Time Series Analysis: Project

Times Series Analysis *26/11/2019*

Enseignant : M. Wintenberger

ISUP 1ère année de Master Actuariat

Time Series Analysis
Project

Aurélien ROUSSEAU
Corentin BOYEAU
Ramzi FAROUI

Année scolaire 2019-2020

Table des matières

Dataset introduction

1) Description

In this project we use the statistical software R to analyze a time series. The time series that we use were made publicly available by the NYC Taxi & Limousine Commission and consists of 1.1 billion taxi trips from New York between January 2009 to June 2016. The dataset gives us pickups for 30 minutes period.

In our case we use only daily data between 1st April 2013 to 26th June 2016 to solve the problem of missing value and in order to have a number of observations which are a multiple of seven because this time series varies a lot according to the week day. So our dataset is composed of a number of pickups for each day during the period of interest. To obtain daily data we do simply the sum of pickups for a day.

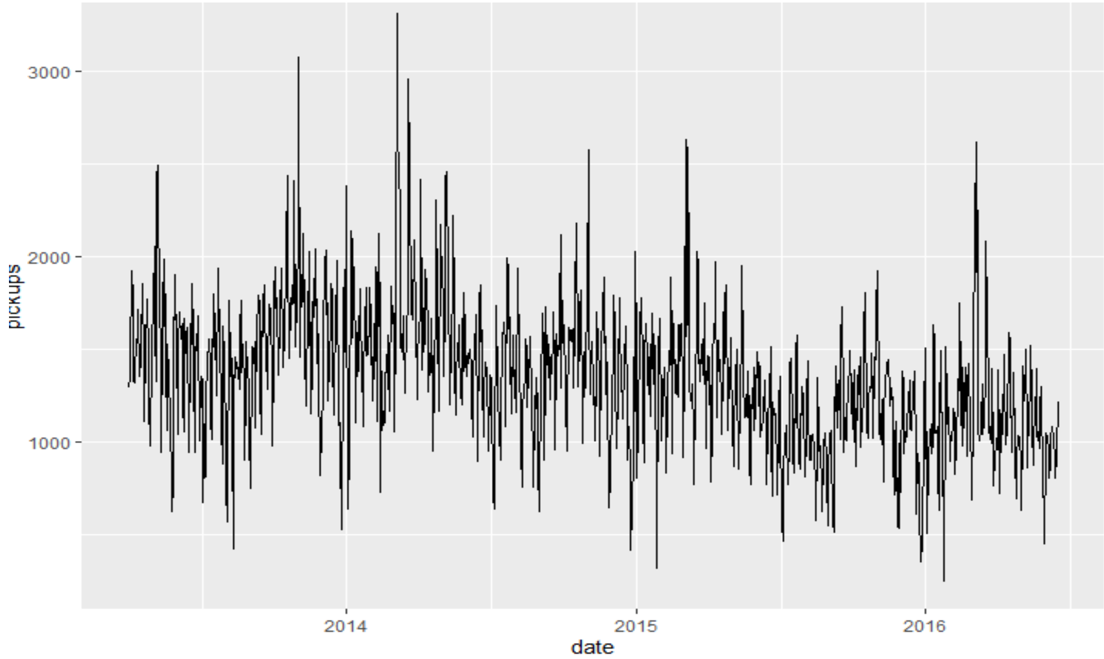
In order to have explanatory variable we use another dataset which contains meteorological variable like for example temperature and precipitation, etc. This weather data for each day was obtained from the National Oceanic and Atmospheric Administration and corresponds to measurements from a weather station located in the Central Park in NYC.

2) Variables

Date : the day of interest
pickups : Number of taxi pickups during the day of interest.
min_temp : minimal temperature during the day of interest.
max_temp : maximal temperature during the day of interest.
wind_speed : average wind speed during the day of interest.
visibility : indicator symbolizing the fog during the day of interest.
pressure : average pressure the day of interest.
precipitation : average precipitation duration the day of interest.
snow_depth : average snow depth on the ground the day of interest.

I) Preprocessing

At the start, without modification, we have the following time series:

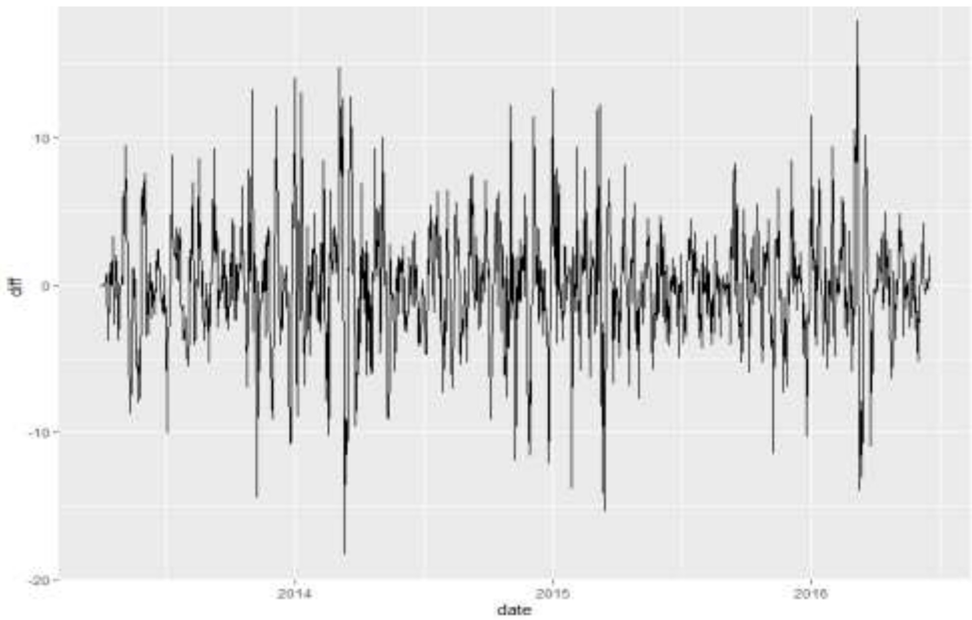


Now we want to remove trend and seasonality of this time series. We observe that the most significant periodic variations correspond to variations according to the week day then we chose a step of seven to stationarize. Then we implement the follows R code:

```
39 data_all <- diff(data, lag=7)/100
```

Which corresponds to: $X_t = \frac{(D_t - D_{t-7})}{100}$

Indeed we have an weekly seasonality and we divide by 100 to increase our results visibility. So We have the following stationarized time series :



II) Model Fitting on the Time Series

In this part, different models such as AR, MA or ARMA models are tested to fit on our time series.

We decompose our dataset in two parts: one for testing (the 10 last observations) and the other for training, with the aim to evaluate our predictions.

Let (X_t) be a centred second order stationary process. We define the following functions, for any h ∈ Z:

- the autocovariance function: $\gamma_{X(h)} = Cov(X_t, X_{th}) = Cov(X_0, X_h) = E[X_0 X_h]$
- the autocorrelation function (ACF): $\rho_X(h) = \rho(X_t, X_{t+h}) = \gamma_X(h)\gamma_X(0)$
- the partial autocorrelation function (PACF):

$$\rho'_X(h) = \rho_X(X_0 - \Pi_{h-1}(X_0), X_h - \Pi_{h-1}(X_h))$$

(with the convention $\Pi_0(X_1) = 0$) where $\Pi_{h-1}(X_0)$ is the projection of X_0 on the linear span of $(X_1, \ldots, X_{h-1}))$

1) Moving Average (MA)

The moving average is the simplest sparse representation of the infinite series in the causal representation $X_t = \sum_{j\geq 0} \psi j Z_t - j$ consisting in assuming $\psi_j = 0 \; for \; j \geq q$.

A MA(q) process, with q ∈ N, is a solution to the equation:

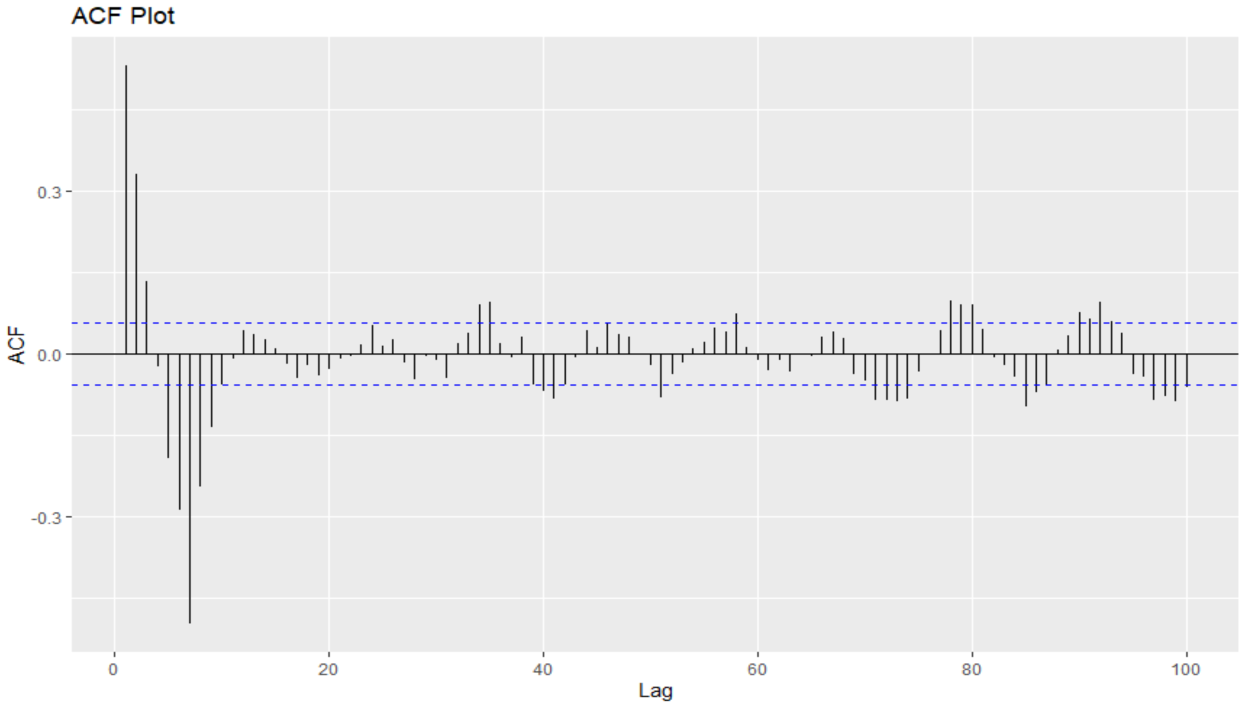
$$X_t = Z_t + \gamma 1 Z_t - 1 + \ldots + \gamma q Z_t - q$$

With t ∈ Z and (Z_t) a white noise.

We need to find the parameter q, for that we use the following property:

If (X_t) is a MA(q) time series, we have $\gamma_X(h) = 0$ for all h ≥ q.

In practice, we use the ACF plot where the order q is corresponding to the last component, which is significantly non-null, i.e. outside the blue confident band.



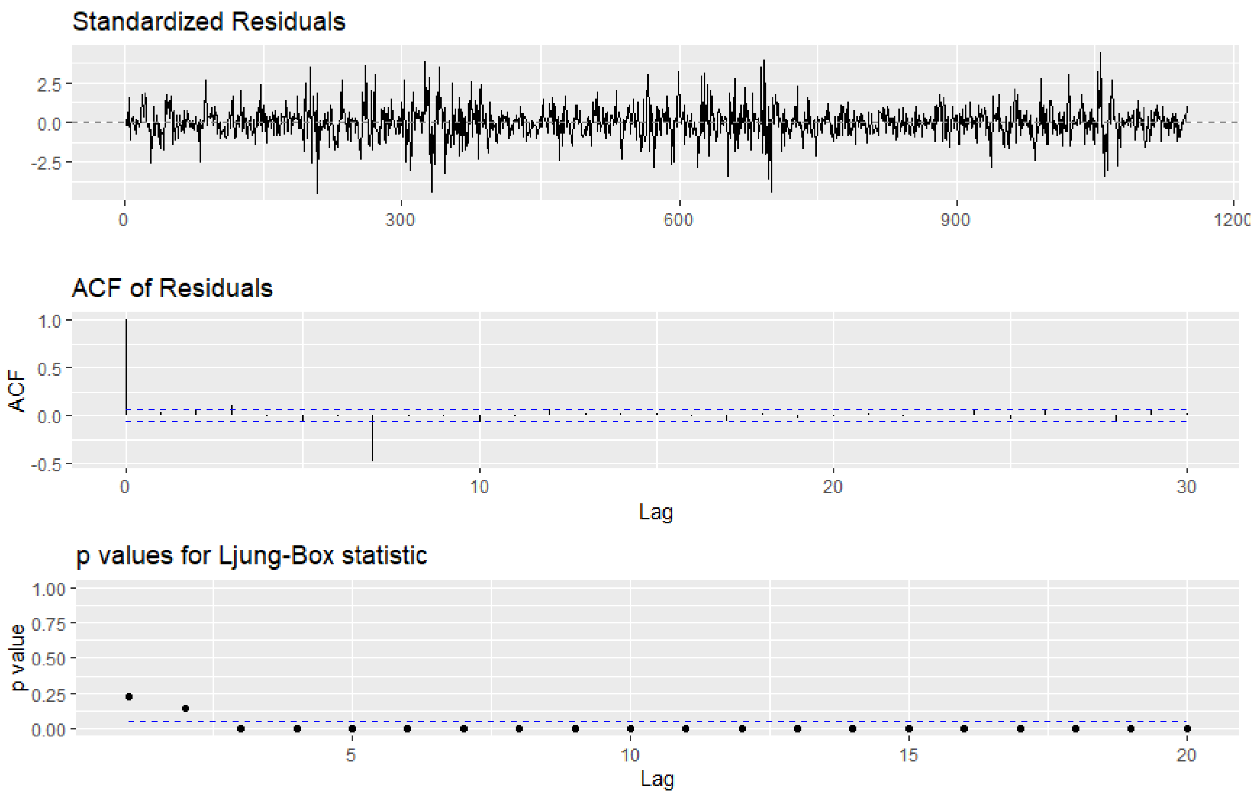
We can see that the last significant value is at lag 9, so the order is 9. We choose a model MA(9).

```
Call:
arima(x = data, order = c(0, 0, 9), include.mean = FALSE)

Coefficients:
         ma1     ma2     ma3     ma4     ma5     ma6      ma7      ma8      ma9
      0.5327  0.3256  0.1024  0.1105  0.1146  0.1110  -0.8799  -0.4123  -0.2227
s.e.  0.0290  0.0291  0.0217  0.0229  0.0235  0.0223   0.0223   0.0317   0.0268

sigma^2 estimated as 6.33:  log likelihood = -2742.78,  aic = 5505.55
```

We have finally:

$$X_t = Z_t + 0.5327.Z_{t-1} + 0.3256.Z_{t-2} + 0.1024.Z_{t-3} + 0.1105.Z_{t-4} + 0.1146.Z_{t-5} + 0.1110.Z_{t-6} - 0.8799.Z_{t-7} - 0.4123.Z_{t-8} - 0.2227.Z_{t-9}$$

Now that we have the model, we need to know if it fit to the time Serie and check if the residuals are a white noise.



We can see on the ACF of residuals a lag at 7. So, we need to check with Ljung-Box test that is a test of autocorrelation in which it verifies whether the autocorrelations of the time series are different from 0(hypothesis H0). In other words, if the result rejects the hypothesis, this means the data is independent and uncorrelated; otherwise, there remains serial correlation in the series.

On our graph, we have only two p-values >5%, so we can reject the hypothesis H0.

In conclusion, the MA(9) model is not appropriate.

2) Auto Regressive (AR)

An AR(p) process, with p ∈ N, is a solution of the equation:

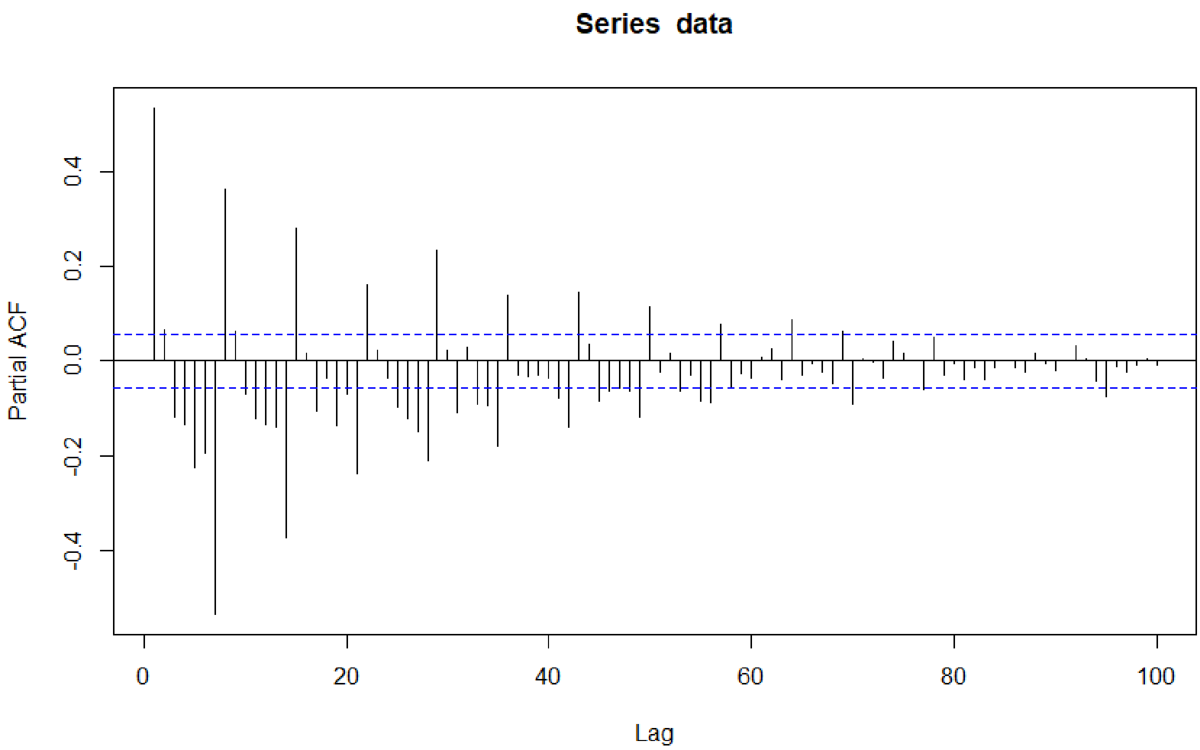
$$X_t = \varphi_1 X_{t-1} + \ldots + \varphi_p X_{t-p} + Z_t$$

with t ∈ Z and (Z_t) a white noise.

To find the order of this model we use the PACF, because the ACF is not relevant in AR models.

The partial autocorrelations are used to determine graphically the order of an AR(p) model. We also have the property for auto-regressive model:

The PACF of an AR(p) time series satisfies: $\rho' X(h) = 0 \ \forall h > p$.



We can see that according to the PACF, the order of the AR model is likely 29, but in order to have a time of compute reasonable, we decide to take p = 8.

```
Call:
arima(x = data, order = c(8, 0, 0), include.mean = FALSE)

Coefficients:
         ar1     ar2     ar3      ar4      ar5     ar6      ar7     ar8
      0.5010  0.0682  0.0012  -0.0096  -0.0705  0.0069  -0.4669  0.2792
s.e.  0.0281  0.0285  0.0286   0.0285   0.0285  0.0286   0.0285  0.0280

sigma^2 estimated as 8.59:  log likelihood = -2909.39,  aic = 5836.78
```

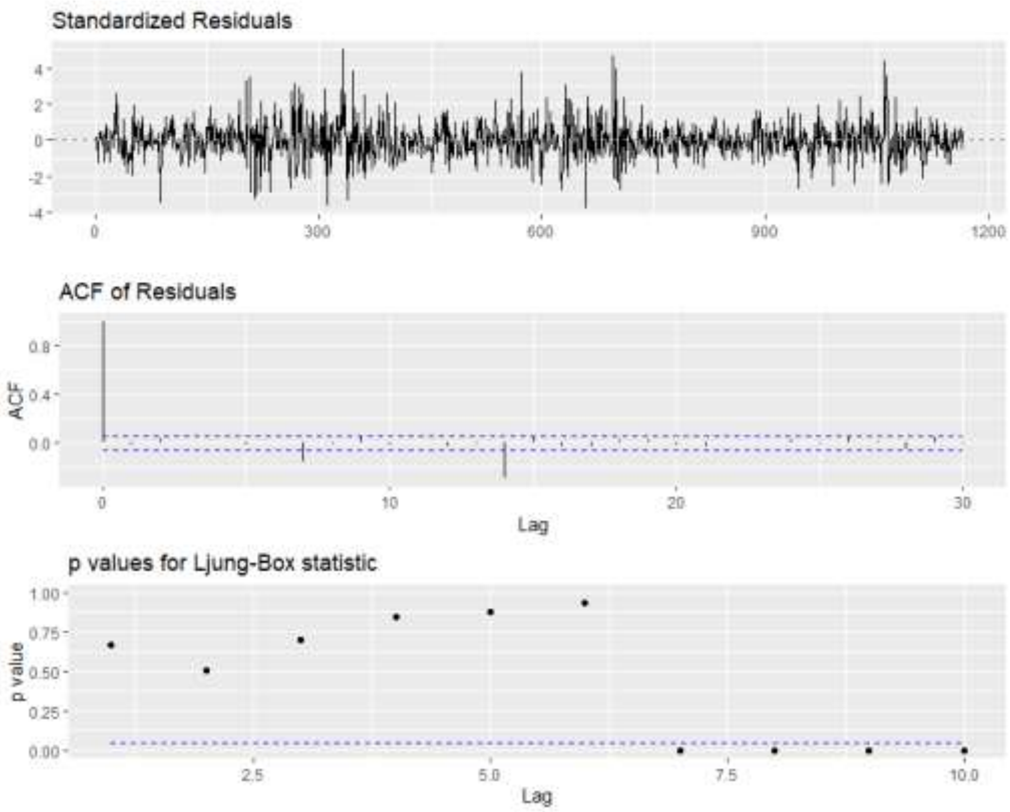
We have finally:

$$X_t = 0.5010X_{t-1} + 0.0682X_{t-2} + 0.0012X_{t-3} - 0.0096X_{t-4} - 0.0705X_{t-5} + 0.0069X_{t-6} - 0.4669X_{t-7} + 0.2792X_{t-8}$$

Which can be approximated:

$$X_t = 0.5010X_{t-1} - 0.4669X_{t-7} + 0.2792X_{t-8}$$

Now that we have the model, we need to know if it fit to the time Serie and check if the residuals are a white noise.



We can see on the ACF of residuals a lag at 7 and 14. But thanks to the Ljung-Box test, we can see that the P-values are in majority ≥ 5%. In conclusion, we cannot reject the hypothesis H0, so the AR(8) model is appropriate.

3) Auto Regressive Moving Average (ARMA)

An ARMA(p,q) time series is a process solution of the model :

$$X_t = \varphi_1 X_{t-1} + \ldots + \varphi_p X_{t-p} + Z_t + \gamma_1 Z_{t-1} + \ldots + \gamma_q Z_{t-q}$$

with $\theta = (\varphi_1, \ldots, \varphi_p, \gamma_1, \ldots, \gamma_q)\ \theta \in R_{p+q}$ the parameters of the model and (Z_t) a white noise.

In order to find the orders of ARMA, we will need to use information criterion that is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, the information criterion estimates the quality of each model, relative to each of the other models. We define one information criterion as penalized log-likelihood L:

The Akaike Information Criterion: $AIC = 2(p+q) - 2L$

The AIC criterion offers an estimate of the relative information lost when a given model is used to represent the process that generated the data. In doing so, it deals with the trade-off between the goodness of fit of the model and the simplicity of the model. The best model is is the one with the smallest AIC.

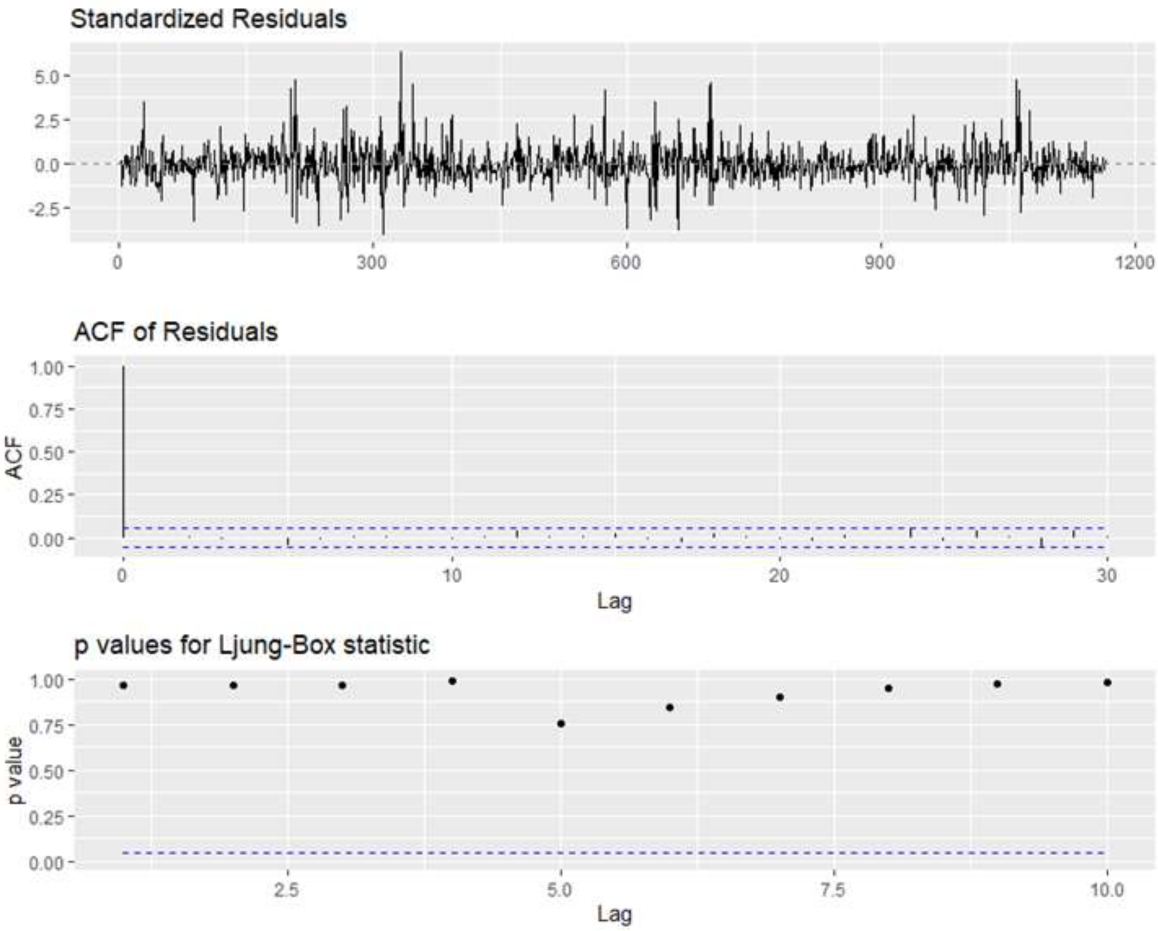
We have the following matrix of the AIC for each p,q from 1 to 9 :

	1	2	3	4	5	6	7	8	9
1	6163.174	6154.917	6155.613	6134.585	6081.980	5658.348	5492.896	5495.669	5488.465
2	6134.114	6059.630	5996.669	6001.598	5947.296	5634.504	5494.450	5491.360	5484.042
3	6062.383	6060.801	5992.525	5929.199	5849.091	5617.737	5487.663	5486.939	5476.049
4	6060.928	6047.803	5926.271	5812.740	5796.736	5614.000	5486.609	5488.057	5473.904
5	6076.544	6026.410	5844.838	5846.218	5794.333	5576.305	5486.705	5487.848	5475.623
6	6053.948	6021.503	6014.981	5794.006	5728.262	5549.101	5485.716	5485.239	5479.295
7	5873.117	5806.973	5834.718	5762.204	5730.944	5542.292	5484.441	5486.440	5477.372
8	5837.089	5834.564	5836.414	5728.174	5717.569	5546.874	5486.439	5487.985	5481.537
9	5832.148	5802.619	5838.448	5718.603	5658.313	5544.769	5487.222	5488.360	5481.800

So the best model is the ARMA(4,9).

```
Call:
arima(x = data, order = c(p_opt, 0, q_opt), include.mean = FALSE)

Coefficients:
         ar1      ar2     ar3     ar4      ma1     ma2     ma3     ma4     ma5     ma6      ma7     ma8      ma9
      1.6137  -1.4046  0.3456  0.0704  -1.0760  0.9228  0.0419  0.0360  0.0219  0.0278  -0.9464  1.0934  -0.8967
s.e.  0.0706   0.1199  0.0732  0.0351   0.0663  0.0717  0.0118  0.0125  0.0131  0.0190   0.0148  0.0569   0.0719

sigma^2 estimated as 6.12:  log likelihood = -2722.95,  aic = 5473.9
```

Now that we have the model, we need to know if it fit to the time Serie and check if the residuals are a white noise.



We can see no significant lag. In addition, thanks to the Ljung-Box test, we can see that the P-values are $\geq 5\%$. Finally, we cannot reject the hypothesis H0, so the ARMA(4,9) model is appropriate.

4) Residuals

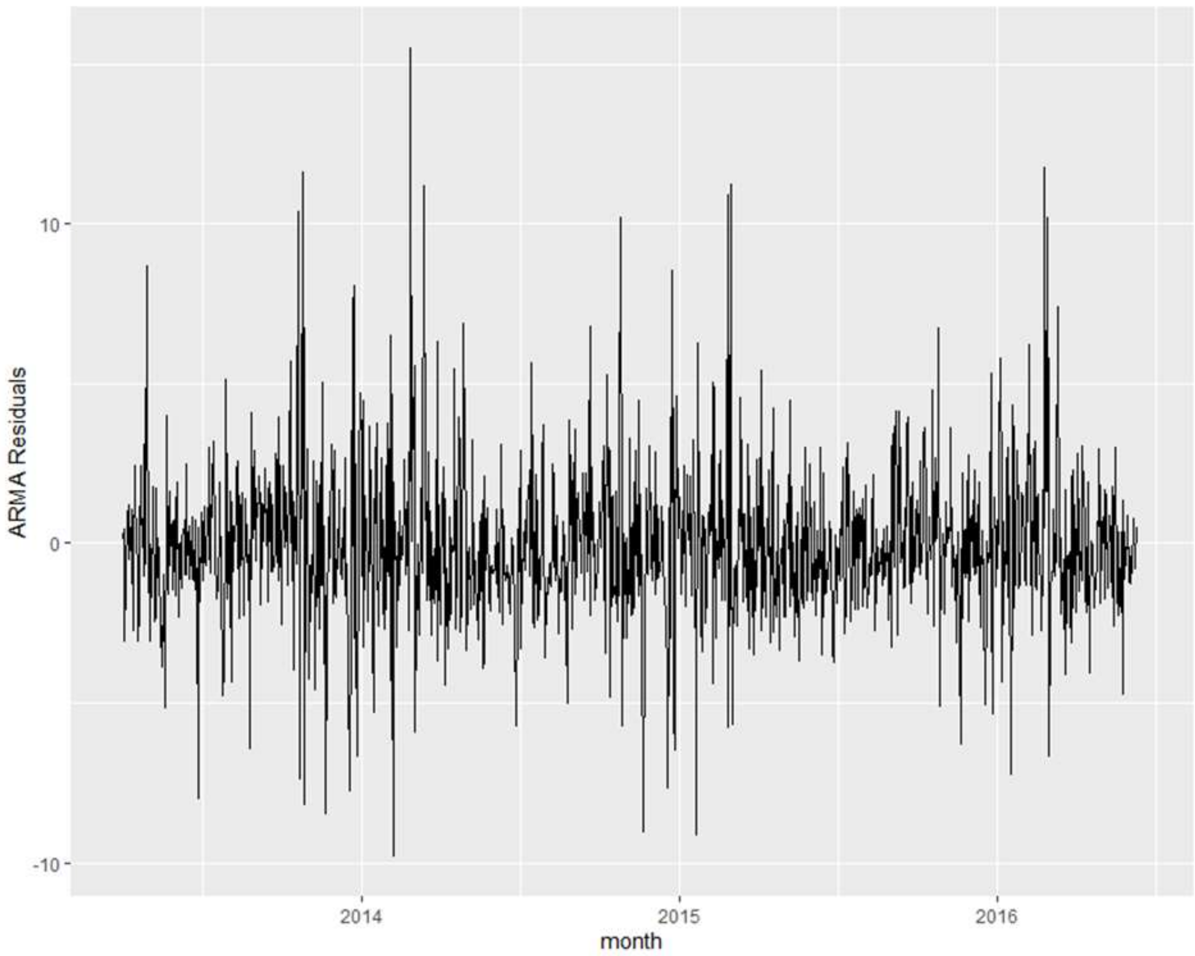
We want to check if the residuals are gaussian. First, we choose the best model thanks to the AIC.

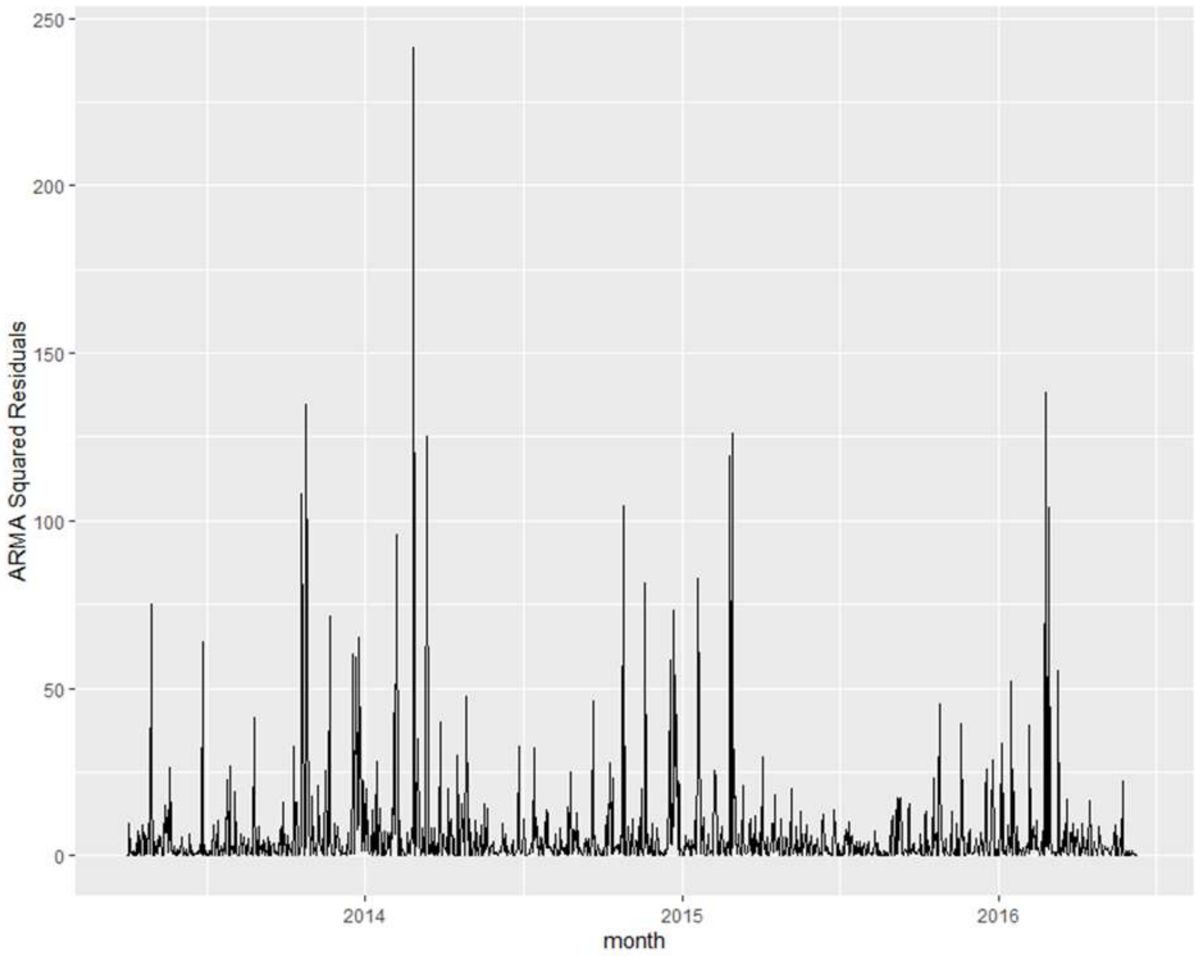
We have:

	df	AIC
fitar	9	5836.778
fitma	10	5505.551
fitarma	14	5473.904

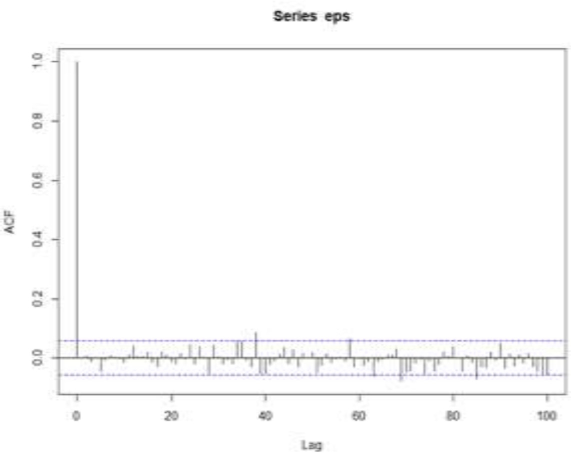
So, we take the ARMA model because it has the smallest AIC value.

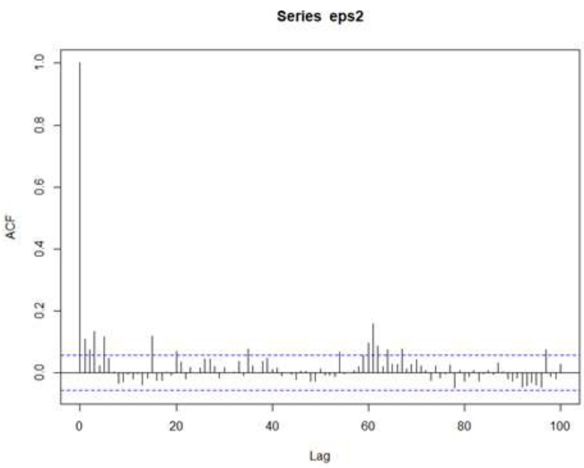
We represent the residuals trough time:





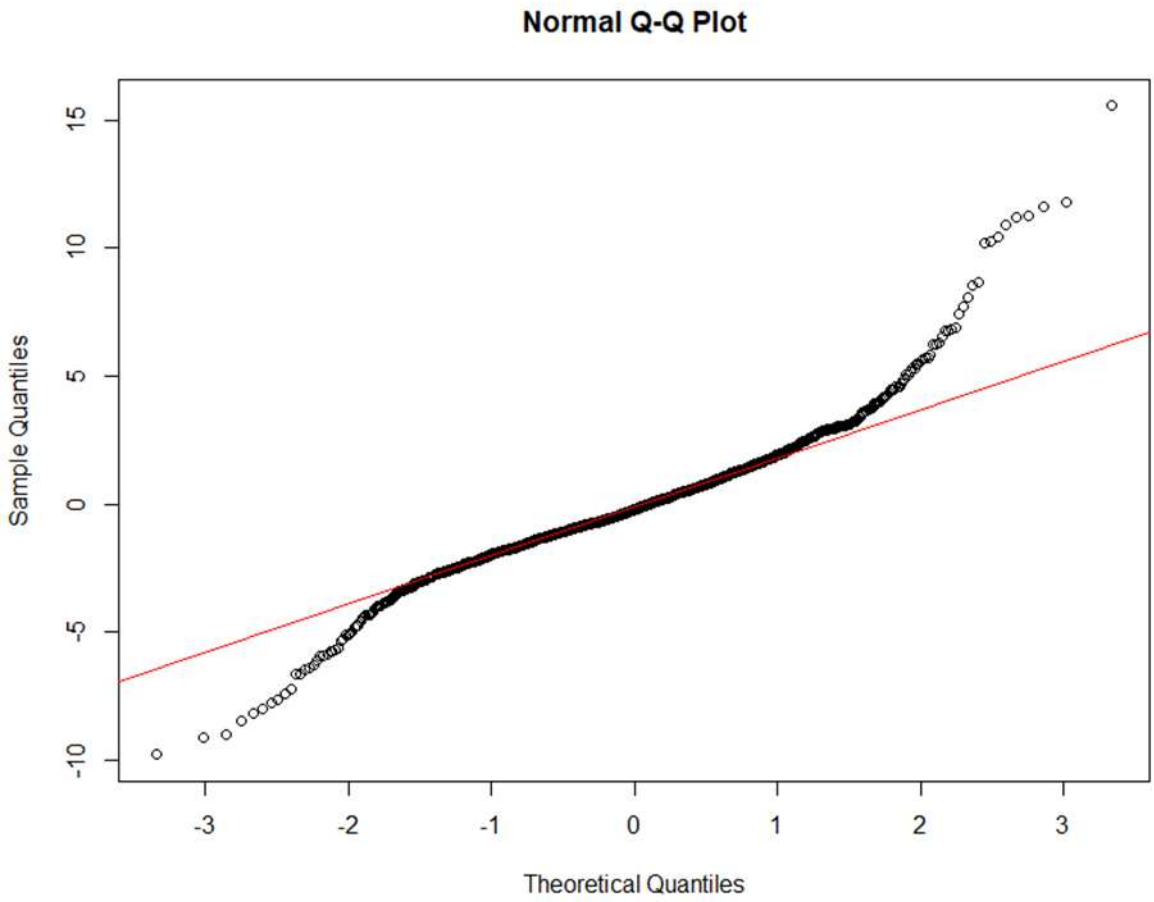
Then, we plot the ACF on the residuals (eps) and the squared residuals (eps2):





There is some autocorrelation left in the residuals (seen in the significant spike in the ACF plot). This suggests that the model can be improved, although it is unlikely to make much difference to the resulting forecasts.

Finally, we can look at the Q-Q plot of the residuals to determine if it is gaussian.



According to the graph, the residuals show too many extreme negatives and positives values. Moreover, we can see that the relation is not linear.

In conclusion, the residuals are not gaussian.

5) Generalized Autoregressive Conditional Heteroskedastic (GARCH)

Even if the ARMA(4,9) model is appropriate, we can see in the residuals some little lags that reflects some volatility. We use ARCH/GARCH models to predict this volatility.

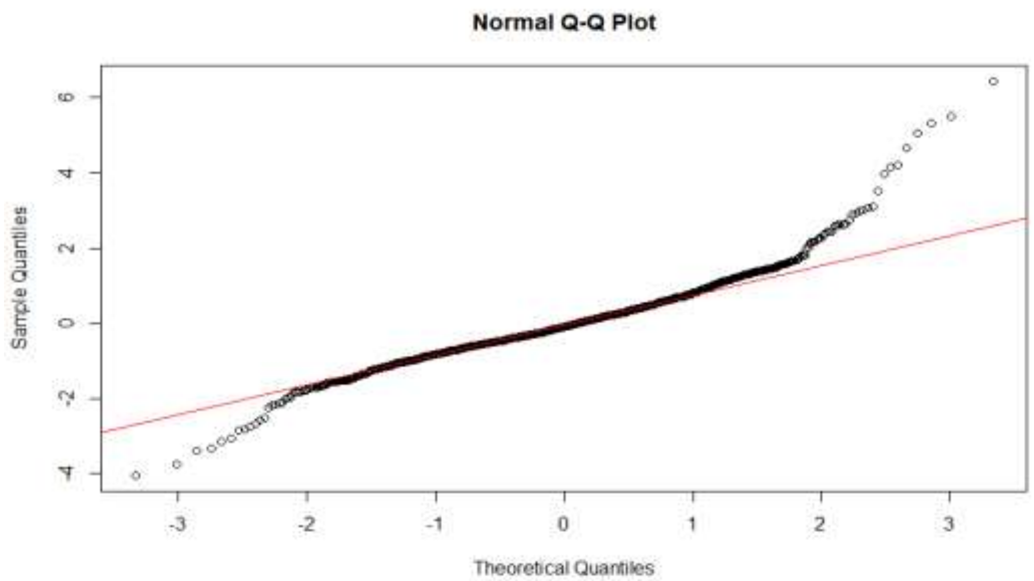
We consider (Z_t) an observed white noise. The GARCH(p,q) model (Generalized Autoregressive Conditional Heteroscedastic) is solutions, if it exists, of the system :

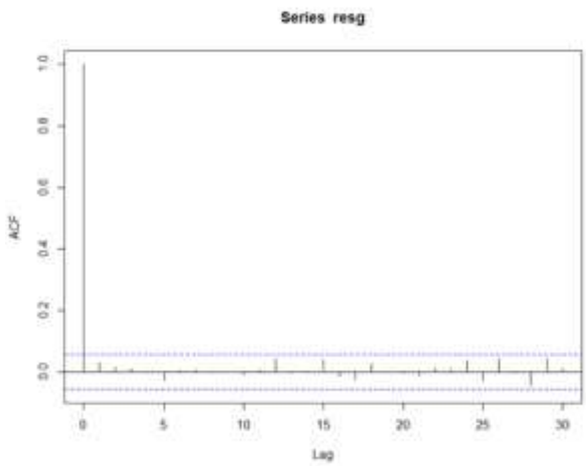
$$Z_t \;=\; \sigma_t W_t \; \sigma_t^2 \;=\; \omega \; + \beta_1 \sigma_{t-1}^2 + \ldots + \beta_p \sigma_{t-p}^2 \; + \alpha_1 Z_{t-1}^2 + \ldots + \alpha_q Z_{t-q}^2$$

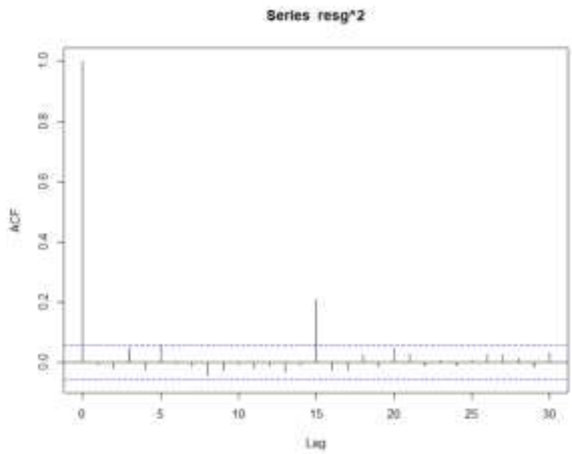
Here, we focus for simplicity on p = q =1 for simplicity i.e. GARCH (1,1).

$$\sigma_t^2 \;=\; \omega \; + \beta_1 \sigma_{t-1}^2 \; + \alpha_1 Z_{t-1}^2$$

Thanks to some function seen in courses, we fit a GARCH (1,1) model on the residuals.

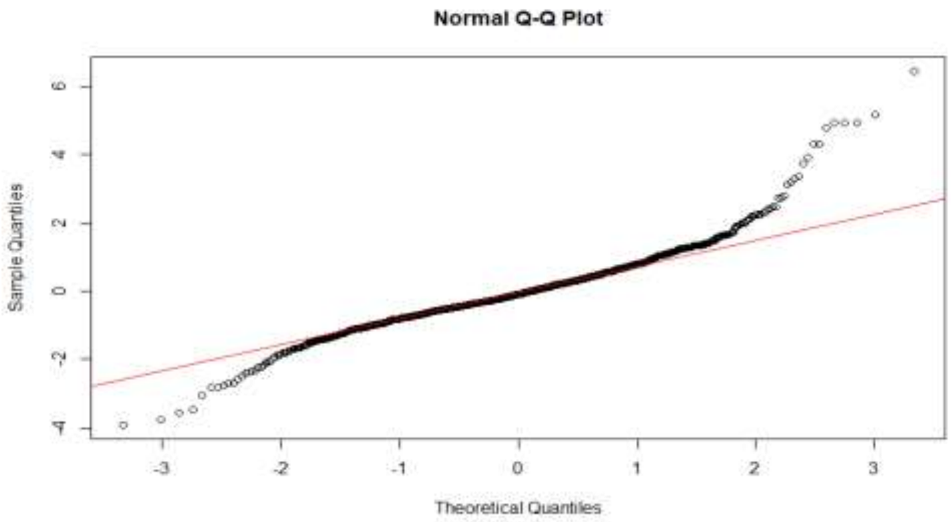


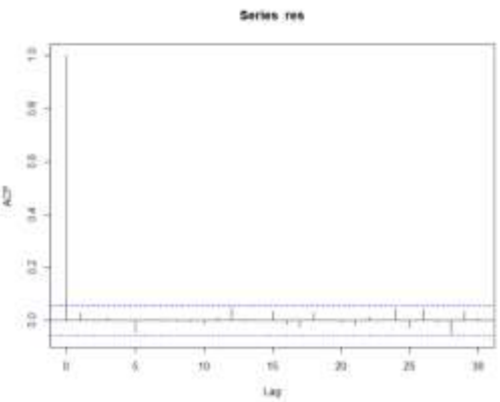


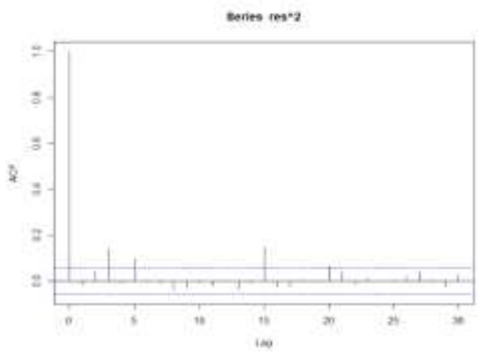


We can see only one significant lag, but the QQ plot shows that the residuals are not gaussian.

In order to see if the GARCH (1,1) model is relevant, we can test the nullity of β. We compute a p-value = 0.925, so we cannot reject the hypothesis that β=0 (at 5%). Then, we test an ARCH (1) (= GARCH (0,1)) model because β=0:





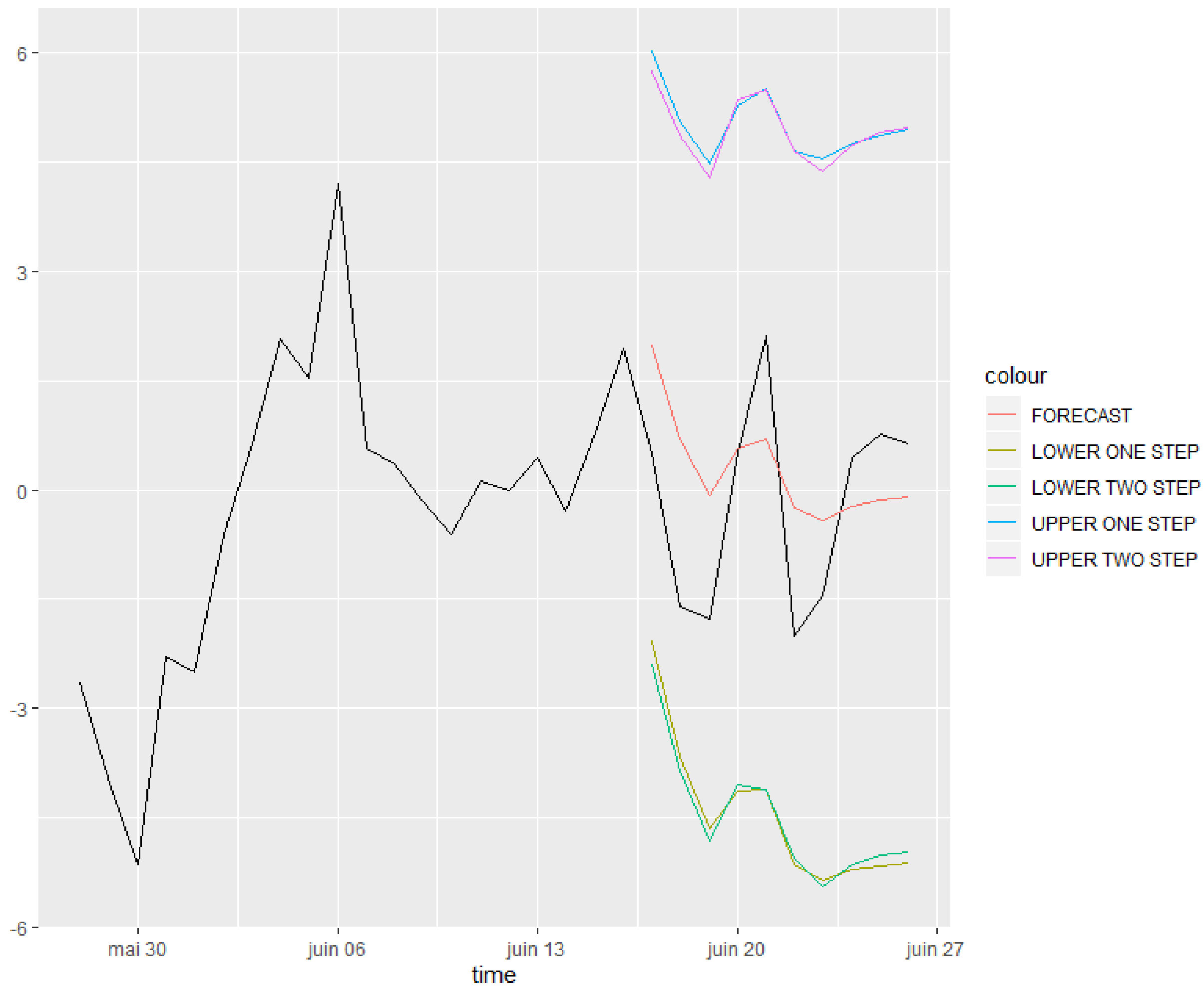


We can see more significant lag compared to the GARCH model and the QQ plot confirms us that the residuals are not gaussian.

So we test the nullity of α : We find a p-value = 0.9476611 so we cannot reject the hypothesis (at 5%).

6) Prediction intervals for the 10 most recent data

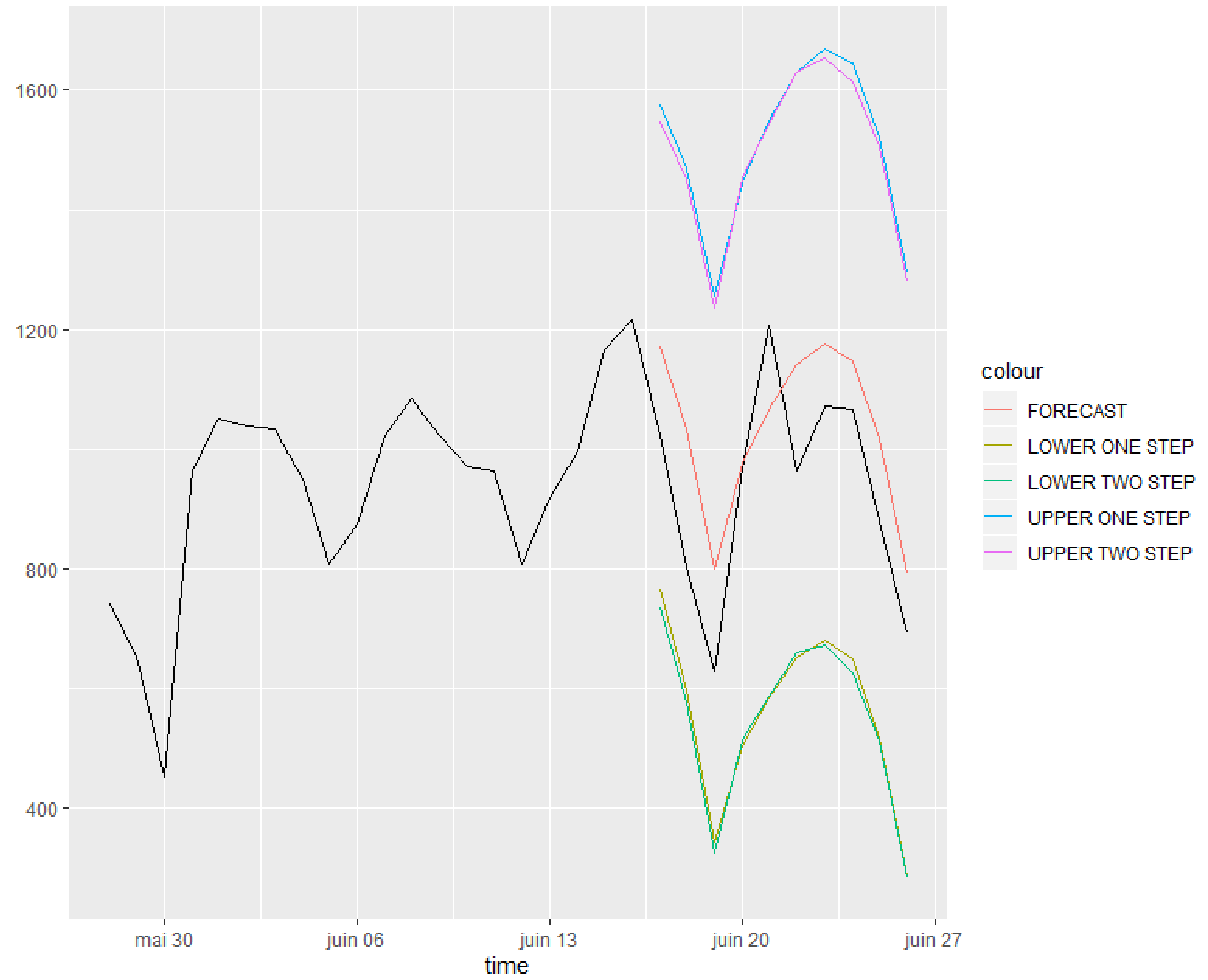
First, we built an interval with a two-step procedure and then we use a one-step estimator given by the rugarch package. We obtain:



However, this is not really what we want. Here, we have a forecast with prediction intervals but for our differenced time series X_t ! The aim of the project is to give a prediction on the "true" time series D_t. To get back to D_t, we just have to do the opposite operation that we did to find X_t :

$$X_t = \frac{D_t - D_{t-7}}{100} \Rightarrow D_t = 100 * X_t + D_{t-7}$$

Finally, we have:



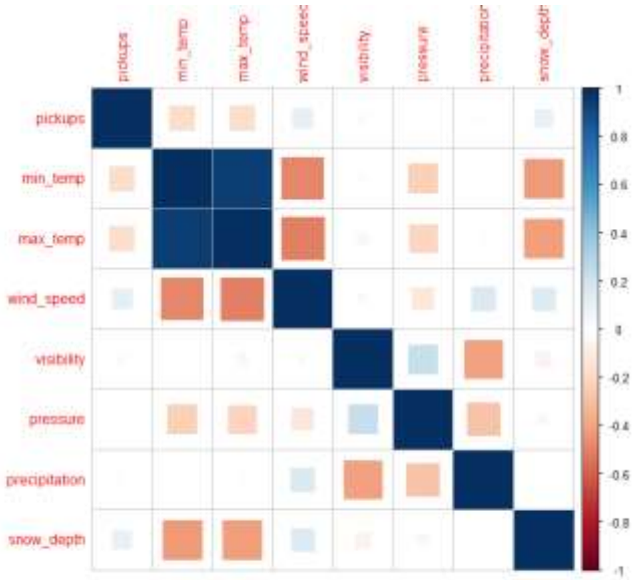
The predictions on the 10 most recent data seem to be quite good and follow the evolution of the test data.

III) Training on the times series of interest using explanatory times series

1) Preprocessing

a) Variables choice

Firstly, we should observe explicative variable and the correlation between them and also with the pickups number which are our interest variable. Then we analyze the following correlation matrix:



We can see that minimum and maximum temperature are very correlated as we could anticipate. Then we don't keep max_temp in our variable set in order to reduce the number of variable.

Furthermore we have that the pickups number are very few correlated with visibility, pressure and precipitation. However we do the choice to keep only visibility because the using of this variable allow us to improve our prediction significatively. Then we delete pressure and precipitation in order to reduce the computing time.

Finally we observe that the number of records is positively correlated with bad weather conditions like low temperature, strong wind or presence of snow.

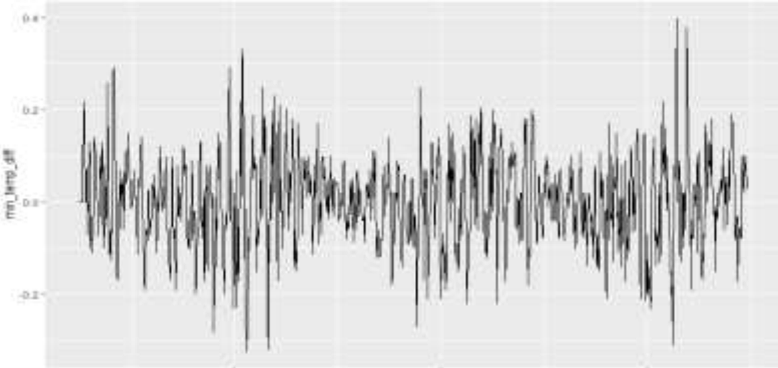
b) Remove trend and seasonality

In the same way that we did in Part I, we stationarize time series with a step of seven which corresponds to a weekly seasonality. We have the following formula and we apply this to our six explicative variables:

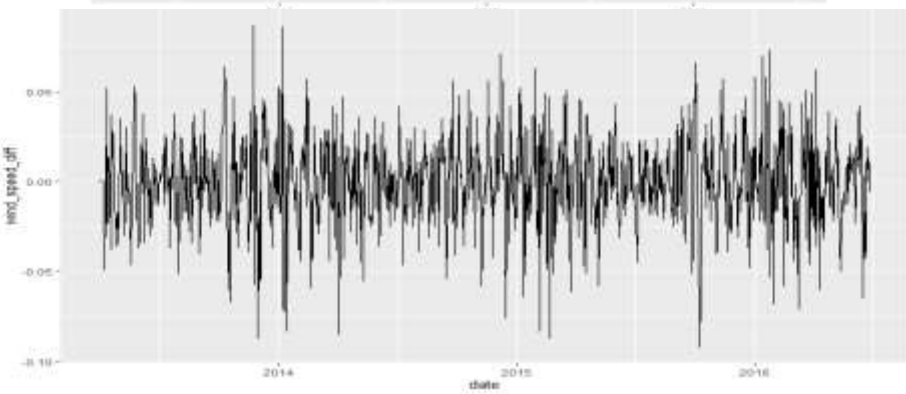
$$X_t = \frac{(D_t - D_{t-7})}{100}$$

We obtain the following plot which corresponds to time series of explicative variable without seasonality:

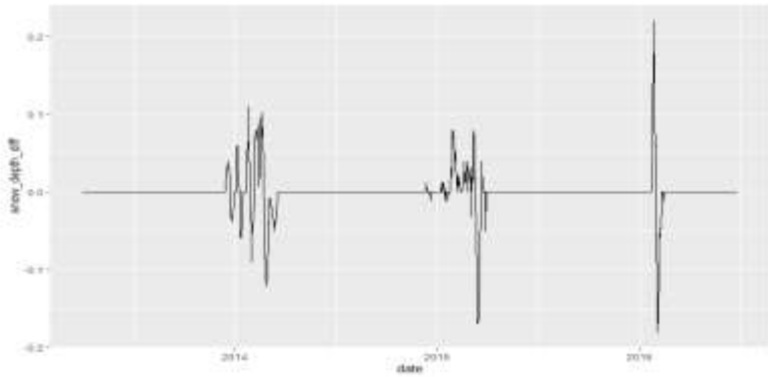
Min temprature



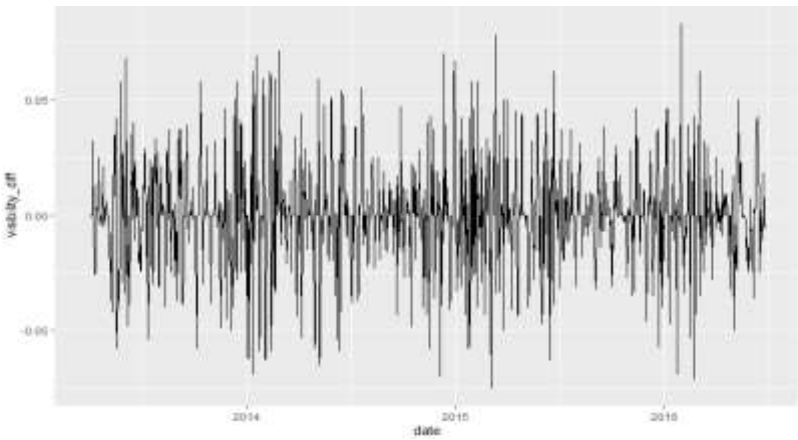
Wind speed



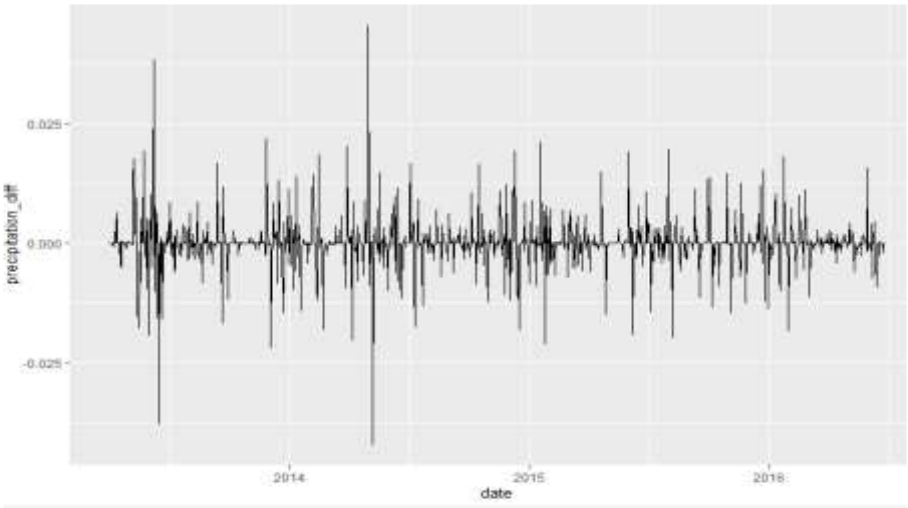
Snow depth



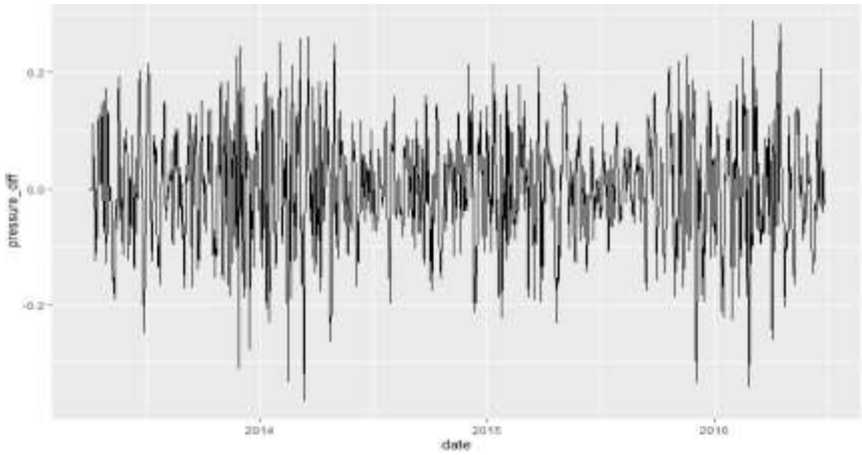
Visibilty



Precipitation



Pressure



We don't have time series which are well stationarize, indeed we use the same process as in Part 1 whereas weather data are not relative to a weekly seasonality.

2) Time varying coefficients

We construct the following dynamic model :

```
n<-length(all_taxi$pickups)
y<- ts(all_taxi$pickups, frequency=7, start=1)
ytraining <-ts(y[-((n-9):n)],frequency=7, start=1)

Y<-ts(cbind(ytraining,lag(ytraining,1), lag(ytraining,7),lag(ytraining,8),lag(ytraining,9),lag(ytraining,10),
    lag(ytraining,11),lag(ytraining,12),lag(ytraining,13),lag(ytraining,14),lag(ytraining,15),lag(ytraining,16),
    lag(ytraining,17)))

Y <- Y[-c(1:17,(length(Y[,1])-6):length(Y)),]

model <- SSModel (Y[,-c(1,2,3)] ~-1+SSMregression(~ min_temp[-c(1:7,(n-9):n)] + wind_speed[-c(1:7,(n-9):n)] +
    snow_depth[-c(1:7,(n-9):n)]+ visibility[-c(1:7,(n-9):n)]+ Y[,3]+Y[,2]+Y[,1],
    Q=diag(NA,7),R=t(matrix(rep(diag(1,7),10),nrow=7)), H = diag(1,10))


fit <- fitSSM(model, inits = c(0.1,0.1,0.1,0.1,0.1,0.1,0.1), method = "BFGS")

model <- fit$model
```

We had in part II an AR model of order 8, then we train the model on the eight previous values of Y. However we observe that for the 2nd to the 6th previous values of Y the information given to the prediction is very low. Because coefficients corresponding to this variables are closer to zero in the AR(8) model in part II.

So to predict Y we use four explicative variables (snow depth, minimum temperature, visibility and wind speed) and three previous value of the interest variable (Y_{t-1}, Y_{t-7} and Y_{t-8}). This reduction allows us to reduce the computing time.

Finally, this model predicts the interest variable, here pickups, according to our four explicative variables and the precedent value of the interest variable, here the number of pickups at three different previous day.

Furthermore we fit the model with the BGFS method. It's an optimization algorithm that approximates the Boyden-Fletcher-Goldfarb-Shanno algorithm in a fast way.

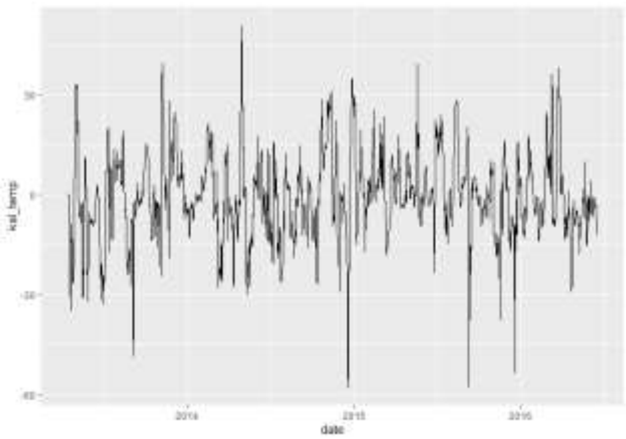
3) QLIK

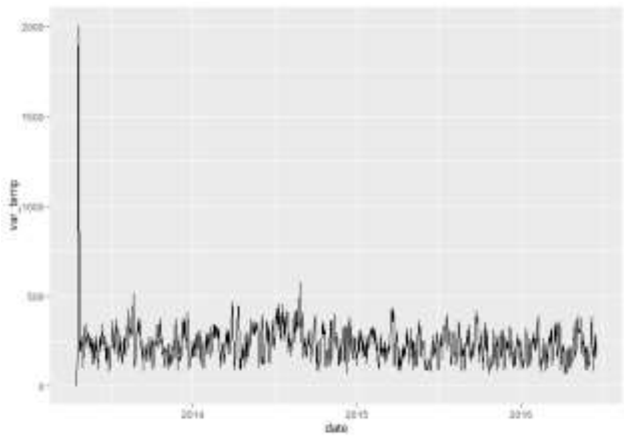
Now we use the KFAS package in order to tune the hyperparameters and we obtain the matrix of disturbance:

```
         [,1]     [,2]     [,3]     [,4]       [,5]        [,6]       [,7]
[1,] 60.19352   0.0000   0.0000   0.0000 0.00000000 0.000000000 0.00000000
[2,]  0.00000 340.9083   0.0000   0.0000 0.00000000 0.000000000 0.00000000
[3,]  0.00000   0.0000 156.9507   0.0000 0.00000000 0.000000000 0.00000000
[4,]  0.00000   0.0000   0.0000 108.0397 0.00000000 0.000000000 0.00000000
[5,]  0.00000   0.0000   0.0000   0.0000 0.01314659 0.000000000 0.00000000
[6,]  0.00000   0.0000   0.0000   0.0000 0.00000000 0.008955509 0.00000000
[7,]  0.00000   0.0000   0.0000   0.0000 0.00000000 0.000000000 0.02075429
```

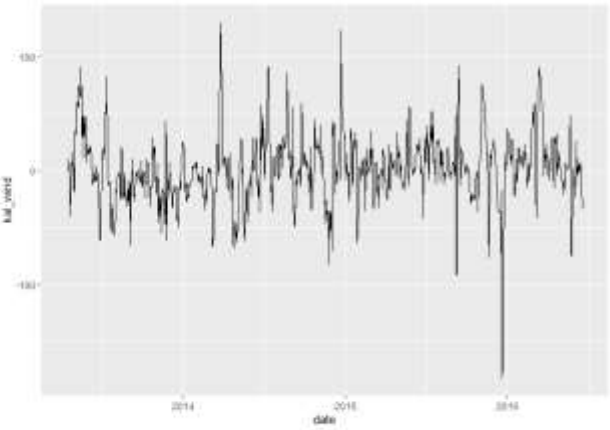
Now we plot the parameters value for the one step predictions with the KFAS command (at the left) and the associated variance (at the right) :

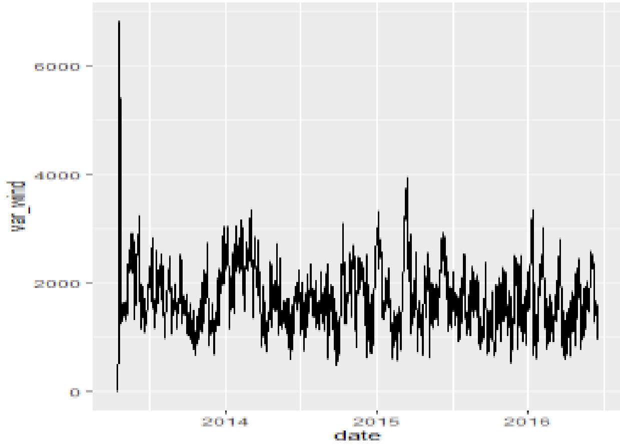
Minimum temperature



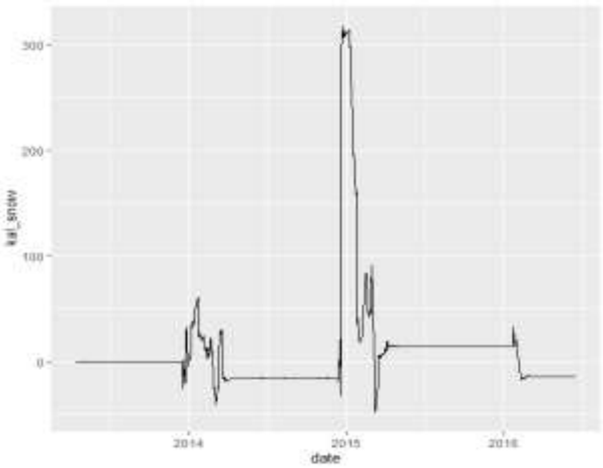


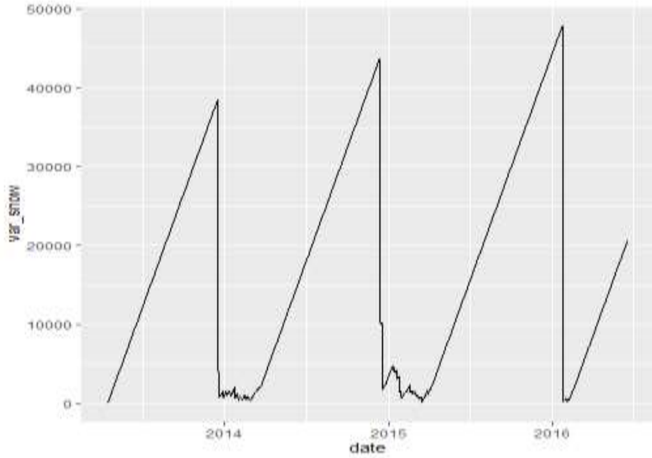
Wind speed



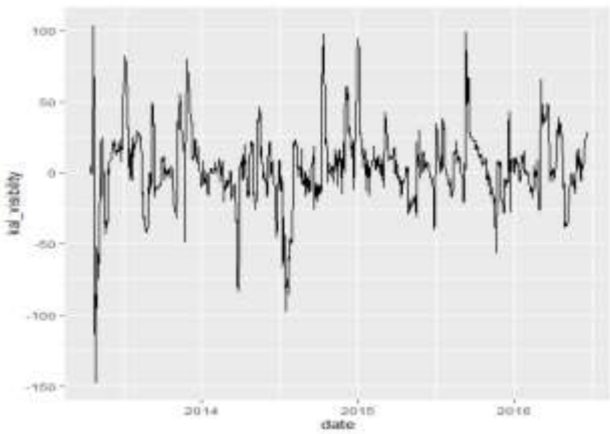


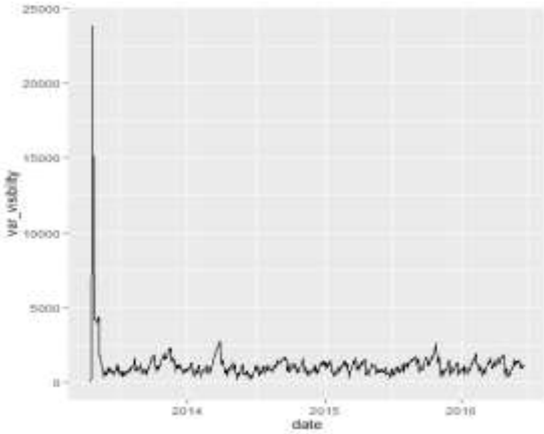
Snow depth



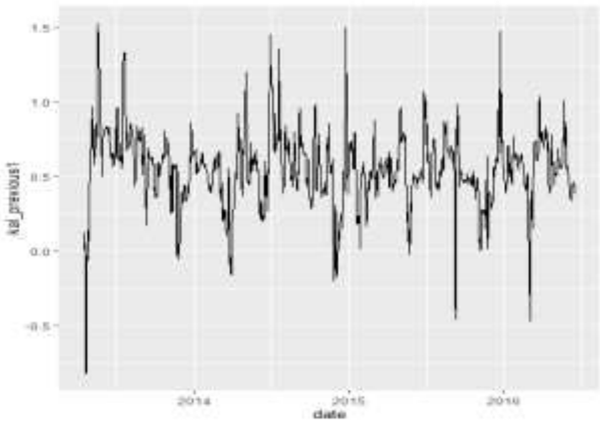


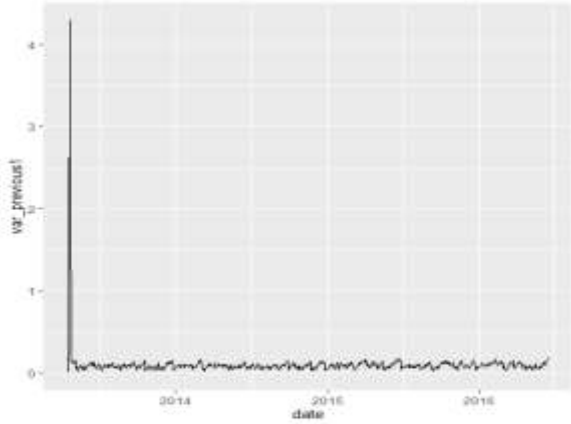
Visibility



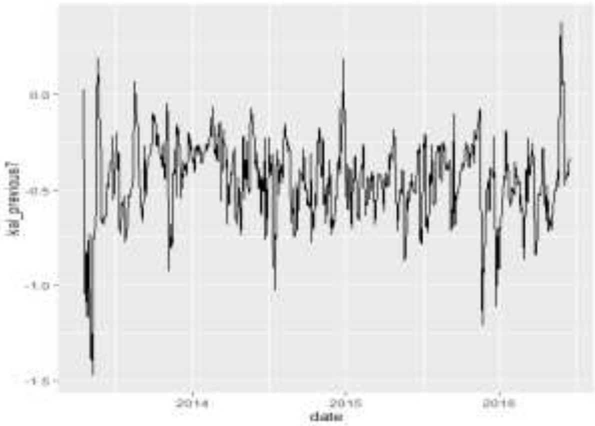


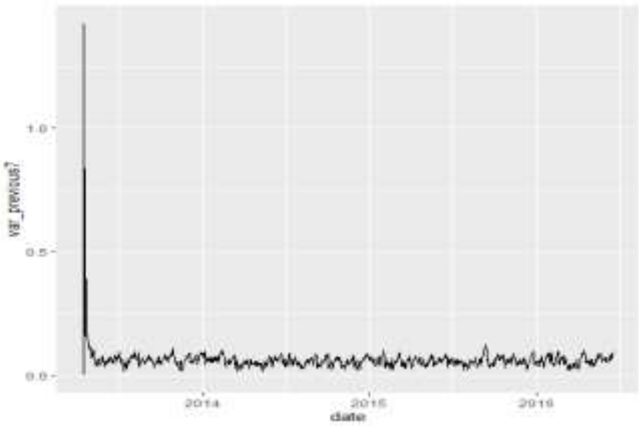
$\underline{Y_{t-1}}$



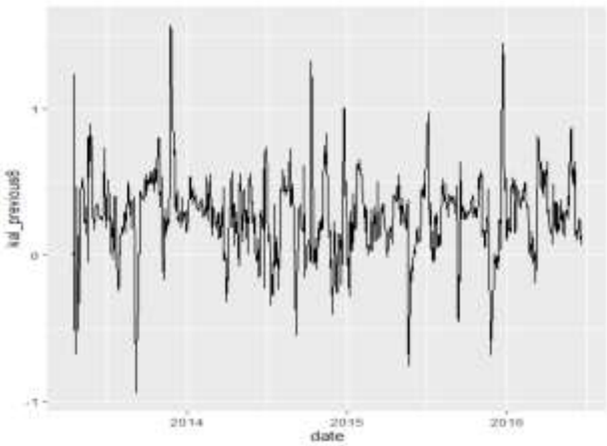


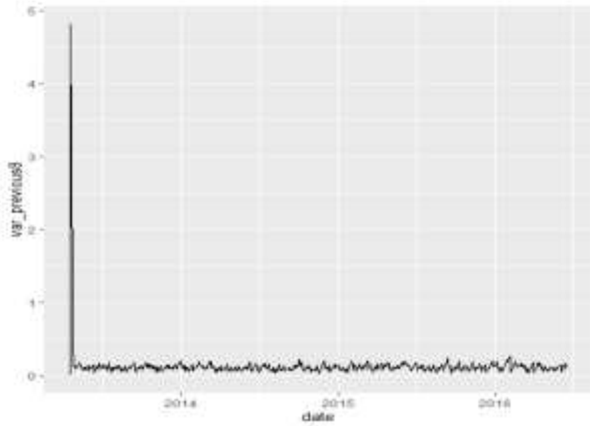
$\underline{Y_{t-7}}$





$\underline{Y_{t-8}}$



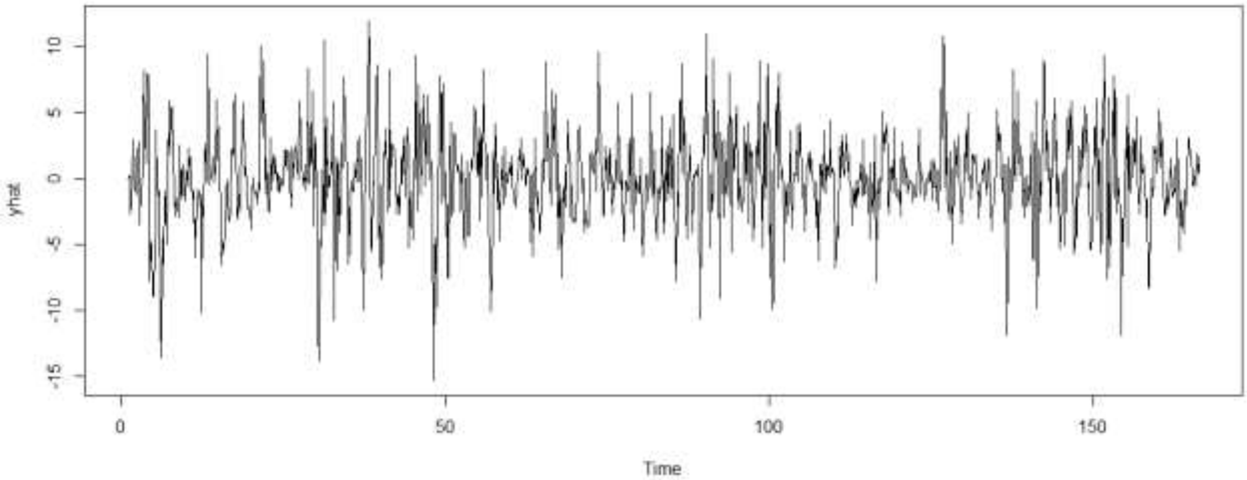


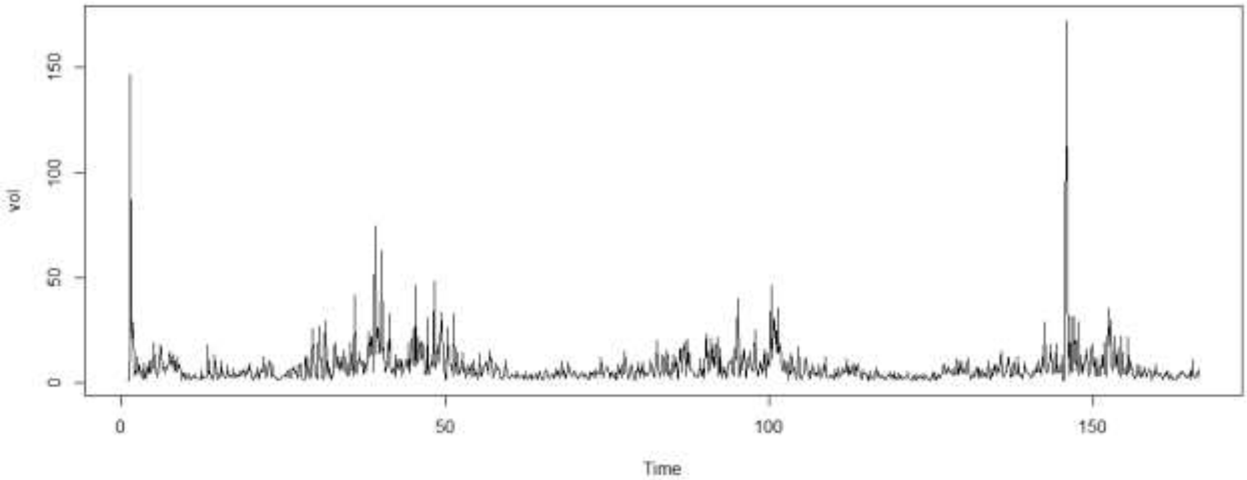
We have for the variables majority that there is a high volatility only for the first observations and decreases quickly to 0, indeed in this case we don't have enough data then it's the riskiest period. Moreover for the snow coefficient the volatility is high for some period because this variable is very uncertain in winter, it's also the case for wind speed which varying a lot for certain period of the year.

Now we plot the prediction sample and also the conditional variance of the one-step prediction with the following code:

```
yhat<-ts(kal$m[,1],frequency =7, start=1)
ts.plot(yhat)

vol<-ts(kal$F[1,],frequency = 7, start = 1)
ts.plot(vol)
```

We write frequency equal to 7 because it's the step of our seasonality. Then we have 165 period in the prediction.

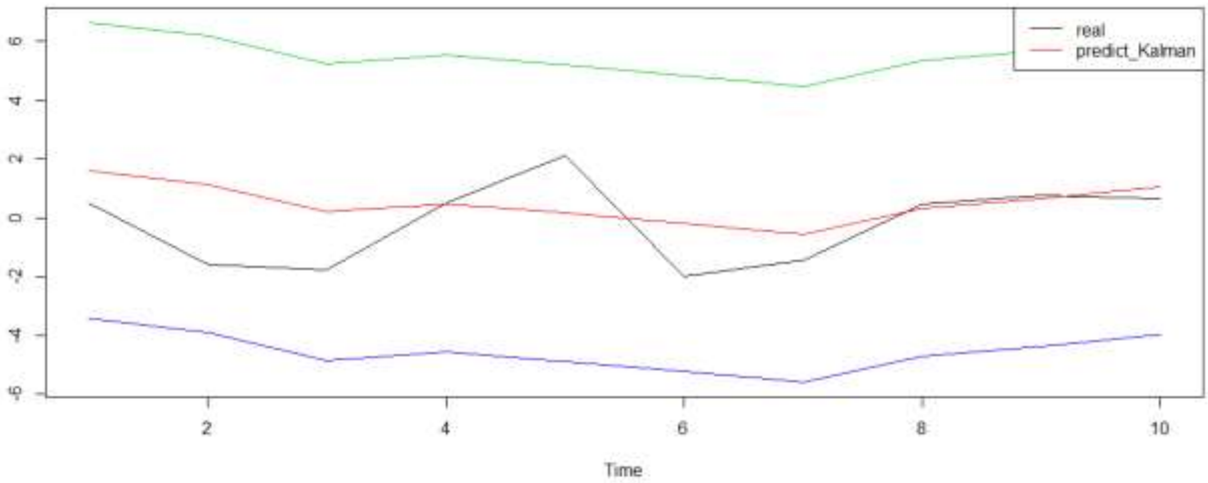




There is a high volatility effect due to the time varying coefficients. At the begining the high volatility is due to the uncertainty prediction (yhat) due to the lack of observations.

4) Prediction

We use the Kalman's recursion on the tuned dynamical model to produce an interval of prediction for the ten most recent data:

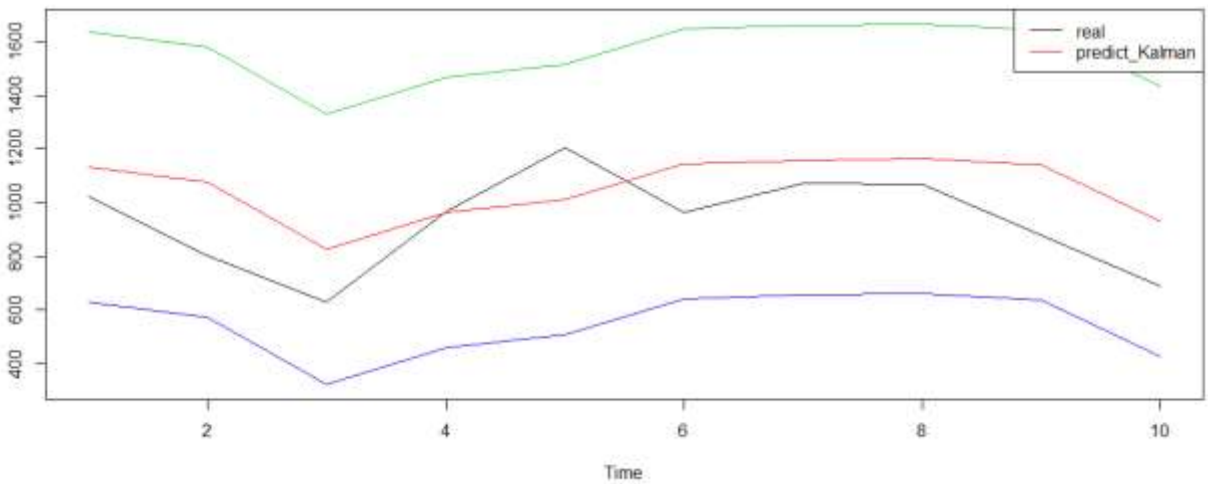


We obtain a real observations which are in the interval and a prediction which fits well with real observations. However this graph doesn't represent the real value of data but only stationarized data. Then we should transform this data as follow:

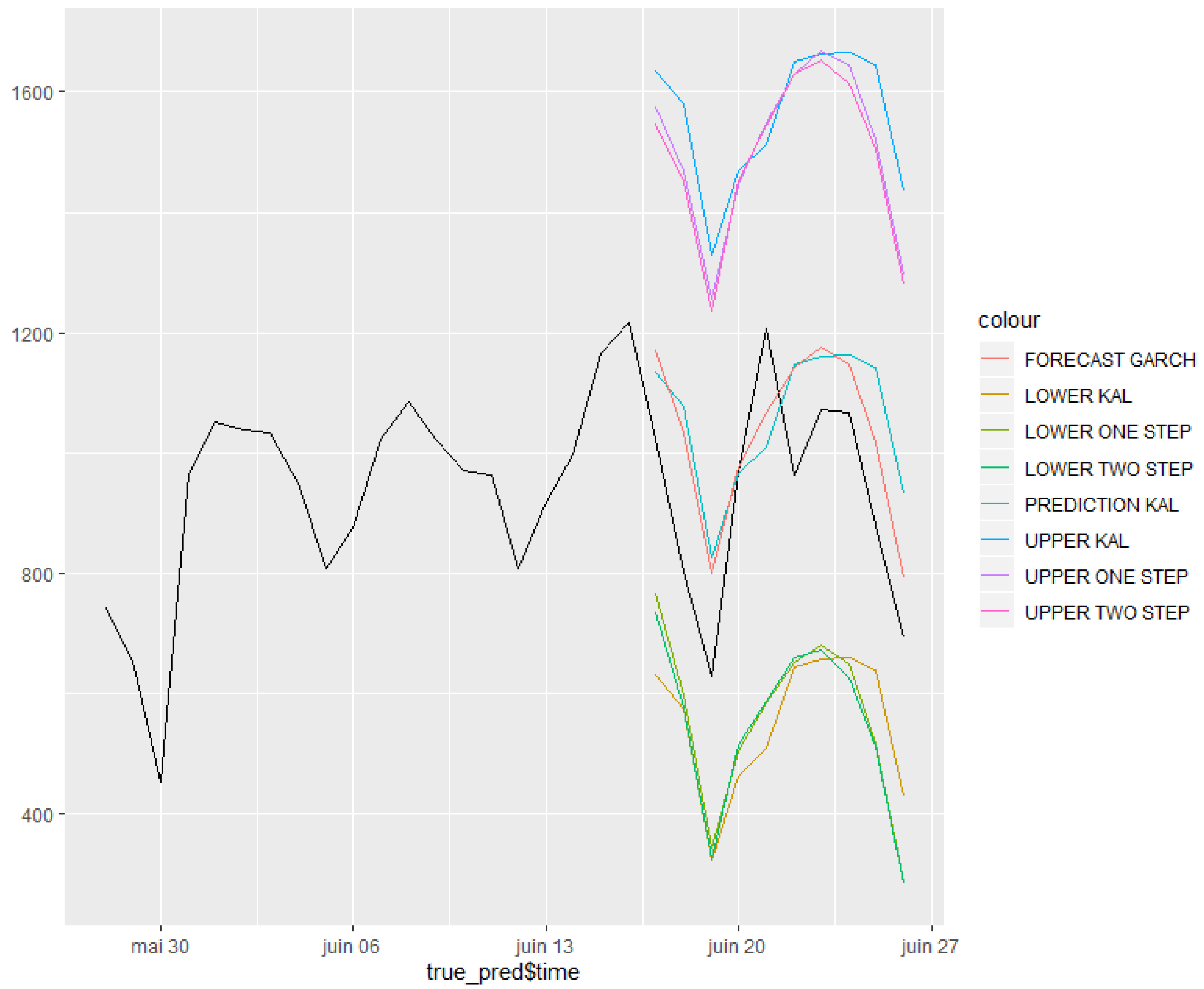
Donc $\boldsymbol{D_t = 100 * X_t + D_{t-7}}$

Indeed we had in first part : $X_t = \frac{D_t - D_{t-7}}{100}$

Finally we obtain the following interval which contains non stationarized data:



Conclusion



To conclude, we can compare our three intervals of predictions for the 1-step procedure, the 2-step procedure and the Kalman recursion. The three intervals are quite large, and any of these three intervals seems to be better than the others. However, the prediction seems to be great for the two models and give us a good idea of the evolution through time of the taxis pickups in this terminal.