Prediction of Spanish Energy prices

Presented by Group F

MBD Oct 2020

Content

1 Intro

2 The Spanish energy market

3 Data Description

4 EDA

5 Feature Engineering

6 Modelling and Interpretation

- Linear Regression
- Random Forest
- XGBoost

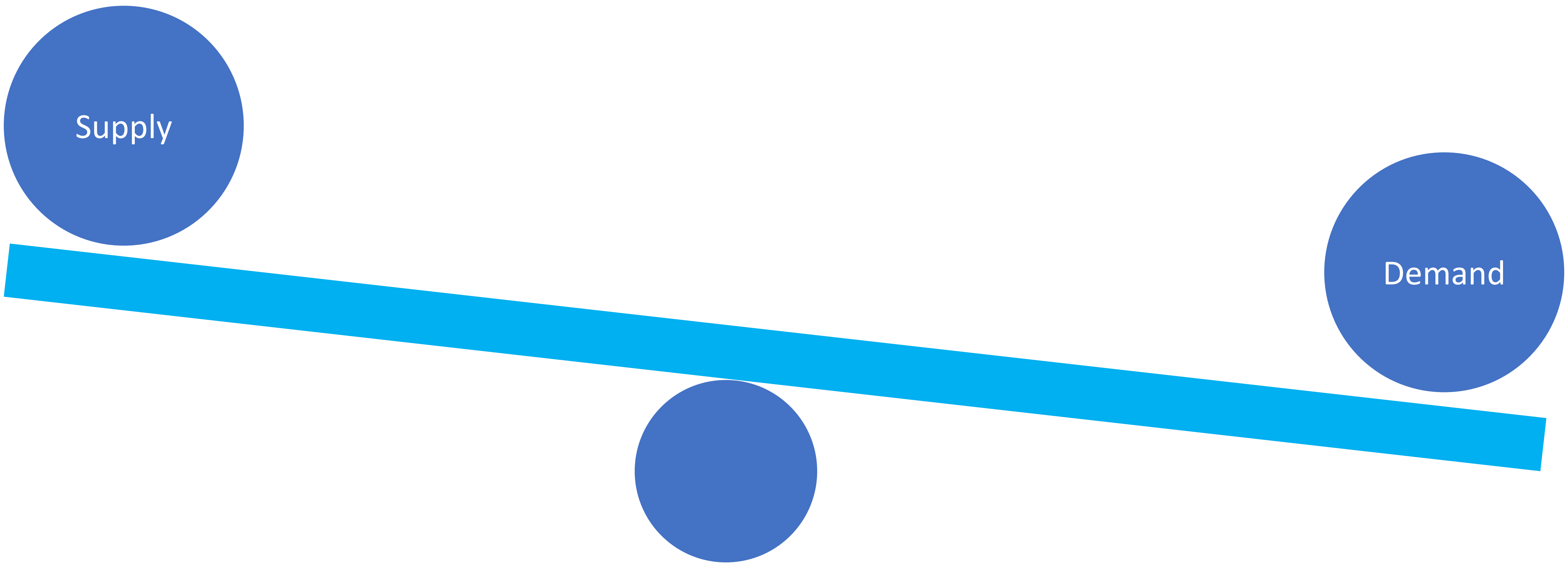
7 Conclusion

1. Intro

The purpose of this project is to predict the hourly price of electricity for the Spanish market by making use of the market's historical movements and its parameters (components). The nature of this topic suggests the use of machine learning tools as adequate for the prediction. This document intends to explain the process and methodology used for the prediction.

2. The Spanish power market

Spain's electricity market is based on daily auctions that determine the price of energy according to the expected energy consumption (demand) and the expected production of energy (supply). It is important to highlight that market bids are prioritized by production price, meaning that those energy sources with lower production cost will have priority over more costly ones.



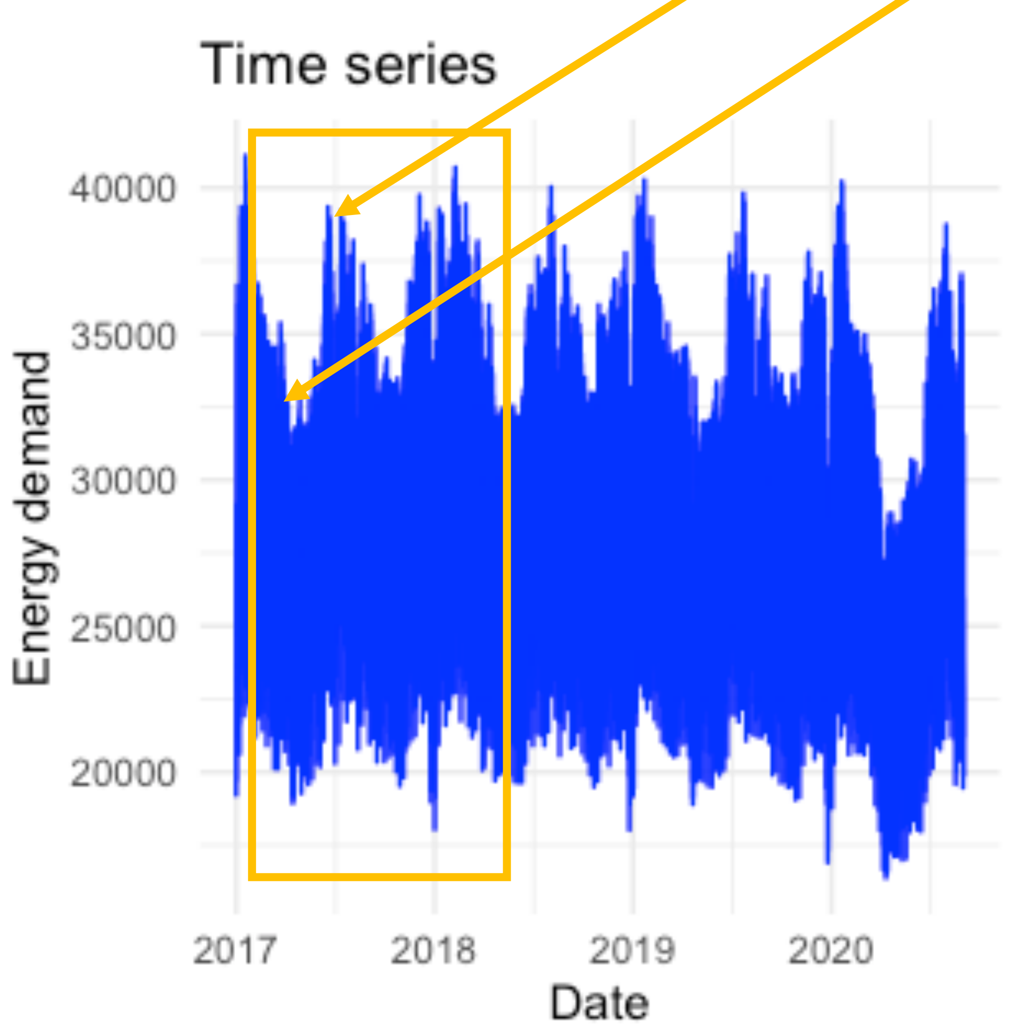
3. Data Description

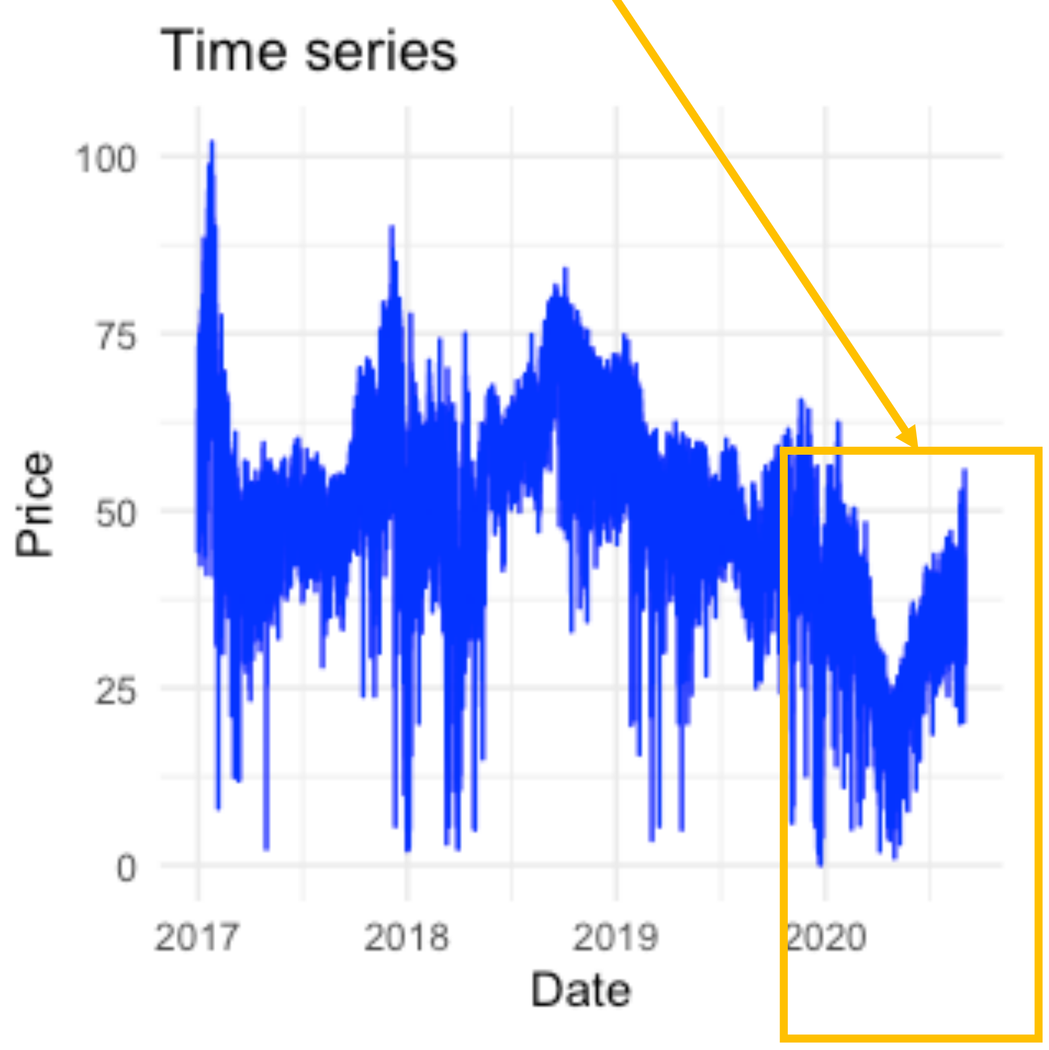
The dataset utilized during the project contains 32.135 hourly observations dating from January 1st, 2017 to August 31, 2020. For each of the observations the following variables are included:

- *date*: date of the observation "%Y-%m-%d"
- *hour*: hour of the observation, [0 - 23]
- *fc_demand*: forecast of demand in MWh
- *fc_nuclear*: forecast of nuclear power production in MWh
- *import_FR*: forecast of the importing capacity from France to Spain in MWh
- *export_FR:* forecast of the exporting capacity from Spain to France in MWh
- *fc wind*: forecast of wind power production in MWh
- *fc_solar_pv:* forecast of PV solar (solar panels) power production in MWh
- *fc_solar_th*: forecast of thermal solar power production in MWh
- *price*: power price for each hour in €/MWh.

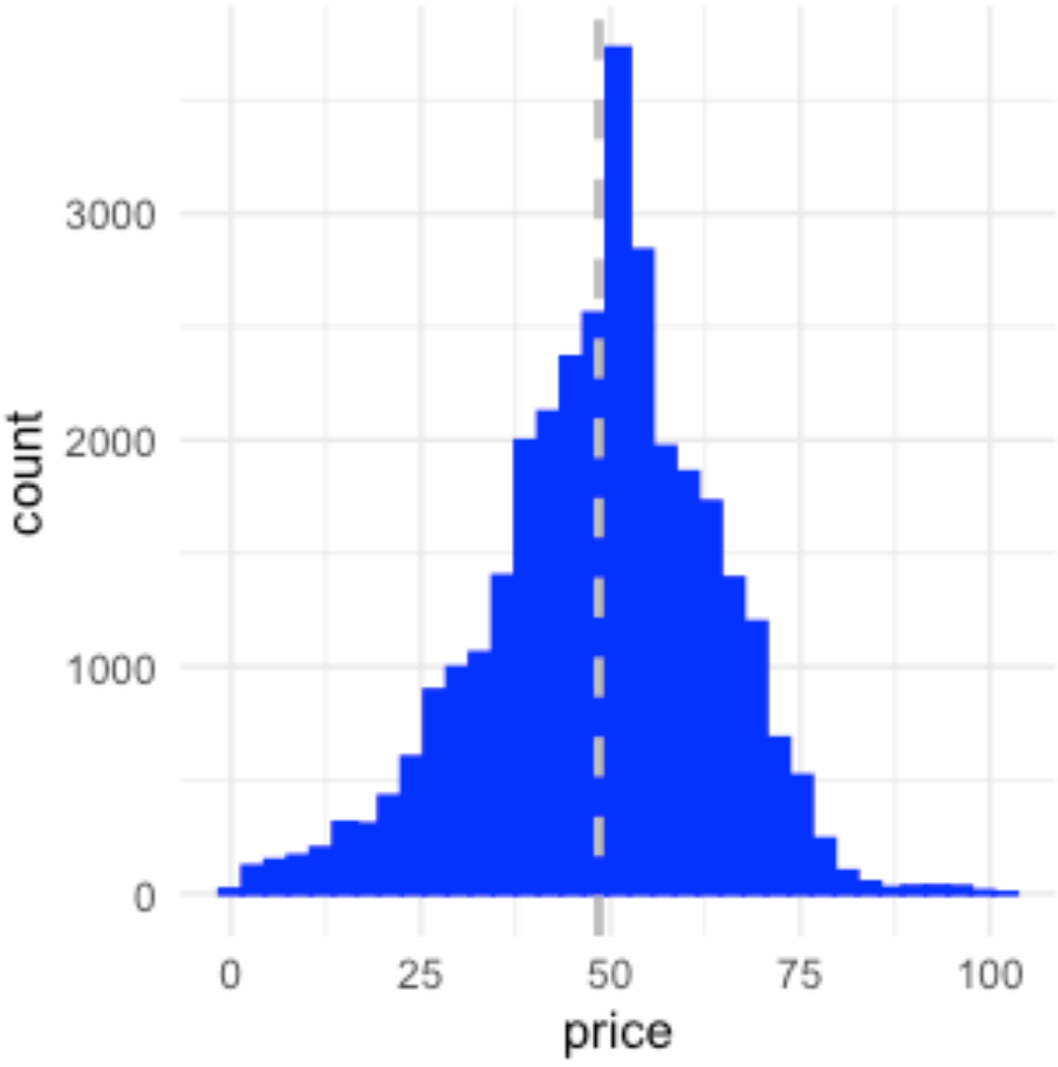
4. Exploratory Data Analysis (EDA)

As the first step of our exploratory data analysis, we started exploring the data by plotting the demand and price over time. As seen below, we can clearly get an idea of the timeframe of the data and other characteristics. It is important to highlight that there are two main observations. The first one was done to obtain a sense of the **seasonal component** of the price, the second was created to observe if there is a clear **impact** of COVID-19 in the dataset.





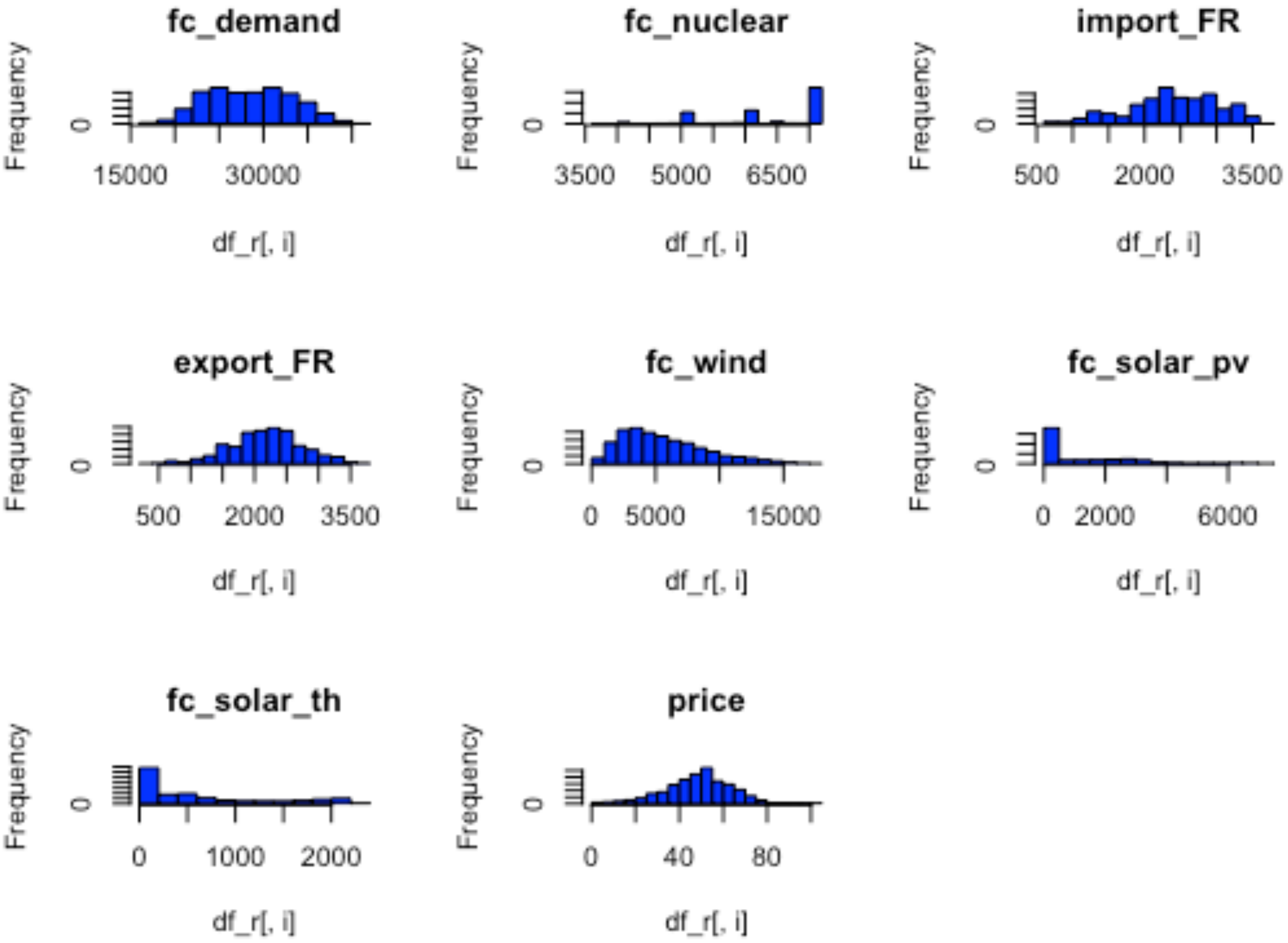
In order to further explore our data, we decided to plot the distribution of our target variable price:



4. Exploratory Data Analysis (EDA)

It becomes obvious that the price variable (the one we want to predict) looks like a normal distribution. Therefore, we can assume that most observations are relatively similar and that we have few outliers in our data set.

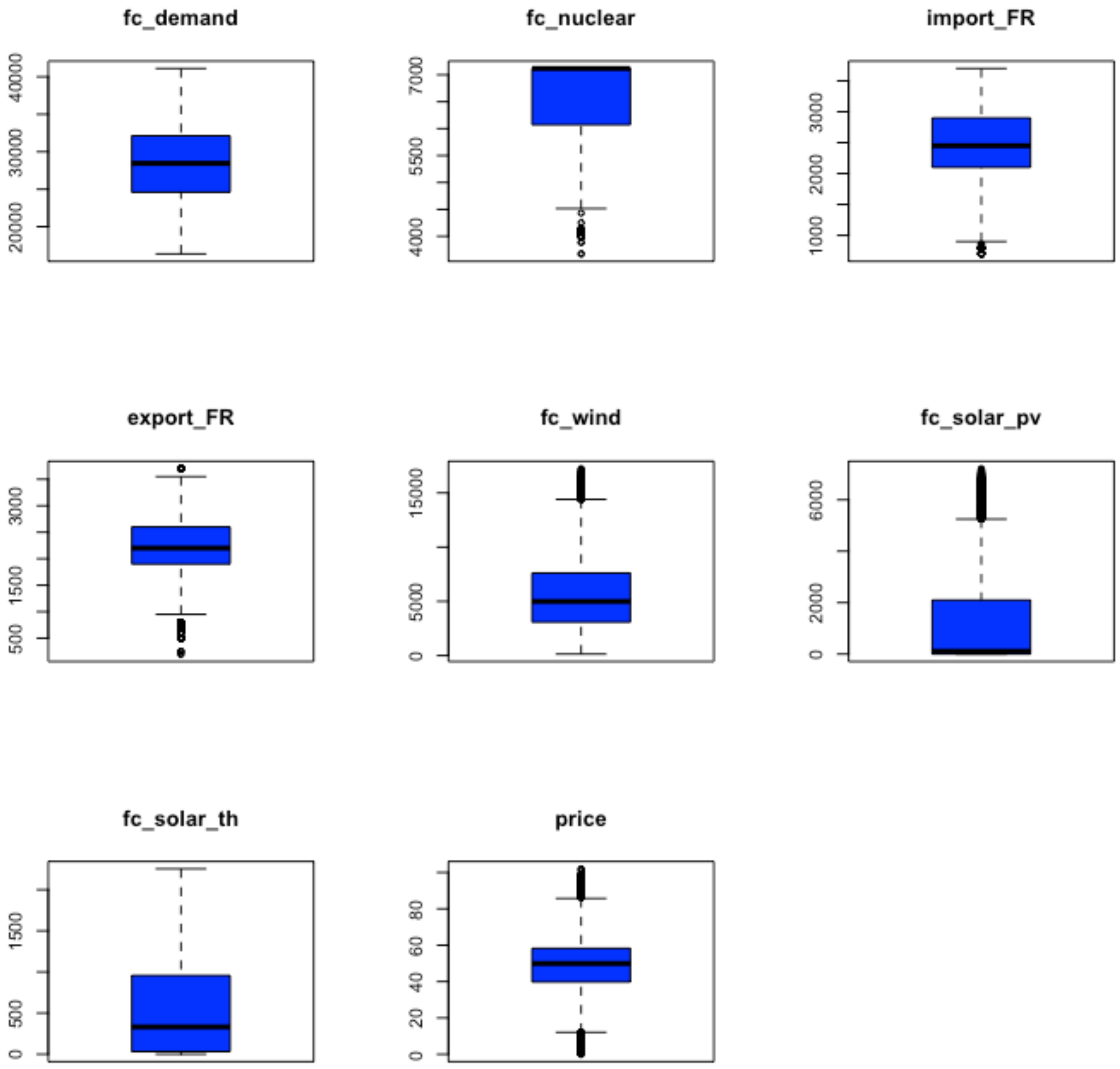
After having explored our most important features, we decided to further loot at our data by plotting the distribution of all variables:



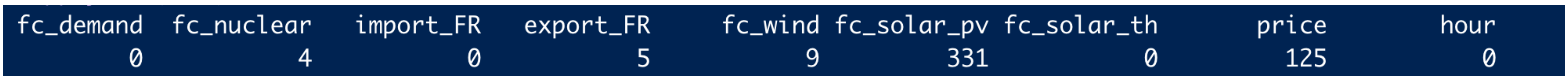
Except for the price variable, the forecast for the export capacity and the demand, we can clearly see that most features are not normally distributed but rather skewed to the right or the left. This observation might be useful later on in the future engineering and modelling part as this might have an effect on the modelling performance.

4. Exploratory Data Analysis (EDA)

The next logical step was to identify potential outliers in our data. We started by building boxplots for each variable as presented below to get an initial idea. Then, we calculated the number of outliers based on different ranges.



As can be seen, we clearly have some outliers for most of the features. Therefore, we decided to calculate the numbers of outlier manually (with 2*IQR) and we obtained the following results:



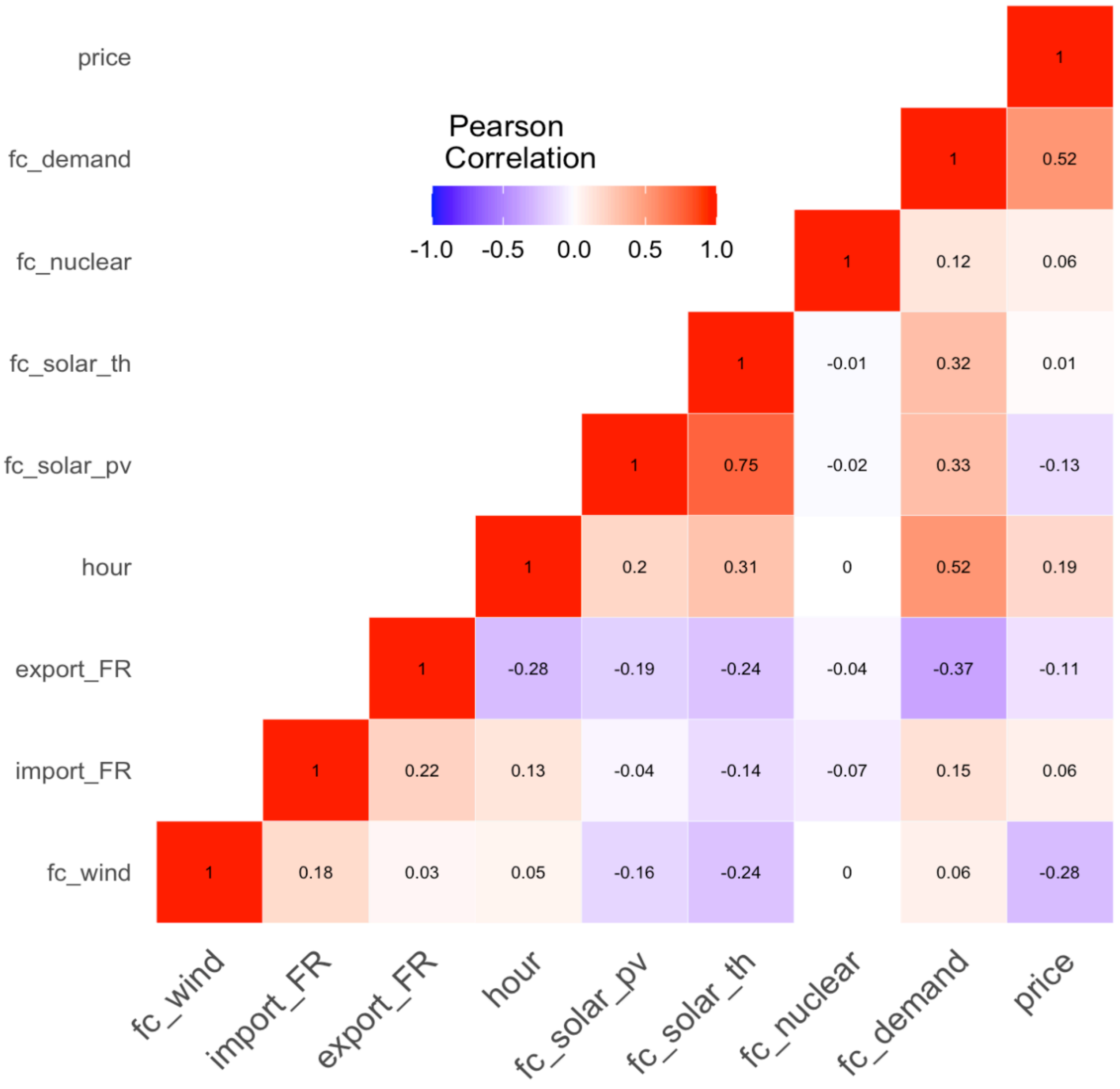
```
fc_demand  fc_nuclear    import_FR    export_FR      fc_wind fc_solar_pv fc_solar_th        price        hour
        0           4            0            5            9         331           0          125           0
```

After increasing the range (3*IQR), all previously observed outliers are not identified anymore. Finally, due to the aspect that we have time series data, we decided to include those observations because otherwise we would risk to disregard times of very high and low prices (high/low energy production) in the energy market.

4. Exploratory Data Analysis (EDA)

Coming now to the aspect of missing values, we found out that the data set had a very little amount of missing observations. We only found 13 values for import_FR and export_FR. Since there were not many missing values, we decided that removing those observations is a valid method. Nevertheless, we also tried applying an imputing algorithm called missForest which imputes the missing values by predicting each of the missing values using random forest. Later on, during the modelling process we realized that the technique of handling the missing values did not influence the performance of our models, so both methods used here are valid ones.

Moreover, we chose to explore our dataset by analyzing the correlation between variables, as we may identify high correlated ones, which might bias our model. The graph presented below displays the correlation matrix based on the Pearson correlation coefficient.



4. Exploratory Data Analysis (EDA)

At a first glance we can observe that our coefficients ranges from around -0.4 to 0.75. We can find evidence that energy coming from both solar sources, photovoltaic and thermal, are highly correlated having a coefficient of 0.75. Likewise, the coefficients computed for energy demand and price achieve a value of 0.52, which seems to be logical as both renewable energy sources depend on sunlight. The variables ‘hour’ and the corresponding one for the energy demand also showed a coefficient of 0.52. This correlation analysis gave us a rough overview of which features might have a higher impact on our dependent variable price as well as helps us to understand the relationship in between the different features.

5. Feature Engineering

First of all, we started building the models including all features as we wanted to prevent missing important information. In addition to that, we tried to include the information of seasonality by adding a variable for the month as well as for the week in the year and the type of the day indicating if it is a week day or a weekend day.

Due to the different behavior of the energy market during the covid-19 crisis (which we already saw in the EDA), we tried to find a variable which could explain this behavior. Therefore, we thought the total deaths per week in Spain (total and not covid deaths) could be a good indicator for the crisis and its intensity. We calculate the correlation between number of deaths and prices getting a value of -0.5 meaning more deaths is lowering demand therefore decreasing energy prices.

To add another feature explaining the interaction between the variables, we thought it was useful to include a thermal gap variable which is the difference between the high cost and the low cost energies. We found that this a best practice in predicting energy prices which was then also proven by our analysis as it turned out to be one of the most important explanatory variables in predicting prices.

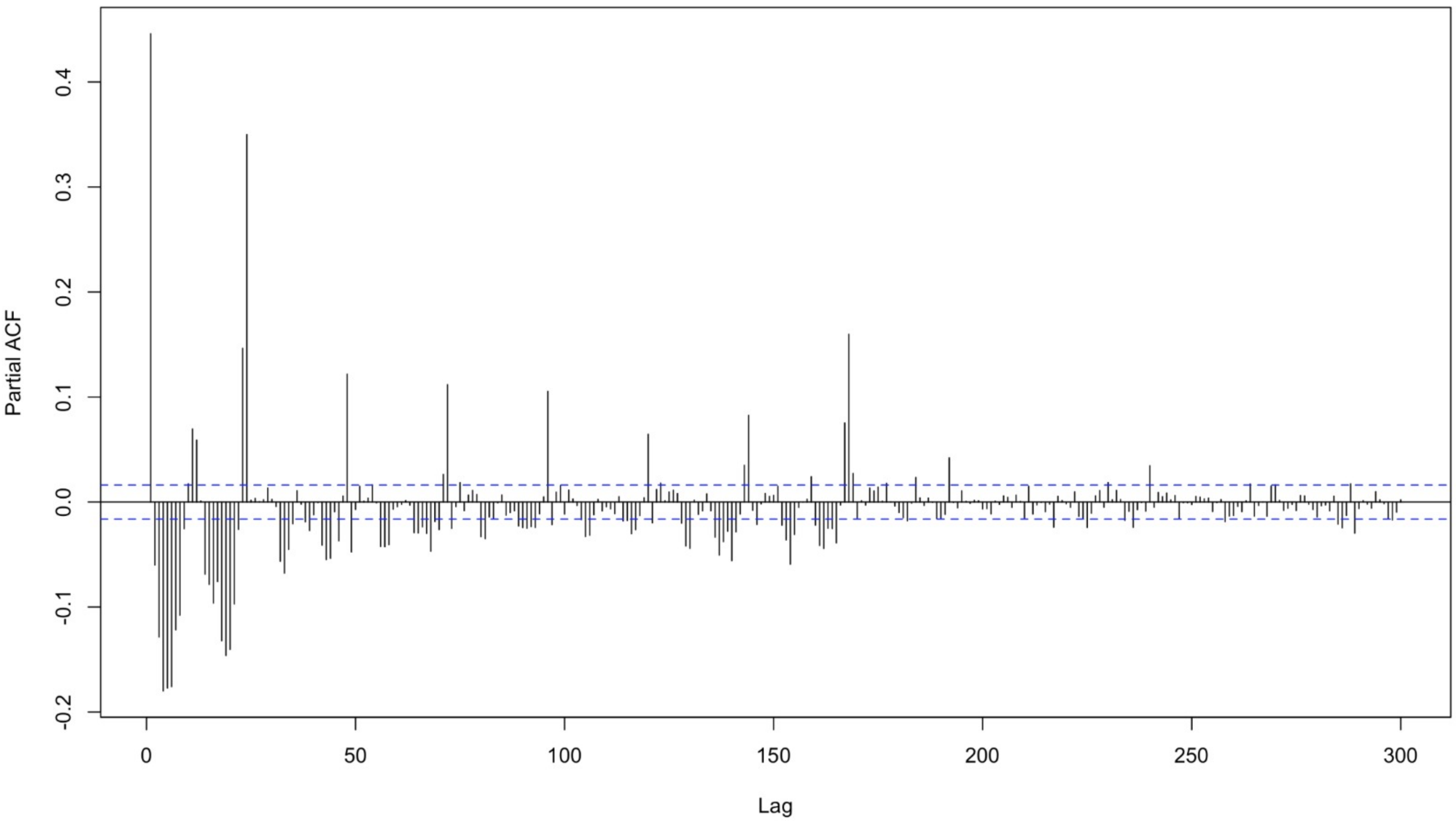
Fourth, we were thinking the price lags would have been a valid feature in our goal predicting prices but as we don’t have real time data we cannot know those prices lags. So in case our prediction would only be for the next day, we could have made use of that feature but since the problem here is to predict over multiple weeks we were not able to make use of this feature.

Since will still tried to include this time series behavior we thought it was useful to include the lags of the feature with highest explanatory power which in our case was the thermal gap.

5. Feature Engineering

Below a representation of the most important lags by plotting the Partial ACF (Auto correlation function):

Price lags (for stationary data)

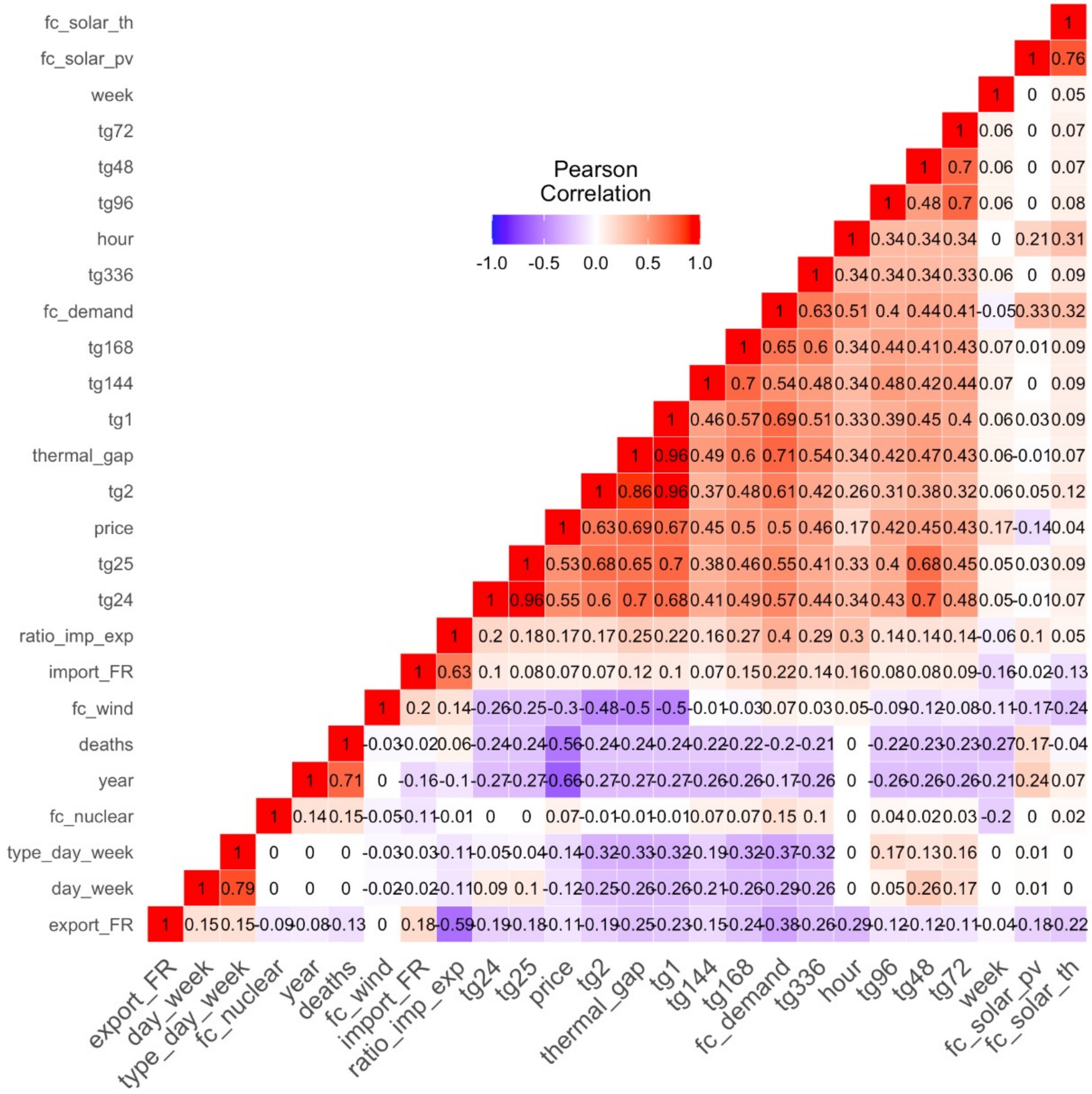


So what we did, is analyzing the price series in order to know which lags had the highest correlation with the actual price. The most significant ones were the first, the 24th, 48th and the 168th.

As we couldn’t use those lags (the price lags), we substituted them with the lags from the thermal gap variable to still account for the time series behavior (special days/seasonality) of the data.

5. Feature Engineering

After adding all features, we plotted again the correlation matrix we saw during the EDA and obtained the following results:



We can see that the price is highly positively correlated with the thermal gap and most of its lags, as well as highly negatively correlated with the deaths meaning that we included powerful and meaningful features.

6. Modelling and Interpretation

Feature selection:

Given the findings in the feature engineering section as well as during the exploratory data analysis, we chose to include not all but rather a selected groups of features which showed high correlations to our dependent variable price. Furthermore, we tried also different combinations throughout the modelling process and we arrived to choose the following variables: tg1, tg2, tg24, tg25, tg48, tg168, deaths, thermal_gap, fc_demand and year.

Evaluation of the models:

Initially, we tried running a sliding window validation by means of the package caret in R but sadly encountered the issue of not having enough computational resources to run it for our model. Therefore, we built our own sliding window which basically does the same but in a more simpler format.

Baseline model 1: Linear regression

For the beginning of our analysis we decided to use a normal linear regression. This is one of the simplest methods in machine learning, so it may not be the best method for complex data, but it still offers good interpretability.

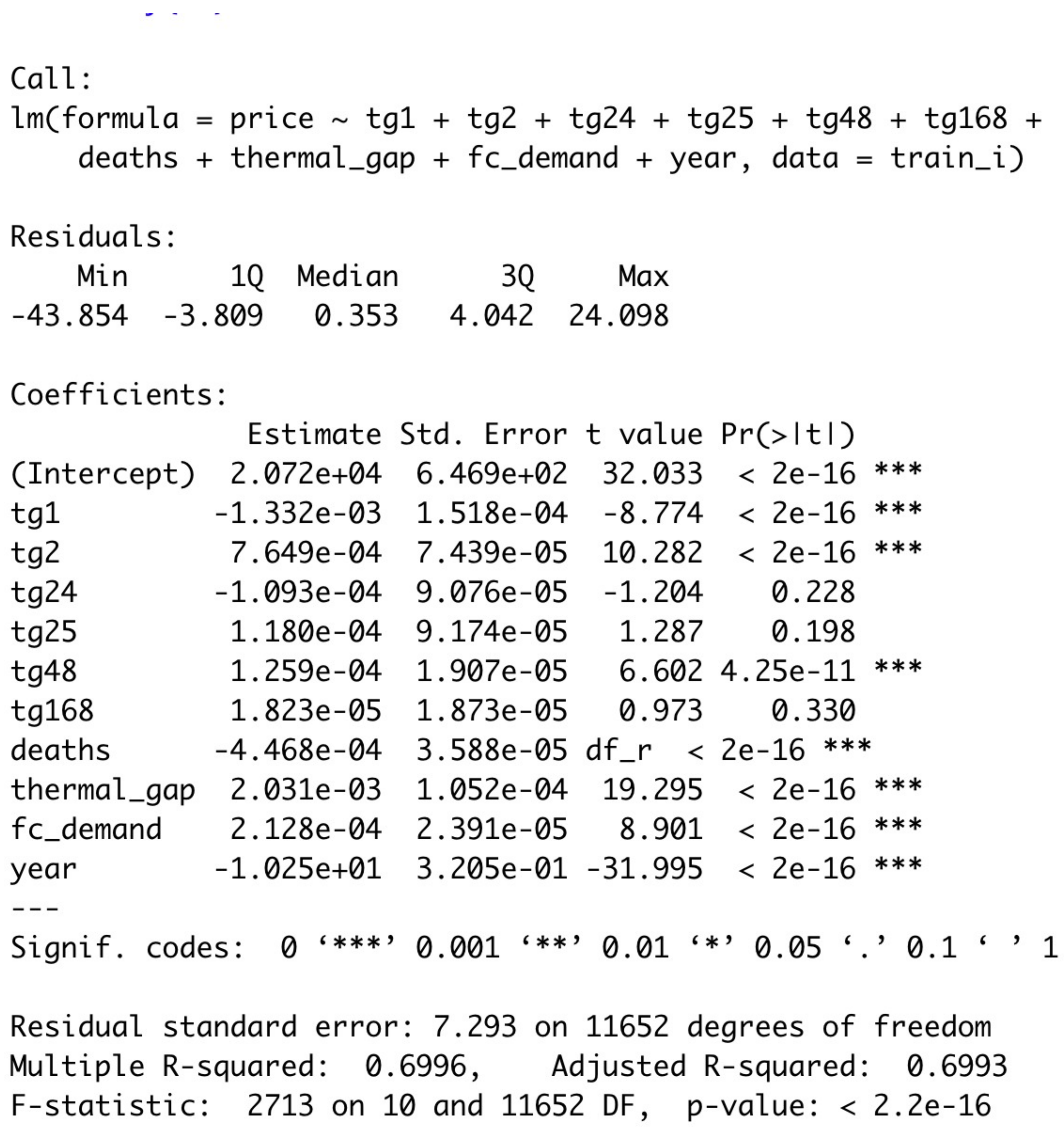
The basic structure of a linear regression is described as follows:

$Y = \beta 1 + \beta 2X + \epsilon$

In linear regression, we try to predict the values of one variable in our case the prices with the help of one or more other variables. As the name already tells us, linear regression only looks at linear relationships.

Applying multiple linear regression to our problem and trying to predict the price we achieved the following results: (please see next page)

6. Modelling and Interpretation



```
Call:
lm(formula = price ~ tg1 + tg2 + tg24 + tg25 + tg48 + tg168 +
    deaths + thermal_gap + fc_demand + year, data = train_i)

Residuals:
    Min      1Q  Median      3Q     Max
-43.854  -3.809   0.353   4.042  24.098

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.072e+04  6.469e+02  32.033  < 2e-16 ***
tg1         -1.332e-03  1.518e-04  -8.774  < 2e-16 ***
tg2          7.649e-04  7.439e-05  10.282  < 2e-16 ***
tg24        -1.093e-04  9.076e-05  -1.204    0.228
tg25         1.180e-04  9.174e-05   1.287    0.198
tg48         1.259e-04  1.907e-05   6.602 4.25e-11 ***
tg168        1.823e-05  1.873e-05   0.973    0.330
deaths      -4.468e-04  3.588e-05 df_r  < 2e-16 ***
thermal_gap  2.031e-03  1.052e-04  19.295  < 2e-16 ***
fc_demand    2.128e-04  2.391e-05   8.901  < 2e-16 ***
year        -1.025e+01  3.205e-01 -31.995  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.293 on 11652 degrees of freedom
Multiple R-squared:  0.6996,    Adjusted R-squared:  0.6993
F-statistic:  2713 on 10 and 11652 DF,  p-value: < 2.2e-16
```

6. Modelling and Interpretation

Baseline model 1: Linear regression

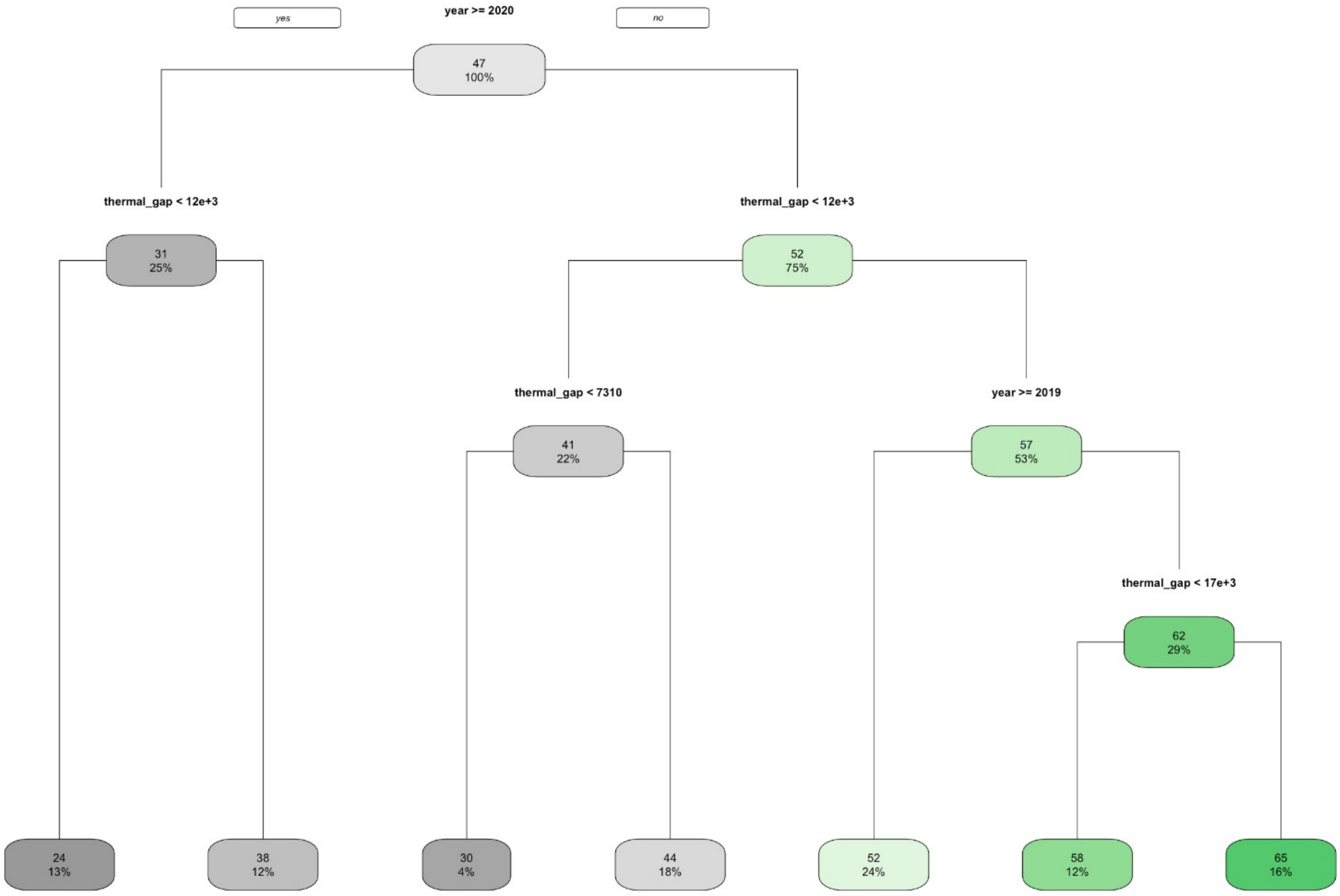
The regression output shows that most variables are significant within a significance level of 1%. Validating our regression model we obtain a Root mean squared error (RMSE) using the sliding window of 9.2 which is fairly good but defintely can be improved by using more avdanced techniques.

As in our multiple linear regression we just took into consideration all pre-selected features and its very likely that our model overfit or just took into consideration features which are not mandatory needed, we ran another method called best subset regression. This method is automatically adding or removing features and by means of that tries to predict the best subset of features. Since the forward stepwise method gave us a worse prediction than the one we ran with out pre-selected group of features we did not look further into this method.

6. Modelling and Interpretation

Baseline model 2: Decision tree

As another baseline model and keeping in mind that we will use further decision tree methods we built a basic decision tree and the outcome can be seen below:
Similar to what we have expected it definetely makes a difference in which year we are in, as 2020 includes the effect of covid. Further, it becomes obvious that the thermal gap plays a siginifcant role. Applying simple decision trees we achieved the following results (with a test RMSE using sliding window of 6.9):



6. Modelling and Interpretation

XGBoost

After running normal decision trees we decided to try out one more advanced method called XGboost. This library enables supervised machine learning with the Boosted Tree algorithm, a tree algorithm with gradient boosting. Boosting is a method that combines different simpler and weaker models to make better predictions of the target variable. Models are added until no more improvements in the predictions occur. Gradient boosting uses a special gradient algorithm called the gradient descent algorithm to add the models. The tree algorithm with gradient boosting adds new branches that predict errors of previous branches and minimize the errors or losses. By linking the trees and the branches, respectively, the final predictions are created.
Our model in Xgboost, including also hyperparameter tuning within a grid search, we finally obtained a rolling window RMSE of 6.4

Random Forest

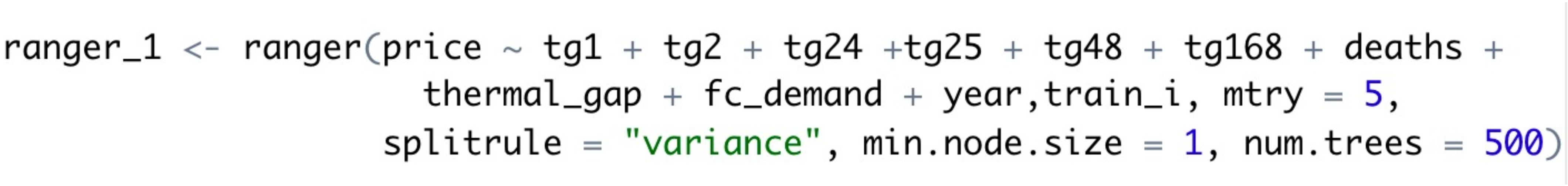
After exploring more simpler techniques we looked into a more advanced technique called Random Forest. Implied by its name, the algorithm random forest ("forest" instead of "tree") essentially implies multiple trees instead of considering only one. While the regression tree is built on the training set including all observations, random forest considers randomly at each split, a random sample of m features as split candidates from the full collection of p features. Then these split can choose only one out these m features. After repeating this procedure multiple times, each tree determines a certain class and the class which is determined most, will be finally chosen for the split.

Using a grid search for hyperparameter tuning and validating the model using the sliding window, we were able to compute a relatively stable and accurate model. Our final RMSE for this method was 4.9.

6. Modelling and Interpretation

Random Forest

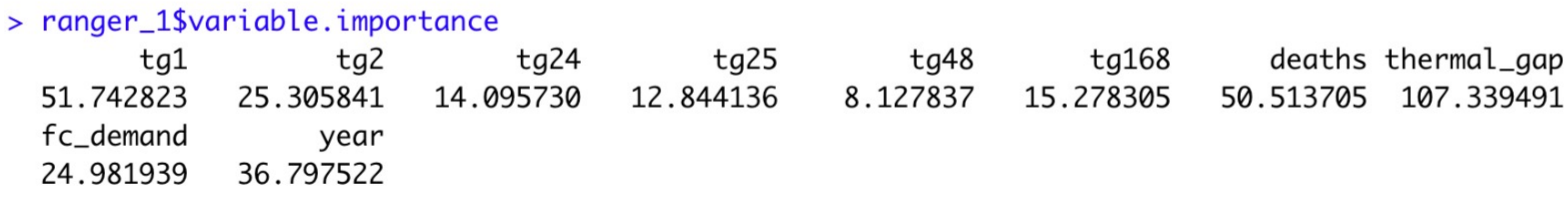
As random forest is less likely to overfit and we obtained the lowest RMSE we decided to make this model as our final one:



```
ranger_1 <- ranger(price ~ tg1 + tg2 + tg24 +tg25 + tg48 + tg168 + deaths +
                    thermal_gap + fc_demand + year,train_i, mtry = 5,
                   splitrule = "variance", min.node.size = 1, num.trees = 500)
```

We chose to only run it with 500 trees as otherwise our model was giving us worse results during the validation procedure as it was overfitting. Furthermore, we chose the min node size (minimum node size) of one, the split rule and the mtry (Number of variables to possibly split at in each node) of 5 accordingly to the grid search we have done before.

Just for completeness, we also calculated the variable importance of each of our features and arrived to the conclusion that thermal_gap was definitely a good choice to add and that variables having a high correlation with the dependent variable also proved to be important while running our model:



```
> ranger_1$variable.importance
        tg1         tg2        tg24        tg25        tg48       tg168      deaths thermal_gap 
  51.742823   25.305841   14.095730   12.844136    8.127837   15.278305   50.513705  107.339491 
  fc_demand        year 
  24.981939   36.797522 
```

7. Conclusion

In summary, with the help of extensive data analysis, we have managed to build a model that can predict Spanish energy prices fairly accurately. Especially in a market like Spain, this can be very important, as prices fluctuate a lot.

Below are the **steps of our analysis and its outcomes:**

The exploratory data analysis has shown that we clearly have a seasonality as well as a strong effect of covid. In addition, the dataset was already of high quality and there was only little work to be done to clean the data.

Given the findings in the exploratory data analysis and the domain knowledge, we designed feature engineering accordingly. We added the week as well as the month to account for seasonality. In addition, we added total deaths in Spain because it best captured the effect of covid. Furthermore, given the dependency of energy production and demand and the time series data we have here, we added several lag variables.

For the modelling process, we then used three different approaches. First, we started with a multiple linear regression and decision trees. By means of those methods, we obtained fairly good results in terms of model performance and we were able to get a feeling on the importance of variables and it helped us to interpret the model better. After that, we tried different decision tree techniques, starting with Xgboost and moving then to random forest. This method gave us the best results and was the least overfitted so we decided to further improve it. By means of a grid search as well as validating the model with a sliding window we were finally able to obtain a RMSE of around 4. This analysis was of course not 100% optimized due to the time restriction. It could be taken further by adding more feature engineering, also with the help of external data on the energy market. In addition, with more computation power, a more extensive grid search could have been performed to optimize the models.