Time Series Project

Matteo Biglioli

December 1, 2020

Abstract

The following project has the goal of fitting a time series model to gas consumption data of a city in northern Italy.
We will address the periodicity of our data using Fourier series and fit an ARMA model on the residuals; we will then use the reconstructed model to compute a forecast and evaluate it using the Diebold and Mariano test agains a naive forecast, computed as the consumption of the same day and hour of the previous month.

Introduction

In this first part of the project we will briefly explain the structure of the gas distrubution network and some technical aspects of the device that collected the data we will use in this project.

The Natural Gas Network

The Natural Gas (NG) is one of the most used fuels in the world, it actually satisfies almost 22% of the world's primary energy need (TPES - Total Primary Energy Supply) and over 40% of the Italian need. This resource is transported between states, regions and cities using a network of pipes that connects every end user to the main grid. The whole network is divided in smaller subgrids that operate at different pressures: there are high pressure pipes that carry NG from other countries and low pressure pipes that run below our cities and distribute NG to the end users. This pressure difference is mantained by pressure regulators located in different nodes of the grid. Almost every city has one or more main pressure regulators (Re.Mi.) which connect the high pressure grid, used to transport NG nationwide, and the medium pressure grid, used to distribute NG in the city.

Massflow Meter

Our massflow meter, the device used to collect data, is located at the city gate of a nothern italian town and collects the value of the flow of NG moving through the regulator; because this particular city has only one city gate, the whole NG consumption is collected by our sensor. Here we present two main issues related to the sensor we used to collect our dataset that we addressed in the data cleaning part of the project:

1. Because the city gates are usually located in a remote plant outside the city center, our device is far from stable. Even thogh we programmed the device to have a long retention time, our dataset still has a few missing data due to technical issues.

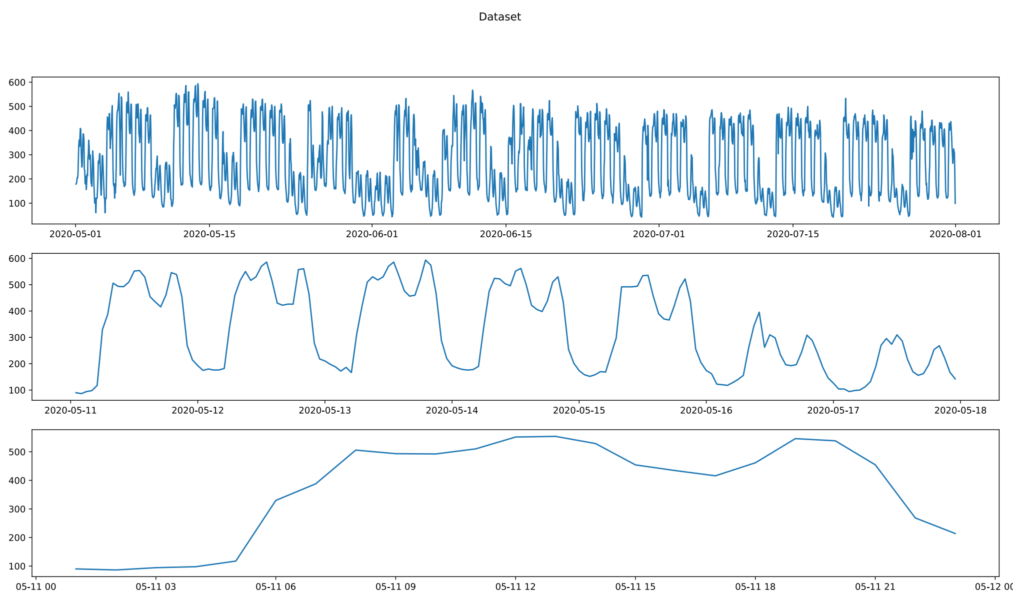
2. The kind of meter used to measure NG flow does not have a constant sampling frequency; our sensor sends an impulse every time a sm^3 of NG passes through it, regardless of the time it takes. This kind of configuration, useful for gas distribution companies whose main focus is the cumulated value of the NG passed through the regulator, must be addressed to work with an equispaced time series.

Data

Our dataset starts on the 2020-05-01 and ends on the 2020-08-01. We choose to use this period for two main reasons:

- The free version of the software used for the project, Eviews, has some limits regarding the number of observations that we can use.
- The summer period allows us not to take into consideration the trend caused by the residential heating that would need more than a few months of data to model.

Below we reported our full dataset and two subsamples from which we can better understand the structure of the data.



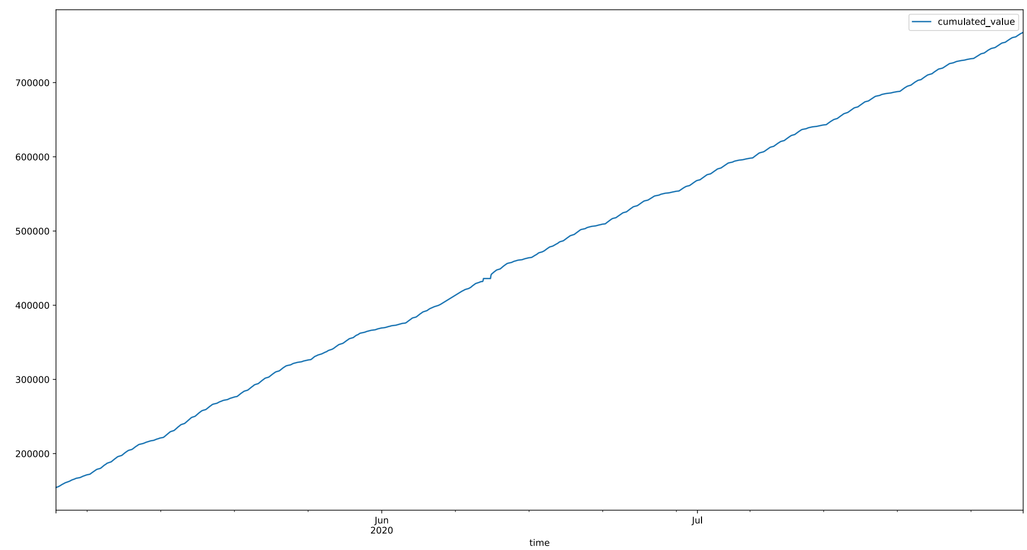
For the purpose of our project, that is forecasting future data, we will split our dataset as folllows:

- from 2020-05-04 to 2020-06-28 as training set (8 weeks)
- from 2020-06-29 to 2020-07-26 as test set (4 weeks)

We will select and estimate the model on the training set and use the test set just to compute the forecast errors.

Data Cleaning

A time series is a sequence of points taken at successive **equally spaced** points in time, but, as discussed in the previous section, our data are not equispaced due to the way they are collected by our sensor. We also highlight that some data are missing due to technical problems.
In order to solve both problems we computed a time-moving integral over our data that gave us the cumulated consumption as shown below:



We then proceed to interpolate the function representing the cumulated consumption to fill the gaps caused by missing data.

Finally we subsampled the cumulated function with a costant sampling frequency to derive an actual time series from our initial data.

Model Selection and Estimation

AutoRegressive Moving Average (ARMA) models are generally used to model time series data, however they do not directly handle seasonality. The ARMA model regresses the current data value against historical data value(s) in the time series. In order to deal with multiple seasonality, external regressors need to be added to the ARMA model.
To incorporate the multiple seasonality in the gas consumption behavior, we added additional Fourier terms to the ARMA model as shown in the generic equation below:

$$y_t = c + \sum_{i=1}^{M} \sum_{k=1}^{K_i} [\alpha_k^{(i)} sin(\frac{2\pi kt}{p_i}) + \beta_k^{(i)} cos(\frac{2\pi kt}{p_i})] + u_t$$

Where u_t is a generic ARMA model.

Fourier Model

A Fourier series is a periodic function composed of harmonically related sinusoids, combined by a weighted summation. As introduced above our goal is to model seasonality using fourier terms as external regressors of an ARMA model. This approach is flexible, and allows us to incorporate multiple periods; in our case we identified two seasonal components, a daily one and a weekly one, which have period respectively $p_1 = 24h$ and $p_2 = 24 \times 7 = 168h$. For each of the periods p_j we added different Fourier terms as shown below:

$$\sum_{k=1}^{K_j} [\alpha_k sin(\frac{2\pi kt}{p_j}) + \beta_k cos(\frac{2\pi kt}{p_j})]$$

In order to find the right number of Fourier terms corresponding to each of the periods we decided to use the Schwarz Information Criterion (SIC), that is computed as:

$$SIC = k \ln(n) - 2 \ln(\mathcal{L})$$

Where k is the number of parameters estimated in the model (in our case two parameters for each Fourier term) and $\mathcal{L}$ is the maximized value of the likelihood function of the model.

The data for the SIC values with varying number of Fourier terms for the two periods $p_1 = 24h$ and $p_2 = 168h$ are shown below:

168 \ 24	1	2	3	4	5
1	12.00744	11.70132	11.66293	11.57809	11.58878
2	11.90488	11.55302	11.50592	11.40260	11.41328
3	11.89481	11.53373	11.48489	11.37768	11.38853
4	11.89493	11.52903	11.47916	11.36985	11.38052
5	11.87220	11.49059	11.43746	11.32124	11.33190
6	11.82668	11.41689	11.35798	11.22892	11.23958
7	NA	NA	NA	NA	NA
8	11.81415	11.39202	11.33065	$\boxed{11.19578}$	11.20664
9	11.82165	11.39776	11.33598	11.20025	11.21091

We notice that we don't have a Fourier term with $k = 7$ for p_2: this is because a term for p_2 with $k = 7$ is equal to the term for p_1 with $k = 1$.

From the figure, it can be seen that the best model, which minimizes the SIC criteria, is one which has four Fourier term for $p_1 = 24h$ and eight Fourier terms for $p_2 = 168h$, with corresponding SIC value of 11.19578. We can interpret this result as a way of the Fourier series to adapt to the two non-symmetric periodicities we have in our dataset that are:

- Weekly periodicity: 2 low-consumption days (weekend) and 5 high-consumption days (working days).
- Daily periodicity: 8 low-consumption hours (night) and 16 high-consumption hours (day) with 3 main peaks (breakfast, lunch and dinner hours).

We can then estimate the coefficents for each one of those terms to get the actual fourier model that would be:

$$y_t = 272.7358 + d_t + w_t$$

Where we define the daily component d_t and the weekly component w_t as

$$\begin{aligned}
d_t = &-24.85sin(\frac{2\pi t}{24}) - 121.12cos(\frac{2\pi t}{24}) - 59.53sin(\frac{2\pi 2t}{24}) - 39.1cos(\frac{2\pi 2t}{24}) + \\
&11.91sin(\frac{2\pi 3t}{24}) - 11.30cos(\frac{2\pi 3t}{24}) + 21.02sin(\frac{2\pi 4t}{24}) + 27.18cos(\frac{2\pi 4t}{24}) \\
w_t = &-88.27sin(\frac{2\pi t}{168}) + 45.35cos(\frac{2\pi t}{168}) + 23.53sin(\frac{2\pi 2t}{168}) + 38.04cos(\frac{2\pi 2t}{168}) + \\
&7.79sin(\frac{2\pi 3t}{168}) - 16.79cos(\frac{2\pi 3t}{168}) + 7.05sin(\frac{2\pi 4t}{168}) + 11.11cos(\frac{2\pi 4t}{168}) + \\
&3.18sin(\frac{2\pi 5t}{168}) - 22.86cos(\frac{2\pi 5t}{168}) - 29.22sin(\frac{2\pi 6t}{168}) - 0.84cos(\frac{2\pi 6t}{168}) + \\
&12.68sin(\frac{2\pi 8t}{168}) - 13.31cos(\frac{2\pi 8t}{168})
\end{aligned}$$

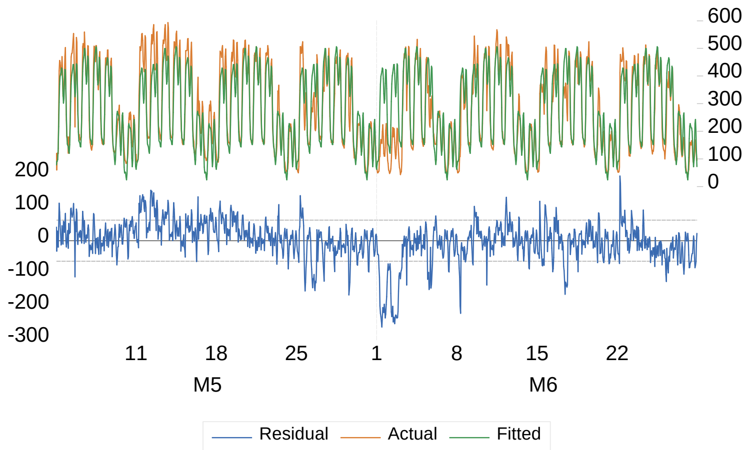
Here we also report the actual output from Eviews of the coefficents' estimation:

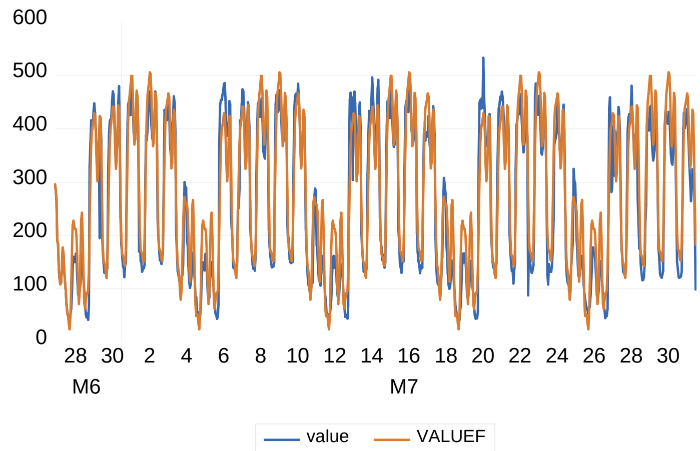
Dependent Variable: VALUE
Method: Least Squares
Date: 11/26/20 Time: 22:12
Sample: 5/04/2020 01:00 6/28/2020 23:00
Included observations: 1343
HAC standard errors & covariance (Bartlett kernel, Newey-West fixed bandwidth = 8.0000)

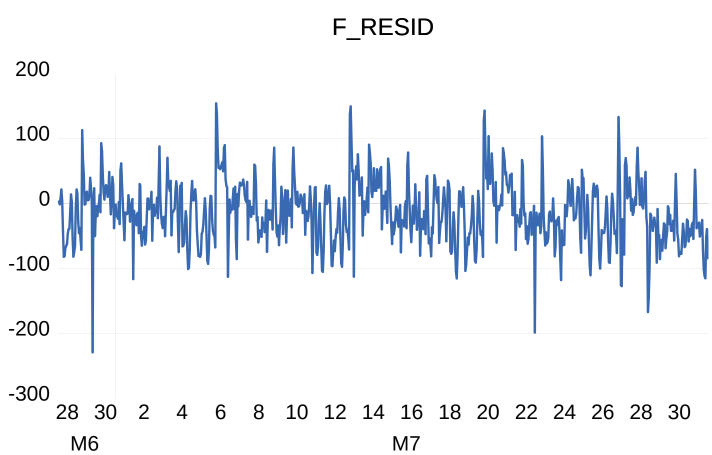
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	275.7358	3.939162	69.99859	0.0000
SIN_D_1	-24.85258	4.995089	-4.975403	0.0000
COS_D_1	-121.1203	4.709698	-25.71721	0.0000
SIN_D_2	-59.53211	2.937746	-20.26455	0.0000
COS_D_2	-39.09610	3.538791	-11.04787	0.0000
SIN_D_3	22.90844	2.443606	9.374848	0.0000
COS_D_3	-11.30386	1.789414	-6.317074	0.0000
SIN_D_4	21.01741	1.784900	11.77512	0.0000
COS_D_4	27.18075	2.179832	12.46919	0.0000
SIN_W_1	-88.26955	5.056609	-17.45628	0.0000
COS_W_1	45.35097	6.013724	7.541245	0.0000
SIN_W_2	23.52908	5.463921	4.306263	0.0000
COS_W_2	38.03760	5.557054	6.844922	0.0000
SIN_W_3	7.786788	4.675350	1.665499	0.0961
COS_W_3	-16.78644	6.099349	-2.752168	0.0060
SIN_W_4	7.049990	4.861264	1.450238	0.1472
COS_W_4	11.11314	5.757218	1.930297	0.0538
SIN_W_5	3.181043	5.166246	0.615736	0.5382
COS_W_5	-22.85625	5.223283	-4.375839	0.0000
SIN_W_6	-29.22292	5.095691	-5.734830	0.0000
COS_W_6	-0.844737	4.975696	-0.169773	0.8652
SIN_W_8	12.68754	4.830246	2.626687	0.0087
COS_W_8	-13.31585	4.469207	-2.979465	0.0029

R-squared	0.827824	Mean dependent var	275.8946
Adjusted R-squared	0.824955	S.D. dependent var	148.0084
S.E. of regression	61.92436	Akaike info criterion	11.10668
Sum squared resid	5061707.	Schwarz criterion	11.19578
Log likelihood	-7435.137	Hannan-Quinn criter.	11.14006
F-statistic	288.4815	Durbin-Watson stat	0.334549
Prob(F-statistic)	0.000000	Wald F-statistic	169.7560
Prob(Wald F-statistic)	0.000000		

Before addressing the residual, we can further analyze the Fourier model to understand how well it behaves in fitting the sample data and forecasting future consumption values as shown below:


600
500
400
300
200
100
0
200
100
0
-100
-200
-300
11
18
25
1
8
15
22
M5
M6
Residual
Actual
Fitted

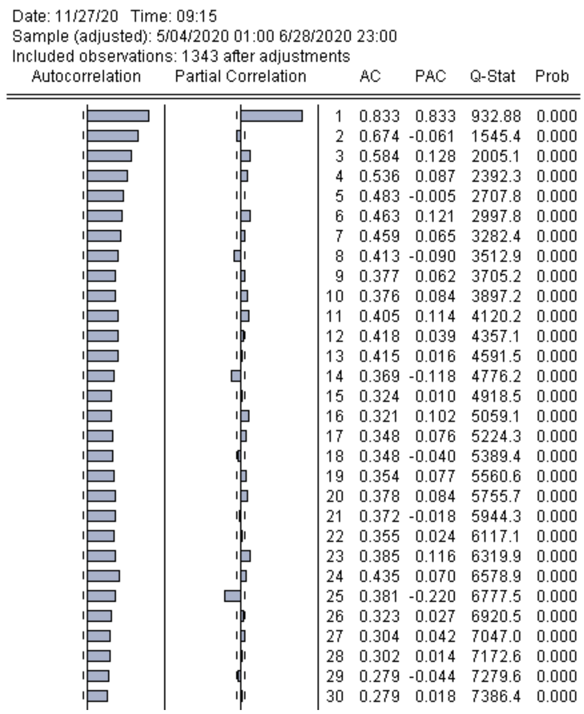

600
500
400
300
200
100
0
28 30 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
M6
M7
value
VALUEF


F_RESID
200
100
0
-100
-200
-300
28 30 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
M6
M7

From the graphs above we can notice that, even if the Fourier model is a deterministic model and we estimated its coefficents using only the training set, it actually fits very well both training and test data. We can also briefly discuss the high spike in residuals that we can see in the first days of June: in that case the model is estimating an higher consumption than the one we actually registered; this is due to the fact that in those days there was a nationwide holiday in Italy that reduced NG consumption of companies and factories.

Residual Fitting

We can then proceed to estimate an ARMA model on the residuals. Here we show the correlogram of our residual time series:


Date: 11/27/20 Time: 09:15
Sample (adjusted): 5/04/2020 01:00 6/28/2020 23:00
Included observations: 1343 after adjustments

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.833	0.833	932.88	0.000
		2	0.674	-0.061	1545.4	0.000
		3	0.584	0.128	2005.1	0.000
		4	0.536	0.087	2392.3	0.000
		5	0.483	-0.005	2707.8	0.000
		6	0.463	0.121	2997.8	0.000
		7	0.459	0.065	3282.4	0.000
		8	0.413	-0.090	3512.9	0.000
		9	0.377	0.062	3705.2	0.000
		10	0.376	0.084	3897.2	0.000
		11	0.405	0.114	4120.2	0.000
		12	0.418	0.039	4357.1	0.000
		13	0.415	0.016	4591.5	0.000
		14	0.369	-0.118	4776.2	0.000
		15	0.324	0.010	4918.5	0.000
		16	0.321	0.102	5059.1	0.000
		17	0.348	0.076	5224.3	0.000
		18	0.348	-0.040	5389.4	0.000
		19	0.354	0.077	5560.6	0.000
		20	0.378	0.084	5755.7	0.000
		21	0.372	-0.018	5944.3	0.000
		22	0.355	0.024	6117.1	0.000
		23	0.385	0.116	6319.9	0.000
		24	0.435	0.070	6578.9	0.000
		25	0.381	-0.220	6777.5	0.000
		26	0.323	0.027	6920.5	0.000
		27	0.304	0.042	7047.0	0.000
		28	0.302	0.014	7172.6	0.000
		29	0.279	-0.044	7279.6	0.000
		30	0.279	0.018	7386.4	0.000

from wich we can try to deduce the p and q values of the $ARMA(p,q)$ that would be the best fit for our residuals:

- The high spikes of the PACF at the first lag, combined with an autocorrelation that goes down in time, suggest the presence of a few AutoRegression components.
- The high spikes of the PACF at lag 23, 24 and 25, that we can easily interpret as the correlation with the conusumption values at the same hour of the day before, suggest that there could be one or two Moving Average components with lag 23, 24 and 25.

We decided to use an Information Criteria to find the best ARMA model to fit our residuals; so we used the *automatic ARIMA forecasting* procedure of Eviews with SIC, that returned the following:

Automatic ARIMA Forecasting
Selected dependent variable: R_FOURIER_TRAIN
Date: 11/27/20 Time: 11:40
Sample: 5/01/2020 01:00 7/31/2020 23:00
Included observations: 1343
Forecast length: 0

Number of estimated ARMA models: 121
Number of non-converged estimations: 1
Selected ARMA model: (9,7)(0,0)
SIC value: 9.80050325995

Model Selection Criteria Table
Dependent Variable: R_FOURIER_TRAIN
Date: 11/27/20 Time: 11:40
Sample: 5/01/2020 01:00 7/31/2020 23:00
Included observations: 1343

Model	LogL	AIC	BIC*	HQ
(9,7)(0,0)	-6516.213988	9.730773	9.800503	9.756894
(8,8)(0,0)	-6516.407584	9.731061	9.800792	9.757182
(8,10)(0,0)	-6513.596463	9.729853	9.807331	9.758876
(6,5)(0,0)	-6541.273288	9.760645	9.811006	9.779510
(7,6)(0,0)	-6538.074781	9.758860	9.816969	9.780628
(7,10)(0,0)	-6527.357846	9.748858	9.822462	9.776429
(6,10)(0,0)	-6533.008357	9.755783	9.825513	9.781904
(5,3)(0,0)	-6564.265345	9.790417	9.829157	9.804929
(7,5)(0,0)	-6549.966621	9.775081	9.829315	9.795397
(4,4)(0,0)	-6565.646071	9.792474	9.831213	9.806985
(10,4)(0,0)	-6545.263800	9.771056	9.833038	9.794274
(5,4)(0,0)	-6563.998246	9.791509	9.834122	9.807472
(7,3)(0,0)	-6560.527312	9.787829	9.834316	9.805243
(6,6)(0,0)	-6553.863671	9.780884	9.835119	9.801200
(8,5)(0,0)	-6552.137074	9.779802	9.837911	9.801569
(10,8)(0,0)	-6534.233298	9.760586	9.838064	9.789609
(7,9)(0,0)	-6541.588830	9.768561	9.838292	9.794682
(6,4)(0,0)	-6563.735800	9.792607	9.839094	9.810021
(7,4)(0,0)	-6560.345105	9.789047	9.839408	9.807912
(4,3)(0,0)	-6575.410235	9.805525	9.840390	9.818586
(7,2)(0,0)	-6571.768724	9.803081	9.845694	9.819043
(6,8)(0,0)	-6555.235280	9.785905	9.847888	9.809124
(5,8)(0,0)	-6565.485792	9.799681	9.857790	9.821448
(3,8)(0,0)	-6572.996865	9.807888	9.858249	9.826753
(3,9)(0,0)	-6572.010614	9.807909	9.862143	9.828225

From the Information Criteria is easy to identify that the best model for our data is an ARMA(9,7).

We then estimated the model on the residuals and got the following coefficents (notice that we remove the constant that is already taken care of in the Fourier component of the model):

Dependent Variable: R_FOURIER_TRAIN
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 11/27/20 Time: 11:53
Sample (adjusted): 5/04/2020 10:00 6/28/2020 23:00
Included observations: 1334 after adjustments
Failure to improve likelihood (non-zero gradients) after 35 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 5/03/2020 17:00 5/03/2020 23:00

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.414083	0.036473	11.35303	0.0000
AR(2)	0.180518	0.028898	6.246683	0.0000
AR(3)	-0.357735	0.012610	-28.36881	0.0000
AR(4)	0.549805	0.014235	38.62402	0.0000
AR(5)	-0.282625	0.016092	-17.56358	0.0000
AR(6)	0.410384	0.009326	44.00497	0.0000
AR(7)	0.679266	0.012474	54.45312	0.0000
AR(8)	-0.758124	0.032528	-23.30673	0.0000
AR(9)	0.118302	0.028650	4.129253	0.0000
MA(1)	0.483695	0.025475	18.98673	0.0000
MA(2)	0.111057	0.035813	3.101028	0.0020
MA(3)	0.397593	0.035929	11.06592	0.0000
MA(4)	-0.166534	0.043120	-3.862136	0.0001
MA(5)	0.097550	0.035753	2.728469	0.0064
MA(6)	-0.300571	0.035407	-8.488958	0.0000
MA(7)	-0.914222	0.025132	-36.37709	0.0000

R-squared	0.752569	Mean dependent var	-0.250977
Adjusted R-squared	0.749753	S.D. dependent var	61.44444
S.E. of regression	30.73737	Akaike info criterion	9.700757
Sum squared resid	1245228.	Schwarz criterion	9.763077
Log likelihood	-6454.405	Hannan-Quinn criter.	9.724110
Durbin-Watson stat	1.995406		

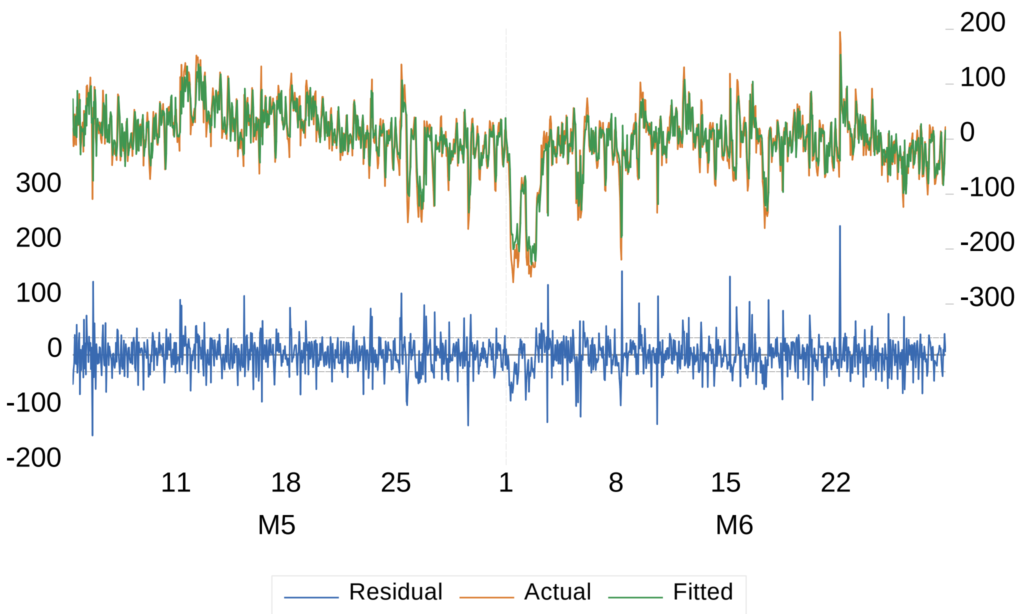
Inverted AR Roots	.98	.63	.53+.85i	.53-.85i
	.19	-.26-.96i	-.26+.96i	-.97-.26i
	-.97+.26i			
Inverted MA Roots	.92	.53-.85i	.53+.85i	-.27-.96i
	-.27+.96i	-.96+.26i	-.96-.26i	

From which we conclude that the model fitted on the residuals is the following:

$$
\begin{aligned}
u_t = 0.41u_{t-1} + 0.18u_{t-2} - 0.35u_{t-3} + 0.55u_{t-4} - 0.28u_{t-5} + \\
0.41u_{t-6} + 0.68u_{t-7} + 0.76u_{t-8} - 0.35u_{t-3} + 0.12u_{t-9} + \\
0.48\epsilon_{t-1} + 0.11\epsilon_{t-2} + 0.4\epsilon_{t-3} - 0.17\epsilon_{t-4} + 0.01\epsilon_{t-5} - \\
0.3\epsilon_{t-6} - 0.91\epsilon_{t-7}
\end{aligned}
$$

We can further notice, checking the inverted roots printed at the bottom of the table, that the model we are considering seems to present some common factors (we actually expect the common roots to be different at some decimal digits). For the purpose of this project we will keep the analysis using the $ARMA(9,7)$ model selected by SIC, but we highlight that a feasible next step would be to compare the choosen model to the one that can be built removing the common factors, following the **parsimonious modeling** approach.

Before merging the two models to actually fit our whole training dataset, we briefly analyze the fitting of the ARMA model with the following graphs:



Date: 11/27/20 Time: 12:18
Sample (adjusted): 5/04/2020 10:00 6/28/2020 23:00
Q-statistic probabilities adjusted for 16 ARMA terms

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.001	0.001	0.0016	
		2	-0.007	-0.007	0.0658	
		3	-0.005	-0.005	0.0972	
		4	0.046	0.046	2.9733	
		5	0.007	0.007	3.0484	
		6	-0.075	-0.074	10.558	
		7	0.011	0.012	10.724	
		8	0.035	0.033	12.384	
		9	-0.006	-0.008	12.432	
		10	-0.070	-0.064	19.039	
		11	-0.018	-0.018	19.496	
		12	0.100	0.092	32.937	
		13	0.030	0.031	34.156	
		14	-0.040	-0.031	36.344	
		15	-0.055	-0.055	40.381	
		16	0.027	0.010	41.344	
		17	0.010	0.007	41.482	0.000
		18	-0.079	-0.061	49.953	0.000
		19	-0.010	-0.004	50.088	0.000
		20	0.036	0.021	51.840	0.000
		21	0.050	0.042	55.226	0.000
		22	-0.035	-0.013	56.895	0.000
		23	0.034	0.046	58.499	0.000
		24	0.178	0.159	101.63	0.000
		25	-0.000	-0.017	101.63	0.000
		26	-0.076	-0.070	109.52	0.000
		27	-0.027	-0.012	110.55	0.000
		28	0.060	0.040	115.45	0.000
		29	0.038	0.034	117.46	0.000
		30	-0.046	-0.013	120.29	0.000

For the purpose of this project we will stop our estimation here, even if we can notice from the correlogram of the residuals that we actually still have some correlations that can be modeled. We highlight in particular the spike in the PACF plot at $lag = 24$ that is obviously the correlation with the consumption at the same hour of the previous day.

Final Model

We can finally reconstruct our model using the components defined above (d_t, w_t, u_t) as:

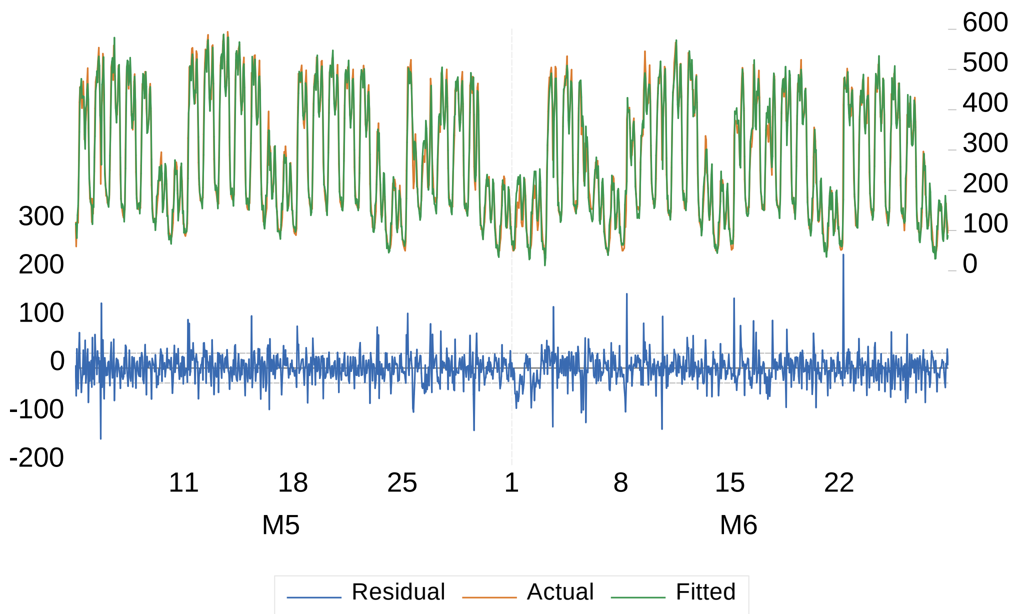
$$y_t = c + d_t + w_t + u_t$$

where the estimates for the coefficents are the following:

Dependent Variable: VALUE
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 11/27/20 Time: 12:25
Sample: 5/04/2020 01:00 6/28/2020 23:00
Included observations: 1343
Failure to improve likelihood (non-zero gradients) after 37 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 5/03/2020 17:00 5/03/2020 23:00

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	272.4340	13.28114	20.51285	0.0000
SIN_D_1	-25.24204	3.478126	-7.257368	0.0000
COS_D_1	-121.0722	3.476154	-34.82936	0.0000
SIN_D_2	-59.70626	2.538170	-23.52335	0.0000
COS_D_2	-39.06646	2.539988	-15.38057	0.0000
SIN_D_3	22.80074	1.898948	12.00704	0.0000
COS_D_3	-11.29232	1.901154	-5.939716	0.0000
SIN_D_4	20.83585	1.416078	14.71377	0.0000
COS_D_4	27.09769	1.416454	19.13066	0.0000
SIN_W_1	-86.47748	8.221161	-10.51889	0.0000
COS_W_1	47.29184	8.233948	5.743520	0.0000
SIN_W_2	22.82198	5.375379	4.245650	0.0000
COS_W_2	36.82446	5.412082	6.804121	0.0000
SIN_W_3	7.956465	4.503855	1.766590	0.0775
COS_W_3	-15.85818	4.530666	-3.500188	0.0005
SIN_W_4	7.193375	4.102460	1.753430	0.0798
COS_W_4	10.42171	4.108662	2.536521	0.0113
SIN_W_5	2.860359	3.850469	0.742860	0.4577
COS_W_5	-22.39572	3.842662	-5.828179	0.0000
SIN_W_6	-28.82728	3.652534	-7.892404	0.0000
COS_W_6	-1.085645	3.643594	-0.297960	0.7658
SIN_W_8	13.01033	3.318722	3.920283	0.0001
COS_W_8	-13.21310	3.322947	-3.976320	0.0001
AR(1)	0.411303	0.036540	11.25623	0.0000
AR(2)	0.180393	0.029042	6.211535	0.0000
AR(3)	-0.357465	0.012675	-28.20138	0.0000
AR(4)	0.548903	0.014213	38.62051	0.0000
AR(5)	-0.281080	0.016155	-17.39927	0.0000
AR(6)	0.409545	0.009320	43.94425	0.0000
AR(7)	0.680168	0.012485	54.47931	0.0000
AR(8)	-0.755365	0.032580	-23.18469	0.0000
AR(9)	0.117843	0.028838	4.086348	0.0000
MA(1)	0.483954	0.025403	19.05124	0.0000
MA(2)	0.112626	0.035733	3.151820	0.0017
MA(3)	0.398516	0.035873	11.10898	0.0000
MA(4)	-0.169196	0.042995	-3.935249	0.0001
MA(5)	0.095359	0.035581	2.680089	0.0075
MA(6)	-0.301367	0.035214	-8.558240	0.0000
MA(7)	-0.915007	0.025000	-36.60070	0.0000

Before the computation of the forecast, we briefly analyze the fitting of our final model on the training set with the following graph:



from which we can see that the combination of the two models behave very well in both standard periodic situations and once-in-a-while holidays like the one that happens on the 2nd of June.

Forecast Evaluation

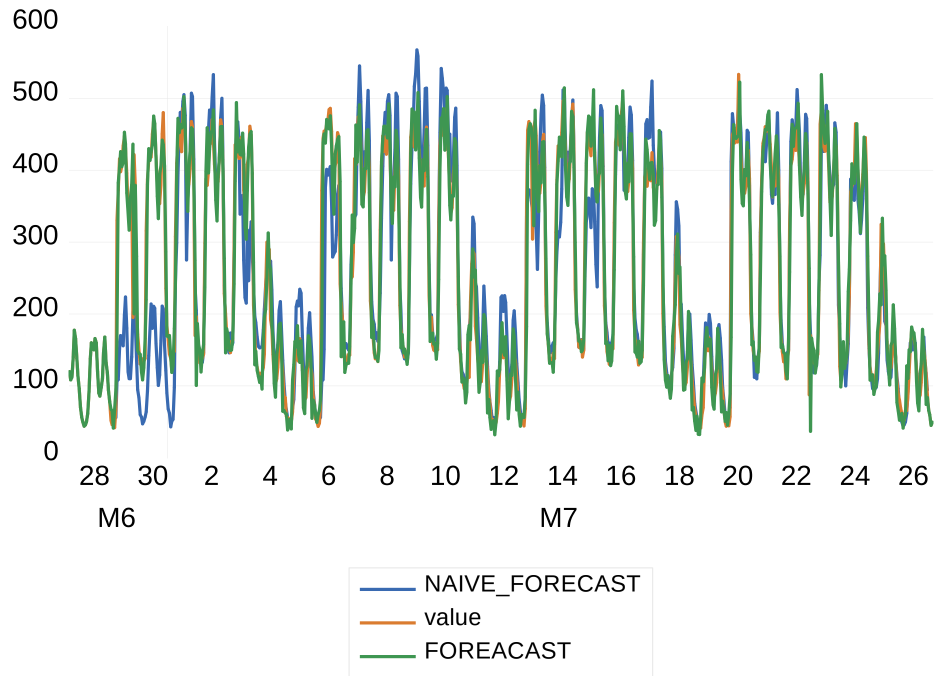
In this last section we will compute and evaluate our forecast using the Diebold-Mariano Test.

Forecast Computation

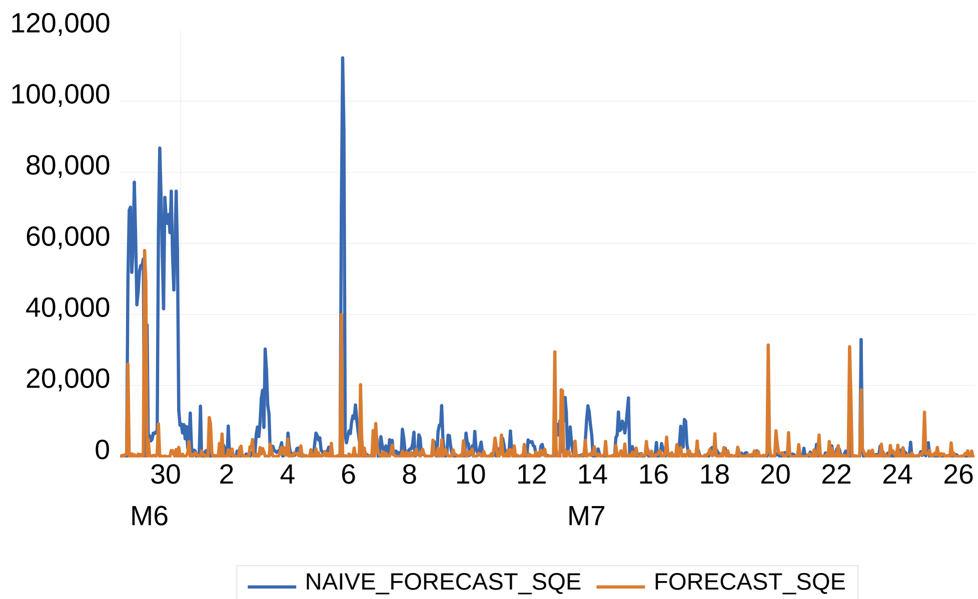
We used the complete model to compute a forecast for the period 2020-06-29 to 2020-07-26.
We also computed a naive forecast using the consumption value of the same hour of the previous month, that can be modeled as:

$$\hat{y}_t = y_{t-(24\times7\times4)} = y_{t-672}$$

Before the actual evaluation of the forecast, we compared the two forecasts mentioned above with the actual test data in the following graph:



We also computed the Squared Errors, $e_t = (\hat{y}_t - y_t)^2$, of both forecasts against our test set in order to better visualize the difference between the two forecasts, given that they are not clearly distinguishable from the basic plot. In the following graph we show the comparison of the two squared errors:



Even if from the graph above it seems that our forecast performs better than the naive one, we also compared their Mean Squared Error to get a unique simple metrics from which we can easily identify the best forecast:

$$\begin{aligned} MSE_{naive} &= 5162.17 \\ MSE_{forecast} &= 1341.77 \end{aligned} \tag{1}$$

The Diebold-Mariano Test

From a basic MSE comparison we can notice that the forecast computed using our model seems to perform significantly better than the naive forecast; we just need to check that the difference between the two is actually statistically significant.
In order to do so we define :

$$d_t = g(e_t^{(naive)}) - g(e_t^{(forecast)}) \quad \text{where} \quad g(x) = x^2 \tag{2}$$

and we test the null hypotesis $H_0 : \{\mathbb{E}[d_t] = 0\}$.

This test in Eviews can be easily computed estimating a constant model on the series $g(e_t^{(naive)}) - g(e_t^{(forecast)})$ (with HAC covariance method), which will return both the coefficent estimate and the result of our test as we show below:

Dependent Variable: FORECAST_SQE-NAIVE_FORECAST_SQE
Method: Least Squares
Date: 11/28/20 Time: 15:26
Sample: 6/29/2020 01:00 7/26/2020 23:00
Included observations: 671
HAC standard errors & covariance (Bartlett kernel, Newey-West fixed bandwidth = 7.0000)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-3820.399	1194.115	-3.199356	0.0014

R-squared	0.000000	Mean dependent var	-3820.399
Adjusted R-squared	0.000000	S.D. dependent var	14267.81
S.E. of regression	14267.81	Akaike info criterion	21.97089
Sum squared resid	1.36E+11	Schwarz criterion	21.97761
Log likelihood	-7370.233	Hannan-Quinn criter.	21.97349
Durbin-Watson stat	0.428171		

From this output we can notice that our model performs significantly better than the naive forecast, and that the difference between the two is statistically significant even at the 5% level.

Conclusions

From this study we can conclude that a time series model that includes fourier terms to account for periodic patterns may be both one of the most elegant and performant ways to address this kind of datasets.
We also recall that at the beginning of the study we imposed ourselves a strong limitation regarding the dataset dimension, that has lead us to fit our model on a well defined time span that does not include the winter season.
Two next steps to further analyze this kind of problem and provide better and more general forecasts could be:

1. Relax the limitations about the dataset size and add a Fourier component accounting for the seasonal periodicity.

2. Further analyze the ARMA component of the model to check if it can be simplified according to the **parsimonious modeling** approach.