Monthly Electric Consumption

Kevin Eng

4/28/2020

Introduction

The goal of this analysis is to explore factors which influence electric consumption at buildings within New York City. To accomplish this, we collect three data sets: one which describes the surrounding weather, one which measures a buildings enery consumption, and one which desribees the physical characteristics of the building.

Ideally, in order to seperate electric consumption due to seasonal trends, hourly data would have been preferable. Indeed the inductive basis for making conclusions with high temporal resolution data would be stronger since we would not have to worry about confounding with say energy differences due to holidays. Alas the highest resolution data set that was found was monthly.

Data

Historical weather data was collected off NOAA's database. Fortunately, they have a simple API which allows for quick querying of basic weather data. A particular aspect of their database is that NOAA only offers access to select daily weather summeries. Hourly, and monthly averages are only given as "normal" averages which the NOAA defines as 30-year averages. The observations include daily minimum temperature, maximum temperature, perciptation, snow fall, and average wind speed.

Information on the physical characteristics of a collection of building complexes are recorded in the NYCHA data set. The dataset also contains a number of administrative details such as whether or not it is a senior development. In database venacular the primary key of this data set is the TDS number. Since the TDS number refers to a building complex is it not possible to identify sub-units within a complex.

Details on the monthly energy consumption can be found in the NY open data website. the website claims the data set only contains readings from 2010 up to March 2019. This is not quite true since in reality it contains readings up to September 2019. The monthly readings are given for an individual meter. This means that there are several readings for a given TDS number since each sub-unit within a complex has its own meter.

Data Processing

All three data sets needed some preprocessing in order to get into tidy form. Since the electric consumption measurements are given on a monthly basis, daily weather readings must be aggregated into monthly data. This can be accomplished by extracting the month and and year from each date record using the handy `year()` and `month()` function from the `lubridate` package. Additionally, it would also nice to have some notion of the typical monthly temperature. We can estimate this by computing the median of the monthly average maximum and minimum temperature using purr's `map()` function.

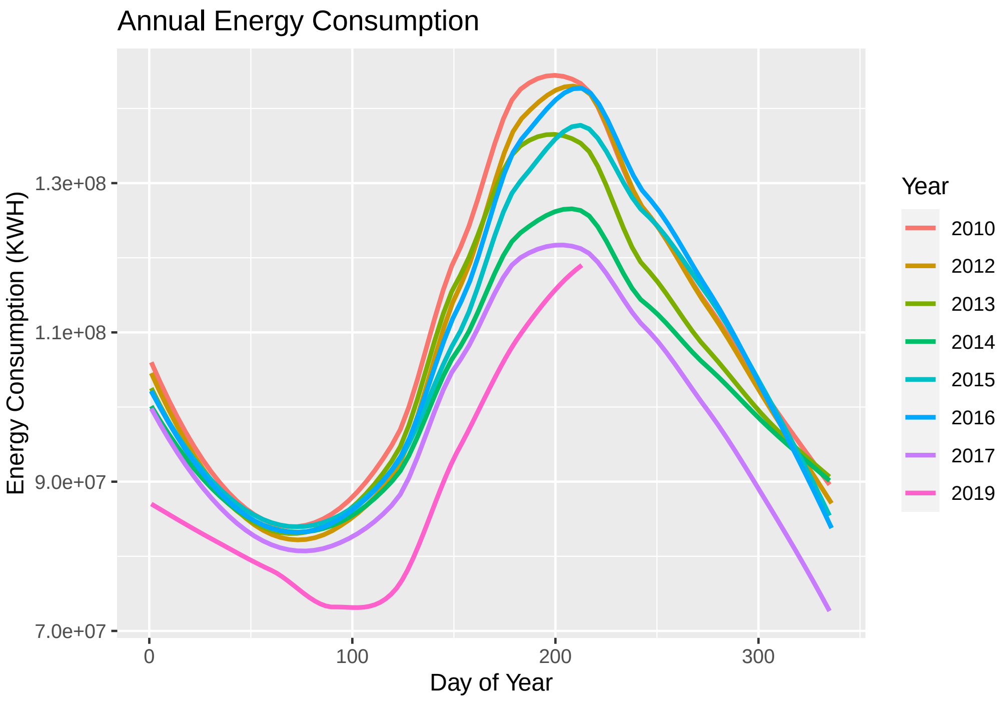
The NYCHA data set contains a great deal of unwanted administrative details. And as we will see, the data is also non-tidy as some rows contain multiple observations. On a somewhat cosemtic level, the naming convention for the columns is overly verbose and makes for combersome data referencing. We can easily fix these issues using dplyr's `select()` and `rename()` functions. There `TDS` column is somewhat problematic because it is a source of multiple observations and possible data entry errors. It likely that some building designs were reused so several apartment building share the same physical makeup on paper. In regards to possible data entry error, several rows contain `*`'s. To seperate the multiple observations we can use the `separate_row()` function which allows us to split a row into multiple rows based on a delimiter. The `*`'s can be easily dealt with using `str_replace()`. The last issue concerns the column that specifies the number of stories in a complex. Since we do not have the data resulution to seperate sub-units within building complexes we must find a way to reduce the number of recorded stories into one value. An easy approach is to take the mean of all the stories. This can be accomplished by first splitting the recording using `str_split` and then using `map_dbl` to produce the mean.

The electric consumption data, like the weather data, provides higher resolution data then can be used. In order to conform to the NYCHA data, total electric consumption and costs must be grouped by TDS which can be done in a similiar fashion as the weather data. Minor data issues were present where several rows contained missing TDS values. These were easily dropped from the table using `drop_na`.

Finally all three data sets were merged using `inner_join`'s. The weather data was joined to the electricity data using the date. Next, the resulting merged table was joined with the NYCHA data using TDS. Because we used `inner_join`'s the final table is complete and because we set each table to the same "scale" the final table is tidy.

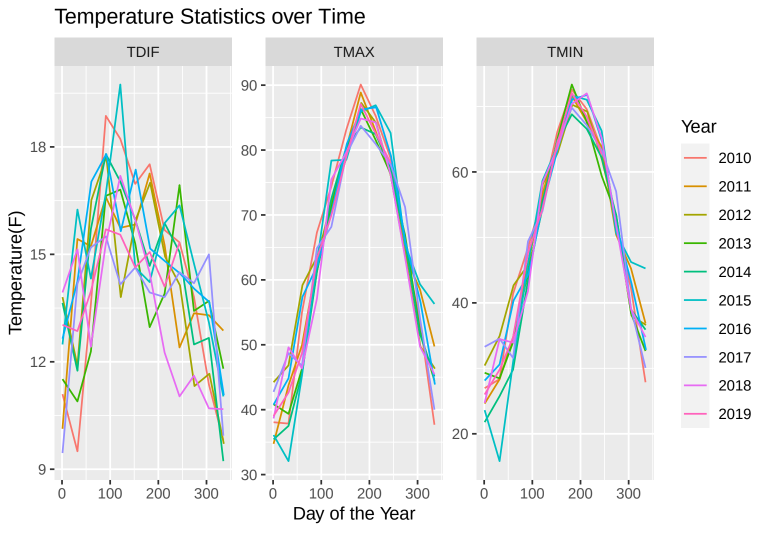
Exploratory Data Analysis

To get a rough idea what what annual energy consumption over time looks like, we group the data by data by year and month and compute the total energy consumption. From there we can create a plot of supper imposed monthly energy consumption curves over several years.

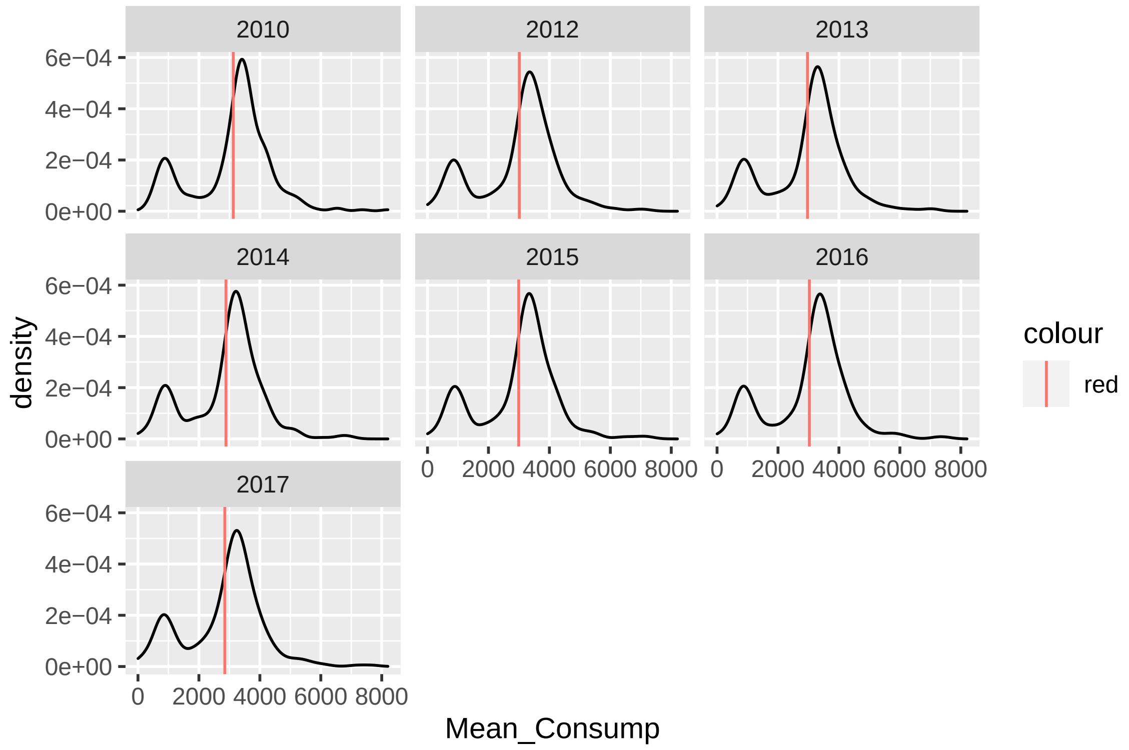


The visualization shows a striking downward trend in enery consumption. In an optimistic take one might attribute the apparent decrease in electrical consumption to more energy efficient technologies.

In exploring the weather data, an interesting question is whether not it supports the notion of seasons arriving "late" or "early". In other words, we want to see whether or not ground hogs should be afraid of their shadow.



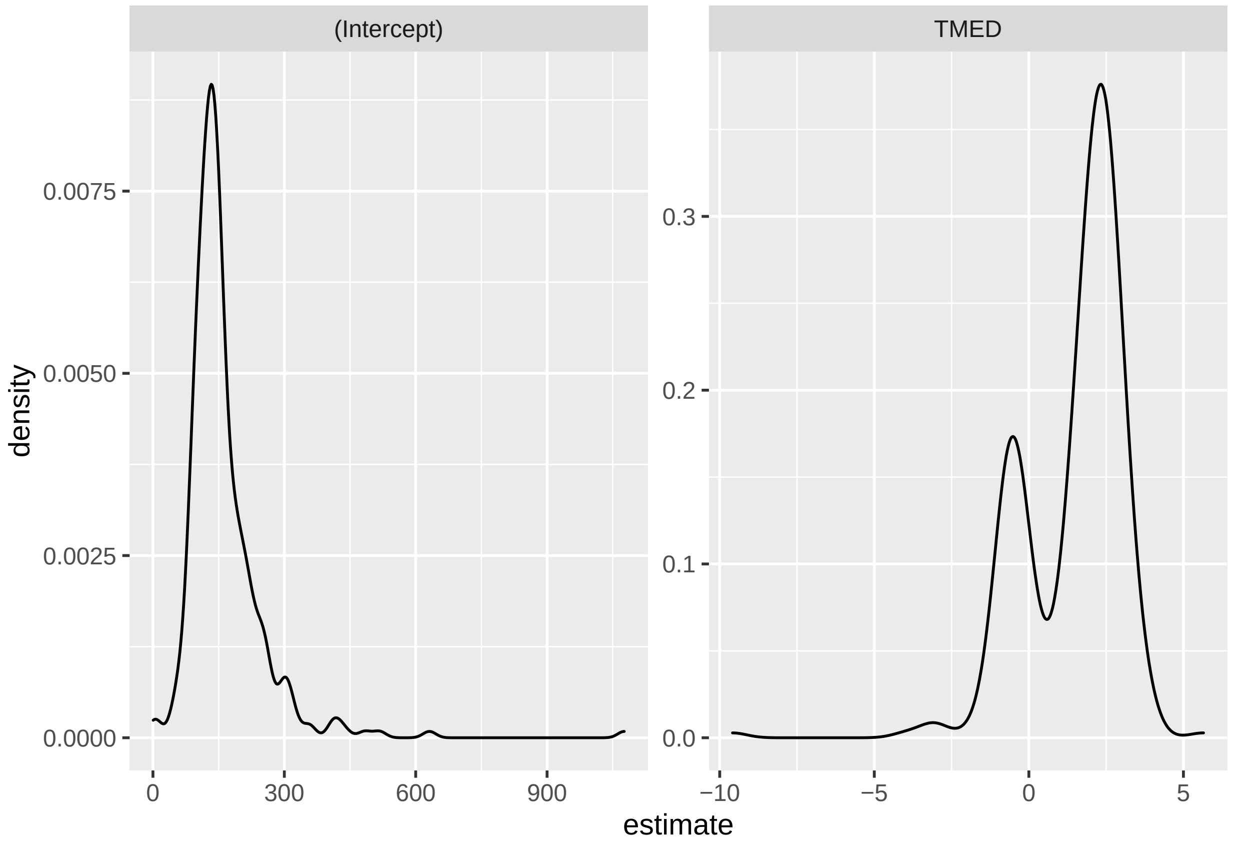
To explore the validity of this narrative we can calculate distribution of annual energy consumption per person by building



Calculate simple linear mixed model and show the distributions of beta.

```
consumption_model <- function(data) {
    lm(PP_KWH_CONSUMP ~ TMED, data = data)
}

full_data %>% mutate(PP_KWH_CONSUMP = KWH_CONSUMP / TOTAL_POPULATION) %>%
    group_by(TDS) %>%
    nest() %>%
    mutate(model = map(data, consumption_model)) %>%
    mutate(summary = map(model, tidy)) %>% unnest(summary) %>%
    ggplot(aes(estimate)) +
        geom_density() +
        facet_wrap(vars(factor(term)), scales = 'free')
```


(Intercept)
TMED
density
estimate