

Modeling ZNE Consumption Targets

Prepared by:

Lucas Beslow, Leslie Nelson, and Taher Othman

University of California, Davis

June 8, 2021

Advisors: Dr. Kurt Kornbluth, Josh Morejohn, and Rhys Davis

Project Background	3
1.2 Problem description	3
1.3 Relevant prior art/literature review	4
Methodology	4
2.1 Major considerations, constraints, assumption and metrics	4
2.2 Equipment and Procedure	5
2.2.1 Model Logic	5
2.2.2 Data Sources	6
2.2.3 Model Development	6
2.2.3.1 Linear Regression	7
2.2.3.2 Hourly Budgets	7
Results	8
Discussion	9
Recommendations and Conclusions	10
Bibliography	11
Appendix	12
7.1. Appendix A	12
7.2. Appendix B	14
7.3. Appendix C	17
7.4. Appendix D	18

1. Project Background

1.1 Brief history and context

In 2013, construction was completed at the University of California, Davis West Village. At the time, this was the largest planned zero net energy (ZNE) community in the United States. Based on the immense public interest in the project and the connection to UC Davis, researchers have studied the energy performance of the West Village closely to determine if it is meeting ZNE goals. This level of data-verified ZNE status analysis is uncommon with ZNE projects, however. Over 80% of commercial buildings declaring a zero net energy (ZNE) goal do not have data-verified actual ZNE status (Sanguinetti et al., 2018). Dr. Angela Sanguinetti and her team at the UC Davis Institute of Transportation Studies have sought to address this issue.

Dr. Sanguinetti has been leading a project for several years that aims to increase West Village occupant awareness of the ZNE progress of the West Village buildings. The hope is that increased occupant awareness will affect the energy usage behavior of occupants, thereby helping reduce energy loads and assist in the achievement of ZNE at West Village. In pursuit of these ends, Dr. Sanguinetti and team have developed a West Village Energy Dashboard (<http://wved.ucdavis.edu/#/>) and supporting ZNE budgets to inform occupants if the building they are occupying is on track for ZNE status at the end of the year. The dashboard relies on a model that determines monthly, weekly, daily, and hourly energy “budgets” that occupants must stay under to remain on track for ZNE status. To clarify further, these “budgets” are essentially an amount of energy consumption a building should not exceed in a given time period to ensure a building reaches ZNE status by the end of the year.

1.2 Problem description

Energy budgets are based on the idea that energy demand must equal or fall short of energy production in order for a building or collection of buildings to reach ZNE status. The usefulness and accuracy of these budgets is thereby reliant on an accurate accounting of the anticipated energy production of onsite generators as well as onsite energy demand. For the West Village, both of these are predicted by an existing model. This model is what informs the West Village Energy Dashboard and is pivotal in the accuracy and success of the project as a whole.

The current model framework limits its ability to accurately predict both energy production and demand. Most notably, the model is built off the base assumption that monthly energy demand and production remain the same year-to-year. While this is a good baseline to start predictions and budgets at the beginning of a year, the model lacks any dynamic inputs and therefore is inflexible when there is a discrepancy between the current and previous years’ energy generation and production. These discrepancies can occur due to fluctuations in weather, occupancy behaviors, and more. Our team was tasked with improving this model’s accuracy.

1.3 Relevant prior art/literature review

Judd et al. (2013) conducted a study on occupant behavior and found that having a designated person who would regularly track occupants' energy consumption and share the results with them, as well as suggest changes in their behaviour, was the most effective in making a difference in occupancy behaviour.

Coleman et al. (2013) on the other hand, suggests the utilization of wireless technologies to track occupancy behavior and provide the results directly to the occupants was an effective method in changing occupancy behavior. The study was found to be encouraging as occupants appreciated the personalized feedback, however, it was also found to be somewhat impractical to implement many changes in behavior when it came down to shared office space. Nevertheless, the engagement of occupants in energy performance is the first step towards a ZNE pathway.

Medrano. et al. (2018) suggests the utilization of linear regression and the software R to build a model that correlates energy consumption and weather. The model takes into account *degree days* to include temperature data. According to Dobson (2019), degree days are defined to be a measure of the duration and magnitude of the difference between the external temperature in any given day and a base temperature. A model that incorporates the duration of a temperature difference (i.e. degree days) is potentially a better energy consumption indicator than one that merely only includes a temperature difference.

Kneifel and Webb (2016) on the other hand, develop a regression model that utilizes two weather variables to predict the energy consumption of occupants on a daily level. The model implements explanatory variables of temperature, plane of array solar irradiance, and the day of every week. This model was found to be more accurate in its estimation of yearly energy consumption than the more popular building energy models widely used, like EnergyPlus and TRNSYS. Even more so, the study by Kneifel and Webb states that a more accurate model predicting energy consumption is possible if artificial neural networks (ANNs) are implemented. However, the complexity of ANNs make it somewhat impractical to implement in a simple model.

2. Methodology

2.1 Major considerations, constraints, assumption and metrics

The design process for the model logic was largely guided by the existing budget modeling logic. The existing model logic can be found in Appendix A. Each step of the existing model logic was analyzed to assess whether it should remain the same, be improved, or be redesigned. The only constraints given by the client were that the model output should be of a similar format to the existing model. The existing model outputs were hourly budgets by end use (HVAC, kitchen, plug loads, and lighting) for an entire year.

From our literature review, we were interested in utilizing weather and occupancy data to better predict energy consumption. Considering future weather predictions become less accurate farther into the future, the question of how far into the future the model must predict was raised. After talking with the client and reviewing the dashboard graphics that the model's output would eventually feed into, it was agreed that predicting energy budgets one week out would be sufficient.

The next consideration was the availability of data. To incorporate weather data into the model, the dashboard would have to be able to connect to a data source and pull in hourly temperature predictions periodically. The team noticed that the existing dashboard graphics include an outdoor air temperature value. The client shared that the temperature is displayed through an API connection to the website Dark Sky (<https://darksky.net/dev>). Dark Sky also has hourly predictions of outdoor temperature for over a week into the future, so it was determined to be the best data source. Dark Sky does not have historical temperature data, so temperature data from a sensor on campus that is stored in UC Davis' PI Database was determined to be the best option. Occupancy data was determined to be best approximated by specifying if a given day is a weekend or not, holiday or not, and if UC Davis school is in session or not. This data was all readily available.

The final consideration was whether a dynamic model that is updated every week or more frequently with data connections through API could be utilized and incorporated into the West Village Energy Dashboard. The idea of a dynamic version of the model was described to the client and her programming team and determined to be technically feasible if the model was coded in python and the outputs stored in an SQL database.

When designing the model, the assumptions made were that the Dark Sky and PI Database temperature values are in agreement, hourly temperature predictions a week into the future are accurate, and the historical metered data of the end uses at West Village is accurate and representative of a typical year.

2.2 Equipment and Procedure

2.2.1 Model Logic

The model is built to predict how much electricity a building can consume by end use (HVAC, lighting, kitchen, and plug loads) in any given hour for the building to be on track to have total energy consumption for the year be less than the photovoltaic generation. A relationship between temperature and occupancy inputs and predicted energy consumption has been developed based upon a year of data from 2018/2019.

The inputs to the model are occupancy characterizations (weekday or weekend, holiday or not holiday, and UC Davis school in session or no school in session) and cooling degree days (CDD) and heating degree days (HDD). Degree days are a means of incorporating outdoor temperature that reflect both temperature value and duration of temperature value. CDD and HDD can be calculated using the equations:

$$CDD = \sum_{hour=1}^{24} (T_{out, hour} - 65) \text{ for hours when } T_{out} > 65 \quad (1)$$

$$HDD = \sum_{hour=1}^{24} (65 - T_{out, hour}) \text{ for hours when } T_{out} < 65 \quad (2)$$

Where $T_{out, hour}$ is the outdoor temperature for each hour of the day. The occupancy characterizations, CDD, and HDD for each day are then input into the linear regression equation for each end use and the output given is an energy fraction, $EF_{end\ use, daily}$, for each end use in each day. $EF_{end\ use, daily}$ is explained by equation 3. Linear regression equations developed can be found in Appendix B.

$$EF_{end\ use,\ daily} = \frac{\text{predicted daily end use energy consumption}}{\text{average daily end use energy consumption}} \quad (3)$$

To calculate the ZNE daily energy budget for each end use, the average daily photovoltaic production, $PV_{budget,\ daily}$, and the fraction of daily budget allocated to each end use, $F_{end\ use}$, must be taken into account. These values are given by equations 4 and 5:

$$PV_{budget,\ daily} = \text{annual building PV production} / 365 \quad (4)$$

$$F_{end\ use} = \sum_{day=1}^n (\text{daily end use energy consumption})/n \quad (5)$$

The PV production for each building is found by weighting the total community annual PV production by each building's proportion of the community's annual electricity consumption.

To obtain the end use budgets for each day, equation 6 is used.

$$Budget_{end\ use,\ daily} = EF_{end\ use,\ daily} \times PV_{budget,\ daily} \times F_{end\ use} \quad (6)$$

Finally, to convert the daily energy budget for each end use, $Budget_{end\ use,\ daily}$, into hourly budgets, the daily budget is multiplied by the average percentage of energy that each end use consumes in each hour of the day. Separate percentages are used for weekdays and weekends. These values are given in Appendix C.

The model logic was coded in python for inclusion in the West Village Energy Dashboard. The python code for the model is included in Appendix D.

2.2.2 Data Sources

To develop the linear regression equations, end use energy consumption data collected from HOBO loggers in the West Village and hourly temperature data from the UC Davis PI database (tag aiTIT4045), are used. When running the model, hourly temperature predictions for the next seven days from Dark Sky (<https://darksky.net/dev>) are used. In addition, $PV_{budget,\ daily}$ and $F_{end\ use}$ values are used that were calculated from past HOBO logger data.

2.2.3 Model Development

The development of the model relies heavily on historic solar production and energy consumption by end use data. This information has been collected by HOBO loggers in each of the West Village buildings (1715, 1605, 1590, 215) and was provided by the client. The data provided had gaps in data collection and many outlier values due to challenges with the loggers. Aggregating and cleaning the data was time consuming and therefore was only conducted for building 1715. The data provided by the client for 1715 was: nov 2018, dec 2018, jan 2019, feb 2019, mar 2019, apr 2019, may 2019, jun 2019, jul 2019, *aug 2018*, sep 2019, oct 2019. A python script was developed to combine each month of data into one dataframe. To clean the data, duplicate readings from different files were removed and HOBO logger power readings in watts were converted to kWh for each minute, each hour, and each day. For building 1715, the values for September 19 to October 11 were removed because the data was missing.

2.2.3.1 Linear Regression

Based on findings from the literature review, it was decided that the energy consumption predictions that informed the greater model would be based on linear regression. These linear regression models were created in R and were used to predict the various end uses' daily consumption factors (the models' response variables). Based on the data available to the group, the explanatory variables that could be leveraged to predict the response variable included: heating degree day, cooling degree day, and whether it was a holiday, weekday, or if UC Davis classes were in session.

The strategy for determining the best model for each end use was to start by including all explanatory variables in a linear model. Based on the results of the initial model, variables that were shown to have low statistical significance to the response variable were removed and the model was run again. This iterative process was then repeated based on the statistical significance of explanatory variables and the adjusted R^2 value of the model. Beyond adding and subtracting variables, incremental changes in each model iteration also included taking the logarithm of the response variable, creating interaction variables between the different explanatory variables, and creating heating degree day and cooling degree day threshold "dummy variables" to allow the model to adjust based on different degree day thresholds. These thresholds were added upon inspection of the relationship between heating/cooling degree days and end use loads. To illustrate, an example plot showing HVAC load vs heating degree day is included in Appendix B. In the plot, it can be seen that above a certain threshold of heating degree day values, the relationship becomes more linear. Once the best (i.e. most predictive) model based on the explanatory variables available was determined to be found by the modeler, the process was complete. Results from these models used to predict daily end use consumption factors were then passed along to inform the other aspects of the overall budget model.

Through this process, it was found that the explanatory variables available were very poor predictors of Plug and Kitchen load end use consumption factors. This was determined based on very low adjusted R^2 values of all models iterated for these end uses as well as results showing non-statistically significant explanatory variables. In summary, results showed that weather (cooling and heating degree days) and occupancy indicators (weekend, weekday, school being in session) had very little effect on the plug and kitchen loads in West Village building 1715. In the end the group concluded to use the original model's (i.e. the existing model before this project) logic for kitchen and plug loads. The HVAC and lighting end use models, on the other hand, showed higher adjusted R^2 values and were therefore utilized. The resulting linear regression equations along with their adjusted R^2 values based upon data from 2018/2019 can be found in Appendix B.

2.2.3.2 Hourly Budgets

The hourly budgets were based on the historical energy consumption data collected for Building 1715 2018/2019. The data was collected throughout the year on an hourly-basis depending on end use (HVAC, kitchen, plug loads, and lighting). Microsoft Excel was used to organize and clean the data, which was then uploaded into the Python model. The hourly budgets were calculated using the following algorithm: first, the hourly consumption data depending on end use was set up in a way that would take weekdays

versus weekends and school in-session versus holidays into consideration. Second, the entire year was split up into its months. Third, heating degree days and cooling degree days were assigned to each day of the month. Fourth, the average consumption during any given hour throughout each day of the month was calculated depending on the end use. Extensive data cleaning was required in this step since the HOBO loggers used to track energy consumption produced a significant amount of values that were undefined. After cleaning the data to exclude undefined values for the entire twelve months of the year, the model then produced twenty-four averages (for 24 hours) for every hour of the day for each of the four end uses. The same calculation was iterated 12 times (for the twelve months). Fifth, all of these averages were then converted into percentages providing information on how energy is consumed depending on end use throughout every hour of the day in any given month.

3. Results

To determine if the developed model performs better than the existing model in daily predictions, the mean square error (MSE) for each model compared to measured values was calculated. The formula to calculate MSE is:

$$MSE = \frac{1}{n} \sum_{y=1}^n (Y_i - \widehat{Y}_i)^2 \tag{7}$$

Where Y_i is an observed value, \widehat{Y}_i is a predicted value, and n is the number of data points. MSE is a measure of how well a model predicts values. For this analysis, the MSE was calculated based on predicted energy use from each model, not ZNE budget. This was done so the values could be compared to actual end use consumption.

The mean square error was calculated for the 2018/2019 timeframe that was used to develop both the new and existing model. This MSE value is not as accurate because it is compared to the training dataset and a model may be overfitted to that data, meaning it is good at predicting the values in that dataset but not new values. Therefore the MSE was also calculated for May and June of 2018. An MSE value for a whole year of data was desired, but data quality issues limited the range to only two months. All calculated MSE values are given in Table 1.

Table 1: MSE values for HVAC and Lighting predictions

		MSE	
2018/2019	HVAC	Existing Model	555
		New Model	526
	Lighting	Existing Model	80
		New Model	153
5/18 and 6/18	HVAC	Existing Model	1220
		New Model	627
	Lighting	Existing Model	303
		New Model	70

In addition to calculating MSE, the predictions for each model and actual value for May and June of 2018 were graphed, shown in Figures 1 and 2.

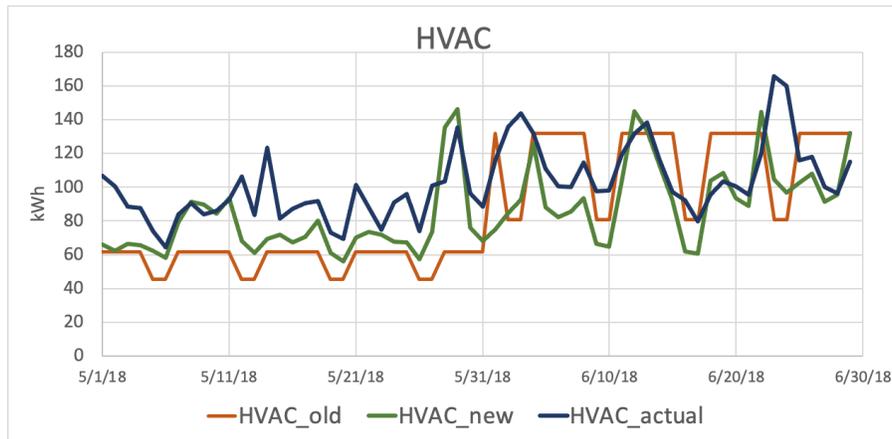


Figure 1: HVAC predictions for building 1715 from the existing (old) and new models compared to actual recorded values.

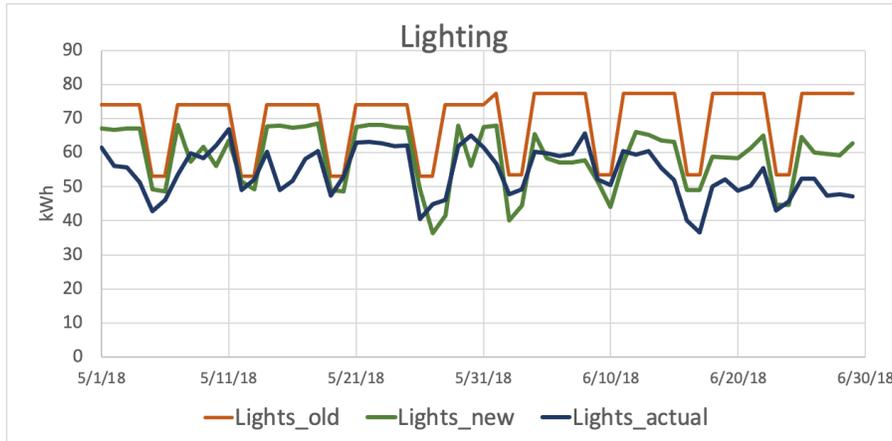


Figure 2: Lighting predictions for building 1715 from the existing (old) and new models compared to actual recorded values.

4. Discussion

The goal of this modeling effort was to accurately predict energy consumption of each end use so that those values could be utilized to develop ZNE budgets. The accuracy of each model to predict daily energy consumption for each end use was evaluated using MSE. The hourly predictions were not evaluated because the same methods were used in both models to obtain hourly predictions from daily predictions. For HVAC, the MSE for the new model outperformed the existing model. For Lighting, the MSE of the new model for May and June 2018 outperformed the existing model, but performed worse for the 2018/2019 timeframe. Our team is confident that the new model is a better predictor of HVAC energy consumption. For lighting, our team would have more confidence that the new model is better if we could test it on a larger dataset. For May and June 2018, the MSE values are drastically improved, which indicates the existing model is likely over-fitted to the training dataset and is a poor predictor of

data outside of the 2018/2019 timeframe. Though more data to evaluate the models would increase confidence, our team believes the new model is a better predictor for HVAC and lighting end uses. Because both models utilize energy predictions to calculate ZNE budgets, we also believe the new model leads to more accurate ZNE budgets. It's important to mention, however, uncertainties did exist, especially in the metered data. As previously mentioned, the metered data recorded values that were undefined, which makes it difficult to determine what was happening during those certain hours. Additionally, in the current model, a single year was only analyzed to calculate the hourly consumption data. More years would be needed to do a better job in capturing seasonality and determining any outliers in the data.

5. Recommendations and Conclusions

Since its inception, the West Village at UC Davis has served as a case study for ZNE community design and implementation. Like many ZNE buildings and communities, it has also served as a reminder that simply designing for ZNE does not mean it will be achieved. Data verification of ZNE status is an important tool that should be used to ensure best laid plans for ZNE are realized. Beyond understanding whether a facility or community achieves ZNE in a given year, researchers like our client, Dr. Angela Sanguinetti, are hoping that informing tenants in their building's ZNE performance can influence whether or not that building will reach ZNE status by the end of the year.

Building occupant engagement efforts have included the creation of the West Village Energy Dashboard, which informs West Village occupants of energy consumption budgets that should be met in a given hour, day, week, and year to achieve annual ZNE status. Before this project, these budgets were informed by a model that was built on inaccurate assumptions and lacked flexibility and accuracy. To address this, our group, informed by reviewing relevant literature in the field of ZNE modeling, built a new model. The model created is dynamic and offers a moving energy budget based on weather and occupancy indicators. Linear regression was leveraged to increase the accuracy of end use energy consumption factors that were the basis for creating new energy budgets. Model validation proved that the new model outperformed the old model significantly in terms of energy consumption predictions for both HVAC and lighting loads for Building 1715 in West Village. Kitchen and plug load predictions were not able to be improved based off the data available to the team, so the original model's logic was implemented for these end uses. Future work on implementing our model to the three other commercial buildings (1590, 1605 and 215) in West Village is recommended to calculate more accurate consumption predictions at the least for the HVAC and lighting loads. Overall, however, the team accomplished its goal of creating a more accurate energy model that can be applied to all the buildings at West Village and inform its Energy Dashboard.

6. Bibliography

Coleman, Michael J. et al. "Promoting Behaviour Change through Personalized Energy Feedback in Offices." *Building Research & Information*, vol. 41, no. 6, 2013, pp. 637-51, doi:10.1080/09613218.2013.808958.

Dobson, M. (2019, October 25). Introduction to Degree Days. Stark. <https://www.stark.co.uk/resources/news/introduction-degree-days/>.

Judd, Kathleen S. et al. "The Role of Occupant Behavior in Achieving Net Zero Energy: A Demonstration Project at Fort Carson." 2013-09-30 2013. doi:10.2172/1110475.

Kneifel, Joshua and David Webb. "Predicting Energy Performance of a Net-Zero Energy Building: A Statistical Approach." *Applied energy*, vol. 178, 2016, pp. 468-83.

Medrano, Marc et al. "Assessing the Nearly Zero-Energy Building Gap in University Campuses with a Feature Extraction Methodology Applied to a Case Study in Spain." *International Journal of Energy and Environmental Engineering*, vol. 9, no. 3, 2018, pp. 227-47, doi:10.1007/s40095-018-0264-x.

Sanguinetti, A., McMurry, R., Favetti, M., Huang, B., Sikand, S., Dombrovski, K., . . . Meier, A. 2018. "Keeping Track to Stay on Track for Zero Net Energy: Modeling Building and End Use Consumption Targets for a ZNE Community." In Proceedings of the ACEEE 2018 Summer Study on Energy Efficiency in Buildings.

7. Appendix

7.1. Appendix A

The existing ZNE budget model developed by Dr. Sanguinetti and team:

Calculating ZNE Budgets

- Step 1: Calculate **Community Annual ZNE Budget**. The annual ZNE budget for the community of West Village energy and transportation research centers (all four buildings combined) is equal to the amount of energy produced annually from the photovoltaic (PV) solar panels that are dedicated to those spaces.

$$\text{Community Annual ZNE Budget} = \sum(\text{Building}_{k(1-4)} \text{ Annual Production})$$

- Step 2: Calculate **Building Annual ZNE Budgets**. To calculate an annual ZNE budget for each building, we weighted the community annual ZNE budget by each building's proportion of the community's annual electricity consumption, using a year's worth of historical consumption data. This yielded an annual ZNE budget for each building that would be fair for the competitive elements of the dashboard, essentially creating a handicap factor to cancel out building (dis)advantages.

$$\text{Building}_k \text{ Annual ZNE Budget} = \text{Community Annual ZNE Budget} * \left(\frac{\text{Building}_k \text{ Annual Consumption}}{\text{Community Annual Consumption}} \right)$$

SIDEBAR: Below are two data tables that show calculations of building annual ZNE budgets based on two different timeframes. This can give you a sense of how the Production ranges. I think ideally, we would have a dynamic model that calculates the annual ZNE budgets based on the past 365 days of consumption and production, but if we can't do that, how valid is it to use the budgets created based on past data into the future?

Annual Production and Consumption for 2017-2018, and Annual ZNE Budgets

Building	Annual Consumption (MWh/yr)	Annual Production (MWh/yr)	ZNE Performance	Weight	ZNE Budget (163*Weight)	Pooled ZNE Performance
1590 Tilia	46.0	40.0	87%	46/174	43	93%
1605 Tilia	53.6	47.6	89%	53.6/174	50	93%
1715 Tilia	--	--	--			
215 Sage	74.4	74.9	99%	74.4/174	69.69	93%
Total	174	162.5	94%	N/A		

Annual Production and Consumption for 2018-2019, and Annual ZNE Budgets

Building	Annual Consumption	Annual Production	ZNE	Weight	ZNE Budget	Pooled ZNE
----------	--------------------	-------------------	-----	--------	------------	------------

	(MWh/yr)	(MWh/yr)	Performance		(187*Weight)	Performance
1590 Tilia	46.1	34.4	75%	46.1/244.4	35.2	76%
1605 Tilia	49.3	46.1	94%	49.3/244.4	37.7	76%
1715 Tilia	71.9	37.8	53%	71.9/244.4	55	76%
215 Sage	77	68.6	89%	51.8/244.4	39.6	76%
Total	244.4	187	76.5%			

- Step 3: Calculate **Building Weekday and Weekend Hourly ZNE Budgets for Each Month**. In order to convey ZNE building performance throughout the year (e.g., weekly, daily, and even hourly), we created ZNE budgets for every hour of the day, separately for weekdays and weekend days, for each month, for each building (a total of 48 hourly budgets for each month*building, 24 for weekdays and 24 for weekends). We used the same weighting logic, this time weighting each building's annual ZNE budget by the proportion of consumption for each hour of weekday/weekend*month relative to the building's annual consumption, again using a year's worth of historical consumption data. These hourly ZNE budgets are used in the building dashboard weekly leaderboard and heat map 24-hour and weekly views. Hourly ZNE budgets are summed up for daily ZNE budgets, used in the heat map monthly and annual views.

$$\text{Building}_k \text{ Month}_k \text{ (Jan-Dec) Weekday Hour}_k \text{ (0-24) ZNE Budget} = \text{Building}_k \text{ Annual ZNE Budget} * \left(\frac{\text{Building}_k \text{ Month}_k \text{ Mean Weekday Hour}_k \text{ (0-24) Consumption}}{\text{Building}_k \text{ Annual Consumption}} \right)$$

$$\text{Building}_k \text{ Month}_k \text{ Weekend Day Hour}_k \text{ ZNE Budget} = \text{Building}_k \text{ Annual ZNE Budget} * \left(\frac{\text{Building}_k \text{ Month}_k \text{ Mean Weekend Day Hour}_k \text{ (0-24) Consumption}}{\text{Building}_k \text{ Annual Consumption}} \right)$$

- Step 4: Calculate **Building End Use ZNE Budgets**. Using the same logic as Steps 2 and 3, we created annual, and hourly ZNE budgets for each end use in each building for each month (separate for weekdays and weekends). These goals are used in the heat map visualizations on the building dashboards.

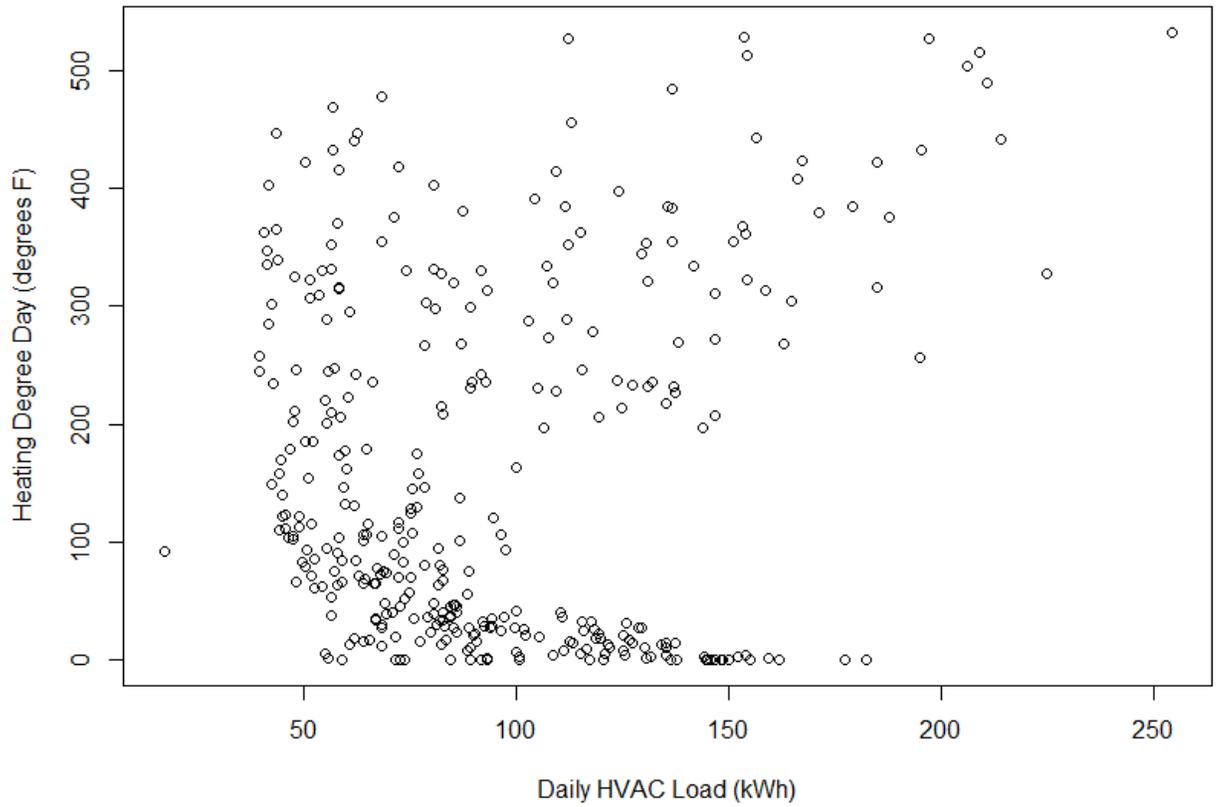
$$\text{Building}_k \text{ End Use}_k \text{ Annual ZNE Budget} = \text{Building}_k \text{ Annual ZNE Budget} * \left(\frac{\text{Building}_k \text{ End Use}_k \text{ Annual Consumption}}{\text{Building}_k \text{ Annual Consumption}} \right)$$

$$\text{Building}_k \text{ Month}_k \text{ End Use}_k \text{ Weekday Hour}_k \text{ ZNE Budget} = \text{Building}_k \text{ End Use}_k \text{ Annual ZNE Budget} * \left(\frac{\text{Building}_k \text{ Month}_k \text{ End Use}_k \text{ Mean Weekday Hour}_k \text{ Consumption}}{\text{Building}_k \text{ End Use}_k \text{ Annual Consumption}} \right)$$

$$\text{Building}_k \text{ Month}_k \text{ End Use}_k \text{ Weekend Day Hour}_k \text{ ZNE Budget} = \text{Building}_k \text{ End Use}_k \text{ Annual ZNE Budget} * \left(\frac{\text{Building}_k \text{ Month}_k \text{ End Use}_k \text{ Mean Weekend Day Hour}_k \text{ Consumption}}{\text{Building}_k \text{ End Use}_k \text{ Annual Consumption}} \right)$$

7.2. Appendix B

The following plot shows the relationship between heating degree day and HVAC load. After a certain threshold, the relationship appears more linear:



The developed linear regression equations for HVAC and lighting in building 1715 are:

```
call:
lm(formula = (HVACfactor) ~ hddlimit * weekday * cdd * hdd, data = Cleaned1715)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.76864 -0.12870 -0.01501  0.12474  1.05981
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.608e-01  1.875e-01   2.458 0.014492 *
hddlimitTRUE      8.072e-02  2.528e-01   0.319 0.749670
weekdayTRUE     -5.977e-02  2.128e-01  -0.281 0.778976
cdd               2.211e-02  7.039e-02   0.314 0.753585
hdd               3.368e-04  5.411e-04   0.623 0.534036
hddlimitTRUE:weekdayTRUE  2.484e-01  2.868e-01   0.866 0.387173
hddlimitTRUE:cdd  -2.106e-02  7.039e-02  -0.299 0.764959
weekdayTRUE:cdd   4.489e-01  1.294e-01   3.469 0.000592 ***
hddlimitTRUE:hdd  -5.064e-04  1.365e-03  -0.371 0.710901
weekdayTRUE:hdd   2.509e-03  6.158e-04   4.074 5.8e-05 ***
cdd:hdd          -1.003e-04  3.447e-04  -0.291 0.771245
hddlimitTRUE:weekdayTRUE:cdd -4.478e-01  1.294e-01  -3.461 0.000610 ***
hddlimitTRUE:weekdayTRUE:hdd -2.503e-03  1.567e-03  -1.597 0.111297
hddlimitTRUE:cdd:hdd  9.954e-05  3.448e-04   0.289 0.773017
weekdayTRUE:cdd:hdd -1.819e-03  5.603e-04  -3.246 0.001291 **
hddlimitTRUE:weekdayTRUE:cdd:hdd  1.799e-03  5.604e-04   3.210 0.001459 **
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.25 on 326 degrees of freedom
Multiple R-squared:  0.6748,    Adjusted R-squared:  0.6599
F-statistic:  45.1 on 15 and 326 DF,  p-value: < 2.2e-16
```

```
call:
lm(formula = log(Lightsfactor) ~ cddlimit * (holiday + hdd +
  weekday) + hddlimit * (holiday + hdd + weekday), data = Cleaned1715)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.69172 -0.13230  0.01705  0.16084  0.40781
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.320114   0.116136  -2.756  0.00617 **
cddlimitTRUE       0.177677   0.109702   1.620  0.10627
holidayTRUE      -0.575445   0.303871  -1.894  0.05914 .
hdd              -0.002109   0.002524  -0.836  0.40399
weekdayTRUE       0.497745   0.092005   5.410 1.22e-07 ***
hddlimitTRUE      0.050961   0.105804   0.482  0.63037
cddlimitTRUE:holidayTRUE  0.231879   0.293380   0.790  0.42988
cddlimitTRUE:hdd    0.001656   0.002517   0.658  0.51105
cddlimitTRUE:weekdayTRUE -0.167479   0.087183  -1.921  0.05560 .
holidayTRUE:hddlimitTRUE  0.105416   0.220717   0.478  0.63325
hdd:hddlimitTRUE  -0.003354   0.002038  -1.646  0.10082
weekdayTRUE:hddlimitTRUE -0.078058   0.077253  -1.010  0.31304
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.2035 on 328 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.433,    Adjusted R-squared:  0.414
F-statistic: 22.77 on 11 and 328 DF,  p-value: < 2.2e-16
```

7.3. Appendix C

The average percentage of energy that each end consumes in each hour of the day for building 1715 are given in the table below. The hour 0 corresponds to 12:00 AM.

	Unnamed: 0	HVAC	Kitchen	Lights	Plugs	HVAC_wknd	Kitchen_wknd	Lights_wknd	Plugs_wknd
0	0	0.005246	0.016336	0.042274	0.034157	0.017537	0.040651	0.043678	0.041492
1	1	0.004848	0.015588	0.039341	0.034348	0.015687	0.038388	0.046595	0.039888
2	2	0.004689	0.036323	0.005703	0.041106	0.014220	0.052706	0.039919	0.041274
3	3	0.004531	0.035105	0.005287	0.036804	0.014016	0.041194	0.034330	0.039934
4	4	0.004082	0.031623	0.005185	0.034240	0.013953	0.041953	0.036481	0.040544
5	5	0.016219	0.125650	0.004871	0.035086	0.013918	0.041336	0.032303	0.040143
6	6	0.054164	0.419607	0.004732	0.044239	0.013321	0.041150	0.027732	0.039644
7	7	0.055573	0.015907	0.039323	0.041020	0.040322	0.043256	0.031892	0.040797
8	8	0.052264	0.014756	0.041922	0.041483	0.041303	0.032995	0.031437	0.044498
9	9	0.056096	0.019805	0.053348	0.044820	0.050093	0.033995	0.031234	0.042495
10	10	0.065888	0.018903	0.061107	0.047978	0.052282	0.036680	0.037791	0.042099
11	11	0.068376	0.018997	0.062682	0.049970	0.055773	0.040138	0.038060	0.041187
12	12	0.075182	0.025292	0.061830	0.051876	0.056502	0.048742	0.043206	0.041310
13	13	0.077964	0.027292	0.062234	0.053055	0.064764	0.037762	0.044848	0.042079
14	14	0.081316	0.023395	0.061384	0.050577	0.067794	0.040634	0.047478	0.040015
15	15	0.083818	0.018172	0.058306	0.049426	0.071741	0.069202	0.047226	0.042416
16	16	0.082280	0.017175	0.055407	0.046438	0.077138	0.036732	0.043948	0.042336
17	17	0.079177	0.016310	0.052986	0.041867	0.071415	0.040082	0.048442	0.042572
18	18	0.029336	0.015944	0.046549	0.037733	0.062650	0.045178	0.049872	0.042660
19	19	0.030850	0.017823	0.044568	0.037368	0.061521	0.042937	0.051503	0.042105
20	20	0.022062	0.018566	0.049987	0.037096	0.048320	0.038413	0.051013	0.043466
21	21	0.015438	0.016897	0.050178	0.038978	0.039662	0.040329	0.051092	0.043326
22	22	0.015231	0.017510	0.046838	0.035638	0.021770	0.040213	0.046291	0.041775
23	23	0.015371	0.017022	0.043960	0.034696	0.014300	0.035335	0.043629	0.041946

7.4. Appendix D

Model Logic in Python code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.tseries.holiday import USFederalHolidayCalendar
```

In [2]:

```
## utilize existing model to get the annual budget for each buidling in kWh
annual_1715 = 55.01350245*1000
#annual_1590 = 35.27291326*1000
#annual_1605 = 37.72135843*1000
#annual_215 = 58.91571195*1000
```

In [3]:

```
## get average daily PV budget for each building in kWh
day_avg_1715 = annual_1715/365
```

In [4]:

```
# get average daily end use budget for each building in kWh

# % for each end use based on Aug 2018 through Jul 2019
avg_pct_1715_hvac = 0.477918249531868 #all 1715 calcs exclude 9/18 through 10/12 data
avg_pct_1715_kitchen = 0.0221182004710866
avg_pct_1715_light = 0.295831392155982
avg_pct_1715_plug = 0.204132157841063
```

In [5]:

```
#read in hourly weather data for the the week
from google.colab import files
uploaded = files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving sample_OAT.csv to sample_OAT.csv

In [6]:

```
import io
future_temp = pd.read_csv(io.BytesIO(uploaded['sample_OAT.csv']))
```

In [7]:

```
#format temperature data
future_temp["Timestamp"]=pd.to_datetime(future_temp["Timestamp"], utc=True) #make index date time
future_temp.set_index("Timestamp", inplace=True)
future_temp["aiTIT4045"] = pd.to_numeric(future_temp["aiTIT4045"],errors='coerce') #make correct datatype
```

In [8]:

```
#get info from data
pred_year = future_temp.index[1].year #get the year we are predicting budgets for
```

In [10]:

```
# import baseline predictions for the year
#there is another python script to generate this file that should be used to make this file more dynamically
from google.colab import files
uploaded = files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving old_model_budgets_1715_2021.csv to old_model_budgets_1715_2021.csv

In [11]:

```
import io
budget = pd.read_csv(io.BytesIO(uploaded['old_model_budgets_1715_2021.csv']))
```

In [28]:

```
#read in hourly weather data for the past before the week being predicted
#in the future will need to make an API connection to the pi database, for now use a manual file upload
# in this example we upload hourly temp data Jan 1 through May 2
from google.colab import files
uploaded = files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

In [14]:

```
import io
past_temp = pd.read_csv(io.BytesIO(uploaded['previous_2021_OAT.csv']))
```

In [15]:

```
#format temperature data
past_temp["Timestamp"]=pd.to_datetime(past_temp["Timestamp"], utc=True) #make index date time
past_temp.set_index("Timestamp", inplace=True)
past_temp["aiTIT4045"] = pd.to_numeric(past_temp["aiTIT4045"],errors='coerce') #make correct datatype
```

In [16]:

```
#combine all temp data into one dataframe
temp = past_temp.append(future_temp)
```

In [17]:

```
#remove duplicates
temp = temp[~temp.index.duplicated(keep='last')]
```

In [18]:

```
#calculate cdd and hdd from temperature data
temp["diff"] = temp["aiTIT4045"]-65
temp["cdd"] = temp.loc[temp["diff"] > 0, "diff"]
temp["hdd"] = temp.loc[temp["diff"] <= 0, "diff"]
temp["hdd"] = temp["hdd"]*-1
temp = temp.resample("D").sum()
```

In [19]:

```
#include only useful columns
temp.drop(columns=["aiTIT4045", "diff"], inplace=True)
```

In [20]:

```
#classify each day in the week - school/noschool, weekend/weekday, holiday/noholiday
```

```
def is_school(i):
    q_su18 = (i > pd.Timestamp("2018-06-25")) & (i < pd.Timestamp("2018-09-15"))
    q_f = (i > pd.Timestamp("2018-09-24")) & (i < pd.Timestamp("2018-12-08"))
    q_w = (i > pd.Timestamp("2019-01-07")) & (i < pd.Timestamp("2019-03-16"))
    q_sp = (i > pd.Timestamp("2019-04-01")) & (i < pd.Timestamp("2019-06-07"))
    q_su19 = (i > pd.Timestamp("2019-06-24")) & (i < pd.Timestamp("2019-09-14"))
    school = q_w | q_sp | q_su18 | q_f | q_su19
    return school
```

```
def is_weekday(i):
    return i.dayofweek // 5 == 0 # equivalent to i.dayofweek < 5
```

In [21]:

```
temp.index = temp.index.tz_localize(None) # tz_aware --> tz_naive
temp["school"] = is_school(temp.index)
temp["weekday"] = is_weekday(temp.index)
```

In [22]:

```
cal = USFederalHolidayCalendar()
holidays = cal.holidays(start=temp.index.min(), end=temp.index.max())
temp["holiday"] = temp.index.isin(holidays)
```

In [23]:

```
#utilize linear regression to calculate the energy factor for each day in the week for each end use
and also total
#add columns to temp dataframe: hvac_ef, light_ef, kitchen_ef, plug_ef
```

```
temp["hdd_lim_hvac"] = temp["hdd"] < 185
```

In [24]:

```
#linear regression equation for HVAC
#lighting linear regression equation needs to be input in the future
temp["hvac_ef"] =
0.460846226321002+(0.080724243377688*temp["hdd_lim_hvac"])-(0.0597729876288952*temp["w
eekday"])+(0.0221138821784112*temp["cdd"])+(0.000336844583232445*temp["hdd"])+(0.24837176
9331151*temp["hdd_lim_hvac"]*temp["weekday"])-
(0.0210618352410019*temp["cdd"]*temp["hdd_lim_hvac"])+(0.448886014572328*temp["weekday"]*t
emp["cdd"])-(0.000506370685544067*temp["hdd"]*temp["hdd_lim_hvac"])+(0.00250904388724317*t
emp["weekday"]*temp["hdd"])-(0.000100309969562149*temp["cdd"]*temp["hdd"])-(0.447821031678
69*temp["hdd_lim_hvac"]*temp["weekday"]*temp["cdd"])-(0.0025026499525651*temp["hdd_lim_hva
c"]*temp["weekday"]*temp["hdd"])+(0.0000995371331836865*temp["hdd_lim_hvac"]*temp["hdd"]*te
mp["cdd"])-(0.00181900859623164*temp["weekday"]*temp["cdd"]*temp["hdd"])+(0.00179899888128
381*temp["hdd_lim_hvac"]*temp["weekday"]*temp["cdd"]*temp["hdd"])
```

In [25]:

```
#apply energy factor to get the kWh budget for each day in the week for each end use and also total
temp["hvac"] = temp["hvac_ef"]*avg_pct_1715_hvac*day_avg_1715
#temp["light"] = temp["light_ef"]*avg_pct_1715_light*day_avg_1715
#temp["kitchen"] = temp["kitchen_ef"]*avg_pct_1715_kitchen*day_avg_1715
#temp["plug"] = temp["plug_ef"]*avg_pct_1715_plug*day_avg_1715
```

In [26]:

```
temp = temp.resample("H").ffill()
```

In [29]:

```
# get hourly kWh budgets for each day in the week
from google.colab import files
uploaded = files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving hourly_budgets.csv to hourly_budgets.csv

In [30]:

```
import io
hourly = pd.read_csv(io.BytesIO(uploaded['hourly_budgets.csv']))
```

In [31]:

```
#apply hourly budgets
temp["hvac_hr"] = 0
for i in temp.index:
    hour = i.hour
    if i.dayofweek // 5 == 0:
        temp.loc[i, "hvac_hr"] = hourly.loc[hour, "HVAC"]*temp.loc[i, "hvac"]
    else:
        temp.loc[i, "hvac_hr"] = hourly.loc[hour, "HVAC_wknd"]*temp.loc[i, "hvac"]
```

In [32]:

```
#output is temp dataframe with ZNE budget for HVAC as temp["HVAC_hr"]
temp
```

Out [32]:

	cdd	hdd	school	weekday	holiday	hdd_lim_hvac	hvac_eff	hvac	hvac_hr
Timestamp									
2021-01-01 00:00:00	0.00000	430.875302	False	True	True	False	1.627296	117.218641	0.614950
2021-01-01 01:00:00	0.00000	430.875302	False	True	True	False	1.627296	117.218641	0.568307
2021-01-01 02:00:00	0.00000	430.875302	False	True	True	False	1.627296	117.218641	0.549604

2021-01-01 03:00:00	0.00000	430.8753 02	False	True	True	False	1.6272 96	117.2186 41	0.5311 65
2021-01-01 04:00:00	0.00000	430.8753 02	False	True	True	False	1.6272 96	117.2186 41	0.4784 88
...
2021-05-09 20:00:00	307.524 67	0.000000	False	False	False	True	0.8651 01	62.31560 1	3.0110 61
2021-05-09 21:00:00	307.524 67	0.000000	False	False	False	True	0.8651 01	62.31560 1	2.4715 61
2021-05-09 22:00:00	307.524 67	0.000000	False	False	False	True	0.8651 01	62.31560 1	1.3566 29
2021-05-09 23:00:00	307.524 67	0.000000	False	False	False	True	0.8651 01	62.31560 1	0.8910 85
2021-05-10 00:00:00	8.51898	0.000000	False	True	False	True	0.7482 04	53.89521 1	0.2827 44

3097 rows × 9 columns