My project consists of primarily two data sets: (1) electricity data from the EIA, which include electricity demand, interchange, and generation and (2) hourly, local climatological data (LCD) downloaded from NOAA. To limit the scope of the project, I retrieved both data sets from the city of Los Angeles exclusively (specifically, Los Angeles Department of Water and Power data from EIA and the Los Angeles Downtown LCD data). Electricity data were retrieved using EIA's API and then unpacked into a pandas `DataFrame`. Since LCD data are not available via NOAA's API, I manually downloaded LCD data from the NOAA website as a CSV file which I imported to a pandas `DataFrame`. Electricity data contain hourly entries from July 2015 to present. To align the data correctly, I downloaded the LCD data from July 2015 to September 2018 and later cut the electricity data to fit the date range.

The electricity data required some cleaning--there were very few null values in the set, however there were a very small number of outliers where data was either entered in as zero or a large negative number. Removing outliers cut only ~.01% of the dataset. LCD data on the other hand required more cleansing. Throughout the cleaning of the LCD data, I frequently referred to the LCD documentation for guidance on how to treat the features. Firstly, all features (except daily cooling/heating degrees) that did not record hourly entries, i.e., monthly and daily records, were dropped. I kept hourly heating and cooling degrees because they are very relevant to energy consumption. I used `bfill` to fill the the hourly entries of a specific day with that day's heating/cooling degrees. Additionally I added hourly heating and cooling degrees to get higher granularity on a metric which is relevant to energy consumption. The hourly heating/cooling degree days are the number of degrees below/above 65 degrees Fahrenheit (negative values are set to zero) and represents the amounting heating/cooling necessary for that temperature. I dropped a number of hourly columns with excessive null entries compared to the size of the data set and those with low predictive power. I dropped hourly pressure tendency and change because we of excess null values and because measures of pressure were already present, hourly wind direction and gust speed for the same reasons, and hourly present weather type because of excess null values and the dispersity of possible values, indicating low predictive power.

One of the main challenges in cleaning the LCD data was that there were in some cases multiple entries for the same hour. I wanted to have just one entry per hour such that I could eventually align LCD data with the hourly entries in the electricity data. I wrote a function that goes through the weather data and for each instance where there are multiple hourly entries, I keep the hourly entry closest to the hour and drop the remaining rows. For instance, if there are entries at 11:05, 11:36, and 11:56, I kept the entry corresponding to 11:05 and dropped the other rows. I performed the cleaning this way because the data closest to the hour is most representative for that hour. I then renamed the date such that the minute column is set to 00 to once again align with the electricity data. Either way, the values for multiple, per-hour entries are very similar, so the choice of which entry to keep doesn't make a huge difference.

The hourly sky condition feature required additional cleaning. The hourly sky condition feature contains information about each cloud layer (up to three) using six different categories to

describe the cloud coverage. As stated in the documentation, the full state of the sky can be best determined the the category of the last (i.e., highest) layer. With this in mind, I replaced the sky condition feature with the condition of the upper most layer to make the feature consistent. This feature also included some information about the cloud layer height which I excluded because these data were only present for a small number of entries. Finally, there were a number of features of mixed float and string types. After looking at their unique values, I saw that some values were string and floats arbitrarily. I simply converted strings to floats to make the features consistent.

Next we deal with null values and outliers. To determine whether to fill null values with median of `ffill`, I plotted histograms, violin plots, and printed stats for each feature. Features where the median was close the mean suggests few outliers and that the median is an appropriate statistic to fill null entries. Conversely, null values in features with outliers present were filled using `ffill`, assuming that previous values have some predictive value for subsequent values since we are dealing with time-stream data. For outliers, because there was not a single feature where the presence of outliers significantly shifted the median from the mean and because the data is such high granularity, outliers were kept without further cleaning. Lastly, we cut both data sets to fit the same date range as mentioned above, and combine the electricity and weather data into a single `DataFrame` on date index. We are left with ~27K data points from July 2015 to September 2018 to perform statistical analysis and predictive modeling.