_pt.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

plot(Lag(pt.d.df[,2]), pt.d.df[,2], col = "blue", 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="red")



# compute autocorrelation of order 1 as sample correlation

cor(Lag(pt.d.df[,2]), pt.d.df[,2], use = "pairwise.complete.obs" )
```

```
dev.off()
#test for stationarity with dick-fueller root test
#since by inspection of graph it has no constant linear trend it is not worth using ADF-GLS

#test for Prices
#daily log prices
ADF.d.pt<-adf.test(pt.d.df$Pt.d)
#daily prices
ADF.d.Pt<-adf.test(Pt.d.df$pt.d)
#weekly log prices
ADF.w.pt<-adf.test(pt.w.df$pt.w.all)
#monthly log prices
ADF.m.pt<-adf.test(pt.m.all$pt.m.all)

ADF.y.pt<-adf.test(pt.y.all$pt.y.all)



#test for daily returns
ADF.d.rt<- adf.test(rt.d.df$rt.d[-1])
#test for weekly returns
ADF.w.rt<- adf.test(rt.w.df$rt.w[-1])
#test for monthly returns
ADF.m.rt<- adf.test(rt.m.df$rt.m[-1])

ADF.y.rt<- adf.test(rt.y.df$rt.y[-1])



ADF<-rbind(ADF.d.Pt,ADF.d.pt,ADF.w.pt,ADF.m.pt,ADF.y.pt,ADF.d.rt,ADF.w.rt,ADF.m.rt,ADF.y.rt)
#report table
ADF<- as.data.frame(ADF[,-c(2,5,6)])
ADF <- apply(ADF,2,as.character)
```

```
ADF

#write.csv(ADF,file='ADF2.csv')

#ACF

#postscript(file="../figures/SP500_ACF_pt.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

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

# acf of monthly prices for recent subsample: still persistent!

ss_dates_temp <- "20101231/20201231"

# Prices in subsamples

pt.m.temp <- pt.m.all[ss_dates_temp];

acf(pt.m.temp, main="Autocorrelations of the monthly Prices")

# Histogram of daily prices and normal density

# daily


#postscript(file="../figures/hist_qqplot_SP500.eps",width=9,height=6,horizontal = FALSE,
onefile=FALSE)

par(mfrow=c(2,4))

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

hist_OUT <- hist(data2plot, freq = FALSE, breaks = 50, col=10,  xlab="", main=TeX('daily log-return'), )

norm_y <-  dnorm(hist_OUT$mids, mean=mean(data2plot, na.rm=TRUE), sd=sd(data2plot,
na.rm=TRUE));

lines(x=hist_OUT$mids, y=norm_y,col="red", lwd=1)


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

hist_OUT <- hist(data2plot, freq = FALSE, breaks = 40, col=10,  xlab="", main=TeX('weekly log-
return'), )

norm_y <-  dnorm(hist_OUT$mids, mean=mean(data2plot, na.rm=TRUE), sd=sd(data2plot,
na.rm=TRUE));

lines(x=hist_OUT$mids, y=norm_y,col="red", lwd=1)


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

hist_OUT <- hist(data2plot, freq = FALSE, breaks = 40, col=10,  xlab="", main=TeX('monthly log-
return'), )
```

```
norm_y <-  dnorm(hist_OUT$mids, mean=mean(data2plot, na.rm=TRUE), sd=sd(data2plot, na.rm=TRUE));

lines(x=hist_OUT$mids, y=norm_y,col="red", lwd=1)


data2plot = rt.y.df[,2]; # yearly returns

hist_OUT <- hist(data2plot, freq = FALSE, breaks = 30, col=10,  xlab="", main=TeX('yearly log-return'), )

norm_y <-  dnorm(hist_OUT$mids, mean=mean(data2plot, na.rm=TRUE), sd=sd(data2plot, na.rm=TRUE));

lines(x=hist_OUT$mids, y=norm_y,col="red", 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 = 2, lwd = 2)


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

qqline(rt.w.df[,2], col = 2, lwd = 2)


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

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


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

qqline(rt.y.df[,2], col = 2, lwd = 2)



# QQ-plots: daily, weekly, monthly, annual

# daily


#postscript(file="../figures/qqplot_sp500annual.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

par(mfrow=c(2,2))
```

```
qqnorm(rt.d.df[-1,2], pch = 1, frame = FALSE, xlab="Normal Quantiles", main=TeX('daily log-return'))

qqline(rt.d.df[-1,2], col = 2, lwd = 2)


qqnorm(rt.w.df[-1,2], pch = 1, frame = FALSE, xlab="Normal Quantiles", main=TeX('weekly log-
return'))

qqline(rt.w.df[-1,2], col = 2, lwd = 2)


qqnorm(rt.m.df[-1,2], pch = 1, frame = FALSE, xlab="Normal Quantiles", main=TeX('monthly log-
return'))

qqline(rt.m.df[-1,2], col = 2, lwd = 2)


qqnorm(rt.y.df[-1,2], pch = 1, frame = FALSE, xlab="Normal Quantiles", main=TeX('annual log-
return'))

qqline(rt.y.df[-1,2], col = 2, lwd = 2)

# Kernel density of : similar to a smooth histogram!

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

data2plot <- rt.d.df[-1,2]; # daily returns

mean.data <- mean(data2plot, na.rm=TRUE)

sd.data   <- sd(data2plot, na.rm=TRUE)


density.eval.points <- seq(from = min(data2plot), to = max(data2plot), length.out = 300)


sm.density.out <- sm.density(data2plot, h = 0.05, eval.points = density.eval.points, dispaly="none",
col=1)

sm.density.out<- sm.density(data2plot, h = 0.0017, eval.points = density.eval.points, dispaly="none",
col=2)

sm.density.out <- sm.density(data2plot, h = 0.0016, eval.points = density.eval.points,
dispaly="none", col=3)

sm.density.out <- sm.density(data2plot, h = 0.0015, eval.points = density.eval.points,
dispaly="none", col=4)


# h is the 'bandwidth parameter': --> the larger the bandwidth, the smoother the histogram

sm.density.out.default <- sm.density(data2plot, eval.points = density.eval.points, dispaly="none")
```

```
# different bandwidth can produce different kernel densities.....


# R default function is:

d <- density(data2plot)

plot(d)

norm_y <-  dnorm(sm.density.out$eval.points, mean=mean(data2plot, na.rm=TRUE),
sd=sd(data2plot, na.rm=TRUE));

#stud_y<- dt(sm.density.out$eval.points,df=1)

lines(x=sm.density.out$eval.points, y=norm_y,col="red", lwd=2, lty=1)


# Kernel density and histogram, with normal distribution on top!

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


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

# Uses 'sm' package: controls parameters of kernel smoothing

#postscript(file="../figures/hist_kern_SP500daily.eps",width=12,height=6,horizontal = FALSE,
onefile=FALSE)

par(mfrow=c(1,2))

plot(x=sm.density.out$eval.points, y=sm.density.out$estimate ,xlim =c(-0.06,0.06), typ = "l", col=10,
lwd = 2,

    xlab = TeX('Daily returns'), ylab = TeX('Density'), main=TeX('Kernel density: daily log-return'))

norm_y <-  dnorm(sm.density.out$eval.points, mean=mean(data2plot, na.rm=TRUE),
sd=sd(data2plot, na.rm=TRUE));

lines(x=sm.density.out$eval.points, y=norm_y,col="red", lwd=2, lty=1)

abline(v = mean(data2plot), col="black", lwd=1, lty=2)

#legend("topleft", legend=c("Kernel desnity: data" , "Normal"),

#       col=c("blue", "red"), lwd=c(2,1),lty=c(1,2))


hist_OUT <- hist(data2plot, freq = FALSE, breaks = 70, col=10,  xlab = TeX('Daily returns'),
main=TeX('Histogram: daily log-return'), )

norm_y <-  dnorm(hist_OUT$mids, mean=mean(data2plot, na.rm=TRUE), sd=sd(data2plot,
na.rm=TRUE));
```

```
lines(x=hist_OUT$mids, y=norm_y,col="red", lwd=1, lty=1)

#legend("topleft", legend=c("Histogram: data" , "Normal"), col=c("lightgreen", "red"),
lwd=c(1.75,1),lty=c(1,1))


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

# simple plot: Kernel density of annual data

data2plot <- rt.y.df[-1,2]

sm.density(data2plot)


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

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

# # more advanced kernel density plot (NOT SHOWN IN THE SLIDES !!!!)

 density.eval.points <- seq(from = min(data2plot), to = max(data2plot), length.out = 300)

 sm.density.out <- sm.density(data2plot, h = 0.08, eval.points = density.eval.points)

 plot(x=sm.density.out$eval.points, y=sm.density.out$estimate ,xlim =c(-0.5,0.5), typ = "l",
col="blue", lwd = 2,

     xlab = TeX('Annual returns'), ylab = TeX('Density'), main="Kernel density plot vs Normal for
annual returns")

 norm_y <-  dnorm(sm.density.out$eval.points, mean=mean(data2plot, na.rm=TRUE),
sd=sd(data2plot, na.rm=TRUE));

 lines(x=sm.density.out$eval.points, y=norm_y,col="red", lwd=2, lty=2)

 abline(v = mean(data2plot), col="black", lwd=1, lty=2)

 points(data2plot, rep(0,length(data2plot)), pch='|')

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

# QQ plot of annual data

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

qqline(rt.y.df[,2], col = 2, lwd = 2)

# SUMMARY STATISICS

#####################################################################
```

```
#########################################################################
# very intuitive use of individual R functions
# some summary statistics...not really the ones we want!
summary(rt.d.df[,2])

# Sample Skewness,    note the difference:
skewness(rt.d.df[,2], na.rm = TRUE)
##MDV## #gives NA so need to remove NA
moments::skewness(rt.d.df[,2], na.rm= T)
# pay attention.....I have loaded the package moment, which contains a 'skewness'  function
#   but there are other packages with skewness functions....
# same with kurtosis!
# I prefer the skewness, kurtosis, and all other moments form the package 'moments'...
# which by default divides the summations of the estimators by 1/T ....


# Sample Kurtosis,
kurtosis(rt.d.df[,2], na.rm= T) # daily
kurtosis(rt.m.df[,2], na.rm= T) # monthly
kurtosis(rt.y.df[,2], na.rm= T) # annual -> much smaller!

#########################################################
# OPTIONAL : some nicer descriptive statics. Note that Skewness and Kurtosis are computed
differently from how we defined them in class!
# function form package 'summarytools'

mydata <- descr(rt.d.df[,2], style = "rmarkdown")
View(mydata)
st_options(round.digits=4)
#####goooood shit#####
# Produces latex:
```

```
summary_table(rt.d.df)

#######


print(descr(rt.m.df))

# compute Normality tests and sample summary statistics

library(tseries)

#Jarque-Bera Normality tests

jarque.bera.test(rt.d.df[-1,2])

jarque.bera.test(rt.w.df[-1,2])

jarque.bera.test(rt.m.df[-1,2]) # the '-1' removes the first observation (a NA) so that NA are not an issue!

jarque.bera.test(rt.y.df[-1,2])

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

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

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

# Produce Summary statistics for ANNUAL data only

data <- rt.y.df[-1,2]; T<-length(data)


# JB test statistics computed manually from formula in the slides:

((T/6)*(skewness(data))^2)+ (T/24*(kurtosis(data)-3)^2)

# JB test statistics computed form function: they are exactly the same...if moments package is used ot compute skenwss and kurtosis!

jb.norm.test(data)

jarque.bera.test(data)


# #perfomr JB test manually

a<-jarque.bera.test(data)$statistic

x <- data

n <- length(x)        ## Number of observations

m1 <- sum(x)/n        ## Mean

m2 <- sum((x-m1)^2)/n  ## Used in denominator of both

m3 <- sum((x-m1)^3)/n  ## For numerator of S
```

```
m4 <- sum((x-m1)^4)/n  ## For numerator of K
b1 <- (m3/m2^(3/2))^2  ## S
b2 <- (m4/m2^2)        ## K
STATISTIC <- n*b1/6+n*(b2-3)^2/24; STATISTIC

# p-value of JB statics: we can compute it! Computes quantile of chi^2
?pchisq
# method 1: compute area below chi2 for interval
# (-inf, STATISTICS)
one.minus.p_jb <- pchisq(STATISTIC,2);
1 - one.minus.p_jb

# method 2: compute area below chi2 for interval
# (STATISTICS, +inf)
p_jb <- pchisq(STATISTIC,2, lower.tail = F); p_jb
jarque.bera.test(rt.y.df[-1,2])



# different ways to compute skewness !!!!!
c(T,n)
c(skewness(data), m3/m2^(3/2))
c(moments::skewness(data), m3/m2^(3/2))
c(timeDate::skewness(data), m3/m2^(3/2))

# MANUAL COMPUTATION of  JB TEST FOR DAILY DATA
# T.JB <- 16862
# S.JB <- -1.0017
# K.JB <- 30.1344
# JB.stat <- (T.JB/6)*(S.JB^2) + (T.JB/24)*((K.JB-3)^2)
# JB.stat
# pchisq(520114,df=2)
```

```
# DIFFERENT NORMLAITY TESTS:
jarque.bera.test(data)          # Jarqu-Bera
ks.test(rt.y.df[-1,2], "pnorm")  # Kolmogorov Smirnov -->
ad.test(rt.y.df[-1,2])          # Anderson Dalring test for normlaity
lillie.test(rt.y.df[-1,2])      # Lilliefors test
lillie.test(rnorm(100, mean = 5, sd = 3)) # try Lilliefors test on normally distributed data

# GENERATES TABLE EXACTLY EQUAL TO THE ON IN SLIDE N. 91
##############################################################################
##############################################################################
# personalized table of summary statistics
# creates a 'list' with the data of DAX 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]);

# list of daily data with and without 87 crash!
#X.d.87crash.YN <-list("daily all" = rt.d.df[-1,2],
#                      "daily NO 87 Crash" =rt.d.all.noOct87crash.df[-1,2])

sum(is.na( rt.d.df[-1,2]))

#sum(is.na(rt.d.all.noOct87crash.df[-1,2]))
#which(is.na(rt.d.all.noOct87crash.df))
#  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,
   KS.test.stat=ks.test(x, "pnorm")$statistic,
   KS.test.pvalue.X100 =ks.test(x, "pnorm")$p.value*100,
   AD.test.stat =ad.test(x)$statistic,
   AD.test.pvalue.X100 =ad.test(x)$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 (you can copy-paste frm there to excel directly!)
print(a)
# and return results in a nice and tidy table
round(a, digits = 5) # show nicer-looking table!!!!!
```

```
# save TABLE  on .csv file (to be opened with excel/any other editor)
write.csv(a,file='table_try.csv')

# or do copy paste form variable 'editor 'pane'
############################################################
############################################################
############################################################
############################################################
# Compute summary statistics with / without 87 crash data crash data
# Skip in class: just repetition of what has been done above, with different dataset !
#a.crash.YN <- sapply(X.d.87crash.YN, multi.fun)
#round(a.crash.YN, digits = 5) # show nicer-looking table!!!!!

############################################################
############################################################
 # SHOWS TABLE IN LATEX, too advanced for Master students
# modify for the normal thing
 n_series <- length(X); n_stats  <- nrow(a)
 digits_m <- rbind(4*(matrix(1,n_stats-1,n_series+1)), 0*(matrix(1,1,n_series+1)))
 xtable(a, digits = digits_m)

############################################################
############################################################
# ACF of returns: Autocorrelation function

# use 'Acf function', and not 'acf', though they are very similar
```

```
#postscript(file="../figures/SP500_ACF_rt_dwm_1953_2018.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

par(mfrow=c(1,4))

 # chnage the limit

lag.max.acf = 40;  lim.y.axes = c(-0.2,0.2)

data2plot = rt.d.df[-1,2]; # daily returns

Acf(data2plot, main=TeX('Daily returns : $r_t$'), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes, xlim = c(1,40))

data2plot = rt.w.df[-1,2]; # weekly returns

Acf(data2plot, main=TeX('Weekly returns : $r_t^w$'), lag.max = lag.max.acf, xlab = "lag in weeks", ylim=lim.y.axes, xlim = c(1,40))

data2plot = rt.m.df[-1,2]; # monthly returns

Acf(data2plot, main=TeX('Monthly returns : $r_t^m$'), lag.max = lag.max.acf, xlab = "lag in months", ylim=lim.y.axes, xlim = c(1,40))


#no interesting information for yearly results andit just makes other graphs harder to present

lim.y.axes = c(-0.4,0.4)

data2plot = rt.y.df[-1,2]; # monthly returns

Acf(data2plot, main=TeX('Annual returns : $r_t^y$'), lag.max = lag.max.acf, xlab = "lag in years", ylim=lim.y.axes, xlim = c(1,33))

#-------------------------------------------------------------

# display ACF values on R-console: daily data

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)
#####################################################################
#####################################################################
# Box-Pierce and Ljung Box tests
# 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)
# test 20th lag
Box.test(my.data, lag = 20, type = c("Ljung-Box"), fitdf = 0)
Box.test(my.data, lag = 20, type = c("Box-Pierce"), fitdf = 0)

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

#to be used
# 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)

round(a, digits = 3) # show nicer-looking table!!!!!
tab.2.print <- base::round(a,digits=4)
# show on screen (latex format to export in latex directly from screen!)
 print(xtable(my.table.df, digits=c(0,0,3,3,3,3,3,3,3,3)), include.rownames=FALSE)
#-----------------------------------------------------------
# display ACF values on R-console: daily data
#my.data <- rt.d.df[-1,2];
 my.data <- rt.m.df[-1,2];
# my.data <- rt.w.df[-1,2]
# my.data <- rt.y.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))



#to be used
# 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)


round(a, digits = 3) # show nicer-looking table!!!!!

tab.2.print <- base::round(a,digits=4)

# show on screen (latex format to export in latex directly from screen!)

 print(xtable(my.table.df, digits=c(0,0,3,3,3,3,3,3,3,3)), include.rownames=FALSE)

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

# Squared RETURNS: ACF

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

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

# ACF of squared returns

#postscript(file="../figures/SP500_ACF_SQUAREDrt_dwm_1953_2018.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

par(mfrow=c(1,4))

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 months", ylim=lim.y.axes)

lim.y.axes = c(-0.40,0.40)

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

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


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

# ACF of absolute returns
```

```
# postscript(file="../figures/SP500_ACF_ABSrt_dwm_1953_2018.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

par(mfrow=c(1,3))

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 months", ylim=lim.y.axes)
```

---

```
# LEVERAGE

#------------------------------------------------------------

# CROSS-COVARINACE OF DAILY RETURNS with squared returns

# uses ccf function

# omputes the cross-correlation or cross-covariance of two univariate series.

# reading the help of the function you find:

# "The lag k value returned by ccf(x, y) estimates the correlation between x[t+k] and y[t]."


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

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




ss_dates3 <- "19871230/20201230"

rt.d.subsamp3 <- rt.d.all[ss_dates3];


# Cross correaltion plot in the slides


dev.off()
```

```
#postscript(file="../figures/SP500_corr_rtmj_rt2_1988_2018.eps",width=9,height=6,horizontal = FALSE, onefile=FALSE)

par(mfrow=c(1,1))

ss_dates3 <- "19871231/20201231"

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'))


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

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

#-------------------------------------------------------

# VIX vs. sp500 returns

# daily values of VIX:

#VIX.d.all  <- VIX$VIX.Adjusted ; names(VIX.d.all)  <- "VIX.d"


#a <- merge(pt.d,VIX.d.all) # merge two datasets (log prices and VIX ) with different legths --> NA are kept

#b <- diff(a)             # compute changes in pt and VIX conmpared to previus period


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

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


#a.comp <- a.df[complete.cases(a.df),] # remove all rows in which there is at least one NA -> generate balanced (complete) matrix

#b.comp <- b.df[complete.cases(b.df),]


#head(a.comp); tail(a.comp)

#head(b.comp); tail(b.comp)


# Level of SP500 and VIX ('a.comp' matrix)
```

```
################################################

#postscript(file="../figures/SP500_VIX_1988_2018.eps",width=8,height=6,horizontal = FALSE, onefile=FALSE)

#par(mfrow=c(1,1))

#data2plot <- a.comp;

#plot(x = data2plot[,1], y = data2plot[,2], type ="l", ylab=TeX("SP 500 (blue)"),    main=TeX('SP500 (left, blue) and VIX (right, red) indexes'),

#     xlab="", xaxt ="none",   col = "blue", lwd =2)

#par(new = TRUE)

#plot(x = data2plot[,1], y = data2plot[,3], type = "l", xaxt = "n", yaxt = "n",

#     ylab = "", xlab = "", col = "red", lty = 1)

#axis(side = 4)

#mtext("VIX", side = 4, line = 3)

#seq_sel <- endpoints(data2plot$date, on = 'years'); date_seq = data2plot$date[seq_sel]; date_lab = format(date_seq,"%b-%y")

#axis(1, at = date_seq, label = date_lab, las = 1, cex.axis=1.0);

#dev.off()

# changes of SP500 vs change VIX

# postscript(file="../figures/SP500_corr_rt_dVIX_1988_2018.eps",width=8,height=6,horizontal = FALSE, onefile=FALSE)

#par(mfrow=c(1,1))

#data2plot <- b.comp;

#plot(x = data2plot[,2], y = data2plot[,3], xlab=TeX("SP 500 log return"),    main=TeX('($VIX_t-VIX_{t-1}$) vs. $r_t$'),

#     ylab="VIX Change", col = "blue", lwd =1)

#abline(lm(data2plot[,3]~data2plot[,2]), col="red", lwd=2) # regression line (y~x)

#dev.off()

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

# cross - correlations in the different SUBSAMPLES (not in slides!)

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

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

# subsamples:
```

```
ss_dates1 <- "19900101/20181231"
ss_dates2 <- "19970101/20181231"
ss_dates3 <- "19980101/20181231" # we look at this sample: after 87 crash, till end 2018
ss_dates4 <- "20000101/20181231"

rt.d.subsamp1 <- rt.d.all[ss_dates1];
rt.d.subsamp2 <- rt.d.all[ss_dates2];
rt.d.subsamp3 <- rt.d.all[ss_dates3];
rt.d.subsamp4 <- rt.d.all[ss_dates4];
#---------------------------------------------------------------
dev.off()
par(mfrow=c(2,4))
# Cross-correlations between lagged and future return, and squared returns
# Subsamp 1
ret  <- as.numeric(rt.d.subsamp1); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "correlation",  plot = TRUE)
# Subsamp 2
ret  <- as.numeric(rt.d.subsamp2); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "correlation",  plot = TRUE)
# Subsamp 3
ret  <- as.numeric(rt.d.subsamp3); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "correlation",  plot = TRUE)
# Subsamp 4
ret  <- as.numeric(rt.d.subsamp3); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "correlation",  plot = TRUE)
#---------------------------------------------------------------
# cross-covariances
# Subsamp 1
ret  <- as.numeric(rt.d.subsamp1); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "covariance",  plot = TRUE)
# Subsamp 2
```

```
ret  <- as.numeric(rt.d.subsamp2); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "covariance",  plot = TRUE)
# Subsamp 3
ret  <- as.numeric(rt.d.subsamp3); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "covariance",  plot = TRUE)
# Subsamp 4
ret  <- as.numeric(rt.d.subsamp3); ret2 <- ret^2
ccf(ret, ret2, lag.max = 10, type = "covariance",  plot = TRUE)
######################################################################
######################################################################
######################################################################
######################################################################
######################################################################
######################################################################
#  Other manual computations to check results in the slides :
# Confidence interval
(0.9701/sqrt(16862))*qnorm(0.975)
cl = 0.0276 - (0.9701/sqrt(16862))*qnorm(0.975)
cu = 0.0276 - (0.9701/sqrt(16862))*qnorm(0.975)
ci = c(cl,cu)

stat.test = 0.0276/(0.9701/sqrt(16862))
0.009701*252
22.89/0.97
(-22.900-0.0276)/0.970
#################################################################################################
par(mfrow=c(1,4), pty="s")
 nu.t <- 3;
 my.data <- rt.d.df[,2];
 samp.size <- length(my.data);
```

```
seq.quant <- (seq(1,samp.size,1) -0.5)/samp.size;

y.qqplot <- scale(my.data);

x.qqplot <- (1/sqrt( (nu.t/(nu.t-2))) )*qt(seq.quant,df=nu.t);

qqplot(x.qqplot, y.qqplot, main="Location scale Q-Q Plot", ylab="Sample standardized  Quantiles",
xlab="standardized t(3)")

abline(0,1,col="blue", lwd = 2)


nu.t <- 4;

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

samp.size <- length(my.data);

seq.quant <- (seq(1,samp.size,1) -0.5)/samp.size;

y.qqplot <- scale(my.data);

x.qqplot <- (1/sqrt( (nu.t/(nu.t-2))) )*qt(seq.quant,df=nu.t);

qqplot(x.qqplot, y.qqplot, main="Location scale Q-Q Plot", ylab="Sample standardized Quantiles",
xlab="Standardized t(4)")

abline(0,1,col="blue", lwd = 2)


#

nu.t <- 5;

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

samp.size <- length(my.data);

seq.quant <- (seq(1,samp.size,1) -0.5)/samp.size;

y.qqplot <- scale(my.data);

x.qqplot <- (1/sqrt( (nu.t/(nu.t-2))) )*qt(seq.quant,df=nu.t);

qqplot(x.qqplot, y.qqplot, main="Location scale Q-Q Plot", ylab="Sample standardized Quantiles",
xlab="standardized t(5)")

abline(0,1,col="blue", lwd = 2)


nu.t <- 6;

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

samp.size <- length(my.data);

seq.quant <- (seq(1,samp.size,1) -0.5)/samp.size;
```

```
 y.qqplot <- scale(my.data);

 x.qqplot <- (1/sqrt( (nu.t/(nu.t-2))) )*qt(seq.quant,df=nu.t);

 qqplot(x.qqplot, y.qqplot, main="Location scale Q-Q Plot", ylab="Sample standardized Quantiles", xlab="standardized t(6)")

 abline(0,1,col="blue", lwd = 2)


 dev.off()

# ROBUSTENSS: same analysis with and without 87 crash !

# check results yourself ..... they are not in the slides still they are interesting.....

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

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

# ACF daily rtns with and without 87 crash

#par(mfrow=c(1,2))

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

#data2plot = rt.d.df[-1,2] ; # daily squared returns

#Acf(data2plot, main=TeX('Daily returns : $r_t$'), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes)

#data2plot = rt.d.all.noOct87crash.df[-1,2] ; # daily squared returns

#Acf(data2plot, main=TeX('Daily returns : $r_t$, NO 87 CRASH'), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes)


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

# ACF daily squared rtns with and without 87 crash

#par(mfrow=c(1,2))

#lag.max.acf = 80;  lim.y.axes = c(-0.10,0.40)

#data2plot = (rt.d.df[-1,2])^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.d.all.noOct87crash.df[-1,2])^2 ; # daily squared returns

#Acf(data2plot, main=TeX('Daily squaredvreturns : $r_t^2$, NO 87 CRASH' ), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes)


#################################################################
```

```
# ACF daily absolute rtns with and without 87 crash

# par(mfrow=c(1,2))

# lag.max.acf = 80;  lim.y.axes = c(-0.10,0.40)

# data2plot = abs(rt.d.df[-1,2]) ; # daily squared returns

# Acf(data2plot, main=TeX('Daily squared returns : $|r_t|$'), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes)

# data2plot = abs(rt.d.all.noOct87crash.df[-1,2]) ; # daily squared returns

# Acf(data2plot, main=TeX('Daily squaredvreturns : $|r_t|$, NO 87 CRASH' ), lag.max = lag.max.acf, xlab = "lag in days", ylim=lim.y.axes)


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

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

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

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

# ADVANCED: performs Lilliefors test figure as in slides...try to understand code by yourself !

# NOT IMPORTANT FOR THE EXAM!

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


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

# postscript(file="../figures/lillie_test_sp500annual.eps",width=12,height=6,horizontal = FALSE, onefile=FALSE)

par(mfrow=c(1,2))

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

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


my.data.ordered <- sort(my.data);

mean.data <-mean(my.data);

sd.data <-sd(my.data);

samp.size <- length(my.data)

seq.ind <- seq(1,samp.size,1);

emp.cdf <-seq.ind/samp.size

emp.cdf.2 <-(seq.ind-1)/samp.size
```

```
theor.cdf <- pnorm(my.data.ordered,mean.data,sd.data)
# plot(y=emp.cdf,  x=my.data.ordered, col="blue", lwd=1, lty=1)
# points(y=emp.cdf.2,  x=my.data.ordered, col="green", lwd=1, lty=1)

plot(y=emp.cdf,  x=my.data.ordered, col="blue", pch=1, lwd=1, lty=1, cex=0.5,
    xlab='DAX sorted annual log returns', ylab='cdf')
points(y=theor.cdf, x=my.data.ordered, col="red", pch=0,lwd=1, lty=1, cex=0.5)
legend("topleft", legend=c("Empirical cdf" , "Normal cdf"), pch=c(1,0), col=c("blue", "red"))

KS.L.stat1 = max(abs(emp.cdf-theor.cdf))
KS.L.stat2 = max(abs(emp.cdf.2-theor.cdf))
KS.L.stat = max(c(KS.L.stat1,KS.L.stat2))
KS.L.stat1
KS.L.stat2
KS.L.stat
plot(y=abs(emp.cdf.2-theor.cdf),x=my.data.ordered, type='l',
    xlab='DAX sorted annual log returns', ylim=c(0,0.3), col='blue')
abline(h=0.805/sqrt(samp.size),lwd = 4, lty = 1, col='orange')
abline(h=0.886/sqrt(samp.size),lwd = 4, lty = 1, col='red')
abline(h=1.031/sqrt(samp.size),lwd = 4, lty = 1, col='darkred')
text(x=-0.5, y=0.805/sqrt(samp.size)-0.006, TeX('10% crit. value $KS_L$ = 0.1401'),  adj = c(0,0), cex=
0.7)
text(x=-0.5, y=0.886/sqrt(samp.size)+0.002, TeX('5% crit. value $KS_L$ = 0.1542'),  adj = c(0,0), cex=
0.7)
text(x=-0.5, y=1.031/sqrt(samp.size)+0.002, TeX('1% crit. value $KS_L$ = 0.1795'),  adj = c(0,0), cex=
0.7)
text(x=-0.5, y=0.032, TeX('$ G(\\tilde{x}_t)-\\Phi(\\tilde{x}_t, \\hat{\\mu}, \\hat{\\sigma}^2)$'),  adj =
c(0,0))

# Critical values of Lilliefors test for T = length dataset
c(0.805,0.886,1.031)/sqrt(samp.size)
#########################################################################
```

```
# Rolling standard deviation on subset of daily data from 1980 to 2018

ss_dates <- "19871230/20201231"

rt.d.subsamp <- rt.d.all[ss_dates];

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


wind.length <- 252; # 1 year of daily data = 252 working days

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

# a <- rollapply(data = rt.d.subsamp.df[,2],

#             width = wind.length,

#             function(y){c(mean(y),sd(y),skewness(y),kurtosis(x))},

#             #by = 252,

#             align = "right",

#             by.column=FALSE)

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

# mean

roll.mom <- rollapply(data = rt.d.subsamp,

                width = wind.length,

                function(y){c(mean(y),sd(y),skewness(y),kurtosis(y),

                        moment(y, order = 4, central = TRUE))},

                align = "right", # save rolling statistics on the last date on which they are computed

                by.column=FALSE)

?rollapply

#

mean.plot    <- roll.mom[,1];

mean.plot.ub <- roll.mom[,1]+1.96*roll.mom[,2]/sqrt(wind.length);

mean.plot.lb <- roll.mom[,1]-1.96*roll.mom[,2]/sqrt(wind.length);

a <-cbind(mean.plot, mean.plot.lb, mean.plot.ub)

mean.plot.all <- cbind(index(a), data.frame(a)); names(mean.plot.all)[1] <- "date";
```

```
data2plot <- mean.plot.all[complete.cases(mean.plot.all),]

seq.dates.x <- data2plot[,1]

#postscript(file="../figures/SP500_MEAN_rolling_1981_2018.eps",width=12,height=6,horizontal =
FALSE, onefile=FALSE)

plot(x = seq.dates.x, y = data2plot[,2]*100, type = 'l', col="blue"   , lty = 1, lwd = 2,

    xlab=""  , ylab=TeX("mean (in percentage)"),    main=TeX('Rolling mean(on 252 days)  \\%'),   xaxt
="none",

    ylim=c(-0.5,0.5))     # do not diaply x and y-axes labels/ticks)

lines(x=seq.dates.x, y=data2plot[,3]*100, col="red", lwd=1, lty=1)

lines(x=seq.dates.x, y=data2plot[,4]*100, col="red", lwd=1, lty=1)

# just to produce nice x axis with dates

seq_sel  <- endpoints(data2plot$date, on = 'years');

date_seq <- data2plot$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 = 1) # add zero line

# compute rolling mean manually using look: get same results!

T <- length(rt.d.subsamp);

# creates empty matrix to store data

roll.mom.manual <- NA*matrix(0,T,5);

# run for loop

for (i in wind.length:T){

  est.window <- (i-wind.length+1):(i)

  y <- rt.d.subsamp[est.window]

  roll.mom.manual[i,1] <- mean(y)

  roll.mom.manual[i,2] <- sd(y)

  roll.mom.manual[i,3] <- skewness(y)

  roll.mom.manual[i,4] <- kurtosis(y)

  roll.mom.manual[i,5] <- moment(y, order = 4, central = TRUE)

}

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

# plot results of manually computed rolling mean!

mean.plot.man    <- roll.mom.manual[,1];
```

```
mean.plot.man.ub <- roll.mom.manual[,1]+1.96*roll.mom.manual[,2]/sqrt(wind.length);

mean.plot.man.lb <- roll.mom.manual[,1]-1.96*roll.mom.manual[,2]/sqrt(wind.length);

data2plot.NA <-cbind(mean.plot.man, mean.plot.man.lb, mean.plot.man.ub)


data2plot <- data2plot.NA[complete.cases(data2plot.NA),]

plot(x = seq.dates.x, y = data2plot[,1]*100, type = 'l', col="blue"   , lty = 1, lwd = 2,

   xlab=""  , ylab=TeX("mean (in percentage)"),    main=TeX('Rolling mean(on 252 days)  \\%'),   xaxt
="none",

   ylim=c(-0.5,0.5))     # do not diaply x and y-axes labels/ticks)

lines(x=seq.dates.x, y=data2plot[,2]*100, col="red", lwd=1, lty=1)

lines(x=seq.dates.x, y=data2plot[,3]*100, col="red", lwd=1, lty=1)

axis(1, at = date_seq, label = date_lab, las = 1, cex.axis=1.0); abline(0,0, lty = 1) # add zero line

# St. deviation

sd.plot    <- roll.mom[,2];

mu4        <- roll.mom[,5];

sd.plot.ub <- roll.mom[,2]+1.96*(1/(2*sd.plot)*sqrt(mu4-sd.plot^4))/sqrt(wind.length);

sd.plot.lb <- roll.mom[,2]-1.96*(1/(2*sd.plot)*sqrt(mu4-sd.plot^4))/sqrt(wind.length);

a <-cbind(sd.plot,sd.plot.lb,sd.plot.ub)

sd.plot.all <- cbind(index(a), data.frame(a)); names(sd.plot.all)[1] <- "date";



data2plot <- sd.plot.all[complete.cases(sd.plot.all),]

#postscript(file="../figures/SP500_stdev_rolling_1981_2018.eps",width=12,height=6,horizontal =
FALSE, onefile=FALSE)

plot(x = seq.dates.x, y = data2plot[,2]*100, type = 'l', col="blue"   , lty = 1, lwd = 2,

   xlab=""  , ylab=TeX("st.dev (in percentage)"),    main=TeX('Rolling st. dev.(on 252 days)  \\%'),
xaxt ="none",

   ylim=c(0,4))     # do not diaply x and y-axes labels/ticks)

lines(x=seq.dates.x, y=data2plot[,3]*100, col="red", lwd=1, lty=1)

lines(x=seq.dates.x, y=data2plot[,4]*100, col="red", lwd=1, lty=1)

axis(1, at = date_seq, label = date_lab, las = 1, cex.axis=1.0); abline(0,0, lty = 1) # add zero line
```

```
# Skewness
skew.plot    <- roll.mom[,3];
skew.plot.ub <- +1.96*sqrt(6)/sqrt(wind.length);
skew.plot.lb <- -1.96*sqrt(6)/sqrt(wind.length);
a <-cbind(skew.plot, skew.plot.lb, skew.plot.ub)
skew.plot.all <- cbind(index(a), data.frame(a)); names(skew.plot.all)[1] <- "date";


data2plot <- skew.plot.all[complete.cases(skew.plot.all),]
#postscript(file="../figures/SP500_skew_rolling_1981_2018.eps",width=12,height=6,horizontal = FALSE, onefile=FALSE)
plot(x = seq.dates.x, y = data2plot[,2], type = 'l', col="blue"   , lty = 1, lwd = 2,
    xlab=""  , ylab=TeX("skewness"),    main=TeX('Rolling Skewness (on 252 days) '),   xaxt ="none",
    ylim=c(-4,2))     # do not diaply x and y-axes labels/ticks)
lines(x=seq.dates.x, y=data2plot[,3], col="red", lwd=1, lty=1)
lines(x=seq.dates.x, y=data2plot[,4], col="red", lwd=1, lty=1)
axis(1, at = date_seq, label = date_lab, las = 1, cex.axis=1.0); abline(0,0, lty = 1) # add zero line
# kurtosis
kurt.plot    <- roll.mom[,4];
kurt.plot.ub <- 3+1.96*sqrt(24)/sqrt(wind.length);
kurt.plot.lb <- 3-1.96*sqrt(24)/sqrt(wind.length);
a <-cbind(kurt.plot, kurt.plot.lb, kurt.plot.ub)
kurt.plot.all <- cbind(index(a), data.frame(a)); names(kurt.plot.all)[1] <- "date";


data2plot <- kurt.plot.all[complete.cases(skew.plot.all),]
#postscript(file="../figures/SP500_kurt_rolling_1981_2018.eps",width=12,height=6,horizontal = FALSE, onefile=FALSE)
plot(x = seq.dates.x, y = data2plot[,2], type = 'l', col="blue"   , lty = 1, lwd = 2,
    xlab=""  , ylab=TeX("kurtosis"),    main=TeX('Rolling kurtosis (on 252 days)'),   xaxt ="none",
    ylim=c(2,11))     # do not diaply x and y-axes labels/ticks)
lines(x=seq.dates.x, y=data2plot[,3], col="red", lwd=1, lty=1)
```

```
lines(x=seq.dates.x, y=data2plot[,4], col="red", lwd=1, lty=1)

axis(1, at = date_seq, label = date_lab, las = 1, cex.axis=1.0); abline(0,0, lty = 1) # add zero line
```