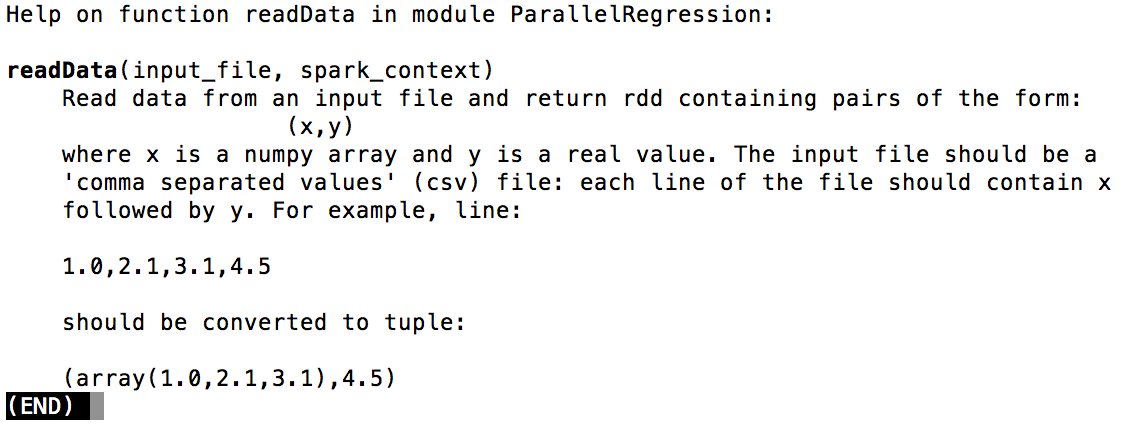
**HW 2**

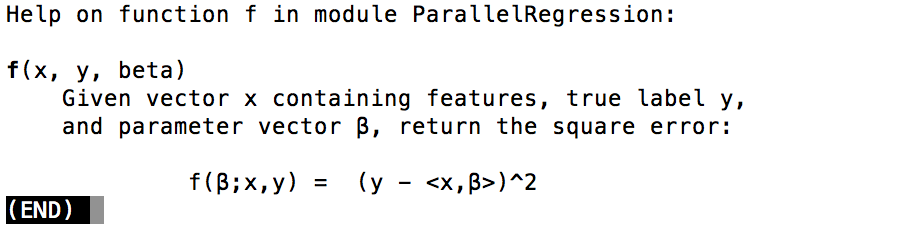
**Xianlong Zhang**

**Question 0:**

The help(PR.readData) will print as following:



The help(PR.f) will print as following:



The two parts are printed because they are in the annotation part under readData() and f() functions.

The Figure 1 is a part of small.test, which is set of numbers, and Figure 2 is an element of resulting RDD, the RDD is a tuple of those elements as format (x, y), x is an array.

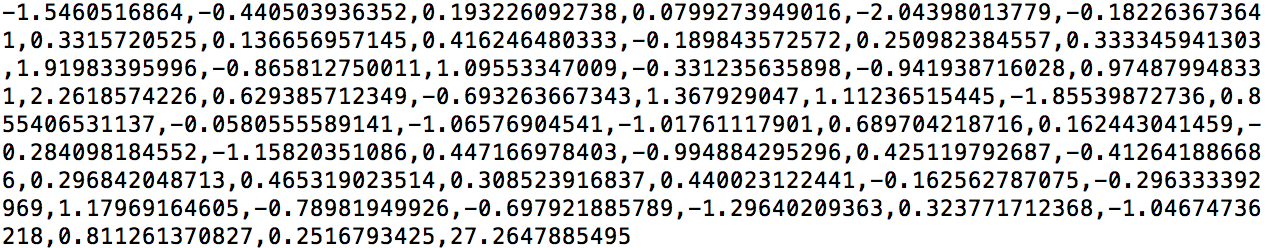
****

Figure 1

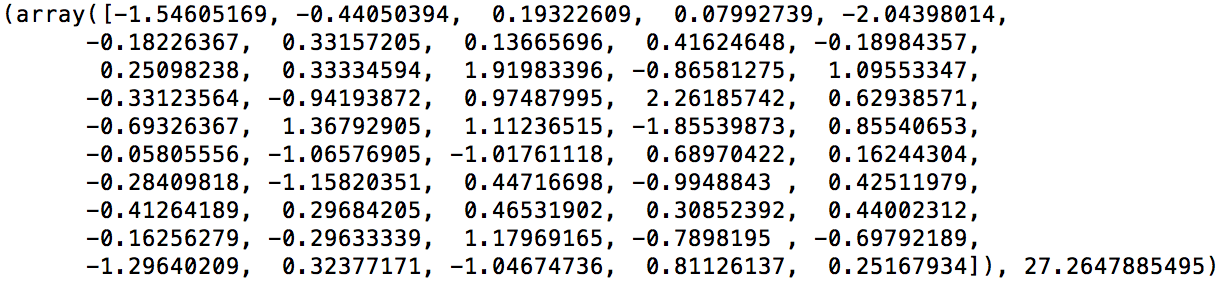
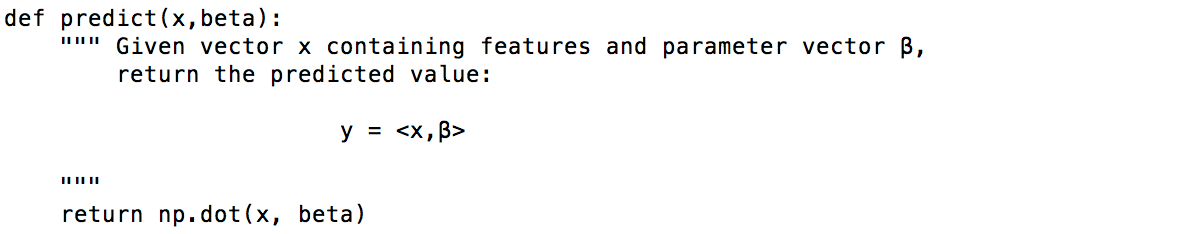


Figure 2

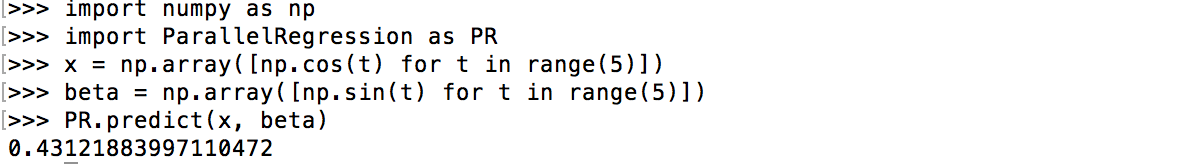
**Question 1:**

**(a)**

The modified code is as following:

****

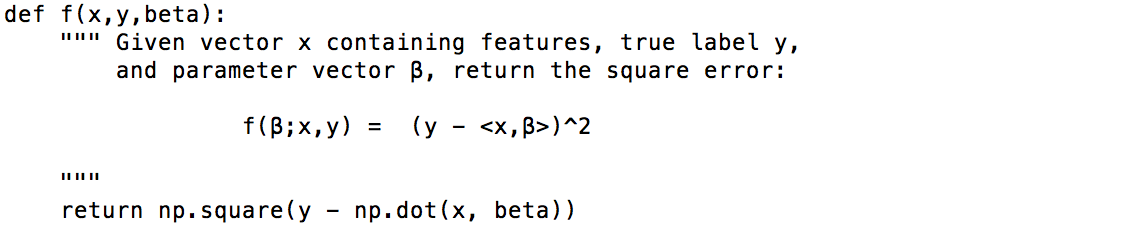
**(b)**

****

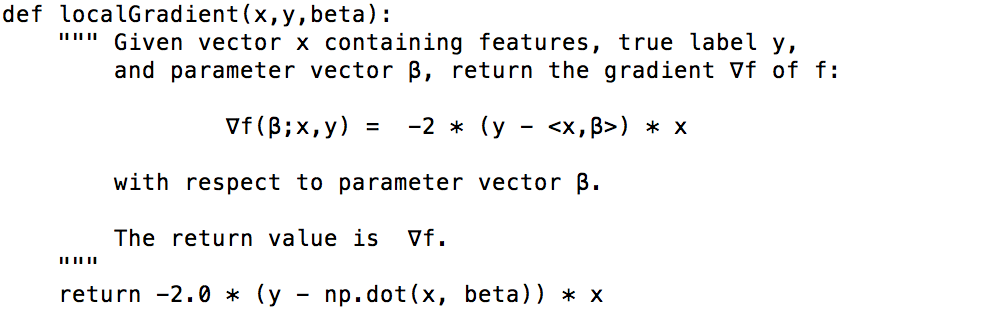
**Question 2:**

**(a)**

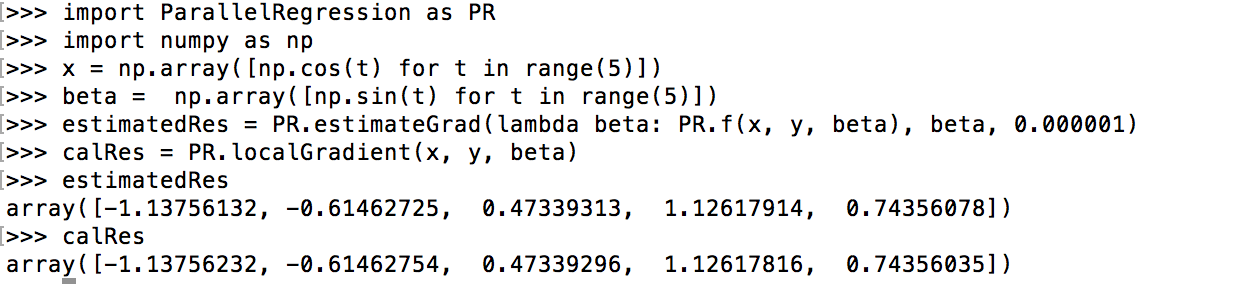
**(b)**



**(c)**

****

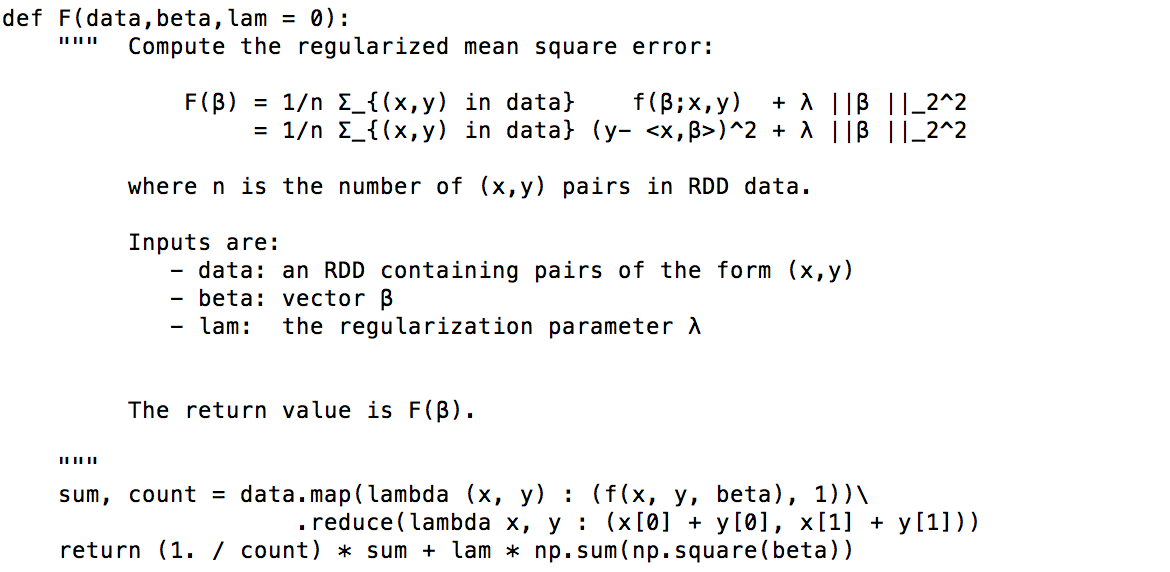
**(d)**



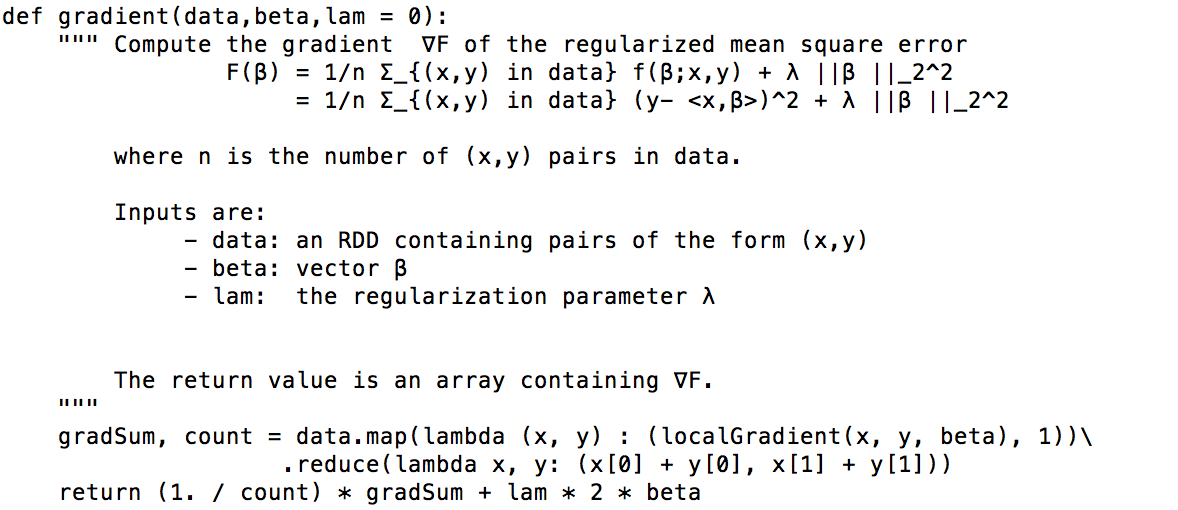
We set small enough, and from results above, we find they are nearly identical, so it’s correct since localGradient agrees with the estimate produced by estimateGrad.

**Question 3:**

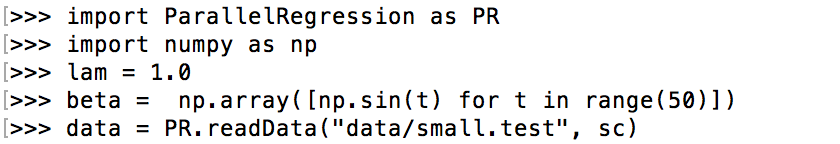
**(a)**

****

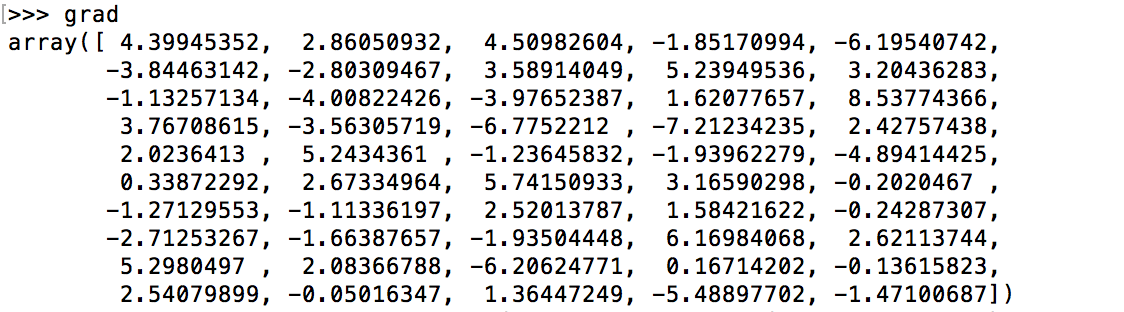
**(b)**

****

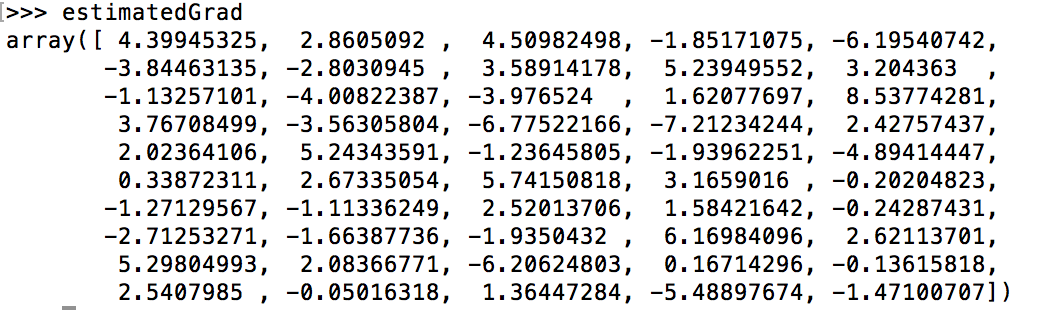
**(c)**

****

**../../../../Screen%20Shot%202017-10-27%20at%2001.31.00.png**

****

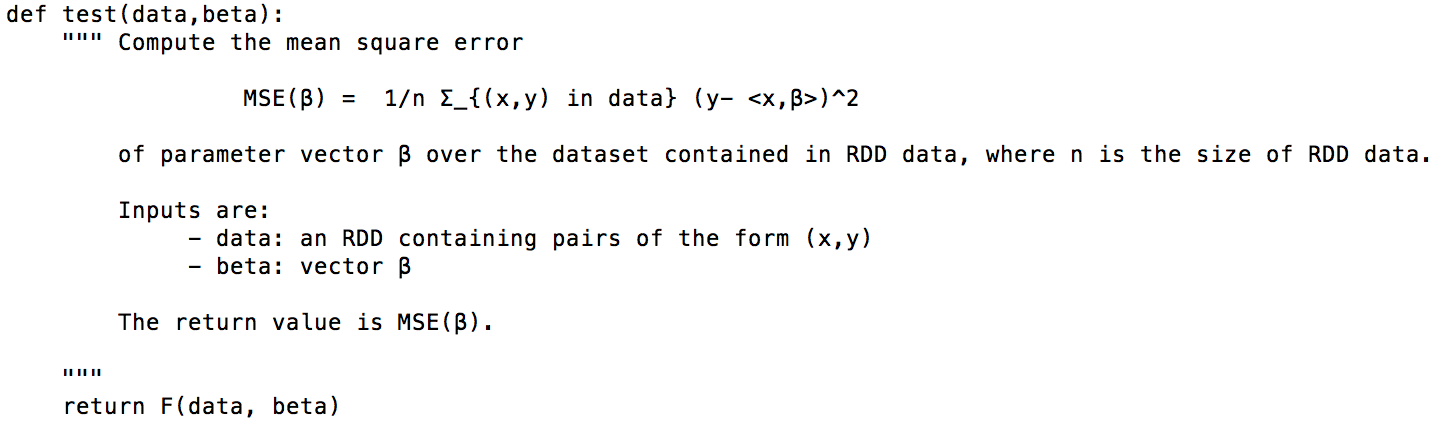
**../../../../Screen%20Shot%202017-10-27%20at%2002.42.34.png**

****

**Question 4:**

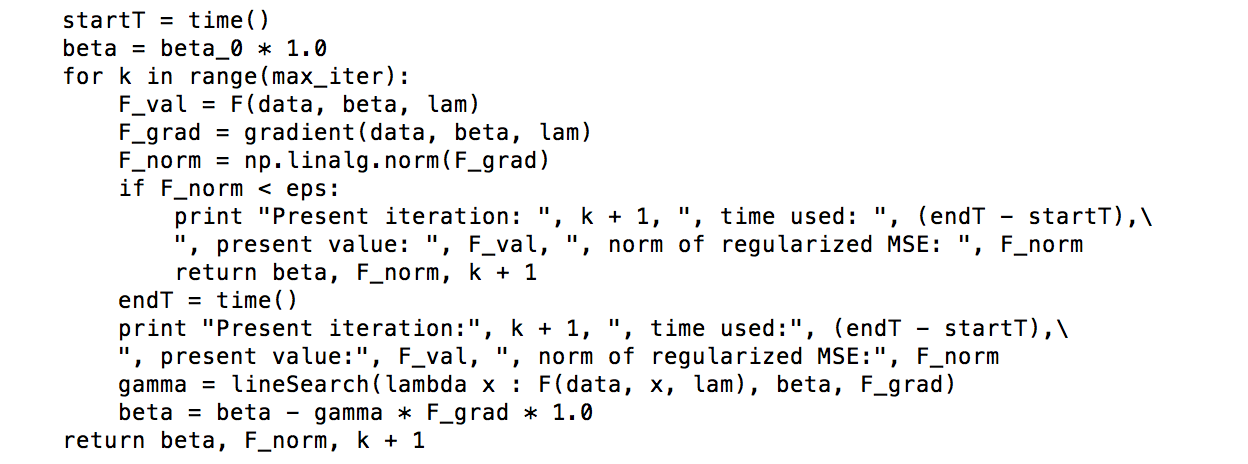
**(a)**

The code I modified is as following:

****

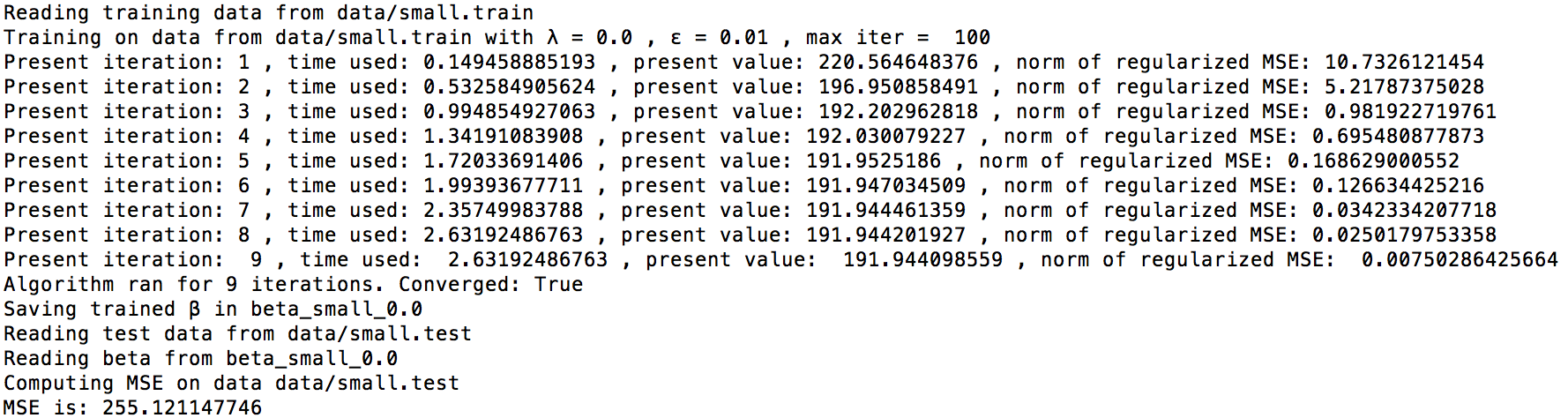
**(b)**

The train() function I modified is as following:

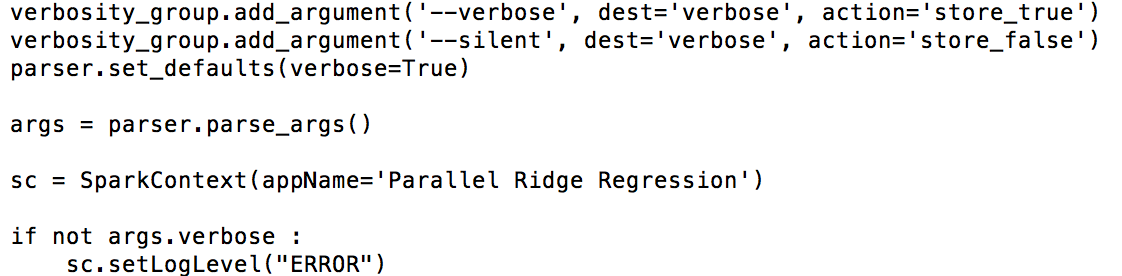
****

**(c)**

The following is the result we get from execution:

****

The part in the following screenshot has the effect of “--silent”. We can see that there are two modes –silent and –verbose. If we choose –silent the sc.setLogLevel will be ERROR, and some trival information won’t appear.



**Question 5:**

The table of the resulting MSE values are as following, and from the table we can find that when, the smallest MSE can be got, which is 3967.17658354.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 4151.49371162 | 4000.09930202 | 3977.8384858 | 3971.3275161 | 3968.86820762 | 3967.83265427 | 3967.38615943 |
|  |  |  |  |  |  |  |
| 3967.21396808 | 3967.17658354 | 3967.2071355 | 3967.27230693 | 3967.35461275 | 3967.44420838 | 3967.53515038 |
|  |  |  |  |  |  |  |
| 3967.6248401 | 3967.7117152 | 3967.79442828 | 3967.87275232 | 3967.94720755 | 3968.0177581 | 3968.08323632 |

**Question 6:**

The average running time for different partitions are as following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N=1** | **N=2** | **N=4** | **N=8** | **N=16** | **N=32** | **N=64** |
| 61.845 | 38.330 | 24.210 | 16.133 | 13.464 | 15.399 | 18.821 |

From table above, when N = 16, the smallest mean convergence time can be got, which is 13.464, so we can conclude that adding more partitions not always decrease the convergence time for the reason that every partition will be assigned a task, if there are too many tasks, the data size is too small in each partition, or sometimes workload is not well balanced, which would cause the cost of switching thread and make some tasks wait. It’s a very inefficient use of resources.