Energy performance of buildings

Technical University of Denmark

Applied statistics and statistical software(02441)

January 21, 2020

Done by:

Stanley Chris Ardiente Frederiksen , s140425

Huda Ibrahim Muhudin, s153485

Reda Lahlou, s192431

GROUP 14

1 Summary

The aim of the project is to predict the worst insulated buildings, in order to enable Høje Taastrup municipality (HTK) to prioritize the renovation of these buildings first. The initial model is based on data provided by HTK and have been merged into a single data frame subsequently to data cleansing (i.e normalization, removing invalid and incomplete data). We went from an initial regression model based on the physics of heat loss to a model containing climatic explanatory variables and their second order interactions. Based on the slopes of the initial regression model, it was possible to identify the worst (14 buildings) as well as the best (11 insulated buildings) - i.e the ones with a slope significantly different from a reference slope. An analysis of contingency table showed that no building type was over-represented as the worst or best insulated building.

Finally, it is concluded that the physical model was too simple, as it did not take other variables into account that proved to have a significant impact on the consumption, explaining 6% more of the consumption variation.

Table of contents

2 Introduction

The Høje Taastrup municipality (HTK) wants to use statistical methods to identify buildings with poor energy performances, in order to restore and optimize them. Heat loss through a simple wall is given by the following equation:

$$Q_{heat} = U_A \cdot (T_{indoor} - T_{outdoor}) \tag{1}$$

Here, U_A represent a measure of the insulation in the building. Throughout this project the aim is to estimate the U_A and to understand how the energy consumption is affected by different climatic variables. Since its not possible for HTK to restore all the buildings the "worst" insulated buildings will be identified [1]

2.1 Description of Data

The data was collected over approximately 4 months (2018.09.01 to 2018.12.28) and consists of a total of 9794 real-world observations between 83 different public buildings in HTK. Each building is given a unique ID and has 118 observations of their total energy, equivalent to each day during the whole period. There are 10 explanatory variables or factors that determine total energy consumption.

We are also given a data set; *HTK_ building_ data_ share* that shows what each building is used for and also that each building has different heated areas. Since there exist differences between the heated areas, we assume that larger buildings have a higher consumption relative to the smaller buildings. This could lead to the false impression that larger buildings have poorer insulation during the statistical analysis. Therefore, the data requires a standardization that considers the relative size differences. Moreover, the fog and rain variables from the original data are not considered in this report.

Table 1 shows the data type categorized according to the output of the R function *str*. A short description of the data is also given and the range of the values for each of the individual variables. Even though dates are intervals and the IDs are themselves continuous, they are both considered as factorial for the sake of computation. Furthermore, the IDs are converted to a range from 1 to 83 corresponding to one of the IDs from the original data set to identify the buildings easier.

Figure 1 shows the scatter plots against each of the different explanatory variables/factors and also how they correlate with the dependent variable, the consumption. Collinearity exists between the two independent variables temperature and dew point. In addition, there seems to be a correlation between the consumption and the temperature. The consumption decreases when the outdoor temperature increases.

Variable name	**Data type**	**Description of the data**	**Unit**	**Range**
Date	Factorial	Date of the observation	N/A	2018.09.01-2018.12.28
ID	Factorial	ID of building	N/A	1:83
Consumption	Continuous	Energy consumption	[MWh]	0:8.703
Temp	Continuous	Outdoor temperature	[°C]	-1.9:18.615
Dew_{pt}	Continuous	Temperature where water vapor condenses into liquid water	[°C]	-3.650:15.125
Hum	Continuous	The concentration of water vapour present in air	[°C]	49:98.6
$wind_{spd}$	Continuous	Wind speed	$[\frac{meter}{hour}]$	3.84:42.27
dir	Factorial	Wind direction	[East,ENE,ESE,NE,NNE,NNW, North,NW,SE,South,SSE,SSW,SW, West,WNW, WSW]	1:16
vis	Continuous	Visibility	[meters]	1.965:50
pressure	Continuous	Pressure	[mPa]	986.48:1040.35
cond	Factorial	Condition of the weather	[clear, Fog, Light rain, Light snow, Mist, Mostly cloudy,Overcast, Patches of fog,Partly cloudy]	1:10

Table 1: Overview of Data

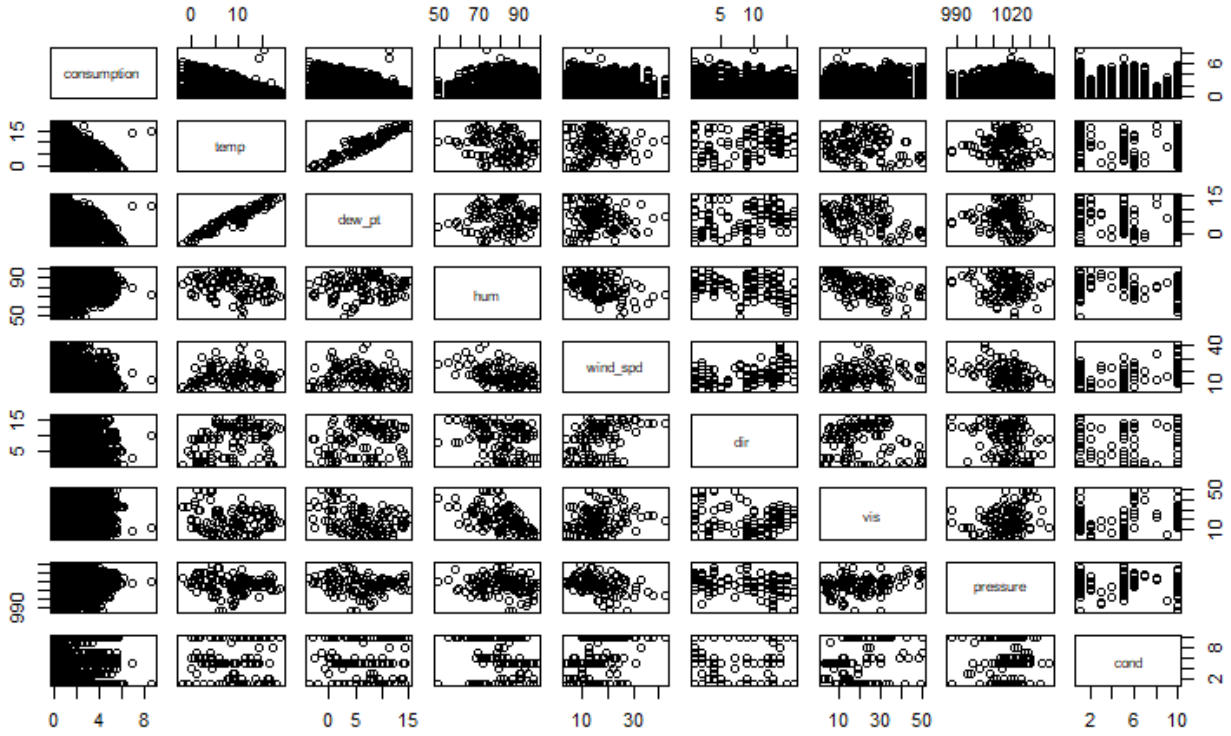


Figure 1: Plot of the dependent and independent variables against each other

The correlation between the temperature and dew point is shown in figure 2, which effects the formulation of our model as they cannot independently predict the consumption. This would be elaborated later in our report.

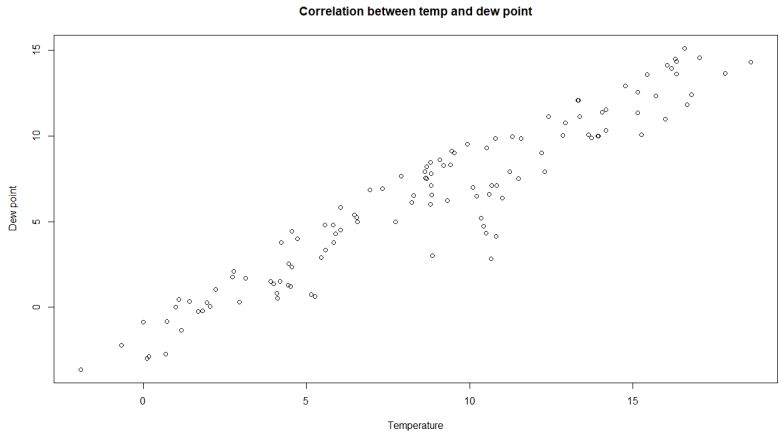


Figure 2: Collinearity, between temperature and dew point

We also show the boxplot in figure 3 of the independent variables to see their normality and the variation of the values. Some of the variables seem to be normally distributed (e.g. temp and dew_pt), while others are skewed to either left or right which can be visualized through the whiskers (e.g. if the lower whiskers is longer than the upper, the data is left-skewed). We can also detect some outliers in the wind speed and pressure variables.

Furthermore, the pressure is divided by 10 so the values fit within the range of the other independent variables for visualization purposes. The time series of the consumption is shown in figure 5a where each color corresponds to each building (ID).

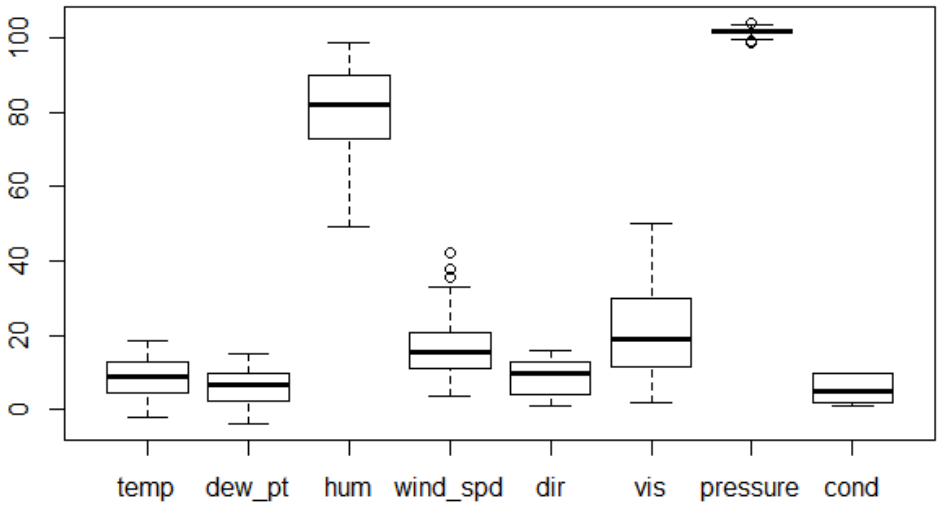


Figure 3: Boxplot of the independent variables

3 Statistical analysis

The process of our statistical analysis is shown in figure 4. When constructing a linear model the assumptions can be checked through diagnostic plotting. This process is iterated until the assumptions are satisfied, transformation may be needed.

For linear models, the following assumptions have to be fulfilled:

- The variance (σ^2) of the residuals are homogeneous and independent of location
- Residuals are normally distributed
- Linear dependency between the explanatory and response variables.

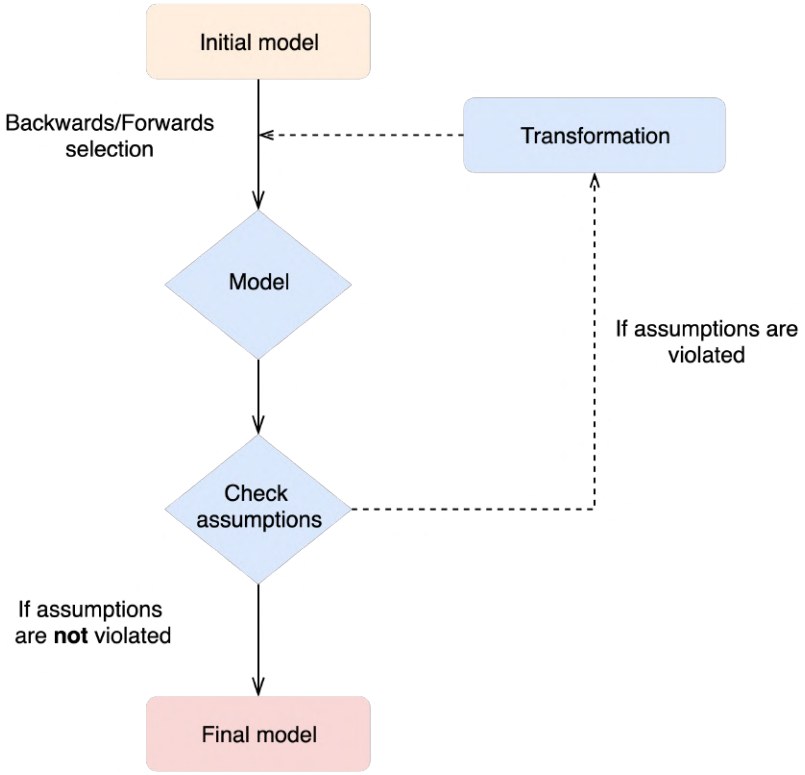


Figure 4: Flow chart showing the model selection process.

A significance level $\alpha = 0.05$ will be used throughout this report.

The statistical method depends on the given data (e.g. type of explanatory variables/factors). Based on our examination of the data set, the statistical model leads to a *General Linear Model*, ANCOVA, since we have a continuous dependent variable and at least one explanatory variable that is categorical and one continuous. The model is given by the following equation:

$$\begin{aligned} Y_{gi} &= \mu_g + \beta_g x_{gi} + \epsilon_{gi} \\ g &= 1, ..., k \\ i &= 1, ..., n \end{aligned} \tag{2}$$

where

- k denotes the number of levels
- n denotes the number of observations
- Y_{gi} is the consumption given by the g-level and i-observation of our data
- β_g (slope) corresponds to U_A in equation 1
- x_{gi} are the variables that we use in our model
- μ_g is the intercept and will only change according to the different levels
- ϵ_{gi} is the residuals and consists of the factors that are not explained by the model

Throughout the report, the R syntax will be used to introduce the models. For instance, if Y is the response variable, and x_1 and x_2 are the explanatory variables. Models will be presented in the following ways:

- lm(Y$\sim$ $x_1 + x_2$) is when the model contains additive explanatory variables to explain the impacts of x_1 and x_2 on Y.
- lm(Y$\sim$ $x_1 * x_2$) contains the additive terms but also includes the interaction effect between x_1 and x_2 to explain Y.

3.1 Data Cleansing

Two different data sources, *WUnderground* and *meterdata.zip* was given for the statistical modelling of 97 buildings in HKT. Some of the variables from the *WUnderground* contained N/A's and some were fixed to one value. These were removed prior to the merging with the *meterdata*. Furthermore, for each day the mean value of the continuous variables were calculated per building and the mode of the factor variables were also counted. Meters with less than 121 records were excluded from the data set.

To ensure comparability between the consumption (difference between daily readings) of each building, the daily reading were interpolated at 11:59 pm which is the reference point of each measurement. Finally, we compare our merged data with the data set *merged_data.csv* given during the lecture.

3.2 Normalization of data

As mentioned previously in the description of the data, larger buildings have a higher energy consumption and might aggravates the impression of the insulation in these kind of buildings. To account for this in our model, we perform a normalization of the data by the size of each building. We assume that the size of each building is proportional to its mean consumption. The day-to-day consumption for each of the building will be divided by the building's mean consumption over the whole period, and is calculated by the following equation:

$$Q^n_{heat} = \frac{Q_{heat}}{\bar{Q_{heat}}} = \frac{U_A}{\bar{Q_{heat}}}.(T_{indoor} - T_{outdoor}) = U^n_A.(T_{indoor} - T_{outdoor}) \quad (3)$$

where $\bar{Q}_{heat}$ denotes the mean consumption of the particular building for the whole period, Q^n_{heat} the normalized consumption of the building and U^n_A its normalized insulation coefficient.

From now on, any mention of "the slope" to the "insulation coefficient" in the report will refer to the **standardized insulation coefficient**, since it is the one that allows to perform a relevant comparison between the buildings.

In addition to the normalization of the data, unusual values which are not consistent with reality are removed. That is when the energy consumption is equal to zero since this cannot be the case unless the building are not in use. The normalized plot with all the observation can be seen in Fig. 13, found in the Appendix.

3.3 Construction of the model

To investigate the buildings with the highest energy consumption, we build a linear model step by step, starting with a simple model that only includes the temperature and ID. This is given by the following *R-syntax*:

$$Model = lm(consumption \sim temp \cdot ID) \tag{4}$$

We suppose that the buildings with the worst insulation have the highest slopes, U_A (denoted as β in equation 2) as these buildings are subject to a small temperature difference according to equation 1. The variable *temp* in equation 4 represents the difference between indoor and outdoor temperature. The indoor temperature is fixed to $21°C$ throughout the report.

3.3.1 Fitting the data

We built the linear model stated in the equation 4 before normalizing the data, and then we check for violations whether the model satisfies the the assumption of a linear regression model as mentioned in section 3. We then repeat the model fitting after normalization of the data.

3.3.2 Detection of the worst building

The worst buildings in terms of insulation are the ones with the highest slope (i.e. highest U_A value). An estimation of the slopes for each building can be found through the estimated coefficients in the R-output. In order to have a quantitative criteria on the detection of the highest slopes, we consider the building with the median slope and the corresponding 95% confidence interval as a reference. The other slopes (buildings) can therefore be considered as significantly different from the reference slope, if their 95% confidence interval does not overlap with the interval of the reference slope.

The confidence interval for the slope of each building was found by the following equation:

$$I = \beta \pm t_{(0.975,df)} \cdot se \tag{5}$$

where

1. I is the confidence interval (lower and upper boundaries).
2. β is the estimate of the slope.

3. $\pm t_{(0.975,df)}$ is the standard normal deviate corresponding to a significance level of 0.05 with *df degrees of freedom*
4. se is the standard error.

Calculating the standard error of each slopes requires to study how the uncertainty propagates, since the slopes are a linear combination of the parameters estimated by the linear model.

Let us define the matrix A as:

$$B = Ax \tag{6}$$

where B is the vector of the slopes, and x the vector of the estimated parameters.

If Σ_x if the covariance matrix of the estimated parameters, then the variance of B can be calculated as:

$$V[B] = A\Sigma_x A^T \tag{7}$$

The standard error for each slope can then be found by taking the square root of the diagonal element of the corresponding line of V[B].

3.3.3 Categorization of the buildings

Extra information on 77 buildings were given in an Excel file: *HTK building data share.xlsx*. Especially, the buildings are split among categories: schools, empty buildings, libraries etc.

In order to find if poorly and good insulated buildings are over-represented - or under-represented- in each of these categories, we build the corresponding contingency table. An analysis is done by performing a Fisher-test, which indicates if the distribution found is significantly different from a binomial distribution.

3.3.4 Final Model

The final selection was done through iteration with the built-in function in R, *step*, until the model contains the main effects and interactions that are significant. However, we decided to restrain our model only up to 2. order interactions for computational reasons. We set the minimum model to contain features as shown in equation (4) and with the normalized consumption values as the response. We defined the maximum model to include all the features that is shown in figure 3. Also, the dates are included but were converted to weekdays, and therefore, reduces to only 7 levels (i.e. monday to sunday). Equation 8 shows the formulation in R.

$$\begin{aligned} Maximal\ Model = lm(normalized \sim (temp + ID \\ + hum + cond + wind\_spd + dir + vis + pressure + weekday)^2) \end{aligned} \tag{8}$$

4 Results

4.1 Part 1: Data cleansing

An overview of the provided merged data is given in the following output from R-studio (table 2) along with our own merged data (table 3) for comparison. There are some minor differences, but overall the two merged data frames looks almost identical except the categorical variables (i.e *cond*, *fog* and *rain*). For the given merged data these variables were considered as continuous and not categorical. The original hour-to-hour table had these variables as categorical, but it seems that in the merged data that was given, their mean for each day was calculated instead of their mode. These variables were excluded in the further data analysis, since they do not make sense as continuous variables.

```
        date           ID            consumption          temp            dew_pt          hum
 2018-09-30:  89   Min.   : 4529799   Min.   :0.00000   Min.   :-1.900   Min.   :-3.650   Min.   :49.00
 2018-11-20:  86   1st Qu.: 6627217   1st Qu.:0.07465   1st Qu.: 4.556   1st Qu.: 2.333   1st Qu.:72.68
 2018-10-30:  85   Median :65005112   Median :0.15160   Median : 8.833   Median : 6.600   Median :82.07
 2018-11-06:  85   Mean   :37890916   Mean   :0.43617   Mean   : 8.724   Mean   : 6.309   Mean   :81.02
 2018-11-13:  85   3rd Qu.:69429582   3rd Qu.:0.33618   3rd Qu.:12.857   3rd Qu.:10.000   3rd Qu.:89.81
 2018-09-03:  84   Max.   :78673711   Max.   :8.70266   Max.   :18.615   Max.   :15.125   Max.   :98.61
 (Other)   :9280
    wind_spd          dir             vis             pressure                    cond           fog
 Min.   : 3.84   West   :1239   Min.   : 1.965   Min.   : 986.5   Scattered Clouds:3066   Min.   :0.00000
 1st Qu.:11.24   SW     :1079   1st Qu.:11.706   1st Qu.:1011.2   Mist            :2325   1st Qu.:0.00000
 Median :15.44   SE     :1076   Median :18.878   Median :1017.4   Clear           :1990   Median :0.00000
 Mean   :16.37   East   : 828   Mean   :20.541   Mean   :1016.5   Mostly Cloudy   : 998   Mean   :0.05203
 3rd Qu.:20.72   South  : 748   3rd Qu.:29.815   3rd Qu.:1022.5   Fog             : 580   3rd Qu.:0.00000
 Max.   :42.27   ESE    : 745   Max.   :50.000   Max.   :1040.4   Light Rain      : 252   Max.   :0.65000
                 (Other):4079                                     (Other)         : 583
      rain
 Min.   :0.00000
 1st Qu.:0.00000
 Median :0.05263
 Mean   :0.12691
 3rd Qu.:0.20000
 Max.   :0.61905
```

Table 2: R output of summary for the **given** merged data

```
        id          consumption             day             temp            dew_pt          hum
 2018-09-30:  89   Min.   :0.00000   2018-09-30:  89   Min.   :-1.800   Min.   :-3.600   Min.   :49.00
 2018-11-20:  86   1st Qu.:0.07458   2018-11-20:  86   1st Qu.: 4.579   1st Qu.: 2.190   1st Qu.:73.10
 2018-10-30:  85   Median :0.15153   2018-10-30:  85   Median : 8.905   Median : 6.833   Median :82.32
 2018-11-06:  85   Mean   :0.43395   2018-11-06:  85   Mean   : 8.733   Mean   : 6.317   Mean   :81.01
 2018-11-13:  85   3rd Qu.:0.33469   2018-11-13:  85   3rd Qu.:12.833   3rd Qu.: 9.947   3rd Qu.:89.30
 2018-09-03:  84   Max.   :8.00929   2018-09-03:  84   Max.   :18.500   Max.   :15.583   Max.   :98.39
 (Other)   :9280                     (Other)   :9280
    wind_spd          vis             pressure          dir                    cond           fog
 Min.   : 3.713   Min.   : 1.965   Min.   : 985.8   SE     :1078   Scattered Clouds:2982   0      :9464
 1st Qu.:11.305   1st Qu.:11.667   1st Qu.:1011.1   South  : 998   Mist            :2329   1      : 330
 Median :15.195   Median :17.645   Median :1017.4   SW     : 998   Clear           :1825   character(0):   0
 Mean   :16.363   Mean   :20.601   Mean   :1016.5   West   : 907   Mostly Cloudy   : 916
 3rd Qu.:20.786   3rd Qu.:29.571   3rd Qu.:1022.4   East   : 830   Fog             : 663
 Max.   :41.929   Max.   :50.000   Max.   :1040.2   ESE    : 830   Partly Cloudy   : 250
                                                    (Other):4153   (Other)         : 829
      rain            snow
 0      :9461   0      :9794
 1      : 333   character(0):   0
 character(0):   0
```

Table 3: R outout of summary for our **own** merged data

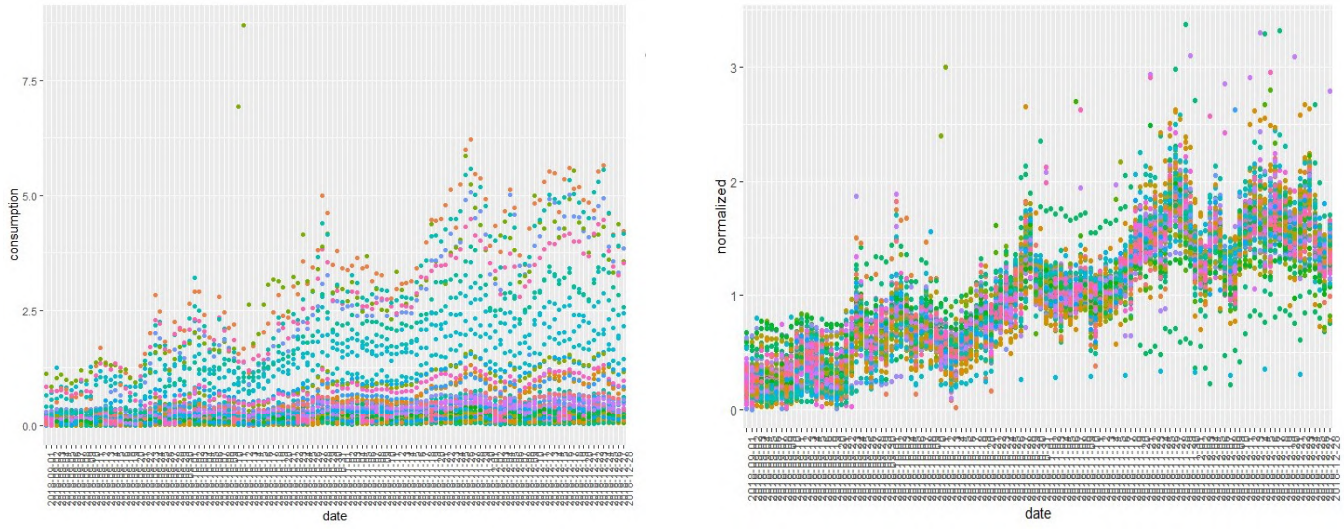
Another difference is the presence of the *snow* variable in table 3. During the data cleansing, variables showing fixed values were removed before calculating their means/modes for each day - and *snow* was not one of these fixed values at this point. After calculating its modes, it appears that no day showed more a majority of snowy hours - hence only values of 0 now appear in the final dataset.

Our final frame has 9794 rows, describing 83 different meters - just as the merged data that we were given.

4.2 Part 2: Data analysis

4.2.1 Normalization of the data

We normalized our data according to equation 3. Fig. 5a illustrates how the consumption evolves by time/date prior to normalization. We see an increasing variation of the consumption as the date proceeds towards winter. The different colors represents the different buildings/IDs.



(a) Before normalization

(b) Normalized Energy consumption

Figure 5: Time series of the energy consumption before (a) and after (b) normalization. The x-axis and the y-axis represents the dates and consumption, respectively

As seen in figure 5b, the normalization and removal of zeros results in a stabilized variance of the time series. The colors and their corresponding IDs can be seen in the appendix figure 13.

We can visually observe an increasing tendency of the consumption as the time approaches winter season in HTK. This is also expected since the consumption as shown in equation 1 can be described as a function of differences between the indoor and outdoor temperature (e.g. higher difference leads to a higher consumption).

The several peaks in figure 5b reflect the temperature peaks as seen in the appendix figure 12 which indicates a linear relationship between the consumption and temperature.

4.3 Fitting the model

The R-output given in table 4 shows the results from our linear model given in equation 4. The temperature difference, the mean of the consumption denoted new_{ID} and their interaction are all significant ($p<0.05$). This means that the temperature difference has a significant effect on the energy consumption for each building. The mean consumption for each building is significantly

different from one another and that the slope of the consumption over the temperature difference is significantly different between at least some of the buildings.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
I(21 - temp)	1	400.1485	400.1484505	15402.5701	< 2.2e-16
new_ID	82	4774.3620	58.2239271	2241.1635	< 2.2e-16
I(21 - temp):new_ID	82	1095.1057	13.3549471	514.0605	< 2.2e-16
Residuals	9628	250.1290	0.0259793	NA	NA

Table 4: Output of R-function for the ANOVA model **before normalization**

When inspecting the diagnostic plots of our first initial model (before normalizing the data), the variance of the residuals in the scale-location plot is not constant. Moreover, the cook's distance is present in the residuals vs. leverage plot. We also identify several observations that are considered as outliers; observation 3357 and 3282 appears in all the plots. However, observation 8946 doesn't appear in the Residual vs. leverage plot but instead observation 3440 appears which is not shown in the other plots.

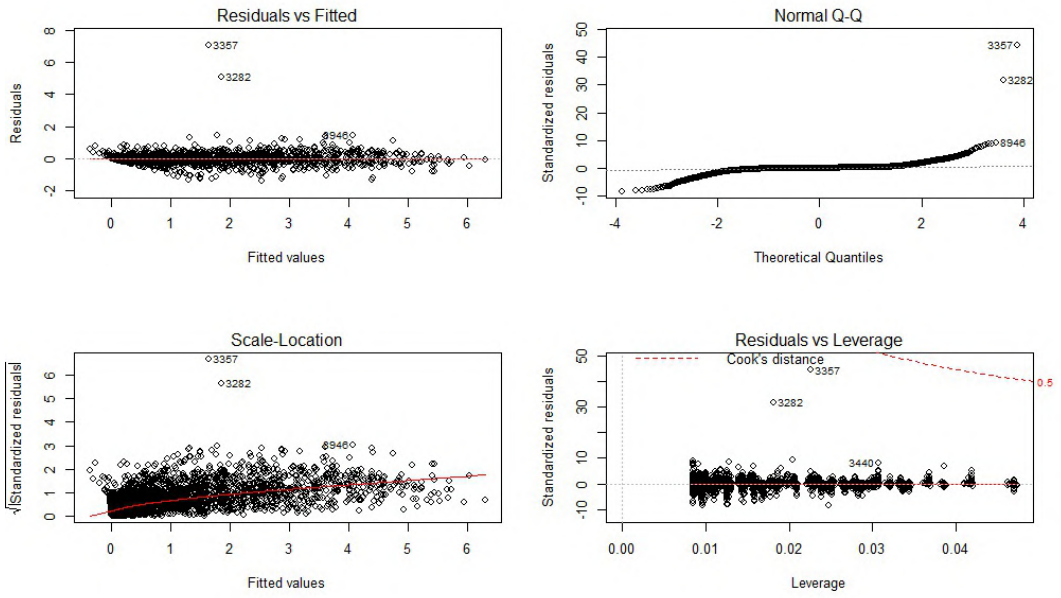


Figure 6: Diagnostic plots for linear model **before normalization**: Residual vs fitted plot, QQ-plot, scale-location plot and Cook's distance.

Table 5 illustrates the output of R after the normalization of the data. The temperature is a significant variable ($p<0.05$). However, buildings (IDs) are no longer significant ($p>0.05$) since they are normalized relative to their size. The ID's is only significant when interacting with the temperature. The model explains approx. 83.2% of the consumption variance.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
I(21 - temp)	1	2074.396116	2074.3961164	45135.182393	< 2.2e-16
ID	82	3.987328	0.0486259	1.058015	0.339
I(21 - temp):ID	82	97.110648	1.1842762	25.767751	< 2.2e-16
Residuals	9576	440.109382	0.0459596	NA	NA

Table 5: Output of R-function for the ANOVA model **after normalization**

Fig. 7 illustrates the diagnostic plots after normalization. The variance of the residuals are more constant compared to the one observed in Fig. 6. The variance illustrated in scale-location is more constant and we can barely see the cooks distance. It can therefore be concluded that normalization of the data and the exclusion of extreme observations, consumption = 0, is closer in fulfilling the model assumptions.

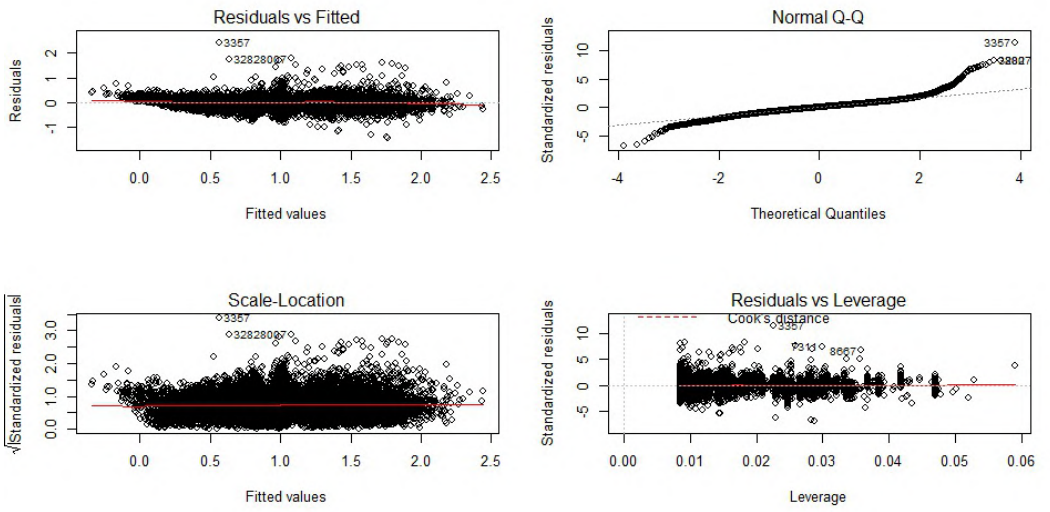


Figure 7: Diagnostic plots for linear model **after normalization**: Residual vs fitted plot, QQ-plot, scale-location plot and Cook's distance.

Lastly, we fit the data after the normalization of the data and also excluded the observations with zero consumption. The diagnostics can be found in the Appendix Fig. 14. In contrast to the data that includes zero consumption, the lower tail in the QQ-plot seems more smooth and closer to be in line, indicating a relative closer normal distribution of the residuals. Furthermore, the Residuals vs. fitted plot seems to have a narrower variance of the residuals and indicates a better fit of the data. However, the detected outliers are the same in both cases.

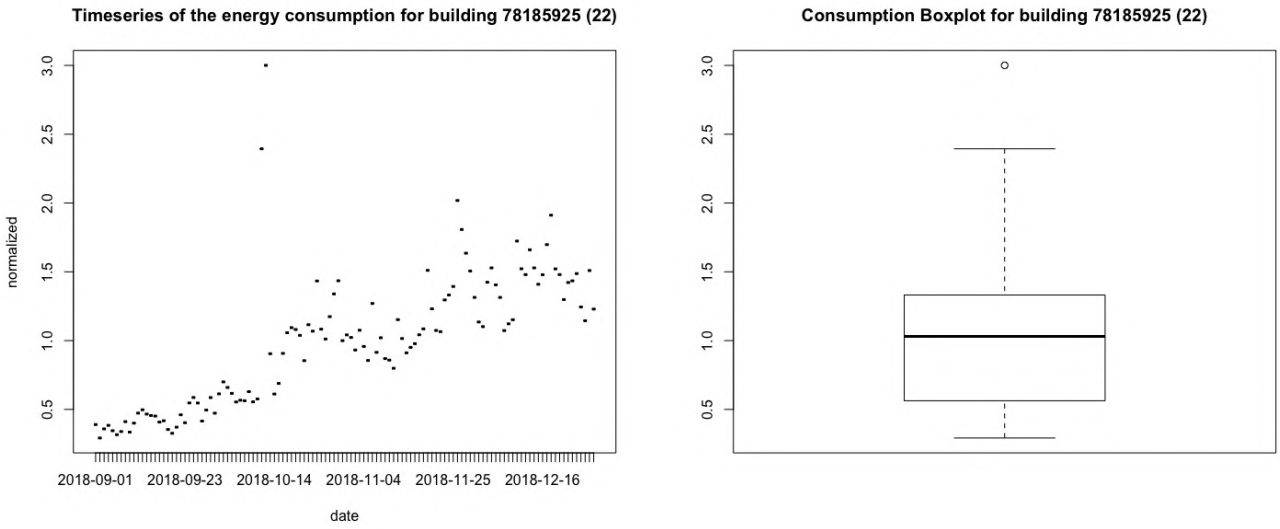
4.3.1 Outliers

After inspecting the diagnostic plots given in figure 7 we found one outlier, observation 3357, that belongs to building with ID 78185925. Fig 8a shows the consumption of building ID: 78185925, where it is obvious that the observation is high. It is, although, not possible to tell if this observation is due to an intentional excessive energy consumption or because of measurement failure. However, a possible explanation can also be found by expanding our model to include other explanatory factors. Table 6 shows the observation number, date, ID (new ID), the consumption (normalized) and temperature.

	date	ID	consumption	temp	new_ID	normalized
	<fctr>	<int>	<dbl>	<dbl>	<fctr>	<dbl>
3357	2018-10-12	78185925	8.702658	15.15	22	3.000493

Table 6: Output from R for the outliers 3357 and 3282 **After normalization and exclusion of zero consumption**

Figure 8a shows the time series of building ID: 78185925 and figure 8b shows the boxplot of the consumption from the same building. Based on the boxplot, we only consider one outlier, which is observation 3357 and not 3382, although both observations seems to be outliers when looking on the time series plot in figure 8a.



(a) Time series of building ID: 78185925

(b) Boxplot of building ID: 78185925

Figure 8: Visualization of building ID: 78185925 with consumption as a function of time (a) and the distribution of the consumption shown by a boxplot (b). Observation 3357 is detected as an outlier.

4.4 Finding the buildings

Fig. 9 shows a histogram of the slopes for each buildings to check if the they are normally distributed. The x-axis contains the slope values and the y-axis shows how frequent they occur. The red graph is a reference for comparison of our histogram and the shape of a normally distributed data set. However, the slopes do not exactly follow a normal distribution and looks more like a bimodal distribution with two peaks. Moreover, we observe slopes close to zero and also many individual slopes above 0.12. We therefore choose to use the median as the reference slope from which the worst and best buildings would be determined.

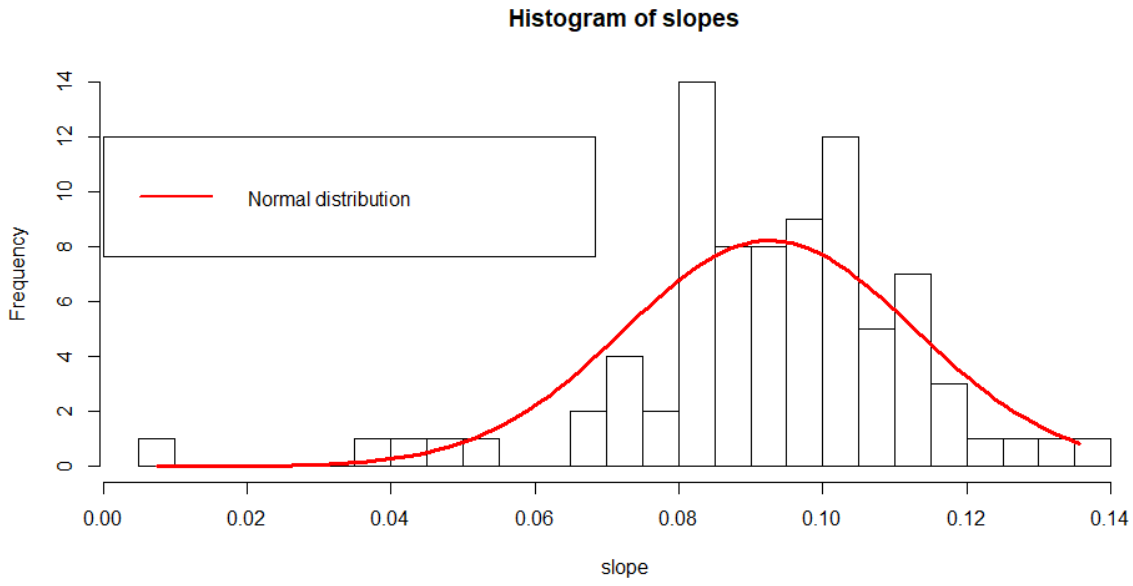


Figure 9: Histogram over the slopes with normal distribution graph as reference

The boxplot of the slopes are plotted in figure 10. The x-axis are the building ID's and the y-axis are the slope values. Each of the boxes represents one building ID with their slope values marked with the white horizontal line in the center of the boxes. The upper and lower edges of the boxes are the upper and lower confidence intervals of the individual slopes, calculated from equation 5. The black horizontal lines are the confidence interval of the median building (ID: 67). The upper black line represents the upper confidence and likewise, the lower black horizontal line represents the lower confidence boundary. One building (red box, ID: 32) seems to have a very low slope, which corresponds to the one in the farthest left side of the histogram in figure 9. The ID's in figure 10 are not shown in a chronological order, but we have ordered the sequence from the lowest to the highest slopes for visualization purposes.

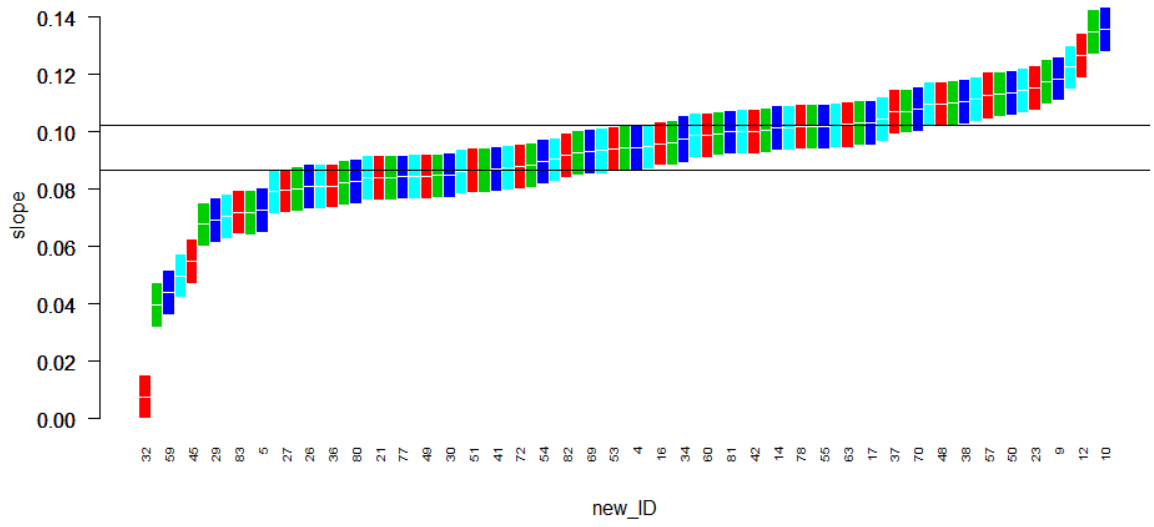


Figure 10: Slopes against building ID's with 95 % confidence interval. The interval is used for selection of the worst buildings

By comparing each slopes, it is possible to determine which building that diverge from the median building and the corresponding confidence intervals. We consider all the boxes (slope and confidence interval of each building) which are not overlapping within the confidence interval of the median building are either the worst or best buildings in terms of insulation. We found 14 buildings above the the upper confidence interval of the median ID (worst buildings) and 11 buildings were found below the lower confidence bound (best buildings). These total 25 buildings will then be further evaluated.

Table 7 recaps the worst and best buildings.

Worst insulation	**Best insulation**
6940321	4529799
65118812	6567326
6681894	78185925
6618580	6790785
5325295	6842603
5093913	69999051
7072231	69478883
78673711	4839509
69861509	5140250
69429582	6627258
69469107	4866195
6392057	
69585544	
7072241	

Table 7: Buildings with an insulation coefficient that is significantly different from the median one

4.5 Categorization of the buildings

The extra-information given for 77 specific buildings allows us to group these buildings in different categories. First, the 24 different categories are merged into 3 big categories: *institutions*, *buildings where people live* and *other buildings*.

Table 8 is the contingency table showing how many buildings of each categories are also found among the worst, best and normally insulated buildings found previously in the study (cf 7).

	Normal	**Best**	**Worst**
Institution	35	8	10
Living	12	2	3
Other	6	1	0

Table 8: Contingency tables displaying the relation between the categories of buildings and their insulation performance

The Fisher-test performed on this data indicates a p-value of 0.8987. We are therefore unable to reject the null hypothesis, which means that this distribution is not significantly different from a binomial distribution. Particularly, there is no under- nor over-representation of badly insulated buildings in any of the 3 categories defined. Figure 9 gives a visual representation of this analysis through the built-in R function *mosaicplot*.

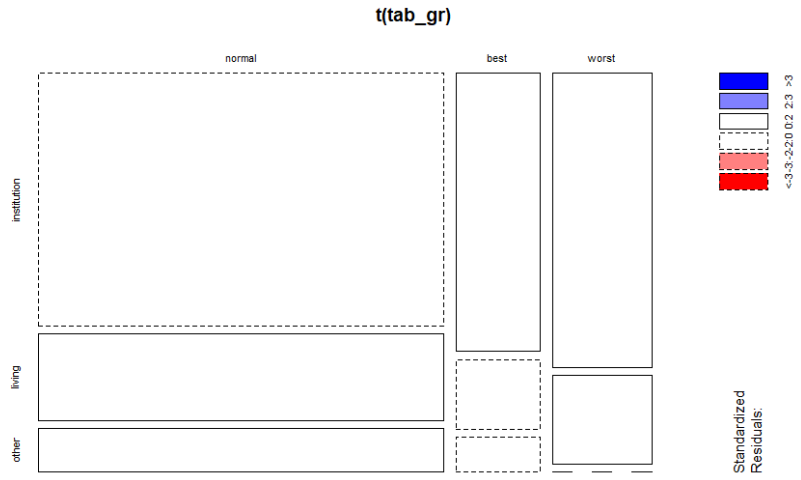


Table 9: Mosaic plot displaying the relation between the categories of buildings and their insulation performance

4.6 Final Model

Table 10 shows the ANCOVA test for our final model selection. All main effects are significant. However, only four of the 2. order interactions were significant. These are the temperature that interacts with pressure, condition and wind_spd. Moreover, the direction interacting with weekdays and condition are also significant. We can see from the summary of the model, that our final model explains approx. 89% of the consumption variance. That is 6% more of the variance, compared to the simple model with only IDs and temperature.

Anova Table (Type II tests)

Response: normalized

	Sum Sq	Df	F value	Pr(>F)	
I(21 - temp)	248.186	1	8139.9517	< 2.2e-16	***
ID	4.199	82	1.6795	0.0001261	***
weekday	17.508	6	95.7021	< 2.2e-16	***
dir	14.680	15	32.0986	< 2.2e-16	***
cond	4.283	9	15.6087	< 2.2e-16	***
pressure	1.360	1	44.5888	2.567e-11	***
wind_spd	0.631	1	20.7110	5.407e-06	***
I(21 - temp):ID	97.175	82	38.8674	< 2.2e-16	***
weekday:dir	26.751	31	28.3021	< 2.2e-16	***
I(21 - temp):weekday	4.669	6	25.5218	< 2.2e-16	***
dir:cond	13.845	12	37.8414	< 2.2e-16	***
I(21 - temp):dir	4.117	7	19.2899	< 2.2e-16	***
Residuals	288.495	9462			

Table 10: ANCOVA test of the final model

The model diagnostics is shown in figure 11. In comparison with the simple model residuals as seen in figure 7, the Final model is very similar in terms of constant variance and an S-shaped QQ-plot, and the same outlier was detected, namely observation 3357.

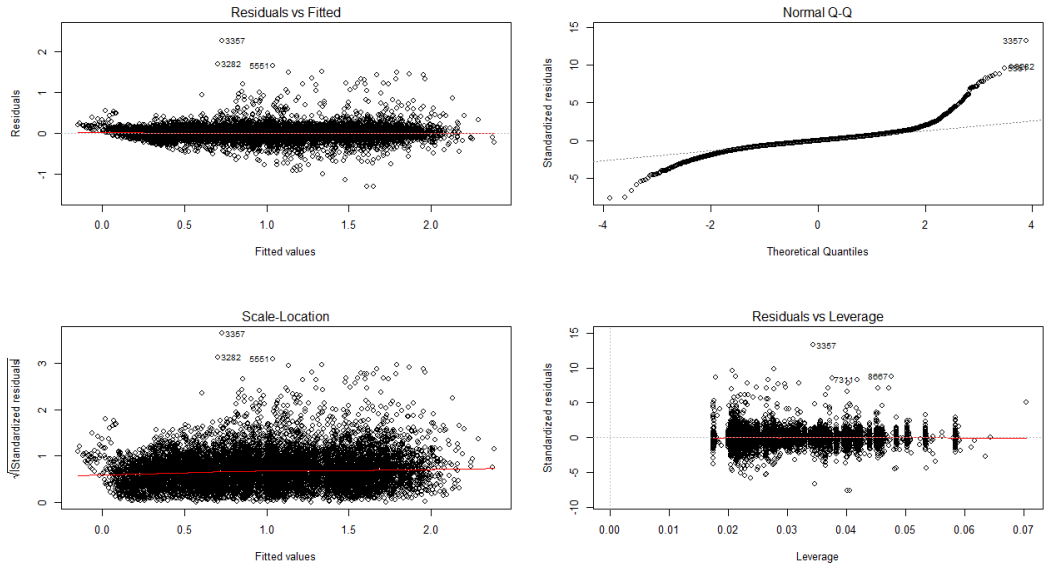


Figure 11: Caption

We compared the simple and final model as shown in table 11. It shows that the final model is significantly different from the simple model ($p<0.05$).

```
Analysis of Variance Table

Model 1: normalized ~ I(21 - temp) * ID
Model 2: normalized ~ I(21 - temp) + ID + weekday + dir + cond + pressure +
    wind_spd + I(21 - temp):ID + weekday:dir + I(21 - temp):weekday +
    dir:cond + I(21 - temp):dir
  Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
1   9576 440.11
2   9462 288.50 114    151.61 43.619 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 11: Model comparison between the simple and final model

5 Conclusion

The aim of the project was to estimate the insulation, U_A, in order to identify the buildings with the highest consumption (i.e worst insulated buildings). Furthermore, we wanted to predict the effect of the climatic variables. The final model consists of all the selected explanatory variables from table 1 and their second order interaction. As expected, we found that the temperature difference has a significant effect on the energy consumption of the buildings, and that there is a significant difference between some of the slopes found for each building. Upon interpreting the model, we found 14 buildings with significant high slopes compared to the median slope, making them the worst insulated buildings. In addition 11 buildings with significantly low slopes were also identified (see Table 7). Moreover, an analysis of a contingency table showed that none of the building types were over-represented as either the worst or the best insulated buildings.

On the basis of our analysis we can conclude that the physical model we based our main analysis on was too simple. It did not account for other climatic variables, which proved to be significant - the final model explains 6% more of the relationship between the consumption and the temperature.

Further analysis could be done by estimating more parameters than only the insulation coefficient - for example a coefficient linking the consumption to the wind speed. Additionally, some assumptions could be discussed more thoroughly. For example, the indoor temperature was fixed to 21°C , which is probably not realistic - especially since we are considering various types of buildings.

References

[1] Andreas Baum. *Applied Statistics and Statistical Software (02441). Case 2: HTK Case: Energy performance of buildings.*

6 Appendix

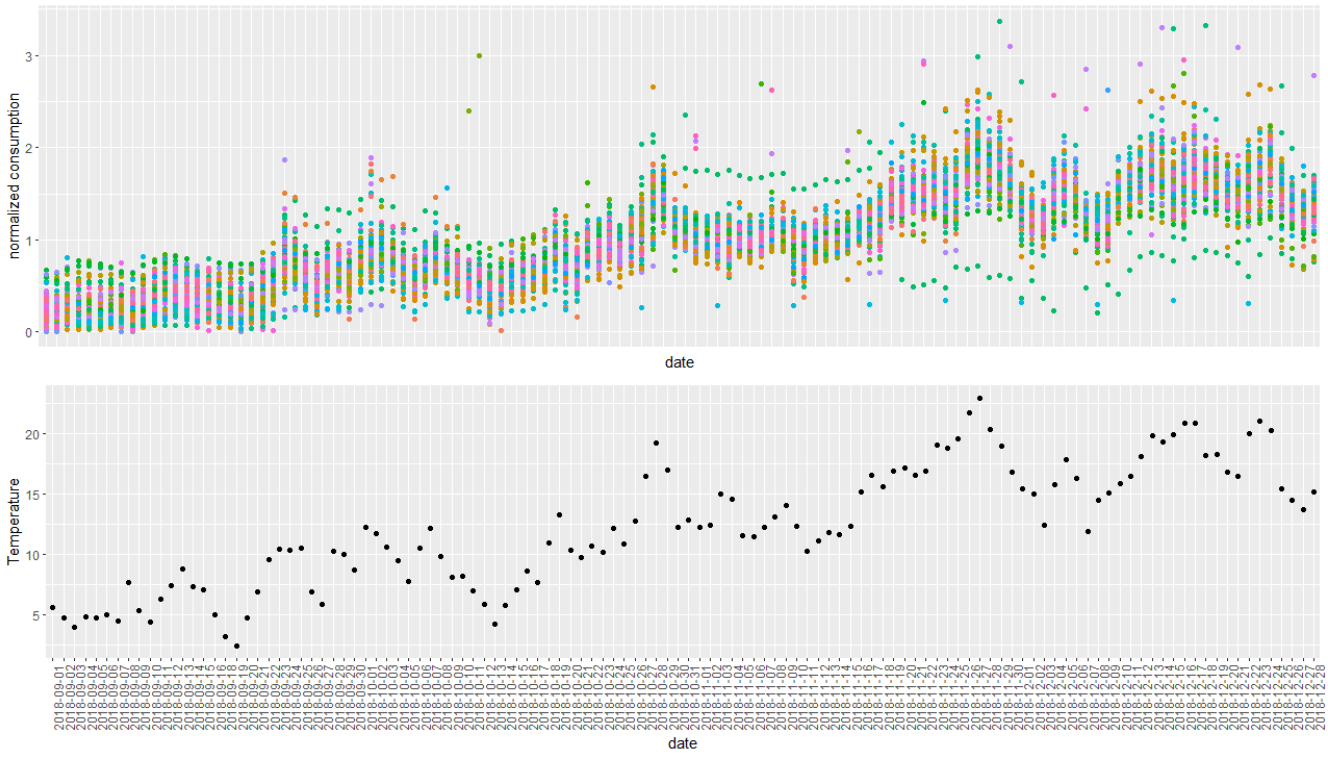


Figure 12: Consumption and Temperature

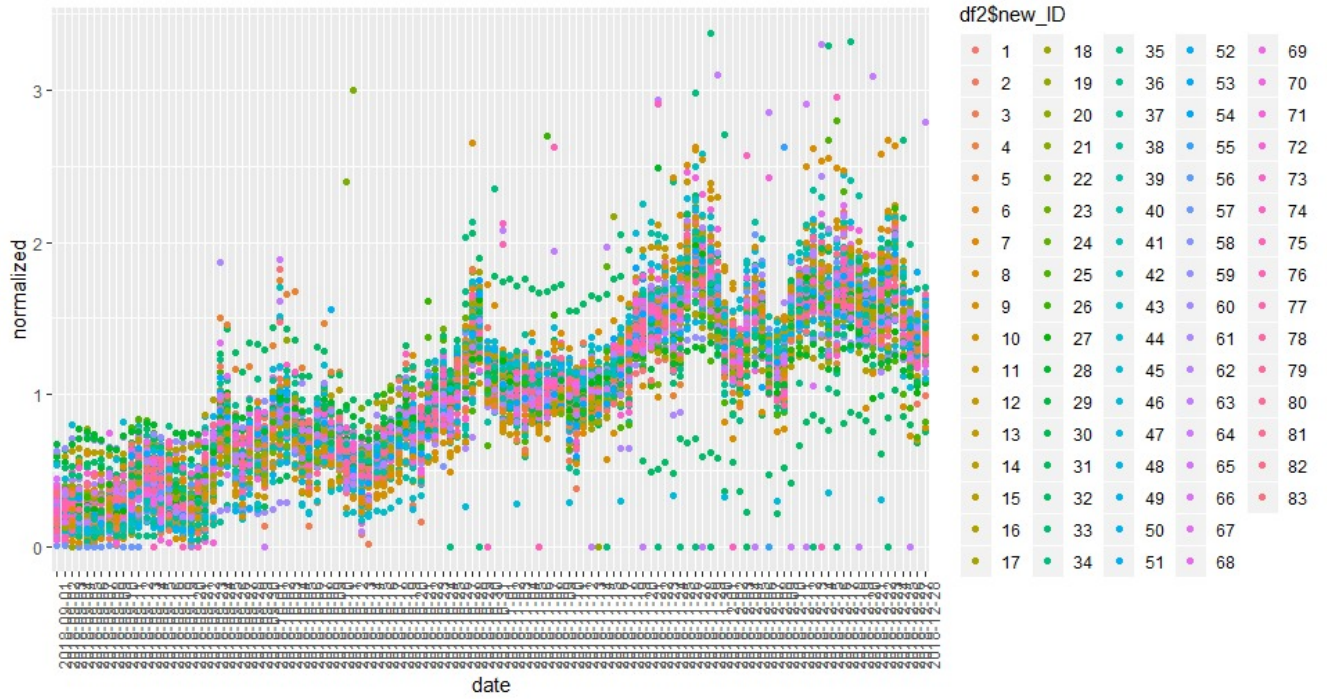


Figure 13: Normalized data

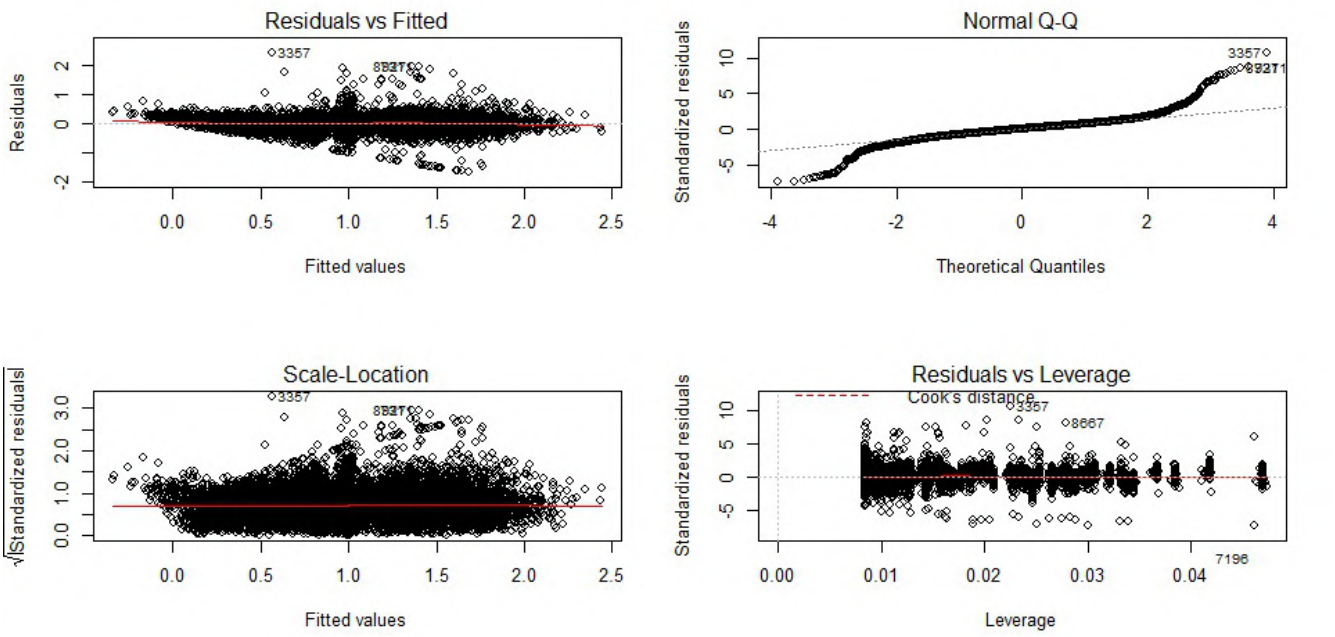


Figure 14: Diagnostic plots for linear model **after normalization and with observations equal to zero**: Residual vs fitted plot, QQ-plot, scale-location plot and Cook's distance

6.1 R script

R Notebook

```
library(car)
```

```
## Loading required package: carData
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.0
```

```
library(plyr)
```

##WUnderground data

```
load("WUndergroundHourly.RData")
str(WG)
```

```
## 'data.frame':    2184 obs. of  21 variables:
##  $ date         : POSIXct, format: "2018-09-01 00:00:00" "2018-09-01 01:00:00" ...
##  $ temp         : num  14 14 15 14 15 14 15 15 16 16 ...
##  $ dew_pt       : num  13 13 14 14 14 14 14 14 14 14 ...
##  $ hum          : num  90 95 96 92 93 93 91 90 85 81 ...
##  $ wind_spd     : num  11.1 7.4 13 9.3 9.3 11.1 9.3 7.4 13 14.8 ...
##  $ wind_gust    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ dir          : chr  "NNW" "WNW" "NW" "NNW" ...
##  $ vis          : num  45 40 10 40 40 30 35 30 40 35 ...
##  $ pressure     : num  1021 1021 1022 1022 1022 ...
##  $ wind_chill   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ heat_index   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ precip       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ precip_rate  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ precip_total: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ cond         : chr  "" "Scattered Clouds" "Rain" "Clear" ...
##  $ fog          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ rain         : num  0 0 1 0 1 0 0 0 0 0 ...
##  $ snow         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ hail         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ thunder      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ tornado      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
#Remove columns with only NA
df <- WG[, colSums(is.na(WG)) != nrow(WG)]
```

```
#Checking which columns have a variance different from 0
app <- apply(df,MARGIN = 2,function(x) var(x,na.rm = TRUE) != 0)
```

```
## Warning in var(x, na.rm = TRUE): NAs introduits lors de la conversion
## automatique
```

```
## Warning in var(x, na.rm = TRUE): NAs introduits lors de la conversion
## automatique
```

```
## Warning in var(x, na.rm = TRUE): NAs introduits lors de la conversion
## automatique
```

```
#Extracting the names of the columns which have a variance different from 0, or don't have a varaince (
n <- names(app[app==TRUE | is.na(app)==TRUE])
dfa <- df[,n]
```

```
#Splitting the day and the hour
split <- strsplit(as.character(dfa$date),' ')
merged <- as.data.frame(do.call("rbind",split))
colnames(merged) <- c("day","hour")
```

```
#Combining into new df
dfb <- cbind(merged,dfa)
dfb$date <- NULL
```

```
str(dfb)
```

```
## 'data.frame':    2184 obs. of  13 variables:
##  $ day     : Factor w/ 120 levels "2018-09-01","2018-09-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ hour    : Factor w/ 24 levels "00:00:00","01:00:00",..: 1 2 3 4 5 7 8 9 10 11 ...
##  $ temp    : num  14 14 15 14 15 14 15 15 16 16 ...
##  $ dew_pt  : num  13 13 14 14 14 14 14 14 14 14 ...
##  $ hum     : num  90 95 96 92 93 93 91 90 85 81 ...
##  $ wind_spd: num  11.1 7.4 13 9.3 9.3 11.1 9.3 7.4 13 14.8 ...
##  $ dir     : chr  "NNW" "WNW" "NW" "NNW" ...
##  $ vis     : num  45 40 10 40 40 30 35 30 40 35 ...
##  $ pressure: num  1021 1021 1022 1022 1022 ...
##  $ cond    : chr  "" "Scattered Clouds" "Rain" "Clear" ...
##  $ fog     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ rain    : num  0 0 1 0 1 0 0 0 0 0 ...
##  $ snow    : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
#Changing some variables to factors
change <- c("dir","cond","fog","rain","snow")
for (i in 1:length(change)){
  x <- change[i]
  dfb[,x] <- as.factor(dfb[,x])
}
str(dfb)
```

```
## 'data.frame':    2184 obs. of  13 variables:
##  $ day     : Factor w/ 120 levels "2018-09-01","2018-09-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ hour    : Factor w/ 24 levels "00:00:00","01:00:00",..: 1 2 3 4 5 7 8 9 10 11 ...
##  $ temp    : num  14 14 15 14 15 14 15 15 16 16 ...
##  $ dew_pt  : num  13 13 14 14 14 14 14 14 14 14 ...
##  $ hum     : num  90 95 96 92 93 93 91 90 85 81 ...
##  $ wind_spd: num  11.1 7.4 13 9.3 9.3 11.1 9.3 7.4 13 14.8 ...
##  $ dir     : Factor w/ 17 levels "","East","ENE",..: 7 16 9 7 15 9 16 16 9 7 ...
##  $ vis     : num  45 40 10 40 40 30 35 30 40 35 ...
##  $ pressure: num  1021 1021 1022 1022 1022 ...
##  $ cond    : Factor w/ 21 levels "","Clear","Drizzle",..: 1 19 17 2 17 13 19 2 13 19 ...
##  $ fog     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ rain    : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 1 1 1 1 ...
##  $ snow    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
#Splitting columns between continuous and factors
fact <- sapply(dfb,is.factor)
fact_names <- names(fact[fact])
cont_names <- names(fact[!fact])

#Removing unrelevant elements
fact_names <- fact_names[-c(1,2)]
```

```r
#Getting the means for continuous variables
means <- apply(dfb[,cont_names],2, aggregate,list(dfb$day),mean,na.rm=TRUE)
```

```r
#Getting the modes for factor variables
get_mode <- function(col){
  name <- colnames(col)
  c <- count(col,name)
  m <- max(c$freq)
  mode <- c[c$freq==m,][1,1]
  if (mode==""){
    m2 <- max(c$freq[c$freq!=m])
    mode <- c[c$freq==m2,][1,1]
  }
  return(mode)
}

modes <- sapply(dfb[,fact_names], aggregate,list(dfb$day),get_mode)
```

```r
#Merging means and modes into a df
col_names <- c("day",cont_names,fact_names)
days <- unique(dfb$day)
df_means <- as.data.frame(matrix(0,ncol = length(col_names), nrow = length(days)))
colnames(df_means) <- col_names
df_means[,1] <- as.character(days)
for (i in 1:length(days)){
  for(j in 1:length(cont_names)){
  df_means[i,j+1] <- means[j][[1]][i,"x"]
  }
  for(j in 1:length(fact_names)){
  df_means[i,j+length(cont_names)+1] <- as.character(modes[[2*j]][i])
  }
}
```

##Meter data

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## caractère(s) 'nul' au milieu de l'entrée
```

```
#Checking which ID have at least 121 records
occ_id <- count(md$ID)
occ_id <- occ_id[occ_id$freq>120,]

md2 <- md[md$ID %in% occ_id$x,]
str(md2)
```

```
## 'data.frame':    10043 obs. of  3 variables:
##  $ ID     : int  4529799 4529800 4839509 4866195 4887707 4962433 5037175 5093913 5093998 5140250 ...
##  $ Time   : Factor w/ 1455 levels "01-09-2018 02.00",..: 10 12 2 12 2 1 2 2 1 12 ...
##  $ Reading: Factor w/ 11254 levels "0,905","1031,40",..: 83 82 23 19 11 38 75 60 90 15 ...
```

```
#Converting reading to continuous variable
md2$Reading <- gsub(",",".",md2$Reading)
md2$Reading <- as.numeric(md2$Reading)
```

```
#Converting time to date type
md2$Time <- strptime(md2$Time,format = "%d-%m-%Y %H.%M")
md2$Time <- as.POSIXct(md2$Time)
```

```
#Splitting the day and the hour
split <- strsplit(as.character(md2$Time),' ')
merged <- as.data.frame(do.call("rbind",split))
colnames(merged) <- c("day","hour")

#Binding day and hour splitted to the df
md2 <- cbind(md2,merged)
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
```

```
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values
```

##Merging WUnderground and meter data

```
days <- unique(inter$day)
day <- days[1]
X <- inter[which(as.character(inter$day)==as.character(day)),c("ID","cons")]
v <- df_means[which(df_means$day==day),]
Y <- as.data.frame(matrix(v,nrow = nrow(X),ncol = ncol(v),byrow = TRUE))
final <- cbind(X,Y)
colnames(final) <- c("id","consumption",colnames(df_means))

for (i in 2:length(days)){
  day <- days[i]
  X <- inter[which(as.character(inter$day)==as.character(day)),c("ID","cons")]
  v <- df_means[which(df_means$day==day),]
  Y <- as.data.frame(matrix(v,nrow = nrow(X),ncol = ncol(v),byrow = TRUE))
  Z <- cbind(X,Y)
  colnames(Z) <- c("id","consumption",colnames(df_means))
```

```
  final <- rbind(final,Z)
}
```

##Correcting the types of some column of the data frame

```
final[,c("temp","dew_pt","hum","wind_spd","vis","pressure")] <- sapply(final[,c("temp","dew_pt","hum","

final[,c("id","day","dir","cond","fog","rain","snow")] <- sapply(final[,c("day","day","dir","cond","fog

final[,c("id","day","dir","cond","fog","rain","snow")] <- sapply(final[,c("day","day","dir","cond","fog
```

#Removing remaining NAs

```
final <- final[-which(is.na(final$temp)),]
```

```
summary(final)
```

```
##       id             consumption          day                 temp
##  Length:9794        Min.   :0.00000   Length:9794        Min.   :-1.800
##  Class :character   1st Qu.:0.07458   Class :character   1st Qu.: 4.579
##  Mode  :character   Median :0.15153   Mode  :character   Median : 8.905
##                     Mean   :0.43395                      Mean   : 8.733
##                     3rd Qu.:0.33469                      3rd Qu.:12.833
##                     Max.   :8.00929                      Max.   :18.500
##      dew_pt            hum           wind_spd           vis
##  Min.   :-3.600   Min.   :49.00   Min.   : 3.713   Min.   : 1.965
##  1st Qu.: 2.190   1st Qu.:73.10   1st Qu.:11.305   1st Qu.:11.667
##  Median : 6.833   Median :82.32   Median :15.195   Median :17.645
##  Mean   : 6.317   Mean   :81.01   Mean   :16.363   Mean   :20.601
##  3rd Qu.: 9.947   3rd Qu.:89.30   3rd Qu.:20.786   3rd Qu.:29.571
##  Max.   :15.583   Max.   :98.39   Max.   :41.929   Max.   :50.000
##     pressure          dir                cond               fog
##  Min.   : 985.8   Length:9794        Length:9794        Length:9794
##  1st Qu.:1011.1   Class :character   Class :character   Class :character
##  Median :1017.4   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1016.5
##  3rd Qu.:1022.4
##  Max.   :1040.2
##      rain               snow
##  Length:9794        Length:9794
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```