École nationale
de la statistique
et de l'analyse
de l'information

An Investigation into Water Evaporation: Monitoring and Analysis

Students:
Louis ALLAIN
Léonard GOUSSET
Julien HEURTIN

Teacher:
Youssef Esstafa

Time Series Project

December 2023

Introduction

The investigation into surface water evaporation assumes paramount importance in the context of environmental change. Understanding and monitoring this natural process are essential for anticipating and comprehending climate variations, thereby offering valuable insights into more effective management of water resources. Examining surface water evaporation becomes a pivotal element in adapting strategies to mitigate environmental impacts, particularly within the realms of agricultural management, drought prevention, and the overall preservation of ecological balance.

The scrutiny of surface water evaporation through the analysis of time series data proves to be an invaluable approach in capturing the nuances of this variable over time. Utilizing sequential data enables the identification of trends, seasonal cycles, and anomalies, providing a profound comprehension of fluctuations in surface water evaporation. This method also aids in highlighting intricate patterns that may be challenging to discern through point observations alone. By amalgamating the benefits of time series analysis with surface water evaporation data, we can refine our predictive models and make informed decisions for sustainable water resource management, taking into account the temporal dynamics of this essential variable.

Our current endeavor involves a comprehensive exploration of surface water evaporation using time series analysis and machine learning models. Initially, we will examine surface water evaporation as an independent time series, delving into trends and variations over time. Subsequently, we plan to integrate the temperature variable into our analysis, aiming to unravel the intricate relationships between temperature and surface water evaporation. This integrated approach will yield in-depth insights into the mechanisms governing surface water evaporation and foster a better understanding of the influence of temperature on this critical process.

The first phase of our project will entail a thorough exploratory analysis of the two variables in question. Moving into the second phase, we will strive to model these variables using time series and machine learning methodologies. At this juncture, we will leverage temperature variable predictions to evaluate their efficacy in early predictions of surface water evaporation. While we retain the option to incorporate other environmental variables, it is imperative to note that predictions pertaining to these variables will be observational, with the sole aim of understanding their impact on surface water evaporation.

The overarching objective remains the development of a robust tool capable of monitoring surface water evaporation, holding promising prospects for prevention and support in agricultural practices.

Contents

1 Data aquisition and preprocessing

Our data monitoring project relies on the utilization of Google Earth Engine (GEE) to extract the necessary information. Google Earth Engine is a cloud-based platform that enables large-scale geospatial data analysis. This platform provides access to a broad range of satellite and geospatial data, making it particularly well-suited for our long-term monitoring objectives.

The choice of Google Earth Engine was motivated by its capability to handle massive datasets and provide advanced analytical tools, a task that would have been challenging with a simple local database. Importantly, the Google Earth Engine API is primarily accessible in Python, which led to the inclusion of this language in our project. This choice allows us to continuously retrieve data for ongoing monitoring efforts.

To collect our data, we opted for ERA5, a Copernicus database. Copernicus is a European program providing Earth observation services with the goal of understanding and monitoring our planet's environment. It encompasses various data types, including climate, atmosphere, oceans, and more.

ERA5, specifically, is a product of the Copernicus Climate Change Service (C3S). It is a high-resolution climate dataset that offers detailed information on various climate parameters, including temperature and water evaporation. While ERA5 data extends back to 1950, for reasons of data storage considerations, we have chosen to focus on the past 40 years. This extended period allows us to observe and analyze long-term trends, essential for understanding climate changes and variations in environmental conditions over decades.

Firstly, we will explain the data retrieval process. Then, in the second phase, we will present the preprocessing operations for the data.

1.1 Data Retrieval using Google Earth Engine API

The Python script we wrote utilizes the Google Earth Engine (GEE) Python API to process ERA5 climate data for the region of France. The process involves selecting a specific climate variable, defining a date range, and retrieving relevant data from the GEE image collection. The region of interest is filtered to correspond to France, and the data is then processed by aggregating it temporally and spatially.

The key steps include iterating through the specified date range, extracting images for the selected variable, and converting the Earth Engine array to a Pandas DataFrame. This DataFrame undergoes various transformations, such as extracting date components, computing mean values, and creating a point geometry for each data entry. The result is a processed DataFrame containing climate data ready for further analysis.

The script demonstrates how to instantiate the class with specific parameters, such as the selected climate variable and date range. The process_data method is then called to retrieve and process the data accordingly. The resulting DataFrame (result_df) serves as the output, containing the processed climate data for the specified region and time period. This file is intended for further analysis and treatment in R that we're going to explain now.

1.2 Data preprocessing using R

To prepare the data for analysis, several preprocessing steps were undertaken to enhance its usability and relevance. Initially, unnecessary details, particularly those related to precise geographical locations denoted by the variable "Point," were removed. This strategic omission is driven by the objective of aggregating data at the national level, specifically focusing on France. While this simplifies the study and renders it more manageable, it's important to note that it introduces a trade-off, reducing the precision of the analysis.

For the variable temperature, subsequent steps involved organizing the dataset by date and calculating the mean temperature for each month. This temporal aggregation aims to distill key trends over time, facilitating a more comprehensive understanding of temperature patterns. To ensure consistency and ease of interpretation, the temperature values, initially presented in Kelvin, were converted to Celsius. This standardization aligns with common meteorological practices and simplifies the interpretation of temperature variations.

For the water evaporation variable, we did the same preprocessing but without the kelvin transformation. For all variables, temperature, water evaporation and all exogenous variables considered later are normalized.

The resultant dataset, now refined and devoid of unnecessary geographical details, is structured for further analysis. It encapsulates a summarized view of temperature trends over time at the national level, providing a foundation for meaningful insights without the complexities associated with precise geographical coordinates.

2 Exploratory Data Analysis

In the upcoming exploratory analysis, our focus will be directed towards two pivotal variables: temperature and water evaporation. These variables have been carefully chosen for their significance in unraveling essential insights about the environmental conditions.

2.1 Analysis of the variable Temperature

We now have the temperature variable that we will represent over time. The variable 'temperature' represents the average temperature in France each month. For the initial graphical representation, we leave the values as they are, but subsequently, we will normalize them to facilitate calculations. Here is the resulting graph:

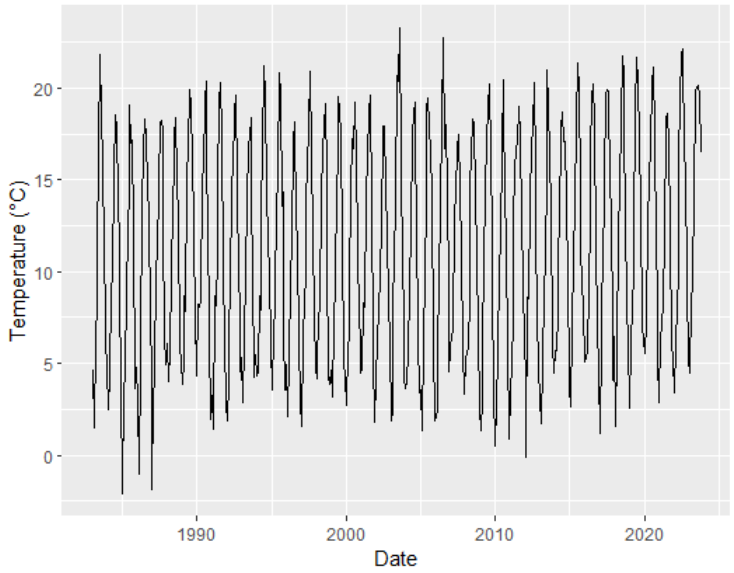


Figure 2: Temperature each months between 1983 and 2023

Due to the nature of the data, a clear seasonality can be observed in temperature variations over the years. Additionally, there appears to be an upward trend in temperature, with higher values around the 2020s compared to 1980. Furthermore, it is evident that minimum temperatures seem to be decreasing, becoming less cold over time. Therefore, these two elements contribute to the non-stationarity of the series. Next, we will decompose the time series to verify the presence of a positive trend. We obtain the following chart:

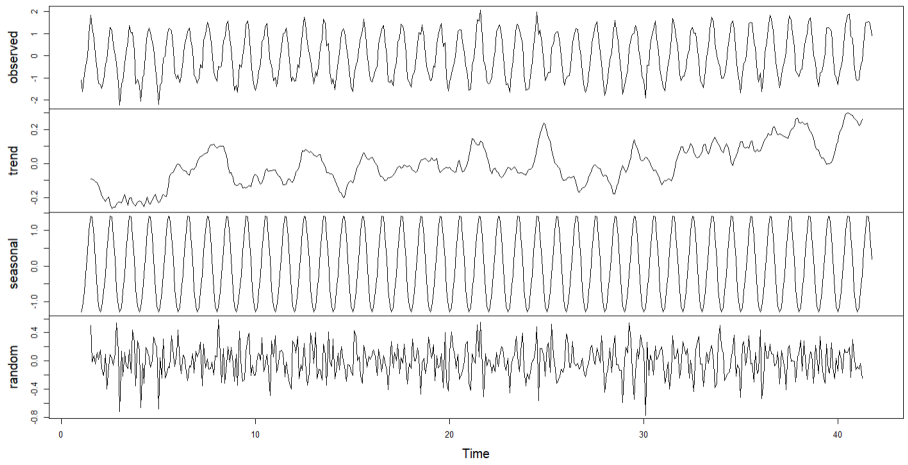


Figure 3: Decomposition of temperature time series

One can then confirm the hypothesis of a positive trend. We will now graphically represent the autocorrelation function (ACF) of the temperature as well as the differenced series with a lag of 12 time steps, considering that each season spans 12 months. The resulting graph is as follows :

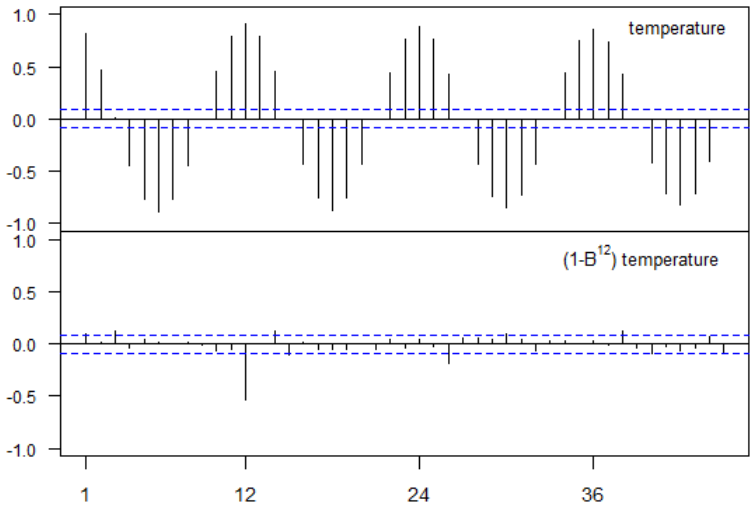


Figure 4: Autocorrelation Function (ACF) of the temperature and the differenced temperature.

The series exhibits an intriguing behavior, characterized by a sequential and significant negative autocorrelation starting at lag 6 and repeating every 12 months. This distinctive pattern can be attributed to the seasonal variation, where the temperature differences arise due to the changing seasons. For example, if today is winter with cold temperatures, in 6 months, we expect higher temperatures in the summer, leading to the observed negative autocorrelation. These temperature trends typically move in opposite directions. Moreover, at every 12 lags, there is a notable peaks that decay slowly, indicating the presence of seasonal non-stationarity.

Let's now examine the differenced series. It exhibits a peak at lag 12, a value significantly different from 0 shortly after 24, followed by a sharp attenuation—characteristics indicative of a stationary series with seasonality.

In the context of a stationary time series with seasonality, the SARIMA (Seasonal Autoregressive Integrated Moving Average) model is often regarded as ideal. This preference stems from the model's ability to effectively address key aspects of such time series. Therefore, we will proceed to attempt modeling this time series using a SARIMA model.

2.2 Analysis of the water evaporation

This variable represents the amount of evaporation from surface water storage like lakes and inundated areas. We specifically do not take into account oceans. The unit of measurement is meter of water equivalent. In a first time, we are going to sum the water evaporation by month. That way we have less points but a more readable time series. We can start by plotting the time series as is.

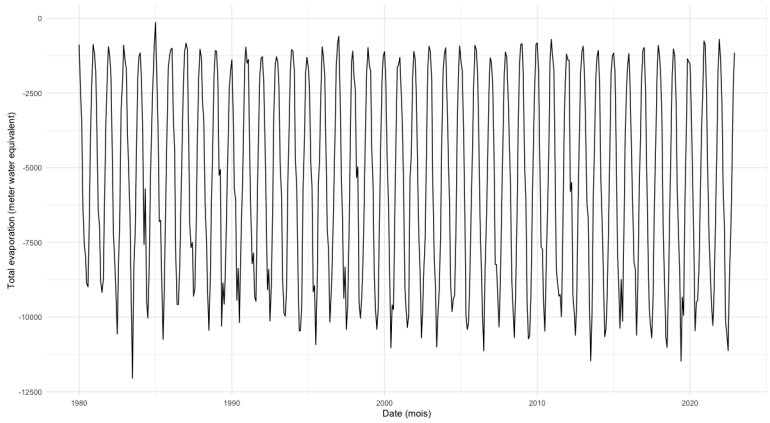


Figure 5: Total evaporation evolution between 1980 and 2023

We clearly see a seasonality over the years. Naturally, we can expect to have a link with seasons and thus have a 12 month seasonality. From this view we do not clearly identify a trend. We are going to check the seasonality with a season plot.

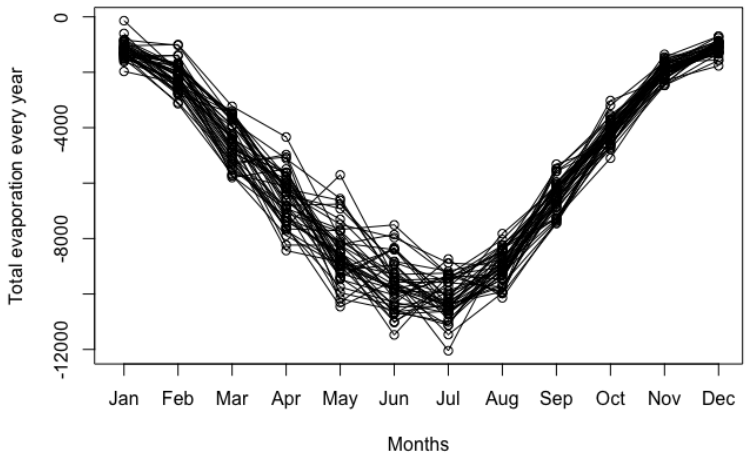


Figure 6: Total evaporation evolution every year from 1980 to 2023

As we can see, the total evaporation is seasonal, but do not present a trend over the years. The months of June and July have a lot more evaporation compared to winter months like December, January and February. Before moving on to the ACF and PACF studies we are going to check the stationnarity of our time series. First of all an Augmented Dickey-Fuller test gives us a pvalue of 0.01 < 0.05. Therefore this test suggests that our series is stationary. We can also use the decompose function to take a look at the trend. As we can see on the figure below, the seasonality is indeed present, but we notice a light downward trend.

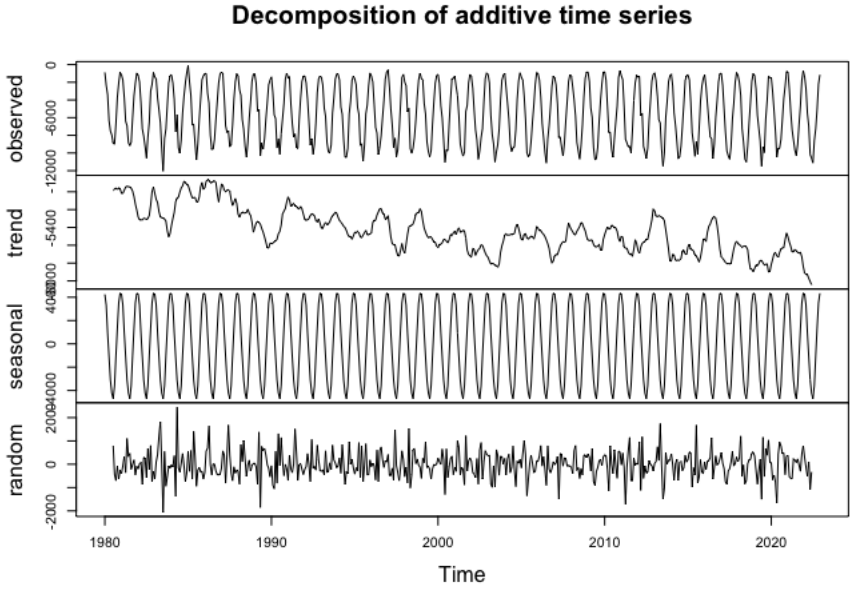


Figure 7: Decomposition of the totale evaporation time series

3 Modeling temperature and water evaporation

After completing the Exploratory Data Analysis (EDA), our next step involves delving into the modeling phase. In this initial stage, our focus will be on modeling the temperature data, and subsequently, we will turn our attention to modeling soil evaporation.

Initially, we partition both datasets into training and test samples. About eighty percent of the data will be utilized for training, while the remaining portion will be reserved for model validation. Consequently, the training data spans from January 1983 to 2015, while the test data encompasses 2016 to 2022.

3.1 Modeling the time series of temperature

As previously elucidated, a SARIMA model appears to be suitable. Consequently, we will commence by employing this model. Subsequently, we will explore the utilization of machine learning algorithms.

3.1.1 Modeling temperature using a SARIMA model

First, let's define what is a SARIMA model. A Seasonal Autoregressive Integrated Moving Average (SARIMA) model is a time series forecasting method that extends the ARIMA model to account for seasonality. The SARIMA(p, d, q) (P, D, Q)s model is characterized by three components:

- Autoregressive (AR) component denoted by (p, P): Captures the relationship between the current observation and its past observations in both the original time series and the seasonal differences.
- Integrated (I) component denoted by d and D: Represents the order of differencing needed to achieve stationarity in the original time series and the seasonal differences.
- Moving Average (MA) component denoted by (q, Q): Models the relationship between the current observation and the residual errors of past observations in both the original time series and the seasonal differences.
- The seasonal parameter (s) represents the periodicity of the seasonal pattern, indicating the number of observations per season (for monthly data with yearly seasonality like us, s = 12)

The mathematical representation of a ARIMA model is given by:

$$Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \ldots - \phi_p Y_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}$$

For the ARIMA with the seasonal part:

$$Y_t - Y_{t-s} - \phi_1(Y_{t-1} - Y_{t-s-1}) - \phi_2(Y_{t-2} - Y_{t-s-2}) - \ldots - \phi_P(Y_{t-P} - Y_{t-s-P})$$

$$= \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_Q \varepsilon_{t-Q}$$

Where: Y_t is the observed value at time t, ε_t is the error term at time t, ϕ_i and θ_i are the autoregressive and moving average coefficients, p, d, q are the non-seasonal ARIMA orders, P, D, Q, s are the seasonal SARIMA orders.

Now, to obtain the parameters for our SARIMA model, we will analyze the autocorrelation function (ACF) and partial autocorrelation function (PACF) of both the original time series and the differenced series. We obtain the following plots:

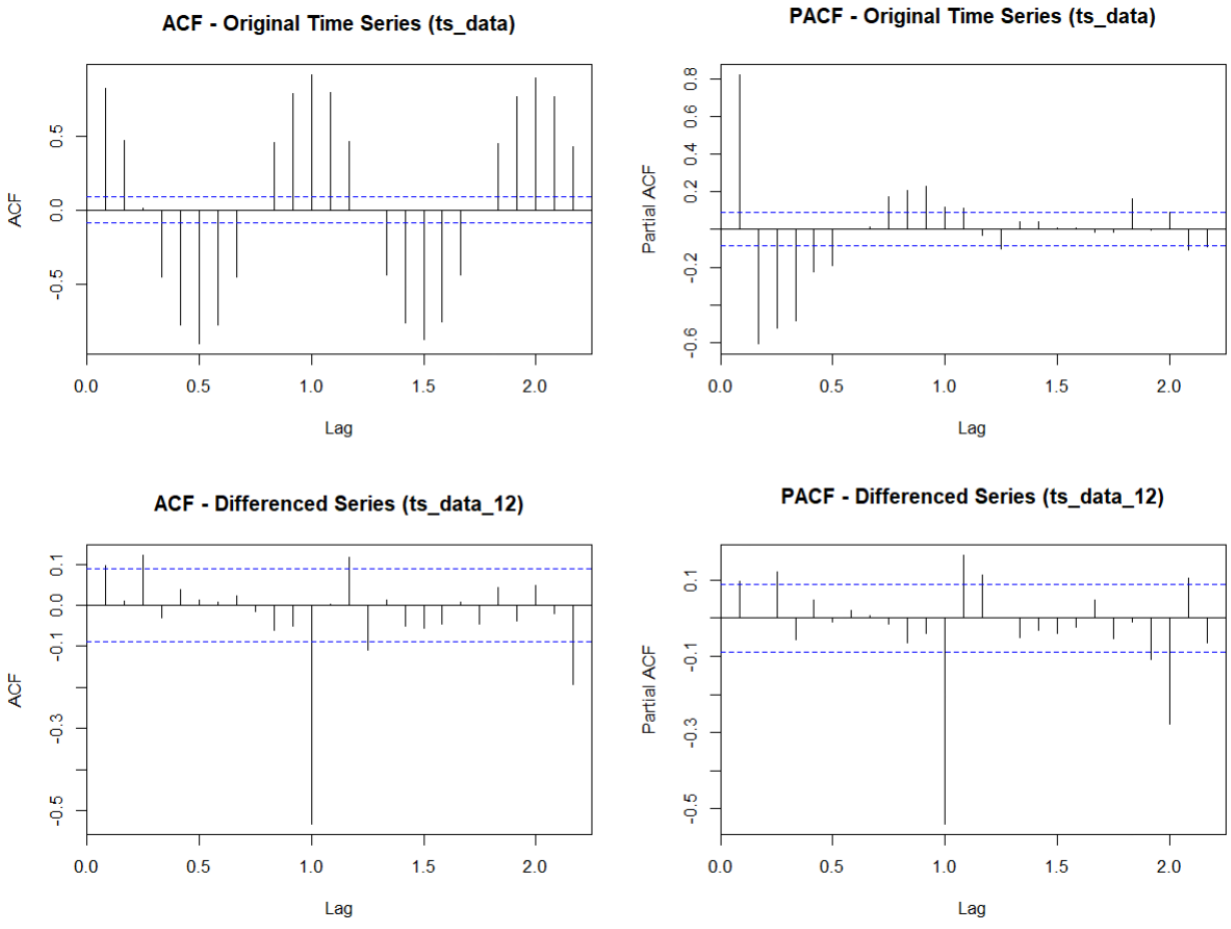


Figure 8: ACF and PACF for both the original time series and the differenced series

We have already analyzed the ACF, so we will now focus on the PACF. We can see in the regular PACF a peak at the first lag, justifying the inclusion of the AR(1) term. For the seasonal PACF, the presence of periodic peaks at lags multiple of 12 suggests a seasonal component, justifying the seasonal MA(1) term.

Regarding the parameters d and D, two scenarios are considered:

- If $d = 0$ and $D = 1$, it means $y_t = Y_t - Y_{t-s}$, where y_t is the differenced series and Y_{t-s} is the original seasonal lag.
- If $d = 1$ and $D = 1$, then $y_t = (Y_t - Y_{t-1}) - (Y_{t-s} - Y_{t-s-1}) = Y_t - Y_{t-1} - Y_{t-s} + Y_{t-s-1}$.

In the context of seasonal series like temperature, these considerations about seasonal (D) and non-seasonal (d) differencing are crucial to capture trends and seasonal variations in the data. These adjustments contribute to making the model more suitable for the specific structure of the time series. We're going to focus on the seasonal variation by choosing $D = 1$.

After the evaluation of the ACF and PACF, the parameters that seem appropriate for a SARIMA model are (1,0,0) (1,1,0). We also add a "drift", an option in SARIMA is suitable when the time series exhibits a positive trend. It introduces a constant factor to better capture and represent this trend in the data. The model parameters, including the drift term, appear to be statistically significant:

$$\text{ARIMA(1,0,0)(1,1,0)[12] with drift}$$

Coefficients:

$$\begin{aligned} &\text{ar1: } 0.1385, \quad \text{sar1: } -0.5263, \quad \text{drift: } 9 \times 10^{-4} \\ &\text{s.e.: } 0.0501, \quad 0.0434, \quad 1 \times 10^{-3} \end{aligned}$$

$$\begin{aligned} \text{sigma}^2 &= 0.09183 \\ \text{log likelihood} &= -91.52 \end{aligned}$$

However, challenges arise when conducting the Box-Ljung test on the model residuals:

$$\text{Box-Ljung test}$$

$$\begin{aligned} \text{X-squared} &= 96.882, \\ \text{df} &= 24, \\ \text{p-value} &= 1.017 \times 10^{-10} \end{aligned}$$

We chose lag=24 in the Box-Ljung test because it is advisable to select a lag value that is at least twice the seasonal period for seasonal time series data. In our case, since the seasonal period is 12, choosing a lag of 24 allows for a thorough examination of autocorrelation in the residuals across multiple seasonal cycles. The p-value of the Box-Ljung test is extremely low, suggesting a lack of independence in the residuals. Despite the significance of the model parameters, the autocorrelation in the residuals poses a problem, indicating that the model might not adequately capture some temporal patterns in the data. Further investigation or model refinement may be needed to address this issue.

The time series under consideration proves too complex to discern its parameters through a simple analysis of the autocorrelation function (ACF) and partial autocorrelation function (PACF). To address this complexity, we are opting to use a function named "get.best.arima"[1]. This function is designed to streamline the process of identifying the optimal parameters for an ARIMA model, given a time series (x.ts). Instead of relying on manual inspection of ACF and PACF plots, this function automates the search for the best-fitting model by systematically exploring various combinations of non-seasonal and seasonal ARIMA orders.

In essence, the function initializes with a high value for best.aic and then iterates through possible combinations of ARIMA orders, fitting models and calculating the consistent AIC (cAIC)

for each configuration. The cAIC serves as a criterion for balancing model fit and complexity, penalizing additional parameters.

The function continuously updates the best-fitting model and corresponding parameters whenever a configuration yields a smaller cAIC, indicating an improved fit. This systematic search concludes with the function returning a list containing the parameters of the ARIMA model that minimizes the cAIC.

This automated approach is particularly useful when dealing with time series data that exhibits complexity not easily discernible through visual inspection alone. By leveraging the function, one can efficiently navigate the parameter space and arrive at a data-driven decision regarding the most suitable ARIMA model for the given time series. With this function, we then obtain this model:

$$\text{ARIMA(2,0,1)(3,1,2)[12] with drift}$$

$$\text{Box-Ljung test}$$

$$\text{X-squared} = 34.879,$$

$$\text{df} = 24,$$

$$\text{p-value} = 0.07021$$

The refined model, with its improved parameters and lower AIC, suggests a better fit to the training data. Nevertheless, the persistence of autocorrelation in the residuals raises concerns about the model's ability to capture all temporal patterns in the data. The lack of independence in the residuals indicates that the refined model may not be sufficiently significant, emphasizing the importance of continued model refinement and assessment.

With the aim of enhancing the SARIMA model's ability to capture more intricate seasonal patterns, we have decided to increase the q parameter. This parameter is associated with the seasonal Moving Average (MA) component of the model, and its augmentation will allow the model to better capture temporal dependencies and respond in a more detailed manner to seasonal residuals. By increasing q, we seek to refine the model's capability to represent subtle variations and more complex patterns present in the time series. However, it is crucial to maintain a balance between model complexity and generalization ability to avoid overfitting to the specific features of the training data.
After gradually increasing q and comparing various models, we obtain the following SARIMA model:

$$\text{ARIMA(2,0,11)(3,1,2)[12] with drift}$$

$$\text{Box-Ljung test}$$

$$\text{X-squared} = 11.626,$$

$$\text{df} = 24,$$

$$\text{p-value} = 0.9838$$

We now have a model with residuals that exhibit white noise characteristics. The model is thus significant. Consequently, we can proceed to evaluate the model on the test data. To compare models, we will use Mean Squared Error (MSE) as a common metric. The choice of MSE is judicious for time series due to its effectiveness in capturing the magnitude and direction of prediction errors over continuous temporal data. This approach enables us to compare SARIMA models with other machine learning models. On the training data, we achieve an MSE of 5.88, which appears to be a promising result, especially considering the nearly 100-sample validation

set size. This result will be subject to comparison with other models. We obtain the following plots by comapring the predictions with the test set:

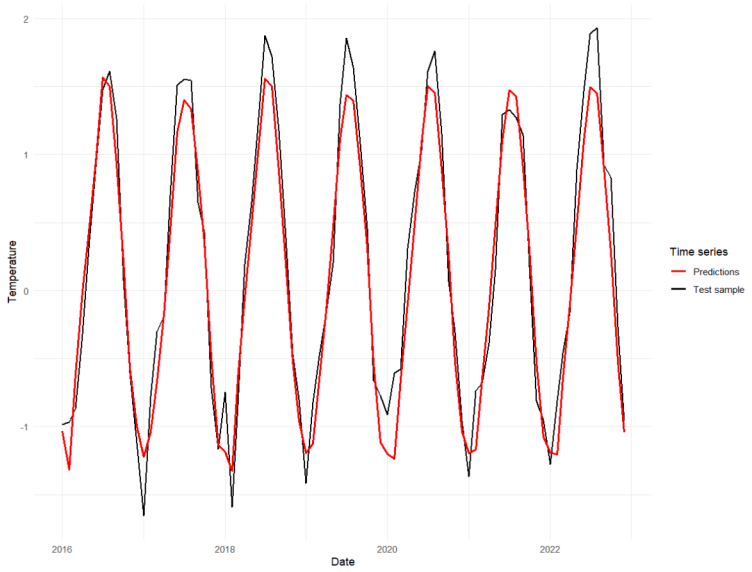


Figure 9: Fitting our model to the test sample with a SARIMA

We observe that our SARIMA model effectively captures temperatures during months when temperatures are not extreme (i.e., excluding approximately June-July and January-February). Representing these data would therefore require implementing modeling for extreme values, such as Generalized Extreme Value (GEV) in the general case first. Furthermore, we think the increase in temperature due to environmental changes introduces a positive trend that the SARIMA model struggles to capture, even with the inclusion of a drift component.

Subsequently, we will attempt to implement a machine learning algorithm to model the temperature variable.

3.1.2 Modeling temperature using boosting

We will implement two machine learning algorithms using boosting. Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to reduce learning errors. Boosting algorithms exhibit excellent predictive properties, hence the decision to employ them. The two algorithms will be gbm and xgboost.

For our data, we will create relevant features. We will provide the models with lagged temperature values for the past 6 months and the temperature of the previous year's same month. Additionally, we will include an encoded variable for the month, taking the value of 1 if it corresponds to the variable's month and 0 otherwise. We will adjust the parameters using cross-validation with the R "Caret" package.

After tuning the hyperparameters of XGBoost through cross-validation, we trained the model on our training data. We obtained a mean squared error (MSE) of 9.58, which is considerably

less satisfactory compared to the SARIMA model. We obtain the following plots by comapring the predictions with the test set:

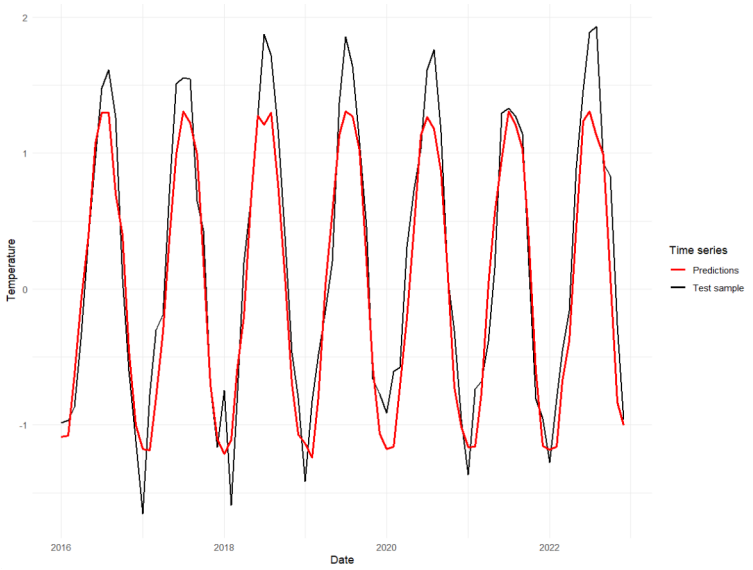


Figure 10: Fitting our model to the test sample with Xgboost

We will now perform the same operations for the gbm model. We choose a Gaussian distribution for squared error. After cross-validation and training, we test our model on our test data. We obtain a MSE of 7.87; thus, the model is better than XGBoost but still not as good as SARIMA. We obtain the following plots by comapring the predictions with the test set:

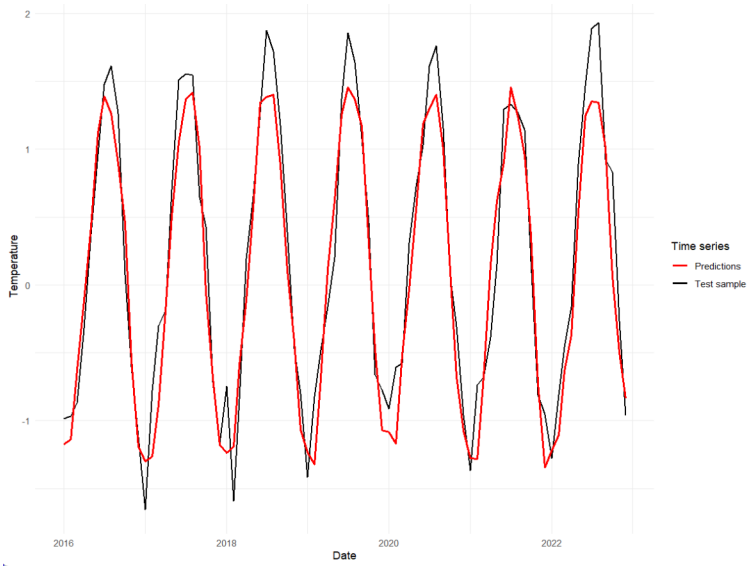


Figure 11: Fitting our model to the test sample with gbm

For both boosting models, we observe similar limitations as the SARIMA model in modeling extremes

Now, having examined the temperature time series, our focus will shift to water evaporation. The goal was to achieve an accurate prediction of future temperature to use as a variable and assess whether it enhances the modeling quality of water evaporation. This will be discussed in Part 4, where we will introduce a Sarimax model. But first, we will focus on modeling water evaporation without incorporating the temperature variable in the next part.

3.2 Modeling the time series of water evaporation

We recall that our variable is the sum by month of the total water evaporation. From a brief data analysis we suggested to consider a SARIMA model. That is what we are going to use in the following part. We are going to compare it to two other machine learning models, XGBoost and Random Forests.

3.2.1 Modeling with a SARIMA model

As seen in a previous section during our analysis of the temperature the model's parameters are quite hard to evaluate.

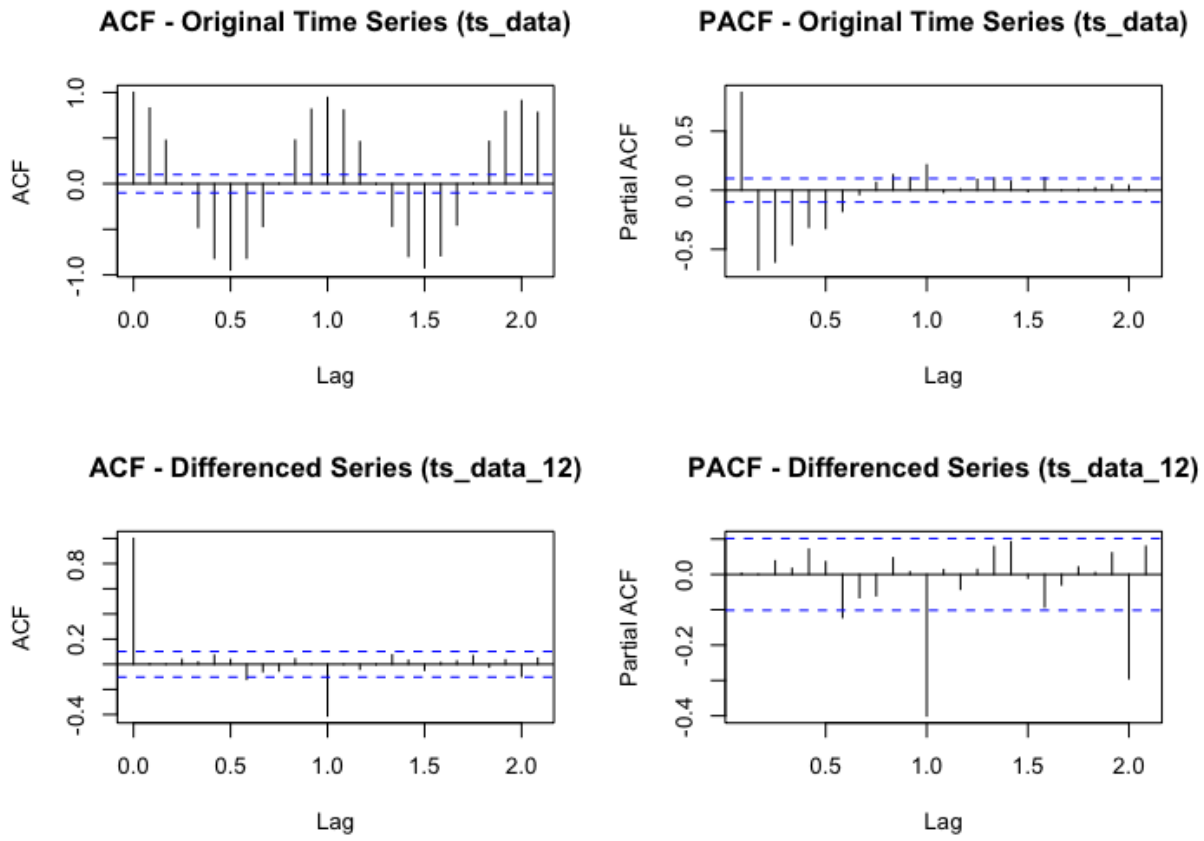
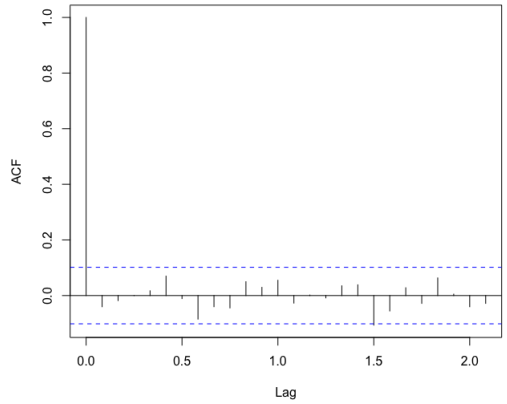


Figure 12: ACF and PACF for both the original time series and the differenced series

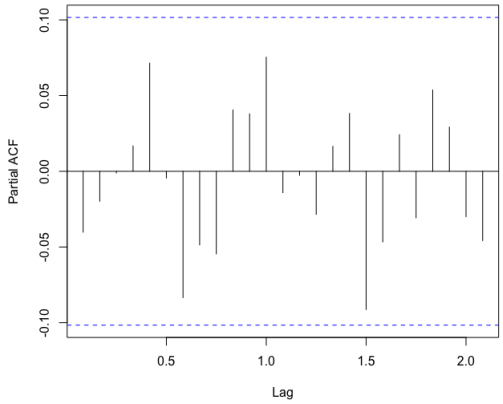
The ACF shows autocorrelation that are repeating each twelve months. In a very similar behavior to our ACF from the temperature variable. Therefore the model we will consider for this

variable is again a SARIMA of seasonality 12. Taking a look at the differenced ACF and PACF neither have tapering spikes. This suggests a 0 moving average **MA(0)** and a 0 autoregressive part **AR(0)**. Regarding the seasonal moving average and autoregressive, we see that the differenced PACF tapers in multiples of 12 (two negative spikes) and that the differenced ACF has one significative spike at lag 1. This suggests a seasonal moving average of 1 **SMA(1)**. The differenced ACF does not taper, we have only the presence of a significative spike at lag 12. So we can only suppose a seasonal autoregressive of 0, **SAR(0)**. Because we have a slight downward trend, figure 6, in a seasonal time series we are going to use set D=1.

Finally, the model we are going to examine is the following one: SARIMA$(0,0,0)(0,1,1)_{12}$. Moreover we checked the best model using a function called `get.best.arima`[1]. The choice criteria is based on the consistent AIC (page 144), and the chosen model for SARIMA parameters inferior or equal to 2 is the same as the one we identified with our ACF and PACF analysis. In order to validate our model we are going to check the reisudals.



(a) ACF of the residuals of the model SARIMA(0,0,0)(0,1,1)12



(b) PACF of the residuals of the model SARIMA(0,0,0)(0,1,1)12

The big positive spike in the ACF at lag 1 and the fact that the PACF shows no significative lag, indicates that our residuals are white noise. Checking the histogram of the residuals we can see that they are normally distributed. We can use the Box-Ljung test to finally convince our self that the residuals are indeed white noise. For a lag of 24 (which is twice our seasonality) we have $p_{value} = 0.69 > 0.05$. Therefore we can say that our residuals are white noise and our model is correct.

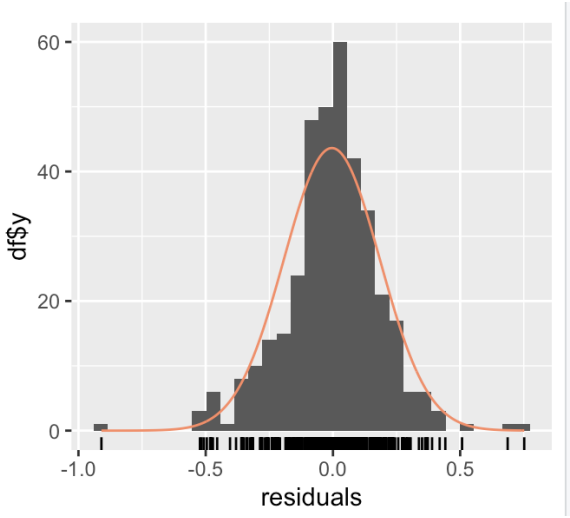
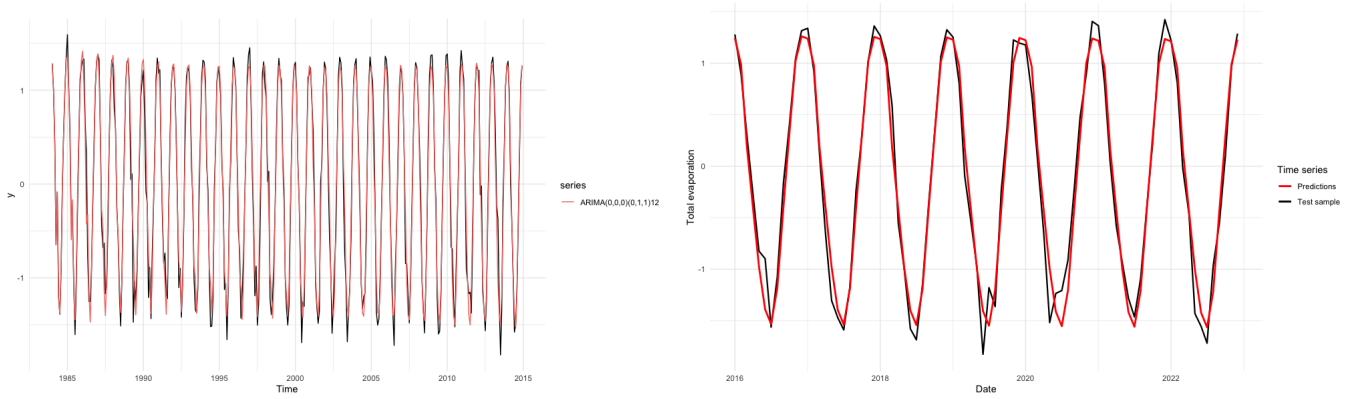


Figure 14: Histogram of the residuals

But how good in prediction is our model ? We are going to check the predictions on both the train and test samples and calculate the empirical risk on the test sample.



(a) Fitting our model to the train sample

(b) Fitting our model to the test sample

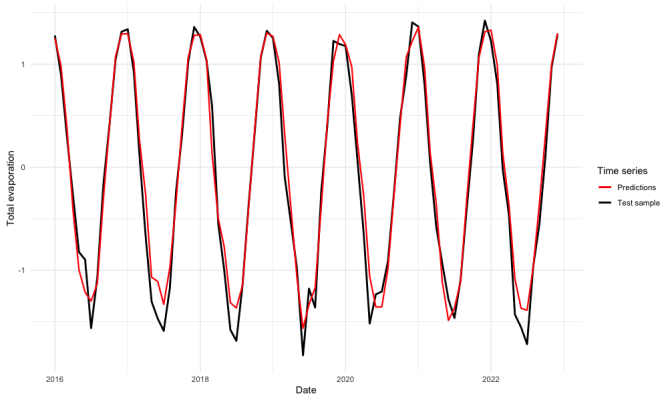
Our models seems to predict well the test sets. It has a hard time predicting sharp spikes, especially in 2019 and 2021, and is sometimes a little late in its predictions. Nevertheless, with a empirical risk of $\text{MSE}_{\text{SARIMA}} = 2.76$, we can consider our model to be very good.

3.2.2 Modeling with a Random Forest Model

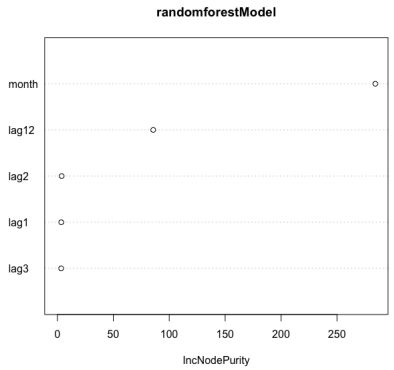
The main problem when using a machine learning problem is what features to use. Following what we did when using the XGBoost and GBM models for the temperature, we are going to use lag variables as feature variables. Here we are using the value of water evaporation of the previous month, two months prior, three months prior as well as 12 months prior (a year prior).

For this model we are going to use the packages `randomForest` and `Caret` for cross validation. We do not need to one-hot encode our month variable as trees handle categorical variables very well. This is very powerful in random forest models, you can you use both tabular data and

categorical variables. This is why we are performing this model to compare it to other machine learning algorithm. We thus perform cross-validation on the parameter *mtry* (the only one we can tune using those packages). We found that mtry $= 8$ is the best value.



(a) Fitting the predictions of the Random Forest to the test sample

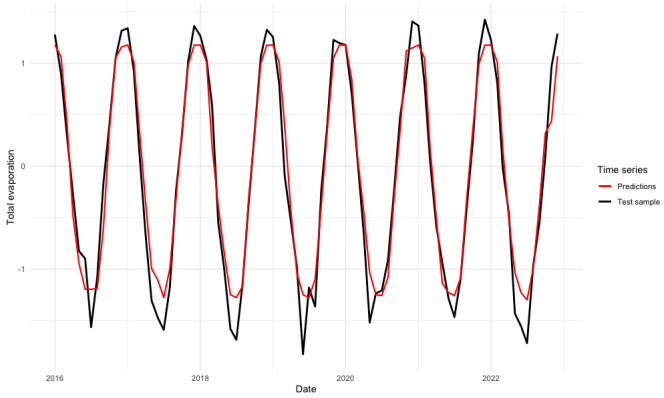


(b) Feature importance for our Random Forest model

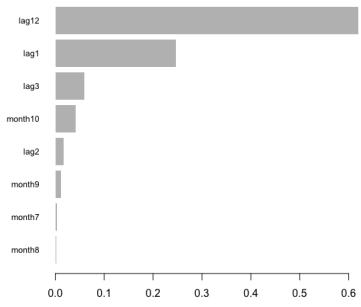
We found a very good accuracy with this model: $\text{MSE}_{\text{RandomForest}} = 2.86$. It is slightly above the SARIMA model. What is interesting with this model is to look at the features importance. As we can see, it is clearly the month variable that make the model perform, and not so much the water evaporation of the previous month.

3.2.3 Modeling with a XGBoost and GBM

In this section we are going to compare our Random Forest model to XGBoost and GBM models.



(a) Fitting the predictions of the XGBoost model to the test sample



(b) Feature importance of the 9 most important variables for our XGBoost model

The XGBoost model has a worst empirical risk compared to the SARIMA or the Random Forest ones: $\text{MSE}_{\text{XGBoost}} = 4.15$. This can be caused by the one-hot encoding of the month variable. It is very effective in the in the Random Forest model, unfortunately the one-hot encoding seems that lose some information that the XGBoost model tries to compensate with the lag12 variable.

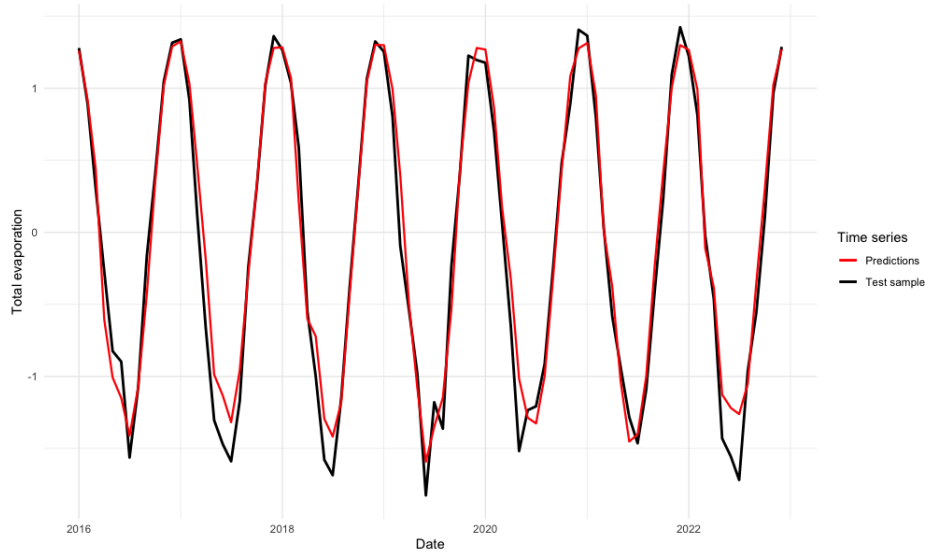


Figure 18: Fitting the predictions of the GBM model to the test sample

With an empirical risk of $\text{MSE}_{\text{GBM}} = 3.21$ the GBM model does better than the XGBoost one but is not as good as the SARIMA and Random Forest ones. Unfortunately we did not found how to measure feature importance with this method. It would have been interesting to see on which feature GBM relies the most compared to XGBoost.

4 Using other variables to predict the water evaporation

In this section we are going to try to predict the water evaporation using the temperature in a first time and then multiple other covariates in a second time, using a SARIMAX model but also comparing it to machine learning models such as Random Forest, XGBoost and GBM.

4.1 Using lag variables of the temperature variable

First of all we are going to use lag variables on the temperature to predict the water evaporation. Those variables are going to play the role of exogenous variable in a SARIMAX model. For the machine learning models those features will simply be added to the previous features.

The exogenous variables do not change the time series. Therefore our ACF and PACF analysis in 3.2.1 stays the same. We consider the following parameters (0, 0, 0)(0, 1, 1). After checking that the residuals are still white noise (normally distributed, a pvalue of 0.66 for the Ljung-Box test), we look at the predictivity of the model.

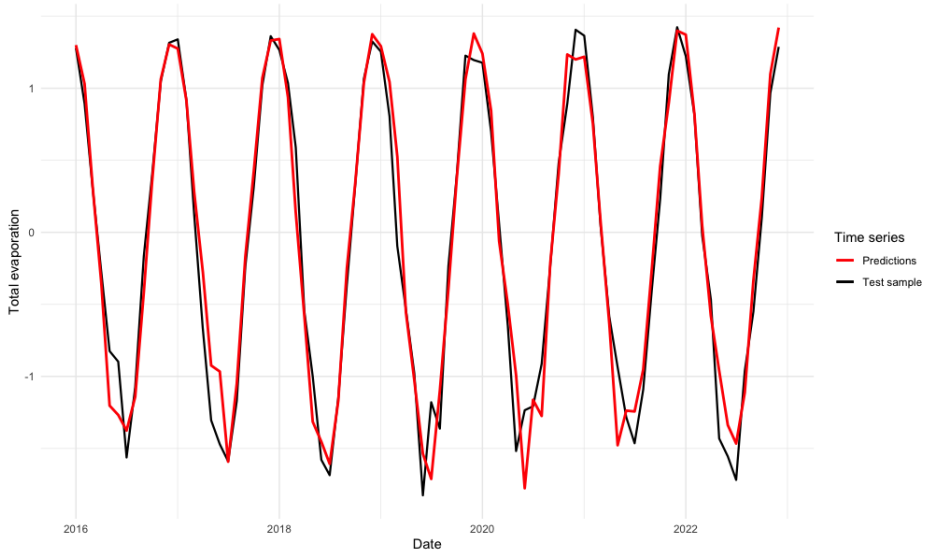


Figure 19: Fitting the predictions of the SARIMAX model with lag variables to the test sample

With an empirical risk of $\text{MSE}_{\text{SARIMAXlag}} = 3.23$ this model is worst than the SARIMA fitted earlier. This is surprising consider the fact that we added information to the model...

Without going into too much depth we found the following empirical risk for machine learning models using the lag of the temperature as additional features:

- Random Forest : $\text{MSE}_{\text{Random Forest lag}} = 2.81$
- XGBoost : $\text{MSE}_{\text{XGBoost lag}} = 3.8$
- GBM : $\text{MSE}_{\text{GBM lag}} = 3.05$

They are better the SARIMAX model, but none of them manages to beat the first SARIMA model. Therefore we can question our use of the lag of the temperature as new features.

4.2 Using our first model to predict the temperature

The idea here is to use our first model on temperature and use it to create a variable of today's predicted temperature and use it as new information. Therefore, rather than using the previous month temperature we can use the predicted temperature for a given month to predict the evaporation of the same month.

This does not work as expected. Our SARIMAX predictions are always above the reality on the test sample. This translates into a poorer empirical risk of $\text{MSE}_{\text{SARIMAXped}} = 11.6$.

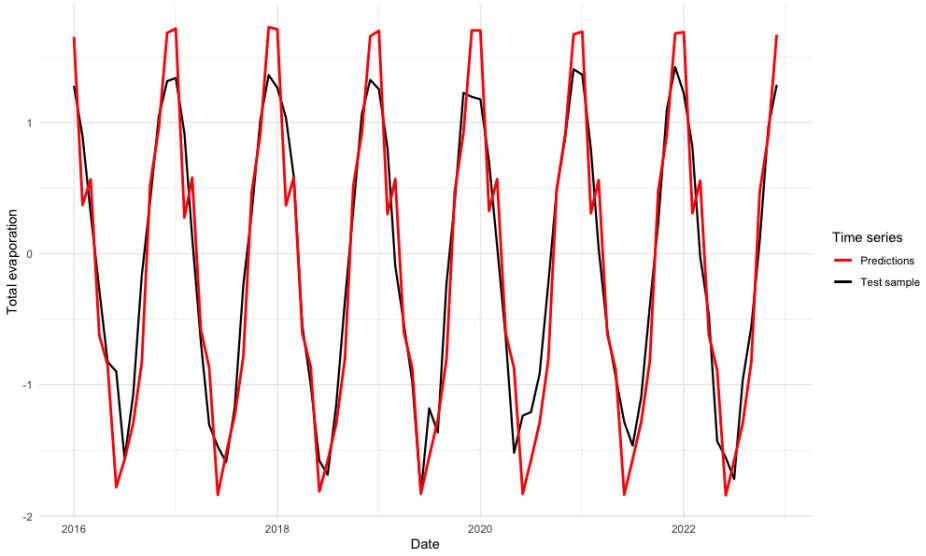


Figure 20: Fitting the predictions of the SARIMAX model with predicted variables to the test sample

The machine learning models are also performing worst with this feature than before.

- Random Forest : $\text{MSE}_{\text{Random Forest pred}} = 8.1$
- XGBoost : $\text{MSE}_{\text{XGBoost pred}} = 8.8$
- GBM : $\text{MSE}_{\text{GBM pred}} = 8.9$

Just like the SARIMAX model, the machine learning models perform poorly with the new variable.

Why is there such a difference in performances while we added information ? Our supposition is that modelling the temperature is very hard. As we can see on the figure x.x our model cannot predict well the peaks of temperature (positive as well as negative). Even thought the SARIMA model on the temperature has a low empirical risk, it is not good enough to be used in another model. Furthermore, our train sample is based from 1984 to 2015, and as we know the climate change is exponential. This is why it has difficulty making accurate predictions for today's temperature.

4.3 Using many variables and their lag

In this section we are going to incorporate many variables into our models through their lags. The idea being that the strength of the wind in the past can influence the evaporation of water lakes. Here is a list of all variables that we are considering.

- Temperature
- Quantity of radiation
- Surface pressure
- water temperature on the surface and deep
- Bottom temperature of lakes
- Total precipitations
- Wind in both directions

For all those variables we consider their lag at 1, 2, 3, 4, 5, 6 and 12 months.

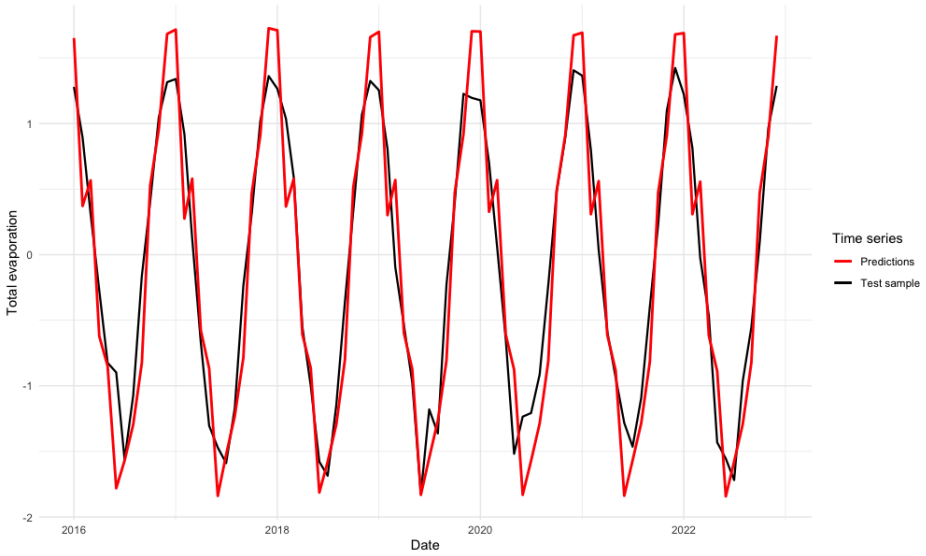


Figure 21: Fitting the predictions of the SARIMAX model with many variables to the test sample

Our SARIMAX model performs well with an empirical risk of $\text{MSE}_{\text{SARIMAXmany}} = 4.35$. But it is still not as good as the simple SARIMA we fitted in section 3.2.1. Can one of the machine learning model outperform our very first model ?

- Random Forest : $\text{MSE}_{\text{Random Forest many}} = 3.0$
- XGBoost : $\text{MSE}_{\text{XGBoost many}} = 3.9$
- GBM : $\text{MSE}_{\text{GBM many}} = 3.4$

With a lot of data, machine learning models performs better than the SARIMAX one, but cannot out perform the simple SARIMA model that we performed at the beginning. Finally to better our models, we could look out for better features rather than "better" models.

5 Conlusion

The project has proven to be a challenging endeavor, primarily due to the inherently complex nature of time series, compounded by variations undoubtedly linked to climate change. Modeling these variations has proven challenging, highlighting increased complexity when attempting to account for the myriad factors influencing temporal data.

Our analysis suggests that simpler models appear to be more effective in this context. We observed that the SARIMA model, despite its relative simplicity (comapred to the other models), outperformed more sophisticated approaches such as SARIMAX and other machine learning models. This observation underscores the significance of simplicity in time series modeling, where complex phenomena can often be better understood by simpler models.

However, despite our successes in modeling, practical hurdles hindered the realization of a real-time monitoring tool. Limitations associated with a non-professional account on Google Earth Engine impeded token generation for connection, rendering real-time monitoring unfeasible.

Notwithstanding these challenges, results obtained with the SARIMA model for predicting surface water evaporation have been highly promising. This model demonstrated an impressive ability to forecast over an extended time scale, offering considerable prospects for long-term prediction.

As a perspective for improvement, the monitoring tool could benefit from expanding its features by incorporating predictions of other variables, as explored with the temperature variable. This expansion could potentially enhance the tool's robustness by providing a more holistic understanding of the factors influencing the studied time series.

References

[1] Paul S.P. Cowpertwait and Andrew V. Metcalfe. *Introductory Time Series with R*. Springer, 2008.

[2] Yves Aragon. *Séries temporelles avec R*. Springer, 2011.