48 Hour Dataset Analyis

Author: Lena Schwertmann

Course: Software for Analyzing Data

Objective: Analyze the `buildings` data set within 48 hours. What is the story of this dataset? What can we learn from it?

Timeframe for analysis: Saturday, 2020-Feb-01 11 am - Monday, 2020-Feb-03 11 am

How to run this code:

1. Below, set your working directory
2. Add the data file "building_1319.csv" to that directory
3. If necessary, install the required packages that are loaded below

1 Preliminaries

```
# INSERT THE PATH TO YOUR WORKING DIRECTORY HERE
setwd("/home/lena/Nextcloud/Documents/Master_Data_Science/3_Semester/SAD/48 hours analysis/")

library(rmarkdown)    # for using RMarkdown
library(tidyverse)    # for tidying and cleaning data
library(magrittr)     # for additional pipe operators
library(lubridate)    # for easier handling of timestamp data
library(ggplot2)      # for advanced plotting based on tidyverse
library(ggfortify)    # for more plotting options in ggplot, e.g. PCA objects
library(GGally)       # for nice heatmap plots of correlation matrix
library(grid)         # for more control on plotting in RMarkdown
library(gridExtra)    # for more control on plotting in RMarkdown
library(modEvA)       # for model diagnostics
```

2 Data Import

The dataset is imported, correctly specifying the variable types, as it is indicated in the accompanying README file.

```
# import the data using suitable data types
original_data <- read_csv("building_1319.csv",
                col_types = cols(cloud_coverage = col_factor(),
                                 meter = col_factor(),
                                 timestamp = col_datetime(format = "%Y-%m-%d %H:%M:%S")))
original_data %<>%
  mutate(meter = recode(meter,
                        "0" = "Meter Type 0",
                        "1" = "Meter Type 1",
                        "3" = "Meter Type 3"))%>%
  rowid_to_column(.)

glimpse(original_data)
print(paste("Time goes from",min(original_data$timestamp),"to",max(original_data$timestamp),
            "covering", length(unique(date(original_data$timestamp))),"days."))
```

The data initially contains 26,331 observations of meter readings in kWh, measuring the energy consumption of a building. There are 3 different meter types, thus 8777 observations per meter. The data is timestamped over the period of the year 2016 in hourly intervals, starting on the 1st of January and ending on the 31st of December. The rest of the variables are weather variables, measuring temperature (air and dew, probably in °C), cloud coverage (factor variable), precipation in the 1 hour interval (probably in mm), pressure at sea level (probably in hPa), wind direction (circular, interval 0-360°) and wind speed (probably in m/s).

3 Data Cleaning

3.1 Removing columns that contain no information

It looks like this data is the subset of a larger one. Thus, some variables contain no meaningful information as they have the same values for all rows. They are thus excluded:

```
# function that returns the unique values of a variable
printUniquevalue <- function(variable){
  return (print(paste("Unique value of ", deparse(substitute(variable)), "is:",
                      unique(variable))))
}
# building_id
printUniquevalue(original_data$building_id)
# site_id
printUniquevalue(original_data$site_id)
# primary_use
printUniquevalue(original_data$primary_use)
# square_feet
printUniquevalue(original_data$square_feet)
# year_built
printUniquevalue(original_data$year_built)
# floor_count
printUniquevalue(original_data$floor_count)

# drop these columns because they contain no value for the analysis!
data <- select(original_data, -building_id,-site_id,-primary_use,
               -square_feet,-year_built,-floor_count)
```

There is still some context information about the data to be obtained from these variables: The data in the dataset stems from a site with the ID 14 (variable: `site_id`) and the building ID 1319 (variable: `building_id`), it is a building used for entertainment and public assembly (variable: `primary_use`) having a square footage of 287,419 sq.ft. The year that the building was build (variable: `year_built`) and the number of floors (variable: `floor_count`) are unknown.\

3.2 Identifying Missing Values (NAs)

Often, datasets have missing values. Using `summary(data)`, provides a first impression about the occurrence of missing values.

So which variables have how many missing values?

Air Temperature: 3, Cloud Coverage: 9933, Dew Temperature: 3, Precipitation Depth: 93, Sea Level Pressure: 330, Wind Direction: 870, Wind Speed: 69

This shows that the variable `cloud_coverage` with 9933 NAs has by far the most missing values. As this corresponds to ca. 40% of missing data, this column is dropped. The remaining missing values are

relatively few. I assume that dropping these rows will not change the outcome noticeably. This leaves 25,056 observations as compared to 26,331 before removing them.

```
# variable cloud coverage is excluded
data <- select(data, -cloud_coverage)
# all the rows with any NA value are dropped
data <- drop_na(data)
dim(original_data) # 26331
dim(data) # 25056
```

The boxplots below show that the overall distribution of the target variable, the meter reading in kWh, does not change noticeably when the rows containing missing values are removed.

```
# Are the meter readings different when NAs are removed?
boxplot_original <- ggplot(data = original_data, aes(meter, meter_reading)) +
  geom_boxplot() +
  ggtitle("Original Data") + xlab("Type of Meter") + ylab("Meter Reading [KWh]")

boxplot_noNAs <- ggplot(data = data, aes(meter, meter_reading)) +
  geom_boxplot() +
  ggtitle("Data after NAs were dropped") + xlab("Type of Meter") + ylab("Meter Reading [KWh]")
grid.arrange(boxplot_original, boxplot_noNAs, ncol = 2)
```

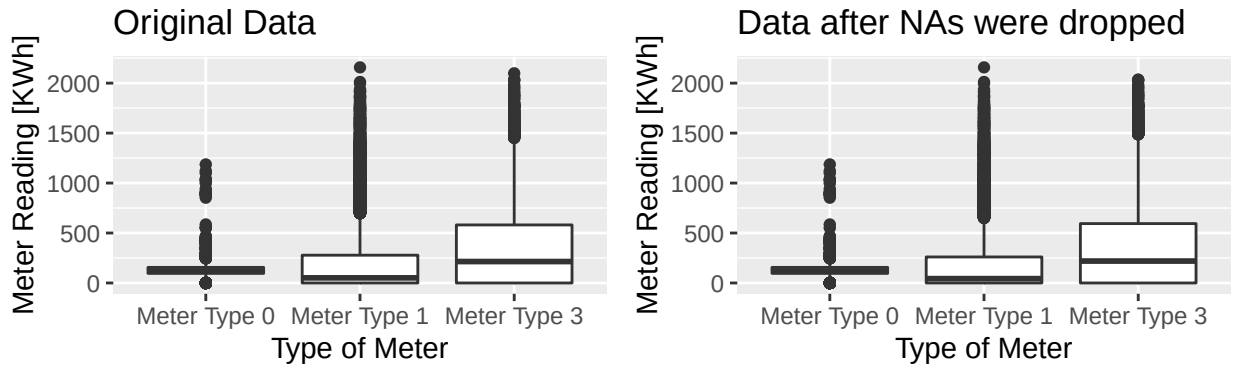

Figure 1: Comparing the distribution of the target variable before and after removing NAs.

3.3 Replacing Negative Precipitation Values

Besides the missing values, the summary of the data also showed that the precipitation variable has -1 as its minimum value. This is most likely a measurement error, as the precipitation measured (within one hour) can only have positive values. Therefore the -1 values are replaced with 0.

```
summary(data$precip_depth_1_hr)
# There are 1323 rows where precipitation is negative
select(filter(data, precip_depth_1_hr == -1), precip_depth_1_hr)
# change these values!
data %<>%
  mutate(precip_depth_1_hr = case_when(
    precip_depth_1_hr == -1 ~ 0,
    TRUE ~ precip_depth_1_hr))

# Check the result
filter(data, precip_depth_1_hr == -1)
```

4 Data Exploration - What are interesting observations?

4.1 Are there different patterns in the meter readings for different meter types?

There is no information about the type of meter available, so what can the data tell us about the meter types? Plotting the meter readings for all meters as a simple time series plot gives a first intuition. This plot shows that the meters are most likely measuring different things. For instance, the meter of type 1 has the highest recordings in the winter months, while the meter of type 1 has its highest recordings during the summer months. This might indicate that meter 3 shows the energy used for heating, which is more required in winter, and meter 1 measures energy used for electricity e.g. air conditioning which is more required in summer.

```
# plot of energy consumption grouped by meter
ggplot(data, aes(x = timestamp, y = meter_reading,
                 group = meter,
                 colour = meter)) +
  geom_line() +
  xlab("Time") +
  ylab("Meter Reading [KWh]")
```

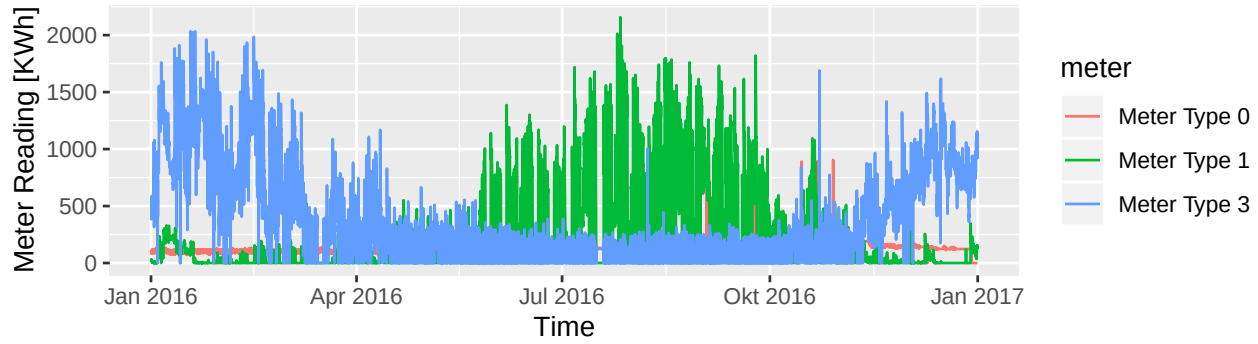

Figure 2: Time series plot of the meter readings

The following plot shows the frequency plots for the meter readings differentiated by the meter type. This shows the differences in distributions, the high prevalence of values equal to zero for meter type 1 and 3 and the non-normal distribution for all meter types.

```
ggplot(data, aes(x = meter_reading)) +
  geom_histogram(bins = 100) +
  facet_grid(~meter)  + xlab("Meter Reading [KWh]") + ylab("Count")
```

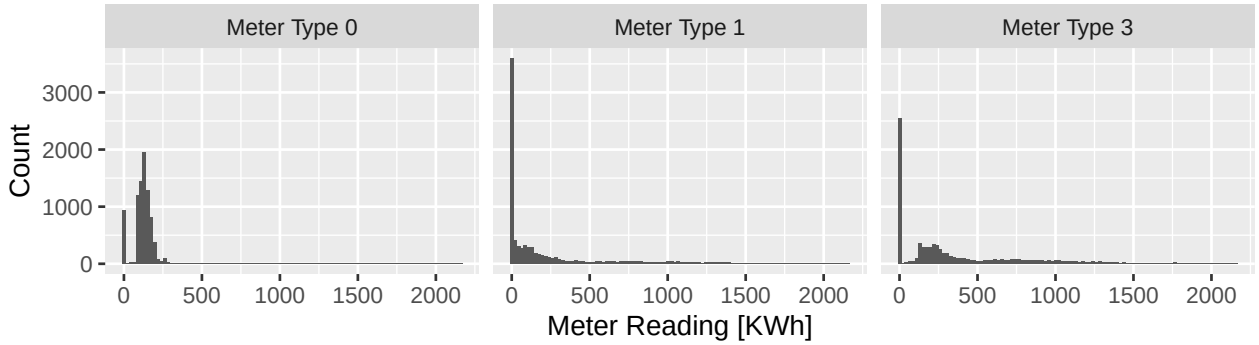

Figure 3: Frequency plots for the meter readings, distinguished by meter type.

4.2 Are some of the weather variables measuring very similar things?

4.2.1 Correlation Matrix

As a first intuition for similarity in the weather variables, I look at correlation. This is done using the Pearson correlation as all the variables are continuous.

```
cor_mat <- cor(select(data, meter_reading:last_col()), method = "pearson")

# plot a heatmap from correlation values
ggcorr(data = NULL, cor_matrix = cor_mat, label = TRUE,
       label_size = 2.5, size = 3, label_round = 2, hjust = 1, layout.exp = 3,
       label_color = 'black', low = 'darkred',  mid = 'white', high = 'darkgreen')
```

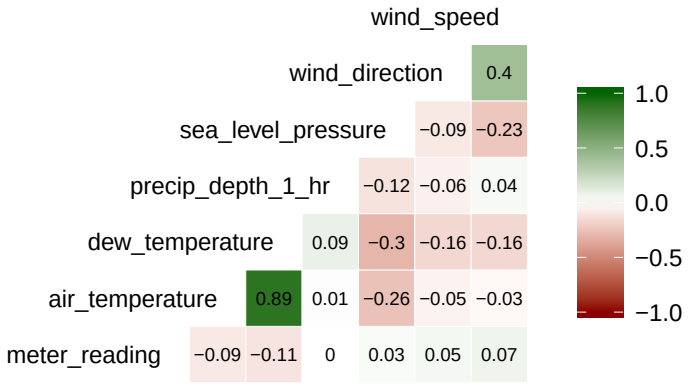

Figure 4: Heatmap of the correlation values between the continuous explanatory variables.

Overall, there are very few high correlations present. The highest correlation exists between air and dew temperature with a value of 0.89. The second-highest correlation of 0.4 is between wind speed and wind direction.

4.2.2 Principal Component Analysis (PCA)

A Principal Component Analysis (PCA) gives information about similarity between variables, as it is a tool commonly used for discovering structure in data and then reduce the dimensionality based on it. Here it is used to look at redundancy in the explanatory weather variables, therefore the time variables, response variable (meter reading) and meter type are excluded.

```
# PCA of all explanatory variables using all data
pca_all <- prcomp(select(data, air_temperature:last_col()), scale = TRUE)

autoplot(pca_all, x = 1, y = 2,
         colour = 'grey', xlim = c(-0.02,0.02), ylim = c(-0.02,0.02),
         loadings = TRUE, loadings.label = TRUE, loadings.label.size = 3,
         loadings.label.repel = TRUE,  loadings.label.vjust = 2,
         loadings.label.colour = 'black', loadings.colour = 'black') +
  theme_bw()
```

The PCA indicates that the variables dew temperature and air temperature are similar, as well as wind direction and wind speed. Arrows that are orthogonal to each other indicate dissimilarity, meaning that they explain different aspects of the data. This is the same pattern found by the correlation as shown in figure 4. Looking towards modeling, variables orthogonal to each other have a higher predictive power for the variance in the data than those close together. As all variables were scaled to unit variance, the variance can also be

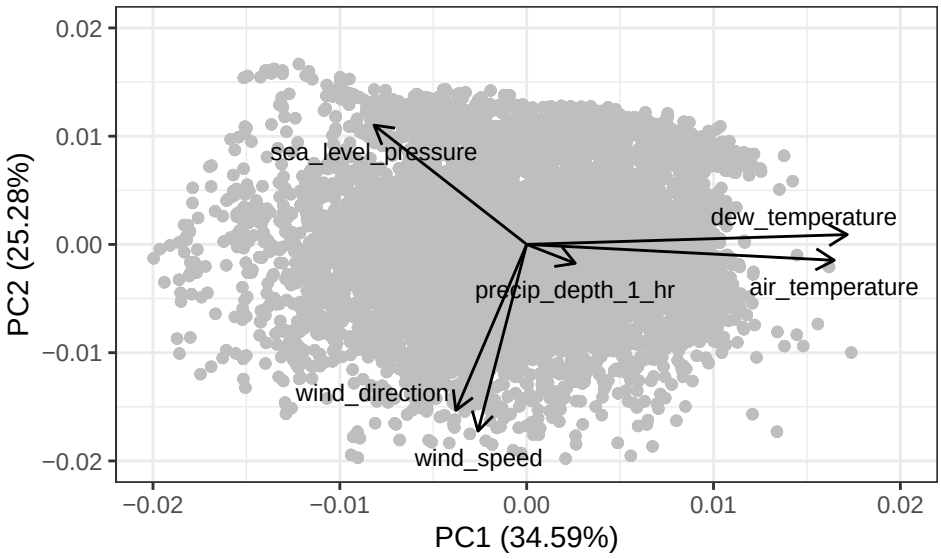

Figure 5: Biplot of the principal component analysis (PCA) including all continuous explanatory variables.

compared. The degree of variance of a variable is indicated by the length of its respective vector. This shows that the precipitation variable has quite low variance compared to the other variables.

Overall, it looks like the two temperature variables show a very similar behaviour, as well as the two wind variables.

4.3 Are there generally changes in energy consumption, based on the month, the weekday and the time of day?

To be able to answer this question, I need to extract more specific information from the timestamp. To that end, three new factor variables are created: The month variable has 12 levels, corresponding to the 12 months of the year. The weekday variable has 7 levels, corresponding to the 7 days of the week. The TimeOfDay variable has 4 levels, each one corresponds to a quarter (= 6 hours) of the day, defined as night (24h - 5h), morning (6 - 11h), afternoon (12 - 17h) and evening (18 - 23h).

```
# adding month + weekday + time of day variable as factors
data %<>%
  mutate(month = as_factor(month(timestamp))) %>%
  mutate(weekday = as_factor(wday(timestamp))) %>%
  mutate(TimeOfDay = case_when(
    hour(timestamp) >= 0 & hour(timestamp) <= 5 ~ "night",
    hour(timestamp) >= 6 & hour(timestamp) <= 11 ~ "morning",
    hour(timestamp) >= 12 & hour(timestamp) <= 17 ~ "afternoon",
    hour(timestamp) >= 18 & hour(timestamp) <= 23 ~ "evening",
    TRUE ~ "NA")) %>%
  mutate(TimeOfDay = factor(TimeOfDay,
                            levels = c("night", "morning", "afternoon", "evening"))) %>%
  select(rowid, meter, meter_reading,
         timestamp, TimeOfDay, weekday, month,
         air_temperature, dew_temperature, precip_depth_1_hr,
         sea_level_pressure, wind_direction, wind_speed)
```

So let's look at some boxplots that differentiate by these factors. Due to the brevity of the report, I'll focus on meter type 1.

```r
# by time of day
energy_TimeOfDay <- ggplot(data = filter(data, meter == "Meter Type 1"),
                           aes(x = TimeOfDay, y = meter_reading)) +
  geom_boxplot() + xlab("Time of Day") + ylab(element_blank())
# by weekday
energy_weekday <- ggplot(data = filter(data, meter == "Meter Type 1"),
                         aes(x = weekday, y = meter_reading)) +
  geom_boxplot() + xlab("Day of the Week") + ylab("Meter Reading [KWh]")
# by month
energy_month <- ggplot(data = filter(data, meter == "Meter Type 1"),
                         aes(x = month, y = meter_reading)) +
  geom_boxplot() +  xlab("Month")  + ylab(element_blank())

grid.arrange(energy_TimeOfDay, energy_weekday, energy_month, nrow = 3)
```

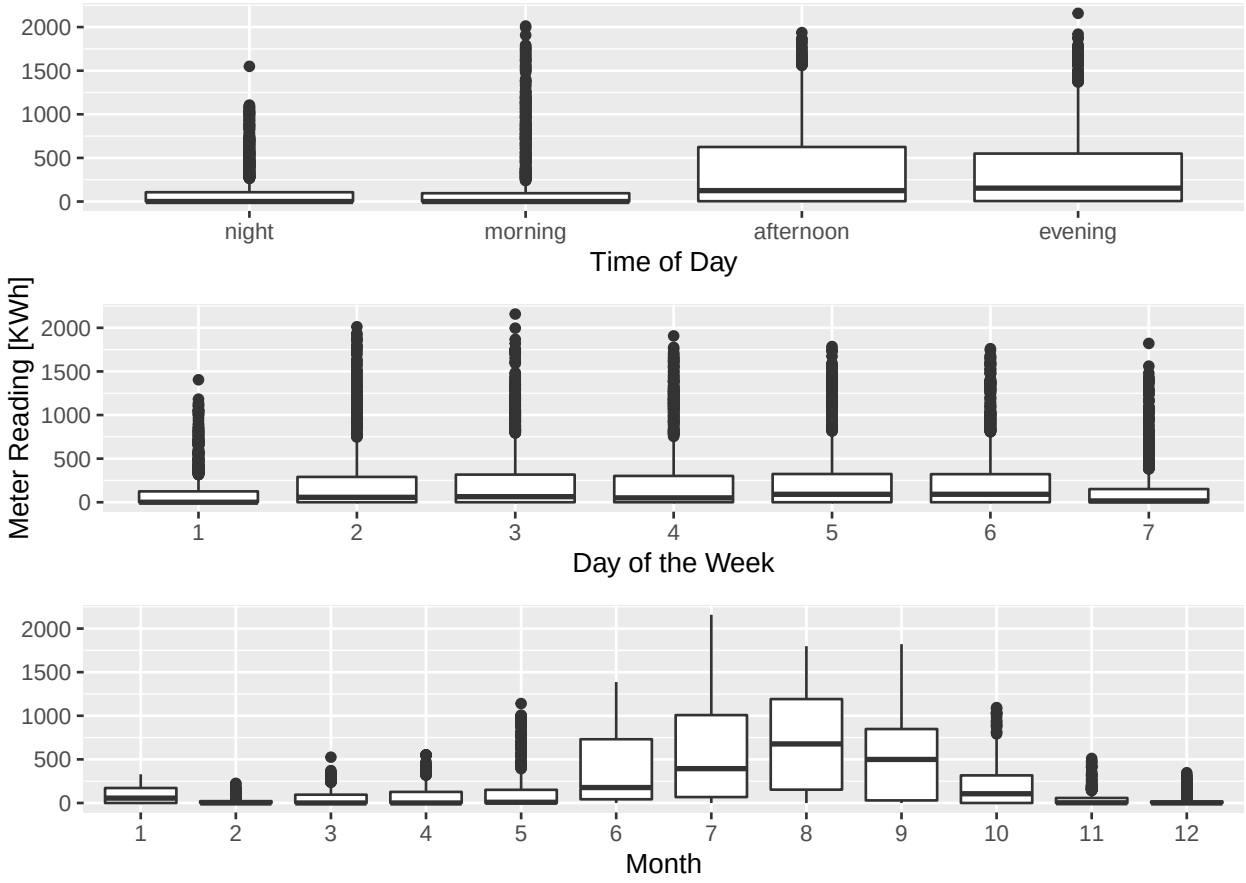

Figure 6: Comparison of energy consumption measured by meter of type 1 based on the time of day, weekday and month.

The plot shows that the energy consumption as measured by meter 1 measures noticeably by the time of day and month, but not so much by the day of the week. The building from which the data is obtained needs considerably more energy during the summer from June to September and in the afternoon and evening time. This might be due to its primary use, remember that is used or entertainment and public assembly. Event of this type occur generally more often in the evening than earlier on the day.

5 Modeling Energy Consumption

What are the variables having the highest predictive power for the energy consumption?

As the data is not normally distributed (shown by figure 3) and there are categorical variables that should be included as explanatory variables, I will use a generalized linear model (GLM) for modeling the response variable meter reading. As in the previous chapter, the analysis only considers data measured by meter type 1. The Gaussian error structure with the default link function is used. All continuous explanatory variables are scaled to account for different units. The models do not consider interactions between variables.

```
# this library is loaded here, because an error is thrown if loaded at the top
library(car)          # for model diagnostics

data_glm <- data %>%
  filter(meter == "Meter Type 1")
attach(data_glm) # make column names directly accessible

# model including all explanatory variables
add_all <- glm(meter_reading ~ TimeOfDay + weekday + month +
                 scale(air_temperature) + scale(dew_temperature) +
                 scale(precip_depth_1_hr) + scale(sea_level_pressure) +
                 scale(wind_direction) + scale(wind_speed),
               family = gaussian(link = "identity"))
# check model statistics
summary(add_all)
# check the variance inflation factor
vif(add_all)
# check pseudo R² values
RsqGLM(add_all)

# reduced model (non-significant variables removed)
add_reduced <- glm(meter_reading ~ TimeOfDay + weekday + month +
                     scale(air_temperature) + scale(sea_level_pressure) +
                     scale(wind_direction), family = gaussian(link = "identity"))
summary(add_reduced)
vif(add_reduced)
RsqGLM(add_reduced)

# model using only the weather variables
add_weather <- glm(meter_reading ~ scale(air_temperature) +
                   scale(precip_depth_1_hr) + scale(sea_level_pressure) +
                   scale(wind_direction) + scale(wind_speed),
                 family = gaussian(link = "identity"))
summary(add_weather)
vif(add_weather)
RsqGLM(add_weather)

# model using only the time variables
add_time <- glm(meter_reading ~ TimeOfDay + weekday + month,
                family = gaussian(link = "identity"))
summary(add_time)
vif(add_time)
RsqGLM(add_time)
```

```r
# store main results in a dataframe for each model
results_glm <- tibble(model = "add_all", AIC = 116922,
                      null_deviance = 1195038453, residual_deviance = 583753436,
                      Nagelkerke = 0.51,
                      vif = "too high for month + air temp + dew temp") %>%
  add_row(model = "add_reduced", AIC = 116939,
          null_deviance = 1195038453, residual_deviance = 585358759,
          Nagelkerke = 0.51, vif = "too high for month + air temp") %>%
  add_row(model = "add_weather", AIC = 119427,
          null_deviance = 1195038453, residual_deviance = 791911009,
          Nagelkerke = 0.34, vif = "all ok") %>%
  add_row(model = "add_time", AIC = 117228,
          null_deviance = 1195038453 , residual_deviance = 606385000,
          Nagelkerke = 0.49, vif = "all ok")
```

Details from the results of the fitted GLM can be extracted from the respective summaries contained in the accompanying .Rmd file. In the following, an overview over the results is given.

The summary from the "add_all" model show the three time variables (TimeOfDay, weekday, month) as highly significant. In contrast, multiple of the weather variables (precipitation and both wind variables) are not significant ($p > 0.001$). This model can explain 51% percent of the variance in the response variable, based on the Pseudo-R^2 value (Nagelkerke). The results for the "add_reduced" model are very similar (Pseudo R^2 (Nagelkerke) = 51%), despite having excluded the variables dew temperature, precipitation and wind speed based on the observations above.

For the weather model, the dew temperature variable is removed due to high correlation with the air temperature variable, as indicated by the variance inflation factor check. This model has an R^2 (Nagelkerke) of only 0.34. The model "add_time" using only the factorial time variables has an R^2 (Nagelkerke) of 0.49, indicating that the time variables are generally explaining a higher proportion of the variance in the data.

The models can be further improved and compared using candidate modeling based on the AICc and the analysis of more model diagnostics (residual plots).

To draw a preliminary conclusion to the question asked in this subchapter: It seemes like the variables based on time explain more variance in the meter readings than the weather variables.

6 Concluding Thoughts on the Data

Above it has been shown with exploratory plots that the energy consumption of the building under analysis depends largely on the month and time of day. Correlation plots and a PCa were able to show the similarity of some of the weather variables.

Considerably more time can be spent with the formulation and optimization of predictive models. The suitability of the chosen model formulation needs to be further evaluated using more model diagnostics, as well as a theory-driven approach.