Question 1 (a)

List the indices of the removed data points.

```
Int64Index([21, 43, 111, 146, 232, 303], dtype='int64')
```

then Remove all rows of the data that contain a missing ('NA') value

	transactiondate	age	nearestMRT	nConvenience	latitude	longitude	price
0	2012.917	32.0	84.87882	10.0	24.98298	121.54024	37.9
1	2012.917	19.5	306.59470	9.0	24.98034	121.53951	42.2
2	2013.583	13.3	561.98450	5.0	24.98746	121.54391	47.3
3	2013.500	13.3	561.98450	5.0	24.98746	121.54391	54.8
4	2012.833	5.0	390.56840	5.0	24.97937	121.54245	43.1
...	...	...	...	...	...	...	...
403	2013.000	13.7	4082.01500	0.0	24.94155	121.50381	15.4
404	2012.667	5.6	90.45606	9.0	24.97433	121.54310	50.0
405	2013.250	18.8	390.96960	7.0	24.97923	121.53986	40.6
406	2013.000	8.1	104.81010	5.0	24.96674	121.54067	52.5
407	2013.500	6.5	90.45606	9.0	24.97433	121.54310	63.9

408 rows × 7 columns

delete all features from the dataset apart from: age, nearestMRT and nConvenience.

	age	nearestMRT	nConvenience
0	32.0	84.87882	10.0
1	19.5	306.59470	9.0
2	13.3	561.98450	5.0
3	13.3	561.98450	5.0
4	5.0	390.56840	5.0
...	...	...	...
403	13.7	4082.01500	0.0
404	5.6	90.45606	9.0
405	18.8	390.96960	7.0
406	8.1	104.81010	5.0
407	6.5	90.45606	9.0

408 rows × 3 columns

Normalization

```
age             0.406079
nearestMRT      0.162643
nConvenience    0.412010
dtype: float64
```

Question 2

Print out the first and last rows of training set

	age	nearestMRT	nConvenience
0	32.0	84.87882	10.0
203	38.5	665.06360	3.0

After normalization and add price is

	age	nearestMRT	nConvenience	price
0	0.730594	0.009513	1.0	37.9
203	0.878995	0.099260	0.3	34.2

Print out the first and last rows of test set

	age	nearestMRT	nConvenience
204	11.5	1360.13900	1.0
407	6.5	90.45606	9.0

After normalization and add price is

	age	nearestMRT	nConvenience	price
204	0.262557	0.206780	0.1	26.2
407	0.148402	0.010375	0.9	63.9

Question 3

In this question all the $w^t x^i$= w0+ w1x1+w2x2+w3x3

$$Lc(y^i, \hat{y}^i) = \sqrt{\frac{1}{c^2} * (y^i - w^t x^i)^2 + 1} - 1$$

$$\frac{dLc(y^i, \hat{y}^i)}{dw_k}$$

$$= \frac{1}{2} * \left(\frac{1}{c^2} * \left(y^i - w^t x^i\right)^2 + 1\right)^{-\frac{1}{2}} * \frac{2}{c^2} * (-x^k) * (y^i - w^t x^i)$$

$$= \frac{x^k * (w^t x^i - y\hat{}i)}{c^2 * \sqrt{\frac{1}{c^2}(w^t x^i - y^i)^2 + 1}}$$

$$= \frac{x_k * (w^t x^i - y\hat{}i)}{c * \sqrt{(w^t x^i - y^i)^2 + c\hat{}2}}$$

so

$\frac{dLc(y^i, \hat{y}^i)}{dw_0}$ will be $\frac{(w^t x^i - y\hat{}i)}{c * \sqrt{\left(w^t x^i - y^i\right)^2 + c\hat{}2}}$

$\frac{dLc(y^i,\hat{y}^i)}{dw_1}$ will be $\frac{x_1*(w^t x^i - y\hat{}i)}{c*\sqrt{(w^t x^i - y^i)^2 + c\hat{}2}}$ and so on(others just change x1 to x2 and x3)

Question 4

GD

i = 0

#iteration number

While not convergence:

#check if convergence

Wx_change= $sum(\frac{(w^t x^k - y\hat{}i)}{c*\sqrt{(w^t x^k - y^k)^2 + c\hat{}2}}$)for k from 0 to index

Wx = Wx – lr*Wx_change/index*xi

i+=1

#add one iteration time

SGD:

While not convergence:

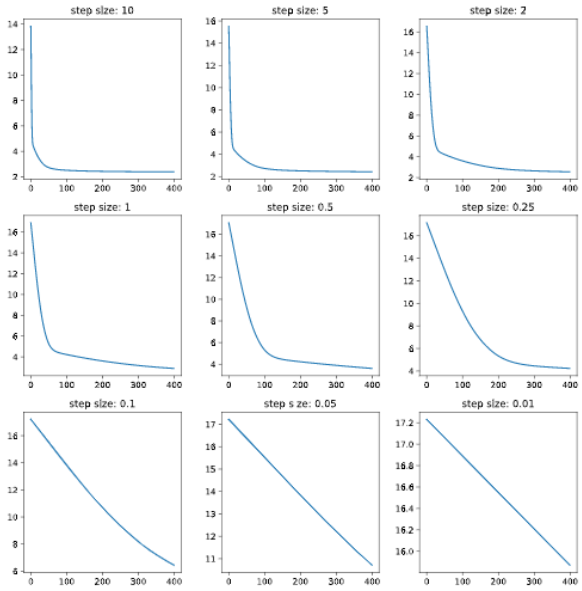
Theta = theta +learning_rate*($\frac{x_k*(w^t x^i - y\hat{}i)}{c*\sqrt{(w^t x^i - y^i)^2 + c\hat{}2}}$)

i+=1

#add one iteration time

Question 5

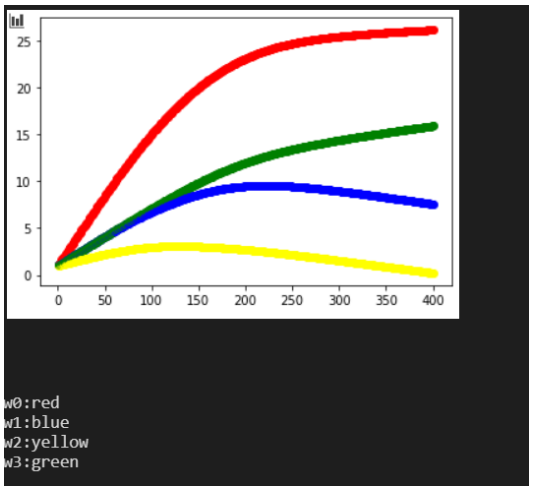
(a)



(b)

for step size I chose 0.5， cause all the lines are smooth so we pick the ones that have the most obvious change but aren't sudden,I think is 0.5s

(c)



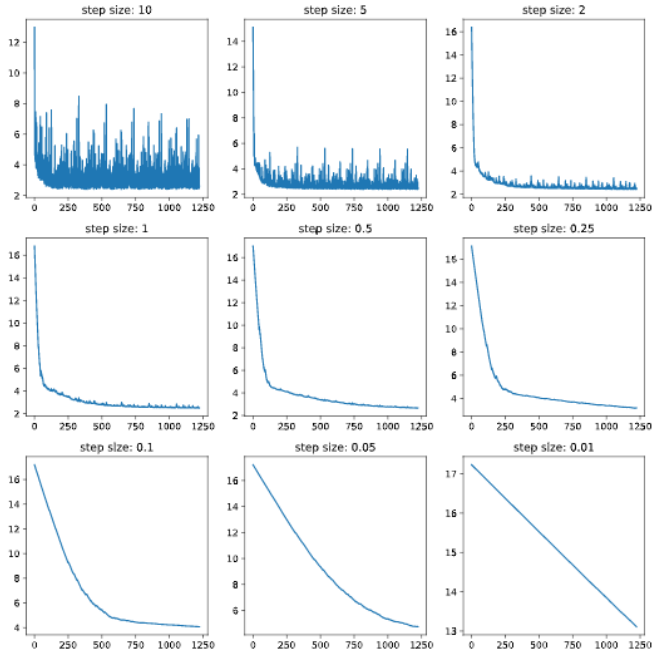
and print out the final weight vector , train loss and test loss

```
[26.11838075862762, 7.554954286320743, 0.22091266810394947, 15.882161070414401]
```

```
Train loss: 4.089850739201337
Test loss: 3.827500929218835
```

Question 6

(a)



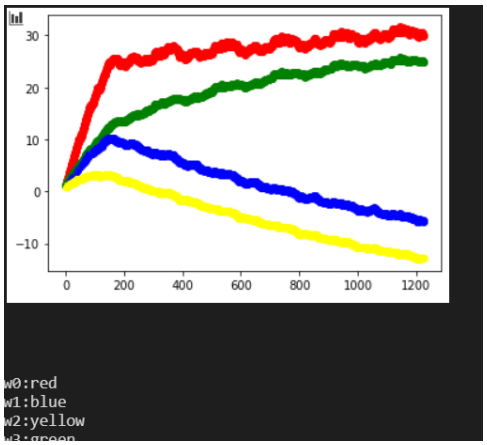
(b)

for step size I chose 0.1,

firstly since we allways use big step size to check vast amounts of data to Save the number of code operations and small step size for small amounts of data to ensure the

accuracy, If step size too small like 0.01 is hard to show the change and also waste too much time but too big step size like 10 is unstable so it has poor accuracy,so i chose '0.1' which is smooth and not cost a lot number of runs

(c)



and print out the final weight vector , train loss and test loss

```
[30.132063157428234, -5.583681450744382, -12.859753310061139, 24.875357185884447]
```

```
Train loss: 2.7802386966007977
Test loss: 2.8448220512160356
```

Question 7

step size for GD and SGD is use to precise the graph,if too big the graph will be unstable,if too small then the change will tends to be constant,so find a suitable step size is important.

After we done question 5 and 6,that we can find the line in question 5 is much smooth than question 6

First we have to think about what kinds of data are GD and SGD oriented,

Gd check all of the data in every iteration to get the mean value of weight that will take long time but the data will be more accurate.(so we always use it to check the dataset that not to big)

SGD choose one random data to calculate the lose that save time but not accurate.(so we always use it to check huge amount dataset)

They are like two extremes, so for the same data set there will always have the better one in GD and SGD， for this question the largest dataset that we have is training data and test data which is 208,it's not a big number of data set,so obviously GD is

better,thats the reason why it looks much smooth than SGD