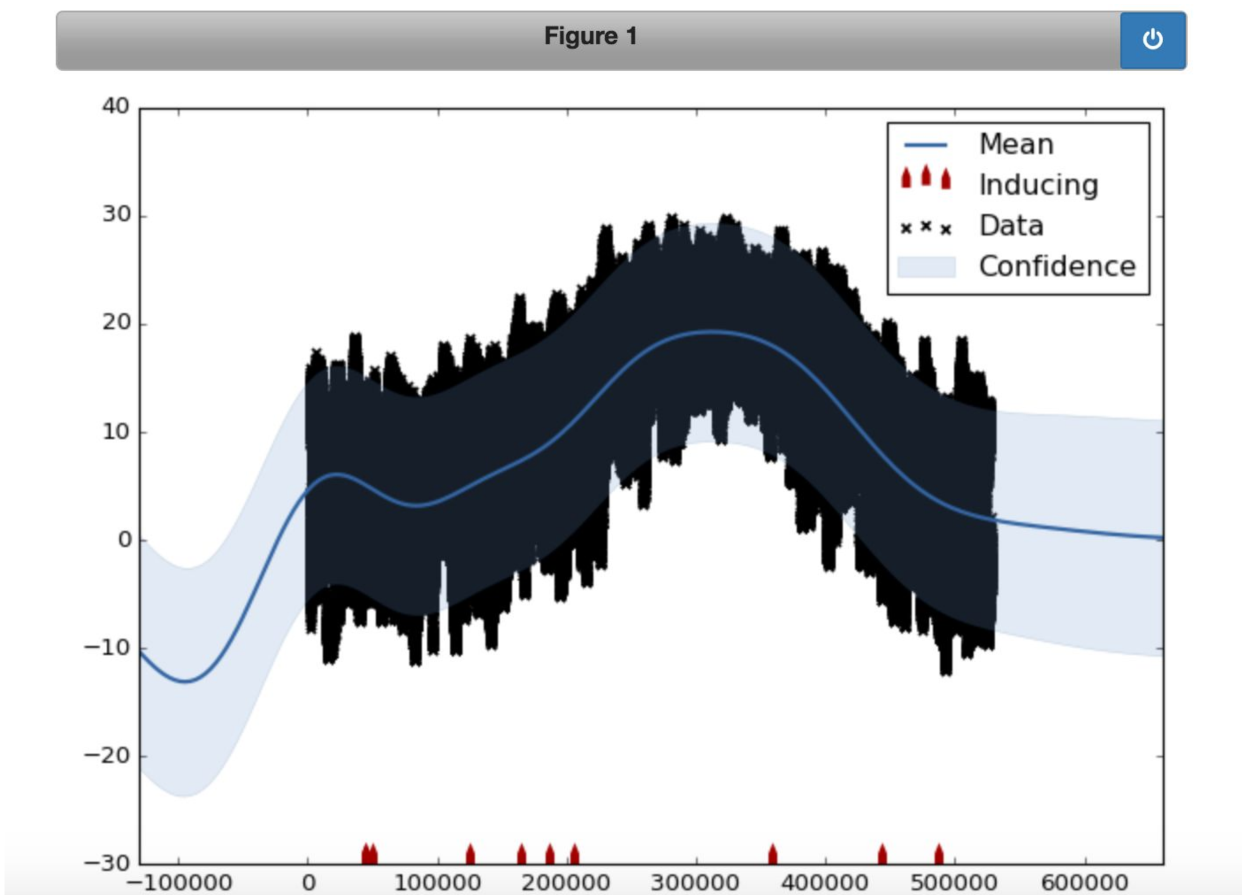
Sparse Gaussian Process

This project uses the temperature data (yearly weather) from Yosemite Village. This data covers 6 years, so I split the data into a training set of the first 5 years, and a testing set of the 6th year.

1/ I used a sparse Gaussian process to estimate the temperature over time of day (0:00-23:59) and time of year (1-365).

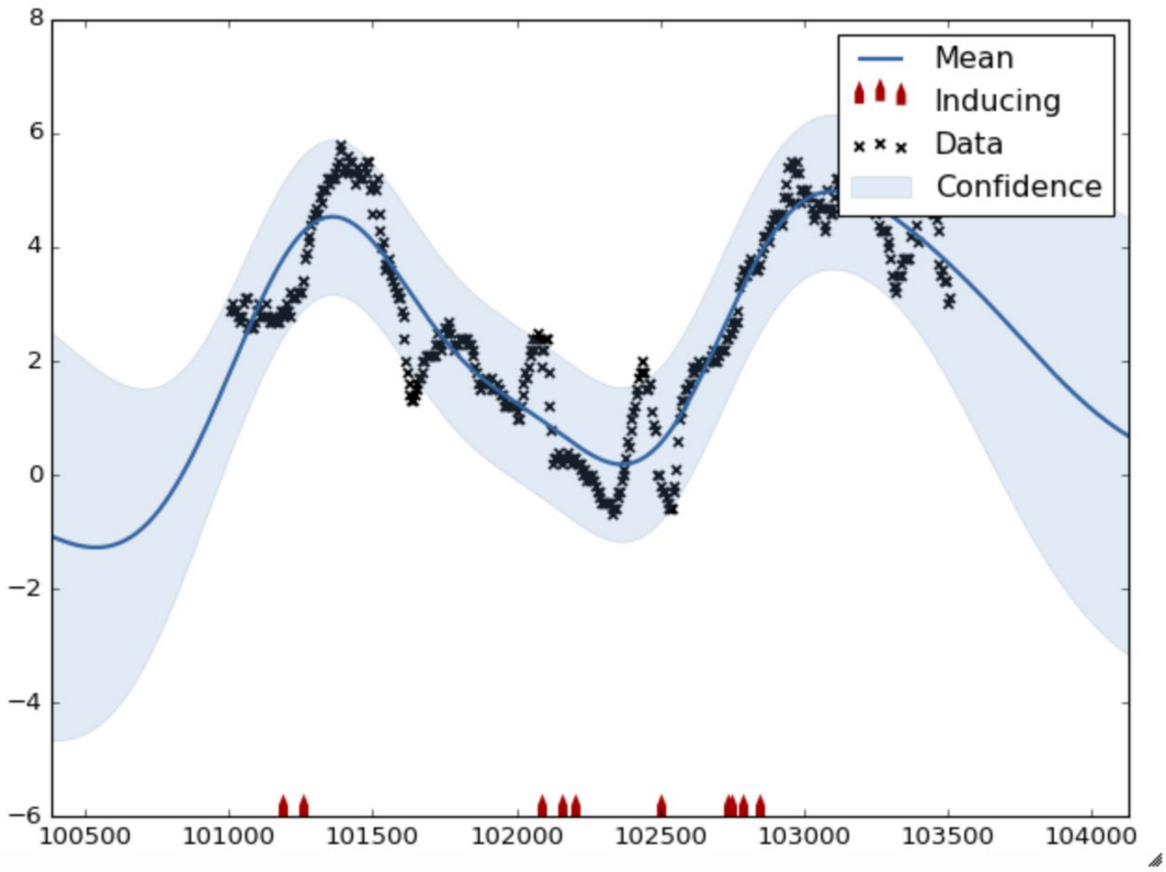
The covariance function I chose is an RBF function with a length scale of 10000.. After I ran optimize(), the lengthscale was updated to the closest local optimal solution which is 45860.8864 and the variance and Gaussian_noise.variance are also updated. Those are the three hyperparameters that were fit. The value 45860.8864 for the lengthscale makes sense because with this lengthscale, the graph would be pretty smooth, and there is a strong correlation between data points that are close to each other, and the covariance will decrease when the difference increases. It means that we have medium or high correlations until the point c*sqrt(45860)



```
Name : sparse_gp
Objective : 1608268.1098
Number of Parameters : 13
Number of Optimization Parameters : 3
Updates : True
Parameters:
  sparse_gp.                 |          value  |  constraints  |  priors
  inducing_inputs            |        (10, 1)  |     fixed     |
  rbf.variance               |  298.650570343  |      +ve      |
  rbf.lengthscale            |  163385.284028  |      +ve      |
  Gaussian_noise.variance    |  26.5369665518  |      +ve      |
Run time: 90.98

Mean squared error: 25.26
```

Running it on one section of the data set:



2/ I measured the mean squared error (MSE) of myGaussian process regression, and contrasted it with the MSE of my linear parameter model (on another project). I compared and contrasted the training time of the linear parameter model with the training time of the Gaussian process model:

```
Name : sparse_gp
Objective : 1607349.04234
Number of Parameters : 13
Number of Optimization Parameters : 3
Updates : True
Parameters:
  sparse_gp.                 |           value  |  constraints  |  priors
  inducing_inputs            |         (10, 1)  |     fixed     |
  rbf.variance               |  0.728121491572  |      +ve      |
  rbf.lengthscale            |   45860.8864747  |      +ve      |
  Gaussian_noise.variance    |   26.3933398326  |      +ve      |
Run time: 61.27

Mean squared error: 25.26
```

	Run time	Mean Squared Error
Gaussian process regression	61.27	25.26
Linear parameter regression	0.85	23.85

We can see that the runtime of Gaussian process regression is much longer than that of Linear paramater regression. The mean squared error is around the same.

3/ I made a 3D plot showing temperature as a function of (day, time):