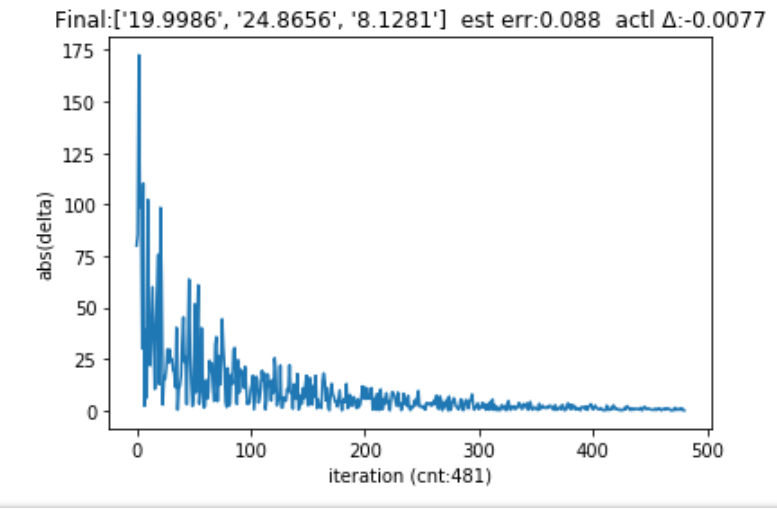
The algorithm used was linear regression, based off the iterative method of changing the estimated price off of the following method.

**estimatedUnitPrice = estimatedUnitPrice - (delta \* ALPHA \* randomMealPortions)**

**Round 1:**

Note that the starting prices all started at $10

estimatedUnitPrice = np.array([10, 10, 10])



**# PLACEHOLDER\_1#start:** set your own stopping conditions and learning rate

MAX\_ITERATION = 100000

MIN\_DELTA = 0.01

ALPHA = .001 # 1e-3

**# PLACEHOLDER\_1#end**

**# %%%PLACEHOLDER\_2#start**

delta = expectedTotalPrice – cashierPrice

deltaHistory.append(abs(delta))

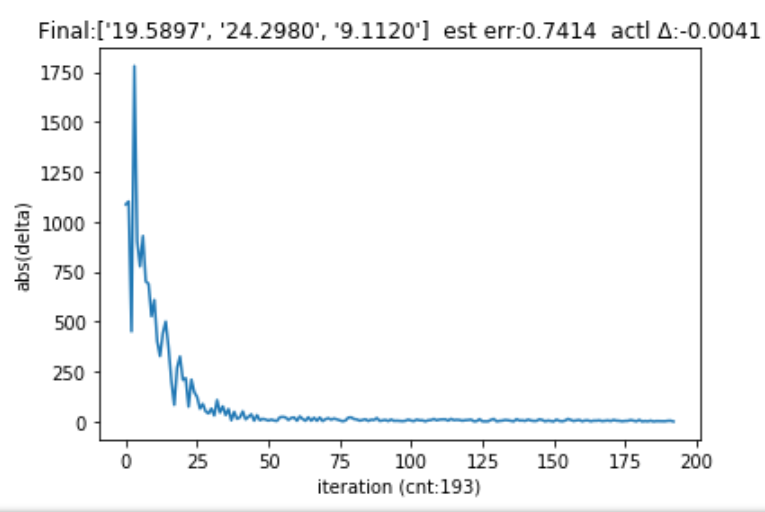
estimatedUnitPrice = estimatedUnitPrice - (delta \* ALPHA \* randomMealPortions)

**# %%%%PLACEHOLDER\_2#end**

**Round 2:**

Note that the starting prices all started at $100

estimatedUnitPrice = np.array([100, 100, 100])



**# PLACEHOLDER\_1#start:**

MAX\_ITERATION = 500, I changed this because during the first round only 481 iterations happen

MIN\_DELTA = 0.01

ALPHA = .001 # 1e-3

**# PLACEHOLDER\_1#end**

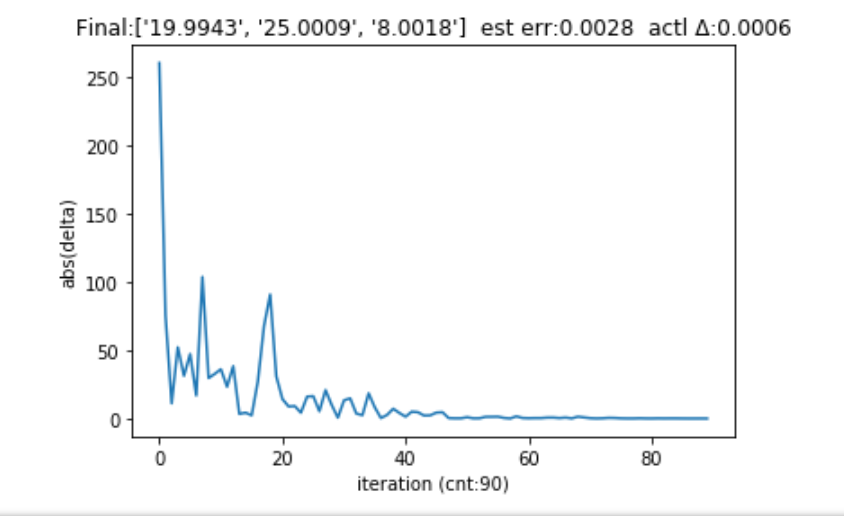
**The second placeholder content never changed**

**Round 3:**

Note that the starting prices all started at $1

estimatedUnitPrice = np.array([1, 1, 1])

I changed the alpha learning rate just to understand how delta would be effected and is now .01 while leaving the MAX\_ITERATION at 500



**The second placeholder content never changed**

Results:

I believe the first round had the most accurate estimated prices. The second round converged faster than any other round but didn’t achieve as accurate estimated unit prices.

I have attached the following python files that go with these results.