
# Seismic Time Series Analysis
# Statistics for Spatio-Temporal Data
## *– project draft –*

Marco Ciotola, 848222

June 2, 2020

# Contents

# 1 Introduction

### Motivation and objective

Thanks to the recent coronavirus outbreak worldwide, we currently have data about many different phoenomena from periods with standard human activity, and from a period with reduced human activity.

Inspired by an article published during the initial phases of the Belgium-wide lockdown (1), we want to assess whether the lockdown period can be correlated with a visible change in seismic activity. It is important to notice that some industrial events, such as mine explosions, are already mapped by seismologic institutions since they interfere with seismic surveys (2). Such events are sometimes comparable with natural seismic events, depending on the distance from the seismograph.

### Data source and description

The Royal Observatory of Belgium (ROB) gathers and publishes (3, 4) data about vertical ground displacement from 9 stations through the country, linking them to seismic events whenever possible (5). They also offer a data visualization tool for the same data (6).

The vertical ground displacement is represented in by the minimum and maximum displacement of the ground in the vertical axis, with *nm* as unit of measure. This can easily be transformed in total ground displacement for that second by their difference. For each day, a pair per second is available (86400 per day, around 30 million per year).

ROB collects data from 9 stations. Of those, the following 6 have interesting positions, related to possible correlations with human activity:

- **Uccle**, **Sart-Tilman**, and **Ében-Émael** are near a city (Bruxelles, Liege, and Maastricht).
- **Ostenda** This station is near the city of Ostenda, that borders with the sea.
- **Membach** and **Dourbes** are near a natural park.

Data collected from near a city and from near a natural park could be selected, in order to compare possible seasonality results and differences from before and after the lockdown. Considered these factors, we selected the **Uccle** and **Membach** stations.

It is important to notice that the *day*ly data in each file is considered in UTC time, an optimal choice to publish data in a machine-readable way, even if this could cause some issues while interpreting the results.

## 1.1 Data cleanup and transformation

### Data gathering and cleanup

The data is provided with a CSV per day, with only two columns related to the min-max pair (4). This causes two problems:

1. Before merging different days, we need to add a column with the complete datetime for each row
2. Days with some `NA` values cannot skip rows, so we will find placeholder values that indicate them $(\min, \max) \in \{(0,0), (-1,1)\}$

Data from each station have different domains, as distinct equipment can register a specific amplitude of the seismic movements. In fact, we found some values outside the domains, transformed in `NA`.

In particular, we find that out of $\approx$ 50 million seconds, from 27-10-2018 to 28-05-2020, only 0.67% (Uccle) and 0.37% (Membach) have invalid values. We will see later how they are distributed for each station.

#### Data aggregation and initial transformation

Since we are not interested in second-precise analysis, and it would be computationally expensive to deal with the whole dataset, we are going to aggregate data by hour. The aggregation will get us around 14 thousand points, that can still result in interpretable models.

While performing this operation, we should consider the correct timezone: any seasonal analysis dealing with UTC data would be out of phase between the half-year when Belgium follows the UTC+01 timezone, and the half-year when it follows the UTC+02 timezone. With the help of the `lubridate` R package we converted to the actual timezones, and then we kept the actual objects without being timezone explicit, because all the methods we are going to use are smart enough to work on the UTC conversion of datetimes. This transformation made it possible to have, for example, the time 6:00 in the timeseries to represent the local time in Belgium (*Europe/Bruxelles*), without being influenced by the actual timezone on that specific day.

The aggregation was performed considering the mean max-min difference, for the following reasons:

- When an hour contains some missing seconds, these do not heavily affect the aggregation result, in contrast with summing the differences
- Seismic events, such as quakes and mine explosions, normally last some minutes at most, and will affect the result only slightly, rather than keeping the hourly maximum
- Normally, maximum values in defined period of times do not follow the Normal distribution, while some models that we are going to use assume the data follow a Normal distribution

## 2 Analysis and decomposition

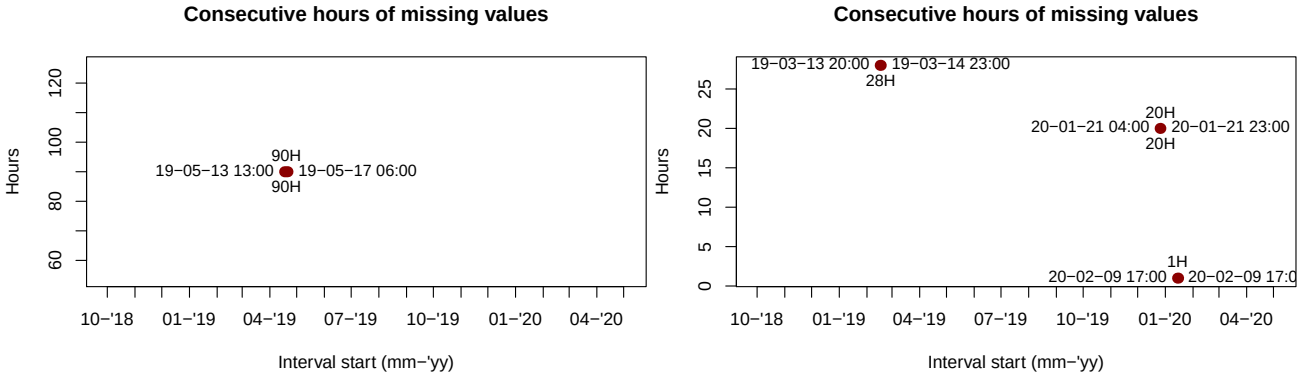
#### Missing values analysis

After aggregating by hour, the percentages of missing values remain quite stable. As can be seen from Figures 1a and 1b, both stations missing values are grouped in few consecutive periods. To deal with them when R methods need complete timeseries, we will define different completion strategies depending on the seasonality we will be studying, always based on the R function `na.aggregate`.

In particular, after a visual analysis after decomposition, we can suppose the 90 hours gap from the Uccle station is related to some changes to the seismic equipment, that resulted in slightly more precise measurements.

#### Seasonality assumptions

The already cited article (1) suggests the existence of a weekly seasonality. Looking at the data it seems confirmed, in addition to a daily seasonality.




(a) Uccle station

(b) Membach station

Figure 1: Missing values representation

Since our main focus is on the seasonalities themselves, in order to prevent assumptions based on the objective, so that searching for correlation with human activity becomes imposing human-based seasonality periods, we will use an analytical method to extract seasonalities that the data itself suggests to us. Still, we recognize that the already performed transformations impose an human-based time alteration to the data itself (Section 1.1).

The method we are going to apply is based on the Fourier Transform, and generates a *periodgram* highlighting the most interesting frequencies that might correspond to a seasonality component. Applying the FT for any frequency, we transform our 1-dimensional data into a 2-dimensional data mapped around a circle. Then, we can sum all the resulting vectors (that are our new data points) to obtain a vector for each frequency, whose absolute value represents the power level of the same frequency. An high power level represents a significant periodicity on that frequency in our original data.

A common operation to increase the difference between significant and non-significant periodicities is to apply a MA smoothing filter with a smaller window than the interesting ones. We will use such a filter with $f = 2$.

On this note we must specify that, when a significant periodicity is found, it influences smaller periodicities given by $f * \{2, 3, \dots\}$ the significant frequency. Also, if any periodicity is found that is longer than the original dataset, it must be considered as a *coincidence*, rather than a surprising serendipity.

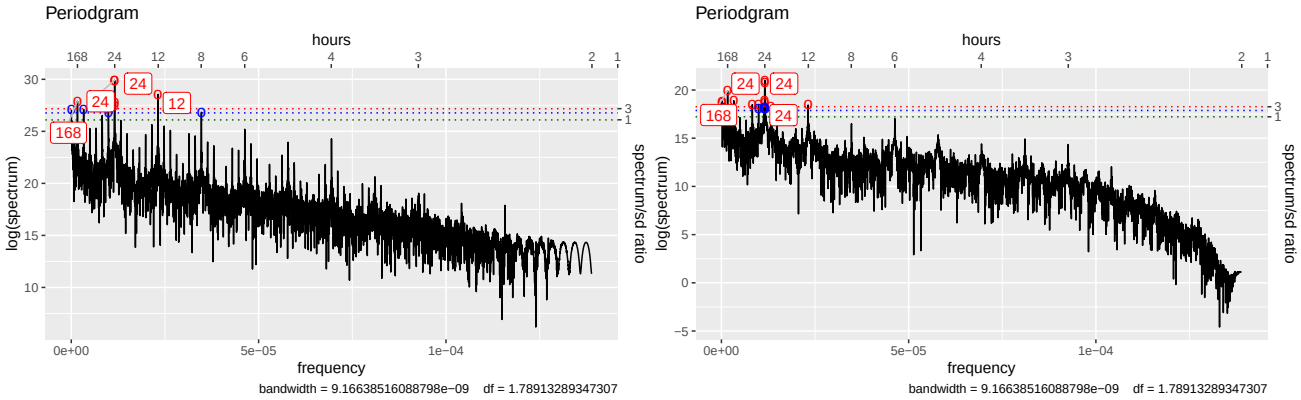
Since we will be applying this method on the dataset indexed by a datetime object, which uses the second as the unit, the transformation of frequency $f$ into hourly periodicity $p$ is obtained by $p = (1/f)/3600$. We can analyse the resulting periodgrams for Uccle and Membach stations in Figures 2a and 2b.

Both highlight multiple 24-hour periodicities. This is given by approximation factors, since we round the divisions result to the nearest natural number. It also highlights that using a single 24-hour seasonality could leave some information on the remaining components. Still, in both stations it is the most significant periodicity.

Both stations highlight a significant 168-hour (1-week) periodicity, that also confirms our initial assumptions.

Both stations show a possibly significant 12-hour seasonality, but since it corresponds to double the frequency of the 24-hour seasonality, we can safely ignore it.

In particular from the Uccle station, we can see a great number of spikes, that will probably undermine our models.




(a) Uccle station (b) Membach station

Figure 2: Periodgram results

## Dataset division

As best practices suggest, we are going to divide the dataset into training and testing datasets with a ratio around 85%/15% – 75%/25%. Since one of our objectives is to assess any significant difference from before and after the Belgium lockdown (on March, $14^{th}$ the *soft* lockdown, while on March, $18^{th}$ the complete lockdown), we selected data from the start of our timeseries until 20-01-2020 (64 weeks ≈ 82%) as training, while the test follows and ends on 27-04-2020 (14 weeks ≈ 18%).

This initial division demostrated to be computationally heavy while fitting some models with weekly seasonality, and we noticed a possible yearly seasonality on Uccle data (Figure 3). Even if it is not demonstrable given the short period our data comes from, applying a 24-hour MA smoothing filter on the whole dataset (546 days) and then producing the periodgram as previously done, it suggests a 8748-hours significant periodicity (364.5 days).

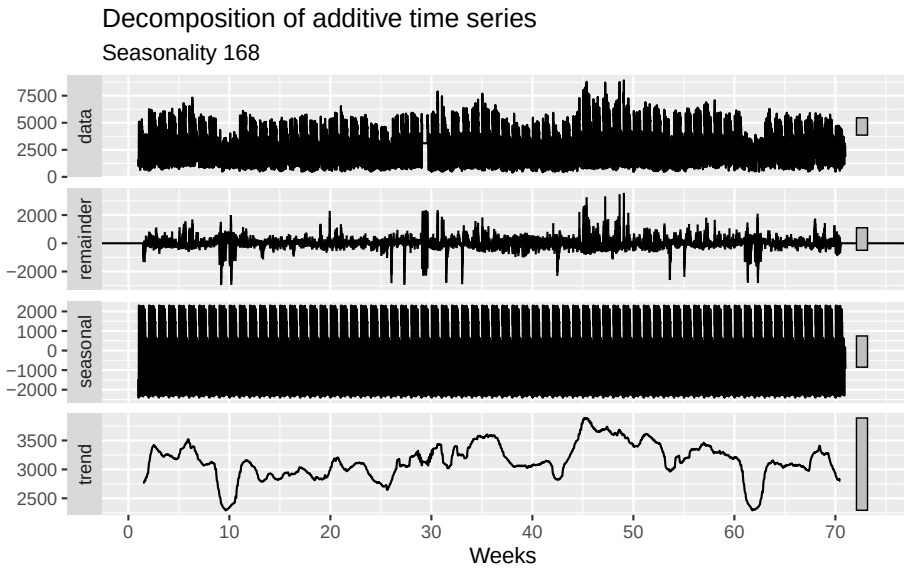



Figure 3: Whole Uccle data decomposition
Weeks 10 and 62 include January, $1^{st}$ 2019/2020

Given those issues, we choose the training/testing sets to hold 52/18 weeks: this is the greater train-to-test ratio we deem usable (≈ 75%/25%), while maintaining computational feasibility, and we hope it will render the possible yearly seasonality less intrusive.

**Decomposition**

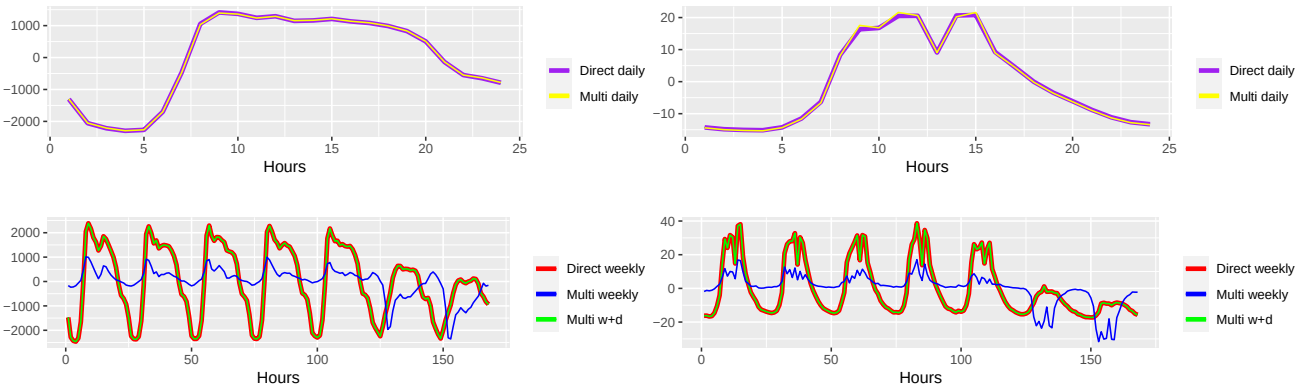
In order to decompose our dataset into trend and seasonality(-ties) components, we followed these steps:

1. Apply a smoothing filter to extract the trend component
2. Remove the trend from the dataset and extract the seasonal figure using the local mean method
3. Remove the trend and the seasonal components to obtain the residuals

While doing so, we tested different smoothing filters: simple filter (p=seasonality), Spencer's 15-point filter and MA smoothing filter (f=seasonality). Since they do not produce particularly different seasonalities, we choose to keep the results obtained by applying the MA smoothing filter, for its significance in terms of seasonally adjusted data.

Since the periodgrams suggested the presence of two significative periodicities, we also tested the extraction of both corresponding seasonalities:

1. Apply the MA smoothing filter (f=24) on the dataset to extract the trend component
2. Remove the trend from the dataset and extract the 24-hour seasonal figure using the local mean method
3. Remove the 24-hour seasonal components to obtain a deseasoned dataset
4. Apply the MA smoothing filter (f=168) on the deseasoned dataset to extract the updated trend component
5. Remove the trend from the already deseasoned dataset and extract the 168-hour seasonal figure using the local mean method
6. Repeat steps 3-5 steps alternating the 168/24 seasonalities until convergence (2 to 3 cycles should be enough)
7. Removing the last fitted trend and the two seasonalities we obtain the final residuals




(a) Uccle station (b) Membach station

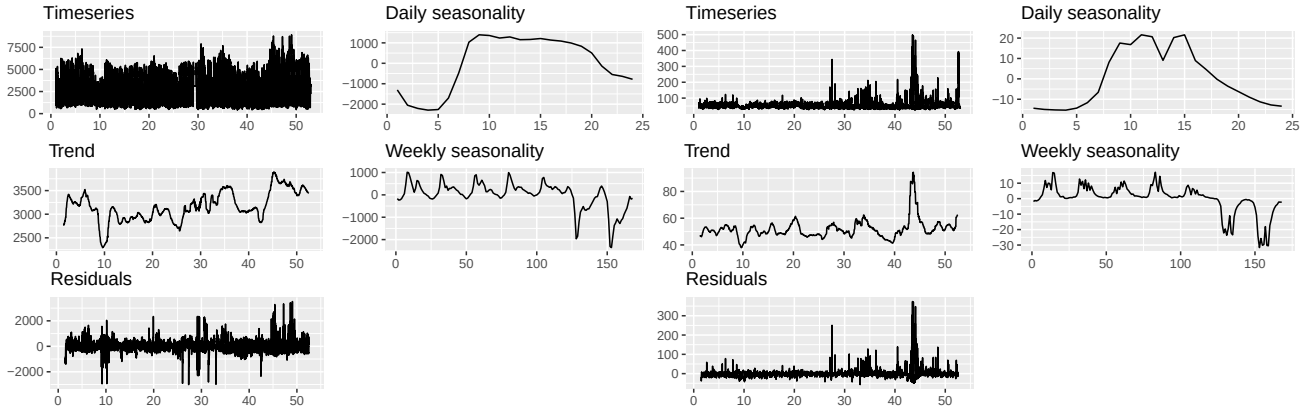
Figure 4: Seasonality figures
*Direct* refers to single seasonality extraction
*Multi* refers to both seasonalities extracted at the same time

In Figure 4 we see both approaches applied to both stations. Those plots show an interesting property of our seasonalities:

- Both ways of extraction result in the same daily seasonality extracted
- The sum of the daily and weekly seasonalities extracted together is the same as the weekly seasonality extracted directly

Given in particular the second property, we will only use the 168-hours seasonality while fitting our models, for both the stations.

In Figure 5 we see the decomposition results. While data from Uccle station have a clear significativity of both trend and seasonalities, data from Membach could seem not significant. Still, if we notice that the main timeseries lies between values of 0 and 100, we can find out that around half of the value will be composed by the trend, a 30% by the joined seasonalities component, and the remaining value by the residuals. We will need to better handle the spikes from around half our dataset.




(a) Uccle station (b) Membach station

Figure 5: Timeseries decomposition
Only the second method results are shown for clarity

# 3 Parameter estimation

Having two different datasets, we need to fit two separate models that represents those stations. During this estimation, our aim is to obtain a model that generalises our training dataset in such a way that forecasting becomes effective, also assessing errors analytically. Obtaining such a reliable model will be the first step to verify any significant difference forecasting over the lockdown period.

The first model we are going to estimate is the ARIMA model, including the seasonality components. To identify the order of the different components described by ARIMA, we will analyse the autocorrelation and partial autocorrelation of our datasets. As a reminder:

- non-seasonal AR process of order $p$ is described by an exponentially decaying ACF, and by a cut-off of significant PACF values at lag $p$
- non-seasonal MA process of order $q$ is described by an exponentially decaying PACF, and by a cut-off of significant ACF values at lag $q$
- seasonal AR process of order $P$ is described by an exponentially decay at seasonal lags on ACF values, and by a cut-off of significant PACF values at each seasonal lag until the $P^{th}$

- seasonal MA process of order $Q$ is described by an exponentially decay at seasonal lags on PACF values, and by a cut-off of significant ACF values at each seasonal lag until the $Q^{th}$
- given how significance is assessed for the ACF/PACF spikes, one over 20 spikes (5%) could be *coincidentally significant* or *coincidentally* ***non****-significant*

## 3.1 Uccle station

### ARIMA

Our training set, despite the precautions taken during the initial data cleanup and aggregation, doesn't satisfy the Normality assumption, even after applying the *log* or *Box-Cox* transformations. Thus, we will estimate the models using the original scale.

Since our dataset shows a relevant trend component, resulting in non-constant mean, we consider the usage of the *diff* operator to smooth the mean. After applying the *diff* operator with lag 168, in conjunction with the simple *diff* or not, we come up with two possible models by checking the ACF and PACF plots:

- from using both *diff*s, the ACF and PACF on the long run hint to $\text{ARIMA}(0,1,0)(0,1,1)_{168}$
- from using just the lagged *diff*, the PACF on the first lags and the ACF/PACF combination on the long run hint to $\text{ARIMA}(2,0,0)(0,1,1)_{168}$

We also employ the *auto.arima* stepwise methodology, locking the *diff* values to 0 or 1, to get some hints on the possible models.

Following both the analytical ways and the *auto.arima* hints, and refining the models through further inspection of ACF/PACF plots of the residuals, we end up with three characteristic models. We selected them through the Akaike Information Criterion (AIC), being the only three out of the fitted models with AIC $< 120.000$: $\text{ARIMA}(2,0,0)(0,1,1)_{168}$, $\text{ARIMA}(3,1,1)(0,1,1)_{168}$, and $\text{ARIMA}(4,1,2)(0,1,1)_{168}$.

Also checking the Mean Absolute Percentage Error (MAPE) on the training set, they actually have the lowest three values (MAPE $< 6.2\%$). Nevertheless, MAPE is not a single valid criterion for selection, since a low error predicting the training set itself does not necessarily mean a good forecast power.

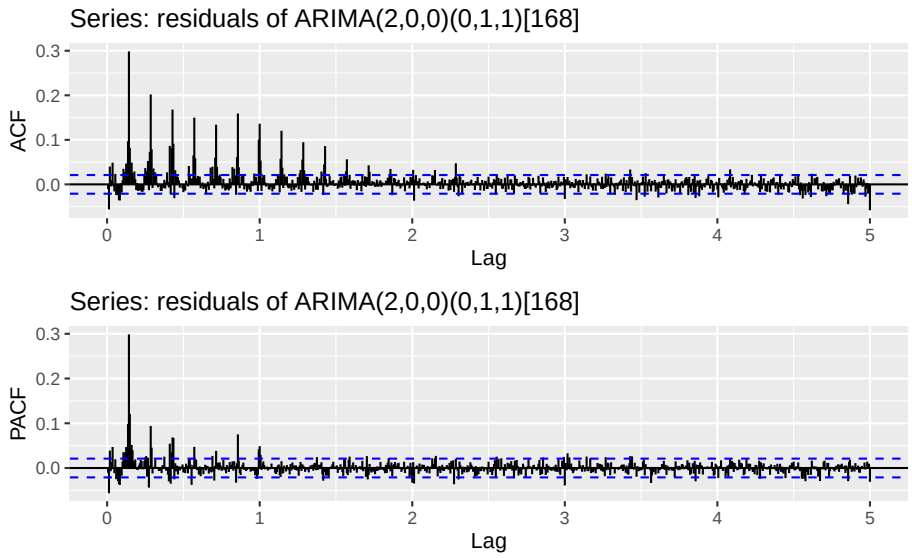
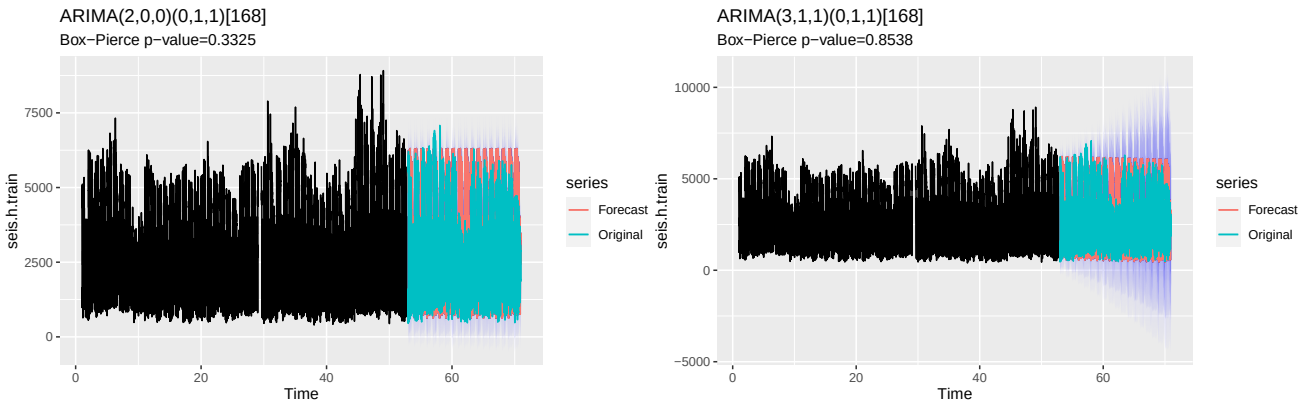



Figure 6: ACF and PACF plots for $\text{ARIMA}(2,0,0)(0,1,1)_{168}$

The ACF and PACF of the residuals for all three models are really similar (Figure 6), even if the models themselves have great differences in meaning and in number of parameters: they highlight a clear seasonal ARMA process with periodicity 24. This shows how our previous conclusion that the 168 hours seasonality could include the 24 hours one is flawed. Unfortunately, the parameters estimator we are using does not permit to estimate multiple seasonalities[1].

Probably related to the 24 hours periodicity that remains on the residuals, together with the non-Normality of the original data, all Q-Q plots show clear divergence on both sides of the distribution. The Box-Pierce test statistics accept the independence hypothesis between residuals.

Then, we assess how the forecasts behave on the test set. The MAPE for all three models more than doubles to around 17% for ARIMA$(2,0,0)(0,1,1)_{168}$ and around 15% for the other two. This is expected, since normally the model is more fitted to the training data than to unknown future data. Also, it must be noted that a January $1^{st}$ period lies in our test dataset, and without being able to include a yearly seasonality, that leads to greater errors just from those couple of weeks (Figure 7a).




(a) ARIMA$(2,0,0)(0,1,1)_{168}$ (b) ARIMA$(3,1,1)(0,1,1)_{168}$

Figure 7: ARIMA forecasts

ARIMA$(4,1,2)(0,1,1)_{168}$ behaves like ARIMA$(3,1,1)(0,1,1)_{168}$ with faster increasing variablity

We can see in Figure 7 the forecast behaviour for our models. We can appreciate the constant span of the confidence intervals in Figure 7a, in constrast with the more complex models that have increasing intervals further away from the training set. It must be noted that the increasing intervals include negative values, that are actually impossible to obtain given the data meaning (difference between maximum and minimum measurements).

The increasing variability also make the percentage of test values falling inside the confidence intervals to be higher than normal:

- ARIMA$(2,0,0)(0,1,1)_{168}$: $\approx$ 81% of data is in the 80% CI and $\approx$ 91% in the 95% CI
- ARIMA$(3,1,1)(0,1,1)_{168}$: $\approx$ 96% of data is in the 80% CI and $\approx$ 98% in the 95% CI
- ARIMA$(4,1,2)(0,1,1)_{168}$: $\approx$ 98% of data is in the 80% CI and $\approx$ 99% in the 95% CI

Given all these factors, we are confident choosing the ARIMA$(2,0,0)(0,1,1)_{168}$ model, being aware of the missed 24 hours seasonality.

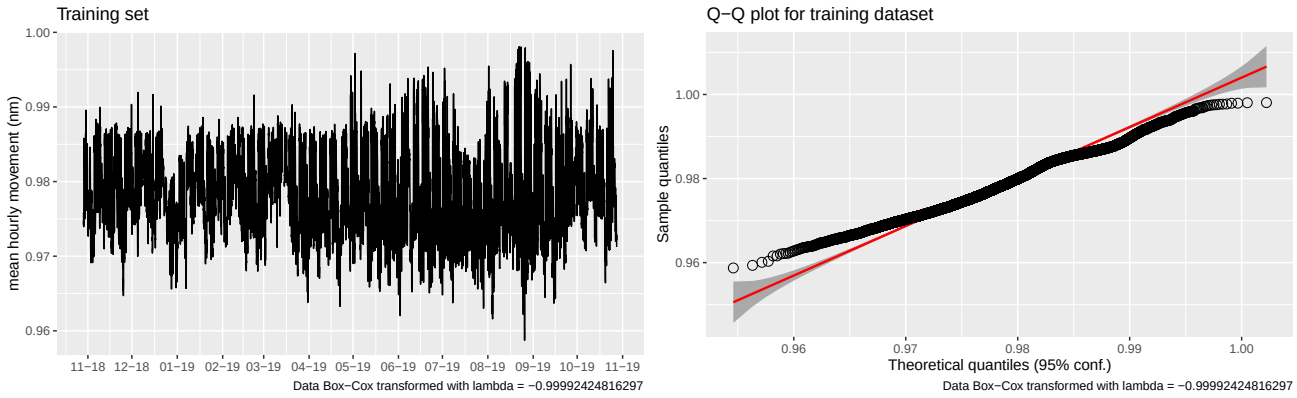
[1] *Arima* method from the *forecast* R package, only specifying the seasonal and non-seasonal orders

## 3.2 Membach station

### ARIMA

Similarly to data from Uccle, data from Membach doesn't respect the Normality assumption. We tested if the Q-Q plot showed significantly better approximation of the Normal distribution after applying the *log* or *Box-Cox* transformations, and the latter makes the data approach Normality with $\lambda \approx -1$ (Figure 8). This solves the high peaks problem we already noticed while decomposing to highlight the trend and seasonalities.

Since the $\lambda$ value approximately simplifies the Box-Cox transformation formula from $(y^\lambda - 1)/\lambda$ into $(y-1)/y$, and since we only have positive values, all transformed values fall into the higher $0-1$ spectrum, in particular $0.95-1$ (Figure 8a). This will probably cause some approximation errors that could rig the final results.




(a) Transformed training set

(b) Normal vs. transformed data distribution

Figure 8: Results of Box-Cox transformation

The transformed data still shows no constant mean, even if it has a way better shape than the original data. Thus, we consider the usage of the *diff* operator, simple and at lag 168. Interestingly, the analysis of ACF and PACF plots hints to similar models to the Uccle station: ARIMA$(0,1,0)(0,1,1)_{168}$ and ARIMA$(2,0,0)(0,1,1)_{168}$.

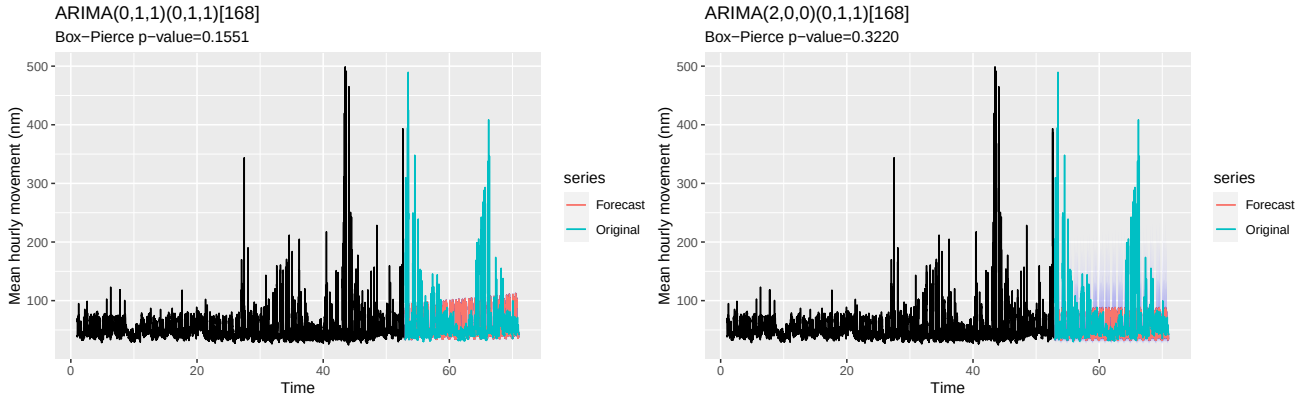
In this case, the employment of *auto.arima* stepwise methodology does not lead to any useful model, since it obtains AICc = inf for the two manually-chosen models, that have an AICc value around -84.500, while the best model found by that methodology has an AICc around -81.000. This can be caused by approximation errors[2], as we supposed earlier.

The analysis of ACF/PACF plots of the residuals from both models do not lead to any new model but, as with the Uccle station, we can notice a clear 24 hours periodicity.

Both models have valid p-value results for the Box-Pierce test, similar AICc values, and similar MAPE on the training set. Thus, we can assess the forecasts for both models in Figure 9.

The first model catches a rapidly-increasing trend, while the latter predicts more stable values moving the most part of the increasing trend into increasing variability. Visually, both could be acceptable, even if the latter seems to better behave on *normal* days losing precision on high-valued ones, while the first tries to better accomodate the high-valued days losing precision on the *normal* ones.

[2]The *auto.arima* methodology was used only with initial approximation, due to high computing time




(a) ARIMA$(0,1,1)(0,1,1)_{168}$ (b) ARIMA$(2,0,0)(0,1,1)_{168}$

Figure 9: ARIMA forecasts

Analytically, we have quite a difference between the two models in terms of adherence of the confidence intervals:

- ARIMA$(0,1,1)(0,1,1)_{168}$: $\approx 98\%$ of data is in the 80% CI and $\approx 99\%$ in the 95% CI
- ARIMA$(2,0,0)(0,1,1)_{168}$: $\approx 73\%$ of data is in the 80% CI and $\approx 89\%$ in the 95% CI

As we see, the first model over-generalises the variability, while the latter seems more precise, even if losing some information. Also considering the MAPE on the test set, the latter model fits the data better with a 16% error, instead of a 18%.

Thus, we can select the ARIMA$(2,0,0)(0,1,1)_{168}$ model on the Box-Cox($\lambda \approx -1$)-transformed data, still being aware of the missed 24 hours seasonality, similarly to the Uccle station.

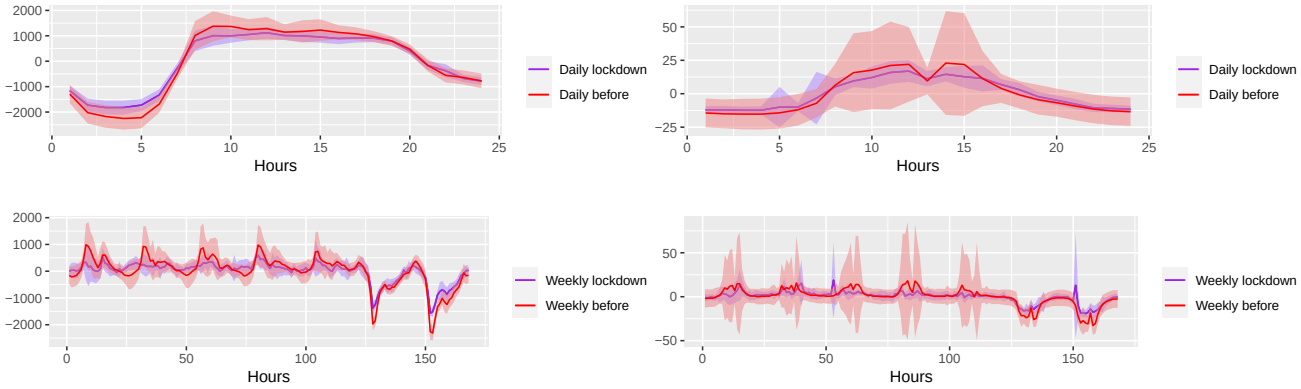
# 4 Conclusion

## 4.1 Lockdown influence

Even if the two models have the same order, they cannot be directly compared, for the Membach model works after a transformation pass.

We can still compare how the seasonalities differ with respect with each original dataset and between each other. We extract the seasonalities for each station highlighting the standard deviation confidence band for an estimated error of 67%. This estimation lies on the assumption that values for each single hour in the hourly/weekly seasonality is Normally distributed, a slightly different assumption that is not necessarily refuted by the already seen Q-Q plots.

As we can see in Figure 10, even if the seasonalities shapes seem very different, in particular regarding the weekly seasonality, the differences are still included in the highlighted confidence bands, apart from a few hours. This is true for both stations. Yet, it is useful to notice that the weekly seasonalities include a possibly interesting daily component, influenced by the weekend drops in values.




(a) Uccle station

(b) Membach station

Figure 10: Seasonalities computed before and after the lockdown
Seasonalities are extracted together

## 4.2 Limitations and future works

The main limitations of this projects that we can identify are: multiple seasonalities and Normality assumption.

The tested models do not effectively include both daily and weekly seasonalities, even if initially we thought so. Also they cannot include the apparent yearly seasonality, mainly because of the short timeseries. To overcome this limitation, we could try different models, also ARIMA-based, that can estimate multiple seasonalities. The first that we tested models seasonalites through Fourier terms. The main problem that lead us not to go further with that approach is that our seasonalities cannot be expressed by a combination of few sine and cosine curves. This requires a high number of curves to estimate, that requires an heavy computational power.

To compensate those difficulties, in the literature it is suggested to test TBATS models, which combines Fourier terms, Box-Cox transformation, and exponential smoothing. This seems a suitable approach given the analyses we performed until now, that can possibly address the non-Normal distribution.

The final suggestion for future work would be to better analyse the differences between stations, in such a way to better correlate different influences of the lockdown on the areas near each station.

# References


[1] Elizabeth Gibney. Coronavirus lockdowns have changed the way Earth moves. *Nature (online)*, March 2020. ISSN 1476-4687. doi: 10.1038/d41586-020-00965-x. URL https://www.nature.com/articles/d41586-020-00965-x.

[2] Royal Observatory of Belgium. Other types of seismic events, April 2020. URL http://seismologie.be/en/seismology/other-types-of-seismic-events.

[3] Royal Observatory of Belgium. Data policy, April 2020. URL http://seismologie.be/en/legal-notices/data-policy.

[4] Royal Observatory of Belgium. Uccle station data from 02-04-2020, April 2020. URL http://seismologie.be/data/csv/SEIS-2020-093-UCCS.csv.

[5] Royal Observatory of Belgium. Uccle station data events from 02-04-2020, April 2020. URL http://seismologie.be/data/csv/events_2020-04-02.json.

[6] Royal Observatory of Belgium. Uccle station data visualization from 02-04-2020, April 2020. URL http://seismologie.be/en/seismology/seismograms/uccs/20200402.