Assignment 2

Ebru Gecici

16/08/2020

This report includes implementation of the **dplyr()** function and **ggplot()** function by using EPIAS 01-31 July 2020 data. To make analysis, first the data is rearranged to analyze and then required calculations are made to get meaningful outputs. By using this outputs, interpretation can be made for July 2020.

More detailed from different time frame can be found in the [EPIAS](https://rapor.epias.com.tr/rapor/xhtml/ptfSmfListeleme.xhtml).

# Implementation of the dplyr() and ggplot()

## EPIAS Electricty Report

Electricity market provide transparency, competition and integration with other electricity markets. Main purpose of this system is to provide stronger achievement in the dynamic system [(1)](https://www.epias.com.tr/en/day-ahead-market/introduction/). For this reason, to provide more information and obtain better prediction about the future energy market, there are available data sets. By analyzing this data, balanced energy market can be achieved. The balanced electricity market means that the produced and consumed energy are in the same level. That is, you produce electricity that you needs. As the price of the electricity in huge amounts, especially in this technology era, this is more important.

To start basic implementation about the electricity price, this report is prepared. In this report, *July 2020* data is used.This data consists of six different values:

1. date,
2. market clearing price,
3. system marginal price,
4. positive imbalance price,
5. negative imbalance price,
6. system marginal price direction.

The **date** is the date information, i.e., day, month, year and the hour, minute, second information. The **market clearing price** is obtain when the produced/supplied is equal to consumed price. This price is also called *equilibrium price*. **System marginal price**, on the other hand, is used in the imbalance/ in-equilibrium point. If there is energy deficit in system, the lowest price for electricity is offered, whereas, if there is energy surplus in the system the highest price is offered. By using these classification, deficit and surplus, the system marginal price direction is defined [(2)](https://www.epias.com.tr/en/balancing-market/system-marginal-price-smp/#:~:text=System%20marginal%20price%20is%20calculated,energy%20surplus%20in%20the%20system.). When the energy demand is more than the predicted demand, **deficit** is occurred in the system. In contrast to deficit, when the energy demand is lower than the predicted demand, **surplus** is observed. If market clearing price is greater than the system marginal price, the energy surplus is obtained, otherwise we get energy deficit in the system. Moreover, by using market clearing price and system marginal price, we obtain positive **imbalance price** and **negative imbalance price**:

* Positive Imbalance Price = min(MCP,SMP)\*0.97
* Negative Imbalance Price = max(MCP,SMP)\*1.03

These calculations show that market operator penalize the system [(3)](https://boun-ie48a.github.io/files/electricity_markets_mini_tutorial) and this process is called imbalance price.

### 1. DATA MANUPILATION

#### 1.1. Data

To make analysis, we first load the packages and manipulate the data to obtain clearer results. For this, after the loading of the data, we check the data type for the calculations.

#we call required packages from the packages which are "tidyverse" and "lubridate"  
library(tidyverse)  
library(lubridate)  
library(ggplot2)  
#we upload data, which is csv file format, by using read.csv() function  
EpiasData <- read.csv("ptf-smf.csv")  
#after the control of the data we can examine columns in the data by using glimpse() function  
EpiasData %>%  
 glimpse()

## Rows: 744  
## Columns: 6  
## $ Date <chr> "01.07.20 00:00", "01.07.20 01:00...  
## $ MCP. <dbl> 323.85, 326.95, 324.31, 322.11, 3...  
## $ SMP. <dbl> 211.00, 201.00, 211.00, 211.00, 2...  
## $ Positive.Imbalance.Price..TL.MWh. <dbl> 204.67, 194.97, 204.67, 204.67, 1...  
## $ Negative.Imbalance.Price..TL.MWh. <dbl> 333.57, 336.76, 334.04, 331.77, 3...  
## $ SMP.Direction <chr> "?Energy Surplus", "?Energy Surpl...

#then we control our date type because in the functions we need date type  
is.Date(EpiasData$Date)

## [1] FALSE

#as the date column is not date type, first we convert "." to "-" by using gsub() function  
EpiasData$Date <- gsub(pattern = "\\.","-",EpiasData$Date)  
#after the changing we transform our date time from character to dttm by using a function of lubridate which is as.POSIXct  
EpiasData$Date <- as.POSIXct(EpiasData$Date,format="%d-%m-%y %H:%M")  
#control of the process  
EpiasData %>%  
 glimpse()

## Rows: 744  
## Columns: 6  
## $ Date <dttm> 2020-07-01 00:00:00, 2020-07-01 ...  
## $ MCP. <dbl> 323.85, 326.95, 324.31, 322.11, 3...  
## $ SMP. <dbl> 211.00, 201.00, 211.00, 211.00, 2...  
## $ Positive.Imbalance.Price..TL.MWh. <dbl> 204.67, 194.97, 204.67, 204.67, 1...  
## $ Negative.Imbalance.Price..TL.MWh. <dbl> 333.57, 336.76, 334.04, 331.77, 3...  
## $ SMP.Direction <chr> "?Energy Surplus", "?Energy Surpl...

#### 1.2. Rename columns and rearrange the data frame

To provide more understandable data, I change the name of the column. This process is provide the uses to get clear information about data.

#then, I changed the name of the column by using transmute() function with pipe operator. And then I assigned this process into  
# raw\_df name as a data frame. After this process, we have usable data frame  
raw\_df <- EpiasData %>%  
 transmute(Date,  
 #rename of the MCP. column  
 MCP = MCP.,   
 SMP = SMP.,   
 PositiveIP = Positive.Imbalance.Price..TL.MWh.,  
 NegativeIP = Negative.Imbalance.Price..TL.MWh.,  
 SMPDirection = SMP.Direction)   
  
#showing first sith raw of the data frame  
head(raw\_df)

## Date MCP SMP PositiveIP NegativeIP SMPDirection  
## 1 2020-07-01 00:00:00 323.85 211 204.67 333.57 ?Energy Surplus  
## 2 2020-07-01 01:00:00 326.95 201 194.97 336.76 ?Energy Surplus  
## 3 2020-07-01 02:00:00 324.31 211 204.67 334.04 ?Energy Surplus  
## 4 2020-07-01 03:00:00 322.11 211 204.67 331.77 ?Energy Surplus  
## 5 2020-07-01 04:00:00 320.00 201 194.97 329.60 ?Energy Surplus  
## 6 2020-07-01 05:00:00 286.21 181 175.57 294.80 ?Energy Surplus

The output shows that, Our data frame is ready for the calculations. That is, the class of the date is changed from character to date-time (dttm). By using this type, we can use *lubridate()* function to calculate, group, etc. After the data manipulation, we start to analyze. The rest of the report is arranged as follow. In the first part of the analysis section, MCP is given. Then the SMP and its relation with the MCP is taken into account. After that, the imbalance price analyzed. Finally, the SMP direction is investigated.

### 2. ANALYSIS

In the analysis of clear data,

* First, I give the visualized outputs to make interpretation,
* Then, I give the calculated outputs

In both, dplyr() function is to prepare and manipulate the data frame for visualization and mathematical analyses. Basic Information About Data (note that the more information about this values are explaned in the previous section):

* **MCP**: Market Clearing Price
* **SMP**: System Marginal Price
* **PositiveIP**: Positive Imbalance Price
* **NegativeIP**: Negative Imbalace Price
* **SMPDirection**: Sytem Marginal Price Direction

#### 2.1. MCP

#### 2.1.1. Hourly MCP

By using minimum and maximum value of the hourly MCP by using following function.

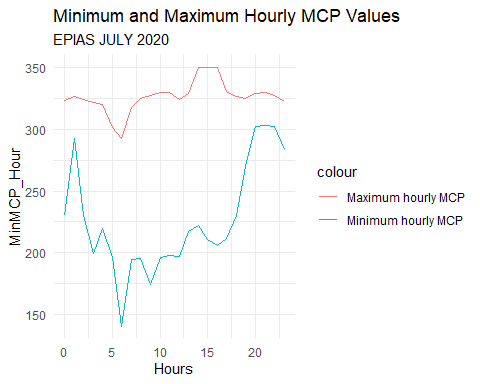
raw\_df %>%  
 select(Date, MCP)%>%  
 #first we group our data according to the hours  
 group\_by(Hour = lubridate::hour(Date))%>%  
 #then we calculate average value of MCP hour each our  
 summarise(MinMCP\_Hour = min(MCP),  
 MaxMCP\_Hour = max(MCP))%>%  
 #ungproup  
 ungroup()%>%  
 #to provide sorting of the data hourly minimum MCP and hourly maximum MCP from smallest to largest   
 arrange(MinMCP\_Hour,MaxMCP\_Hour)%>%  
 print(n=3)

## # A tibble: 24 x 3  
## Hour MinMCP\_Hour MaxMCP\_Hour  
## <int> <dbl> <dbl>  
## 1 6 140. 293   
## 2 9 174. 328.  
## 3 7 194. 318.  
## # ... with 21 more rows

Results show that, in the sixth hour, the minimum MCP value can be observed in the minimum and the maximum MCP values. The sixth hour refers to 5 am.

#### 2.1.2. Graph of the öinimum and maximum values of the MCP according to the hours

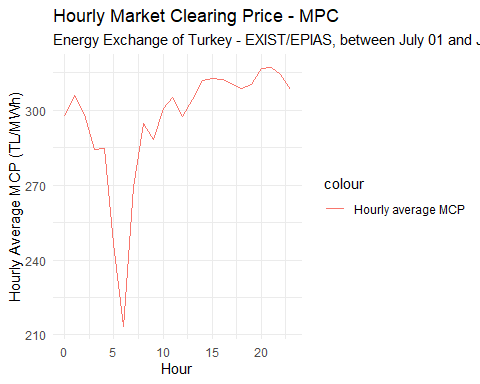
raw\_df%>%  
 #group according to the hours  
 group\_by(Hours=lubridate::hour(Date))%>%  
 #finding minimum and maximum value of the MCP   
 summarise(MinMCP\_Hour = min(MCP),  
 MaxMCP\_Hour = max(MCP))%>%  
 #to visualize these min and max values by using ggplot() function according to the hours  
 ggplot(aes(x=Hours)) +  
 #define lines in the graph  
 geom\_line(aes(y = MinMCP\_Hour, color = "Minimum hourly MCP"))+  
 geom\_line(aes(y = MaxMCP\_Hour, color = "Maximum hourly MCP"))+  
 #give a name to the graph  
 labs( title = "Minimum and Maximum Hourly MCP Values",  
 subtitle = "EPIAS JULY 2020")+  
 #type of the graph, these function provide white canvas   
 theme\_minimal()



After the this calculations for each hour, we can also obtain hourly average MCP values and we can illustrates in the graph like as follow. While the minimum value of the MCP is observed in the sixth hour, the average is minimum in this hour. This is fact. Moreover, if the minimum value is too close to maximum MCP value, we can obtain higher average value of MCP in the hourly results.

#### 2.1.3. Graph of the hourly average MCP value

#After all this process by using raw\_df data frame, we can make analysis about the EPIAS data which is related to electricity  
#Daily average MCP  
plot1<-raw\_df %>%  
 #first we group our data according to the hours  
 group\_by(Hours = lubridate::hour(Date))%>%  
 #then we calculate average value of MCP hour each our  
 summarise(#daily\_total\_MCP = sum(MCP),  
 hourly\_average\_MCP = mean(MCP))  
#to visualize the result we use ggplot() function, which gives the canvas of the graph  
ggplot(plot1, aes(x = Hours)) +  
 #geom\_line() gives the line graph  
 geom\_line(aes(y= hourly\_average\_MCP, color="Hourly average MCP")) +   
 labs(x = "Hour", y = "Hourly Average MCP (TL/MWh)",  
 title = "Hourly Market Clearing Price - MPC",  
 subtitle = "Energy Exchange of Turkey - EXIST/EPIAS, between July 01 and July 31 ")+  
 theme\_minimal()



In addition to the hourly analysis of the MCP, we can also make these calculations for each day. For this purpose, first we define minimum and maximum value of each day, and then we calculate average value of the daily MCP value. Finally, by using visualization tools, the outputs can be analyzed.

#### 2.1.4. Weekly MCP

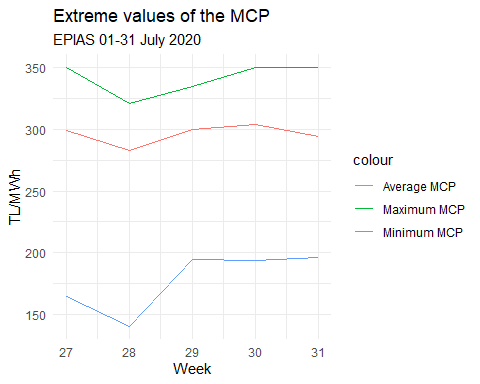
* WA\_MCP : Weekly Average MCP value

EG <- raw\_df %>%  
 group\_by(Week=lubridate::week(Date))%>%  
 summarise(WA\_MCP = mean(MCP),  
 MinMCP = min(MCP),   
 MaxMCP = max(MCP))  
  
knitr::kable(EG, format="markdown")

|  |  |  |  |
| --- | --- | --- | --- |
| Week | WA\_MCP | MinMCP | MaxMCP |
| 27 | 299.3835 | 164.64 | 349.99 |
| 28 | 282.5803 | 140.01 | 321.35 |
| 29 | 300.1358 | 194.65 | 334.99 |
| 30 | 304.2971 | 193.26 | 350.00 |
| 31 | 294.1286 | 196.09 | 349.99 |

#### 2.1.5. Graph of weekly MCP

#Minimum and maximum value of the MCP according to the day  
raw\_df %>%  
 group\_by(Week=lubridate::week(Date))%>%  
 summarise(MinMCP = min(MCP), MaxMCP = max(MCP), Average = mean(MCP))%>%  
 ggplot(aes(x=Week)) +  
 geom\_line(aes(y = MinMCP, color = "Minimum MCP")) +  
 geom\_line(aes(y = MaxMCP, color = "Maximum MCP")) +  
 geom\_line(aes(y = Average, color = "Average MCP")) +  
 labs(x = "Week",  
 y = "TL/MWh",  
 title = "Extreme values of the MCP",  
 subtitle = "EPIAS 01-31 July 2020") +  
 theme\_minimal()



According to the result of the minimum and maximum of the daily MCP analysis, Minimum value is observed in the 13th day of July, and maximum value of the minimum daily MCP is observed in the 3th day of the July. In addition to the minimum MCPs’, minimum and maximum of the maximum daily MCP value are observed in 10th July and 6th, 24th, 27th and 29th July, respectively.

#### 2.1.6. Daily MCP

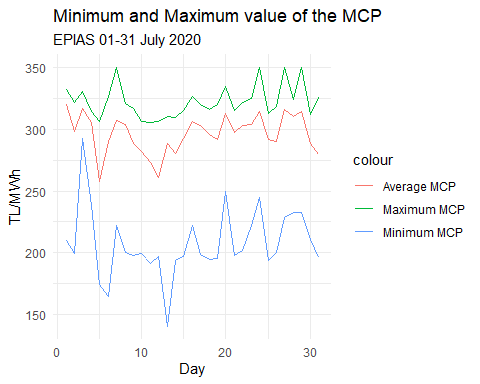
* DA\_MCP : Daily Average MCP value

raw\_df %>%  
 group\_by(Day=lubridate::day(Date))%>%  
 summarise(DA\_MCP = mean(MCP))

## # A tibble: 31 x 2  
## Day DA\_MCP  
## <int> <dbl>  
## 1 1 320.  
## 2 2 298.  
## 3 3 317.  
## 4 4 306.  
## 5 5 258.  
## 6 6 290.  
## 7 7 307.  
## 8 8 304.  
## 9 9 288.  
## 10 10 282.  
## # ... with 21 more rows

#### 2.1.7. Graph of daily MCP

#Minimum and maximum value of the MCP according to the day  
raw\_df %>%  
 group\_by(Day=lubridate::day(Date))%>%  
 summarise(MinMCP = min(MCP), MaxMCP = max(MCP), Average = mean(MCP))%>%  
 ggplot(aes(x=Day)) +  
 geom\_line(aes(y = MinMCP, color = "Minimum MCP")) +  
 geom\_line(aes(y = MaxMCP, color = "Maximum MCP")) +  
 geom\_line(aes(y = Average, color = "Average MCP")) +  
 labs(x = "Day",  
 y = "TL/MWh",  
 title = "Minimum and Maximum value of the MCP",  
 subtitle = "EPIAS 01-31 July 2020") +  
 theme\_minimal()



According to the result of the minimum and maximum of the daily MCP analysis, Minimum value is observed in the 13th day of July, and maximum value of the minimum daily MCP is observed in the 3th day of the July. In addition to the minimum MCPs’, minimum and maximum of the maximum daily MCP value are observed in 10th July and 6th, 24th, 27th and 29th July, respectively.

#### 2.1.8. Percentage of change of daily MCP

EG <- raw\_df %>%  
 select(Date, MCP)%>%  
 group\_by(Day = lubridate::day(Date))%>%  
 summarise(DailyMCP = sum(MCP)) %>%  
 ungroup()%>%  
 mutate(DailyMCPNext=lag(DailyMCP,1))%>%  
 transmute(Day, DailyMCP,DailyMCPNext, percentage=((DailyMCPNext-DailyMCP)/DailyMCP)-1)%>%  
 arrange(desc(percentage))  
  
#knitr::kable(EG, format="markdown")

#### 2.2. SMP

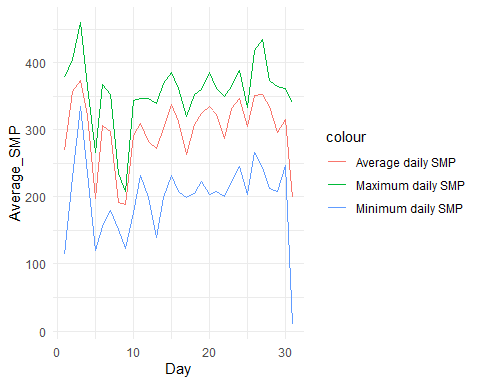
#### 2.2.1 Hourly change of the SMP

raw\_df%>%  
 group\_by(Day=lubridate::day(Date))%>%  
 summarise(Average\_SMP = mean(SMP),  
 Min\_SMP = min(SMP),  
 Max\_SMP = max(SMP))

## # A tibble: 31 x 4  
## Day Average\_SMP Min\_SMP Max\_SMP  
## <int> <dbl> <dbl> <dbl>  
## 1 1 268. 114. 377.  
## 2 2 357. 229. 404.  
## 3 3 373. 335. 460   
## 4 4 321. 235 365   
## 5 5 198. 120 266.  
## 6 6 307. 156. 367.  
## 7 7 298. 179 353.  
## 8 8 191. 151. 235.  
## 9 9 189. 124. 207.  
## 10 10 292. 177. 344.  
## # ... with 21 more rows

#### 2.2.2 Graph of hourly change of the SMP

raw\_df%>%  
 group\_by(Day=lubridate::day(Date))%>%  
 summarise(Average\_SMP = mean(SMP),  
 Min\_SMP = min(SMP),  
 Max\_SMP = max(SMP)) %>%  
   
 ggplot(aes(x=Day, color = "Days"))+  
 geom\_line(aes(y = Average\_SMP, color = "Average daily SMP")) +  
 geom\_line(aes(y = Min\_SMP, color = "Minimum daily SMP")) +  
 geom\_line(aes(y = Max\_SMP, color = "Maximum daily SMP")) +  
 theme\_minimal()



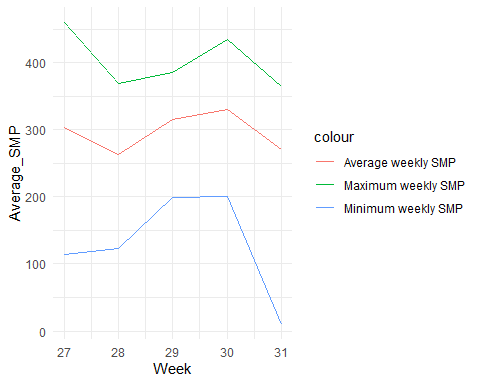
#### 2.2.3 Weekly change of the SMP

EG<- raw\_df%>%  
 group\_by(Week=lubridate::week(Date))%>%  
 summarise(Average\_SMP = mean(SMP),  
 Min\_SMP = min(SMP),  
 Max\_SMP = max(SMP))  
#this line provide table format in the R markdown page  
knitr::kable(EG, format="markdown")

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Average\_SMP | Min\_SMP | Max\_SMP |
| 27 | 303.1767 | 113.75 | 460.00 |
| 28 | 262.5493 | 123.75 | 368.78 |
| 29 | 314.6715 | 198.55 | 385.76 |
| 30 | 329.6039 | 201.18 | 435.00 |
| 31 | 269.8582 | 10.00 | 365.00 |

#### 2.2.4 Graph of weekly change of the SMP

raw\_df%>%  
 group\_by(Week=lubridate::week(Date))%>%  
 summarise(Average\_SMP = mean(SMP),  
 Min\_SMP = min(SMP),  
 Max\_SMP = max(SMP)) %>%  
   
 ggplot(aes(x=Week, color = "Days"))+  
 geom\_line(aes(y = Average\_SMP, color = "Average weekly SMP")) +  
 geom\_line(aes(y = Min\_SMP, color = "Minimum weekly SMP")) +  
 geom\_line(aes(y = Max\_SMP, color = "Maximum weekly SMP")) +  
 theme\_minimal()



#### 2.3. Relationship Between MCP and SMP

#### 2.3.1. MCP and SMP

We can obtain daily average value of the MCP and SMP by using following lines:

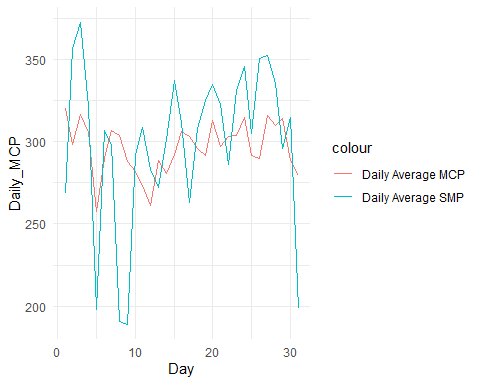
#average value of the MCP in weeks  
EG <- raw\_df %>%  
 group\_by(Day = lubridate::day(Date))%>%  
 summarise(Average\_MCP = mean(MCP),  
 Average\_SMP = mean(SMP),  
 Difference = abs(mean(MCP) - mean(SMP))) %>%  
 ungroup()%>%  
 arrange(desc(Difference))  
  
knitr::kable(EG, format="markdown")

|  |  |  |  |
| --- | --- | --- | --- |
| Day | Average\_MCP | Average\_SMP | Difference |
| 8 | 304.0046 | 190.8692 | 113.13542 |
| 9 | 288.2375 | 188.8638 | 99.37375 |
| 31 | 279.5725 | 198.7842 | 80.78833 |
| 26 | 289.9171 | 350.4929 | 60.57583 |
| 5 | 257.5275 | 197.7721 | 59.75542 |
| 2 | 298.2683 | 356.9079 | 58.63958 |
| 3 | 316.6133 | 372.7733 | 56.16000 |
| 1 | 320.1354 | 268.3675 | 51.76792 |
| 15 | 292.4921 | 337.4846 | 44.99250 |
| 17 | 303.2704 | 262.7879 | 40.48250 |
| 27 | 315.8171 | 352.7588 | 36.94167 |
| 11 | 273.2638 | 309.0046 | 35.74083 |
| 19 | 291.8637 | 324.9617 | 33.09792 |
| 24 | 314.9321 | 345.9442 | 31.01208 |
| 23 | 303.9325 | 331.4333 | 27.50083 |
| 30 | 288.3767 | 315.0079 | 26.63125 |
| 28 | 310.1596 | 335.2754 | 25.11583 |
| 21 | 297.4667 | 322.5717 | 25.10500 |
| 12 | 261.0908 | 283.1854 | 22.09458 |
| 20 | 313.2696 | 334.8942 | 21.62458 |
| 14 | 280.7529 | 301.7425 | 20.98958 |
| 29 | 314.4367 | 295.7825 | 18.65417 |
| 22 | 303.5113 | 286.3546 | 17.15667 |
| 6 | 290.1829 | 306.9533 | 16.77042 |
| 13 | 288.8896 | 272.3062 | 16.58333 |
| 4 | 306.0008 | 321.3442 | 15.34333 |
| 25 | 291.8100 | 304.9679 | 13.15792 |
| 18 | 296.2083 | 308.0858 | 11.87750 |
| 10 | 281.8229 | 291.8733 | 10.05042 |
| 7 | 306.9558 | 298.1183 | 8.83750 |
| 16 | 306.3796 | 311.9146 | 5.53500 |

#### 2.3.2. Graph of the relationship between MCP and SMP

Graph of the daily average value of MCP and SCP with line graph

#average value of the MCP in weeks  
raw\_df %>%  
 group\_by(Day = lubridate::day(Date))%>%  
 summarise(Daily\_MCP = mean(MCP),  
 Daily\_SMP = mean(SMP)) %>%  
   
 ggplot(aes(x = Day)) +  
 geom\_line(aes(y = Daily\_MCP, color = "Daily Average MCP")) +  
 geom\_line(aes(y = Daily\_SMP, color = "Daily Average SMP")) +  
 theme\_minimal()



#### 2.3.3. The weekly relationship between MCP and SMP

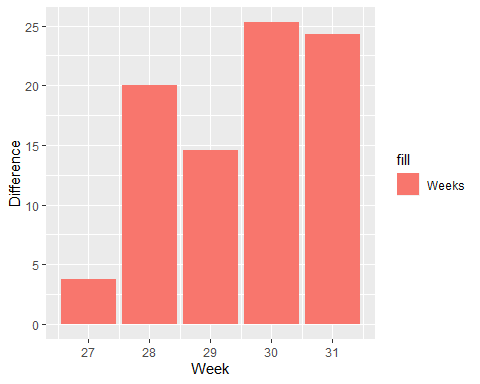
We can obtain weekly average value of the MCP and SMP by using following lines:

#average value of the MCP in weeks  
raw\_df %>%  
 group\_by(Week = lubridate::week(Date))%>%  
 summarise(Weekly\_MCP = mean(MCP),  
 Weekly\_SMP = mean(SMP),  
 Difference = abs(mean(MCP) - mean(SMP))) %>%  
 ungroup()%>%  
 arrange(desc(Difference))

## # A tibble: 5 x 4  
## Week Weekly\_MCP Weekly\_SMP Difference  
## <dbl> <dbl> <dbl> <dbl>  
## 1 30 304. 330. 25.3   
## 2 31 294. 270. 24.3   
## 3 28 283. 263. 20.0   
## 4 29 300. 315. 14.5   
## 5 27 299. 303. 3.79

#### 2.3.4. Graph of the weekly relationship between MCP and SMP

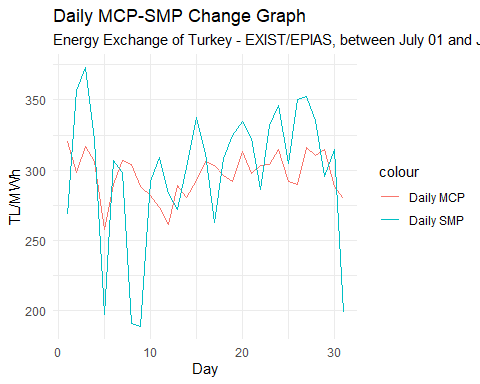
#average value of the MCP in weeks  
raw\_df %>%  
 group\_by(Week = lubridate::week(Date))%>%  
 summarise(Weekly\_MCP = mean(MCP),  
 Weekly\_SMP = mean(SMP),  
 Difference = abs(mean(MCP) - mean(SMP))) %>%  
  
 ggplot(aes(x = Week, y = Difference, fill = "Weeks")) +  
 geom\_bar(stat = "identity")



#### 2.3.5. Graph of the daily relationship between MCP and SMP

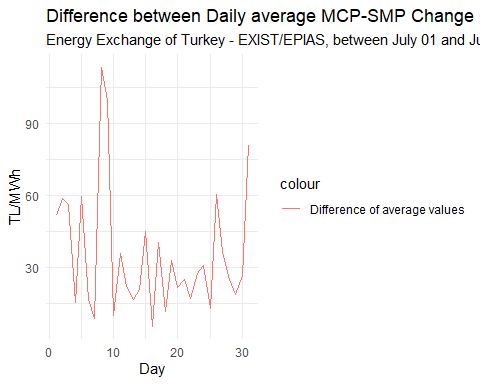
After the calculation of the weekly average, we can visualize the daily average of MCP and SMP values by following.

#Daily change of average change MCP-SMP  
raw\_df %>%  
 group\_by(Time=lubridate::day(Date))%>%  
 summarise(Daily\_MCP = mean(MCP), Daily\_SMP = mean(SMP)) %>%  
 ggplot(aes(x=Time)) +   
 geom\_line(aes(y=Daily\_MCP, color = "Daily MCP")) +  
 geom\_line(aes(y=Daily\_SMP, color = "Daily SMP")) +  
 labs(x = "Day", y = "TL/MWh",  
 title = "Daily MCP-SMP Change Graph",  
 subtitle = "Energy Exchange of Turkey - EXIST/EPIAS, between July 01 and July 31 ")+  
 theme\_minimal()



#### 2.3.6. Graph of the daily relationship between MCP and SMP

#Daily change of average change MCP-SMP  
raw\_df %>%  
 group\_by(Time=lubridate::day(Date))%>%  
 summarise(Daily\_MCP = mean(MCP),   
 Daily\_SMP = mean(SMP),  
 Difference = abs(mean(MCP)-mean(SMP))) %>%  
 ggplot(aes(x=Time)) +   
 geom\_line(aes(y = Difference, color = "Difference of average values")) +  
 labs(x = "Day", y = "TL/MWh",  
 title = "Difference between Daily average MCP-SMP Change Graph",  
 subtitle = "Energy Exchange of Turkey - EXIST/EPIAS, between July 01 and July 31 ")+  
 theme\_minimal()

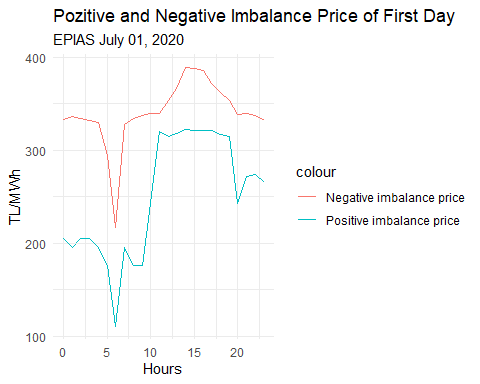


The results show that the highest difference between the daily average value of the MCP and daily average value of the SMP is observed in the eighth day.

### 3. Posivite and Negative Imbalance Price

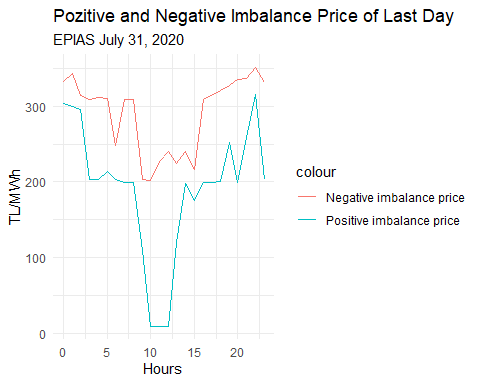
#### 3.1. Negative and positive inbalance price in first day

raw\_df %>%  
 select(Date, PositiveIP, NegativeIP)%>%  
 group\_by(Day = lubridate::day(Date), Hour = lubridate::hour(Date))%>%  
 select(Day,Hour, PositiveIP, NegativeIP)%>%  
 filter(Day==1)%>%  
   
 ggplot(aes(Hour, color = "Hours"))+  
 geom\_line(aes(y=PositiveIP, color = "Positive imbalance price")) +  
 geom\_line(aes(y=NegativeIP, color = "Negative imbalance price")) +  
 labs(x = "Hours", y = "TL/MWh",  
 title= "Pozitive and Negative Imbalance Price of First Day",  
 subtitle = "EPIAS July 01, 2020") +  
 theme\_minimal()



#### 3.2. Negative and positive inbalance price in last day

raw\_df %>%  
 select(Date, PositiveIP, NegativeIP)%>%  
 group\_by(Day = lubridate::day(Date), Hour = lubridate::hour(Date))%>%  
 select(Day,Hour, PositiveIP, NegativeIP)%>%  
 filter(Day==31)%>%  
   
 ggplot(aes(Hour, color = "Hours"))+  
 geom\_line(aes(y=PositiveIP, color = "Positive imbalance price")) +  
 geom\_line(aes(y=NegativeIP, color = "Negative imbalance price")) +  
 labs(x = "Hours", y = "TL/MWh",  
 title= "Pozitive and Negative Imbalance Price of Last Day",  
 subtitle = "EPIAS July 31, 2020") +  
 theme\_minimal()



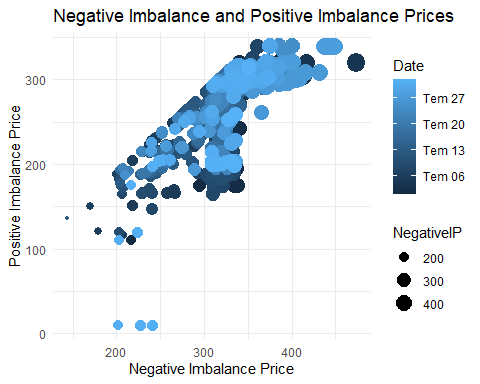
#### 3.3. Daily difference of negative and positive inbalance price

EG <- raw\_df %>%  
 select(Date, PositiveIP, NegativeIP)%>%  
 group\_by(Day = lubridate::day(Date), Hour = lubridate::hour(Date))%>%  
 select(Day,Hour, PositiveIP, NegativeIP)%>%  
 transmute(Day,Hour,PositiveIP, NegativeIP, difference=NegativeIP-PositiveIP)%>%  
 ungroup()%>%  
 arrange(desc(difference)) %>%  
 print(n=10)

## # A tibble: 744 x 5  
## Day Hour PositiveIP NegativeIP difference  
## <int> <int> <dbl> <dbl> <dbl>  
## 1 31 12 9.7 241. 231.  
## 2 31 11 9.7 227. 217.  
## 3 31 10 9.7 202. 192.  
## 4 1 9 176. 338. 162.  
## 5 1 8 176. 335. 159.  
## 6 3 14 320. 474. 154.  
## 7 3 15 321. 474. 153.  
## 8 8 0 174. 322. 148.  
## 9 7 23 174. 320. 146.  
## 10 8 1 175. 320. 146.  
## # ... with 734 more rows

#### 3.4. Graph of daily difference of negative and positive inbalance price

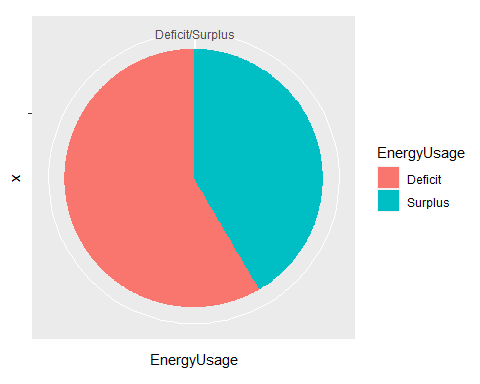
raw\_df%>%   
 ggplot(aes(x=NegativeIP, y=PositiveIP, color=Date, size = NegativeIP)) +   
 geom\_point() +   
 labs(x ="Negative Imbalance Price",   
 y ="Positive Imbalance Price",   
 title="Negative Imbalance and Positive Imbalance Prices") +  
 theme\_minimal()



Point data graph shows that, negative imbalance price and positive imbalance price concentrate in higher value. #### 4. SMP Direction Another implementation for the energy usage is about SMP direction.

#### 4.1. Pie chart of SMP direction

raw\_df %>%  
 group\_by(Hours = lubridate::hour(Date))%>%  
 transmute(EnergyUsage = ifelse(SMPDirection == "?Energy Surplus", "Surplus", "Deficit")) %>%  
 ggplot(aes(x = "", y=EnergyUsage, fill=EnergyUsage)) +  
 geom\_bar(stat = "identity", width = 1) +  
 coord\_polar("y")



Pie chart shows that, the percentage of the deficit is higher than the percentage of the surplus. That is, the energy demand is higher than the predicted energy demand.

#### 4.2. Number of deficit according to the days

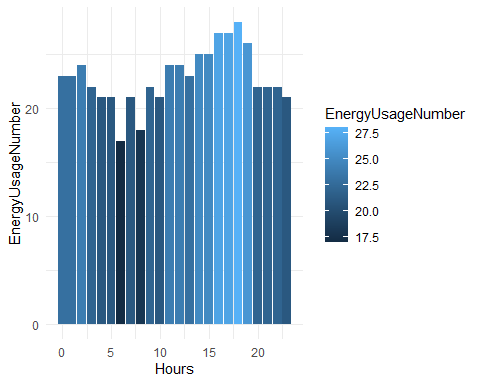
raw\_df %>%  
 group\_by(Hours = lubridate::hour(Date))%>%  
 transmute(EnergyUsage = ifelse(SMPDirection == "?Energy Surplus", "Surplus", "Deficit")) %>%  
 group\_by(EnergyUsage, Hours) %>%  
 filter(EnergyUsage=="Deficit") %>%  
 summarise(EnegyUsageNumber = n())%>%  
 print(n=10)

## # A tibble: 24 x 3  
## # Groups: EnergyUsage [1]  
## EnergyUsage Hours EnegyUsageNumber  
## <chr> <int> <int>  
## 1 Deficit 0 23  
## 2 Deficit 1 23  
## 3 Deficit 2 24  
## 4 Deficit 3 22  
## 5 Deficit 4 21  
## 6 Deficit 5 21  
## 7 Deficit 6 17  
## 8 Deficit 7 21  
## 9 Deficit 8 18  
## 10 Deficit 9 22  
## # ... with 14 more rows

Deficit value is important because when we find the deficit value, we can obtain the under predicted demand. That is, if the deficit occurs, we use more electricity than prediction.

#### 4.3. Graph of number of deficit according to the days

raw\_df %>%  
 group\_by(Hours = lubridate::hour(Date))%>%  
 transmute(EnergyUsage = ifelse(SMPDirection == "?Energy Surplus", "Surplus", "Deficit")) %>%  
 group\_by(EnergyUsage, Hours) %>%  
 filter(EnergyUsage=="Deficit") %>%  
 summarise(EnergyUsageNumber = n())%>%  
   
 ggplot(aes(x = Hours, y = EnergyUsageNumber, fill = EnergyUsageNumber)) +  
 geom\_bar(stat = "identity") +  
 theme\_minimal()



This bar chart shows the hourly number of deficit energy usage. This result shows that the more deficit energy usage observed in the 6 pm.In other words, this graph also shows the energy usage.In toher words, this data show that the eighth and ninth days have no deficit.

#### 4.4. Classification of the energy surplus, energy deficit and balance

When we use the equality to calculate SMP direction, we can obtain surplus, deficit, and balance

EG <- raw\_df %>%   
 group\_by(Hour = lubridate::hour(Date))%>%   
 transmute(Hour, Surplus = sum(MCP>SMP), Deficit=sum(MCP<SMP), Balance=sum(MCP==SMP))%>%  
 print(n=5)

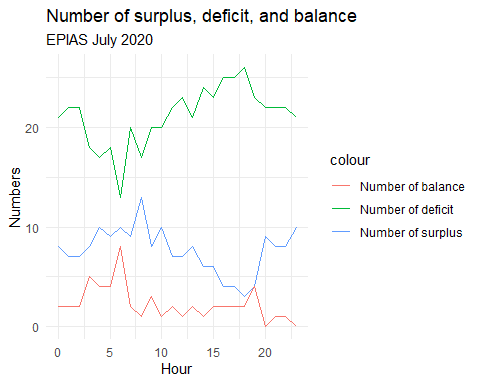
## # A tibble: 744 x 4  
## # Groups: Hour [24]  
## Hour Surplus Deficit Balance  
## <int> <int> <int> <int>  
## 1 0 8 21 2  
## 2 1 7 22 2  
## 3 2 7 22 2  
## 4 3 8 18 5  
## 5 4 10 17 4  
## # ... with 739 more rows

This data shows the number of deficit, surplus, and balance according to hour in a day. To make more clear analysis for this data we can use visualization, which are given in the next section.

#### 4.5. Graph of the energy surplus, energy deficit and balance

When we use the equality to calculate SMP direction, we can obtain surplus, deficit, and balance

raw\_df %>%   
 group\_by(Hour = lubridate::hour(Date))%>%   
 transmute(Hour, Surplus = sum(MCP>SMP), Deficit = sum(MCP<SMP), Balance = sum(MCP==SMP))%>%  
   
 ggplot(aes(x = Hour, color = "Hours")) +  
 geom\_line(aes(y = Surplus, color = "Number of surplus")) +  
 geom\_line(aes(y = Deficit, color = "Number of deficit")) +  
 geom\_line(aes(y = Balance, color = "Number of balance")) +  
   
 labs(x = "Hour",  
 y = "Numbers",  
 title = "Number of surplus, deficit, and balance",  
 subtitle = "EPIAS July 2020") +  
 theme\_minimal()



This graph shows that, especially between 15 and 20 pm, the number of deficit is greater than both number of surplus and balance. The deficit means that, the energy demand is more than the predicted energy. That is, in this time interval, the more energy used than the predicted. In contrast to this information, even though there is still more deficit value according to the surplus and balance, this value is closer to surplus and balance value.However, according to the daily analysis, in each hour, we have more deficit value than surplus and balance. That is, we use more energy than predict.

This report gives some basic information about monthly energy market, i.e., MCP,SMP, positive imbalance price, negative imbalance price, and SMP direction. More information about the electricity energy market, you can visit web page of the [EPIAS](https://www.epias.com.tr/).