Duke Energy Final Draft

Dylan Demo, Denny Dunmire, John Roettig, Winnie Yang

Elon University Statistics Practicum

December 4, 2022

**INTRODUCTION**

The purpose of this report is to better understand the relationship between outdoor temperature and energy usage (kWh) of an individual customer for Duke Energy. Perhaps outdoor temperature has the most impactful influence on a customer’s energy usage due to the significant power their HVAC units consume. Duke Energy has collected a year’s worth of data on 200 customers for each hour of the day (8760 observations per customer), alongside the important records of outdoor temperature and kWh usage for each respective hour. Considering the association between outdoor temperature and kWh usage with confounding variables such as seasons, time of day, time of year, weekdays, weekends, day, night, customer preference, customer demographics, or however else the data may be partitioned, is vital to the success of the modelling process. Analyzing and modelling this data will allow Duke Energy to target customers with inefficient units and homes with programs designed to improve overall efficiency. The primary goal of this project is to determine what variables can be used to best predict kWh across a population of households and fit the resulting model.

**PROCESS**

Data Wrangling

Customers 67 and 148 have been removed from the data due to incorrect or missing entries, as this was causing errors when calculating MSE. Additionally, any NA values from the temperature and kWh variables were filtered out to ensure that we were not considering the missing data in our analyses.

Transforming and Subsetting Data for Explanatory Variables

The main method of approach for this project can be boiled down to different transformations and partitions of the data in order to find the best possible explanatory variables for predicting spikes in kWh usage. After combining all customer data sets (200 customers in total), we created the following new columns for factors that we believe might affect our response variable (kWh): date, time, day of week, weekend, season, working hours, night vs. day, and the binary “season” (heat vs. AC use) (see appendix for specific column definitions). Explanatory variables such as season or “heat vs AC use” will be essential in any model due to their inherent relationship with outdoor temperature. If not used as an explanatory variable in our model, the data will likely be filtered using these variables to reduce variation before any modeling is applied.

Measuring Model Strength with Training vs Testing Data

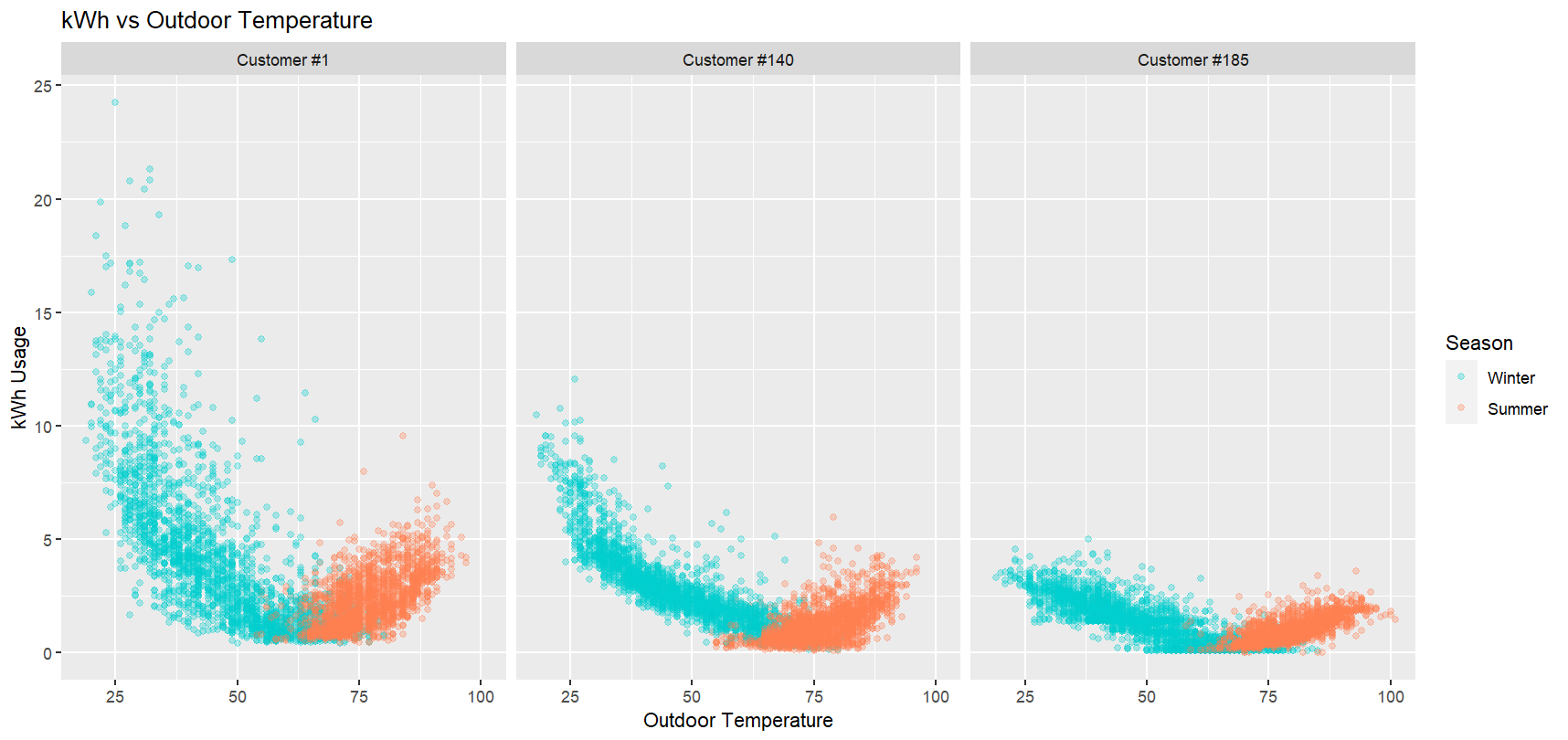
Mean MSE across all customers was used to measure model strength. To calculate this, data for each individual customer was randomly split into 80% training data and 20% testing data, where each customer’s MSE was stored in a vector (198 MSE’s in total, as two customers were removed from the data) to then compute the mean MSE for all customers. The mean MSE for all customers tells us our model’s predictive capability for data it has not yet seen, or in other words, how far off our model’s predicted values (80%) were from the testing values (20%).

General Modelling Approach

If residual plots of non-transformed models were unsatisfactory (mainly regarding response variable kWh), further transformations were applied to reduce outliers and variation. While only a few example models are discussed throughout this report, each of these models was tested numerous times with varying combinations of explanatory variables; the example models themselves serve as method for visualizing the overall process. More details regarding these aspects are discussed at greater length throughout the report.

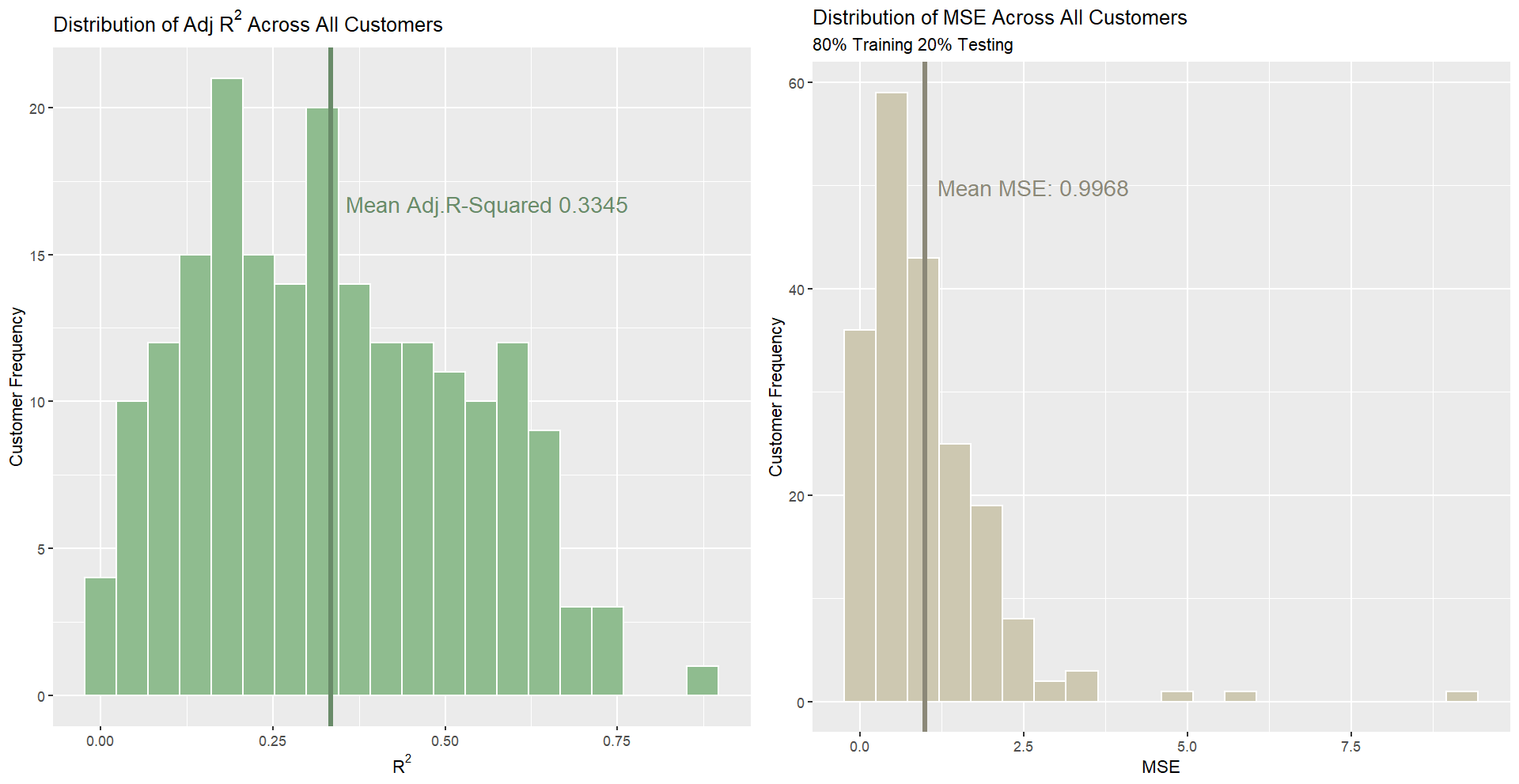
**STARTING SIMPLE**

Before fitting any models, we must first understand the relationship between our response variable (kWh) and our main explanatory variables (outdoor temperature and season) across individual customers. We would ideally expect the relationship between kWh outdoor temperature, and season to look like the following:

These customers in particular follow the general trend of high kWh usage in the winter and summer, but many others do not. Even among the ideal customers above, note the differences in variation among max kWh usage. If we apply our most basic model that coincides with the expected trend, we have:

kwh ~ temperature + factor(season) + temperature\*factor(season)

Note that this model includes an interaction between temperature and season, due to the apparent differences in slopes across the winter and summer seasons. To visualize this model’s strength across all customers, we plotted each customer’s adjusted R-squared and MSE on histograms. For the distribution of MSE, the data was separated into an 80-20% train-testing split.

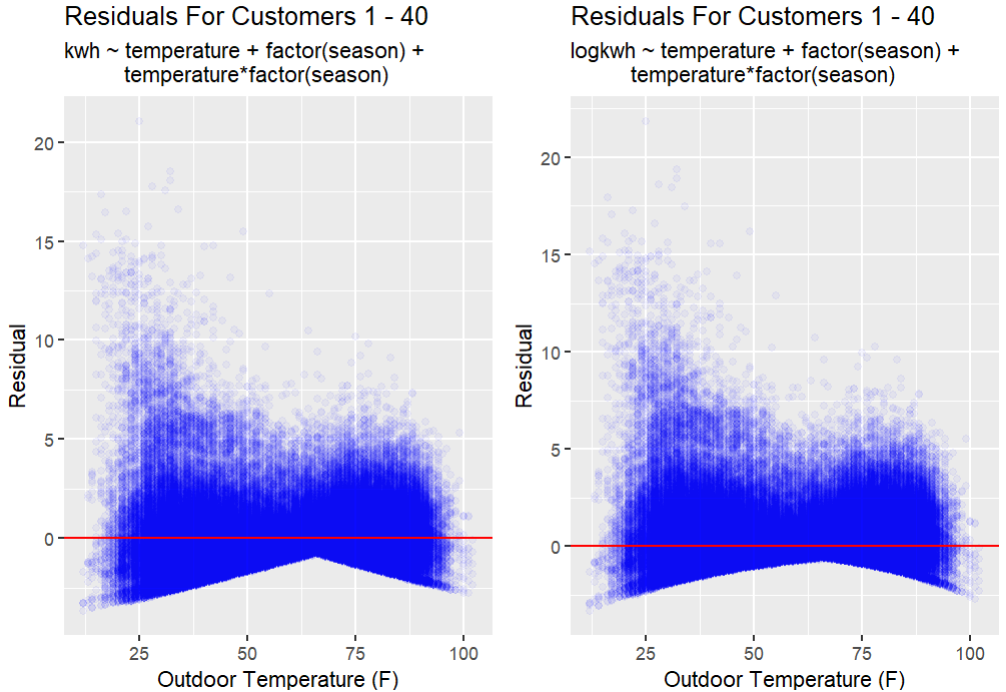


This model produced a mean R-squared of .3345 and a mean MSE of .9968. The adjusted R-squared seems relatively weak compared to the mean MSE, which may be due to the clustered distribution of outdoor temperature values. Beyond this, we continued applying the same training-testing split for models with transformed kWh or different explanatory variables altogether in search of a lower mean MSE.

**DISCUSSION OF TRANSFORMATIONS AND RESULTS**

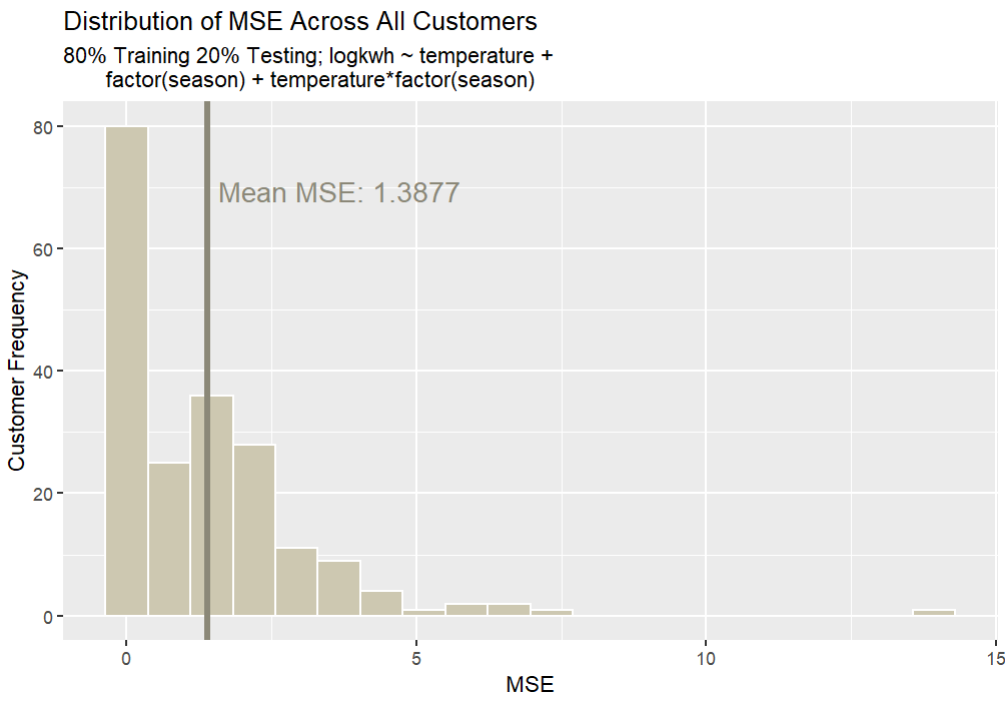
Log Transformation

We knew early in the process that kWh without a transformation could be tricky to work with. Beyond the linear model, one of the first things we tried was performing a log transformation on the variable kWh in hopes of reducing the more extreme variation. For reference, note linear model’s (left plot) residual variation in the winter (approximately below 50 degrees F) relative to its residual variation in the summer. After transforming our response to logkWh to target this variation, we have the resulting residual plot on the right.



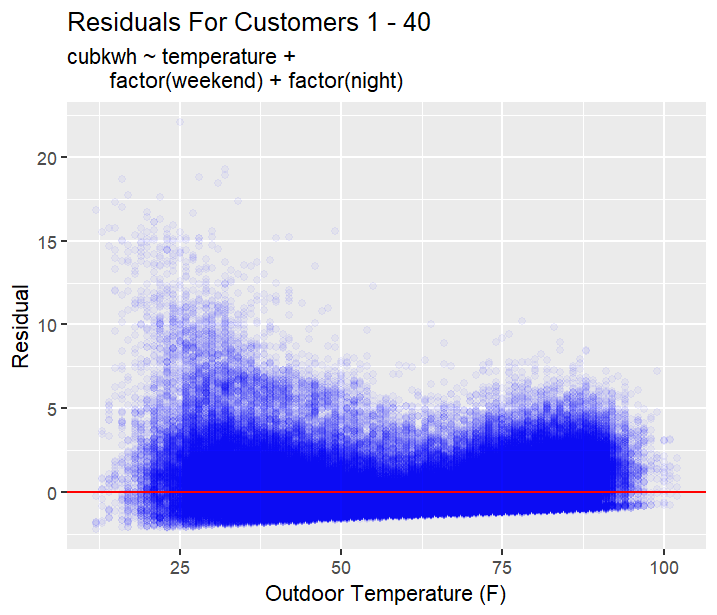
\*Only customers 1-40 were used in computing the residual data above. Graphing all 198 customers (approximately 1.7 million observations) would utilize too much CPU. Additionally, residual axis’ in both plots are expressed in non-transformed kWh

The log transformation of kWh did little to target the variation in the winter, resulting in a slightly different distribution of MSE across individual customers, but overall, a higher mean MSE than the linear model.

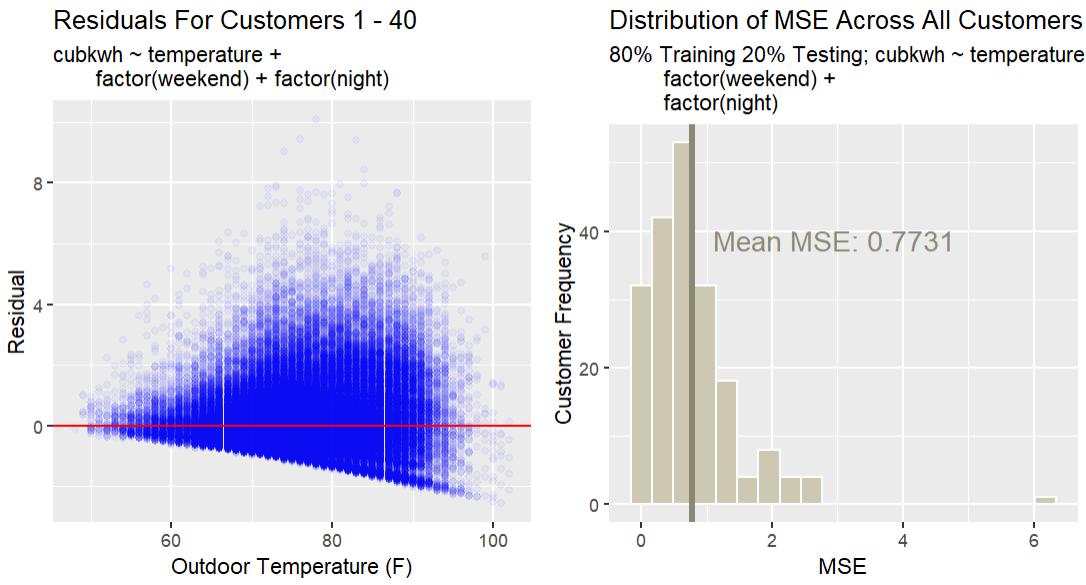


Cube Root Transformation

Relative to the log transformation, the cube root transformation appeared slightly more effective at reducing overall variation, but not in the areas we had hoped. While an improvement, this transformation alone did not reduce the higher-kWh variation in the winter we had hoped to address. The residual plot below displays a noticeable reduction in variation below the model line for both summer and winter.



After testing numerous models, we noticed that our ability to predict kWh output is better in the summer compared to the winter, so we ultimately reduced our data to exclusively summer observations before testing models with different explanatory variables. The residual plot on the left represents summer observations only, which clearly exhibits less variation than winter observations. Relative to other models, this model had the lowest mean MSE of .7731.



**FINAL CONCLUSIONS**

Although we have yet to find a suitable predictive model for the entirety of the sample thus far, we have made substantial progress in partitioning the data and writing code to better visualize and investigate the data. Methods such as the employment of degree days, lag variables, transformations of the data, and shifting the metric of what we determined to be a “good model” to MSE have slowly improved the quality of our models, despite not using some of these variables in the end. The best model we could find, as in the one with the lowest mean MSE across all customers, was cubkwh ~ temperature + night + weekend for only the summer months, with a mean MSE of .7731.

**LIMITATIONS**

Fourier Transformation

Over the course of our work, there were several problems that we encountered. It was recommended to use the Fourier transformation to transform our temperature variable. However, after doing some research on the transformation we realized that we had neither the time nor the expertise to implement this effectively.

Demographic Data

Customer demographics were not included in any modelling due to the sizable number of missing or likely incorrect entries. The variable “home\_year\_built” contained 100 customers with a year value of “0” and 29 customers with “1990,” which were altogether claimed as missing or incorrect. While only 18 customers had missing values for the variable “dwelling\_type,” we discussed early on that the variable levels (“Multiple Family” and “Single Family”) were likely inaccurate and thus unreliable for modeling. Perhaps household square footage had the most potential to predict customer kWh; however, 100 customers had a square footage value of “0.” which we again deemed missing. Among other reasons, variables such as “city, state, and owner\_renter” were not included due to opinions on overall usefulness in predicting kWh.

**APPENDIX**

1The following code was used to partition the data into a variety of variables that could be used in fitting the model. The variables added to the data include:

* heat – Divides the year into days when most people would be running heat in their homes and when most people would be running the AC. The variable is TRUE when most people would be using heat in their home rather than AC and FALSE otherwise.
* dayofweek – Adds the day of the week.
* weekend – Is TRUE if dayofweek is Saturday or Sunday and FALSE otherwise
* season - Specifies the season of each date as a numerical value. 1 = winter, 2 = spring, 3 = summer, 4 = fall
* hour – Gives the hour when a measurement is taken (0-24).
* working – Is TRUE if the measurement is taken between the hours of 8am and 5pm on a weekday, does not account for holidays.
* lkwh – A log transformation of the kilowatt-hour variable.
* night – Is TRUE if the measurement is after 10pm or before 6am, which we determined to be the hours that the majority of households considered to be nighttime.
* lag1 – Uses the lag() function to create a lag variable which can be defined as a variable using past values. This lag variable uses values that are one row behind. Other lag variables were used in the data but are not included in this appendix for fear of redundancy.
* avgtemp – Represents the average temperature on a given day. Included in a separate dataframe entitled “degreeday.”
* degday – A variable for the degree day of each calendar day included in the data. This variable takes the average temperature of a given day and subtracts 65 (the determined threshold between heat use and AC use) from it. This allows us to include a wider range of data in a model over the span of the entire year.

[data] <- [data] %>%

filter(!is.na(temperature), !is.na(kwh)) %>%

arrange(measurement\_dttm\_hb) %>%

mutate(date = date(measurement\_dttm\_hb),

heat = (ymd("2021-10-15")<= date & date <= ymd("2022-04-15")),

dayofweek = wday(date, label=TRUE),

weekend = (dayofweek %in% c("Sat", "Sun")),

hour = hour(measurement\_dttm\_hb),

working = (8 <= hour & hour<=17 & weekend==FALSE),

lkwh = log(kwh),

night = (hour > 22 | hour < 6),

lag1 = lag(temperature, 1))

View([data])

[data]$season = ((yday([data]l$date) >= 355 | yday([data]$date) <= 78) \* 1) +

((yday([data]$date) >= 79 & yday([data]$date) <= 170) \* 2) +

((yday([data]$date) >= 171 & yday([data]$date) <= 264) \* 3) +

((yday([data]$date) >= 265 & yday([data]$date) <= 354) \* 4)

#Winter = 1, Spring = 2, Summer = 3, Fall = 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2This variable “degree day” was used in some testing models for the cube root transformation. This variable was not included in the final model.

degreeday = [data]%>%

group\_by(date, customer\_number) %>%

summarize(mintemp = min(temperature),

maxtemp = max(temperature)) %>%

mutate(avgtemp = (maxtemp + mintemp) /2,

degday = avgtemp - 65)

View(degreeday)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3This function was created to ease the filtering of data into a single customer.

household <- function(data, x){

sub = filter(data, customer\_number == x)

sub

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4This is the for loop that applies the defined model to each customer’s data (excluding 67 and 148) and stores each customer’s R-squared and MSE into their respective vectors. MSE is not included in the summary function outright, resulting in code that is slightly more complicated; thus, the MSE (“MSEp”) calculation for the training-testing split (“SplitRatio = .8”) had to be coded manually. While we did calculate the average R-squared, the mean of all customer MSE’s was the main factor in determining model strength. Also note the seed for reproducing the same results.

allMSE3 = NULL

allrsquares =NULL

set.seed(3049)

for(i in c(1:66, 68:147, 149:200)){

currHouse = household([data], i)

currHouse = filter(currHouse, season %in% c(1,3))

sample = sample.split(currHouse$kwh, SplitRatio = .8)

train = subset(currHouse, sample == TRUE)

test = subset(currHouse, sample == FALSE)

model = lm(formula = kwh ~ temperature + factor(season) +

temperature\*factor(season),

data = train)

test$yhat = predict(model, test)

MSEp = sum((test$kwh - test$yhat)\*\*2)

MSEp = MSEp / nrow(test)

allMSE3[i] = MSEp

allrsquares[i] = summary(model)$adj.r.squared

}

allMSE3 <- allMSE3[!is.na(allMSE3)]

mean(allMSE3)

allrsquares <- allrsquares[!is.na(allrsquares)]

mean(alrsquares)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5This code generates the same R-squared and MSE calculations shown in the linear model above. In this example, the model is using the three variables – temperature, season, and temperature\*season– to predict log transformed kWh. Note that the predicted values for the MSE calculation (exp(test$yhat)) were transformed to regular kWh before computing the residual data and thus the MSE.

allrsquares = NULL

allMSE2 = NULL

set.seed(3049)

for(i in c(1:66, 68:147, 149:200)){

currHouse = household([data], i)

currHouse = filter(currHouse, season %in% c(1,3))

sample = sample.split(currHouse$kwh, SplitRatio = .8)

train = subset(currHouse, sample == TRUE)

test = subset(currHouse, sample == FALSE)

model = lm(formula = lkwh ~

temperature + factor(season) + factor(weekend),

data = train)

test$yhat = predict(model, test)

MSEp = sum(((test$kwh) - exp(test$yhat))\*\*2)

MSEp = MSEp / nrow(test)

allMSE2[i] = MSEp

allrsquares[i] = summary(model)$adj.r.squared

}

allMSE2 <- allMSE2[!is.na(allMSE2)]

mean(allMSE2)

mean(allrsquares)

allrsquares <- allrsquares[!is.na(allrsquares)]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6This code generates the same R-squared and MSE calculations for the cube root transformation. Note that the MSEp variable is created by cubing the yhat, this is because the cubkwh variable is a cube root transformation. It is necessary to “undo” whatever transformation that has been applied to the kWh variable when calculating MSE, similar to the log transformation above. Again, mean R-squared was calculated, but not fully considered in our model strength evaluation compared to mean MSE. Also note that this model only includes the summer season (“season==3'”).

allrsquares = NULL

allMSE = NULL

set.seed(3049)

for(i in c(1:66, 68:147, 149:200)){

currHouse = household([data], i)

currHouse = filter(currHouse, season == 3)

sample = sample.split(currHouse$kwh, SplitRatio = .8)

train = subset(currHouse, sample == TRUE)

test = subset(currHouse, sample == FALSE)

model = lm(formula = cubkwh ~ temperature + weekend + night, data = train)

test$yhat = predict(model, test)

MSEp = sum((test$kwh - (test$yhat)\*\*3)\*\*2)

MSEp = MSEp / nrow(test)

allMSE[i] = MSEp

allrsquares[i] = summary(model)$adj.r.squared

}

allMSE <- allMSE[!is.na(allMSE)]

mean(allMSE)

mean(allrsquares)

allrsquares <- allrsquares[!is.na(allrsquares)]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_