Advanced Project I

The German SPOT and Day Ahead Energy Market 2015-2016

Author:
Mark C Koerner

Instructor:
Prof. Stefan Kettemann

June 9, 2019

JACOBS UNIVERSITY

Executive Summary

The purpose of this report is to examine the hypothesis that Residual Load is highly correlated with the energy market prices, which was previously raised in the literature [1] [2]. In particular, the relationship between both Day-Ahead and Intraday weighted average market prices and aggregated energy generation data is assessed by computing the correlation coefficients as well as visually using heatmaps. After collecting the necessary price and energy generation data, the individual dataframes were first split into hourly and quarter-hourly intervals when necessary and then combined to form two final dataframes. Ultimately, Day-Ahead and Intraday hourly auction prices show a significant positive correlation coefficient with Residual load, which can be attributed to the similar daily patterns shown in the heatmaps. However, quarter-hourly continuous Intraday prices show little correlation with Residual and other Load variables, which seems to be due to the intra-hour price patterns.

Contents

1 Introduction

The share of renewable energies in the German energy grid has increased constantly in recent years, mainly driven by price reductions, government policies and an increased environmental consciousness by Germany's citizens. While currently hovering around 36% [3], the German government targets an increased share between 40 and 45% by 2025 and further increases in subsequent decades[4]. The most popular renewable energy sources have been wind and solar energy as their upfront and installation costs have seen a steady decline. Afterwards, their low marginal cost for energy production has resulted in lower spot energy prices. Wind and solar energy plants themselves provide limited flexibility and low mechanical inertia, however, meaning that they require further measures to stabilize demand-supply balance and thus the energy grid [2].

As a result, the average energy market prices should be fairly dependent on the mix of flexible versus non-flexible energy sources, which is what this report aims to examine. Before introducing the data, Section 2 gives an overview over the German energy market and some relevant definition in the energy sector. Section 3 highlights how the necessary data was collected and Section 4 describes the Data Cleaning and Data Manipulation processes. Finally, Section 5 contains the Methodology and Data Analysis Results and Section 6 offers the Conclusion and a further outlook.

2 Background

As a first step, I will provide some background on the German Energy Market and its current implementation. In 1996, the European Union adopted the first iteration of liberalization directives, ordering the member states to open their energy markets for competition and moving away from its previous monopolization starting in 1998 [5]. Over time, further government directives, participation incentives and market mechanisms have led to an energy market that fully takes advantage of basic economic principles to balance supply and demand. Suppliers can offer their produced energy on the market and companies can bid for said energy supply to satisfy their demand.

The market in Germany allows companies to buy or sell energy in multiple stages. EPEX spot, the main European power exchange, offers bidding options for single hours, multi-hour bloacks, and quarter-hour intervals at different distances from the time in question. The Day-Ahead auction takes place at 12:00 the day before and only allows for trading of blocks of one or multiple hours [6]. This particular auction functions on market-clearing principles, meaning that the price for every transaction for a specific hour is based on the last transaction that is completed [7]. The Intraday Auction also takes place the day before but at 15:00, and allows for trading of all previously mentioned intervals [8]. In contrast to the Day-Ahead auction, it is not bound to market-clearing principles and thus allows for individual bidding principles. The Intraday Auction closes after an hour at 16:00, after which the Continuous Intraday market allows for transactions of quarter-hour energy blocks up until 30 minutes[1] before the actual time interval. Thus, the Intraday market is often seen as a fine-tuning mechanism for individual com-

[1] It was 45 minutes before July 2015. It has since been reduced to 5 minutes, but this change is not relevant for this project.

pany's load profiles after the main purchase in the Day-Ahead market [7].

Supply-Demand balancing on the energy market is necessary for the stability of the energy grid, and has recently been challenged due to the inherent fluctuations in energy consumption as well as energy production from renewable energy sources. With the inherent volatility and seasonality of renewable energy sources such as wind and solar energy and a lack of sufficient energy storage, market mechanisms and other sources of flexibility become vital for a well-functioning grid. Traditional energy sources have typically been able to increase or decrease production in response to energy consumption trends or provided enough mechanical inertia to keep the grid stable, whereas Wind and Solar power plants either produce energy in accordance with the weather conditions or have to be shut off entirely. More recently, Demand Side Management has also seen a rise in popularity, meaning companies can manage their energy consumption according to the market and therefore profit from imbalances through flexibility [9].

As mentioned above, the electricity prices ultimately depend on the balance of energy supply and demand. Although solar and wind energy show large fluctuations in their production profile, they are non-flexible sources of energy, leaving it to the other flexible energy sources to balance the market. Thus, according to Pagnier and Jacquod (2018) [2], spot market prices are usually higher when the share of flexible energy sources of overall energy production is higher. The share of flexible energy sources is usually called *Residual Load*. In equation form, this statement corresponds to

$$Residual\ Load\ =\ Total\ Load - Renewable\ Energy\ Load. \tag{1}$$

Although some correlation between market prices and Residual Load are to be expected, the reported correlation coefficients of up to 0.89 in the literature suggest a very high positive correlation [2]. In other words, the correlation coefficients indicate that market prices generally increase when the amount of Residual Load increases. Investigating this claim will be the focus of this report.

3 Data Collection

In order to compare energy generation to market prices, it was necessary to find data sources for energy generation in Germany broken down into its components, as well as market prices for both the Day-Ahead and Spot markets. As the liberated market depends on a certain degree of transparency, most of the data should be fairly easily accessible.

For the energy data, The ENTSO-E transparency platform [10] provides aggregated load and generation information by production type for all of Germany in 15 minute intervals and allows users to easily export yearly data. Thus, I downloaded .csv files for both generation and load data for 2015 and 2016 respectively, which were then merged later on.

As for the market data, Intraday and Day-Ahead market data is available for payment from the Epex Spot website. In this case, both were provided by Prof. Kettemann. The provided Excel and .csv files contained a multitude of different views and excerpts of the data, and were hence reduced to the relevant parts in the next step. Ultimately, the dataset included hourly Day-Ahead and Intraday auction data, and quarter-hourly data from continuous trading.

4 Data Preprocessing

As a first step, I decided to consolidate the energy generation, Day-Ahead and Intraday data into individual, complete tables respectively. This step was accomplished by simply concatenating the 2016 data at the end of the 2015 data or by identifying and isolating the relevant tables. All dataframes maintained a fairly similar structure, namely being indexed by integer timesteps with one or more columns referencing the actual time of observation and the rest of the columns featuring numerical observations. After taking a closer look at the resulting dataframes, however, I also noticed a couple of differences between them. First of all, the formatting for the timestamp of each observation was very different. Secondly, some of the data was in an hourly or quarter-hourly format, or both. Lastly, the sources differed in how they handled daylight savings time.

4.1 Time Formatting

As previously mentioned, every source offered a different style of formatting for the timestamps. The generation data had a column called "Time (CET)", which contained the starting and ending date and time separated by a hyphen. The day-ahead spot data contained a column with just the starting date and time in addition to a separate column denoting the starting and ending hour separated by a hyphen. Lastly, the intraday dataframe had time encoded in a string with different formats for quarter-hourly and hourly data. The first part of the quarter-hourly string was the hour of the day initialized at 1, the second part was just 'qh' for quarter hour with the integers 1 through 4 denoting the 4 possible quarter-hour segments. As an example, 24qh4 would translate to 23:45-0:00. The hourly intraday data is simply denoted by the first, hourly part of the string. In that case, 24 symbolizes the hour from 23:00 to 0:00. The intraday dataframe also features a non-encoded date column.

In the beginning, I focused most of my efforts in converting the string encodings into proper datetime format, which turned out to be tedious. Even with separating out the different parts of the encoding, the transformation would require a multitude of string operations. In addition, the two hours of the year that were repeated due to daylight savings time were encoded 3A and 3B, which added another obstacle to automating the transformation process. Thus, I decided that instead of joining the dataframes on the datetime column, to just join the dataframes on the integer index after making sure they were lined up perfectly and to then use the starting time from the generation data as a timestamp.

4.2 Further Data Cleaning

In addition to the formatting issues, the sourced dataframes also contained missing or dirty data. For example, the columns "Fossil Oil" and "Fossil Oil Shale" of the Generation data are missing for the entirety of 2015 and 2016, which allowed me to just remove them entirely. Other columns such as "Fossil Peat" and "Marine" in the generation dataframe as well as "Index Price" and "ID3 Price" in the intraday dataframe contained missing values for some of the observations. The likely reason for this is the categories being added at some point during that period, as the missing values are all found at the beginning of the period. These columns were at most used for some aggregations that are discussed in Subsection 4.3. "Index Price" was considered to be used as the price indicator for the intraday market, but the column "Weighted Average Price" features the same values and does not contain NaN's so was used instead. In general, all observations that did not contain numerical values when they should were coerced into the numeric type, which automatically converts all non-numerical data types into NaN format and in turn allows for much easier filtering. Besides the time and date columns, all variables across all sources should only contain numeric values.

Checking for missing values also ultimately helped solve another issue, namely the ability to join the three dataframes. When I first attempted to join them, it became clear that the dataframes were of different lengths. Specifically, the generation dataframe included 8 extra rows, or 2 extra hours worth of data. When checking for NaN values, I realized that the discrepancy in length was caused by the differences in handling daylight savings time. Instead of removing the skipped hour in March, the dataframe included 4 rows of missing values each time. Removing those rows allowed me to safely join all dataframes on the reset index.

4.3 Combining the Sources

Ultimately, I decided to create two separate combined dataframes, one for quarter-hourly data and one for hourly data. The hourly dataframe contained prices for both Spot and Intraday markets, as well as aggregated generation data by summing the 4 quarter-hours for every hour. The quarter-hourly dataframe only contained the given intraday prices as well as the generation data, as the bidding on the Spot market is constrained to hourly intervals. I decided to move forward with both time intervals in order to not lose any of the given information as well as to allow for the ability to compare trends between both. As part of the process, I created the new hourly Energy Generation dataframe, and split the Intraday dataframe into quarter-hourly and hourly components based on the label difference, i.e. every row whose time encoding contained 'qh' was assigned to the quarter-hourly dataframe and the rest to the hourly one. In the joins I only included the chosen Price columns for each market dataframe, whereas I included all columns for the Generation dataframe in case I wanted to individually check the market price correlations for a particular energy source. The generation data did not include Renewable Energy and Residual Load aggregations, however.

The process of creating the Renewable Energy and Residual Load variables is fairly simple. It is achieved by first summing the Solar and Wind Energy observations for every timestamp to create the Renewable Load variable, and then by subtracting the Renewable Load from the Total Load column for the Residual Load variable. This process was followed for both dataframes. In addition to those two extracted features, I also decided to assess the impact of the Day-Ahead Total Load Forecast as that is used as one of the pieces of information at the time of purchase or sale for both auctions. Instead of looking at the raw Load Forecast, I decided to use the difference between forecasted and actual load, i.e. $LoadDiff = LoadForecast - ActualLoad$. This step completed the Data Preprocessing work, and Figure 1 shows a snippet of the final dataframes used for the Analysis.

Delivery Date	Solar [MW]	Wind Onshore [MW]	Wind Offshore [MW]	Intraday WA Price [EUR]	Day-Ahead Price [EUR/MWh]	Renewable Load	Residual Load	Load Diff
2015-01-01 00:00:00	0.0	32512.0	2066.0	16.61	25.02	34578.0	133810.0	235.0
2015-01-01 01:00:00	0.0	33190.0	2065.0	7.40	18.29	35255.0	127406.0	-2783.0
2015-01-01 02:00:00	0.0	34160.0	2056.0	9.17	16.04	36216.0	121823.0	-2787.0
2015-01-01 03:00:00	0.0	34208.0	2071.0	13.62	14.60	36279.0	117870.0	-189.0
2015-01-01 04:00:00	0.0	34574.0	2079.0	11.80	14.95	36653.0	115959.0	1963.0

(a) Hourly Dataframe

Start_Time	Solar [MW]	Wind Onshore [MW]	Wind Offshore [MW]	Intraday WA Price [EUR]	Renewable Load	Residual Load	Load Diff
2015-01-01 00:00:00	0.0	8113.0	520.0	55.55	8633.0	33792.0	530.0
2015-01-01 00:15:00	0.0	8092.0	517.0	62.99	8609.0	33412.0	391.0
2015-01-01 00:30:00	0.0	8161.0	514.0	65.71	8675.0	33393.0	-167.0
2015-01-01 00:45:00	0.0	8146.0	515.0	69.46	8661.0	33213.0	-519.0
2015-01-01 01:00:00	0.0	8183.0	515.0	51.90	8698.0	32532.0	-520.0

(b) Quarter-hourly Dataframe

Figure 1: The first 5 observations for both final dataframes.

5 Data Analysis & Discussion

The Data Analysis component for this project was comprised of two components: correlation and visual analysis. In particular, the goal was to further examine the relationship between Residual Load, Renewable Load and the market prices for the hourly and quarter-hourly data.

5.1 Methodology

In order to assess the relationships between the different variables with some context, I first calculated the correlation between the prices and the generation data and then compared the values to yearly heatmaps of each variable. The heatmaps were generated by aggregating the data by both time-of-day and day of the year, and using those indexes as the axes. Thus, the heatmaps show the relative values for each hour or quarter-hour of the day for the entire year as a pixel, and allows the observer to see seasonal trends. As the correlation coefficient is a measure describing the relationship between two variables, comparing the trends over time for both variables should allow us to confirm and further assess the correlation coefficient.

The correlation coefficient is calculated as

$$CC_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \tag{2}$$

where cov(X,Y) is the covariance between the two variables and σ denotes the Standard Deviation. The correlation coefficient is bound to the interval [-1, 1], with values close to -1 or 1 expressing a large negative or positive relationship and values close to 0 expressing no significant relationship between the variables.

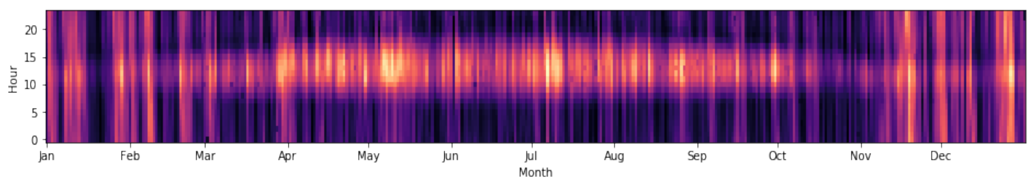
5.2 Hourly Data

First, I wanted to look at hourly data as it contained both Intraday and Day-Ahead auction prices. The relevant correlation coefficients are shown in Figure 2. Similar to previous findings, the correlation between Residual Load and prices for both Day-Ahead and Intraday auctions exhibit a strong correlation, albeit slightly lower than 0.89. Both prices show some negative correlation with Renewable Load, meaning that more Renewable Energy generation should generally result in lower prices. Lastly, the Load Difference does not seem to have a significant impact, as the correlation is very close to 0. It is interesting to note, however, that the Intraday Price has smaller correlation values for all variables besides Load Difference, hinting that the later auction timing might lead to price corrections due to more information being available.

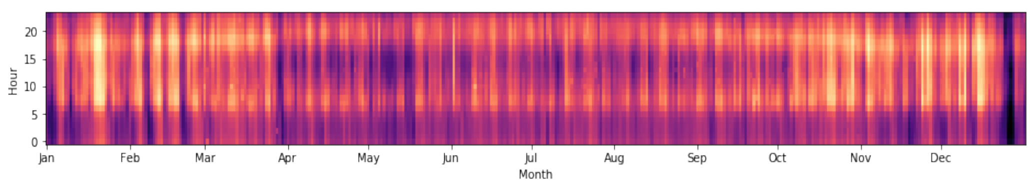
	Renewable Load	Residual Load	Load Diff
Day-Ahead Price [EUR/MWh]	-0.415488	0.842814	-0.057899
Intraday WA Price [EUR]	-0.399821	0.799418	-0.081701

Figure 2: Correlation Coefficients between Intraday and Day-Ahead prices and Load data.

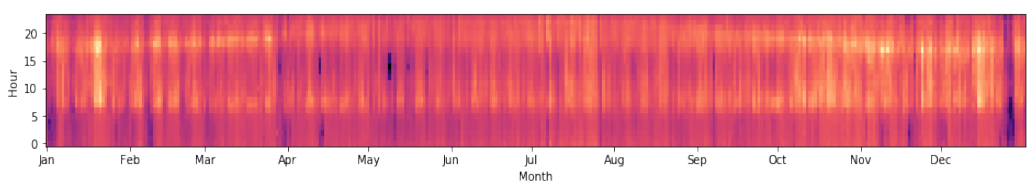
The heatmaps shown in Figure 3 visually support the above-mentioned correlation coefficients. In addition, the heatmaps show clear trends that allow for further insights. For Renewable Load, the graph shows a clear seasonal trend for both Solar and Wind Power, as the Solar component is bound to daylight hours with its highest output in the summer as shown by the lighter orange and red colors, and the wind component with its highest output in the winter. Besides the late night hours, the Residual Load heatmap looks almost like the inverse of the Renewable heatmap, as Residual Load is highest in the early morning and evening hours as well as throughout winter days with little to no wind energy generation. The Price heatmaps look very similar and both seem to mostly mirror the Residual Load heatmap albeit with less variance as evidenced by the little to no change in color. All three heatmaps exhibiting the same trend, however, visualizes the underlying cause of the high correlation coefficient.



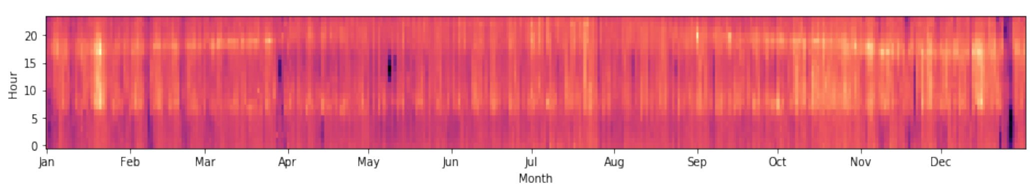
(a) Renewable Load Heatmap



(b) Residual Load Heatmap



(c) Heatmap of Day-Ahead Price



(d) Heatmap of Intraday Price

Figure 3: Heatmaps for hourly Load and Price Data. Lighter colors correspond to higher values.

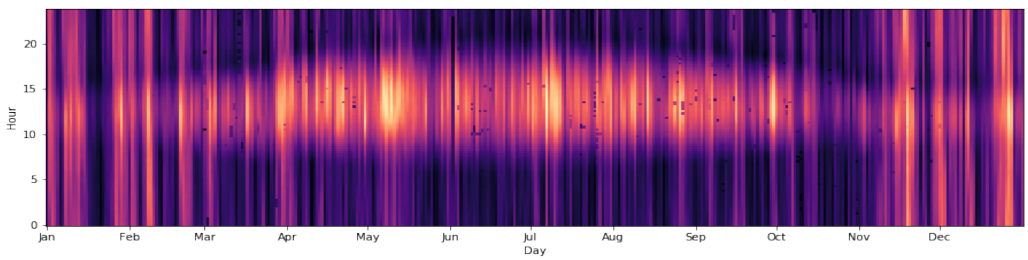
5.3 Quarter-Hourly Data

The Quarter-Hourly price data is only comprised of auction and continuous trading transactions on the Intraday market. Again, these transactions are thought to be used to either smoothen the purchased energy profile for demand-side buyers or to to sell excess generated energy or buy back energy due to worsened generation forecasts. Generally, it can be seen as the market adapting to further information or taking advantage of the shorter time intervals to further match consumption and purchase profiles. Thus, it is rather unsurprising to see low correlation coefficients as shown in Figure 4. Although the Load Diff coefficient is still fairly insignificant, it is interesting to see it being the highest absolute value coefficient, meaning it has the most significant relationship with the Intraday price. It is also interesting to note that the relationship is positive, meaning that when the Load Forecast is increasingly higher than the Actual Load, the Intraday Price also tends to increase. Overall, however, the relationships between quarter-hourly Price and all Load variables are not significant.

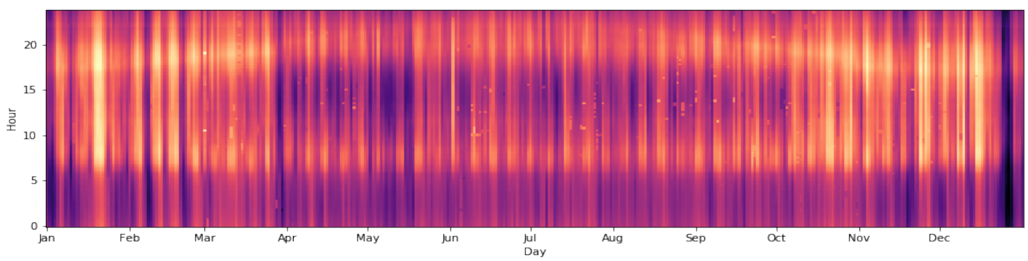
	Renewable Load	Residual Load	Load Diff
Intraday WA Price [EUR]	0.000171	-0.072217	0.126189

Figure 4: Correlation Coefficients between quarter-hourly Day-Ahead prices and Load data.

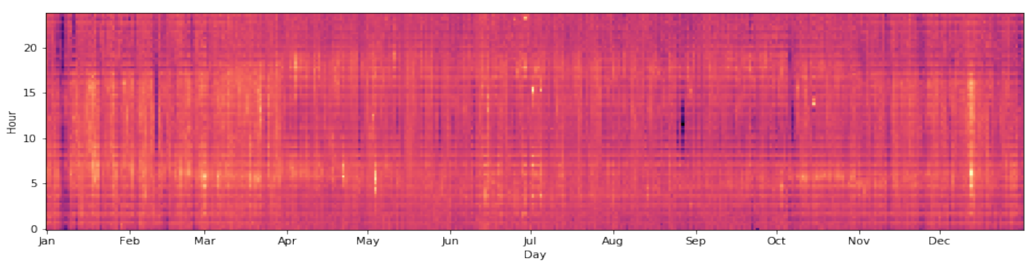
Again, the heatmaps offer some clues on the reason why the correlation coefficient are so low. The Renewable Load and Residual Load heatmaps as seen in Figure 5 show the same trends as the hourly data just on a different scale, which makes sense given that they are based on the same underlying data but on a different scale. The Intraday Price heatmap still shows some of the underlying trend with respect to the early morning and evening hours, yet also reveals hourly trends. In the early morning hours and the evening hours, the last quarter-hour generally shows the highest price as compared to the rest of the hour, whereas in the morning and late night hours it is the lowest and the first quarter-hour seems to be the most expensive. This trend points towards the suspicion of buyers attempting to smooth out their purchasing profile to an extent. Figure 6 offers another visualization of the trend as an average of the entire dataset. Even though early morning hours change throughout the year as evidenced by the heatmaps, the trend is still clearly visible when only looking at the red and blue bars. From 21:00 to 5:00 and 11:00 to 17:00, the last quarter-hour is the most expensive, whereas the rest of the day the first quarter-hour is the most expensive. Thus, the Intraday Price seems to follow this trend much more closely as opposed to the Residual Load pattern, explaining the low correlation scores.



(a) Renewable Load quarter-hourly Heatmap



(b) Residual Load quarter-hourly Heatmap



(c) Heatmap of quarter-hourly Intraday Price

Figure 5: Heatmaps for quarter-hourly Load and Price Data. Lighter colors correspond to larger values.

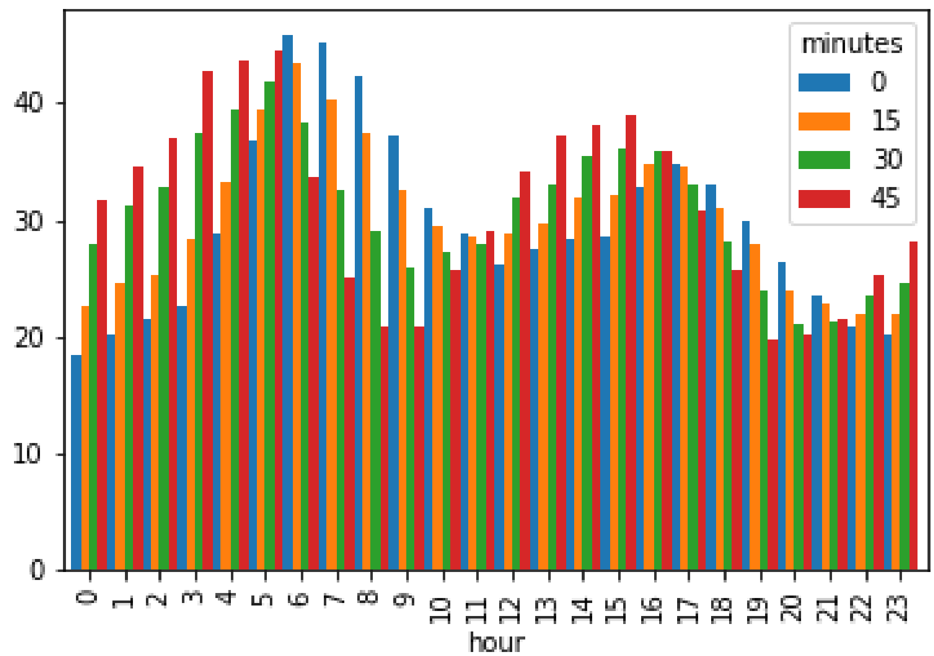


Figure 6: Day Profile for Intraday Continuous Prices. The given minutes in the legend denote the starting time for each quarter-hour.

6 Conclusion & Outlook

Overall, the correlation coefficients and visualizations comfirm the hypothesis that some energy market prices are highly correlated with the energy generation of flexible Residual Load sources. This project demonstrates, however, that this statement is only true for the full-hour transactions in the Day-Ahead and Intraday Market. The quarter-hourly show no correlation with the different types of energy generation but rather follows an hourly pattern that can be linked to the smoothing of production or consumption curves. Lastly, the Load forecasting error seems largely insignificant, but does share the largest absolute correlation coefficient with continuous Intraday price, which point to the market reacting to more accurate information closer to the transmission time.

In future projects, it would be interesting to further investigate the cause of the hourly trends in the quarter-hourly Price data. in order to establish a definite link between the smooth transaction curves and its effect on prices for smaller time interval energy blocks. In addition, I would also like to look at the price patterns of larger time blocks on the Auction markets, and if their pattern are similar to the hour-block patterns or inherently different. Finally, it would be interesting to add more years of data, look at the evolution of the correlation coefficient over the years and then compare it to the installed solar and wind energy capacity.

References

[1] P. Gifvars, *Structural modeling of electricity spot prices, using a residual load and coal generation cost approach*. PhD thesis, Lund University, 2013.

[2] L. Pagnier and P. Jacquod, "How fast can one overcome the paradox of the energy transition? a predictive physico-economic model for the european power grid," *arXiv preprint arXiv:1706.00330*, 2017.

[3] "Zsw und bdew zum anteil der erneuerbaren energien am stromverbrauch 2017." `https://www.zsw-bw.de/presse/aktuelles/detailansicht/news/detail/News/zsw-und-bdew-zum-anteil-der-erneuerbaren-energien-am-stromverbrauch-2017.html`.

[4] F. M. for Economics Affairs and Energy, "The energy transition." `https://www.bmwi.de/Redaktion/EN/Dossier/energy-transition.html`.

[5] F. Gouardères, "Internal energy market," May 2019.

[6] E. SPOT, "Products: Day-ahead auction." `https://www.epexspot.com/en/product-info/auction/germany-luxembourg`.

[7] N. Kraftwerke, "What does intraday trading mean?." `https://www.next-kraftwerke.com/knowledge/intraday-trading`.

[8] E. SPOT, "Products: Intraday auction." `https://www.epexspot.com/en/product-info/intradayauction/germany`.

[9] D. E. A. (dena), "Integrating renewable energy sources – ensuring the security of supply." `https://www.dena.de/en/topics-projects/energy-systems/flexibility-and-storage/`.

[10] ENTSO-E, "entsoe transparency platform." https://transparency.entsoe.eu/, 2019.

A Appendix 1: Code

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:



import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.axes as ax
import seaborn as sns



# In[199]:



#load generation and load data
generation_2015 = pd.read_csv('Actual Generation per Production Type 201
generation_2016 = pd.read_csv('Actual Generation per Production Type 201
load_2015 = pd.read_csv('Total Load - Day Ahead   Actual_201501010000-2
load_2016 = pd.read_csv('Total Load - Day Ahead   Actual_201601010000-2

# make a 2015-2016 df
# concat only 2015 and 2016 generation and load dfs
df_gen1516 = pd.concat([generation_2015, generation_2016], ignore_index=
df_load1516 = pd.concat([load_2015, load_2016], ignore_index=True)
print(df_gen1516.shape)
```

```
print(df_load1516.shape)

#concat into one df, drop redundant columns
df_1516 = pd.concat([df_load1516, df_gen1516], axis=1)
df_1516.drop(columns=['Area', 'MTU'], inplace=True)
print(df_1516.shape)

#split the time field into start and end time fields
time_split = df_1516['Time (CET)'].str.split(' - ', expand=True)
df_1516['Start_Time'] = time_split[0]
df_1516['End_Time'] = time_split[1]
df_1516.drop(columns=['Time (CET)'], inplace=True)

#df_1516.head()
#can still get rid of unnecessary/NaN-valued generation type columns bu

df_1516[-26876:-26868]


# In[200]:


# delete empty rows
df_1516 = df_1516.drop([8360,8361,8362,8363,43308,43309,43310,43311], a
df_1516.reset_index(inplace=True, drop=True)
print(df_1516.columns)
df_1516[43300:43312]
```

```
## Intraday Prices

# In[201]:


intraday_2015 = pd.read_excel('intraday_results_germany_austria_2015.xls
intraday_2016 = pd.read_excel('intraday_results_germany_austria_2016.xls

print(intraday_2015.shape)
intraday_2015.head(3)


# In[202]:


intraday_df = pd.concat([intraday_2016, intraday_2015], axis=0)

print(intraday_df.shape)
intraday_df.head(20)


# In[78]:


intraday_qh = intraday_df[intraday_df['Hour_from'].str.contains('qh')]
intraday_qh = intraday_qh.reset_index(drop=True)
intraday_qh.shape
```

```
# In[79]:


intraday_h = intraday_df[~intraday_df['Hour_from'].str.contains('qh')]
intraday_h = intraday_h.reset_index(drop=True)
intraday_h.shape


# In[9]:


intraday_qh.head(25*4)


# In[10]:


#Index and ID3 Price are NaN for beginning of 2015
#im pretty sure index Price = WAP so just use WAP?
intraday_qh.tail(5)


# # Hourly Intraday

# In[203]:


# get intraday_df and reset index to start from 1/1/15
intraday_h = intraday_df[~intraday_df['Hour_from'].str.contains('qh')]
```

```
#reverse index to start from beginning 2015

intraday_h = intraday_h.reset_index(drop=True)
intraday_h = intraday_h.sort_index(ascending=False, axis=0).reset_index
intraday_h.head()


# In[204]:


# drop unnecessary columns
intraday_h.drop(columns=['Last Price (EUR/MWh)', 'Time_Stamp of Last Pr
                        'Index Price (EUR)'], inplace=True)
intraday_h.tail(3)


# In[205]:


#clean load/gen dataframe
#get numerical col list
cols = df_1516.columns.drop(['Start_Time', 'End_Time'])

#force relevant columns to be numeric and remove cols that only contain
df_1516[cols] = df_1516[cols].apply(pd.to_numeric, errors='coerce')
df_1516.dropna(axis=1, how = 'all', inplace=True)
print(df_1516.columns)
print(df_1516.shape)
```

```
df_1516.head()


# In[206]:


#make load/gen df hourly
df_1516_h = df_1516.groupby(df_1516.index // 4).sum()
df_1516_h['Start_Time'] = list(df_1516.iloc[::4, -2])
df_1516_h['End_Time'] = list(df_1516.iloc[3::4, -1])
df_1516_h.tail()


# In[207]:


#merge dataframes on index
hourly_df = pd.concat([df_1516_h, intraday_h.drop(['Hour from', 'Hour to
hourly_df = hourly_df.iloc[:-2, :]
hourly_df.head()


# In[208]:


hourly_df.drop([#'Day-ahead Total Load Forecast [MW] - Germany (DE)',
                'Low Price (EUR/MWh)', 'High Price (EUR/MWh)'], axis=1,
hourly_df.head()
```

```
## Epex Spot Prices

# Epex Spot Prices are in hour intervals, so maybe make two different d

# In[209]:


epex_2015 = pd.read_excel('PhelixPowerSpotHistory_2015-3.xls', sheet_nam
epex_2016 = pd.read_excel('PhelixPowerSpotHistory_2016.xls', sheet_name
epex_df = pd.concat([epex_2016, epex_2015], axis=0)
epex_df = epex_df.reset_index(drop=True)


# In[210]:


epex_df = epex_df.sort_index(ascending=False, axis=0).reset_index(drop=
epex_df


## Final Hourly DataFrame

# In[211]:


hourly_df = pd.concat([epex_df, hourly_df], axis=1)
hourly_df.shape
```

```
# In[212]:


hourly_df.columns


# In[213]:


#rename some columns
hourly_df.rename(index=str, columns={"Price\nEUR/MWh": "Day-Ahead Price
                                     "Volume Buy (MW)": "Intraday Volume
                                     "Volume Sell (MW)": "Intraday Volume
                                      "Weighted Average Price (EUR)": "In
                                     "Day-ahead Total Load Forecast [MW]
                                     "Actual Total Load [MW] - Germany (I
#create Load Diff variable
hourly_df['Load Diff'] = hourly_df['Day-Ahead Load Forecast [MW]'] - ho
hourly_df.head()


# In[214]:


hourly_df['Delivery Date'].dtype


# In[32]:
```

```
hourly_df.corr()


# In[47]:


col_list= list(hourly_df)
hourly_df['Fossil'] = hourly_df[col_list[6:10]].sum(axis = 1)
col_list


# In[215]:


#create total columns for renewable energies and residual load
ren_list = ['Biomass  - Actual Aggregated [MW]', 'Geothermal  - Actual 
            'Hydro Pumped Storage  - Actual Aggregated [MW]','Hydro Pum
            'Hydro Run-of-river and poundage  - Actual Aggregated [MW]'
            'Hydro Water Reservoir  - Actual Aggregated [MW]',
            'Marine  - Actual Aggregated [MW]', 'Other renewable  - Act
            'Solar  - Actual Aggregated [MW]','Waste  - Actual Aggregate
            'Wind Offshore  - Actual Aggregated [MW]',
            'Wind Onshore  - Actual Aggregated [MW]']
hourly_df['Total Renewable'] = hourly_df[ren_list].sum(axis = 1)

res_load_list = ['Fossil Brown coal/Lignite  - Actual Aggregated [MW]',
            'Fossil Coal-derived gas  - Actual Aggregated [MW]',
```

```
                'Fossil Gas  - Actual Aggregated [MW]','Fossil Hard coal  -
                'Nuclear  - Actual Aggregated [MW]']
hourly_df['Total Residual TR'] = hourly_df[res_load_list].sum(axis = 1)

sh_ren_list = ['Solar  - Actual Aggregated [MW]', 'Wind Offshore  - Act
                'Wind Onshore  - Actual Aggregated [MW]']
hourly_df['Renewable Load'] = hourly_df[sh_ren_list].sum(axis = 1)

#sh_res_load_list = ['Biomass  - Actual Aggregated [MW]', 'Geothermal
- Actual Aggregated [MW]',
#                'Hydro Pumped Storage  - Actual Aggregated [MW]','Hydro Pum
- Actual Consumption [MW]',
#                 'Hydro Run-of-river and poundage  - Actual Aggregated [MW]
#                 'Hydro Water Reservoir  - Actual Aggregated [MW]',
#                 'Marine  - Actual Aggregated [MW]', 'Other renewable
- Actual Aggregated [MW]',
#                 'Solar  - Actual Aggregated [MW]','Fossil Brown coal/Ligni
- Actual Aggregated [MW]',
#                 'Fossil Coal-derived gas  - Actual Aggregated [MW]',
#                 'Fossil Gas  - Actual Aggregated [MW]','Fossil Hard coal
- Actual Aggregated [MW]',
#                 'Nuclear  - Actual Aggregated [MW]']
hourly_df['Residual Load'] = hourly_df['Actual Load [MW]'] - hourly_df[

hourly_df.columns



# In[216]:
```

```
corr_list = ['Day-Ahead Price [EUR/MWh]', 'Intraday WA Price [EUR]', 'R
c = hourly_df[corr_list].reset_index(drop = True).corr()['Day-Ahead Pric
c.iloc[:,2:]


## Hourly Heatmap

# In[217]:


# make Delivery Date into a datetime format
hourly_df['Delivery Date']

#make Delivery Date int datetime format and make it the index
hourly_df['Delivery Date'] = pd.to_datetime(hourly_df['Delivery Date'],
#hourly_df.drop(columns = 'Hour', inplace = True)
hourly_df.set_index('Delivery Date', inplace = True)
hourly_df.head()


# In[218]:


hourly_df.groupby(pd.Grouper(freq = 'm')).mean().head()


# In[219]:
```

```
def show_heatmap(df, ValueColumn, ylabel):
    df['month'] = df.index.month
    df['day'] = df.index.day
    df['hour'] = df.index.hour
    df['minutes'] = df.index.minute
    pivot = df.pivot_table(index=['hour', 'minutes'], columns=['month',
    #pivot = pivot.sort_index(ascending=False, axis=0)
    plt.matshow(pivot, origin= 'lower', interpolation=None, aspect='auto
    plt.tick_params(axis="x", bottom=True, top=False, labelbottom=True,
    plt.ylabel(ylabel)
    plt.xlabel('Month')
    plt.xticks([0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334],
    plt.set_cmap('magma')
    plt.show()


# In[220]:


show_heatmap(hourly_df, 'Solar  - Actual Aggregated [MW]', 'Hour')


# In[54]:


show_heatmap(hourly_df, 'Residual Load', 'Hour')
```

```
# In[55]:

show_heatmap(hourly_df, 'Renewable Load', 'Hour')

# In[56]:

show_heatmap(hourly_df, 'Day-Ahead Price [EUR/MWh]', 'Hour')

# In[57]:

show_heatmap(hourly_df, 'Intraday WA Price [EUR]', 'Hour')

# In[58]:

show_heatmap(hourly_df, 'Actual Load [MW]', 'Hour')

# ### Quarter-hourly data

# In[144]:
```

```
print(df_1516.shape)
print(intraday_qh.shape)
intraday_qh.tail()


# In[141]:


#df_1516['Start_Time'] = pd.to_datetime(df_1516['Start_Time'], format =
#df_1516.head()


# In[64]:


# df_1516 has two entries for the two hours a year when time changes an
#df_1516[df_1516.duplicated(subset = 'End_Time')]


# In[187]:


# merge the two QH dataframes and set Start_Time as the index
print(intraday_qh.shape)
print(df_1516.shape)
qh_df = pd.concat([df_1516, intraday_qh], axis=1)
qh_df['Start_Time'] = pd.to_datetime(qh_df['Start_Time'], format = '%d.%
qh_df.set_index('Start_Time', inplace = True)
qh_df.index = pd.to_datetime(qh_df.index)
```

```
qh_df.head()


# In[188]:


#create total columns for renewable energies and residual load
ren_list = ['Biomass  - Actual Aggregated [MW]', 'Geothermal  - Actual
            'Hydro Pumped Storage  - Actual Aggregated [MW]','Hydro Pum
            'Hydro Run-of-river and poundage  - Actual Aggregated [MW]'
            'Hydro Water Reservoir  - Actual Aggregated [MW]',
            'Marine  - Actual Aggregated [MW]', 'Other renewable  - Act
            'Solar  - Actual Aggregated [MW]','Waste  - Actual Aggregate
            'Wind Offshore  - Actual Aggregated [MW]',
            'Wind Onshore  - Actual Aggregated [MW]']
qh_df['Total Renewable'] = qh_df[ren_list].sum(axis = 1)

res_load_list = ['Fossil Brown coal/Lignite  - Actual Aggregated [MW]',
            'Fossil Coal-derived gas  - Actual Aggregated [MW]',
            'Fossil Gas  - Actual Aggregated [MW]','Fossil Hard coal  -
            'Nuclear  - Actual Aggregated [MW]']
qh_df['Total Residual TR'] = qh_df[res_load_list].sum(axis = 1)

sh_ren_list = ['Solar  - Actual Aggregated [MW]', 'Wind Offshore  - Act
            'Wind Onshore  - Actual Aggregated [MW]']
qh_df['Renewable Load'] = qh_df[sh_ren_list].sum(axis = 1)

#sh_res_load_list = ['Biomass  - Actual Aggregated [MW]', 'Geothermal
```

```
- Actual Aggregated [MW]',
#              'Hydro Pumped Storage  - Actual Aggregated [MW]','Hydro Pum
- Actual Consumption [MW]',
#               'Hydro Run-of-river and poundage  - Actual Aggregated [MW]
#               'Hydro Water Reservoir  - Actual Aggregated [MW]',
#               'Marine  - Actual Aggregated [MW]', 'Other renewable
- Actual Aggregated [MW]',
#               'Solar  - Actual Aggregated [MW]','Fossil Brown coal/Ligni
- Actual Aggregated [MW]',
#               'Fossil Coal-derived gas  - Actual Aggregated [MW]',
#               'Fossil Gas  - Actual Aggregated [MW]','Fossil Hard coal
- Actual Aggregated [MW]',
#               'Nuclear  - Actual Aggregated [MW]']
qh_df['Residual Load'] = qh_df['Actual Total Load [MW] - Germany (DE)']

qh_df['Load Diff'] = qh_df['Day-ahead Total Load Forecast [MW] - German

qh_df.columns


# In[189]:


qh_df.rename(index=str, columns={"Solar  - Actual Aggregated [MW]": "So
                                 "Wind Offshore  - Actual Aggregated [M
                                 "Wind Onshore  - Actual Aggregated [MW
                                "Volume\nMWh": "Day-Ahead Volume [MWh]"
                                "Volume Buy (MW)": "Intraday Volume Buy
                                "Volume Sell (MW)": "Intraday Volume Se
```

```
                                        "Weighted Average Price (EUR)": "Intrada
                                        "Day-ahead Total Load Forecast [MW] - G
                                        "Actual Total Load [MW] - Germany (DE)"
qh_df.columns


# In[182]:


corr_list = ['Intraday WA Price [EUR]', 'Renewable Load', 'Residual Loac
qh_df[corr_list].reset_index(drop = True).corr().iloc[:3,1:]


# In[183]:


def show_qh_heatmap(df, ValueColumn):
    df.index = pd.to_datetime(df.index)
    df['month'] = df.index.month
    df['day'] = df.index.day
    df['hour'] = df.index.hour
    df['minutes'] = df.index.minute
    pivot = df.pivot_table(index=['hour', 'minutes'], columns=['month',
    #pivot = pivot.sort_index(ascending=False, axis=0)
    plt.matshow(pivot, origin= 'lower', interpolation=None, aspect='autc
    plt.tick_params(axis="x", bottom=True, top=False, labelbottom=True,
    plt.xticks([0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334],
    plt.yticks([0, 20, 40, 60, 80], ['0', '5', '10', '15', '20'])
    plt.ylabel('Hour')
```

```
    plt.xlabel('Day')
    plt.set_cmap('magma')
    plt.show()


# In[162]:


show_qh_heatmap(qh_df, 'Solar  - Actual Aggregated [MW]')


# In[163]:


show_qh_heatmap(qh_df, 'Residual Load')


# In[164]:


show_qh_heatmap(qh_df, 'Renewable Load')


# In[166]:


show_qh_heatmap(qh_df, 'Intraday WA Price [EUR]')
```

In[225]:

```
show_qh_heatmap(qh_df, 'Actual Load [MW]')
```

Dataframe Snippets

In[191]:

```
qh_df[['Solar [MW]', 'Wind Onshore [MW]',
       'Wind Offshore [MW]', 'Intraday WA Price [EUR]',
       'Renewable Load', 'Residual Load', 'Load Diff']].head()
```

In[224]:

```
f_hourly_df = hourly_df
f_hourly_df.rename(index=str, columns={"Solar  - Actual Aggregated [MW]'
                                       "Wind Offshore  - Actual Aggregated [MW
                                       "Wind Onshore  - Actual Aggregated [MW]
f_hourly_df[['Solar [MW]', 'Wind Onshore [MW]',
       'Wind Offshore [MW]', 'Intraday WA Price [EUR]', 'Day-Ahead Price
       'Renewable Load', 'Residual Load', 'Load Diff']].head()
```

Other Graph

```
# In[251]:


hour_group = qh_df.groupby(['hour', 'minutes']).mean()
print(hour_group.shape)
print(hour_group.columns)
hour_group.head()


# In[255]:


plt.figure(figsize=(40,20))
hour_group['Intraday WA Price [EUR]'].unstack().plot(kind='bar', width =


# In[ ]:
```