

**Big Data Systems and Statistics joint project**

Israa Atike D00262160

**Department of Computing Science and Mathematics**

**Dundalk Institute of Technology (DkIT)**

Contents

[Project description: 3](#_Toc164545396)

[Research methodology: 3](#_Toc164545397)

[Data wrangling and cleaning 4](#_Toc164545398)

[Exploratory Analysis 8](#_Toc164545399)

[Research Analysis and Findings 20](#_Toc164545400)

[Conclusions and Recommendations 34](#_Toc164545401)

[Bibliography & Appendices 34](#_Toc164545402)

# Project description:

In this project, I will be covering the analysis of 4 files, hourly\_dublin\_17\_18\_19, KeyHourly.txt, JLHome1718Power.csv and JLHome1718Temperature.csv. The first file provides data about weather conditions in Dublin airport from 1-1-17 to 31-12-19, the second file provides information about columns of the first file, the third file provides power consumption in a house in Dundalk from 12-10-17 to 13-10-19 and the fourth file provides data about the same house's temperature measures from 12-10-17 to 13-10-19. The three files have different time intervals, which is one of the challenges that we will try to solve in this project, from a big data point of view, I will be focusing on data preparation and cleaning, choosing an appropriate database, to analyse the data back it up and implement security measures and from a statistics point of view I will be focusing on cleaning and preparing data using R programming language, exploring the data using visualisations, carrying out linear regressions and most importantly developing research questions that we will be able to answer at the end of our analysis, there are many possible research questions for the provided datasets such as:  
1- How do changes in temperature, humidity, rainfall, wind speed, sunlight duration, and visibility affect power consumption?  
2- How does power consumption vary across different seasons?   
3- What factors of our datasets influence power consumption?  
I will adopt the third research question, and I will consider power consumption as my response variable.

Research methodology:  
In this project, I will be using R programming language to do my statistical analysis, most of my variables are numerical except date which is a categorical variable to find the variation in power consumption based on months, find the details of my variables below:  
For my first dataset hourly\_dublin\_17\_18\_19:  
Date: this gives date information in hours of the below variables from 1-1-17 to 31-12-19 which is a categorical variable.  
Rain: Precipitation Amount in mm which is a quantitative continuous variable.  
temp: Air Temperature in degrees Celsius which is a quantitative continuous variable.  
rhum: Relative Humidity percentage which is a quantitative discrete variable.  
wdsp: Mean Hourly Wind Speed in kt which is a quantitative discrete variable.  
sun: Sunshine duration in hours which is a quantitative continuous variable.  
vis: Visibility in meters which is a quantitative discrete variable.  
For my second dataset JLHome1718Temperature.csv:  
Date: this gives date information in the previous 5 minutes of the below variable from 12-10-17 to 13-10-19 which is a categorical variable.  
Temperature: measures the temperature of the house in degrees Celsius which is a quantitative continuous variable.  
For my third dataset JLHome1718Power.csv:  
Date: this gives date information in the previous minute of the below variable from 12-10-17 to 13-10-19 which is a categorical variable  
Power: measures the power usage in watts which is a quantitative discrete variable

hourly\_dublin\_17\_18\_19, JLHome1718Power.csv and JLHome1718Temperature.csv are real data about the weather at Dublin Airport and the temperature and energy use in a house in Dundalk.  
The energy usage and temperature data are gathered using an OpenEnergyMonitor emonPi. OpenEnergyMonitor is a small company based in Wales that uses the emonPi to give real-time energy usage monitoring.  
So the first thing that I had to figure out was how to use all three datasets at once, to apply everything I have learned in statics modules. To attain my objective I decided to start by aggregating my datasets along with cleaning them to handle missing values and outliers also I have created additional variables such as year and month from the date column which will help me more in establishing the categorical relationship that my variables have with date, there are many techniques that I have used to analyse my datasets such as building suitable visualisation to discover how my dataset would look like, calculating descriptive statistics to find out how my data is spread out to try to find solutions to handle any outliers, and finally conducting simple and multiple linear regressions to understand what are the factors that influence our response variable the most, another technique I will be using for this analysis are the test that enables you to compare multiple linear regression models to choose the best one.   
I have chosen to adopt the above research methodology for my statistics part since it will be very helpful in terms of analysing the variables also in terms of ending up with the right interpretation of the whole dataset.

In terms of my databases part I have tried to work on my datasets using both SQL and MongoDB since each one of them serves a different purpose, I have tried to import my datasets in SQL first but since they were very large, SQL could not handle them so I had to break them down into subsets of data to import them, once I imported them I faced another issue which was the nature of my dataset, the nature of my dataset does not give the ability to combine the 3 datasets into one dataset since they are not related by anything and we cannot establish a relationship between all the 3 datasets to store them in one dataset, and keeping them in 3 different datasets will not work either since they are not related so if we want to develop queries that combine the 3 datasets we will have to find a relationship between all of them or a foreign key that will give us the access to go from one dataset to the other, as SQL is known to be a schematic database, but my dataset does not have a schema that will fit in SQL schema, even if SQL is not the best database for my dataset, I have done some interesting analysis using that database that will be provided in the text file.  
I have decided to carry out my work with MongoDB since it is a schema-free database, that gives the user access to all collections without the need for a foreign key that will establish the relationship between the collections, also because it can handle the large volumes of my datasets and I will not have to subset the datasets into small subsections once I decided to carry out my work using MONGO DB I had to convert the three datasets into the same time unit so I can use them in my analysis as well as cleaning them and implementing their measures of database security, and the measures for backing up the database.  
I chose to go with the above methodology since it will be more practical efficient and fast.

# Data wrangling and cleaning

So the first thing I had to figure out was how to convert all three 3 dates to the same unit, so I decided to convert all of them to the unit hour as the first dataset since it was the most convenient after I merged all of the 3 datasets into one dataset using the date column now that they have the same unit, I extracted a subset of the data that has the interval variable from 12-10-17 to 13-10-19 since it is the interval available in all 3 datasets.   
find the details code and explanation below:

library(dplyr)

# transofrming power dataset  
power\_data <- read.csv("JLHome1718Power1.csv", header = TRUE, stringsAsFactors = FALSE)  
colnames(power\_data) <- c("datetime", "power")  
  
power\_data$datetime <- as.POSIXct(power\_data$datetime, format = "%d/%m/%Y %H:%M")  
  
hourly\_total\_power <- power\_data %>%  
 mutate(date\_p = format(datetime, "%Y-%m-%d %H:00")) %>%  
 group\_by(date\_p) %>%  
 summarize(power\_by\_hour = sum(power))  
hourly\_total\_power$date\_p <- as.POSIXct(hourly\_total\_power$date\_p, format = "%Y-%m-%d %H:00")  
  
hourly\_total\_power

temperature\_data <- read.csv("JLHome1718Temperature1.csv", header = FALSE, stringsAsFactors = FALSE)  
colnames(temperature\_data) <- c("datetime", "temperature")  
  
temperature\_data$datetime <- as.POSIXct(temperature\_data$datetime, format = "%d/%m/%Y %H:%M")  
  
temperature\_data$date\_t <- as.POSIXct(round(as.numeric(temperature\_data$datetime) / (60 \* 60)) \* (60 \* 60), origin = "1970-01-01")  
  
hourly\_mean\_temperature <- temperature\_data %>%  
 group\_by(date\_t) %>%  
 summarize(mean\_temperature = mean(temperature, na.rm = TRUE))  
  
hourly\_mean\_temperature

hourly\_total\_power$date\_p <- as.POSIXct(hourly\_total\_power$date\_p, format = "%Y-%m-%d %H:00")  
merged\_data <- inner\_join(hourly\_total\_power, hourly\_mean\_temperature, by = c("date\_p" = "date\_t"))  
merged\_data

weather\_data <- read.csv("hourly\_dublin\_17\_18\_19.csv")  
  
weather\_data$date <- as.POSIXct(weather\_data$date, format = "%d/%m/%Y %H:%M")  
  
# Merge merged\_data and weather\_data  
merged\_data1 <- merge(merged\_data, weather\_data, by.x = "date\_p", by.y = "date", all = TRUE)  
#merged\_data1  
  
# Rename the 'date' column to 'datetime'  
colnames(merged\_data1)[1] <- "datetime"  
  
subset\_data <- subset(merged\_data1, datetime >= as.POSIXct("2017-10-12 23:00:00") &   
 datetime <= as.POSIXct("2019-10-13 18:00:00"))  
#subset\_data  
  
missing\_data <- sapply(subset\_data, function(x) sum(is.na(x)))  
  
missing\_data

## datetime power\_by\_hour mean\_temperature rain   
## 0 290 290 0   
## temp rhum wdsp sun   
## 0 0 0 0   
## vis   
## 0

missing\_percentage <- sapply(subset\_data, function(x) {  
 missing\_count <- sum(is.na(x))  
 total\_count <- length(x)  
 percentage <- (missing\_count / total\_count) \* 100  
 return(percentage)  
})  
  
missing\_percentage

## datetime power\_by\_hour mean\_temperature rain   
## 0.000000 1.653552 1.653552 0.000000   
## temp rhum wdsp sun   
## 0.000000 0.000000 0.000000 0.000000   
## vis   
## 0.000000

##since the data missing is less than 5% we will delete it.  
cleaned\_data <- subset\_data[complete.cases(subset\_data), ]  
#cleaned\_data  
  
cleaned\_data$datetime <- as.POSIXct(cleaned\_data$datetime)  
  
# Extract month from datetime and put it in a new variable called month  
cleaned\_data$month <- format(cleaned\_data$datetime, "%m")  
  
  
# Extract year from datetime and put it in a new variable called year  
cleaned\_data$year <- format(cleaned\_data$datetime, "%Y")  
  
head(cleaned\_data)

I first started by loading my power dataset and cleaning the date column by converting it to the right format and then I converted the unit of 1 minute into a unit of one hour by summing all the power entries of the previous hour, same thing I did with my temperature dataset I converted the date column into the right format, I converted the 5 minutes unit into an hour unit and I got the average of each hour for the temperature value. Then I merged my two datasets using the date column into a dataset called merged\_data.  
After I loaded my weather dataset, I converted the date column into the right format I merged the merged\_data with weather\_data using date again and I called my second mere merged\_data1, after that, I tried to get the subset of the data that will have dates from 12-10-17 to 13-10-19, to dismiss the missing values of power\_by\_hour and mean\_temperature in the other dates and then I tried to look in my data for missing values as you can see above I had a percentage less than 5% of missing values thus I decided to omit them, finally I created two new columns related to datetime column that I called month and year for visualisation and analysis purposes.

For the database, I decided to carry out my work using Mongo DB since it is more beneficial for this dataset than SQL, in terms of cleaning my dataset the first thing I did was started by importing my files in both Mongo DB and SQL, but SQL did not handle the large datasets I had, so I decided to use Mongo DB instead for this project, I imported my datasets in Mongo DB I have specified the types of data in each column before importing that, except the date column since I specified that using a query later on, I will include below some screenshots to show the types of my column

A white background with black text

Description automatically generated

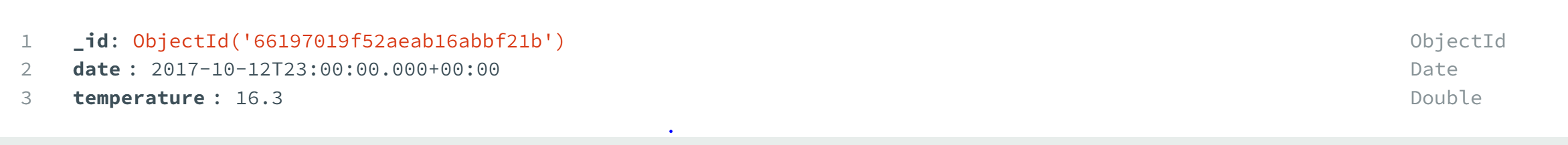
Fig1: types of my column of hourly\_dublin  
  


Fig2: types of my column of temperature



Fig3: types of my column of power

the second step was to convert the files JLHome1718Power.csv and JLHome1718Temperature.csv. into the same time frame as hourly\_dublin\_17\_18\_19, since the file JLHome1718Power.csv had time intervals of a minute JLHome1718Temperature.csv had time intervals of 5 minutes and hourly\_dublin\_17\_18\_19 had time intervals of an hour, thus I converted all of them to have an hour time interval, and I put them in two new collections, for JLHome1718Power.csv I summed all the power consumed within an hour and for JLHome1718Temperature.csv. I got the average temperature within an hour I will include some screenshots of my two new collections below.

A screenshot of a computer

Description automatically generated

Fig4: a screenshot of a sample of temperature dataset by hour

A screenshot of a computer

Description automatically generated

Fig: screenshot of a sample of power dataset by hour

In my text file, I have included the queries that enabled me to clean my data as well as to aggregate those two new collections, I have also included some general queries that we have learned during this course, these queries try to analyse my collections in different ways. I have included them to practice what we have learned in this module.

# Exploratory Analysis

In this section, I will be focusing on analysing my dataset using biviariates and multivariates along with descriptive statistics and I will try to handle any outliers, for missing values I have already handled them in the previous section once I merged my datasets.

Descriptive statistics and univariates:  
power\_by\_hour:  
The power consumption ranges from a minimum of 37 watts to a maximum of 173320 watts with an average consumption of 18276 watts per hour and a median of 9582, looking at the 3rd quartile which is 37256 watts per hour and the maximum which is 173320 watts per hour we can consider the column of power\_by\_hour as very skewed and as full of outliers that is why we will be working with median and IQR for this column.

A diagram of power by hour

Description automatically generated

Fig 6: Boxplot of power by hour.

As we can see in the above boxplot my variable power\_by\_hour is skewed and has a lot of outliers, thus I will try to get rid of some very extreme outliers so the variable is less skewed for further analysis.

A graph of power by hour

Description automatically generated

Fig7: histogram of power by hour.

As we can see from the above histogram power\_by\_hour is very skewed for further analysis accordingly we will get rid of extreme outliers in further analysis.

Mean\_temperature:  
The mean\_temperature ranges from a minimum of 6.038 °C to a maximum of 28.836 °C with an average temperature of 16.445 °C and a median of 16.300 °C and a 3rd quartile of 18.500 °C so we can consider the variable mean\_temperature as a symmetrical non-skewed variable thus we will not need to do any changes regarding outliers.  
As we can see in both the boxplot and the histogram this variable is symmetrical with some outliers, that are not extreme and that do not influence the variable much.  
A diagram of a graph

Description automatically generated

Fig8: Boxplot of the mean of temperature.

A graph of a temperature

Description automatically generated

Fig: Histogram of the mean of temperature

Rain:

The rain ranges from a minimum of 0 mm to a maximum of 12.2 mm with an average rain of 0.08586 mm a median of 0 mm, and a 3rd quartile of 0 mm so we can consider the variable rain as a skewed variable but we will not need to do any changes regarding outliers since they are not very extreme.  
As we can see in both the boxplot and the histogram this variable has a lot of outliers and it is not symmetrical, since most of the time the rain is at 0 mm.  
A graph of rain with black lines

Description automatically generated

Fig10: Boxplot of rain.

A graph of rain and rain

Description automatically generated  
  
fig11:histogram of rain.

Temp:  
The temp ranges from a minimum of -5.6 °C to a maximum of 26.3 °C with an average temperature of 9.81 °C a median of 9.7 °C, and a 3rd quartile of 13.6 °C so we can consider the variable temp as a symmetrical non-skewed variable thus we will not need to do any changes regarding outliers.

As we can see in the boxplot and histogram below the variable temp is symmetrical with a little bit of outliers that are not influencing it a lot.

A diagram of a box plot

Description automatically generated

Fig12: Boxplot of temperature in the air

A graph of a temperature

Description automatically generated  
  
Fig13: Histogram of temperature in the air.

Rhum:  
The rhum variable ranges from a minimum of 25% to a maximum of 100% with an average of 81.53% a median of 84.00%, and a 3rd quartile of 91.00% so we can consider the variable rhum as an almost symmetrical variable thus we will not need to do any changes regarding outliers since they do not influence the variable that much.

As we can see in the boxplot and histogram below rhum variable is almost symmetrical with some outliers that are not very extreme thus we will not need to get rid of them in order not to lose valuable data.

A diagram of a box plot

Description automatically generated

Fig14: Boxplot of relative humidity

A green graph with black text

Description automatically generated

Fig15: histogram of relative humidity

Wdsp:  
The wdsp variable ranges from a minimum of 1 kt to a maximum of 35 kt with an average of 9.856 kt a median of 9 kt and a 3rd quartile of 13 kt so we can consider the variable wdsp as an almost symmetrical variable thus we will not need to do any changes regarding outliers since they do not influence the variable that much.  
As we can see in the boxplot and histogram below our variable is almost symmetrical with a little bit of outliers that we will not omit since they do not influence our variable extremely.

A blue line with black text

Description automatically generated

Fig16: Boxplot of mean hourly wind speed

A graph of a wind speed

Description automatically generated

Fig17: Histogram of mean hourly wind speed.

Sun:   
The sunshine duration variable ranges from a minimum of 0 hours to a maximum of 1 hour with an average of 0.1694 hours a median of 0 hours and a 3rd quartile of 0.1 hour so we can consider the variable sun is skewed.  
As we can see in the boxplot and histogram below the variable sunshine duration is very much skewed with a little bit of outliers that are not influencing it thus we will not get rid of them.

A diagram with a box of sunshine duration

Description automatically generated with medium confidence

Fig18: Boxplot of sunshine duration

A graph with green squares and black text

Description automatically generated

Fig19: histogram of sunshine duration

Vis:  
The visibility variable ranges from a minimum of 100 m to a maximum of 75000 m with an average of 28039 m a median of 30000 m and a 3rd quartile of 35000 m so we can consider the variable visibility as a bit skewed.  
As we can see in the boxplot and histogram below our variable is a bit skewed with a few outliers  
that are not influencing the variable since they are not very extreme, thus we will not omit those outliers.

A diagram with a blue line

Description automatically generated

Fig20: Boxplot of visibility.

A graph of a graph

Description automatically generated with medium confidence

Fig21: Histogram of visibility.

The barplot below shows the frequency of the months we have in our dataset they have almost the same frequency, the slight difference between the frequencies is because this data varies from 12-10-17 to 13-10-19.

A graph of blue bars

Description automatically generated

Fig22: Frequency of months.

The below bar plot shows the years that my dataset contains so the year with the highest frequency is 2018 and then 2019 and finally 2017. The difference between the frequencies is also because this data varies from 12-10-17 to 13-10-19

A green rectangles with numbers and a black text

Description automatically generated

Fig23: Frequency of years.

Below we can see a boxplot that shows the variation of power for the different months we have, there is not a big variation but we can recognise that in hot months the power consumption is lower than in the cold months.

A graph of power by hour

Description automatically generated

Fig24: Boxplot of power by hour.

In the below boxplot, we can see that the reason behind the change in power consumption is the temperature, so when the temperature increases power consumption decreases.

A graph showing a box plot

Description automatically generated with medium confidence

Fig25: Boxplot of mean temperature with months.

For all other variables, they also vary depending on months and seasons.

In the below scatterplot, we can see that most of our variables do not have an obvious correlation with each other, and only one of them has a linear relationship which is temp and mean\_temperature.

A group of black squares

Description automatically generated

Fig26: Pairs plot for my dataset.

# Research Analysis and Findings

After visualising my data now it is time to start analysing it and conducting simple and multiple linear regressions, first thing I will start by analysing the pair panels for the simple linear regression to discover the correlation factors between my variables and knowing that our response variable is very much skewed the same way we see below, so I decided to add the log transformation to make it less skewed to verify the assumptions.

A screenshot of a computer

Description automatically generated

Fig27: panels pair plot

The below pair panel shows the log of power\_by\_hour with every other variable from the dataset, along with the correlation factors and as we can see the log of power\_by\_hour has the highest correlation with rhum sun and wdsp, so these are the 3 variables we will try for our simple linear regression. It also has a high correlation with the power\_by\_hour but it is just because it is the same variable.

A screenshot of a computer

Description automatically generated

Fig28: panels pair plot using the log of power\_by\_hour

The first model: that I developed was lm(log(power\_by\_hour) ~rhum).  
The second model: was lm(log(power\_by\_hour) ~rhum) but without outliers.  
The third model: was lm(log(power\_by\_hour) ~wdsp).  
The fourth model: was lm(log(power\_by\_hour) ~sun.  
After analysing the assumptions of each one of them and after comparing all of them using AIC and BIC everything is included in the R markdown file, I ended up choosing the best model which was the second one, I will include its assumption below along with its summary and its comparison with the second best model which is the first one.

Assumptions of the first model:  
A graph of a graph with numbers and a line

Description automatically generated with medium confidence

Fig 29: Residuals vs fitted for the first model

A graph of a line

Description automatically generated

Fig 30: QQplot for the first model

A graph of a graph

Description automatically generated

Fig31: Histogram for the first model.

Call:

lm(formula = log(power\_by\_hour) ~ rhum, data = cleaned\_data)

Residuals:

Min 1Q Median 3Q Max

-5.8250 -0.8206 -0.1648 0.8412 2.9011

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 10.2724112 0.0491139 209.2 <2e-16 \*\*\*

rhum -0.0114936 0.0005956 -19.3 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9634 on 17246 degrees of freedom

Multiple R-squared: 0.02114, Adjusted R-squared: 0.02108

F-statistic: 372.4 on 1 and 17246 DF, p-value: < 2.2e-16

So as we can see above the assumptions were acceptable.  
We have a constant variance the variance of the residuals is similar across the values of the independent variable.   
it is i.i.d residuals are independently and identically distributed it has a random scatter  
But the above assumptions can be better, for the Adjusted R squared value it is very weak, but we did interpret this model since assumptions are almost verified.  
  
The second model (the best one):  
A graph of black and white dots

Description automatically generated

Fig32: residuals vs fitted for the second model

A graph with a line

Description automatically generated

Fig33: QQplot for second model.

A graph of a graph

Description automatically generated

Fig34: Histogram of the second model

So as we can see above the assumptions were better than the first model.  
We have a constant variance the variance of the residuals is similar across the values of the independent variable.   
it is i.i.d residuals are independently and identically distributed it has a random scatter  
But the above assumptions can be better for the Adjusted R squared value it is still very weak, but it is better than the first model.

A screenshot of a computer

Description automatically generated

Fig35: AIC of the first and second models.

A screenshot of a computer

Description automatically generated  
fig36: BIC of the first and second models.

Call:

lm(formula = log(power\_by\_hour) ~ rhum, data = cleaned\_data[-all\_outliers,

])

Residuals:

Min 1Q Median 3Q Max

-5.8249 -0.8204 -0.1647 0.8412 2.9011

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 10.2722301 0.0491084 209.2 <2e-16 \*\*\*

rhum -0.0114924 0.0005956 -19.3 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9632 on 17238 degrees of freedom

Multiple R-squared: 0.02115, Adjusted R-squared: 0.02109

F-statistic: 372.4 on 1 and 17238 DF, p-value: < 2.2e-16

Looking at the output of the AIC and BIC we can confirm that the best model is the second one since it has a lower value in both tests along with a higher R-adjusted square value and a lower standard error.

**Equation of simple linear regression:  
power\_by\_hour= 10.2722301 – 0.0114924 rhum  
Interpretation:**  
**this model means the power\_by\_hour would already be at 10.2722301 watts and if you multiply rhum by 1 the power\_by\_hour will decrease by 0.0114924.**

Further analysing our data, now we will start looking at the multiple linear regression.  
For my multiple linear regression models, I ended up developing 5 regressions, and I chose the best one of them, as below  
The first model:  
lm(log(power\_by\_hour)~temp+factor(month)+rhum+vis+sun+rain+mean\_temperature+wdsp, cleaned\_data).  
The second model:  
lm\_red<-lm(log(power\_by\_hour)~temp+rhum+sun+rain+mean\_temperature+wdsp+factor(month), cleaned\_data)  
The third model:  
lm\_red2<-lm(log(power\_by\_hour)~temp+rhum+ sun+rain+mean\_temperature+log(wdsp) +factor(month), cleaned\_data).  
The fourth model:  
lm\_red\_final <- lm(log(power\_by\_hour) ~ rhum + sun+ rain + mean\_temperature + log(wdsp) + factor(month), data = cleaned\_data).  
The fifth model:  
lm(log(power\_by\_hour) ~ rhum + sun+ rain + mean\_temperature + log(wdsp) + factor(month), data = cleaned\_data)

I will not include the assumptions of each one of the models since I already included them in my Rmarkdown but I will tell the reason behind choosing the last model as the best one.  
The issue with the first model was that visibility was not significant along with month 11 for the categorical variable, so in the second model, I got rid of visibility.   
The issue with the second model was that the assumptions were not the best as well as the r-adjusted squared value.  
The issue with the third model was that the assumptions were not the best along with the issue of month 11, I did not get rid of month 11 straight away after the first model, to see if its significance would change in case I changed the model. But it was better than the first and second ones since I added the log transformation on wdsp to make it more linear.  
The issue with the fourth model was outliers and month 11 which was still not significant, thus in the next model I decided to get rid of outliers of the column power\_by\_hour as well as month 03 and 11 of the categorical variable since both of them became insignificant.   
I chose the fifth model since it had the best assumptions the best-adjusted r-squared value, and the weakest values of AIC and BIC.

The following figures will present the assumptions for the fifth and best model along with the summary statistics for it, and the AIC and BIC comparison with the second best model which is the fourth model.  
  
A black and white diagram with numbers and a black line

Description automatically generated with medium confidence

Fig37: residuals vs fitted of the fifth model.

A graph with a line

Description automatically generated

Fig38: QQplot of the fifth model.

A graph with numbers and a line

Description automatically generated with medium confidence

Fig39: residuals vs levarge for the fifth model.

So as we can see above the assumptions.  
We have a constant variance the variance of the residuals is similar across the values of the independent variable.   
i.i.d Residuals are independently and identically distributed and it has random scatter also, the cook’s distance plot is good.

GVIF Df GVIF^(1/(2\*Df))

temp 3.547008 1 1.883350

factor(month) 3.997455 11 1.065010

rhum 2.567486 1 1.602338

vis 1.920226 1 1.385722

sun 1.387713 1 1.178012

rain 1.082452 1 1.040410

mean\_temperature 3.732098 1 1.931864

wdsp 1.240332 1 1.113702

We can see that the VIF table is perfectly fine.

Anova Table (Type II tests)

Response: log(power\_by\_hour)

Sum Sq Df F value Pr(>F)

rhum 114.3 1 133.340 < 2.2e-16 \*\*\*

sun 86.8 1 101.197 < 2.2e-16 \*\*\*

rain 11.9 1 13.884 0.0001951 \*\*\*

temp 12.9 1 15.059 0.0001047 \*\*\*

mean\_temperature 249.2 1 290.632 < 2.2e-16 \*\*\*

log(wdsp) 37.2 1 43.430 4.547e-11 \*\*\*

factor(month) 675.4 9 87.519 < 2.2e-16 \*\*\*

Residuals 12259.6 14298

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

We can see that our ANOVA table is perfectly good as well.

A screenshot of a computer

Description automatically generated

Fig40: BIC of the fifth model.

A screenshot of a computer

Description automatically generated

Fig41: AIC of the fifth model.

We can see that this model which doesn’t have outliers of the column power\_by\_hour and the months 03 and 11 from the categorical variable is the best since it has the lowest AIC and BIC values.

Call:

Anova Table (Type II tests)

Response: log(power\_by\_hour)

Sum Sq Df F value Pr(>F)

rhum 114.3 1 133.340 < 2.2e-16 \*\*\*

sun 86.8 1 101.197 < 2.2e-16 \*\*\*

rain 11.9 1 13.884 0.0001951 \*\*\*

temp 12.9 1 15.059 0.0001047 \*\*\*

mean\_temperature 249.2 1 290.632 < 2.2e-16 \*\*\*

log(wdsp) 37.2 1 43.430 4.547e-11 \*\*\*

factor(month) 675.4 9 87.519 < 2.2e-16 \*\*\*

Residuals 12259.6 14298

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

GVIF Df GVIF^(1/(2\*Df))

temp 3.547008 1 1.883350

factor(month) 3.997455 11 1.065010

rhum 2.567486 1 1.602338

vis 1.920226 1 1.385722

sun 1.387713 1 1.178012

rain 1.082452 1 1.040410

mean\_temperature 3.732098 1 1.931864

wdsp 1.240332 1 1.113702

Call:

lm(formula = log(power\_by\_hour) ~ rhum + sun + rain + temp +

mean\_temperature + log(wdsp) + factor(month), data = cleaned\_data)

Residuals:

Min 1Q Median 3Q Max

-5.5184 -0.7092 -0.1599 0.7311 3.2042

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.9483582 0.1084885 82.482 < 2e-16 \*\*\*

rhum -0.0096719 0.0008376 -11.547 < 2e-16 \*\*\*

sun 0.2754919 0.0273858 10.060 < 2e-16 \*\*\*

rain 0.0776558 0.0208405 3.726 0.000195 \*\*\*

temp -0.0106508 0.0027446 -3.881 0.000105 \*\*\*

mean\_temperature 0.0756776 0.0044391 17.048 < 2e-16 \*\*\*

log(wdsp) 0.1018751 0.0154587 6.590 4.55e-11 \*\*\*

factor(month)02 0.1312350 0.0353114 3.717 0.000203 \*\*\*

factor(month)04 -0.1336191 0.0354788 -3.766 0.000166 \*\*\*

factor(month)05 -0.3165934 0.0396753 -7.980 1.58e-15 \*\*\*

factor(month)06 -0.7742846 0.0445882 -17.365 < 2e-16 \*\*\*

factor(month)07 -0.8162313 0.0488726 -16.701 < 2e-16 \*\*\*

factor(month)08 -0.7277660 0.0440574 -16.519 < 2e-16 \*\*\*

factor(month)09 -0.3995114 0.0409040 -9.767 < 2e-16 \*\*\*

factor(month)10 -0.1917486 0.0375797 -5.102 3.40e-07 \*\*\*

factor(month)12 0.1992141 0.0343610 5.798 6.87e-09 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.926 on 14298 degrees of freedom

Multiple R-squared: 0.09704, Adjusted R-squared: 0.09609

F-statistic: 102.4 on 15 and 14298 DF, p-value: < 2.2e-16

As we can see above our model does not have a very high adjusted r-squared value but that is the best model I could get, also I am not very worried since the assumptions are met.  
Thus, I chose this model as a final one since it has the best R squared value, the lowest standard error, the best assumptions, and the lowest AIC and BIC values and all the variables are significant inside the model.

**Equation:  
power\_by\_hour = 8.9222701 -0.0085193 rhum +0.2729654 sun +0.0731852 rain- 0.0106508 temp+ 0.0677064 mean\_temperature+ 1.094643 wdsp+ 0.1431685 month 02 - 0.1444401 month 04 -0.3324378 month 05 -0.7999290 month 06 -0.8610650 month 07 -0.7810960 month 08 -0.4348402 month 09 -0.2134975 month 10 +0.1908624 month 12.**  
**Interpretation:  
power\_by\_hour equals 8. 9222701 when all other variables are zero  
If we multiply rhum by 1 power\_by\_hour will decrease by -0.0085193  
If we multiply the sun by 1 power\_by\_hour will increase by 0.2729654   
If we multiply rain by 1 power\_by\_hour will increase by 0.0731852**

**If we multiply temp by 1 power\_by\_hour will decrease by - 0.0106508  
If we multiply mean\_ temperature by 1 power\_by\_hour will increase by 0.0677064   
If we multiply wdsp by 1 power\_by\_hour will increase by 1.094643  
If we multiply month 02 by 1 power\_by\_hour will increase by 0.1431685  
If we multiply month 04 by 1 power\_by\_hour will decrease by - 0.1444401  
If we multiply month 05 by 1 power\_by\_hour will decrease by -0.3324378  
If we multiply month 06 by 1 power\_by\_hour will decrease by -0.7999290  
  
  
If we multiply month 07 by 1 power\_by\_hour will decrease by -0.8610650  
If we multiply month 08 by 1 power\_by\_hour will decrease by -0.7810960  
If we multiply month 09 by 1 power\_by\_hour will decrease by -0.4348402  
If we multiply month 10 by 1 power\_by\_hour will decrease by -0.2134975  
If we multiply month 12 by 1 power\_by\_hour will increase by 0.1908624**

Comparing the simple linear regression model and the multiple linear regression model, the best one is the multiple linear regression model since it has a weaker AIC and BIC value and better assumptions.  
A screenshot of a computer

Description automatically generated

Fig 42: AIC of the simple and multiple linear regression.

A screenshot of a computer

Description automatically generated

Fig 43: BIC of the simple and multiple linear regression.

For my databases, the reason behind my choice of Mongo DB for this part of the project is, first of all, because Mongo DB has very high scalability, which is what I need for my dataset since it is very large.  
Talking about scalability, Mongo DB has an architecture which is developed to overcome the challenges of handling large amounts of data which is called sharding. Sharding gives Mongo DB the ability to scale horizontally which means your data can be distributed among many and different servers to handle large volumes of data, sharding divides data among shards, each shard is a separate Mongo DB database that stores a part of the data, and as the dataset grows, it is possible to add more shards to divide the load and keep it performing in a fast and smooth way and also gives the ability to process queries in a parallel way. Sharding provides high availability and tolerates faults efficiently, so if for example one shard crashes, the other ones will still be working perfectly like this we will not lose the totality of our data also this will cost less than a highly available vertical server which provides high availability as sharding neither tolerates faults as efficiently, finally sharding uses agile development process which gives the developer the flexibility to adapt his work method based on the project.  
Since Mongo DB has a flexible schema such that it gives the collections the possibility to have different structures you can import documents that have different types in the same collection without having to define a schema, as my data has a variation in the types of its data Mongo DB is the best approach that could be taken, since it will easily handle additional data and different fields also mongo db uses sharding it is very easy to add additional data since sharding can store data horizontally Mongo DB and as the number of data increases we can add more shards for the additional volume, Mongo DB also provides schema validation which enforces restrictions on the format of the document which makes sure only valid data is entered into the database, so if we want to add data we should make sure it is valid to respect data integrity.   
Mongo DB will be able to scale when every home in Britain and Ireland wants to use emonPi to monitor the home since it uses horizontal scalability with sharding, so as the number of houses increases, we will be able to add more shards to handle the increasing volume of data, also because sharding can be based on geographical location so it will make it even easier for us to scale house in different locations like this we will ensure distribution of data and high performance of queries horizontal sharding is known by its fault tolerance using replication features which maintain copies of data, and in big datasets as the one we are working on this project, it will be very crucial to make sure we have a database that tolerates faults also the fact that sharding has the automatic balancer which divides data and moves it between shards to keep the work balanced.

For the backup I used the “mongodumb” query to export my database as either a CSV or a JSON file, The code used to do that is included in the text file, and for later applications we can schedule backups in Mongo DB’s built-in backup features, we can also do the backup in another way which is replica sets that consist of multiple Mongo db shards that host the same data, there are many other ways to backup the database but to stay concise and for my case, I have used the query “mongodump” in Mongo DB to export my database into a backup folder in case I lose my database.

For database security I have implemented users, roles and some views that I will list below

|  |  |
| --- | --- |
| Users | Roles |
| Israa | dbOwner |
| Dr Anesu | read |
| General Data Scientist/ Analyst | read |
| It supports | readWrite |
| Weather Data Scientist/ Analyst | read (hourly\_dublin only) |
| Power Data Scientist/ Analyst | read (hourly\_power only) |
| Temperature Data Scientist/ Analyst | read (hourly\_temperature only) |
| Database Developer | userAdmin |
| Business Analyst | read |
| Data Provider | readWrite |
| Data Steward | readWrite + userAdmin |

Fig44: Database security table.

|  |  |
| --- | --- |
| Views | Granted to |
| hourly\_dublin\_weather\_view | Weather Data Scientist/ Analyst |
| hourly\_power\_consumption\_view | Power Data Scientist/ Analyst |
| hourly\_temperature\_view | Temperature Data Scientist/ Analyst |

Fig 45: views for database security.

A diagram of a company

Description automatically generated

Fig 46: UML diagram for my database.  
  
A screenshot of a computer

Description automatically generated

Fig 47: Class diagram for my database.

Conclusions and Recommendations  
To sum up our project, and answer the research question which was what factor influence power consumption the most?  
Answer: Rhum, sun, rain, temp, mean temperature and months are what influence the power consumption the most.  
The only recommendation I would add is if I had a larger volume of data that covers more than 3 years of power consumption I would have been able to reach deeper insights, otherwise, I did not find any problem with the datasets since they were almost clean they did not have many missing values, thus they had a good quality which helped me focus more in the analysis and the databases side more rather than spending a long time in the cleaning without reaching valuable insights.

# Bibliography & Appendices

lucidchart (2019). *Introducing Types of UML Diagrams | Lucidchart Blog*. [online] Lucidchart.com. Available at: https://www.lucidchart.com/blog/types-of-UML-diagrams.  
Visual Paradigm (2019). *What is a Class Diagram?* [online] Visual-paradigm.com. Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/.  
MongoDB. (n.d.). *How To Back Up And Restore MongoDB*. [online] Available at: https://www.mongodb.com/basics/backup-and-restore.  
www.mongodb.com. (n.d.). *Security — MongoDB Manual*. [online] Available at: https://www.mongodb.com/docs/manual/security/.

‌

‌