Instituto Superior Técnico - UL

Time Series

Project Report

Author:
Pedro Chanca

Student Number:
87101

Prof. Manuel Scotto

Junho 2020

Contents

1. Introduction

The first set of data is regarding PM_{10} particles, where the PM stands for particulate matter and the term is used for a mixture of solid particles and liquid droplets found in the air, such as dust, dirt, soot or smoke. Also, PM_{10} diameters usually have 10 micrometers or smaller. Then, the main focus will be to forecast 5 days ahead using the hourly PM_{10} particles collected at Avenida da Liberdade monitoring station in Lisbon, from the periods of 01/01/2014 to 31/12/2018.

The results are illustrated with detail in chapter 2, which is divided in 4 different sections. In section 2.1, it will be performed an exploratory data analysis, taking advantage of plots, autocorrelation function (ACF) and partial autocorrelation function (PACF), stationary tests and eventual transformations in order to ensure stationarity before fitting the model. In section 2.2, it will be fitted the best SARIMA model by taking into consideration the analysis made on section 2.1, as well as, criteria selection models such as AIC, AICc and BIC values. Additionally, it will also be done the diagnostic testing of the residuals from the chosen fitted models. In section 2.3, it will be made a cross-validation between the selected models in order to back up the results obtained in section 2.2. In section 2.4, it will be done the forecasting with the best model for the 5 days ahead with the respective 95% confidence interval.

The second set of data is regarding the NASDAQ Composite Index, which is is the market capitalization-weighted index of over 2,500 common equities listed on the NASDAQ stock exchange. The types of securities in the index include American depository receipts, common stocks, real estate investment trusts (REITs) and tracking stocks, as well as limited partnership interests. Then, the primary objective of this project is to model a financial time series of log-returns regarding the daily close values of the NASDAQ Composite Index, between the periods of 02/01/2015 to 30/12/2019.

The results are illustrated with detail in chapter 3, which is divided in 2 different sections. In section 3.1, it will be performed an exploratory data analysis of the log-returns, where it will be adjusted an ARIMA model in order to analyze if there is any linear dependency in the data. In section 3.2, it will be adjusted several GARCH-type models, from which will be selected the one's that verify the necessary conditions and minimize the AIC and BIC values. Then, the residuals of each selected model will be analyzed and compared, and the model with the best results will be chosen to model the time series of the log-returns.

2. Results - PM_{10} particles in Avenida da Liberdade

2.1 Exploratory Data Analysis

Initially, the dataframe had a total number of 43824 1-hour PM_{10} particles of which 1391 had missing values, represented as *NA's*. In order to have a better visualization of the data, the PM_{10} particles were converted to 24-hour average levels, by doing the mean of the 24 values of PM_{10} particles available each day. After doing so, the dataframe had in total 1826 24-hour average levels of which 26 were *NA's*.

2.1.1 Handling Missing Values

There are two approaches when dealing with missing values, one being to delete the data where the values were missing and the other would be to to fill the missing values taking advantage of imputation methods, such as mean, median, linear interpolation, etc. In order to decide which one to use, let's first look into what types of missing data there are.

The most common types of missing data are: *Missing completely at random (MCAR)*, *Missing at random (MAR)* and *Missing not at random (MNAR)*. *MCAR* and *MAR* means that the possibility of losing PM_{10} particles data are not related to the fact of the hypothetical values, but instead due to external factors - such as a malfunction of the tools used in the monitoring station or even days when maintenance was needed - cause this data to go missing. Most of the missing data are observed between September and December as well as between February and April, though the cause of the missing data is not due to the values of PM_{10} particles, as shown in Figure 2.1. For example, the number of missing values between October and December shows high variability, while the average of PM_{10} particles is very similar.

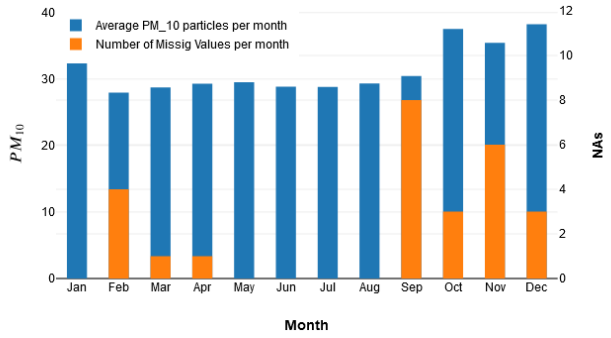


Figure 2.1: Graphical representation comparing the average number of PM_{10} particles per month with the number of missing values within the same month.

Taking the above into consideration and since it is a time series problem, the removal of the data is not often the best approach since there is a low percentage of values missing. Therefore, it would be a better to keep the data and use imputation techniques.

In order to conclude which imputation technique has the optimal performance, it was used the following 5 different approaches,

- Forward Fill;
- Backward Fill;
- Linear Interpolation;
- Cubic Interpolation;
- k-NN.

To start with, it was used the first 500 data points of which it was created two new dataframes, one with the first 400 values in order to train the model and the second with the last 100 values to test the model. To measure the interpolation performance, the test dataframe was filled with 2 random NA's out of the 100, in order to match

the scale of the overall dataframe (26 out of 1826). After using the imputed trained model in the test dataframe, it was measured the *mean squared error* (MSE) of the test dataframe with NA's against the test dataframe with the actual values. Finally, after repeating this approach 21 different times, since the placement of the missing values were random each time, the imputation technique chosen would be the one with the lowest average MSE. As example, in table A.1 is represented the obtained results for one of the runs of this approach.

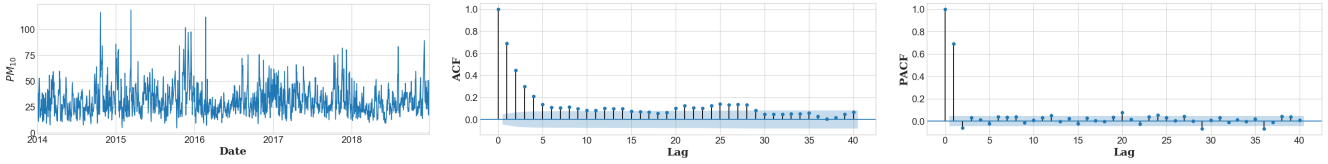
After comparing the average MSE of each imputation method, the *linear interpolation* was the one with the lowest average between the 5 imputation techniques. Therefore, this was the chosen imputation technique to fill the total amount of 26 missing values.

2.1.2 Stationary

After having all the values in place, next next step would be to analyse the times series regarding its stationarity. Before doing so, the dataframe was split in two. One would be the *train dataframe*, which included all the values from the beginning till the 26th of December of 2018, and would be used to train the models. The second, would be the *test dataframe*, which would only be used in cross-validation (section 2.4) to compare the accuracy of each model, and has the values from 27th till 31st of December of 2018.

Then, with the train dataframe, the following methods were used in order to test its stationary:

- Visual analysis of the time series plot, Figure 2.2a, it is apparent that there is no trend or seasonality. On the other side, one can notice the variance has high values and it is not constant along time. Therefore, a data transform is required in order to stabilize the variance. This will take place in section 2.2.1.
- Empiric analysis of the time series by taking advantage of the ACF and PACF plots, Figures 2.2b and 2.2c, respectively. The ACF has its values decreasing exponentially fast to zero and the PACF only has two significant values outside of the confidence interval. Thus, both show good signs of stationary behaviour.
- Function *pmdarima.arima.ndiffs* which estimates the simple differencing operator d by performing a test of stationarity for different levels of d, and in the end it selects the maximum value of d for which the observations resemble a realization of some stationary time series. The difference level obtained was d=0, which tells that the time series already has a good approximation to a stationary behaviour.
- Function *pmdarima.arima.nsdiffs* which estimates the seasonal differencing operator D by performing a test of stationarity for different levels of D, and in the end it selects the maximum value of D for which the observations resemble a realization of some stationary time series. The seasonal difference level obtained was D=0, once again, revelling that the time series already has a good approximation to a stationary behaviour.



(a) TS (b) ACF (c) PACF

Figure 2.2: Graphical representation of the time series (TS) regarding the daily values of PM_{10} particles, autocorrelation function (ACF) and partial autocorrelation function (PACF).

Transformation

Then, taking into consideration the previous statements, the *Box-Cox transformation* will be used in order to stabilize the variance and also to turn non-normal data resemble a normal distribution.

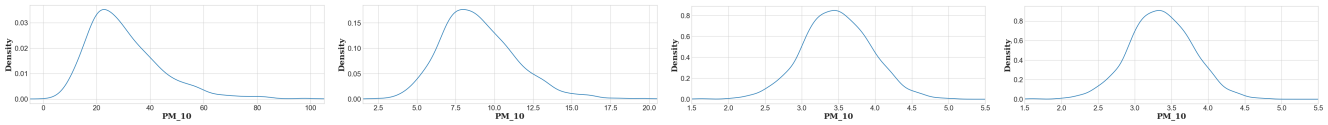
The non-normal behaviour of the original time series can be seen on both Figure 2.3a and Figure 2.4a, where it is represented the Kernel Density Estimation (KDE) and QQ plots, respectively. By looking at the KDE plot it is clear the existence of a asymmetric bell-shape, being the data slightly positive skewed and the QQ plot makes it even more clear the fact of time series not following a normal distribution. Another important variable to be aware of is the variance, which has value of 206, making it a even bigger argument for the use of Box-Cox transformation.

In order to decide which transformation would be well suited for this time series, it was used the function *scipy.stats.boxcox*, where it was obtained as output a λ=0.020, which implies a log transformation of the data, as it is close to 0. The square root transformation was also checked to see how it performed.

The variances resulting from the original data and the three data transformations are summarized in Table 2.1. Also, the KDE and QQ plots are represented in Figures 2.3 and 2.4, respectively.

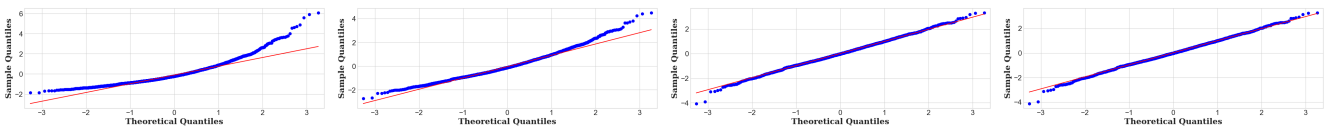
Data	Variance
Original	206.386
Sqrt Transform	5.784
Box Cox (λ=0.020)	0.216
Log Transform	0.188

Table 2.1: Variance of the original data, the square root transformation, Box-Cox transformation with λ=0.020 and log transformation.



(a) Original (b) Sqrt (c) Box-Cox (λ=0.020) (d) Log

Figure 2.3: Graphical representation of the Kernel Density Estimation (KDE) regarding the original data, the square root transformation, Box-Cox transformation with λ=0.020 and log transformation.



(a) Original (b) Sqrt (c) Box-Cox (λ=0.020) (d) Log

Figure 2.4: Graphical representation of the QQ plot regarding the original data, the square root transformation, Box-Cox transformation with λ=0.020 and log transformation.

It is clear that both Box-Cox transformation with λ=0.020 and λ=0 (log transformation) did a good job of stabilizing the variance, as well as, showing a good approximation to a normal distribution, as seen in the KDE and QQ plots. While the Box-Cox with λ=0.020 had the best performance in maximizing the log-likelihood function, the log transformation stabilizes the variance well and the trade-off of slightly less goodness of fit is worth since the log is easier to work with for forecasting. Thus, it was selected the log transformation.

Finally, taking into account the plot in Figure 2.5 looks a lot like what you'd expect from a *White Noise*, one can say the data is ready to be used for forecasting.

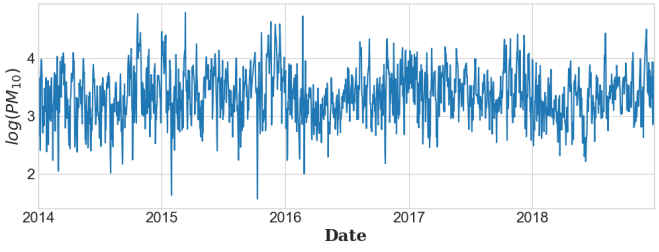


Figure 2.5: Graphical representation of the log transformation of the time series.

2.2 Model Fitting and Diagnostics

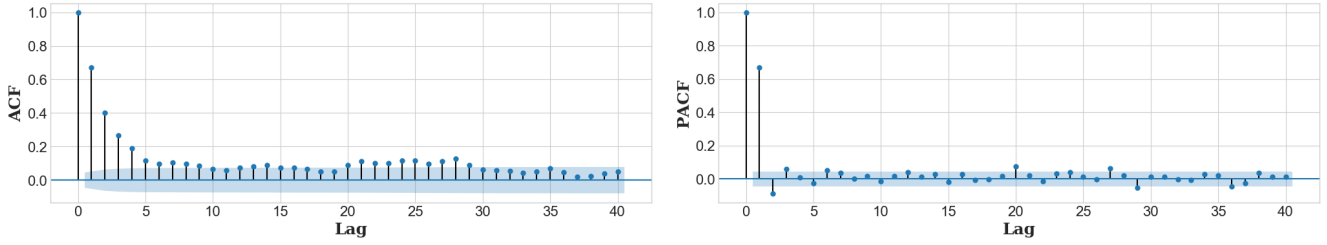
Initially, since the simple and seasonal difference coefficients are equal to 0, then the model as to be an *ARMA(p,q)* of order p and q with the following expression,

$$X_t = \psi_1 X_{t-1} + ... + \psi_p X_{t-p} + Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-p}, \tag{2.1}$$

where $Z_t \sim WN(0, \sigma_t^2)$ and $\psi_q \neq 0$, $\theta_q \neq 0$

2.2.1 Model Fitting

The ACF and PACF plots are a good starting point for model fitting. Looking at the shape and significant points on both plots tell a lot about what will be a good choice for the AR and MA components.



(a) ACF (b) PACF

Figure 2.6: Graphical representation of the autocorrelation function (ACF) and partial autocorrelation function (PACF).

Then, according to Figure 2.6, the PACF has a lag spike at lag 1 and 2, with all the following lags lying inside the confidence interval, only with few exceptions at lags 20 and 29, which were not considered. Overall, it is indicative of an AR(2) model. The ACF shows clear lag spikes up to just lag 3, which indicates a possible MA(3) model.

In order to have a second view of the previous analysis, the function *auto.arima* was used in order to find the best ARIMA(p,d,q) model. It was chosen order 10 for the parameters of the AR and MA models, p and q, respectively. The parameter regarding the simple, d, and the seasonal, D, difference coefficients was set to 0, following the results obtained in functions *pmdarima.arima.ndiffs* and *pmdarima.arima.nsdiffs*. Jointly, it were also used the parameters seasonal=False and stepwise=True. Then, according to this function the best model is an ARMA(1,3).

Thus it was selected the models AR(1), ARMA(1,1), ARMA(1,2), ARMA(1,3), AR(2) and ARMA(2,1) of which, their respective coefficients ψ_i, $i = 1, ..., p$ and θ_j, $j = 1, ..., q$, as well as the values of the Log Likelihood, Akaike information criterion (AIC), Corrected Akaike Information Criterion (AICc) and Bayesian Information Criterion (BIC) are represented in table 2.2.

Model	**Coefficients**		**Log-Likelihood**	**AIC**	**AICc**	**BIC**
	AR	MA				
AR(1)	ψ_1=0.6697	–	-519.322	1044.644	1044.657	1061.166
ARMA(1,1)	ψ_1=0.5882	θ_1=0.1485	-511.287	1030.574	1030.596	1052.602
ARMA(1,2)	ψ_1=0.6549	θ_1=0.0757 θ_2=-0.0658	-509.948	1029.897	1029.931	1057.432
ARMA(1,3)	ψ_1=0.7421	θ_1=-0.0087 θ_2=-0.1364 θ_3=-0.0606	-508.862	1029.723	1029.771	1062.766
AR(2)	ψ_1=0.7279 ψ_2=-0.0868	– –	-512.451	1032.903	1032.925	1054.932
ARMA(2,1)	ψ_1=0.4056 ψ_2=0.1261	θ_1=0.3291 –	-510.617	1031.234	1031.267	1058.769

Table 2.2: Coefficients, Log-Likelihood, AIC, AICc and BIC values for each ARMA(p,q) model.

All the models had their respective roots outside the unit circle, which is a sign of *causality* and *invertibility*. Then, in order to test the significance of the model coefficients, it was used the z-test, which had has a null hypothesis (H_0) the null state of the coefficient.

Between all 6 models, only the ARMA(1,1), AR(1) and AR(2) models had their respective p-values that allowed the rejection of H_0 regarding the terms of the 5% significance level. In the ARMA(1,2), ARMA(1,3) and AR(2) models, the H_0 was not rejected for the coefficients θ_1 and θ_2, θ_1 and θ_3, and ψ_2 and θ_1, respectively, i.e., it was not rejected the possibility of this coefficients being equal to zero, keeping all the others constant. Thus, one could say this could have been caused by *overfitting*.

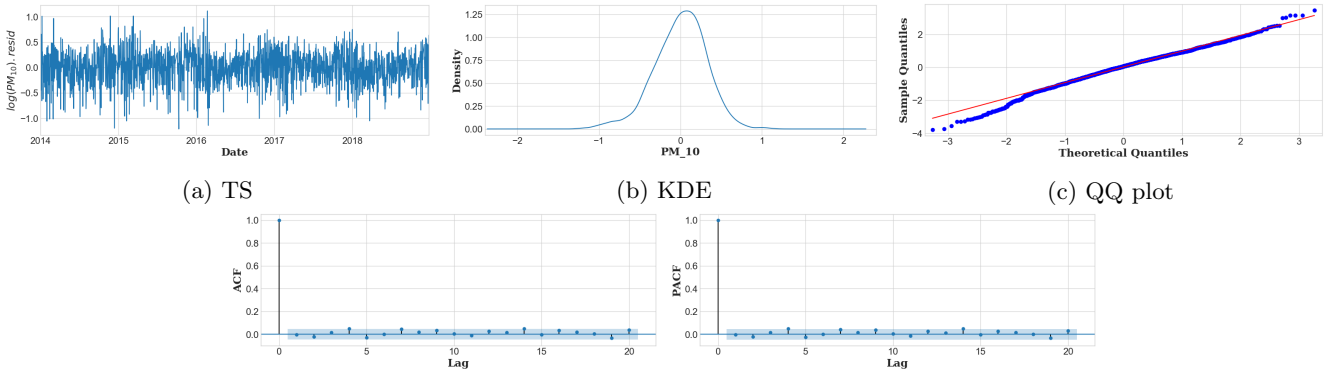
Then, since the models ARMA(1,1), AR(1) and AR(2) had the best results, one has to test its residuals in

order to compare their behaviour to a *White Noise*, i.e., if the residuals are in fact independent and identically distributed with normal distribution.

2.2.2 Diagnostics

The residual diagnostics of the ARMA(1,1), AR(1) and AR(2) are characterized in Figures 2.7, 2.8 and 3.2, respectively, where for each model is represented: the residual plot, KDE and QQ plot, as well as, the empirical ACF and PACF plots. Then, through the analysis of the information contained in these figures, one can conclude the following:

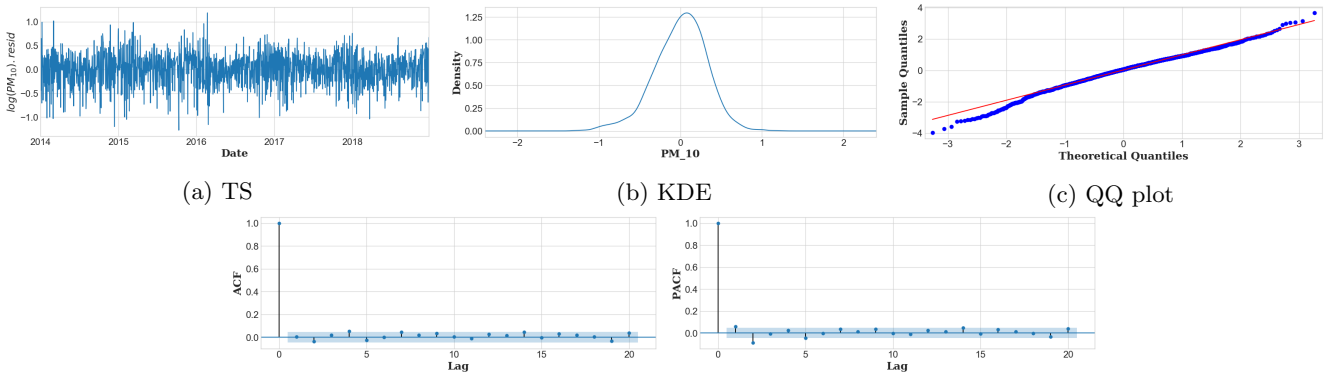
- There are no deviations of the residuals for each model regarding the hypothesis of the mean being close to 0 and the variance being constant. This is backed by the calculated values of the mean for each model, which range between 8.9×10^{-5} and 12.24×10^{-5}.
- The KDE and QQ plots of the residuals for each model have a similar behaviour to a normal distribution with a light tail component.
- The residuals of the AR(1) model have no correlation considering the empirical results of the ACF up to lag 20. But, regarding the PACF, it has one lag spike outside the confidence interval at lag 2, which might imply correlation; on the other side, the residuals of the ARMA(1,1) and AR(2) models have all their values inside the confidence interval up to lag 20 in both ACF and PACF.



(a) TS (b) KDE (c) QQ plot

(d) ACF (e) PACF

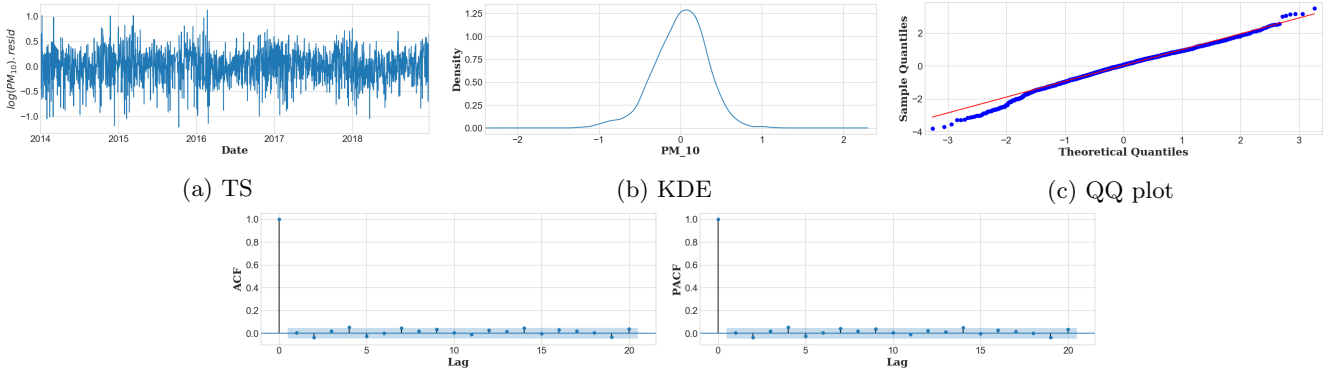
Figure 2.7: Graphical representation of the time series (TS), Kernel Density Estimation (KDE), QQ plot, autocorrelation function (ACF) and partial autocorrelation function (PACF) regarding the residuals of the ARMA(1,1) model.



(a) TS (b) KDE (c) QQ plot

(d) ACF (e) PACF

Figure 2.8: Graphical representation of the time series (TS), Kernel Density Estimation (KDE), QQ plot, autocorrelation function (ACF) and partial autocorrelation function (PACF) regarding the residuals of the AR(1) model.



(a) TS (b) KDE (c) QQ plot

(d) ACF (e) PACF

Figure 2.9: Graphical representation of the time series (TS), Kernel Density Estimation (KDE), QQ plot, autocorrelation function (ACF) and partial autocorrelation function (PACF) regarding the residuals of the AR(2) model.

Then, the residuals of all 3 models were used to apply the following tests: Shapiro-Wilk, Jarque-Bera and Ljung-Box. The Statistical Test Value (SVT) and the p-values associated with each model are represented in Table 2.3. Then, going through the results, one can conclude the following:

- In order to test the normality of the residuals, it was used the Shapiro-Wilk test by using the function *stats.shapiro*, which has as null hypothesis the residuals being drawn from normal distribution. The obtained p-value for all 3 models rejects the null hypothesis and therefore there is evidence that the residuals are not normally distributed for all the values of significance.
- In order to test if the residuals have the skewness and kurtosis matching a normal distribution, it was used the Jarque-Bera test. Using the function *sm.stats.jarque_bera*, the null hypothesis corresponds to the joint hypothesis of the skewness being zero and the excess kurtosis (estimated kurtosis - 3) being zero. Since the p-value obtained for all 3 models is lower than the value of significance of 5%, the null hypothesis is then rejected.
- In order to test if the residuals are independent, it was used the Ljung-Box test. Using the function *sm.stats.acorr_ljungbox* with *lags*=10 and *model_df*=2 (degrees of freedom=p+q, where p is the AR order and q is the MA order), the null hypothesis corresponds to the residuals being independently distributed up to lag 10. For the ARMA(1,1), the p-value obtained of 0.0841 enables one not to reject the null hypothesis at the level of significance of 5%, meaning there is no information in the residuals, which may lead to better forecast results. For the AR(1) and AR(2) their respective p-value of 0.0006 and 0.0351 leads to the rejection of the null hypothesis for the level of significance of 5%, which indicates the dependence between the residuals.

Test	**ARMA(1,1)**		**AR(1)**		**AR(2)**	
	STV	**p-value**	**STV**	**p-value**	**STV**	**p-value**
Shapiro-Wilk	0.9895	3.3535×10^{-10}	0.9890	1.6335×10^{-10}	0.9895	3.4302×10^{-10}
Jarque-Bera	73.2962	1.2132×10^{-16}	78.5514	8.7658×10^{-18}	73.2548	1.2385×10^{-16}
Ljung-Box	13.9104	0.0841	29.2141	0.0006	16.5582	0.0351

Table 2.3: Results of the tests of normality (Shapiro-Wilk and Jarque-Bera) and independence (Ljung-Box) of the residuals of the ARMA(1,1), AR(1) and AR(2) models.

Finally, one can conclude the ARMA(1,1) achieved the best results since it's the only model that had its residuals independent from one another up to lag 10, although the tests of normality rejected the hypothesis of it having a normal distribution. Then, taking this into consideration, the overall results were not satisfactory which might influence the quality of the forecast, but there were no other alternatives that could solve this problem.

2.3 Cross-Validation

In order to back up the results obtained in the previous section, it was done a cross-validation between the predicted values of the next 5 days and the test dataframe which contains the observed values, in order to select

which model has the best accuracy in estimating the values. In table 2.4 are represented the results for the model ARMA(1,1), AR(1) and AR(2), which will allow one to evaluate the model with the best forecast results. Then, going through the results, the following considerations can be taken:

- The coefficient of determination, R^2, measures how well the regression predictions approximate to the observe values. Then, in this particular case, the ARMA(1,1) achieved the highest result between the 3 models.
- Regarding the Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Squared Log Error(MSLE) and Square Root Mean Squared Error (RMSE), the ARMA(1,1) model had the lowest output values on every single one of these error measure techniques.

Model	R^2	**MAE**	**MSE**	**MSLE**	**RMSE**
ARMA(1,1)	0.3028	0.3230	0.1582	0.0084	0.3977
AR(1)	0.2146	0.3478	0.1782	0.0093	0.4221
AR(2)	0.2937	0.3265	0.1602	0.0085	0.4003

Table 2.4: Results of the R^2, MAE, MSE, MSLE and RMSE for the ARMA(1,1), AR(1) and AR(2) models.

Finally, since the ARMA(1,1) achieved the best results in the diagnostics and as a follow up, it also had the best predicting performance between all 3 models, then it was selected to perform the forecast of the next 5 days ahead, whose values are unknown.

2.4 Forecast

The predicted values and the 95% confidence interval were obtained for the period between 1st and 5th of January of 2019, with the respective results represented on Table 2.5, as well as the its plot in Figure 2.10.

The predicted values as time goes by have a tendency to converge to the mean of the time series, as to be expected. Also, the confidence interval increases with longer term predictions which is expected since the uncertainty of a prediction increases as the time period goes further away from the last observed value.

	Day 1	**Day 2**	**Day 3**	**Day 4**	**Day 5**
Predicted Value	3.4489	3.4111	3.3891	3.3761	3.3686
95% Confidence Interval	[2.8207, 4.0771]	[2.6304, 4.1919]	[2.5626 , 4.2155]	[2.5346, 4.2176]	[2.5220, 4.2152]

Table 2.5: Predicted Values and 95% Confidence Interval between the 1st of January and the 5th of January of 2019.

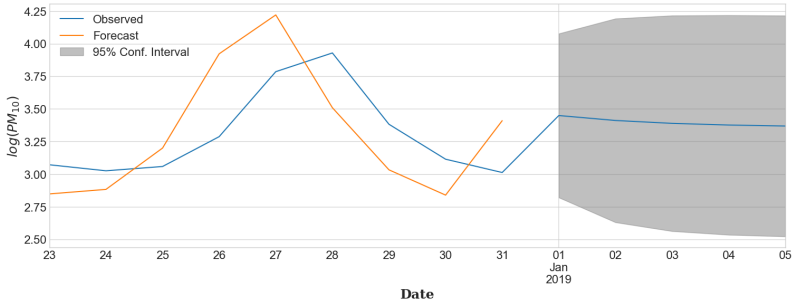


Figure 2.10: Graphic representation of the Predicted Values (blue) and the respective 95% Confidence Interval between the 1st of January and the 5th of January of 2019. The orange line are the observed (real) values available till the 31st of December of 2018.

3. Results - NASDAQ Composite Index

3.1 Exploratory Data Analysis

Initially, it was created a time series of the log-returns, $X_t = log(P_t) - log(P_{t-1})$, associated with the column regarding the daily close values, P_t, of the NASDAQ composite index, with *frequency=252* corresponding to the number of trading days per year, Figure 3.1.

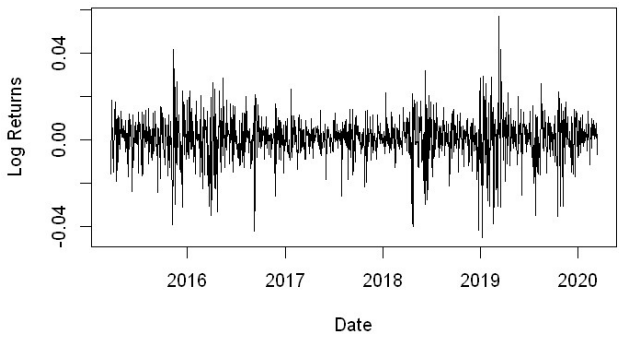


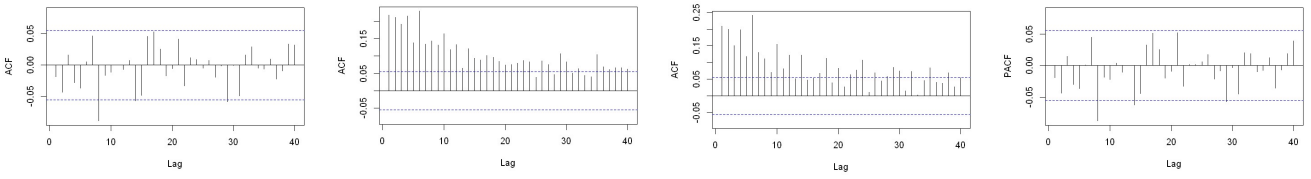
Figure 3.1: Graphical representation of the log-returns of the daily close values of the NASDAQ Composite Index.

Then, it was adjusted an *ARIMA(p,q,d)* model to the time series of log-returns in order to remove linear dependency within the data. So, taking advantage of the R function *auto.arima*, the result was an ARIMA(0,0,0). Therefore, it will not be used an ARIMA model for this data.

Next, it was evaluated some of the characteristics regarding models of conditional variance, such as:

- The sample mean is close to zero (mean=$5,079 \times 10^{-4}$) and the variance is of the order 10^{-4} or smaller (variance=$1,046 \times 10^{-4}$);
- The ACF, Figure 3.2a, is negligible at all lags, with an exception on the first lag. On the other side, the ACF of the absolute or squared values, Figures 3.2b and 3.2c, respectively, are different from zero for a large number of lags and stay almost constant and positive for large lags.

Finally, it was tested the independence of the log-returns by using the Ljung-Box test, where it was obtained a p-value of $0,23$ which then indicates that one can not reject the null hypothesis (independence), which might indicate serial uncorrelation within the data. It was also used the Ljung-Box test on the standardized squared residuals of the log-returns, where it was obtained a p-value of 2.2×10^{-16} and therefore the null hypothesis (independence) is rejected, meaning that the autocorrelation of the standardized squared residuals of the log-returns are different than zero, which is indicative of the existence of an ARCH behaviour.



(a) ACF (b) AACF (c) SACF (d) PACF

Figure 3.2: Graphical representation of the autocorrelation function (ACF), absolute autocorrelation function (AACF), squared autocorrelation function (SACF) and partial autocorrelation function (PACF) of the log-returns.

3.2 Model Fitting and Diagnostics

The next step is to find within the family of GARCH-type models, the one that adjusts better to the time series of the log-returns. To do so, it was used the R function *garchFit* to model 40 different viable combinations of the parameters p and q ($p, q \in [0, 4]$), for both GARCH and APARCH models. Also, it was used the parameter *cond.dist = 'std'* in order to accommodate heavy tails. From the 40 combinations, 16 were not possible to analyse since its computation generated NA's. For the remaining 24, as selection criteria, it was used the AIC and BIC

values, the p-values of the Ljung-Box and LM Arch tests, applied to the residuals and to the squared residuals and the p-values of the Jarque-Bera and Shapiro-Wilk tests, which test the normality of the residuals. All off this results combined will facilitate the selection of GARCH-type model.

Regarding the LM Arch test, its main objective is to check if the residuals still exhibit an ARCH behaviour, i.e., if there stills exist volatility within the residuals' variance. It has as null hypothesis the existence of homoscedasticity, i.e., the residuals' variance being constant and therefore not exhibiting any ARCH behaviour. Therefore, its values have to be high in order to assure the residuals are white noise.

In table 3.1 are represented the coefficients from 3 different models, as well as their respective AIC and BIC values, the p-values obtained in the Ljung-Box and LM Arch tests, which test the independence of the residuals, and the p-values from the Jarque-Bera and Shapiro-Wilk tests, which test the normality of the residuals.

Model	a_0	**Coefficients**		**AIC**	**BIC**	**Ljung-Box(15)**		LM Arch	Jarque-Bera	Shapiro-Wilk
		$a's$	$b's$			R	R^2			
ARCH(4)	$0,3834 \times 10^{-4}$	α_1=0,2218 α_2=0,1783 α_3=0,1618 α_4=0,1996	-	-6,5845	-6,5559	0,2238	0,1598	0,1320	0	0
GARCH(1,1)	$0,3696 \times 10^{-5}$	α_1=0,1500	β_1=0,8265	-6,6151	-6,5946	0,3202	0,3723	0,4172	0	0
APARCH(1,1)	$0,3405 \times 10^{-2}$	α_1=0,1135	β_1=0,8808	-6,5991	-6,5704	0,9997	1	0,6556	0	0

Table 3.1: Coefficients, AIC, BIC values for the selected GARCH-type models as well as the respective p-values obtained in the tests of independence and normality. The APARCH model has the parameter $\gamma = 1$.

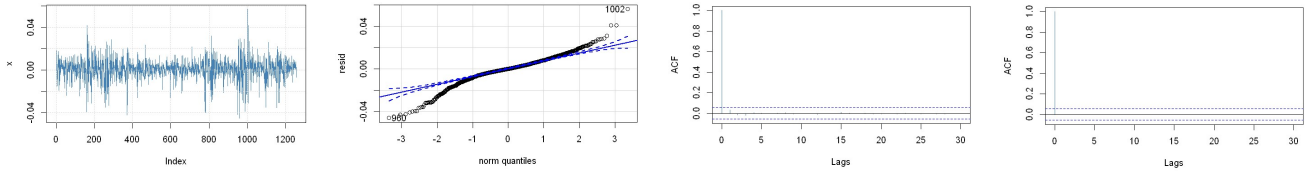
The criteria that led to choose this 3 different models was the following:

- The ARCH(4) was the only ARCH model that had its p-values on the Ljung-Box and LM Arch tests high enough not to reject the null hypothesis (independence of the residuals).
- The GARCH(1,1) was the only GARCH model where its coefficients were statistically significant, i.e., it was the only model that rejected the null hypothesis of the t-test. Regarding the AIC and BIC values, their values were constant for all orders of complexity and therefore were not taking into account. The same occurred with the p-values of the Ljung-Box and LM Arch tests. When it comes to the normality of the residuals, the p-values in the Jarque-Bera and Shapiro-Wilk were 0 throughout all orders of complexity, indicating the non-normality of the residuals.
- The APARCH(1,1) between all the GARCH-type models, achieved the best results on both Ljung-Box and LM Arch tests of independence and its AIC and BIC values are close to the one's obtained with the GARCH(1,1). The results were expected since the APARCH models with $\gamma > 0$ are built in order for negative shocks to have stronger impact on volatility than positive shocks, and since the data is regarding financial time series, quite often bad news have stronger impact on volatility than good news.

Regarding the normality of the residuals, it was not achieved in none of the 3 models, which is reflected on the p-values of the Jarque-Bera and Shapiro-Wilk tests. Therefore, even though the residuals of each model are uncorrelated and do not have ARCH behaviour, they can not be considered *white noise*.

From the 3 selected models, the APARCH(1,1) achieved the best overall results and therefore it was chosen to adjust the time series of the log-returns.

Additionally, in Figure 3.3b is represented the QQ-plot of the residuals where is visible their non-normality and by having a closer look at Figures 3.3c and 3.3d it is also clear the non-correlation and independence of the residuals, which also confirms the results obtained in the Ljung-Box test for both R and R^2.



(a) Residuals Time Series (b) QQ plot Residuals (c) ACF Residuals (d) Squared ACF Residuals

Figure 3.3: Residuals Plots of the APARCH(1,1) model

4. Conclusion

This project was divided in two distinct parts. In the first one, it was adjusted a linear model do the PM_{10} particles collected in Avenida da Liberdade and in the second one, it was adjusted a non-linear model to the data regarding the NASDAQ Composite Index. Then, all the initial objectives for this project were achieved.

Regarding the first part of the project, it was initially handled the missing values by using the linear interpolation which is an imputation technique. Although this achieved good results, the optimal choice would have been the k-NN, since the linear interpolation is not suited when there is more than 5 consecutive days of missing values, which ended up being the case between 15/11/2018 and 06/12/2018. Then, in order to have a stationary behaviour, it was done a log-transformation of the time series leading to a decrease of its variance and mean values close to zero. Next, it was done a model fitting as well as the diagnostic of the residuals of each model. The results were not satisfactory since non of the selected models had their residuals close to a white noise. Although the residuals had no serial correlation, they didn't have a normal behaviour. Finally, after doing a cross-validation to back up the results obtained previously, using the selected ARMA(1,1) model, it was done a forecast into the future up to 5 time periods ahead, with their respective 95% confidence interval.

In the second part of the project, it was also done an initial exploratory data analysis in order to identify characteristic that could indicate which model would be better suited to the data. Since the data didn't show any linear dependency, the next step would be to fit GARCH-type models to the time series of the log-returns as well as the diagnostic of the residuals of the selected models. As in the first part, the residuals had no serial correlation, but didn't have the desired normal behaviour and therefore couldn't be classified as white noise. Overall, the selected model was the APARCH(1,1), which was to be expected from a financial time series.

Bibliography

[1] Prof. Manuel G. Scotto. *Time Series - Lecture Notes.*

[2] António Pacheco Pires. *Notas de Séries Temporais, 2000/01.*

[3] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications With R Examples.*

[4] Selva Prabhakaran. *Time Series Analysis in Python – A Comprehensive Guide with Examples.* URL: https://www.machinelearningplus.com/time-series/time-series-analysis-python.

[5] Paz Moral and Pilar González. *Univariate Time Series Modelling.* URL: http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/tutorials/xegbohtmlframe115.html.

[6] Statsmodels. *Time Series Analysis - Examples.* URL: https://www.statsmodels.org/stable/examples/index.html.

[7] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice.* URL: https://otexts.com/fpp2.

[8] Aileen Nielsen. *Time Series analysis with Python.* URL: https://github.com/Aileennielsen/timeseriesanalysisswi

[9] Jason Brownlee. *How to Use Power Transforms for Time Series Forecast Data with Python.* URL: https://machinelearningmastery.com/power-transform-time-series-forecast-data-python.

[10] Lajos Horváth István Berkes and Piotr Kokoszka. *GARCH processes: structure and estimation.* URL: https://fenix.tecnico.ulisboa.pt/downloadFile/845043405520117/2003%20GARCH.pdf.

A. Appendices

	Forward Fill	Backward Fill	Linear Interpolation	Cubic Interpolation	k-NN (k=5)
1	0.09	1.28	0.29	0.27	0.29
2	0.07	0.42	0.18	0.20	0.72
3	0.51	0.03	0.08	0.14	0.13
4	0.06	1.59	0.41	0.06	0.98
5	0.06	0.24	0.02	0.01	0.05
6	0.66	0.09	0.07	0.10	0.09
7	0.25	0.00	0.06	0.00	0.91
8	0.12	0.15	0.06	0.08	0.27
9	0.26	0.02	0.04	0.01	0.35
10	0.48	0.48	0.02	0.07	0.01
11	0.65	0.31	0.21	0.09	0.76
12	0.67	0.08	0.27	0.29	0.41
13	0.54	0.09	0.16	0.17	0.18
14	0.19	0.04	0.04	0.05	0.03
15	0.58	0.13	0.03	0.30	0.34
16	1.00	0.11	0.40	0.20	0.59
17	0.52	0.03	0.17	0.21	0.24
18	1.01	0.50	0.36	0.22	1.21
19	0.03	0.24	0.10	0.09	0.07
20	0.85	1.50	1.07	0.56	2.00
21	0.14	0.10	0.06	1.42	1.41
Average	0.394	0.511	0.208	0.215	0.526

Table A.1: MSE results for each imputation technique obtained from 21 different placements of NAs in the test dataframe.