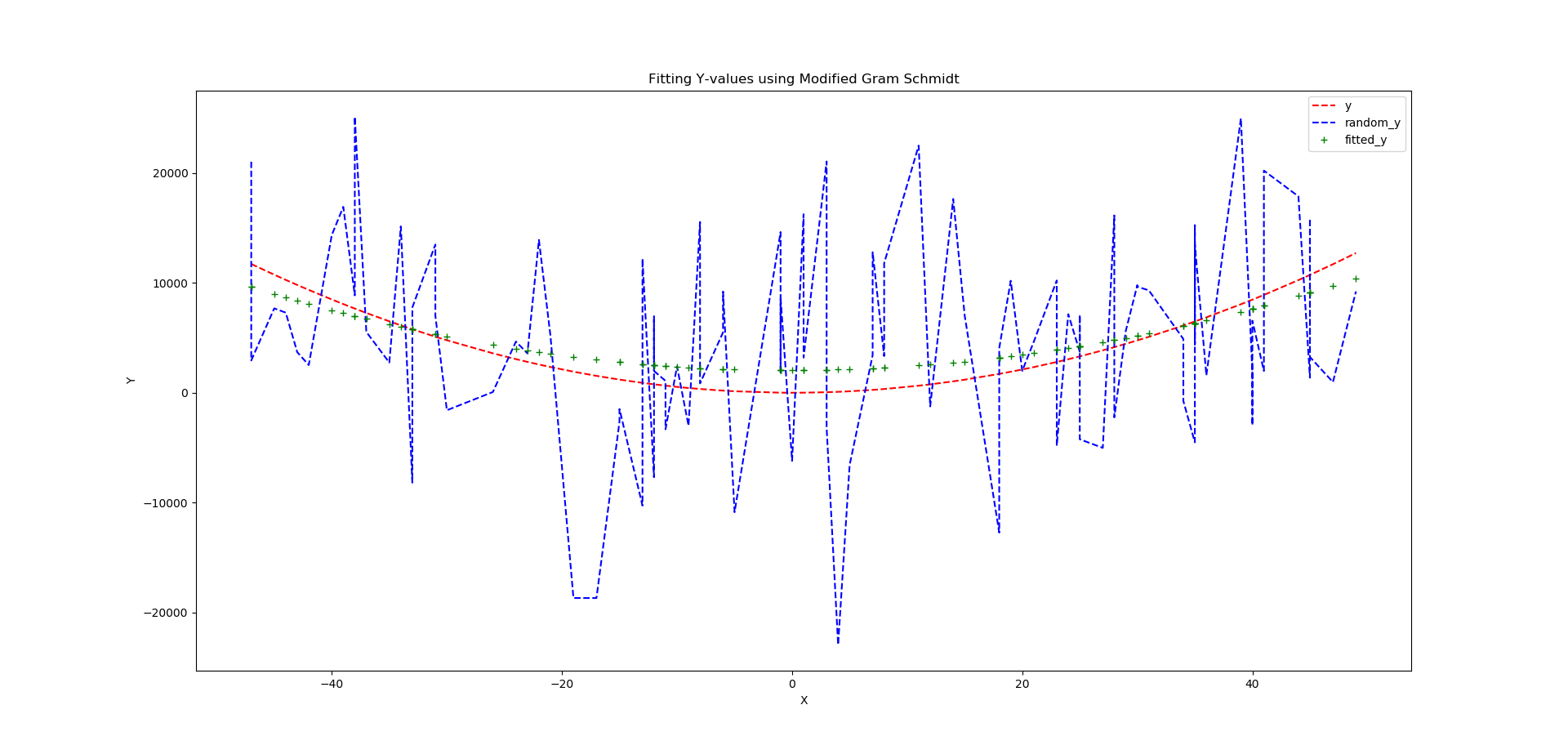
Casey Carr

11/21/18

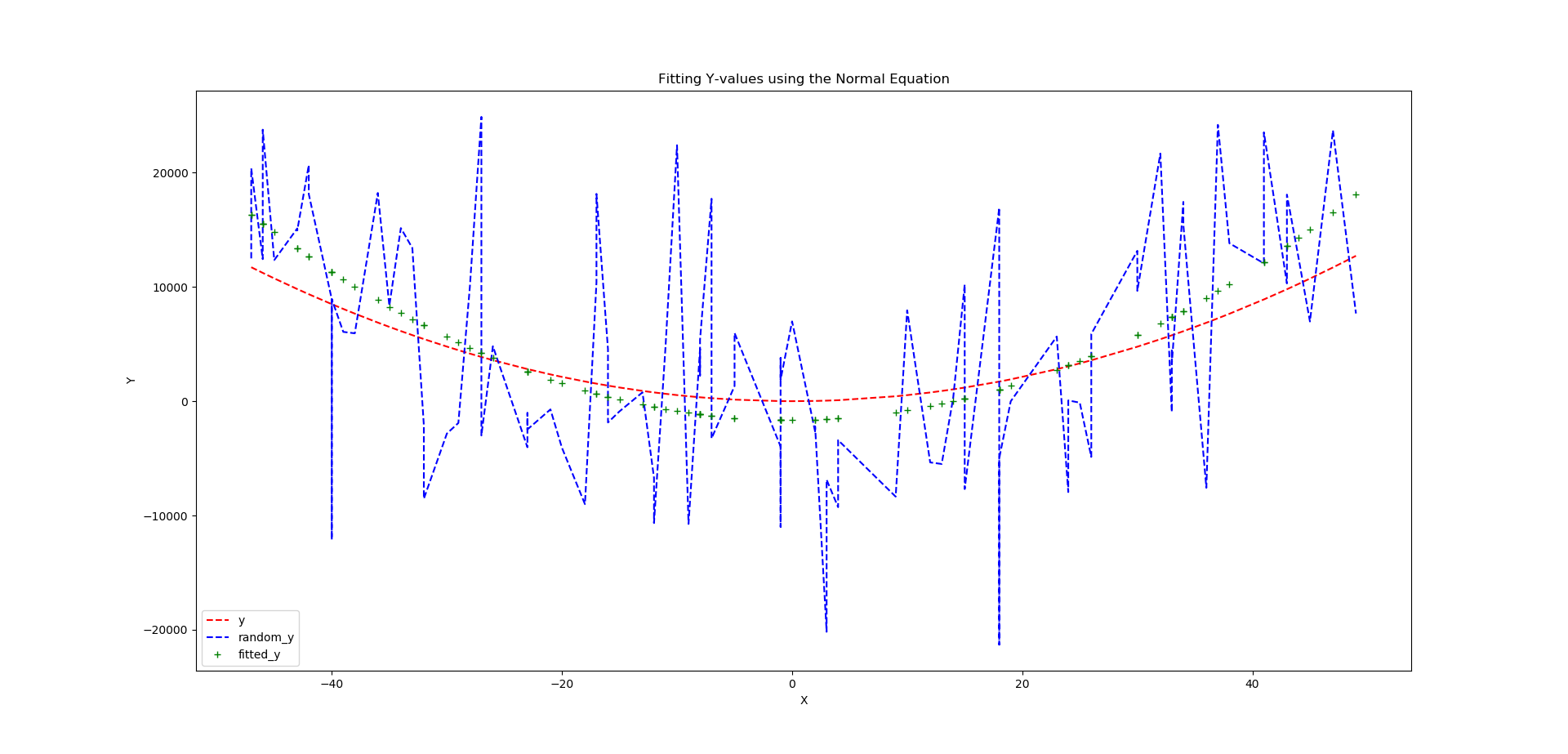
EECE 7220

**HW 4**: Fitting Y-values using various techniques with sigma = 10,000.

Q1:

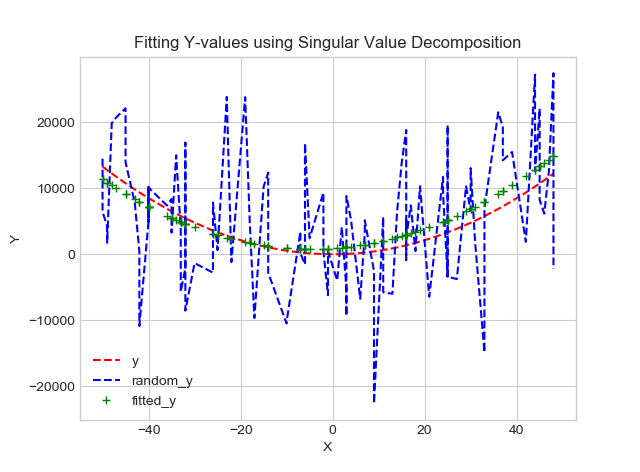
Betas: [[ 1075.73347559 ] [ -15.85284368 ] [ 4.72055966 ]]

As expected, the random Y-values (random\_y) are very volatile from one observation to the next. Therefore, we generate the fitted Y-values using the Modified Gram Schmidt algorithm which has mitigated the randomness inherent to each observation as to minimize the distance between each observation and the fitted curve.

Q2:

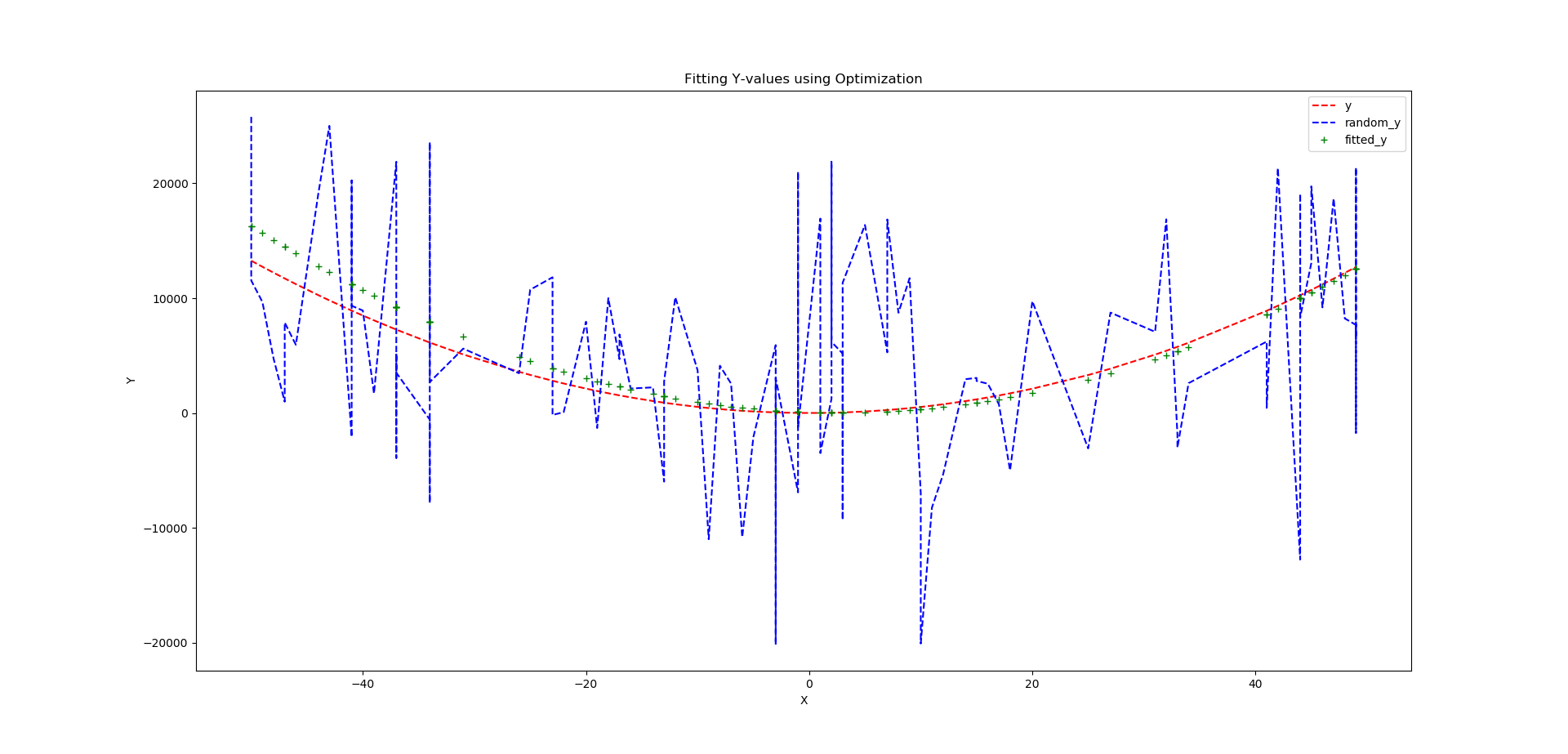
Betas: [[ 1362.29 ] [ -12.109 ] [ 4.318 ]]

As expected, the random Y-values (random\_y) are very volatile from one observation to the next. Therefore, we generate the fitted Y-values using the Normal Equation which has mitigated the randomness inherent to each observation as to minimize the distance between each observation and the fitted curve.

Q3:

Betas: [[ 1613.0448771 ] [ 39.77033235 ] [ 3.98253248 ]]

As expected, the random Y-values (random\_y) are very volatile from one observation to the next. Therefore, we generate the fitted Y-values using the Singular Value Decomposition algorithm which has mitigated the randomness inherent to each observation as to minimize the distance between each observation and the fitted curve.

Q4:

Betas: [[ 59.442 ] [ -31.889 ] [ 5.95 ]]

As expected, the random Y-values (random\_y) are very volatile from one observation to the next. Therefore, we generate the fitted Y-values using the built-in Optimization algorithm, scipy.optimize.fmin(), which has mitigated the randomness inherent to each observation as to minimize the distance between each observation and the fitted curve.