Spain - Electricity Use Related to Temperature

Justin McQuown

May 2024

Contents

1 Introduction

For the choose your own project I chose to try to predict the daily energy use in MWh based on the daily temperature for the country of Spain. The energy use data set came from kaggle and can be found here: https://www.kaggle.com/datasets/pythonafroz/price-of-electricity-and-the-renewable-energy?resource=download. While the license info can be found here: https://creativecommons.org/licenses/by-nc-sa/4.0/. The energy data was was taken in Spain for 2 year period starting on November 1, 2020 and ending on October 31, 2022. The data was collected multiple times an hour giving a fairly decent size dataset of over 100K entries. Temperature data for this time period wasn't available on kaggle so it was gathered from NOAA. The NOAA data was not sampled as frequently and was taken from multiple stations across the country. It initially contained over 260K entries but this was trimmed down several times throughout the analysis. Since both kaggle and NOAA require accounts to download the data, the data sets are provided in the github repository. They are named energyData.csv for the energy data from kaggle and tempData.csv for the temperature data from NOAA.

With the goal of this project being able to predict the energy use based on the daily temperature a validation set of data named testData was generated and is 10% of the original data. The testData was not used to create the various models but was used in the final validation of the algorithms. Several techniques of models were used in an attempt to make the predictions as well as some data engineering to clean the data.

1.1 Loading the data

The data for the project is in two .csv files that are included in the git repository. The files are imported and converted to English style format for decimals (i.e. period is the decimal marker and not a comma) along with a few other formating changes to aid in the analysis. The workspace data in this case was saved as chooseYourOwnWorkspace.Rdata for easy importing into the markdown file. I find it annoying to have to have to re-run the code within the markdown file so it is saved with the git repository as well. This is just a reminder if anyone would want to run the knitr on the markdown file that you will need include the data .csv files and the workspace variable file .Rdata in the same folder as this .Rmd for it to work properly.

1.2 Initial data exploration

In an effort to get a better sense of the data and important features some basic statistics were taken and some plots were generated that may help provide some insight into what techniques are best for this data. Here's a snip of the data set used throughout to get a general idea of the data format and important information.

```
## # A tibble: 654 x 5
##    date        tMax  tMin   eDaily  tAVE
##    <chr>      <dbl> <dbl>    <dbl> <dbl>
##  1 2020-11-01  69.4  46   3259974   67.6
##  2 2020-11-02  69    47.6 3670105.  69.2
##  3 2020-11-03  59.4  50.4 4023419.  72.0
##  4 2020-11-04  53.6  45.8 4159524   67.4
##  5 2020-11-05  56.6  47.4 4135402.  69.0
##  6 2020-11-06  66.4  50.4 4051457.  72.0
##  7 2020-11-07  59.8  46.6 3608911.  68.2
##  8 2020-11-08  60.8  45.2 3324562.  66.8
##  9 2020-11-09  60.4  47.8 3918629.  69.4
## 10 2020-11-11  59.8  41.6 4095811.  63.2
## # i 644 more rows
```

As you'll see, the relevant data columns from the two input files has been cleaned and joined to simplify the data engineering that will take place next. The important columns for this project are the eDaily column

which is the daily energy use, tMax which is the daily max temperature, tMin which is the daily minimum temperature, the other columns were used in initial analysis but were deemed not important after testing various models. They are left in the data frame largely because it really seemed like they would matter at some point but it turned out not to be the case. The eDaily column is in units of MWh while the temperatures are in Fahrenheit even though the data was from Spain, it was easier for me personally to visualize the data in this manner.

2 Data Engineering

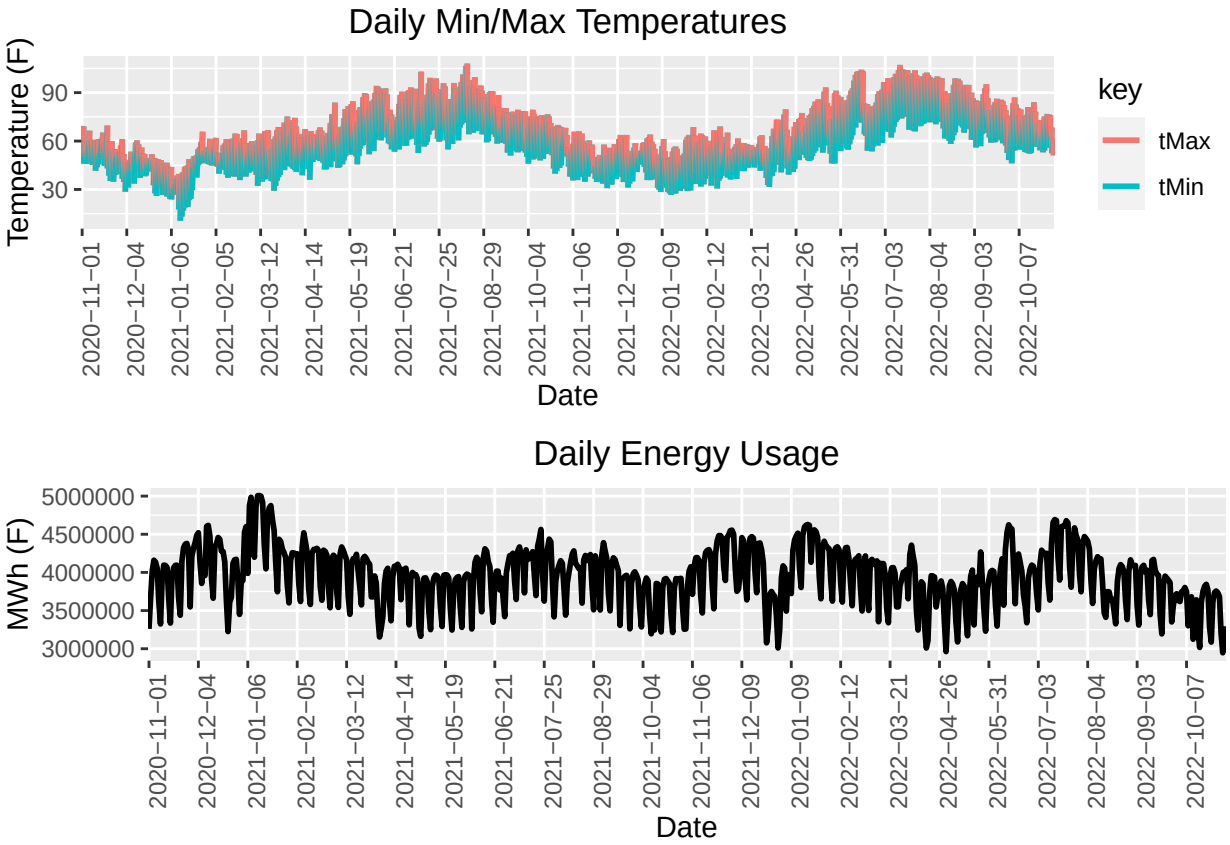
The energy data contained a lot of information including energy total, reference and free market price, reference and market share, and renewable generation source. For this application the price, share, and source were not relevant so they were removed after the initial import leaving just the total energy. The temperature data consisted of the daily high, low, and average temperatures along with the station it was collected, the coordinates of the station and the elevation of the station. In an effort to simplify the number of data points I filtered out all the stations except the ones near Madrid since it was a fairly centralized location within the country. There were a number of stations located in and around Madrid and the daily temperature data was averaged for these locations. Since the energy data was sampled every 4 hours and the temperature data was collected once a day, the energy data was summed to give a single measurement for the day.

2.1 Data Exploration

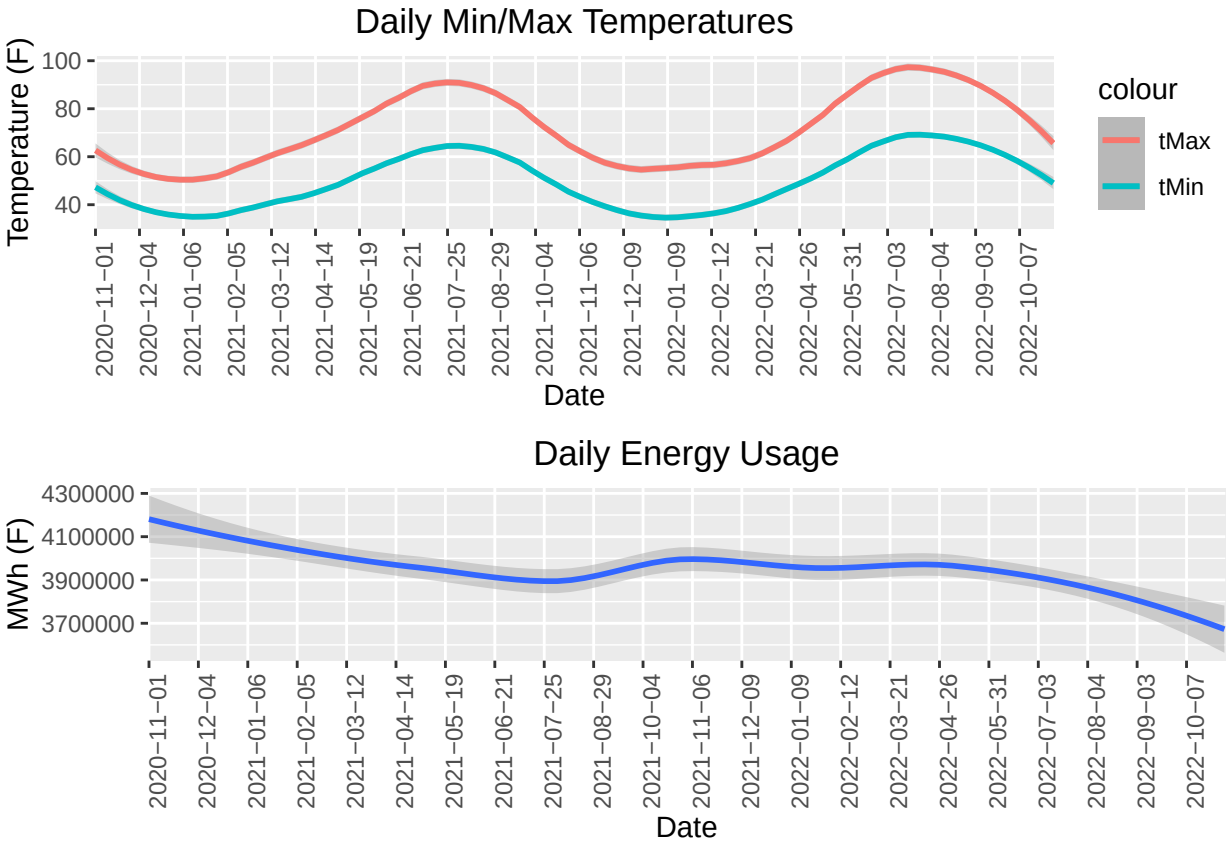
The data exploration started with looking at the basic statistics of the dataset. They are displayed in the table below:

	N	Ave	Median	STDev	High	Low
Max Temp	654	71.24	68.8	16.56	108	32.6
Min Temp	654	49.64	48.6	12.87	77	10.6
Energy Use MWH	654	3958352.91	3973245.9	379254.75	5006539	2947056.6

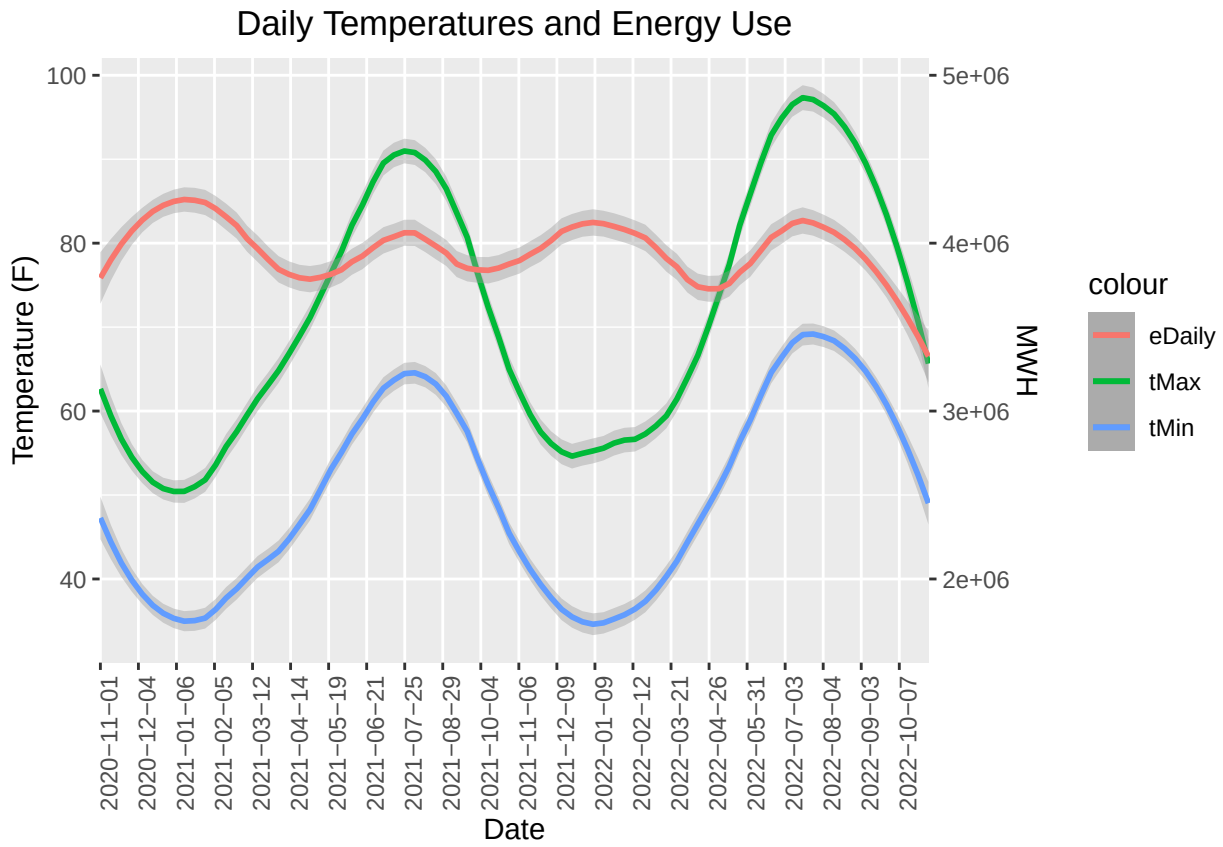
The table in and of itself doesn't provide a whole lot of useful information but you can see that the temperatures are within a reasonable range at least gives a little confidence that the data is realistic. Next, a few plots of the data will be displayed. The first set of plots are the daily Min/Max temperatures and the energy use in MWh vs. the date. You'll notice that this a two year plot so there is definitely a periodic nature of the temperature, which is expected. The daily energy use has some periodic trends but it is less obvious from this plot.



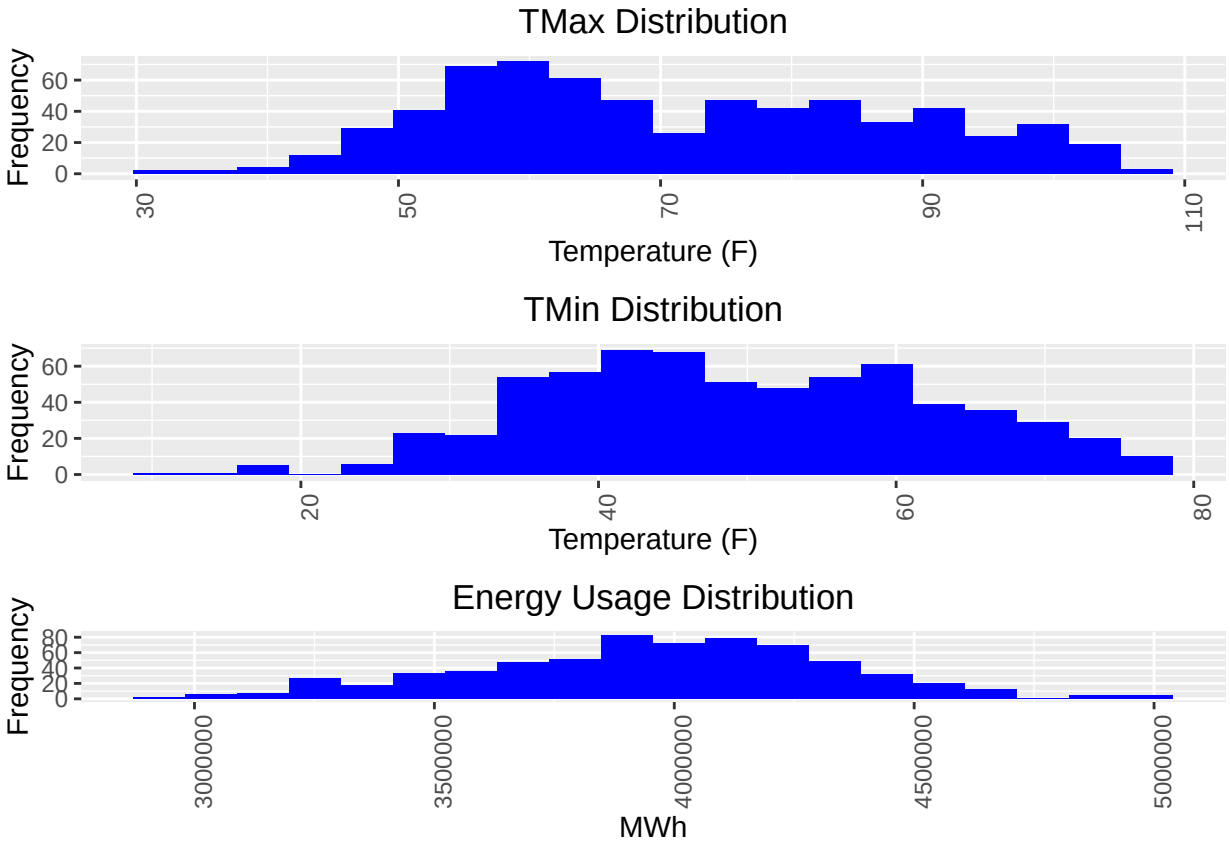
The next set of plots are the smoothed version of the previous plots using the loess method and a span of .3, you'll notice that the temperatures for 2022 seemed to generally warmer than 2021 but that also that lower temperatures seem to show an increase in energy use.



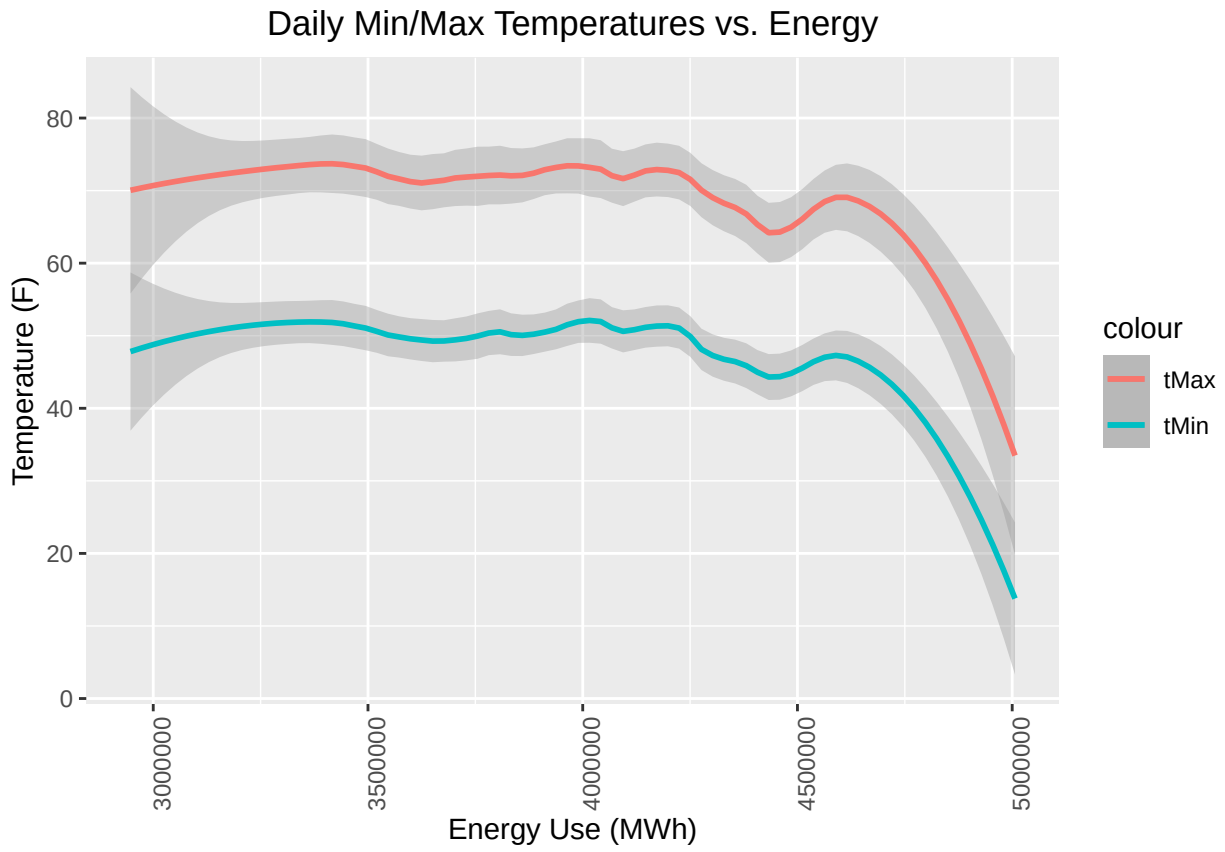
The following plot is the same as the previous except with the temperature and energy use put on the same x-axis just for completeness sake. There isn't a whole lot of new information here but it is just a different way of displaying the previous data. It does show the periodicity a bit better and there seems to be a correlation between the extreme temperatures and energy use with the extreme lows having a larger effect than the extreme highs.



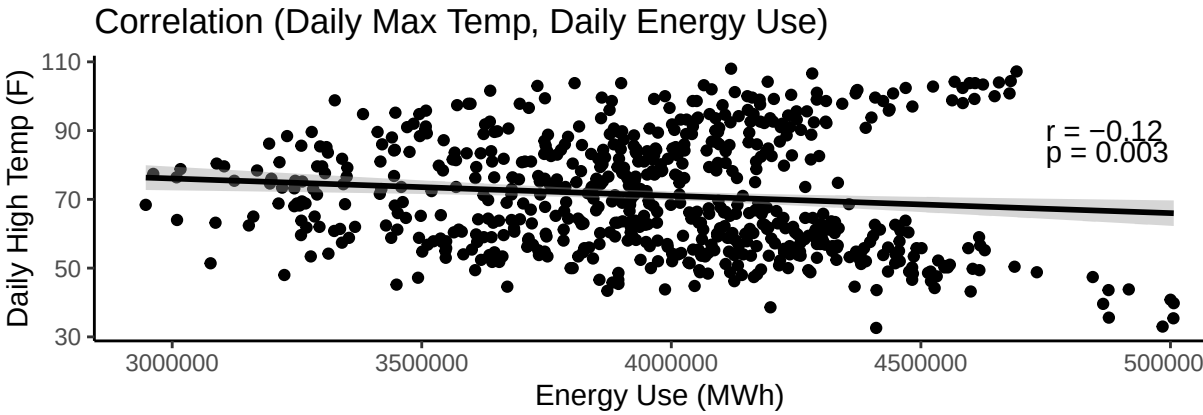
Just out of curiousity the histogram of the data was plotted to see if there's anything that can be of use.

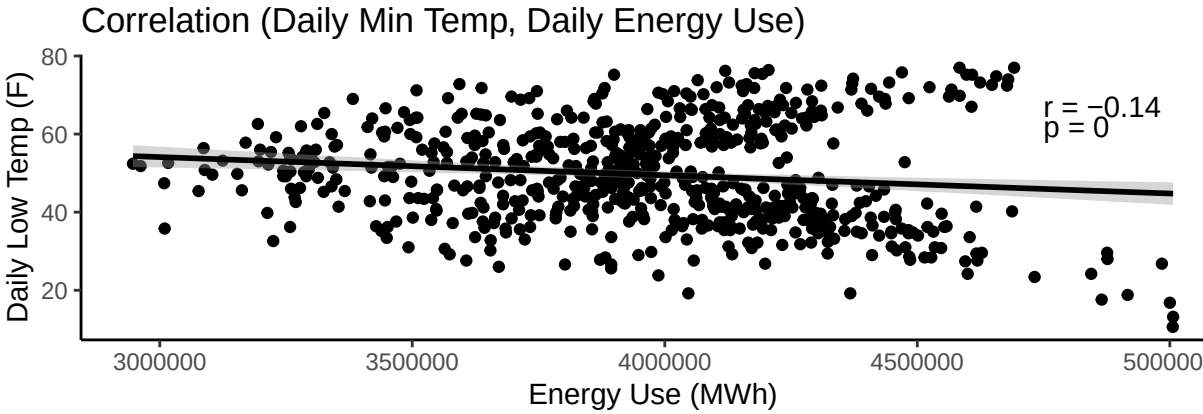


If you squint you can see a normal distribution of data on the temperature min and energy use with a bi-modal distribution of the temperature max. From the plots of the histogram I don't really see much useful information. The next plot is the temperature max/min vs. the daily energy use where the correlation between the extreme temperatures and the energy use use is becoming more apparent.

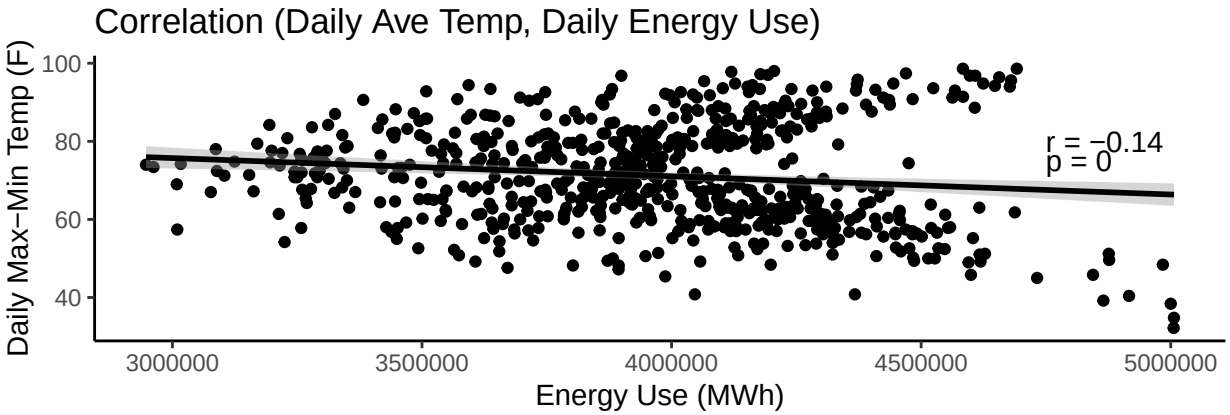


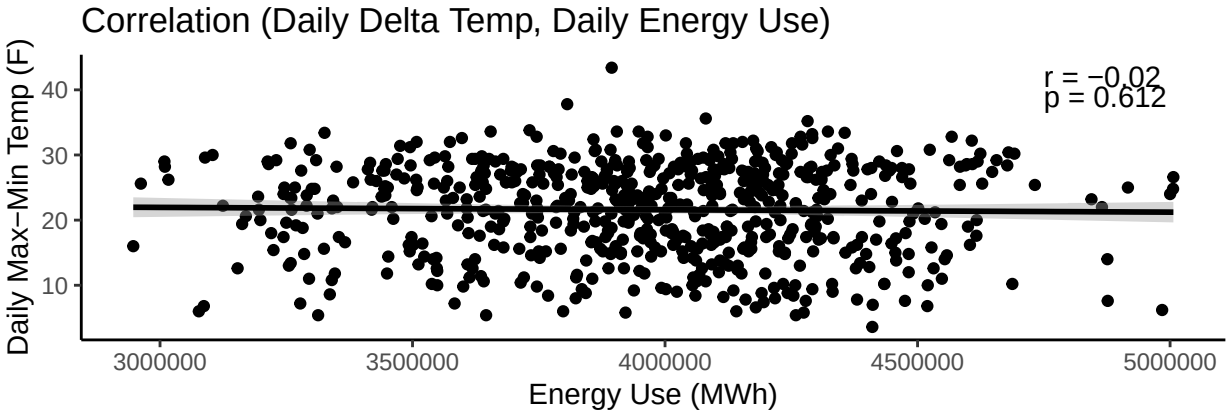
The final set of plots are correlation plots of the three temperature values, max, min, delta between max/min





and the energy use.





The correlation plots seem to show a slightly negative correlation. It is actually lower than I was expecting based on the plots. The delta temperature (tMax - tMin) showed almost no correlation and the average daily temperature resulted in the same correlation as the minimum value so I'm not going to bother using these in any future analysis. Spending some more time with the data and maybe adding additional predictors could probably increase the correlation within these visualizations.

3 Analysis

All of the algorithms tested were executed with the caret train function and executed using the daily energy use as the outcome and the daily temperature max/min as the predictors. The models were run with other predictors such as the daily average temperature, the delta between the max and min temperatures, and the max/min temperatures individually but all of these versions of the model performed worse than the combined version so the data is not discussed here.

3.1 Analysis Precursor

The metric for determine which algorithm performed best was the RMSE but as you'll see later in the report other metrics were used as a comparison, partly of out of my personal interest but also because the data was continuous with a large variation and bias. During the analysis this made it difficult to quickly glance at the error value for an indication of how the algorithm was performing so a percent error measurement was added.

3.2 Initial Models

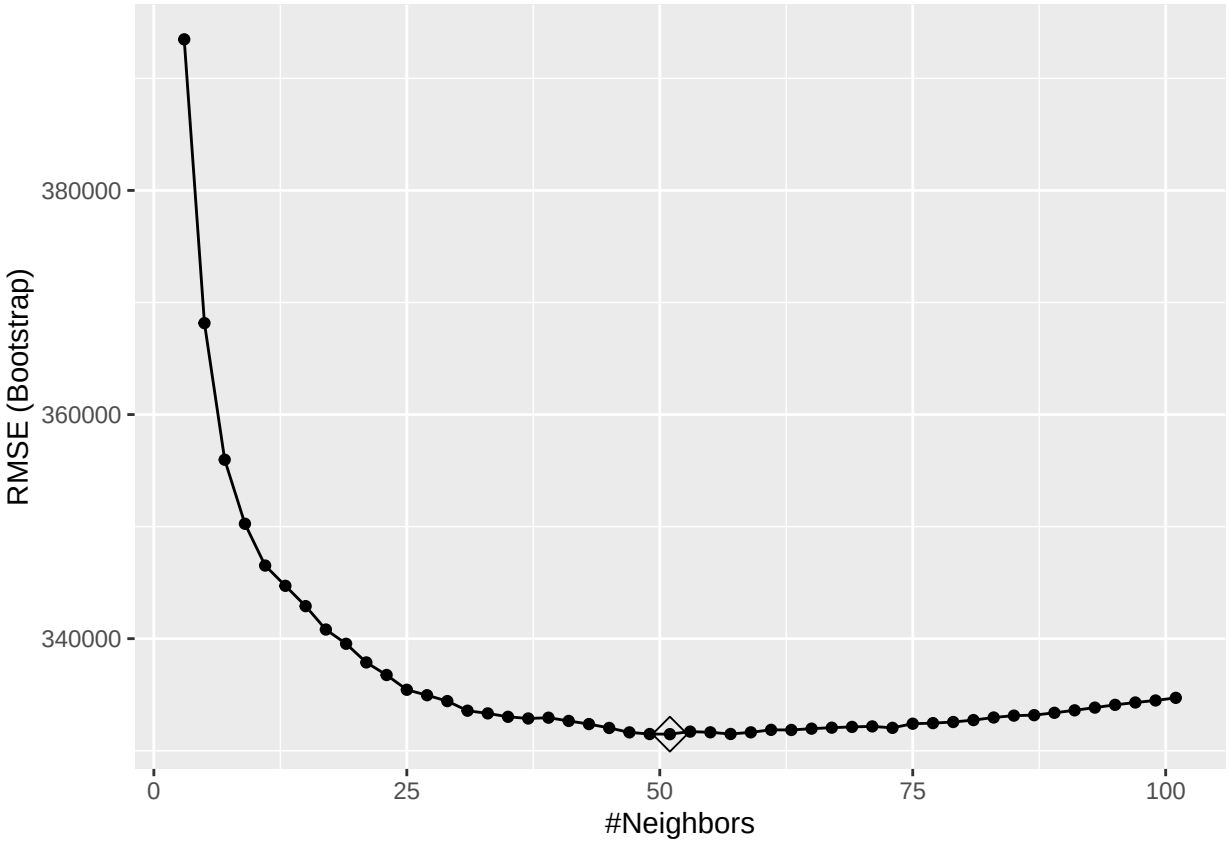
The initial models wer a simple linear model and a generalized linear model providing the baseline measurements. The initial models were taken using the lm and glm methods just to see if there was a significant difference. In this case there clearly wasn't as the results were exactly the same. Here's the results for the linearized and generalized linear model.

	Bias	RMSE	MSE	MAE	MAPE
Linear Model	35102.1	403335.2	162679278846	311036.6	7.85824
Generalized Linear Model	35102.1	403335.2	162679278846	311036.6	7.85824

3.3 Final Models

In effort to try to learn multiple algorihtms or at least get a general feel for how things work, the data was run using the Random Forest, Regularized Random Forest, SVM, L2 Regularized SVM with Linear Kernel, and KNN models. Once the data was run through each individual model it became clear that the KNN model performed the best on every metric, since the mode was originally run with the default k, it was re-rerun after finding the ideal value which was fairly straight forward using the following code.

```
knnFit <- train(eDaily ~ tMax + tMin, method = "knn", data = trainData, tuneGrid = expand.grid(k=seq(3,
#Going to try to find the value of k that minimizes there error
minIdx <- which.min(knnFit$results$RMSE)
minIdx
#So after finding the minimum i realized that the fit has a bestTune variable that gives it without nee
#do any other calculations.  So it goes.
knnHat <- predict(knnFit, testData, type = "raw")
```



You'll notice that the caret train function automatically found the minimum k value within the sequence provided and the predict function will use the optimal value without any real input from the user. The index of the k that was found to have the minimum RMSE was 25 and the value of the RMSE at this k value was 3.3147907×10^5.

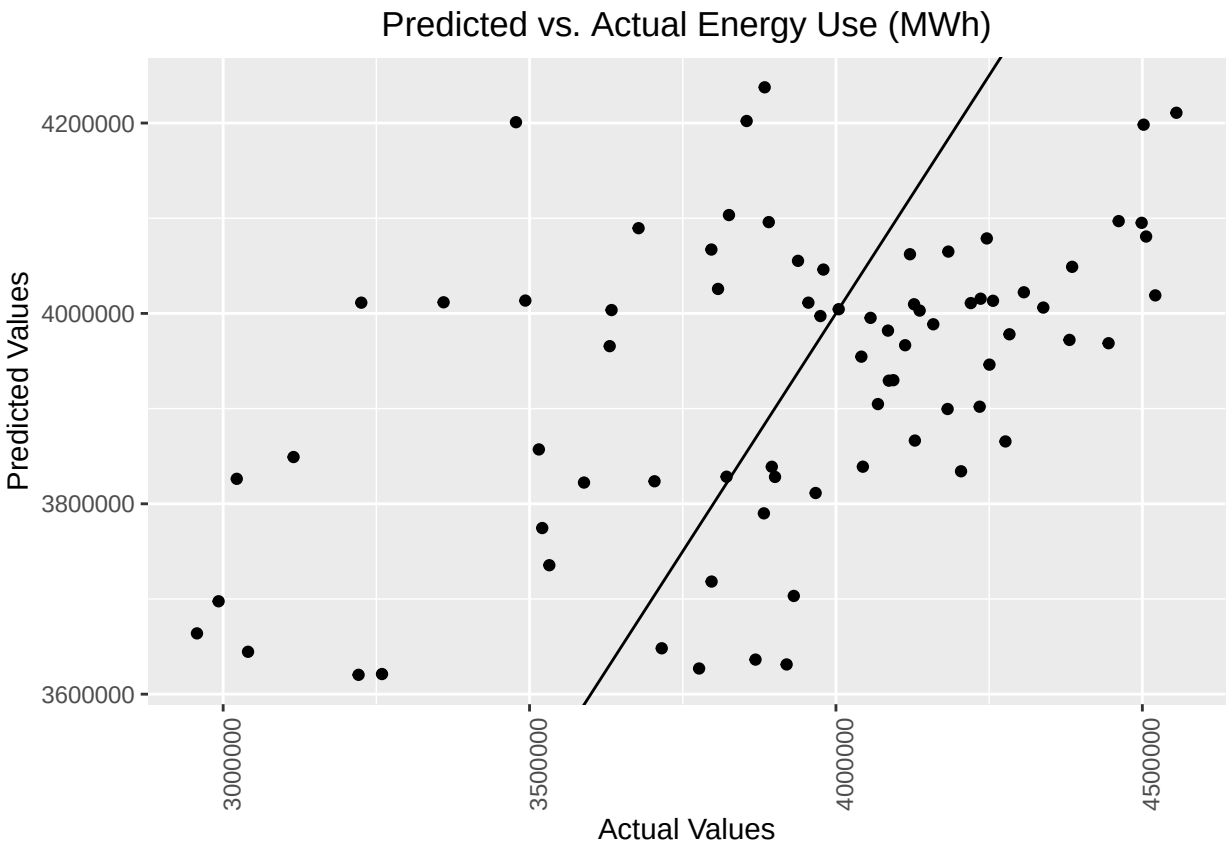
4 Results

The Analysis wasn't as in depth as originally intended due to the limitations I put on the data early on in the process so to provide some reasonable insight into the project I tried multiple models and reviewed multiple metrics for determining the accuracy of the algorithms. You'll notice that the following results table include the calculated Bias, RMSE, MSE, MAE, and MAPE. MAPE was largely included so that a simple percent error was included. The simple percent error was much easier to quickly look at the measurements to determine which one performed the best though I realize that MAPE has it's drawbacks. Though by every measure the optimized KNN algorithm performed the best.

	Bias	RMSE	MSE	MAE	MAPE
Linear Model	35102.105	403335.2	162679278846	311036.6	7.858240
Generalized Linear Model	35102.105	403335.2	162679278846	311036.6	7.858240
KNN	9721.675	343556.0	118030704605	285275.2	7.269751
Random Forest	38185.645	395371.1	156318344522	316405.4	8.018094
Regularized Random Forest	34738.346	394134.7	155342200804	316981.5	8.042404
SVM	61083.060	402950.8	162369366914	309204.2	7.765160

Least Squares SVM	120977.208	422684.6	178662230087	318790.7	7.871297

Here's a plot of the predicted (KNN) vs. actual daily energy use for reference.



You'll notice that the predictions were in the ballpark but had a much narrower range, 6.1722924×10^5 MWh vs the actual test data range of 1.5983274×10^6 which means there are definitely tuning improvements that could be made to the model. The plot makes me feel like I'm on the right track but the data is probably underfitted and there's additional optimizations that can be made to improve the models.

The bias that is being calculated is the historical average error, so on average the predictions overshot the target which for the actual application may be the better outcome. With high bias values I do feel that things should be re-evaluated from the beginning but time is short and I must move on. The RMSE values are the standard root mean square error which is probably the best measure to identify the predictions while the mean square error is essentially the RMSE just without the square root at the end. It didn't really provide much insight here, the mean absolute error was similar to the MSE but since the values weren't squared the negative numbers could counter the positive values bring the total value down a bit. Finally, there was the MAPE which is the mean aboslute percent error which takes into account the mean absolute error but also the value of the outcome for each point in the test data. The formulate for MAPE is: $MAPE = \frac{1}{n} \sum \frac{|e_t|}{d_t}$.

For the error function, maybe I should have used a range of energy use such as +/- 1000 MWh to trim the the error signal to a binary output and used it as a categorical measure but that seemed to be missing the point. Also, since this is energy use and energy predictions that are too large are a problem but too low is catastrophic maybe I should have represented the problem and error conditions with this in mind.

5 Conclusion

The goal of the project was to show what was learned throughout the Harvardx data science course by creating your own project using the techniques discussed throughout the nine courses. With this in mind I believe this project touched every part of the course from the data visualization, modeling, data wrangling, regression, and machine learning. The chosen project was an attempt to try to predict the daily energy use using the daily maximum and minimum temperature for a country, in this case Spain. Several models were run with the best version being an optimized KNN model with the ending prediction resulting in an error of 7.27% using the MAPE metric. The results show that you can get in the ballpark with just the temperature information but the extremes of temperature seem to have a better effect than the actual value and it is a complicated problem so more than just a couple of parameters should be used in a real-world application.

There were several faults in this analysis starting with the fact that the data set probably should have used several full years as opposed to just two, the data set was limited on Kaggle so I used all the available data but it also encompass the Covid-19 epidemic, so it may not have been the most representative data set. Other factors such as adding additional weather stations that span the country and predictors such as precipitation, cloud cover could be added, elevation, and humidity could be added to reduce the error. Several predictors were used based on temperature but they probably don't have the full range of information needed to make a more accurate prediction.

Predicting the energy use most likely is already done by the agencies that govern power generation or at least it should be monitored by agencies such as EIA, IRENA, or IEA. Instead of using daily swings in temperature the agencies probably look at seasonal data since the power systems have large inertia and it probably doesn't make sense to look at such granular data. It would be better for seasonal effects over a number of years to better plan for the necessary generation for any reasonable time period. It would probably also be useful to look at extreme temperatures as opposed to just the min/max temps. If you look through the data you can visually see that the days where there were more energy used the temperature was at one of the extremes.

Given more time there are a bunch of things that I would change from the initial data set to the validation method. I probably should have used a scale based on the MWh error as opposed to the using the absolute error. I threw in the MAPE because a percentage error provides a feel for where things are at where the absolute errors on continuous data are useful but they don't trigger the same response when viewing as a percentage even though MAPE has it's faults.

6 References

Irizarry, Rafael A., "Introduction to Data Science: Data Analysis and Prediction Algorithms in R" https://rafalab.dfci.harvard.edu/dsbook/

"Price of electricity and the renewable energy", Afroz, kaggle. 2024. https://www.kaggle.com/datasets/pythonafroz/price-of-electricity-and-the-renewable-energy?resource=download

"Climate Data Online: Dataset Discovery", National Oceanic and Atmospheric Administration, 2024. https://www.ncdc.noaa.gov/cdo-web/datasets