

Electricity Monthly Mean Consumption Of Buildings

Data Science Project Protocol

Author: Gulst Shem

15/06/2022

Mentoring / Overview / presented to:

Dr. Tomas Karpati MD

A. Introduction

In the early 1990's I was a physics teacher, teaching high school students about the energy crisis of the 1970s - mainly the oil shortage. My students asked me: "why do we (the adults) don't do enough in our daily lives to change the situation."

Fast forwarding 15 years later; the 60-year-old school I was teaching in moved to a new building on the campus, equipped with the most comprehensive energy management technology; such as central climate control systems, lights turned off automatically after school hours, and new solar panels were installed on the rooftops.

All of that, and my physics training, made me very aware of energy usage in public and private buildings, as much so that when I was looking for data for my final project in Data Science and stumbled upon the "[Building Data Genome Project](#)" article describing the use of ML to understand and predict energy usage in buildings around the world, I knew that this is the data I want to get into.

From a more general perspective from my point of view; the world has moved a long way from the 1970s to our days concerning the energy consumption of buildings. The terms "carbon footprint" and "green-buildings", among others, were introduced to culture and economics. Nowadays, when planning a new building or just planning the building usage budget for the next year, one of the major issues to consider is the yearly/monthly energy consumption of the building and its costs.

Following the data sources of the article, I discovered the "sites dashboards" that give real-time several parameters indications of the energy consumption of the site's buildings, and open access to historical data gathered from the systems. This can help in creating a better understanding of the problem of "**what are the major factors that can contribute to more efficient building energy consumption?**" to the owners of the building and to others (like myself) that want to learn about the issue and might have some insights to improve the predictions.

This work will specifically attempt to forecast the "**Next Month mean electricity monthly consumption**" of buildings in sites with the available data up to that month.

Parameters like building characteristics such as location, size, usage, time, climate, and specific weather conditions are the most logical suspects. This work will use all these parameters and others available and try to figure out which ones influence the **Next Month's forecast of mean electricity monthly consumption** the most.

B. Methodology (Project design)

B.1. Data

B.1.1 Source data overview

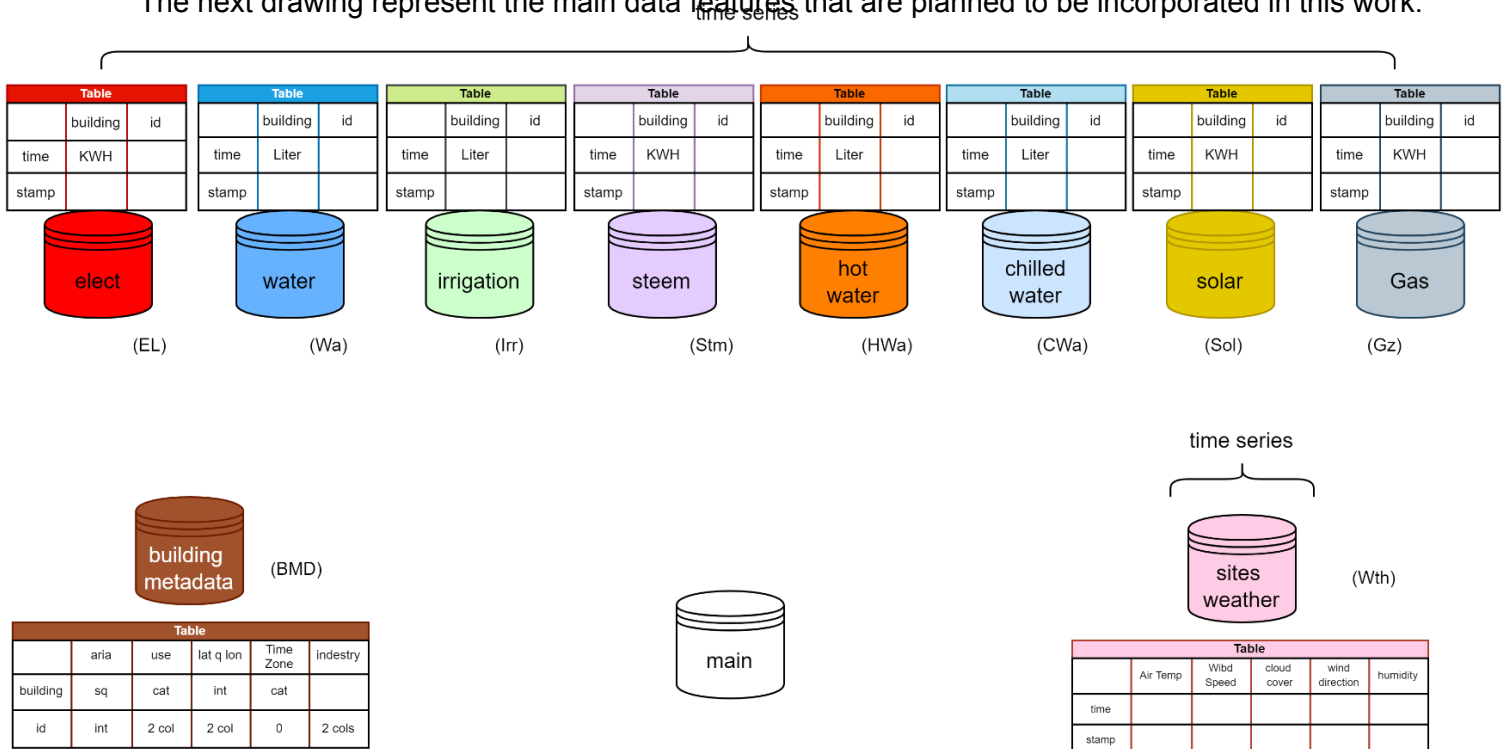
The data that will be used in this project is based on the “[Building Data Genome Project](#)” article and can be found in the affiliated public open to using GitHub repository: [buds-lab/building-data-genome-project-2](#).

There are two main classes of data; The data retrieved from the building sensors and weather in the sites - a Time Series hourly data from 2016, and 2017. The other is the building metadata containing several parameters of buildings characteristics and usage.

In contrast to the original data described, more specific data about the different sites was gathered to make some preliminary decisions (described later).

Treatment of normalizing physical units and cleaning other sensor problems was dealt with by the authors of the article and presented as “Clean Dat”, in contrast to “Raw Data”. During the process of studying the data, both data sets were considered and compared. In this work, the clean data was used as not all sensor information was available and the cleaning process is time-consuming (detailed data normalization and cleaning protocol are presented in the above article).

The next drawing represent the main data features that are planned to be incorporated in this work:



Figur 1: Main data sources

On the top row are the main data tables (csv files) from the 8 different sensors (meters). Each one has the same Time Series time gap of one hour collected over the years 2016 and 2017. In total 17,545 rows (365 X 24 X 2) and 1636 columns represent each building in 19 sites located in 6 different time zones. Several sensors gathered the data in KWH units and others in Letters.

On the right side of the bottom row - the "site's weather" is an 8 parameters Time Series data table, compatible with the sensor's data tables regarding the time gaps and period gathered. The parameters units of the data were gathered in standard common use units.

On the left of the bottom, the row is the metadata table with the building's usage and other characteristics. As mentioned earlier, further data was gathered about the locations of the sites and another table was created as shown in the center bottom row of Figure 1.

Two main issues were addressed during this stage of gathering and understanding the data;

1. Joining the Time Series data and the non-Time Series buildings and sites metadata.

This issue was addressed by electing an outcome and unique key id combined from the building id and the timestamp. Specifically, a building next month's mean electricity consumption as an outcome and a building-monthly id. Five of the other sensor (meter) data gathered were used as additional features to help predict the outcome. They were selected for their level of complete data and their association with air temperature changes during the seasons.

2. Data quality is crucial in creating a successful model and predictions. A careful investigation of the data, looking at the "clean data" sets, and summing up and evaluating which sites and buildings will contribute the most to the process led to the decision of which will be used. The following table presents the summation of the decision process from the [info on sites data](#) file:

Site	chilledwater No	electricity No	hotwater No	steam No	Gas No	Water No	Irrigation No	Buildings	Meters	Met cnt	Imp mt cnt
Panther	25	105	0	0	32	100	37	136	299	5	3
Robin	0	52	15	0	0	0	0	52	67	2	2
Fox	101	137	68	0	0	0	0	137	306	3	3
Bear	0	92	0	0	0	0	0	92	92	1	1
Rat	0	305	0	0	0	0	0	305	305	1	1
Lamb	0	146	0	0	119	0	0	147	265	2	2
Eagle	87	106	60	45	0	0	0	47	106	4	4
Moose	15	13	3	12	0	0	0	15	43	4	4
Gator	0	74	0	0	0	0	0	74	74	1	1
Bull	95	123	0	90	0	0	0	124	308	3	3
Bobcat	22	35	16	0	8	30	0	36	116	4	3
Crow	5	5	5	0	0	0	0	5	15	3	3
Wolf	0	36	0	0	14	16	0	36	66	3	2
Hog	87	152	0	45	0	0	0	163	284	3	3
Peacock	27	45	0	34	0	0	0	106	298	3	3
Cockatoo	71	117	2	92	0	0	0	124	282	4	4
Shrew	0	9	0	0	4	0	0	9	13	2	1
Swan	20	19	16	0	0	0	0	21	55	3	3
Mouse	0	7	0	0	0	0	0	7	7	1	1
buildings with data acording to building metadata - from 1636 buildings								Tot Selc Bui	Tot Sit Met		
	555	1578	185	370	177	146	37	1636	3053	tot m	
	33.92%	96.45%	11.31%	22.62%	10.82%	8.92%	2.26%	914	2112	sel m	
	555	857	170	318	40	1940	63.54%	914/1636	11/19	57.89%	
	100.00%	54.31%	91.89%	85.95%	22.60%	% of data usede from each meter		buildings	sites	data used	
	33.92%	52.38%	10.39%	19.44%	2.44%	63.54%	% from all meters	55.87%	69.18%	%	

Table 1: Summation of the data selection process

Table 1 shows in which sites and in which meters data were gathered (the process cross validates the data presented in the building metadata table with the clean data files to make sure it corresponded to them). To make the decision on which meters and sites to use, two parameters were used;

- A meter has data from at least more than 10% of the buildings.
(marked orange at the table botom).
- A site has at least 3* meters that data was gathered.
(Met cnt column, ≥ 3 marked in red).
- Sites that have at least 3* meters data of more than 10% (crossing the two above).
(imp mt cnt column, ≥ 3 marked in yellow)

(* The 3 number was selected as 1 or 2 meters data from 5 available pre-sites are too few to be useful, and as there were only 5 sites with 4 and 5 meters data selecting ≥ 3 where there are 11 sites seemed logical).

The sites selected by these criteria were marked in green. The meters selected were marked in orange. Summing up (right bottom corner) :

- 57.89% of the sites were used (11/19).
- 55.87 of the total buildings were used (914/1636).
- 63.54% of the meters were used (1940/3053).

Further preliminary investigation of the site's buildings data was conducted (can be found in the second sheet of the [info on sites data](#) file) and revealed that there are several buildings with more than 10% missing data in the selected sites and meters. One in particular "Bobcat_education_Seth" had 10% missing data in 4 of the 5 meters selected. This led to a decision to drop this building too. That leaves a total of 913 buildings.

The source article "[Building Data Genome Project](#)" referred to GitHub Wiki which presents the outcomes of the Kaggle competition RMSLE 4.95 for [long term](#) and 3.585 for winter and 5.105 for summer on [short term](#) predictions. This process of selecting quality data aims to try and get better results.

The following figure 2 shows the complete picture of analyzing the data gathering at this stage.

On the left bottom side, it shows that reducing 19 sites to 11 gives 914 buildings from 1636 in the original data 913 after omitting Bobcat_education_Seth building. Combining building id with month for one year gives 10,956 unique rows for the flat file main dataset for 1 year (second year - 2017 will be used for tests).

More recent data (2018,2019) can be gathered from some of the [sites open website dashboards](#) archives to further test and validate the predictions.

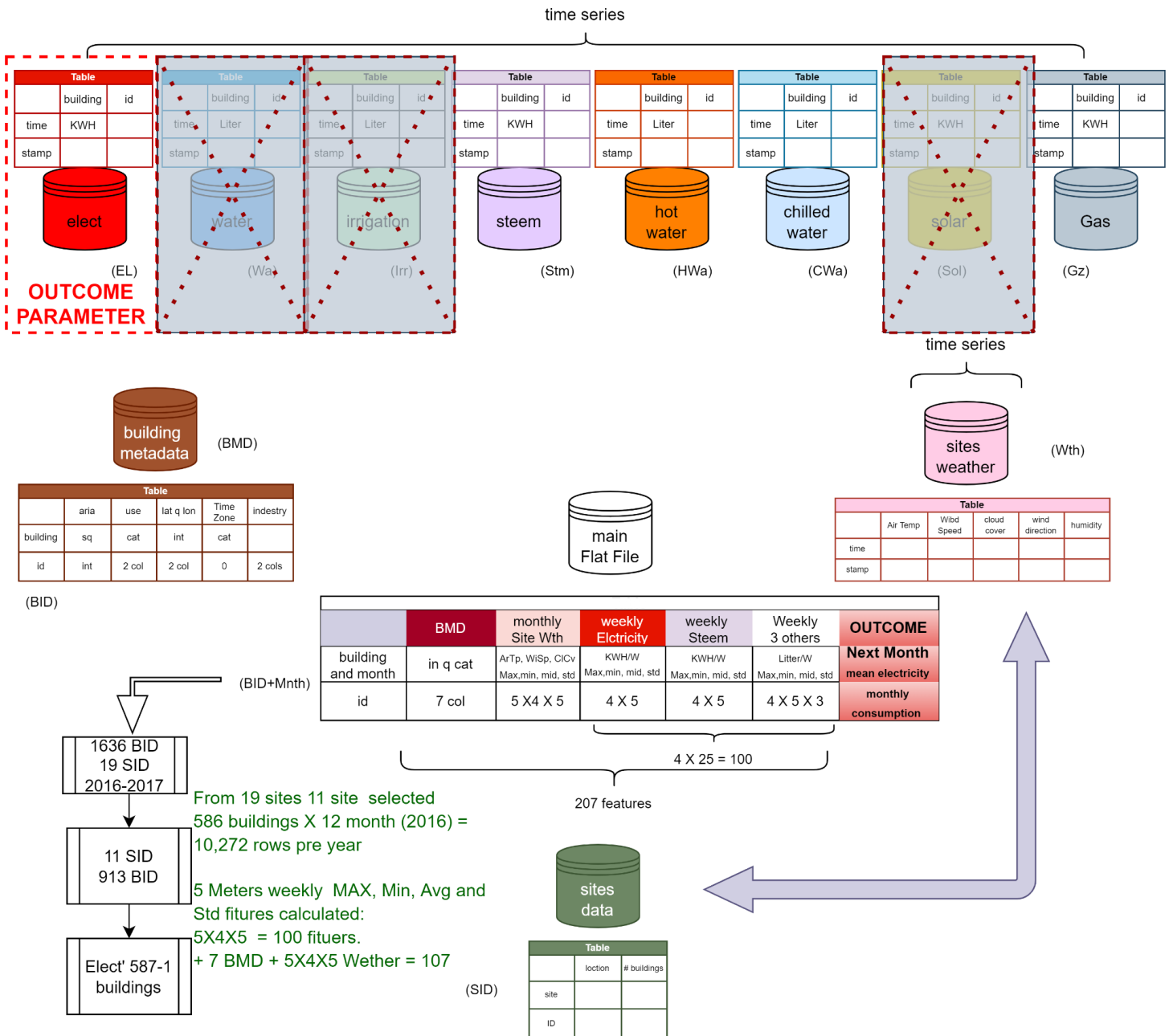


Figure 2: full scope of data gathering analysis and FlatFile assembly

B.1.2 Preliminary flat file design (figure 2 explained) ;

- The unique id key column will be a combination of the building id and month name (M-BID).
- 7 columns driven from the building metadata (BMD)^{note1}
 - Site id (SID) - Categorical (5 cat')
 - Building area in sqm (sqm) - Numeric
 - Building location latitude (lat) - Numeric
 - Building location longitude (lng) - Numeric
 - Building Time Zone (Tmz) - Categorical (5 cat')
 - Building primary usage (PrUs) - Categorical (16 cat')
 - Building sub usage (SbPrUs) -Categorical (102 cat') ^{note3}
- 5(par) X 4(aggr') X 5(weeks) weather parameters ^{note2}
 - Max, min, Avg, Std air temperature in celsius -Numeric, semi categorical (per site).
 - Max, min, Avg, Std dew temperature in celsius -Numeric, semi categorical (per site).
 - Max, min, Avg, Std sea level pressure in atm -Numeric, semi categorical (per site).
 - Max, min, Avg, Std wind speed in Km/h -Numeric, semi categorical (per site).
 - Max, min, Avg, Std wind direction in azimuth angle -Numeric, semi categorical (per site).
- 5 X 4 Electricity monthly summary consumption parameters Max, min, Avg, Std in KWH ^{note2}
- 5 X 4 gas monthly summary consumption parameters Max, min, Avg, Std in liters ^{note2}
- 5 X 4 steam monthly summary consumption parameters Max, min, Avg, Std in KWH ^{note2}
- 5 X 4 chilled water monthly summary consumption parameters Max, min, Avg, Std in liters ^{note2}
- 4 X 4 hot water monthly summary consumption parameters Max, min,Avg, Std in liters ^{note2}
- Categorical season parameter (Autumn, Winter, Spring, Summer)
- **Outcome - Building NEXT MONTH monthly means electricity consumption.**

Notes:

1. Original Building MetaData contains more features, nearly empty ones were disregarded.
2. Even though the outcome is monthly, the time series features (weather and meters) were aggregated to weeks(Max, Min, Avg, and Std), as the source data is hourly gathered, there is enough to have a good weekly parameter and give each of the 5 weeks in a month (4 weeks + residuals days) a fair chance to influence the next month outcome. The last week of the month will likely have a greater effect.
3. Building usage will be converted to numeric as it has more than 100 categories.

Summary:

101 features; 4 catecorial, 12 semi-categorical, 80 numerical
10,956 rows per year,
8 datasets/tables sources.

[Link to the full data retrieval protocols](#) spreadsheet.

B.2. Creating flat file process stages (In SQL):

- Clean data sets were downloaded from the buds-lab/building-data-genome-project-2 repository, using the “view csv” mode and “save to file” option (the regular download option saved only partial data sets).
- Preliminary data reviews were done in MS Excel.
- Datasets were uploaded to SQL Server through SQL import and export app (more than 250 column datasets were uploaded in parts and then rejoined in SQL management studio).
Section (A) in [sb SQLQuery EndPr 09052022 A.sql SQL File](#)
- A protocol to convert hourly time series data to weekly parameters (MAX, Min, Average, and Standard Deviation pivot tables) was created and checked for correct functioning with small datasets (gas meter) for 53 weeks per year, and then applied to all five meters (Electricity, Child Water, Gas, Hot Water, Steam). *Section (B) in [sb SQLQuery EndPr 09052022 B.sql SQL File](#)*
- Breakdown meters weekly tables to five weeks monthly tables were created. For 2016, the week's breakdown was checked with a calendar and matched accordingly. In detail:
 - Jan, Jul, and Oct of 2016 have 6 weeks (partial weeks)
all others have 5 weeks, to have even 5 weeks per month in them,
one week (that manifests in the next month) was dropped from them
 - The weeks to months brake down is:
 - Jan: 1-5, Feb: 6-10, Mar: 10-14, Apr: 14-18, May: 19-23, Jun: 23-27
 - Jul: 27-31, Aug: 32-36, Sep: 36-40, Oct: 40-44, Nov: 45-49, Dec: 49-53

In the process of creating the five-week monthly meter tables a unique building month ID Was created by concatenating the month's name shortcut with the building name (for example: Jan + Hog_office_Garrett = “Jan_Hog_office_Garrett”). *Section (C) in [Sb SQLQuery EndPr 09052022 C.sql SQL File](#)*

- As the week names were mistakenly arbitrarily named Wk1-Wk5 without meter and parameter names, it was corrected at the end of each meter treatment by renaming all the week's column names. *Sections (C2a, C3a, C4a, C5a, C6a, C7a) in [Sb SQLQuery EndPr 09052022 C.sql SQL File](#)*
- One year per meter (Electricity: “El”, Child Water: “ChWt”, Gas: “gas”, Hot Water: “hotwater”, Steam: “Stm”) tables were created by joining (insert) the monthly tables month under a month.
Electricity table first, as a basis for the Flat File, as it has all the Month_Building ID unique rows (858) adding up to 10,296 (=858X12) rows. *Section (D.1) in [sb SQLQuery EndPr 09052022 FF1 D.sql SQL File](#)*

- All parameter (Max,Min,Avg,Std) tables of each meter (EI, ChWt, gas, hotwater, Stm) were joined to a relevant meter table. [Section \(D.2\) in sb_SQLQuery_EndPr_09052022_FF1_D.sql SQL File](#)
- All meter tables were joined together to one temporary Flat File (View) of 101 columns. [Section \(D.3\) in sb_SQLQuery_EndPr_09052022_FF1_D.sql SQL File](#)
- Treatment of the Meta Data Table [Section \(F\) in Sb_SQLQuery_EndPr_09052022_FF1_F.sql SQL File](#)
 - A relevant Building Metadata table was created by:
 - * The columns with more than 30% nulls were dropped
 - ** The rows (buildings id) were filtered to 'Yes' on the electricity column to match the building meter table, then the column was dropped.
 - *** The 11 /19 sites selected in the preview data stage were selected
 - A month short name was added to the building name, multiplying the rows pre each month and then add to one table parallel in monthly building id to the meters table.
 - The two tables; BMD and meters parameters weekly summaries were joined (the procedure included organizing the table so that the BMD will be first)
- Creating the OutCome Column: next month's monthly average electricity consumption:
 - As the EI table is too large (as in the weekly summary), two monthly tables were created by using the same procedure used in the weekly summary (altering the code for DATENAME(WW, TmSt) to MONTH(TmSt)). [Section\(B-OC.1\) in Sb_SQLQuery_EndPr_09052022_FF1_E.sql SQL File](#)
 - Shifted Building monthly ID was created by using the Month number of the EI_Avg, and the Building name. Explicitly; the monthly average electricity consumption of February was attached to January "Jan_building_name" etc. [Section \(B-OC.2.1\) in Sb_SQLQuery_EndPr_09052022_FF1_E.sql SQL File SQL File](#)
 - The two result tables were joined (rows) to one "Sh.A_EL_Mn_OC" table (Rat buildings were omitted as they are not represented in the other meters selected). Unrelevant columns were deleted, and column names were altered to prepare the table to join with the FF. [Section \(B-OC.2.2\) in Sb_SQLQuery_EndPr_09052022_FF1_E.sql SQL File SQL File](#)
 - The final "Sh.A_EL_Nxt_Mn_OC" table was joined with the "Sh.BGP_FF" table, dropping the month number column, after rechecking the monthly average electricity consumption shifted accordingly. [Section \(B-OC.3\) in Sb_SQLQuery_EndPr_09052022_FF1_E.sql SQL File SQL File](#)

C. EDA (Exploratory Data Analysis) stages (In Jupiter R):

- The flat file created as described in the previous section was uploaded directly from SQLServe to Jupiter notebook with R kernel to a data - frame named “df”.
- [Section \(A\) in file: R EDA - descriptive statistics and correletions BGP_01062022](#)
- A regular summary was applied to get the data statistics. [Section \(A.1.\)](#)
- String / Character features were converted to factors. [Section \(A.2.\)](#)
- Mechkar code was used for more rigorous data statistics (table1). [Section \(A.3.\)](#)
- Another statistics diagnostics was applied using *dlooker* library code.
- The distribution of variables and relationship with the outcome was investigated using Mechkar (explore data function). [Section \(A.4.\)](#)

Some obvious results appeared others less. (see samples Tables 2.1, 2.2 below)

Variable	Distribution	Descriptive Statistics	Outliers	Dependent Variable Distribution
sqm		Data type: Continuous Data length: 10272 / 10272 (100 %) Missing: 0 (0 %) Mean: 1.02e+04 StdDev: 1.077e+04 Median: 6930 IQR: 3447 - 1.302e+04 Min: 26.3 Max: 8.004e+04	 Outlier values: There are 46 outlier values	
EI_Min_W5		Data type: Continuous Data length: 9115 / 10272 (88.74 %) Missing: 1157 (11.26 %) Mean: 124.3 StdDev: 215.1 Median: 54.08 IQR: 20.38 - 135 Min: 0.0028 Max: 3062	 Outlier values: There are 878 outlier values	
EI_AVG_W5		Data type: Continuous Data length: 9115 / 10272 (88.74 %) Missing: 1157 (11.26 %) Mean: 177.8 StdDev: 276.7 Median: 89.52 IQR: 37.96 - 193.7 Min: 0.0028 Max: 5658	 Outlier values: There are 959 outlier values	
ChWt_AVG_W5		Data type: Continuous Data length: 5964 / 10272 (58.06 %) Missing: 4308 (41.94 %) Mean: 5.536e+04 StdDev: 2.806e+05 Median: 465.1 IQR: 65.04 - 2213 Min: 0 Max: 7.918e+06	 Outlier values: There are 1058 outlier values	

Table 2.1 - Samples from data visualization & exploration using Mechkar code

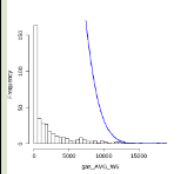
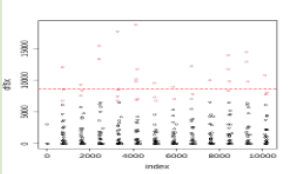
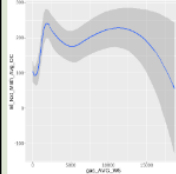
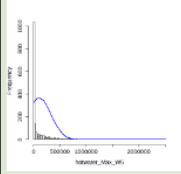
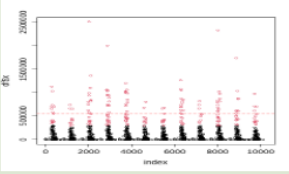
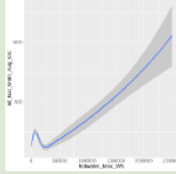
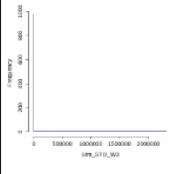
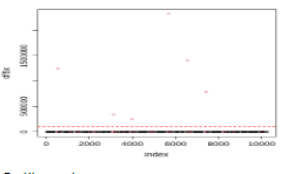
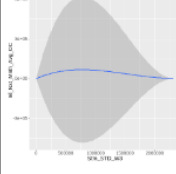
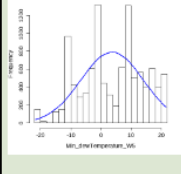
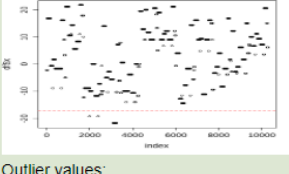
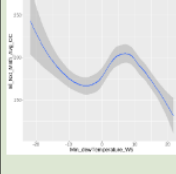
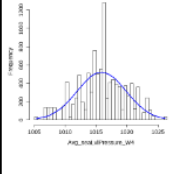
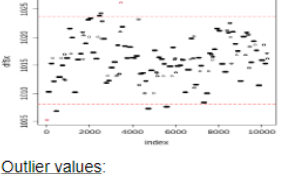
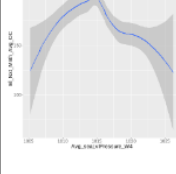
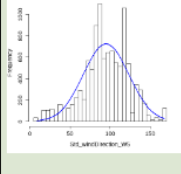
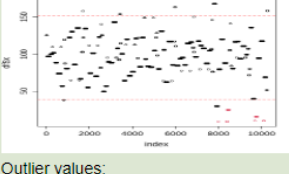
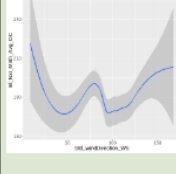
Variable	Distribution	Descriptive Statistics	Outliers	Dependent Variable Distribution
gas_AVG_W5		Data type: Continuous Data length: 374 / 10272 (3.641 %) Missing: 9898 (96.36 %) Mean: 2171 StdDev: 3201 Median: 728.5 IQR: 63.32 - 2686 Min: 0 Max: 1.866e+04	 Outlier values: There are 38 outlier values	
hotwater_Max_W5		Data type: Continuous Data length: 1833 / 10272 (17.84 %) Missing: 8439 (82.16 %) Mean: 1.084e+05 StdDev: 2.226e+05 Median: 1.031e+04 IQR: 127.7 - 1.184e+05 Min: 0 Max: 2.506e+06	 Outlier values: There are 204 outlier values	
Stm_STD_W3		Data type: Continuous Data length: 3656 / 10272 (35.59 %) Missing: 6616 (64.41 %) Mean: 1885 StdDev: 5.14e+04 Median: 32.76 IQR: 9.053 - 102.6 Min: 0 Max: 2.32e+06	 Outlier values: There are 481 outlier values	
Min_dewTemperature_W5		Data type: Continuous Data length: 10272 / 10272 (100 %) Missing: 0 (0 %) Mean: 3.705 StdDev: 10.37 Median: 4.8 IQR: -3.9 - 11.7 Min: -21.7 Max: 21.7	 Outlier values: No outlier values found	
Avg_seaLvPressure_W4		Data type: Continuous Data length: 10272 / 10272 (100 %) Missing: 0 (0 %) Mean: 1016 StdDev: 3.935 Median: 1016 IQR: 1014 - 1018 Min: 1005 Max: 1026	 Outlier values: 1005 - 1026	
Std_windDirection_W5		Data type: Continuous Data length: 10272 / 10272 (100 %) Missing: 0 (0 %) Mean: 94.86 StdDev: 28.23 Median: 94.13 IQR: 79.94 - 116.1 Min: 8.752 Max: 167.4	 Outlier values: 8.752, 24.39, 9.926, 15.28	

Table 2.2 - more Samples from data visualization & exploration using Mechkar code

The full table is in the file: "[report 20052022](#)"

- Parallel to using Mechkar, Sweet Vis in Python was used to get a statistical overview. Some samples of this report can be found below in Table3.

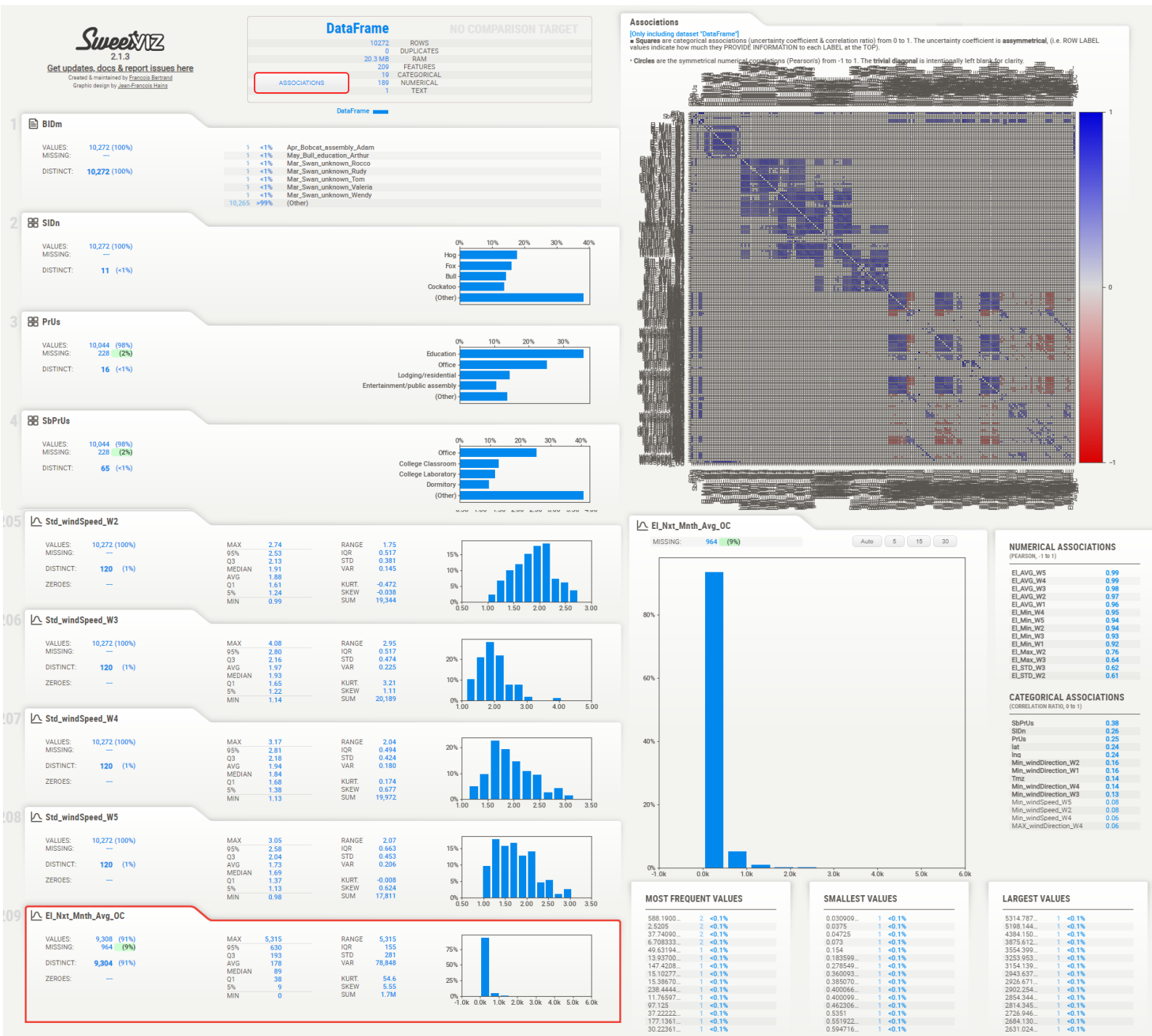


Table 3 - Sample report cards from Sweet Vis

Full table is in file : ["Sh.BGP FF OC 18052022 SW Wth"](#)

- Correlations were explored for numeric variables **corrgram** package (see the result in figure 3).

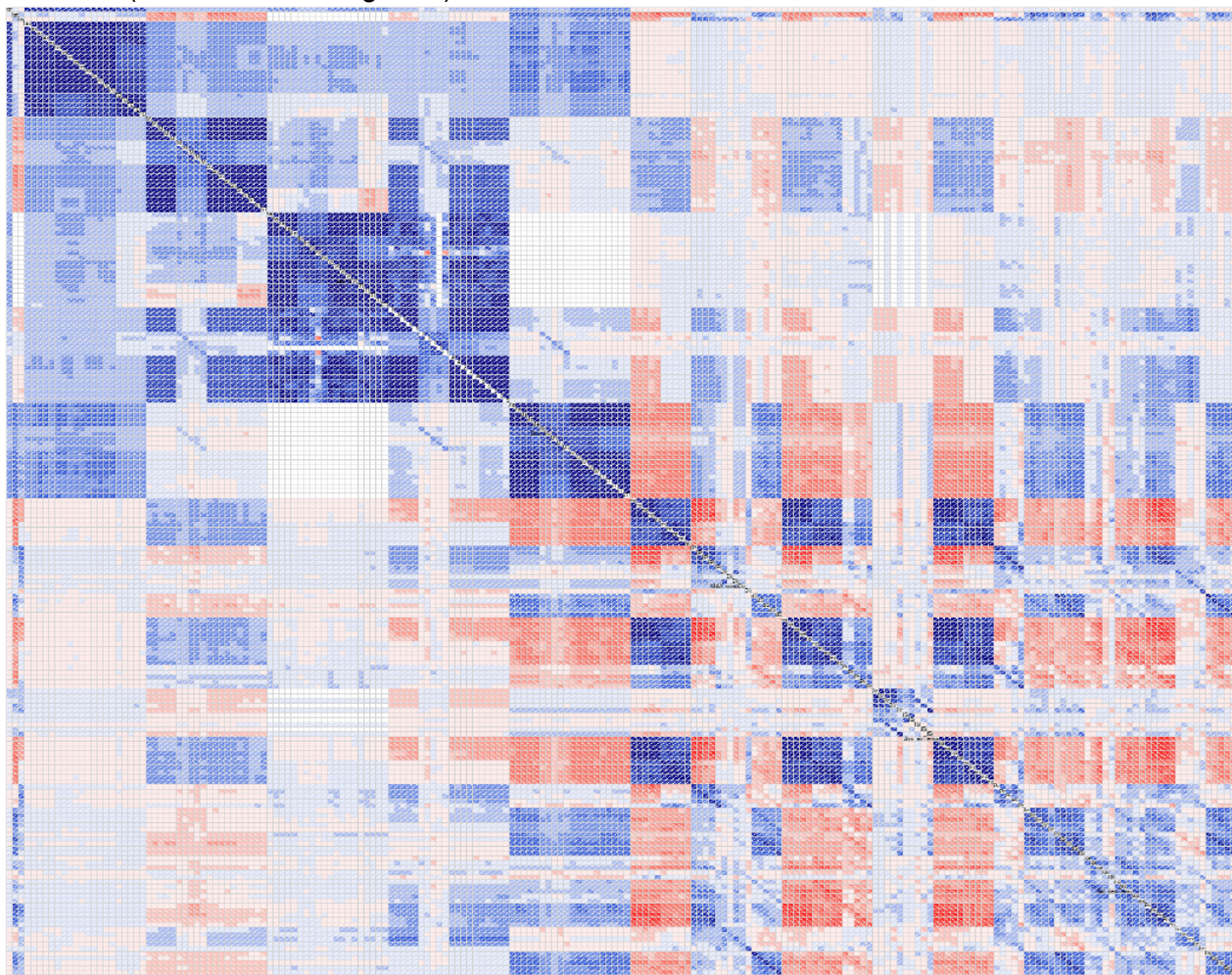


Figure 3 - Heat Map of numeric variables correlations

- An analytic investigation of correlations was performed ([Section \(B.2.\)](#)).
As viewed in the general statistics before and as expected;
Weekly electricity consumption has a high (>95%) correlation with the outcome.
So is the sqm - square meter area of the building (74%).
Other meters such as Steam (<55) and hot water (<3%) were less correlated to the outcome.
Surprisingly, weather variables had a generally low correlation (<2%) with the outcome.
This was seen in the general statistics as well.
At this point, it is early to make assumptions as to the reasons for that.

- A separated relationship between the categorical variables was explored ([Section \(B.3.1\)](#)) with Cramer's V based on ChiSquer (see table 4).

V1	V2	CramerV	p.value
<chr>	<chr>	<dbl>	<dbl>
SlDn	PrUs	0.21317	0
SlDn	SbPrUs	0.51482	0
SlDn	Tmz	1.00000	0
PrUs	SbPrUs	0.95744	0
PrUs	Tmz	0.17487	0
SbPrUs	Tmz	0.40832	0

Table 4 - Correlation table of all categorical variables

- A second test with a **companion** package gave similar results ([Section \(B.3.2\)](#))
- An exploration into the outcome variable “Electricity Consumption of the Next Month”
A distribution plot was drawn using **ggplot** and as the result implied a log plot was drawn. The log plot shows an almost “normal” distribution. It seems that in the Feature transformation stage, it will be worth checking the log function for use on most other meters (see figure 4).

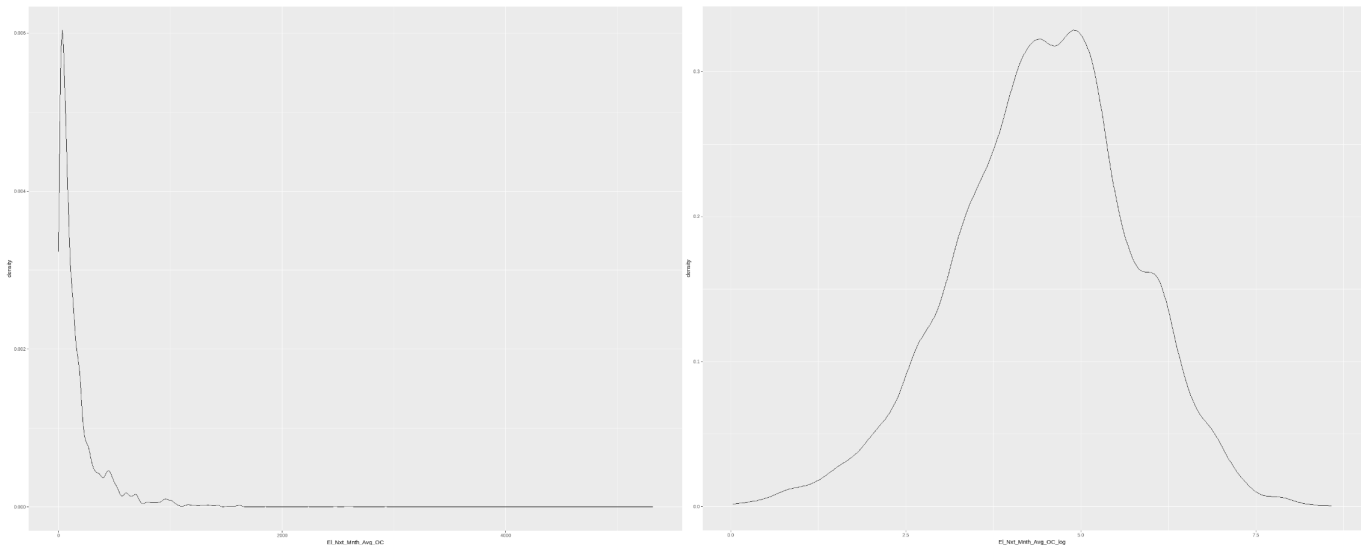


Figure 4 - Outcome plot (left) and log of outcome plot (right)

- Box plot was used to explore the relationships between the outcome and the categorical variables (see example for SIDn -Site ID name in figures 5.1 and 5.2) ([Section \(C.2.\)](#))

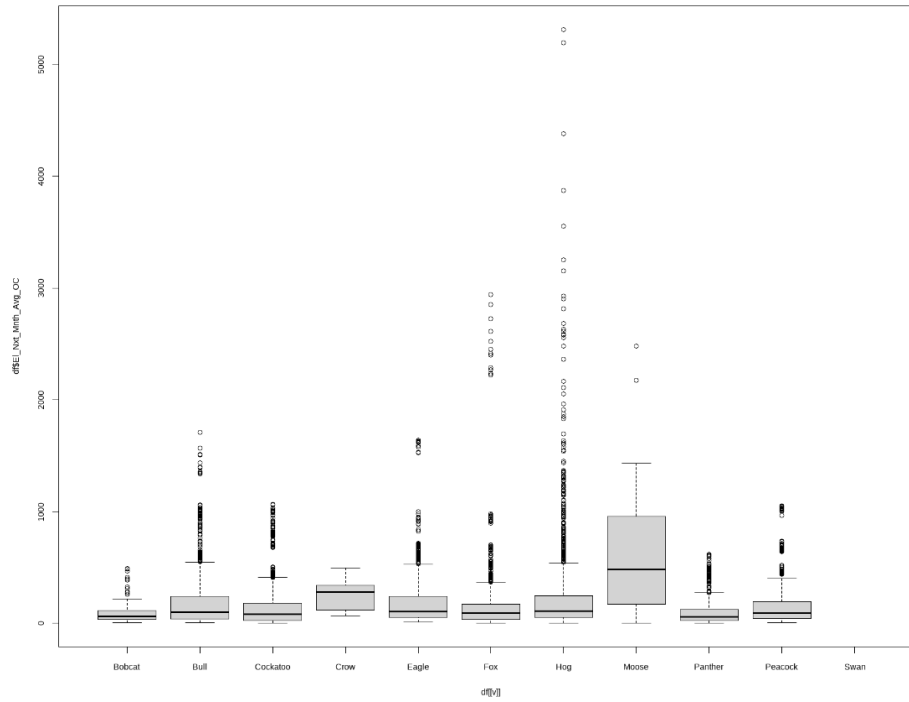


Figure 5.1 - Box Plot for outcome versus SIDn

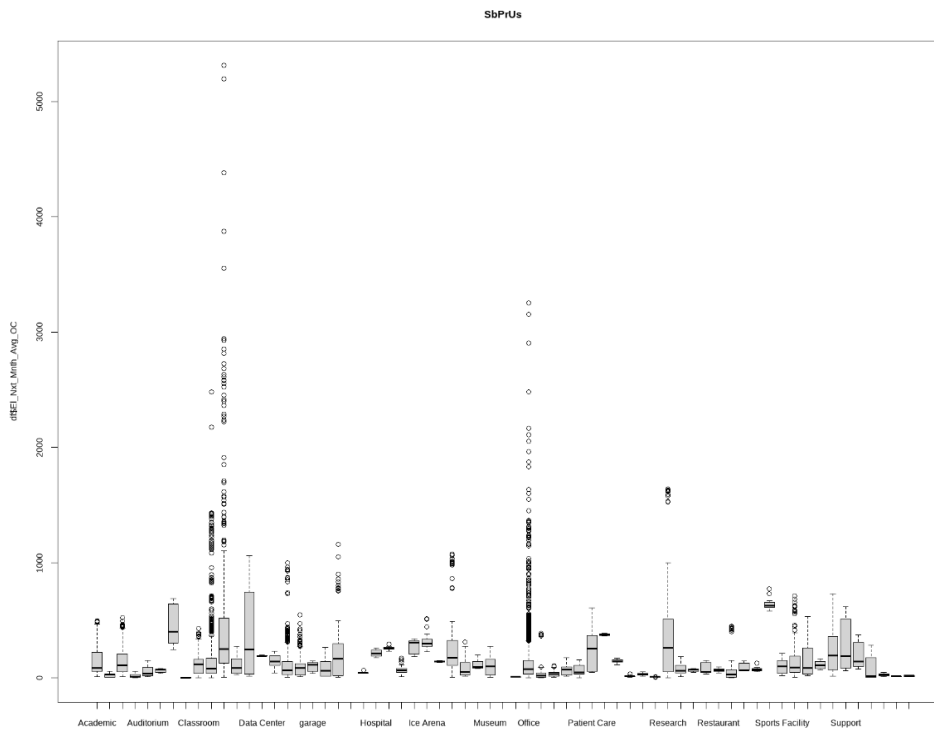


Figure 5.2 - Box Plot for outcome versus SbPrUs

D. Data cleansing - outliers and missing values

- The source data from the SQL server was uploaded to the data frame (DF_BGP)
Section (D) in file: [R Data cleansing -BGP-01062022](#)
- A factorised character variables data frame (DF_BGP_chFc) was created (*Section (D.1.2)*) for use of functions that don't work with characters and strings.
- A data frame (DF_BGP_nFc) with only numeric variables was created (*Section (D.1.3)*) for future use in functions that work with only numeric (like DBSCAN)
- Dr. Karpaty T. function for creating outliers matrix, using inter quartile rate was used for checking univariant outliers (in variables by themselves) (*Section (E.1.1.)*)
- Visualization of univariant outliers done with a heatmap of the outlier matrix. (*Section (E.1.2.)*) Some minor areas were detected. No major issues were visual.
- Box Plot visualization of univariant outliers was plotted for each variable using factorized data frame (DF_BGP_chFc) then was compared with the numeric data frame(DF_BGP_nFc). Many outliers were visually observed on multiple variables. No major difference was detected between the two data frames. (*Section (E.1.3.)*)
- A scatter plot outliers against index - BIDm was drawn for all variables using **scatter.smooth**.
Dew to very few unique values in Min_windDirction and Min_windSpeed weeks 1-4 (only 6-8 values) their relative scatters couldn't be plotted. As was visual in the heat map, also in the scatter plot, not a lot of outliers were visually observed in most variables. (*Section (E.1.3.)*)
- Multivariate outliers (pairs) were reviewed using the **dbscan** library. As dbscan uses only numeric data the numeric data frame (DF_BGP_nFc) was used. Another problem using dbscan didn't work with no full rows on the database. So the Clean Flat File data frame from the end of the file was used to create the outliers pair plot. (*Section (E.2.1, E.2.2)*)
- Partially scattered outliers pairs were plotted, 10 at a time, to get a reasonable view and not to overload the computer. (*Section (E.2.3.1)*)
- Scatter outliers pairs plots of the outcome against the Electricity consumption, Child water, and some wether parameters were drawn. (*Section (E.2.4)*)
On all these plots no visual anomalies were detected. (see some examples in figure 6)

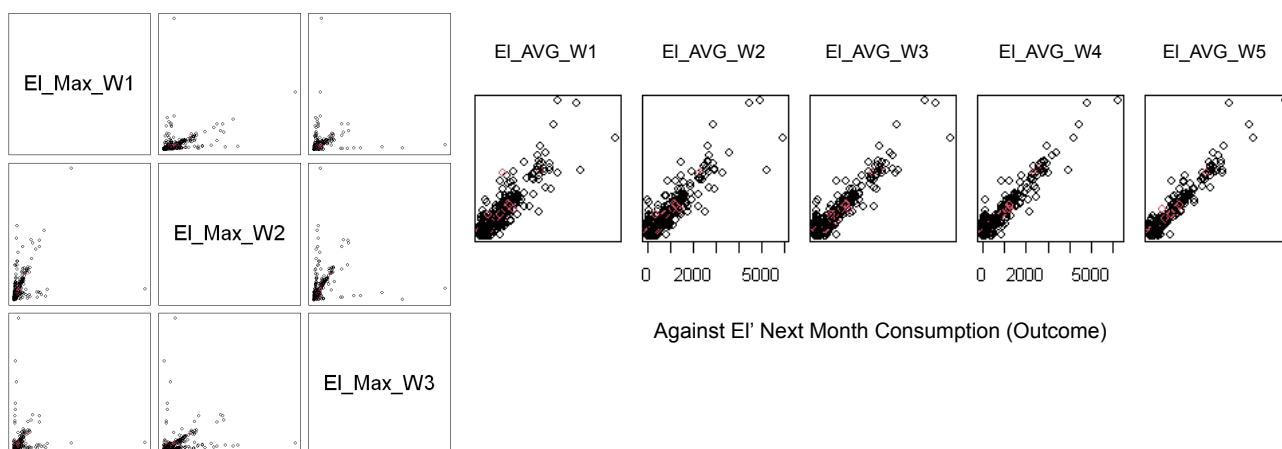


Figure 6 - Scatter plot of some outliers pairs

- The above procedures helped visualize for observing outliers generally. To make decisions, the analytical procedure was conducted using the "OutlierMatrix" code written by Dr. Karpaty Tomas and the Two-sample Kolmogorov-Smirnov test. The purpose is to check if there is a significant change in the distribution of the non-normal distributed variable with and without the outliers of another variable. If the D parameter in the test is less than 0.05 then there was no effect of outliers of one variable on the distribution of the other. The change has to be significant with a p-value less the 0.05 (see some results in table 5 below). ([Section \(E.3\)](#))

The "dis_chg" column is "+" for $D < 0.05$ in the "res4_0" matrix.

The Process was repeated to view possible differences with numeric data (nFc data frame) and factorized characters variables (chFc data frame) with and without NA's (Xna). No significant changes were observed.

A matrix: 38570 × 5 of type chr

var1	var2	OL_n	p-value.KS	dis_chg
sqm	lat	1908	0.0160918372305495	+
sqm	lng	1908	0.0160918372305495	+
sqm	EI_Max_W1	2006	2.79609668751846e-12	+
sqm	EI_Max_W2	2104	2.49200660107363e-12	+
sqm	EI_Max_W3	2104	3.06024094953727e-11	+
sqm	EI_Max_W4	2101	2.25315321955577e-11	+
sqm	EI_Max_W5	1999	5.45874456747697e-12	+
sqm	EI_Min_W1	2112	8.32667268468867e-15	+
sqm	EI_Min_W2	2198	3.77475828372553e-14	+
EI_Nxt_Mnth_Avg_OC	Std_windDirection_W5	247	1	-
EI_Nxt_Mnth_Avg_OC	Std_windSpeed_W1	105	1	-
EI_Nxt_Mnth_Avg_OC	Std_windSpeed_W2	0	1	-
EI_Nxt_Mnth_Avg_OC	Std_windSpeed_W3	303	1	-
EI_Nxt_Mnth_Avg_OC	Std_windSpeed_W4	258	0.999999999877489	-
EI_Nxt_Mnth_Avg_OC	Std_windSpeed_W5	34	1	-

Table 5 - Partial matrix output of Two-sample Kolmogorov-Smirnov test

- A similar test was conducted using the “**Cocor**” coed library to make sure the use was done correctly. The results didn’t show any significant differences between the two test codes. (Section (E.4))
- A visualization of the results of the test described above was plotted using density plots, where the red (0) plot marks the density plot of the source variable data, and the light blue (1) marks the distribution with the influence of the outliers. (Section (E.5))
(No influence can be seen in figure 7.1 below and plots with influence in figure 7.2 below).

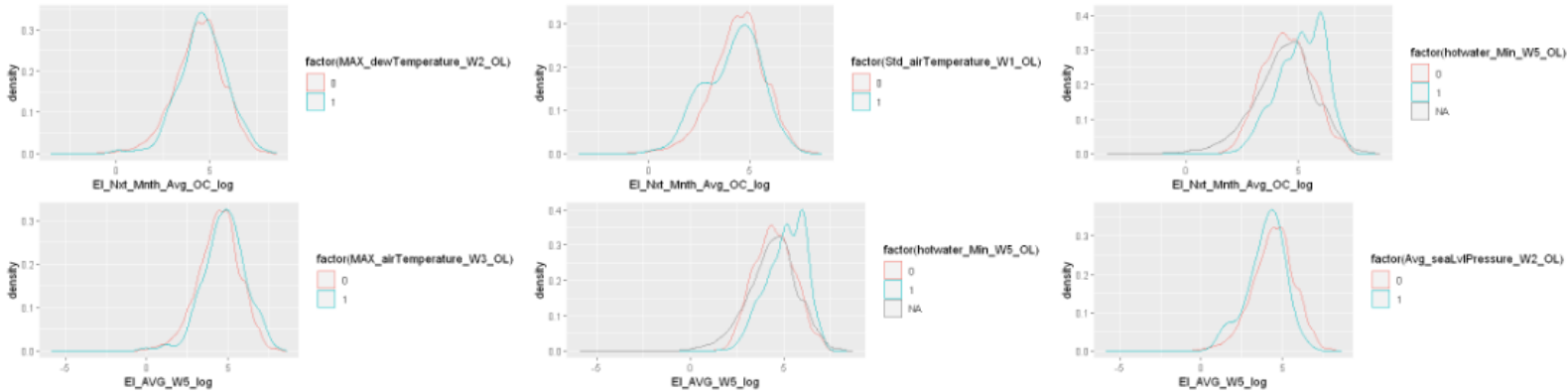


Figure 7.1 - Density plot of no influence of one variable outliers on another.

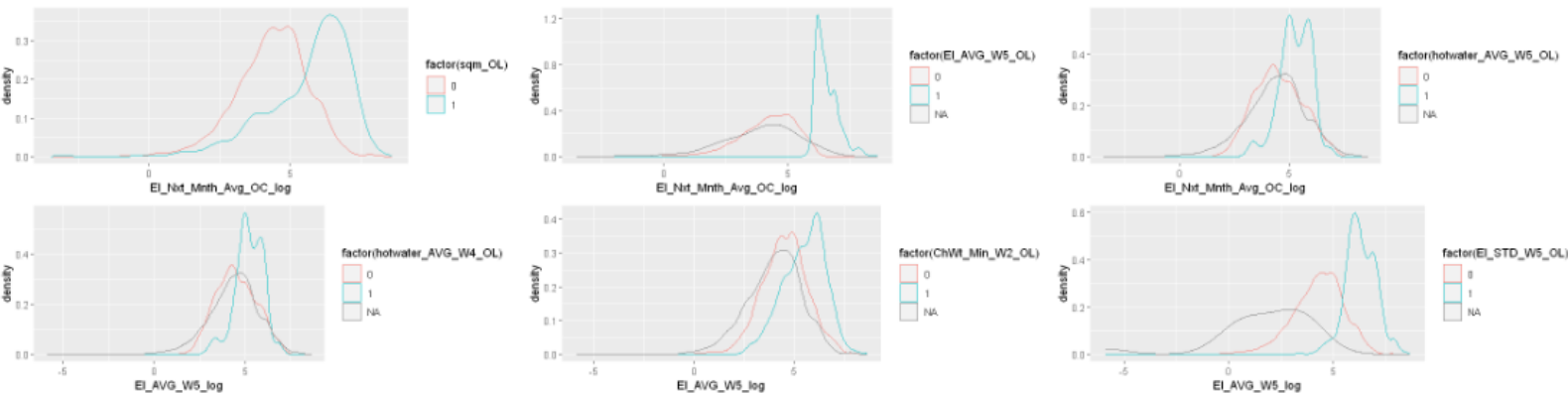


Figure 7.2 - Density plot that shows the influence of one variable outliers on another.

- Finally, to decide which outlier can be treated, a decision matrix was created, combining the correlation test (E.2) and distribution tests (E.3 and E.4). ([Section \(E.6\)](#))
(Partial table can be seen in Table 6.1 below)

var1	OL_n	p.value.KS	dis_chg	p.val.CO	correlation_changed	drop
<chr>	<int>	<dbl>	<chr>	<chr>	<chr>	<chr>
sqm	1908	0.01609184	+	0.000560286084479555	+	No
sqm	1908	0.01609184	+	3.97838557164576e-06	+	No
sqm	1908	0.01609184	+	0	+	No
sqm	1908	0.01609184	+	1.52960491162091e-07	+	No
sqm	1908	0.01609184	+	0.0731857116803039	-	Yes
sqm	1908	0.01609184	+	0.158099484100165	-	Yes
sqm	1908	0.01609184	+	0.610151584467267	-	Yes
sqm	1908	0.01609184	+	0.110873892294057	-	Yes
sqm	1908	0.01609184	+	0.324741340076994	-	Yes

Table 6.1 - Partial output table of OL_dess: outliers decision matrix

- Filtering the decision matrix for treatment needed (“Yes”) and significance ($p\text{-val} < 0.05$) on both correlation and influence, yields an empty data frame determining that **there were no outliers that should be treated /delete** from the data. ([Section \(E.6.2\)](#))
- To be on the “safe side” a check of close to significant values was conducted (see table 6.2)

var1	OL_n	p.value.KS	dis_chg	p.val.CO	correlation_changed	drop
<chr>	<int>	<dbl>	<chr>	<chr>	<chr>	<chr>
hotwater_STD_W1	1908	3.330669e-16	+	0.0501367435853488	-	Yes
hotwater_STD_W1	1908	3.330669e-16	+	0.0501367435853488	-	Yes
hotwater_STD_W1	5414	2.691192e-06	+	0.0501367435853488	-	Yes
hotwater_STD_W1	5413	1.602152e-05	+	0.0501367435853488	-	Yes
hotwater_STD_W1	5469	3.676875e-06	+	0.0501367435853488	-	Yes

Table 6.2 - Outliers decision matrix with p-values max close to 0.05

- Missing values treatment was conducted using Dr. Karpaty Tomas's second part code "missingnMatrix" function. The function marks 1 in place of NA in the source matrix and 0 where there is value. ([Section \(F.1\)](#))
- A summary of the rate of missing values for the different types of the data frames (nFc, chFc) was conducted ([Section \(F.2\)](#))
- A visualization of missing values was needed to generally evaluate the status of the missing values. A heat map (vis_miss) was used in parts as the full data didn't work. ([Section \(F.3\)](#))
- First, missing values in rows were treated. Using missingMatrix function code a percentage of missing values in each row in the data was calculated. ([Section \(F.4.1\)](#))
- It was decided arbitrarily that rows with more than 40% missing values will be deleted. That resulted in 1001 rows from 10272 rows - 9%. ([Section \(F.4.2\)](#))
- As the Gas and Hot Water variables have more than 96% missing values, a check of the same process was conducted on a temporary data frame without those columns. No significant difference in the rest of the variables was detected. ([Section \(F.4.3.\)](#))
- To decide which variables/columns with missing values to treat and how the missingness rate matrix was presented for the different data frames. ([Section \(F.5.1.\)](#))
- A density plot of the numeric variables, with and without the missing values was conducted to visualize the influence of the missing values on their distributions (Some example results are in figure 8.1 - no influence, 8.2 with influence). ([Section \(F.5.2.\)](#))

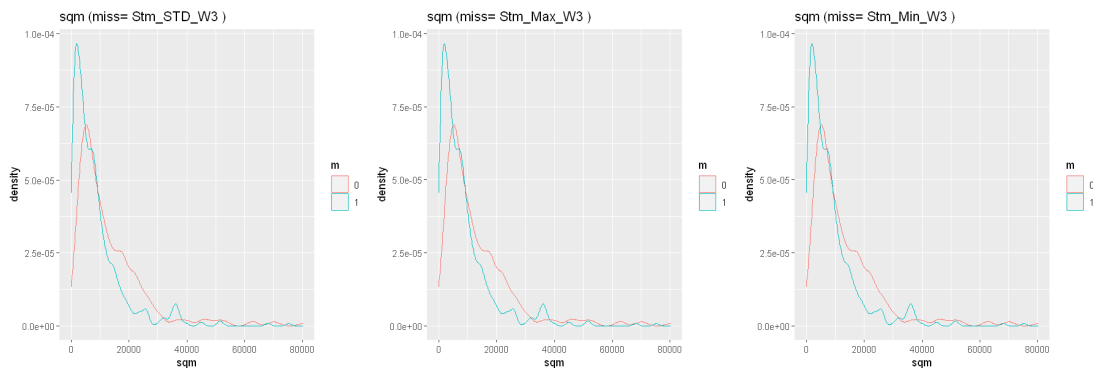


Figure 8.1 - Density plot of no influence of one variable missing values on another.

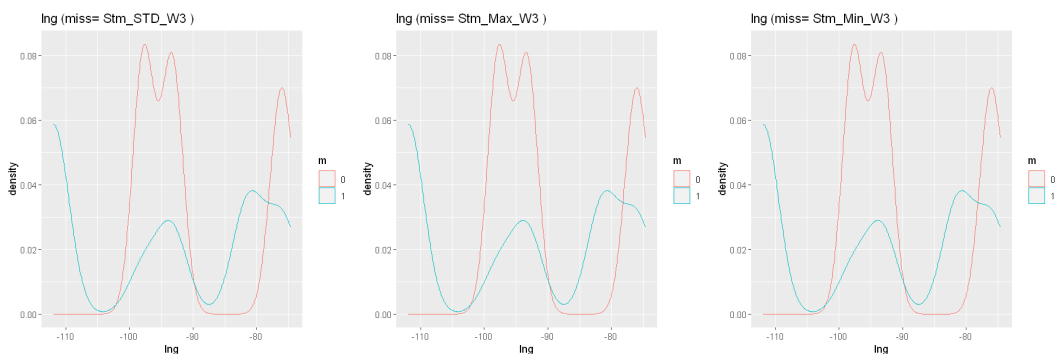


Figure 8.2 - Density plot with the influence of one variable missing values on another.

- Analytically analyzing the missing values' influence of one variable on another was conducted using the missingMatrix function code on the original data frame. ([Section \(F.6.\)](#))
- A table with a summation of the missing matrix and calculation of the significance of the influence for each variable was presented (partial table 7.1 below). ([Section \(F.6.2\)](#))

A matrix: 21721 × 5 of type chr

var	missing	p.value	missing_cnt	distribution_change
sqm	gas_Max_W2	1	0	-
lat	gas_Max_W2	1	1595	-
lng	gas_Max_W2	1	1595	-
EI_Max_W1	gas_Max_W2	1	349	-
EI_Max_W2	gas_Max_W2	1	412	-
EI_Max_W3	gas_Max_W2	1	407	-
EI_Max_W4	gas_Max_W2	1	380	-
EI_Max_W5	gas_Max_W2	1	383	-
EI_Min_W1	gas_Max_W2	1	349	-
EI_Min_W2	gas_Max_W2	1	412	-

Table 7.1 - Analytic analysis of the significant influence of missing values in one variable on the distribution of another variable

- As **no significant influence** was detected it was assumed that the missingness is at list “**missing at random**” (mar) and **imputation is permitted**.
- To make sure of the result - a logistic regression (glm) was used (some results are presented in Table 7.2). The conclusion holds. ([Section \(F.6.3\)](#))

m	var	pvalue
<chr>	<chr>	<dbl>
gas_Max_W2	(Intercept)	0.9991629
gas_Max_W2	lat	1.0000000
gas_Max_W2	EI_Max_W1	1.0000000
gas_Max_W2	EI_Max_W2	1.0000000
gas_Max_W2	EI_Max_W3	1.0000000
gas_Max_W2	EI_Max_W4	1.0000000
gas_Max_W2	EI_Max_W5	1.0000000
Std_seaLvIPressure_W1	Stm_STD_W3	1.0000000
Std_seaLvIPressure_W1	Stm_STD_W5	1.0000000
Std_seaLvIPressure_W1	MAX_seaLvIPressure_W1	0.9989135
Std_seaLvIPressure_W1	EI_Nxt_Mnth_Avg_OC	1.0000000

Table 7.2 - Analytic analysis of missing values influence using Logistic Regression.

- The imputation process was conducted with VIM library code using the kNN model. As is commonly accepted - the imputation process was performed only on variables with less than 40 % missing values. Variables with more the 40% missing values were factorized to high, mid, low, and miss (missing) factors. ([Section \(F.7.\)](#))

As the process needs reference columns with as less as possible missing values, 8 variables with less than 4% missing values were selected as the reference columns (including the outcome variable).

After the process was successful, 17 variables that had 4-5% missing values before imputation was used to impute the 8 columns used as a reference in the first process to have a full table. ([Section \(F.7.3.4\)](#))

- As part of uniforming the factorization process a had the NAs in the character columns (PrUs and SbPrUs) were replaced with “miss”. ([Section \(F.7.4\)](#))
- All 60 variables/columns were factorized as described above. ([Section \(F.7.5.1\)](#))
- A test to check if the imputed and factorized data frame is clean from missing values by using the missingMatrix code process on the new data frame (result table 8). ([Section \(F.7.6.\)](#))

```
[[1]]
[1] var      na.count rate
<0 rows> (or 0-length row.names)

[[2]]
[1] "This dataset has 10272 (100%) complete rows. Original data has 10272 rows."

A data.frame: 0 × 3

  var na.count rate
<chr>    <dbl> <dbl>
```

Table 8 - Result of missing values test on the new full data frame

- A clean and full data frame was created ([DF BGP_CFF](#)) for the next stage. ([Section \(F.8.\)](#))

E. Feature engineering

E1. Feature generation and transformation

- The feature engineering process is presented in the [R_Feature_Eng_BGP_onehot_log_06062022](#) file. The clean and full data frame created in the EDA and data cleansing process was uploaded and basic procedures (such as summary factorization of character features) were applied. ([Section \(A.1. A.2.\)](#))
- To have only numeric data to improve model performance, an index ID column replaced the BIDn column. ([Section \(A.3\)](#))
- To avoid using any characters even in factorize columns to improve model performance one hot encoding was applied using the dummyVars model. ([Section \(A.4\)](#))
- The old factorized and character columns were dropped and a new data frame with only numbers and one hot encoding was generated ([Section \(A.4.4\)](#))
- As was deduced from the EDA; most of the numeric variables have very high values density in the low part and then drop down exponentially (fast). To expand and make clear the range and dispersion of the values, a log transformation on the relevant variables was applied. ([Section \(A.5\)](#))
The new “DF_BGP_OHC_Xnames_log” data frame grow from 209 (in DF_BGP) to 483 columns.
- To investigate the way one hot encoding and log transformation changed the behavior of the data “mechkar” code was applied once more as was done in the EDA. ([Section \(B\)](#))
- The “explorerData” function of “mechkar” creates cards with an overview of variable statistics figures 9.1- 9.6 a “before and after” sample cards are presented. ([Section \(B.1.2\)](#))

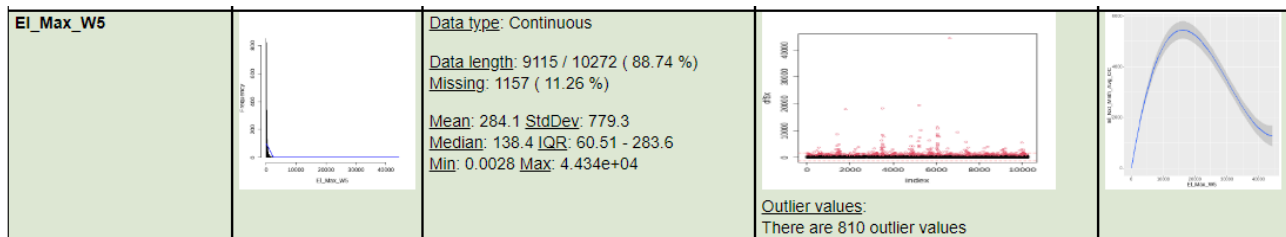


Figure 9.1 - EI_Max_W5 variable card before EDA Data Cleansing, and feature transformation.

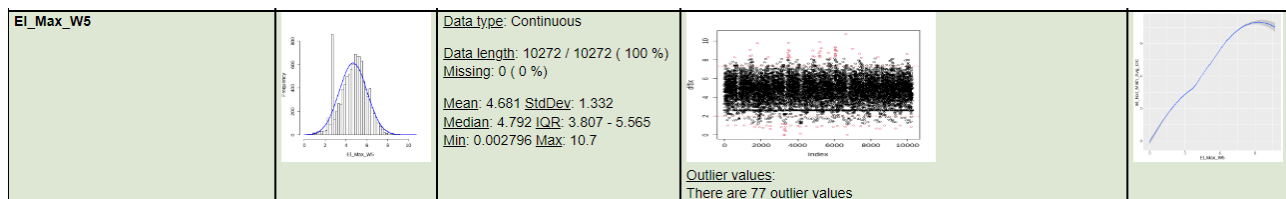


Figure 9.2 - EI_Max_W5 variable card after EDA Data Cleansing, and feature transformation.

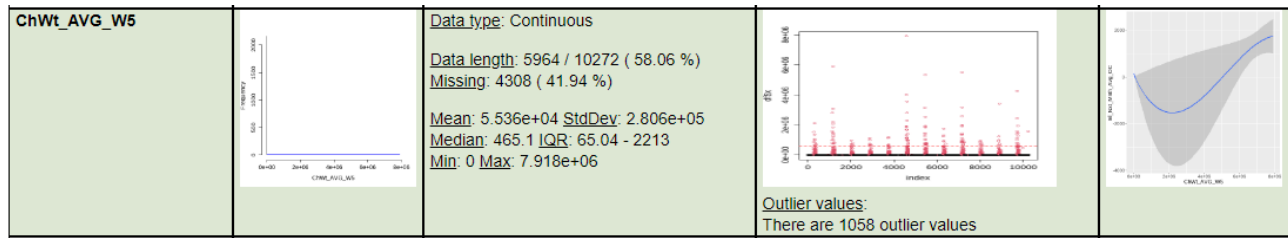


Figure 9.3 - ChWt_AVG_W5 variable card after EDA Data Cleansing, and feature transformation.

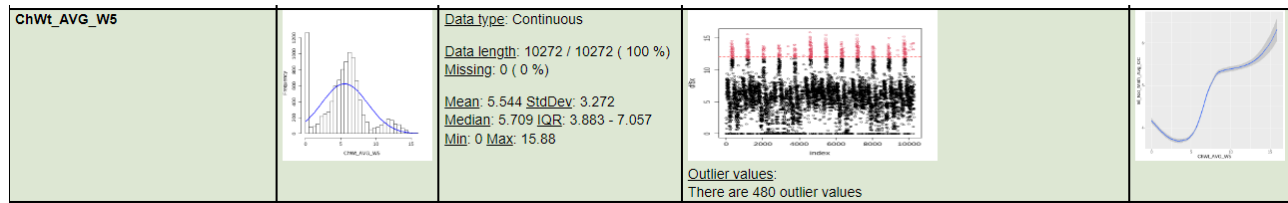


Figure 9.4 - ChWt_AVG_W5 variable card before EDA Data Cleansing, and feature transformation.

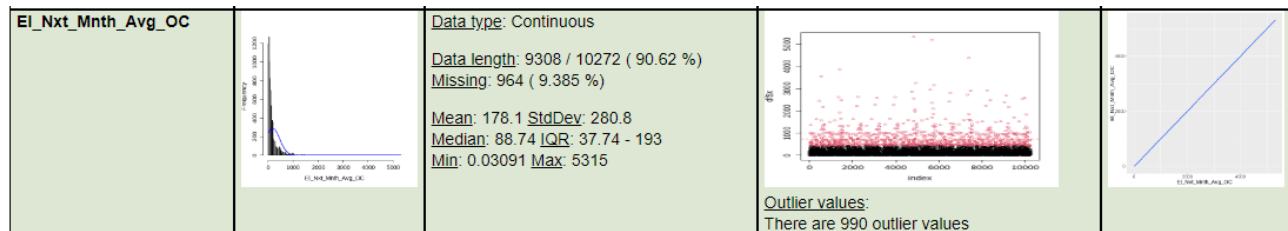


Figure 9.5 - Outcome variable card before EDA Data Cleansing, and feature transformation.

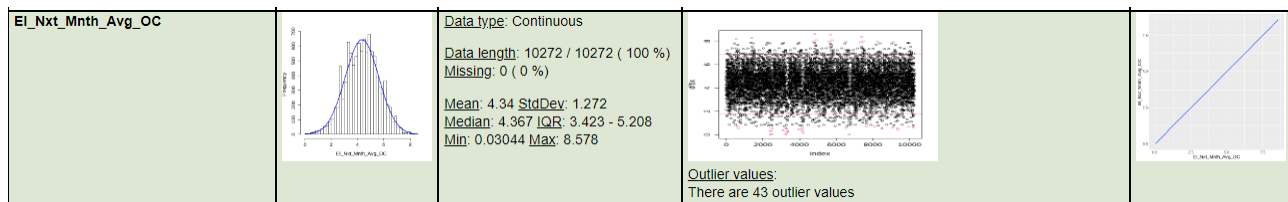


Figure 9.6 - Outcome variable card after EDA Data Cleansing, and feature transformation.

The plots on the right represent a general association between the variable and the outcome with a gray line of tolerance. It is clear from the figures presented here that the process applied bettered the variable's statistics and association with the outcome, das enabling better feature analysis and selection. The full report can be viewed in ["report06062022"](#) file

E2. Feature Selection

- The process of feature selection of a parametric (continuance) outcome and variables take into consideration, among others, the correlation rate and significance between the variable and the outcome. This correlation process was conducted on the new data frame (DF_BGP_OHC_Xnames_log or simply “df”) created after the feature transformation process. ([Section \(C\)](#))
- Library ‘Isr’ with correlate function was used (spearman method). After filtering correlations with more than 70% correlation as they “Tell the same story” and significant correlations (p-value less than 0.05) as we can’t be sure that these correlations are meaningful, the process reduced 483 variables to **230** variables. ([Section \(C.1\)](#))
- A parallel process using “Hmisc” library with a record function was applied to the same data frame. After filtering as described above, 483 variables were reduced to **84!**.([Section \(C.2\)](#))
- The feature engineering process described in the previous section resulted in 230 features in one test and 84 features in the other. Both tests used correlation to filter the relevant features. To continue the process of feature selection the filtered data was exported (csv file) and imported to a Python file:” Feature_Selection_Strategy_11062022” created by Dr. Tomas Karpaty with a unique voting strategy enabling the use of several models to look for the best relevant features to be used in the model training process. ([Section \(A\)](#))
- The **univariable analysis** was conducted, as mentioned, in the previous file. Now the uploaded data was used to input the results into the “voting table”. For each variable/feature that was filtered in the correlation tests, a 1 was marked in the voting table (each test a part). Otherwise, a 0 was marked. ([Section \(B\)](#))
- The **multivariable analysis** used “Lasso”, “Random Forest”, “Gradient Boosting”, and “SVM” models to select the best suitable features to be used. At the end of each running of the model, the results were marked in the voting table as in the univariable. ([Section \(C\)](#))
- The voting of the univariable and the multivariable test was then summarised (see table below) ([Section \(D\)](#))

	Variable	Univariable	Univariable_res	Lasso	RandomForest	GradientBoost	SVM	Sum
0	sqm	1	1	0	0	0	0	2
1	lat	1	1	0	0	0	0	2
2	lng	1	0	1	0	0	1	3
3	EI_Max_W1	0	0	1	0	0	1	2
4	EI_Max_W2	0	0	0	0	0	0	0
...
477	Stm_STD_W4.miss	0	1	0	1	0	0	2
478	Stm_STD_W5.high	0	1	0	1	1	1	4
479	Stm_STD_W5.low	0	1	0	0	0	0	1
480	Stm_STD_W5.mid	0	1	0	0	0	0	1
481	Stm_STD_W5.miss	0	1	0	1	0	0	2

482 rows × 8 columns

Table 9 - Variable Voting table with sum - to select suitable features

- The result of the voting process yield;
 - 0 votings for 208 features
 - 1 voting for 179 features
 - 2 votings for 75 features
 - 3 votings for 11 features
 - 4 votings for 8 features
 - 5 votings for 1 feature

Taking the 3, 4, and 5 votings into consideration - the Feature Selection voting process gives a **20** from 483 features that are most recommended for the Model process.

After considering scale versus quality, it was decided to use the recommendation of 20 features for the first run. If the Model process is not satisfying 2 votings will be added (for larger scale modeling) using 95 features.

- Finally, the twenty selected features were prepared and exported. ([Section \(E\)](#))

F. Data pre-processing test dev train split

- The data “ready to be used” in creating the model from the previous stage was uploaded to an R file([Dataset pre-processing Train Test Dev 13062022](#)) ([Section \(A\)](#))
- The “Train Dev Test split” of Dr. Tomas Karpaty's “Mechkar” library was used to ensure a carefully balanced split that correctly represents the data.([Section \(B\)](#))
The split parameters were chosen to 80% split and 2242 seed number.
- The result was: 8217 rows for the train (temp) and 2055 for the test.
Unfortunately, the balancing process didn't result in a well-balanced split (no matter what variations were applied, including omitting some features).
- The split for dev - train with the same parameters resulted in 6573 rows for train and 1644 rows for dev. ([Section \(C\)](#))
- A quick check with the “ranger” library resulted in the below report and plot (figure 10.1. and 10.2) ([Section \(D\)](#))

Ranger result

Call:
ranger(El_Nxt_Mnth_Avg_OC ~ ., data = temp)

```
Type: Regression
Number of trees: 500
Sample size: 8217
Number of independent variables: 20
Mtry: 4
Target node size: 5
Variable importance mode: none
Splitrule: variance
OOB prediction error (MSE): 0.5502961
R squared (OOB): 0.6589217
```

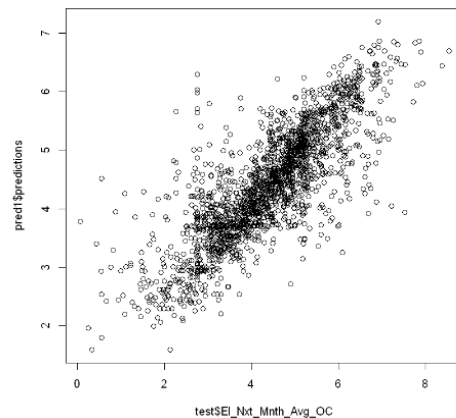


Figure10.1 - ranger report and plot evaluating the temp test splits

Ranger result

Call:
ranger(El_Nxt_Mnth_Avg_OC ~ ., data = train)

```
Type: Regression
Number of trees: 500
Sample size: 6573
Number of independent variables: 20
Mtry: 4
Target node size: 5
Variable importance mode: none
Splitrule: variance
OOB prediction error (MSE): 0.5807321
R squared (OOB): 0.6428985
```

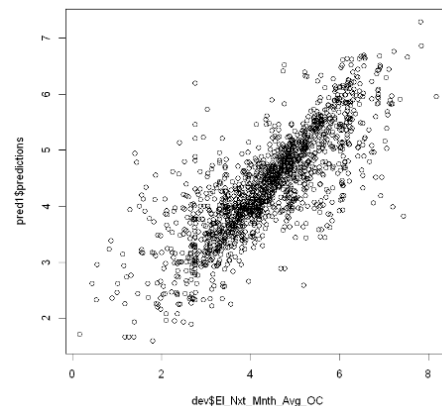


Figure10.2 - ranger report and plot evaluating the train dev splits

- The test train dev data frames were downloaded to CSV files respectively.

G. Models

The work done up to this stage started with generating nontime depended on features, EDA process to learn more deeply about the behavior of the data, data cleansing to check and clear outliers and imputation of missing values, generating more features and transforming it and Finlay selecting the most valuable features and splitting them to prepare for use in training and testing models.

This long process was done to make sure that the data quality going into the modeling process is the best that can be produced from the raw data received.

Parametric (continuous) outcome implies the use of regression models spatially after log transformation made it look close to being normally distributed.

G1. Selecting a suitable regression models

- The process of selecting the suitable regression model was done in the [“BGP_Regression_models_11062022”](#) file.

First, the Shapiro-Wilk normality test was used to make sure that the use of regression models is justified. As the test p-value is not accurate for a large number of rows, 500 random rows from the outcome were selected. ([Section \(A\)](#))

The result was: **Test Statistics: 0.996, P-value=0.333**. Meaning that as the p-value > 0.05, then we fail to reject the null hypothesis i.e. we assume the distribution of the outcome is normal/gaussian. Figures 11.1 and 11.2 below visualize the proximity to normal distribution.

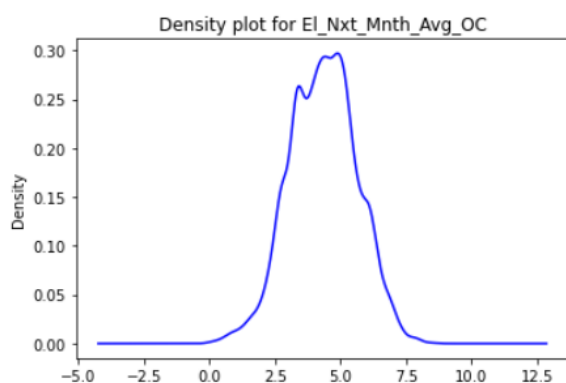


Figure 11.1 Density plot of the outcome

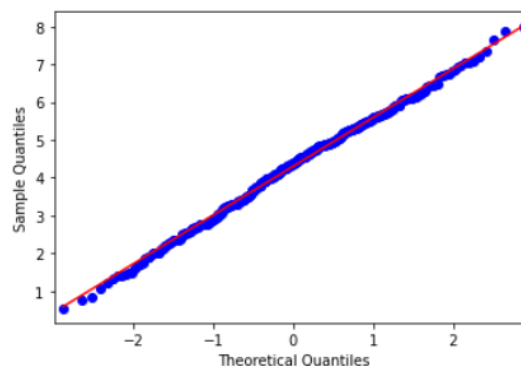


Figure 11.2 QQ plot theoretical / outcome

- The train data from the previous stage was uploaded and separated, variables to X and outcome to y. ([Section \(B\)](#))
- A metrics function was set to RMSLE using the NumPy function “metrics mean squared log error” ([Section \(C\)](#)). This metric was elected as the outcome values have 144 ratios (min/max), and as the source BGP article (in [Github](#)) stated the models' accuracy in the contest in RMSLE (2.9424 for the electricity short-term prediction), it was useful for comparing the results.
- The selection of the suitable model process run on six models:
 - Linear Regression
 - Decision Tree
 - Random Forest
 - Adaptive Boosting (AdaBoost)
 - Gradient Boosting Machine (GBM)
 - Support Vector Machine (SVM)

Using the RMSLE matrix to evaluate which is the best model to be used. ([Section \(D\)](#))

- The linear regression model is then used to determine a baseline, as it is a very basic model. The surprising result of **RMSLE=0.2373** was encouraging regarding the data quality and the preparation process. ([Section \(D.1\)](#))
- The scatterplot in figure 12 clears the overfitting suspicion.

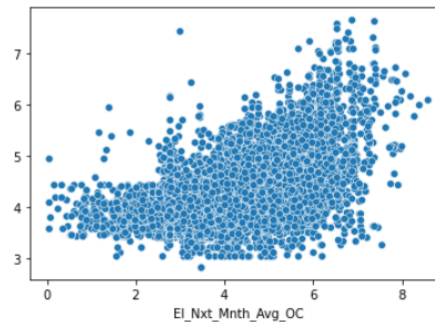


Figure 12 Scatter plot of the outcome in linear regression model prediction/sample

- Table 10 below shows the RMSLE result of the test. ([Section \(E\)](#))

	model	RMSLE
1	Decision Tree	4.135290e-17
2	RandomForest	6.871929e-02
4	GBM	1.892719e-01
3	ADABOOST	2.192662e-01
0	Linear Regression	2.373000e-01
5	SVM	2.655916e-01

Table 10: RMSLE models results

- From table 10, It is clear that the Decision Tree RMSLE result represents overfitting. The best (low) result is of the **Random Forest (RMSLE = 0.0687)** model as it is approximately 3.5 times smaller than our baseline, established by the Linear Regression model.

Figure 13 shows the scatter plot of the Random Forest model. Comparing it to figure 12 of the linear regression above, it is visually clear that the Random Forest works much better.

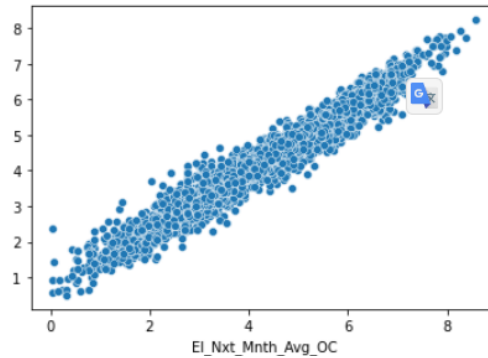


Figure 13 Scatter plot of the outcome in Random Fores model prediction/sample

The Random Forest Modle is thus selected for the full model training process.

G2. Training and testing the data on the Random Forest selected model

- A separate file “ RF_Tr_Ts_14062022 “ was used with Train Dev, and Test split data in the F.pre-processing stage to run the Random Forest model. First, the Train and Dev data were uploaded from the saved CSV files. ([Section \(A\)](#))
- The Random Forest model was then applied to the Train data and compared with the Dev outcome after predict process.([Section \(B\)](#))

RMSLE metric evaluation resulted in: **RMSLE = 0.174 !!!**

- The Test data was then uploaded and used with the prediction model to create the predicted outcome. ([Section \(C\)](#))

RMSLE metric evaluation resulted in: **RMSLE = 0.274 !!!**

G3. Hyperparameter Finetuning

To improve the model prediction a process of Finetuning of the hyperparameters (the model training parameters like a number of iterations or minimum sampling in leaf or split in trees related models) was applied in the file: "[BGP Finetuning 14062022](#)".

- Post feature selection data saved as CSV was uploaded and prepared for the process in a similar way to the modeling process (x variables, y outcome, train, and test...). ([Section \(A\)](#))
- Iteration on the Random forest model was used with randomly selected parameters after setting the range of parameters and number of iterations. ([Section \(B\)](#))
- The results of applying the fine-tuning process was: ([Section \(C\)](#))
 - **Mean Absolute Error of the model before applying finetuning 0.4782**
 - **Mean Absolute Error of the model after applying finetuning 0.4743**
 - **Improvement of 0.83% !!!**
- The best parameters elected by the random search process were:
 - 'n_estimators': 300,
 - 'min_samples_split': 3,
 - 'min_samples_leaf': 2,
 - 'max_features': 'auto',
 - 'max_depth': 20,
 - 'bootstrap': True

H. Deployment of your model

- Who will make the QA of the project?
The last stage (not applied at this stage) will be to get the 2017 data and test the model prediction with it.
- Who is the final user of the predictions?
The final user should be the energy in charge officer or budget manager of the facility (Educational campus, hospital, other public or private large complexes) that want to plan expenses on energy for the next month or energy efficiency act to reduce or carefully control the energy footprint of the facility.
- How the prediction will be presented to the final user?
The best way to present such data and predictions should be through visualization dashboards like one of Cornell [university's](#) dashboards that have the current and historical status of the energy consumption. The next month predicted energy consumption can be added to the dashboard.
- How will the final user be trained to use and interpret the prediction?
This is a preliminary data science project and was not processed to the end user at this stage. Eventually, the system should be able to automatically upload the data from such dashboard systems and create an automated prediction.
Farther more it may have some suggestions list to improve one or the other systems.
- On which platform the predictions will be deployed?
The platform will have to be one that can be synchronized with the backend of the dashboard systems, respectively to the facility.
- How frequently will the model be updated?
The model is based on gathering hourly data to create weekly averages resulting in monthly predictions based on one-year data gathering. Because of that and the fact that global warming and other events (like covid pandemic) it seems that the model should be updated at least once a year or more often if an unpredicted event happens.
- What will happen in cases where the model returns a null prediction (eg. incomplete data)?
If a "meter" (sensor type) is fulted and data is not gathered, the dashboard system should alert the end users and delay the model data gathering thus creating a gap in the next month prediction.
- Which models were used and which were selected for the final prediction.
Random Forest model was selected after testing Linear Regression, Decision Tree, Random Forest, Adaptive Boosting (AdaBoost), Gradient Boosting Machine (GBM), and Support Vector Machine (SVM).

- Which measurements were used to evaluate the prediction?
RMSLE was used to evaluate the prediction - yielding 0.274 at the test stage.
Mean Absolute Error was applied to evaluate the finetuning process with a 0.474 value/
- Which results in we got from those models?
After fine tuning the Random Forest were:
 - 'n_estimators': 300,
 - 'min_samples_split': 3,
 - 'min_samples_leaf': 2,
 - 'max_features': 'auto',
 - 'max_depth': 20,
 - 'bootstrap': True

Results

Here you will present the main results of the process. We will describe:

- The final amount of data used (total, train, test, etc)
The research started with 8 meters (sensor types) of login data every hour for 2016 and 2017 years from 1636 buildings positioned in 19 sites.
11 of the 19 sites were selected and 913 buildings were.
Only data from 2016 was used (to have 2017 as clean final test validation)
At the end of the preliminary process of transforming the hourly data to building data - 209 features were created and 10272 rows representing 856 buildings in 12 months each.
The feature generation process raised the feature number to 483, and the Feature selection process reduced it to 20 (22 with ID and Outcome).
Train data was set to 80% (8217 rows) and then divided once more to train dev 80% (6573 rows). The test was 20% (2055 rows) and Dev (test for train) was 20% of the train data (1644).
- The number of outliers and the way of treating them,
There were more the 1000 outliers in some features, but as their influence on the outcome and other features were not significantly observed ($p\text{-val} > 0.05$) they were not deleted.
- The number of missing values and the methods used for imputing them,
The data had a major problem with some features missing more the 90% values.
kNN imputation, factorizing, and one hot encoding were applied and deleted with this.
- The distribution of the data (timeframes)
Most of the meter data were distributed in a way that when log transformation was applied in the feature engineering stage (feature transformation) it resulted in close to normal distribution. The weather features were more generally normally distributed and did not need a transformation.

Conclusion

This project began as a final assignment in seven months of data science Bootcamp in Bar-Ilan university external courses led by Dr. Eran Shaham and facilitated by Dr. Tomas Karpaty.

I searched for physical data from sensors as opposed to data gathers from people systems (like seals or finance or HR). The Kaggle Building Genom Project was just what I was aiming at.

This was my first almost full data science project. As implied by the name, it was treated as a research project and not a coding-developing use project. This means I used the pre-prepared and learned codes to investigate and understand the data story. From that point of view, the use of the models and evaluations of the process stages helped produce a prediction based on understanding the project stages and development.

On Each stage, I had to struggle with relearning the course materials, re-adjusting the process we learned to the BGP data, and in some parts creating (with Dr. Karpaty's help) new ways to make the data fit the process required in the stages. This forced me to scrutinize the code lines and understand and recode some of them. To make meaning of the results and to understand why some processes didn't work and try (not always with success) to fix them.

As it is the first try, I learned and understand how much I don't know and don't yet understand then learn things that I understand. The overall process is slightly more clear to me, but it is also clear that I have a lot more to dig and check into the specifics of the project stages and overall understanding of the data behavior in the past and the predicted future. This is specific data and data in general. I understand more now how to touch the tip of the edge of the data and its complexity. I know now that I have some tools and procedures to investigate data, but more than that, I humbly understand how vast is the data world, data research tools, and processes, and I sow only the first small step of the infinite staircase.

Regarding the model prediction and viability, it is clear to me that this is a raw fetal experience and most likely there are much better and easier ways to predict electricity consumption for the next month. But for me, the path was more important than the result as I learned that huge amounts of sensor data can be gathered and processed to produce a prediction. I also learned the problems generated by sensor data and the impact of the data science process.

This final assignment of data science was a tedious, long and hard process, but a very challenging and insightful, interesting, invigorating, and humbling at the same time and I am very happy that I chose to go through the data science Bar Ilan university boot camp, and through this BGP assignment... and happier now that I completed it.

Thank you Dr. Tomas Karpaty.