

Paper 004-2013

Fueling the Future of an Energy Company

Stijn Vanderlooy, Ph.D., EDF Luminus, Brussels, Belgium

Wim Mues, Anne De Greef, SAS Institute Inc., Tervuren, Belgium

ABSTRACT

EDF Luminus is a producer and supplier of energy in Belgium. The company is active in several markets for trading commodities including electricity, natural gas, and oil-related products. The market data modeling team is responsible for all data related to these markets. Besides a daily collection and verification of published prices, the team performs a diverse set of data transformations and manipulations in order to extract valuable information from the markets. Examples are numerous and include volatility estimation and price forecasts. In this paper we present a successful application of an in-house developed SAS tool that is used by the market data modeling team to support its core tasks. The tool runs autonomously three times a day.

INTRODUCTION

EDF Luminus is a company that has been present in Belgium for more than half a century. The company not only produces but also trades and supplies commodities such as electricity and natural gas.

The energy business may be said to be challenging at least for a variety of reasons. To illustrate, consider for a moment electricity as a commodity. While we as a customer expect electricity to be available anytime, the commodity is by its nature difficult to store in large amounts. This means that an energy company has to forecast the demand as accurately as possible and, in the end, has to make sure that it can meet its obligation to supply as much as needed (not less and not more in order not to cause capacity problems in the network). Electricity can be generated or it can be bought from another party. Generation can happen, for example, by natural gas-fired power plants. But in order to run such a plant, it is the natural gas that needs to be bought in advance. The expected level of the price is important since the energy company has to pay today for delivery in the future, while customers only pay at delivery time.

For buying and selling commodities such as electricity and natural gas there are several markets that often operate the way stock markets do. As such, an energy company as EDF Luminus is next to retailer and supplier also a trader like a financial institution can be. The collection, verification, and internal re-distribution of the prices published by the energy markets are daily tasks of the market data modeling team. All data are collected in SAS and subsequently used in various statistical analyses (descriptive and predictive). Such analyses include among others accurate predictions of the volatility and future price level of the markets.

This paper will explain how SAS is used at EDF Luminus in order to support the market data modeling team. Scope and confidentiality do not allow us to explain in detail the complete tool that we have built during the years. Instead we will describe its main design and, at several occasions throughout the system, zoom in on important aspects. The main message we want to bring forward is that one can successfully implement full processes with SAS from the very first input to the very last output. Indeed, our SAS tool runs fully automated three times a day, each day.

The remainder of the paper is organized in three large sections as follows. The first section provides some background that will help to clarify the objectives of our SAS tool. The design of the tool is explained in the second section. Some interesting core aspects of the tool will be detailed at the right moments and an application of a data analysis will be presented. The last section concludes the paper.

BACKGROUND

In this section, we provide a background to EDF Luminus and the energy markets in which it is active. This background is needed in order to understand the tasks performed by the market data modeling team.

EDF LUMINUS

EDF Luminus is a company that trades and supplies electricity and natural gas. The company has been present in Belgium for more than half a century. It sells its electricity and gas in Belgium under the brand 'Luminus' to 1.7 million residential and professional customers. This results in a market share of 20% in Belgium.

The retail part of EDF Luminus is clearly essential for its success and continuous growth in market share. However, to meet the demand for (consumption of) energy, there are basically two options on the supply side, namely:

1. Produce with a power plant of some sort;
2. Trade in any of the energy markets.

Sometimes it will be a mix of the two, for example, when demand peaks the power plants may not produce sufficiently so additional amounts have to be bought on the market. In order to illustrate in more detail, which will make clear why market data modeling is much needed, we first look at the different ways of electricity production; see Figure 1.

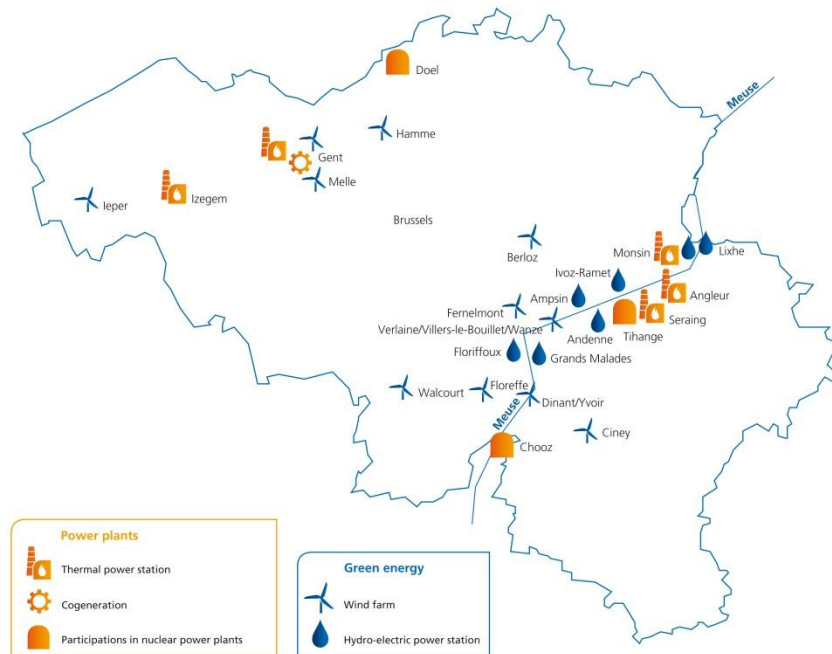


Figure 1: EDF Luminus' production units are spread across the whole of Belgium.

EDF Luminus has more than 20 production sites across Belgium. They can be organized in the following four categories: hydropower plants, wind farms, combined-cycle thermal plants and cogeneration. Hydropower plants use the force of falling water to turn turbine blades causing electricity to be generated in an attached generator. The amount of electricity produced is quite stable. This is in contrast to the wind farms of which the effectiveness depends on volatile and local weather conditions. A thermal power station heats water and the resulting steam spins a turbine which in turn drives a generator. A few plants require some time to be up and running while others, so-called peak-lead units, are up and running in less than 10 minutes. The drawback is that they are less efficient.

Efficiency is an important aspect of the thermal plants since the output energy is always lower than the input energy due to inefficiencies such as friction. So, the more state-of-the-art the power plant, the more electricity can be produced from the same amount of input (gas, coal...). This means for the gas-fired power plants that, if gas prices are high and electricity prices are low, it may not be beneficial to run the plant. In that case, the best option is to buy electricity on the market, which is explained in the next section. We note that the expected future price levels of the commodities (which have to be forecasted) are the driver for the decision whether to produce, and if so, how much.

ENERGY MARKETS

Energy markets are commodity markets, meaning that energy is traded on regulated exchanges. An exchange is often not limited to a single commodity. For example, APX-ENDEX provides a trading platform for electricity and natural gas in the Netherlands, UK, and Belgium. ICE provides worldwide exchanges for many commodities.

In general two types of trading can be distinguished, namely:

1. Spot trading (short-term);
2. Future trading (longer-term).

When a party engages in spot trading the 'delivery' of the commodity takes place immediately or with a short delay needed to ensure that delivery can happen. Trading on the spot is thus short-term trading; often day-ahead: a party agrees with another to buy or sell a predefined amount at a fixed price for delivery the next day. Focusing on electricity as commodity for a moment, the day-ahead market in many cases has an hourly granularity so that the operator of the electricity network can inspect that demand and supply matches at any moment of the day.

Spot trading is in contrast with future trading. In a future contract the parties agree on an amount and price at the moment of making the deal while the delivery of the commodity is in the not-so-near future. Related to energy commodities, these futures often come with standardized maturities denoted as M+1, M+2, ..., Q+1, Q+2, ... Y+1, Y+2, and so on. M+1 stands for a future with delivery next month meaning, for most commodities, each day of next month a uniform part of the commodity is delivered so that, at the last day of next month, the total amount has been transferred. Maturity M+2 stands for the month afterward and so on. In order to have sufficient parties interested in the futures, at some point in time (e.g., after 3 or 6 months ahead) the market switches to deliveries of quarters, then seasons, and ultimately years. An example is given in the Table 1 below.

Future	Price	Delta
M+1 (Mar-13)	47.30 €/MWh	+0.35 €/MWh (+0.7%)
M+2 (Apr-13)	43.34 €/MWh	+0.07 €/MWh (+0.2%)
M+3 (May-13)	39.20 €/MWh	+0.57 €/MWh (+1.5%)
Q+1 (Q2-13)	40.56 €/MWh	+0.68 €/MWh (+1.7%)
Q+2 (Q3-13)	39.11 €/MWh	+0.31 €/MWh (+0.8%)
Q+3 (Q4-13)	49.69 €/MWh	+0.15 €/MWh (+0.3%)
Q+4 (Q1-14)	50.40 €/MWh	+0.43 €/MWh (+0.9%)
Y+1 (Cal-2014)	44.25 €/MWh	+0.87 €/MWh (+2.0%)
Y+2 (Cal-2015)	43.83 €/MWh	+0.83 €/MWh (+1.9%)
Y+3 (Cal-2016)	44.73 €/MWh	+0.52 €/MWh (+1.2%)

Table 1: Market prices published by APX-ENDEX at February, 1st, 2013 for electricity in Belgium. There are three monthly futures, four quarterly futures, and three yearly (calendar) futures that can be traded.

Clearly, future contracts impose quite some risk since it is unknown what will be the value of the commodity at time of its delivery, while the price of the deal has been agreed on today. The profit (or loss) of the deal will be known only at time of delivery by comparing the value of the deal against the spot price. Estimating future prices is one of the main tasks of the market data modeling team in EDF Luminus. It supports risk management and decision making.

MARKET DATA MODELING

The market data modeling team is responsible for all aspects related to the price data published by the markets in which EDF Luminus is active. This responsibility starts with the (automatic) collection and verification of daily market prices. Once the prices pass all quality checks, they reside in the database for further processing and analyses. The results are being distributed to the various departments in the company.

The previous sections have shown that EDF Luminus is active in a challenging environment. Market data modeling plays an important role in several of the activities of the company. In order to support its tasks, it is clear that an IT system has to be chosen that is capable of doing data management at one side and analytics at the other side. As such, naturally, it has been chosen to build the 'market data module' in SAS.

MARKET DATA MODULE

In this section, we will explain the market data module that has been built in EDF Luminus using SAS Base. It runs automatically three times a day (morning, noon, and evening) during the week as well as weekends. Three large parts will be distinguished in the subsections below, namely: input, analyses, and output.

CONFIGURATION AND IMPORT OF MARKET DATA

Each energy market has its own way and format of publishing prices. Nonetheless, it is important that the same SAS code can be used for all input tasks. To that purpose, we decided to configure for each combination of market and commodity a specific spreadsheet to be processed by SAS. Although these spreadsheets may contain whatever data as they like, there is a small set of simple rules to obey so that our system will know how to read the file and give meaning to its contents. In the next paragraphs we will explain the procedure.

A spreadsheet may contain various sheets depending on the type of price. For example, in case of futures, a first sheet may contain the monthly futures, while the second sheet contains the quarterly ones. A third and fourth sheet may contain the seasonally and yearly futures; see Figure 2 for an example. The file contains the daily closing prices for natural gas delivered at Zeebrugge. At a daily basis a new row is added with the closing prices.

It is worthwhile to investigate the column headers in this figure. Since we are in the monthly sheet and considering a publication date in November, one would expect prices for December 2012, January 2013 and so on. But using absolute delivery dates in the columns would mean that manual interventions are needed since extra columns have to be added when the market evolves to futures with new delivery. Therefore, instead of using absolute dates, we use what we call 'rollovers'. This means that the first column contains simply the price of M+1, the second contains the M+2 price, and likewise for the remaining columns. The absolute month can be derived from the number in the column header and the publication date (first column in each sheet). Same principle holds for the other sheets.

Besides the sheets containing the prices, each input file contains a 'Legend' that tells SAS how to interpret the contents. Roughly speaking, the legend is a table that defines for each sheet, and for each of its columns, a code that SAS can decrypt in order to understand the type of price in the column. For example, the code for the second column in Figure 2 will be Quote_X_M_Plus_1_BID indicating that we have the BID price (= quote) of maturity M+1. The 'X' is a dummy and becomes relevant when reading, among others, electricity prices since then it is used to indicate whether it is a base, peak, or off-peak price. Albeit that this kind of legend seems simple, it has proved to be a very flexible way for dealing with the diverse set of data within EDF Luminus. It is a universal and consistent principle that can be used to quickly introduce new market data. Indeed, given new data in this format, we never have to change our SAS code since it is written in a generic way to understand what the legend says (note that it is the legend and its link with the sheets/columns that tells SAS how to read the data and how to store it in the database).

	A	B	C	D	E	F	G	H	I	J
1	Date	Zeebrugge Monthly Rollover Series 1st Month ICIS Heren Bid As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 2nd Month ICIS Heren Bid As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 3rd Month ICIS Heren Bid As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 1st Month ICIS Heren Offer As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 2nd Month ICIS Heren Offer As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 3rd Month ICIS Heren Offer As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 1st Month ICIS Heren Mid As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 2nd Month ICIS Heren Mid As Quoted(GBC: F/THERM)	Zeebrugge Monthly Rollover Series 3rd Month ICIS Heren Mid As Quoted(GBC: F/THERM)
3081	19 October 2012	66.9	66.84	68.25	65.75	68.29	67.75	67.79	69.06	67.6
3082	22 October 2012	68.71	69.31	70.16	68.81	69.71	69.72	68.16	68.95	71.99
3083	23 October 2012	69.07	68.26	68.84	68.54	70.31	68.64	66.91	68.29	69.18
3084	24 October 2012	65.61	67.51	69.47	65.25	66.21	68.39	66.6	68.67	67.87
3085	25 October 2012	64.56	68.23	67.97	65.43	66.97	68.5	64.73	68.21	67.27
3086	26 October 2012	66.12	68.13	69.22	67.24	69.55	68.52	67.54	69.49	68.52
3087	29 October 2012	67.36	66.56	68.6	65.96	69.27	67.53	67.73	67.08	67.63
3088	30 October 2012	67.78	68.4	68.51	67.92	68.46	69.7	66.57	66.01	68.35
3089	31 October 2012	63.9	66.72	68.11	64.13	66.26	67.37	64.99	67.94	66.5
3090	01 November 2012	65.05	65.17	65.49	67.05	65.62	65.23	65.15	66.64	65.82
3091	02 November 2012	65.77	65.36	65.57	67.08	67.56	66.36	64.37	65.69	67.35
3092	05 November 2012	65.42	65.86	64.51	66.29	66.63	65.51	64.25	66.08	65.08
3093	06 November 2012	66.72	66.51	66.2	65.69	68.36	67.73	65.24	67.66	66.36
3094	07 November 2012	64.53	65.29	65.2	65.96	67.53	66.29	65.15	67.26	65.25
3095										

Figure 2: Example of an input file (fictional prices) dealing with natural gas futures for delivery in Zeebrugge.

There is one disadvantage of the rollover concept, unfortunately, and these are the so-called expiry dates. At some point in time, the interpretation of M+1 becomes different. To illustrate, consider that we are dealing with coal futures. Some time for logistics (shipping) is needed in order to deliver coal. As such, one can understand that making a trade for delivery in the next month during the last days of the current month is impossible. The last day when a product can be traded is called expiry date. After an expiry date, the meaning of the second column in our input file (M+1 BID price) is not for delivery in the next month but in fact in the month afterwards (i.e., today plus two months).

One should see in such a case the M+1 as the 'first nearby' month that can be traded on the market. The situation resets itself at the start of a new month meaning that one can construct the absolute month again by taking the quoted date and move one month forward. In order to handle this behavior, for each market and commodity, we have registered the expiry dates in the past and induced rules used to extrapolate expirations in the future. Expiry dates are only one sort of data we collect about the market besides the prices.

The listing below provides a bit of pseudo-code for two macros that are used in the input procedure (focusing on one spreadsheet to read). These macros are used in order to make sure that all input files are stored in tables that have the same layout. The first macro renames the column names with the codes as given in the input file while the second transposes the data so that each record contains one price. Then, the code of each record is split using the underscore character as delimiter and the resulting substrings are used to fill the fields of the record, hereby again detailing the type of price it contains. An example of the end result of the formatting is given in Figure 3. The link with the input file is clear: we can find back the date on which the prices are published, the prices themselves, the period, and other useful information such as absolute dates (called 'start date' in the table, derived from the quoted date and the period). Such information will prove useful during analyses.

	Quoted Date	Period	Observation (ASK, BID, MID)	Subdefinition (BASE, PEAK, OFF-PEAK)	Start Date	Start Time	Unit	Type (PUBLISHED, CORRECTED, ...)	Quote	Creation Stamp	Definition	CET S Time
102557	07NOV2012	M+1	BID		01DEC2012		PENCE/THERM	PUBLISHED	64.53	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102558	07NOV2012	M+1	MID		01DEC2012		PENCE/THERM	PUBLISHED	65.15	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102559	07NOV2012	M+2	ASK		01JAN2013		PENCE/THERM	PUBLISHED	67.53	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102560	07NOV2012	M+2	BID		01JAN2013		PENCE/THERM	PUBLISHED	65.29	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102561	07NOV2012	M+2	MID		01JAN2013		PENCE/THERM	PUBLISHED	67.26	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102562	07NOV2012	M+3	ASK		01FEB2013		PENCE/THERM	PUBLISHED	66.29	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102563	07NOV2012	M+3	BID		01FEB2013		PENCE/THERM	PUBLISHED	65.20	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102564	07NOV2012	M+3	MID		01FEB2013		PENCE/THERM	PUBLISHED	65.25	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102565	07NOV2012	Q+1	ASK		01JAN2013		PENCE/THERM	PUBLISHED	65.44	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102566	07NOV2012	Q+1	BID		01JAN2013		PENCE/THERM	PUBLISHED	65.16	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102567	07NOV2012	Q+1	MID		01JAN2013		PENCE/THERM	PUBLISHED	65.37	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102568	07NOV2012	Q+2	ASK		01APR2013		PENCE/THERM	PUBLISHED	62.92	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102569	07NOV2012	Q+2	BID		01APR2013		PENCE/THERM	PUBLISHED	63.14	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102570	07NOV2012	Q+2	MID		01APR2013		PENCE/THERM	PUBLISHED	63.37	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102571	07NOV2012	Q+3	ASK		01JUL2013		PENCE/THERM	PUBLISHED	62.76	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102572	07NOV2012	Q+3	BID		01JUL2013		PENCE/THERM	PUBLISHED	61.98	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102573	07NOV2012	Q+3	MID		01JUL2013		PENCE/THERM	PUBLISHED	62.56	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102574	07NOV2012	Q+4	ASK		01OCT2013		PENCE/THERM	PUBLISHED	67.93	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102575	07NOV2012	Q+4	BID		01OCT2013		PENCE/THERM	PUBLISHED	68.10	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102576	07NOV2012	Q+4	MID		01OCT2013		PENCE/THERM	PUBLISHED	67.61	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102577	07NOV2012	Q+5	ASK		01JAN2014		PENCE/THERM	PUBLISHED	70.95	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102578	07NOV2012	Q+5	BID		01JAN2014		PENCE/THERM	PUBLISHED	70.95	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102579	07NOV2012	Q+5	MID		01JAN2014		PENCE/THERM	PUBLISHED	70.62	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102580	07NOV2012	Q+6	ASK		01APR2014		PENCE/THERM	PUBLISHED	63.86	08NOV12:11:57:03	Gas Zeebrugge Fictional	
102581	07NOV2012	Q+6	BID		01APR2014		PENCE/THERM	PUBLISHED	63.89	08NOV12:11:57:03	Gas Zeebrugge Fictional	

Figure 3: Each table has the same structure, allowing for widely applicable SAS code.

Once the data reside in the database there is a need to perform a set of quality checks to ensure that all data are reliable and to alert in case of strange behavior. Some of these tests are true sanity checks. For example, on the same publication day ('quoted date' in the table above) you cannot have two records with an M+1 BID price, and not all prices can be zero. Furthermore, if all prices published on a certain day are equal to those of the day before, then something is wrong as well. It may be, for example, that the current day is a holiday and the publishing tool used by the market did not update its values. Or in case of hourly prices we need 24 prices a day except for two days that are impacted by the daylight saving time. If such tests fail an error is generated and logged by the system so that we are informed and can react accordingly. Failure of other quality checks may provide a warning instead of an error.

In total there are 18 quality checks on input data. In some cases a member of the market data modeling team may decide that the quality check should be overruled. For example, suppose that the system reads the prices of today and notices that they are very different compared to those of yesterday. As such, the daily volatility of the market has reached beyond one of our preset thresholds. There is a quality check that will generate an error. However, the input prices may be correct, i.e., the market has behaved very irregular due to bad economic highlights. In such a case, the quality check is overruled and the system unlocks the data to be used for further processing.

Roughly speaking, we can confirm or correct one or more market prices when a quality check fires a warning or error. This interaction with the system happens via a spreadsheet where we identify the uncertain prices and the desired action by our system. There exist also quality checks on our interaction in order to make sure that the database contains the changes as instructed by us. Each data verification and manipulation by a user is logged and reported.

The system is instructed which quality check it has to apply on which (group of) market data. For example, we can instruct applying a specific quality check 'QC_ODD_010' to all price data that belong to the group of futures excluding one special case. Furthermore, each quality check has the same input parameters allowing for one macro that manages the quality checks (when to apply the check, for which data, what options to use, and so on). Albeit that we focus here on input data, there are also quality checks that apply on the output of our system such as price forecasts. For example, some price forecasts have to satisfy some constraints, as we will see in the next section, and we do have a check in place for that. It is always the same SAS macro that takes care of calling the necessary quality checks on input and output and any intermediate step that is needed.

```
%MACRO mcr_mdm_odd_renameColumns(in=, legend=);

  * Get number of columns to be renamed;
  PROC SQL NOPRINT;
    SELECT COUNT(labels) INTO :cnt_labels
    FROM &legend;
  QUIT;

  * Initialise the loop;
  %LOCAL i;
  %LET i = 1;

  %DO %WHILE (&i le &cnt_labels);

    DATA _NULL_;

      * Get the new column name;
      choice = &i;
      SET &leg POINT = choice;
      CALL SYMPUTX("newname", labels);

      * Rename the i-th column;
      dsid = OPEN("work.&in", "i");
      CALL EXECUTE('proc datasets library=work nolist;');
      CALL EXECUTE('modify &in;');
      CALL EXECUTE('rename '||varname(dsid,&i)|| '=&newname;');
      CALL EXECUTE('quit; run;');
      rc = CLOSE(dsid);
      STOP;

    RUN;

    %LET i = %EVAL(&i. +1);

  %END;

%MEND mcr_mdm_odd_renameColumns;

%MACRO mcr_mdm_odd_transformData(in=, out=, expiry=);

  * Sort and transpose so that each record has a day and one quote;
  PROC SORT DATA=&in OUT=&in.0;
    BY quoteddate;
  RUN;

  PROC TRANSPOSE DATA=&in.0 OUT=&in.1;
    BY quoteddate;
  RUN;


```

```

* Add period, observation, and sub-definition;
DATA &in.2;

    RETAIN datetype dateprev;

    MERGE &in.1 (rename=(coll=quote)) &expiry (in=expdate);
    BY quoteddate;

    * datetype = 1 if we have an expiry date;
    * datetype = 2 if expiry date passed and in same month;
    * datetype = 0 if no special treatment is needed;
    IF expdate THEN datetype = 1;
    IF datetype = 1 and dateprev ne quoteddate THEN datetype = 2;
    IF month(dateprev) ne month(quoteddate) THEN datetype = 0;

    * Define the period according to the value of datetype;
    IF datetype = 2 THEN period =
        scan(_NAME_,3,'_')||"+"||compress(sum(scan(_NAME_,5,'_'),1));
    ELSE period = scan(_NAME_,3,'_')||"+"||scan(_NAME_,5,'_');

    * Define observation and sub-definition;
    observation = scan(_NAME_,6,'_');
    subdefinition = compress(scan(_NAME_,2,'_'), 'X');

    * Prepare for next iteration;
    dateprev = quoteddate;
    KEEP quoteddate period subdefinition quote observation;

RUN;

%MEND mcr_mdm_odd_transformData;

```

DATA ANALYTICS

Once the prices reside in the database and are marked as valid, they can be used for data analytics purposes. Interesting questions related to our prices concern among others:

1. Correlation between different markets (e.g., how do prices in France influence the Belgian market?);
2. Volatility of the markets (e.g., used for risk management and asset allocation);
3. Future price levels (e.g., what are the best estimates of the prices in the future?).

By observing that our price data are in essence random variables measured at discrete steps in time, we can use time series analysis to assist in answering questions related to correlations, trends, volatility, seasonality, and much more. In some cases we can also borrow techniques from financial modeling or resort on plain statistics such as probability estimation and predictive modeling. However, such models do make assumptions on the underlying data and we have observed that energy commodity markets have their own special characteristics. It is therefore not surprising that in the last years a large number of specialized data analysis methods have been proposed.

In the introduction and background sections we have explained the importance of having accurate price forecasts (cf. future contracts and the explanation about gas-turbine power plants). Several models exist and are available in our market data module. We will focus here on a basic methodology to derive a 'forward curve'. This is a forecast of prices at which we believe that the market is willing to do business. The forecast has a monthly granularity. Clearly, such a forecast can not only be used for trading purposes but also for risk management and retail applications.

An example of a monthly forward curve is given in Figure 4. Seasonality can be clearly identified with high prices in winter and low prices in summer. It is important to note that such a graph can be stored in a table that has the same structure as the input data sets (cf. Figure 3). We maintain each curve in the database and in case a curve is re-ran (the most recent one or a set in the past) we apply a deletion id concept to the old curve in the sense that the curve is set to inactive, including the timestamp when this happened. Quality checks are also applied.

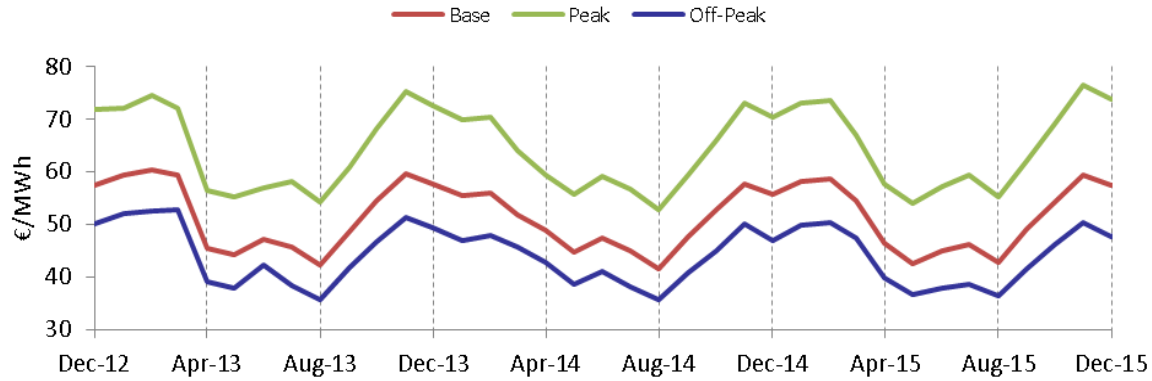


Figure 4: Monthly forward curve for Belgian electricity (fictional price, fictional shape and trend).

The starting point of the forward curve methodology is the most recent set of futures published by the market. Those prices may be as in the left table in Figure 5. Since the first three months are published by the market, we can use these as the first three prices in our curve. Indeed, those prices are the best estimates we have since we consider the market as the true distribution underlying the prices. But note that, if we would afterward use the published quarters and years, then the monthly curve would become piecewise linear, with non-decreasing length of the linear segments. Each segment coincides with a future traded on the market and has a constant price in the months that it covers. But such an estimate is not useful for our purposes since, in the end, all prices in a year will be equal. So we need to find a way to divide yearly prices into quarterly and quarterly into monthly prices, a process often referred to as 'shaping'. We will do the shaping by investigating the closing prices of futures published by the market in the past.

For each day in the past we look at the quarterly prices. In case all three months of the quarter have been published as well, we compute for each month its 'weight' in the quarter by dividing the monthly price by the quarterly. These ratios reflect the shape of the months in the quarter, for that day in the past, where a ratio above one means that the month gets a price higher than the quarterly price. To aggregate over all days in the past, we take a weighted average of these ratios (keeping the quarter/month combination fixed) assigning low weight to ratios coming from prices published a long time ago and slightly increasing the weight when the prices become more recent. The result will be a coefficient matrix that maps each month with a weight to its quarter (see middle table in Figure 5). A coefficient reflects the relation between a parent price and its child as occurred on average in the past. The same procedure can be applied for the distribution of quarters in a year (also shown in Figure 5).

Future	Price	From	To	Coeff	Year	Quarter	Price
M+1	52.00	Q1	M01	1.01327	Y+2	Q1	53.404 (= 50.31 * 1.06149)
M+2	55.50	Q1	M02	1.02413	Y+2	Q2	45.965 (= 50.31 * 0.91364)
M+3	53.90	Q1	M03	0.96260	Y+2	Q3	47.230 (= 50.31 * 0.93879)
Q+1	55.18	Q2	M04	1.01098	Y+2	Q4	54.641 (= 50.31 * 1.08608)
Q+2	49.57	Q2	M05	0.97273			
Q+3	50.07	Q2	M06	1.01629			
Y+1	52.33	Q3	M07	0.99879			
Y+2	50.31	Q3	M08	0.95076			
Y+3	50.28	Q3	M09	1.05045			
		Q4	M10	0.96361			
		Q4	M11	1.03900			
		Q4	M12	0.99739			

Quarter	Month	Price
Y+2 Q1	M1	54.112 (= 53.404 * 1.01327)
Y+2 Q1	M2	54.692 (= 53.404 * 1.02413)
Y+2 Q1	M3	51.406 (= 53.404 * 0.96260)

From	To	Coefficient
Y	Q1	1.06149
Y	Q2	0.91364
Y	Q3	0.93879
Y	Q4	1.08608

Figure 5: Example of a simple forward curve builder. Left: input prices of today. Middle: coefficient matrices (quarter to month and year to quarter). Right: shaping of Y+2 in its four quarter and additional shaping of the resulting first quarter in its three months.

Once these coefficient matrices are computed they can be applied on the prices of today in order to construct the monthly forward curve. First a yearly price is decomposed in four different prices, namely one for each quarter. The way to do this is to consider the price of the yearly future and multiply it with the coefficients we have learned from the historical data (displayed at the top of this page). We have illustrated this with the Y+2 price in the top right table in Figure 5. The Q1 price is computed by multiplying the yearly price with the coefficient for the first quarter. And similarly for the other three quarters. In a second step, the quarterly prices we have just constructed are shaped further in three different prices for the months the quarter covers, as is also illustrated in the figure (Q1 of Y+2 is shaped). Doing this for all four quarters we obtain an estimate of the price for each month in Y+2. And finally, if we apply this methodology for each future product we have (i.e., also for Q+1, Q+2, Q+3 and Y+3 in the left table in Figure 5) the base curve in Figure 4 is obtained.

Assuming that the futures in the market already contain seasonality (e.g., winter quarters are more expensive than summer quarters) the forward curve will reflect these cyclic patterns. The same can be said of a trend: if the market prices are increasing or decreasing in the future, the curve will naturally reflect this pattern.

Note that by construction the average of the three monthly prices is the quarterly price. And the average of the four quarterly prices is equal to the published yearly price. This property is desirable since

1. We have to respect any price published today by the market since it is reality;
2. It is inherently connected to the market.

Indeed, for example, if the average price of three monthly futures is cheaper than the quarterly price, then one would buy those monthly futures and sell them directly as quarterly at a higher price. This is called non-arbitrage.

The outlined methodology is very simple since it depends solely on counting and averaging historical prices. No optimization routines are needed and the process can run in a few minutes when coded efficiently. The only parameters are the weights used to reduce the influence of historical data. However, the methodology explained above does not always satisfy the non-arbitrage property. Assuming that the prices of today are those of a regular day in November 2012, the market provides us with a price for Q1 2013 (Q+1) and for January and February 2013 (M+2 and M+3, respectively). The latter two months have to stay as they are so we are only allowed to construct March by applying the coefficient method. As such, the average of these three months is not guaranteed to be the quarterly quote. This defect can be solved by re-normalizing: $\text{March} = \text{M+4} = [\text{Q+1} - (\text{M+2} + \text{M+3})] / (\text{M+2} + \text{M+3})$.

In case data is available with finer granularity, such as hourly prices, we can use them to derive an hourly forward curve from the monthly curve we have just constructed. For example, for Belgian electricity there exists a day-ahead market (cf. spot trading as discussed in the background section). This market publishes each day in the afternoon a price per hour for the next day. Historical prices are available so one can think of applying a similar counting and averaging scheme as explained above. However, the problem becomes more difficult since daily hourly prices can be very irregular; hereby questioning the usefulness of averaging hourly prices over days.

DASHBOARDS AND OTHER OUTPUT

The market data module is configured to run autonomously three times a day, namely during the morning, afternoon, and evening. In each 'run' the system knows which market prices to collect and what analyses to perform. At the same time, we instruct it with output options; all done via spreadsheets. Figure 6 shows a possible set of options for running and dashboarding some forward curves (fictional).

	A	B	C	D	E	F	G	H	I	J	K	L	
	Short Definition	Definition	Type	Curveenddate	Morning Curvedate Start	Morning Curvedate End	Morning Run	Morning Publish	Noon Curvedate Start	Noon Curvedate End	Noon Run	Noon Publish	Ev Cu St
2	DC_FWD_POWERBE	Belgian Power	MONTHLY	31/12/2016	LAST	LAST	NO	NO	LAST	LAST	NO	NO	LA
3	DC_FWD_POWERBEH	Belgian Power Hourly	HOURLY	31/12/2016	LAST	LAST	NO	NO	LAST	LAST	NO	NO	LA
4	DC_FWD_POWERNL	Dutch Power	MONTHLY	31/12/2015	LAST	LAST	NO	NO	01NOV2012	LAST	YES	YES	LA
5	DC_FWD_POWERFR	French Power	MONTHLY	31/12/2015	LAST	LAST	NO	NO	01NOV2012	LAST	YES	YES	LA
6	DC_FWD_POWERDE	German Power	MONTHLY	31/12/2015	LAST	LAST	YES	YES	01NOV2012	LAST	YES	YES	LA
7	DC_FWD_GASNB	Gas NBP	MONTHLY	31/12/2015	LAST	LAST	YES	YES	LAST	LAST	NO	NO	LA
8	DC_FWD_GASZB	Gas Zeebrugge	MONTHLY	31/12/2015	LAST	LAST	YES	YES	LAST	LAST	NO	NO	LA
9	DC_FWD_GASZB_EUR	Gas Zeebrugge Euro/MWh	MONTHLY	31/12/2015	LAST	LAST	YES	YES	LAST	LAST	NO	NO	LA
10	DC_FWD_CO2EUA	CO2 EUA	MONTHLY	31/12/2015	LAST	LAST	YES	YES	LAST	LAST	NO	NO	LA

Figure 6: Configuration what to run and export.

Depending on the hour when the system starts, it knows whether it has to fetch instructions for the morning run (columns E-H) or noon run (columns I-L) or evening run (not shown). The file also has four columns for an ad-hoc run that we can start at any time we want. For each run, the first two set of columns define the days for which a curve should be constructed. In case of the 'LAST' value, the system uses the most recent data. The third option defines whether the curve should be constructed and the last column of a run states whether an output has to be produced.

In the figure we see that the most recent gas curves are ran in the morning and they should all be published to our internal clients and dashboards. In the noon run we provide some update of the electricity forward curves, and we run them from the first of November up to now. Similar sheets exist for the raw input prices and any other data we have.

In case some data have been marked to be published, the system has to export the requested (part of the) data to some server location. In order to do so, the system reads its spreadsheet with distribution rules that instruct, for each market data, where to copy it to and in which format. Typical formats are csv, xls and zip. Internal clients (either human or system) poll the server for updates of the files and refresh their tools with the latest prices when available.

Exporting the database to our internal clients is not the only type of exporting. Our SAS tool also creates dashboards (intranet pages). One of these dashboards is specifically designed for the market data modeling team. More specifically, it shows the results of a system run. After a run we inspect the dashboard. It may be, for example, that a quality check requires our attention. The dashboard also shows the evolution of the forecasts so that we know where the market is heading to and how it compares to the previous days. Other dashboards are created on request of the internal or external client. Due to the confidential nature, unfortunately, we cannot provide any example in this paper.

CONCLUSION

In this paper we have outlined the 'market data module' that is used in EDF Luminus. It reads all data related to the markets by a single set of SAS macros. We can control the complete tool by spreadsheets containing configurations of the markets, quality checks, forecast algorithms and options, re-distribution rules to internal customers, and much more. All this happens without making any change in the code.

As such, to conclude, we believe that the market data module is an interesting showcase about how SAS can be used to design a system that goes from raw data management to advanced analytics, running multiple times a day and this fully automated. The tool has a lot of value in EDF Luminus. We hope that this paper motivates other users to implement full processes with SAS, from the very first input to the very last output.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Stijn Vanderlooy
 Enterprise: EDF Luminus
 Address: Markiesstraat 1
 City, State ZIP: 1000 Brussels
 Work Phone: +32 222 91976
 E-mail: stijn.vanderlooy@edfluminus.be
 Web: www.edfluminus.be

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.