Introduction to nmecr

Mrinalini Sharma[[1]](#footnote-1)

David Jump[[2]](#footnote-2)

Devan Johnson[[3]](#footnote-3)

November 22, 2019

The nmecr package was written to integrate the past efforts of the energy efficiency industry to streamline meter-based Measurement & Verification of EE projects. It brings together simple linear regression with outside air temperature, the change point models (from ASHRAE 1050-RP), and the Time-of-Week and Temperature model (developed by the Lawrence Berkeley National Laboratory) and builds enhancements upon these by accurately predicting the energy use profiles of facilities with multiple operating modes.

## Normalized Metered Energy Consumption (NMEC)

Meter-based energy efficiency programs are proliferating across the globe. This proliferation is influenced by the widespread availability of high frequency energy use measurements from new metering technology as well as advancements in statistical regression and other empirical energy data modeling methodologies. Program administrators may report savings as the overall reduction in normalized metered energy consumption (NMEC). This method to determine savings is based on analysis of meter data collected prior to and after energy efficiency and conservation measures have been installed. Referred to as advanced measurement and verification (M&V) by the industry, this time-granular data and updated modeling methods provide several advantages over other methods used to quantify the benefits of energy efficiency:

* It reliably determines the actual savings achieved at the meter
* It provides fast feedback on the facility’s energy performance and savings progress
* It enables identification and troubleshooting of issues that prevent savings realization

The nmecr package streamlines the application of the NMEC approach by enabling the:

* Management of high frequency, high volume data.
* Execution of advanced, state-of-the-art time-series data modeling algorithms.
* Comprehensive assessment of the validity of energy data models in specific applications.
* Quantification of uncertainties and risks associated with the energy savings projections in accordance with ASHRAE Guideline 14.

## Data

nmecr includes two datasets: temp and eload. These contain three years (03/01/2012 - 02/28/2015) of outside air temperature data and energy use data from a commercial building in North America. Energy conservation measures were installed between 03/01/2013 and 02/28/2014 in this building and so, we will use the first year (03/01/2012 - 02/01/2013) as the pre-implementation dataset and the third year (03/01/2014 - 02/28/2015) as the post-implementation dataset.

# Load data into an R session:  
  
data(eload)  
data(temp)

eload and temp are data frames with the two variables each. When using nmecr functions, ensure that the column headers of your datasets are the same as those shown below.

Eload:

#> # A tibble: 5 x 2  
#> time eload  
#> <dttm> <dbl>  
#> 1 2012-03-01 00:00:00 21505.  
#> 2 2012-03-02 00:00:00 20892.  
#> 3 2012-03-03 00:00:00 20435.  
#> 4 2012-03-04 00:00:00 15660.  
#> 5 2012-03-05 00:00:00 15864.

Temp:

#> # A tibble: 5 x 2  
#> time temp  
#> <dttm> <dbl>  
#> 1 2012-03-01 00:00:00 38.4  
#> 2 2012-03-02 00:00:00 39.9  
#> 3 2012-03-03 00:00:00 43.0  
#> 4 2012-03-04 00:00:00 49.7  
#> 5 2012-03-05 00:00:00 48.3

## Baseline and Performance Period Dataframes for Modeling

create\_dataframe() combines the eload and temp dataframes into one, filters by the specified start and end dates, and aggregates to an hourly, daily, or a monthly data interval. If an operating mode dataframe is supplied, create\_dataframe creates a list with two objects: dataframe, operating\_mode\_data.

# Baseline Dataframe  
  
baseline\_df <- create\_dataframe(eload\_data = eload, temp\_data = temp,   
 start\_date = "03/01/2012 00:00",   
 end\_date = "02/28/2013 23:59",   
 convert\_to\_data\_interval = "Daily")  
  
head(baseline\_df, 5)  
#> # A tibble: 5 x 5  
#> time temp HDD CDD eload  
#> <dttm> <dbl> <dbl> <dbl> <dbl>  
#> 1 2012-03-01 00:00:00 38.4 26.6 0 21505.  
#> 2 2012-03-02 00:00:00 39.9 25.1 0 20892.  
#> 3 2012-03-03 00:00:00 43.0 22.0 0 20435.  
#> 4 2012-03-04 00:00:00 49.7 15.3 0 15660.  
#> 5 2012-03-05 00:00:00 48.3 16.7 0 15864.

# Performance Period Dataframe  
  
performance\_df <- create\_dataframe(eload\_data = eload, temp\_data = temp,   
 start\_date = "03/01/2014 00:00",   
 end\_date = "02/28/2015 23:59",   
 convert\_to\_data\_interval = "Daily")  
  
head(performance\_df, 5)  
#> # A tibble: 5 x 5  
#> time temp HDD CDD eload  
#> <dttm> <dbl> <dbl> <dbl> <dbl>  
#> 1 2014-03-01 00:00:00 50.0 15.0 0 15989.  
#> 2 2014-03-02 00:00:00 43.5 21.5 0 16033.  
#> 3 2014-03-03 00:00:00 41.8 23.2 0 16938   
#> 4 2014-03-04 00:00:00 51.8 13.2 0 17325.  
#> 5 2014-03-05 00:00:00 51.0 14.0 0 17408.

## Energy Data Modeling

The baseline and performance period dataframes can then be used for energy data modeling using one of the four modeling algorithms available in nmecr:

* model\_with\_TOWT()\*: Time-of-Week & Temperature and Time-Only algorithms
* model\_with\_CP(): 3-Parameter Heating, 3-Parameter Cooling, 4-Parameter, and 5-Parameter algorithms
* model\_with\_SLR(): Simple Linear Regression algorithm
* model\_with\_HDD\_CDD(): Heating Degree Day only, Cooling Degree Day only, and a combination of Heating Degree Day and Cooling Degree Day algorithms

The common arguments for these four algorithms are:

1. training\_list (output from create\_dataframe())
2. model\_input\_options (see below)

\*model\_with\_TOWT() has an additional argument (prediction\_list) for creating data models and predictions together.

### model\_input\_options()

The four modeling algorithms have many specifications in common that can be specified using the assign\_model\_inputs() function. The following code chunk shows the default values for these model inputs

model\_input\_options <- assign\_model\_inputs(timescale\_days = NULL, interval\_minutes = 15,   
 has\_temp\_knots\_defined = FALSE,   
 equal\_temp\_segment\_points = TRUE,  
 temp\_segments\_numeric = 6,   
 temp\_knots\_value = c(40, 45, 50, 60, 65, 70),   
 initial\_breakpoints = c(50,65),   
 regression\_type = "TOWT")

Before creating data models, it is a good practice to visualize the energy use profiles:

#### Baseline Energy Use

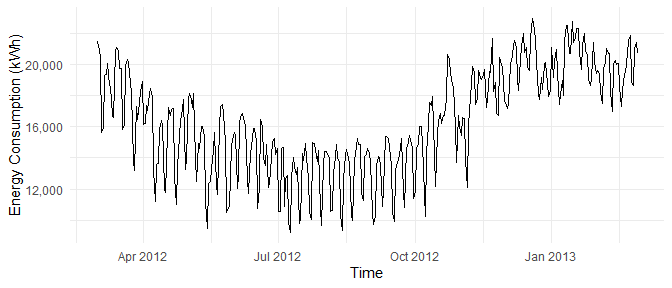


Figure 1: Baseline Energy Use over Time

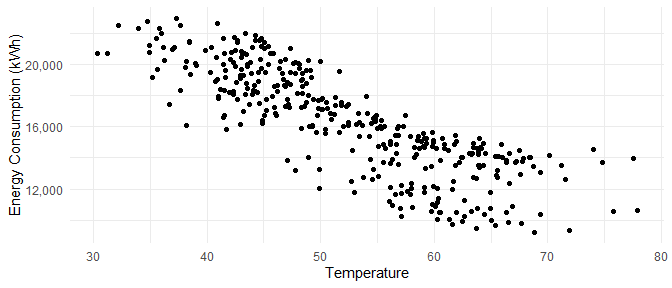


Figure 2: Baseline Energy Use over Temperature

Due to the high dependence on temperature, we can begin the data modeling using Simple Linear Regression and compare that to other modeling algorithms

### Model Creation

**Simple Linear Regression:**

SLR\_model <- model\_with\_SLR(training\_data = baseline\_df,   
 model\_input\_options =   
 assign\_model\_inputs(regression\_type = "SLR"))

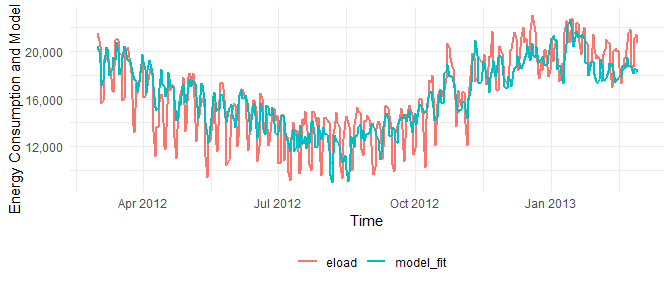


Figure 3: Baseline Energy Use modeled with Simple Linear Regression over Time

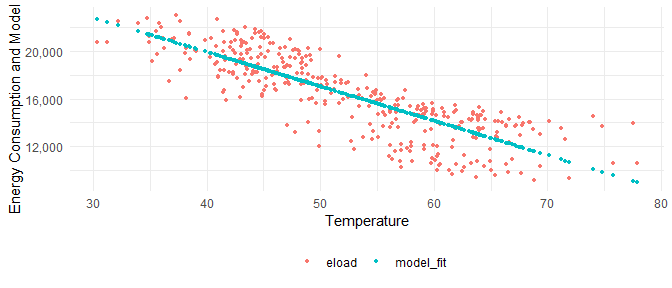


Figure 4: Baseline Energy Use modeled with Simple Linear Regression over Temperature

**Four Parameter Heating:**

Four\_P\_model <- model\_with\_CP(training\_data = baseline\_df,   
 model\_input\_options =   
 assign\_model\_inputs(regression\_type =   
 "Four Parameter Linear Model"))

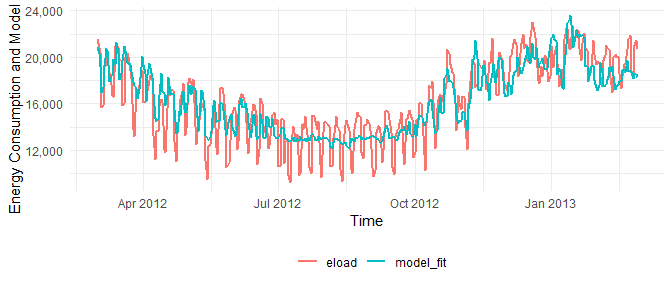


Figure 5: Baseline Energy Use modeled with Three Parameter Heating Regression over Time

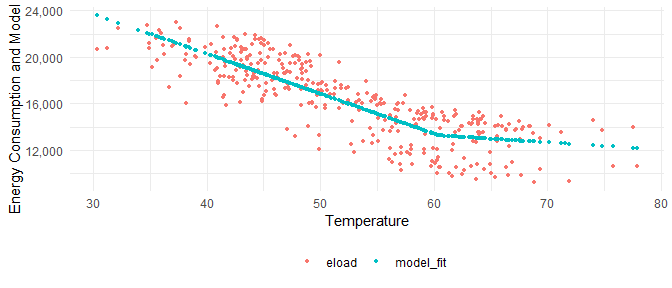


Figure 6: Baseline Energy Use modeled with Three Parameter Heating Regression over Temperature

**Time of Week and Temperature:**

TOWT\_model <- model\_with\_TOWT(training\_data = baseline\_df,   
 model\_input\_options =   
 assign\_model\_inputs(regression\_type = "TOWT"))

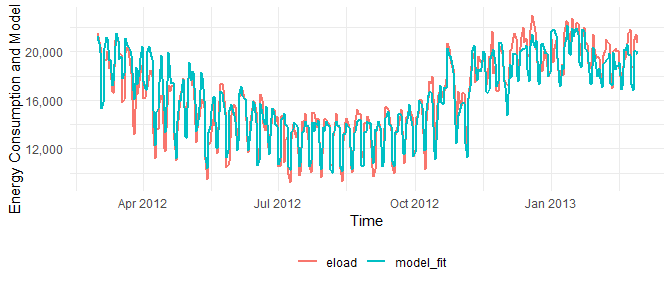


Figure 7: Baseline Energy Use modeled with Time-of-week and Temperature over Time

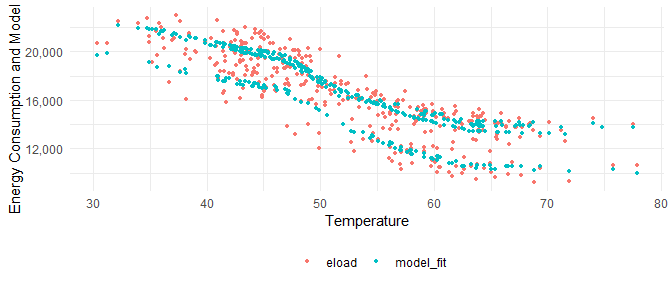


Figure 8: Baseline Energy Use modeled with Time-of-week and Temperature over Temperature

The plots show that the Time-of-Week and Temperature algorithm models the energy use profile better than the other two.

This insight can be confirmed through model statistics by using the calculate\_summary\_statistics() function. The following table summarizes the results from this function for each of the models assessed above:

SLR\_stats <- calculate\_summary\_statistics(SLR\_model)  
  
Four\_P\_stats <- calculate\_summary\_statistics(Four\_P\_model)  
  
TOWT\_stats <- calculate\_summary\_statistics(TOWT\_model)  
  
all\_stats <- bind\_rows(SLR\_stats, Four\_P\_stats, TOWT\_stats)  
  
model\_names <- c("SLR", "Four Parameter", "TOWT")  
  
all\_stats <- bind\_cols("Model Name" = model\_names, all\_stats)  
  
all\_stats  
#> # A tibble: 3 x 6  
#> `Model Name` R\_squared CVRMSE NDBE MBE `#Parameters`  
#> <chr> <dbl> <dbl> <dbl> <dbl> <int>  
#> 1 SLR 0.692 0.115 -6.11e-19 -9.97e-15 2  
#> 2 Four Parameter 0.720 0.109 -1.31e-17 -2.14e-13 4  
#> 3 TOWT 0.897 0.0678 -1.81e-16 -2.96e-12 19

* CVRMSE: Coefficient of Variation of Root Mean Squared Error
* NDBE: Net Determination Bias Error
* MBE: Mean Bias Error
* R\_squared: Coefficient of Determination

## Assessing project fit for NMEC

Using the modeling statistics and the savings uncertainity for 10%, we can determine the validity of the NMEC approach for a certain project. For the building decribed here, following are the key values to be used for this assessment

1. CVRMSE: 6.78% (should be < 25%)
2. NDBE: 0.0000 (should be < 0.005%)

The California Public Utilities Commission requires savings to be detectable above model variations. nmecr interprets this using ASHRAE Guideline 14 - 2014’s formulation for savings uncertainty, which relates the savings uncertainty to the model goodness of fit metric CV(RMSE), the confidence level, the amount of savings, the amount of data used to develop the model, and the amount of data required to report savings. It includes a correction when autocorrelation is present (which occurs mainly in models developed from daily and hourly data). LBNL has shown this uncertainty formulation with correction for autocorrelation underestimates the savings uncertainty. More work on this issue is needed. Until a better formulation is available, nmecr uses ASHRAE’s method only as an estimation.

TOWT\_savings\_10 <- calculate\_savings\_and\_uncertainty(prediction\_df = NULL,   
 modeled\_object = TOWT\_model,   
 model\_summary\_statistics = TOWT\_stats,  
 confidence\_level = 68)  
  
TOWT\_savings\_10$savings\_summary\_df  
#> savings\_pct savings\_uncertainty savings\_pct\_for\_50pct\_uncertainty  
#> 1 0.1 0.1011888 0.02023776  
#> confidence\_level  
#> 1 68

1. Savings Uncertainty for 10% savings (at 68% confidence level): 10.1% (shoule be < 50%).

The savings percentage required to meet the threshold of 50% uncertainty, at the 68% confidence level, is 2% (as shown by the output above, see: savings\_pct\_for\_50pct\_uncertainty).

In addition to evaluating the model metrics, it is essential to ensure that the modeled profile follows the actual energy use closely (see Figure 8 above).

## Energy Load Prediction

The energy use models can be used to predict energy use during a different time period. Predictions in the reporting period are called the Adjusted Baseline Energy Use. The difference between the Adjusted Baseline Energy Use and reporting period actual energy use is called the Avoided Energy Use.

TOWT\_predictions <- calculate\_model\_predictions(training\_data = baseline\_df,   
 prediction\_data = performance\_df,   
 modeled\_object = TOWT\_model)

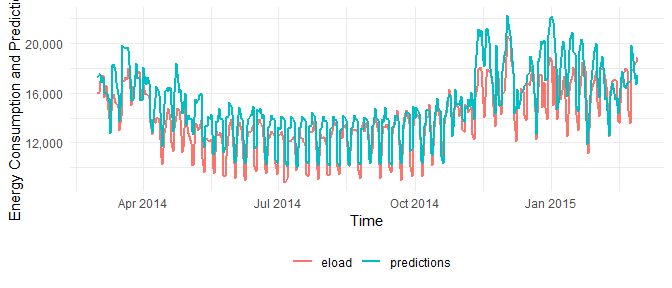


Figure 9: Performance period energy use and TOWT model predictions over Time

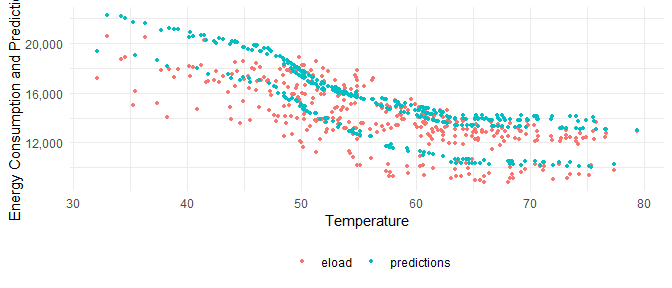


Figure 10: Performance period energy use and TOWT model predictions over Temperature

## Energy Savings Achieved and Uncertainity

TOWT\_savings <- calculate\_savings\_and\_uncertainty(prediction\_df = TOWT\_predictions,   
 modeled\_object = TOWT\_model,   
 model\_summary\_statistics = TOWT\_stats,  
 confidence\_level = 68)  
  
TOWT\_savings$savings\_summary\_df  
#> performance\_period\_use adjusted\_baseline\_use savings savings\_pct  
#> 1 5103905 5509756 405850.6 0.07  
#> savings\_uncertainty savings\_pct\_for\_50pct\_uncertainty confidence\_level  
#> 1 0.1445554 0.02023776 68

Savings achieved at the meter for this project amount to ~7% of the Adjusted Baseline Use with an associated savings uncertainty of 14.5%. ASHRAE requires that the savings uncertainty for a project be below 50% at the 68% confidence level. The 14.5% uncertainty is well below this threshold.

1. [msharma@kw-engineering.com](mailto:msharma@kw-engineering.com) [↑](#footnote-ref-1)
2. [djump@kw-engineering.com](mailto:djump@kw-engineering.com) [↑](#footnote-ref-2)
3. [johnson@kw-engineering.com](mailto:johnson@kw-engineering.com) [↑](#footnote-ref-3)