# Data Exploration

The dataset is comprised of a semi-contiguous observation period spanning from July 6, 2017 to August 6, 2018 (two days of data spanning July 31, 2018 to August 1, 2018 are missing). Null values seem to be represented as zero integers, will replace with NaN’s (excluding the fuel type Boolean columns).

The time series data can be broken up frames, contiguous periods of engine activity (as judged by Fuel Consumption) and other contagious periods of engine inactivity presumably while in port or otherwise not collecting data. I was initially concerned about the ‘spikes’ present in this data, but they generally contain multiple data points and are very well correlated with other readings like engine RPM and Speed through water (not shown) so I have elected not to drop or smooth them.



**Figure 1** Plot of Main Engine Fuel Consumption over the dataset. The red vertical lines denote the frame boundaries we will be using to separate valid from invalid data collection period’s.

The dataset has been hour level qualified and our target variable, Main Engine Fuel Consumption, represents the averaged burn rate over the hour. Because of the aggregate nature of the data, fuel consumption measurements at the beginning and end of voyages have been contaminated by having the engine off for an indeterminate period of time and will be excluded from analysis.

Stopping periods will similarly be excluded, as they will lead to overestimates in true model performance. These data filtering steps reduce the samples in our data set from the original 9.4K down to 6.3K points, a ~33% reduction.

# Train Test Split Design

The model should be fit and evaluated using separate data collection intervals to try and prevent the high degree of temporal correlation within adjacent data points from leading us to have over-confidence in the performance of our model.

Partitioning training and testing data by data collection makes it difficult however to ensure our model will experience the full range of possible input conditions. Will need to use cross validation to fully evaluate model performance across diverse input conditions.

# Feature Engineering

The goal of this exercise can be interpreted in one of two subtly different ways:

1. Build a regression model for tasks like route optimization that must project fuel consumption into the future. With this objective, data from future time points in the same voyage cannot be used in feature engineering but past data points are fair game.
2. Build a deconvolutional model to isolate the effects of independent variables like draft and wind from fuel burn estimates. This type of model would be optimal for quantitative evaluation of vessel performance (vessel optimization). In this case data from past and future time points within the same voyage can be leveraged to best eliminate environmental noise.

I have elected to build the following additional features:

1. Difference in compass heading between current and previous (1 hour pior) data point – presumably turning will impact vessel drag and fuel burn, and may indicate if in port or not. The heading deltas must be normalized to lie within the range +/- pi radians (turning the ship to the left or to right is ambiguous in this dataset without making assumptions about rudder position). Because left turns and right turns will be treated equally we will take the Sin transformation of the heading delta normalizing our feature between +/-1.
2. Trim as recommended by the appendix, thanks! Draft measurements for the first two trips are completely missing and the third are partially missing. We will drop data collection periods zero and one from the data and back fill the missing observations in data collection period 2 with the first valid measurement.
3. Sine transformation of the average rudder deflection angle over the hour. May also be indicative of turning or lateral environmental forces.
4. Continuous rudder deflections without change in heading may be indicative of lateral ocean currents or winds, will use the correlation value (rudder\_deflections(radians)-mean(rudder\_deflection)) /(compass\_heading\_deltas (radians) – mean(compass\_heading\_deltas)) as a feature.
5. Water depth readings of zero will be replaced by the max water depth value as indicated in the appendix.
6. A number of ​water surface temperature readings indicate absolute zero. Will replace those readings with linearly interpolated values between the nearest valid data points.
7. True wind direction relative to compass heading will be used to measure wind; the ships bow and vector of motion are not necessarily synonyms as currents or wind could be driving some lateral movement. To capture wind direction we will need to take both the sine and cosine components of this angle.
8. True wind speed is all that matters in regards to fuel burn so we will use this value as is. Again a few anomalous zero readings here, will fill them with a linear interpolation between points.
9. Some data points were missing Shaft Speed (RPM), and some were missing Shaft Power (kW). We will impute missing Shaft Power and RPM values from the known linear relationship between Shaft Speed, RPM and Power.

In my first Fuel Consumption modeling attempt I included Engine Shaft Power (a direct measure of engine rate of work) in the feature set and produced a very well fitting model. This was of course a silly thing to do; work done by the engine should be directly related to fuel consumption. A simple linear regression model using Shaft Power alone as feature provides a very robust fuel consumption “estimation” with regression coefficient (R2 = 0.99).



**Figure 2.** Correlation between model predicted and true fuel consumption on the holdout set. Coefficient of determination of the fit is 0.99 +/- 0.01 std as estimated from 5 fold cross validation. Redline indicating a perfect fit.

This model provides very good estimate of fuel burn but is otherwise useless, (if we can measure shaft power, absent a broken fuel gauge, we could have just measured fuel consumption directly). Even more concerning, any model trained with a Shaft Power parameter would simply learn to ignore the provided environmental features.

To be useful model for goal number two we would like to learn a model that is most sensitive to its environment, put another way using as few engine related parameters as possible. The difference between speed through the water and speed over the ground should tell us a lot about how hard the engine is working without showing all of our engine cards; lets use that in place of the provided RPM, Torque, and Power so we will (hopefully) be more sensitive to the enviornment.