CS584 Assignment 1: Report

Jingyu Zhu A20311779

1. Problem statement

In this assignment, I tried to do various kinds of solutions for both single feature and multiple feature data sets. In respect of the single feature data sets, I managed to apply linear model and several polynomial models for fitting the data, with which I got the training error and testing error and then compared them with the errors given by the built-in lm() operation. Also, I have reduced the amount of training data and recorded the related errors. I kept recording the errors all the time and tried to figure out how the errors would change among different cases. With regards to the multiple feature data sets, I performed both the linear regression and high dimensional regression and compared their errors and performance. After that, I came up with an iterative solution to find out what's the difference between the errors given by the explicit solution and those from the iterative one.

2. Proposed solution

In terms of the explicit solution, first I built the Z matrix based on all the given features. Then I plugged it into the following formula,

$$\theta = (Z^T \bullet Z)^{-1} \bullet Z^T \bullet Y$$

That gave me all the parameters I needed.

Then, I was able to get the Min Square Error (MSE) and the Relative Square Error (RSE) by following:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

$$RSE = \frac{1}{m}\sum_{i=1}^{m}\frac{(\hat{y}^{(i)} - y^{(i)})^2}{y^{(i)^2}}$$

To come up with the iterative solution, I applied the gradient descent algorithm. The idea of the algorithm is: first, start with a guess "theta0". Second, we iteratively update the guess "theta(i)" using data. The following formula properly summarize the foregoing process:

$$\theta^{(i)} \leftarrow \theta^{(i-1)} - \eta \nabla J(\theta^{(i-1)})$$

3. Implementation details

(1) Single variable regression

First, I opened the data file and got the data I needed. Second, I cut out all the feature columns and combined it with a new column <1, 1, … , 1> on the left side, by which I get the "Z" matrix. Also, I pull out the last column from the data to form the Y vector. Then I was able to calculate the theta vector by the formula mentioned above and figure out what the MSE and RSE are. After that I applied the built-in lm() function to come up with the "theta" once again and redid the process to calculate MSE and RSE one more time. Finally, I recorded these data in the "finalMatrix" and compared them.

To do the polynomial regression for the single variable data set, basically I just repeated the process I went through for the linear regression model. The only difference lied in constructing the Z matrix. Besides the all-one column, I also added a new quadratic column (for the cubic model, both the quadratic and cubic column) into the matrix. The other part before getting the final result was just the same as that in the linear regression model.

In the last step I'm with the single variable data sets, I cut all the four original data sets by half and reran my algorithm again to observe how the errors would change.

(2) Multivariate regression

First, I got the data from the files and mapped them to high dimensions. Then I constructed the high-dimension Z matrix. Second, I came up with the theta vector by the formula (which is mentioned in part2). Finally, I calculated the MSE and RSE, and recorded them.

To implement the iterative solution, I applied the Gradient Descent Algorithm. For the first two sets, I set the learning rate be 0.05 and then have the program iterated 1000 times to converge. For the latter two data sets, since the size of them were relatively large (100000 observations!), I decided to increase the learning rate up to 0.1 and reduce the number of iterations down to 300. At last, I put the result calculated by the iterative solution in a matrix. It turned out to be the algorithm worked pretty well with the parameters I provided.

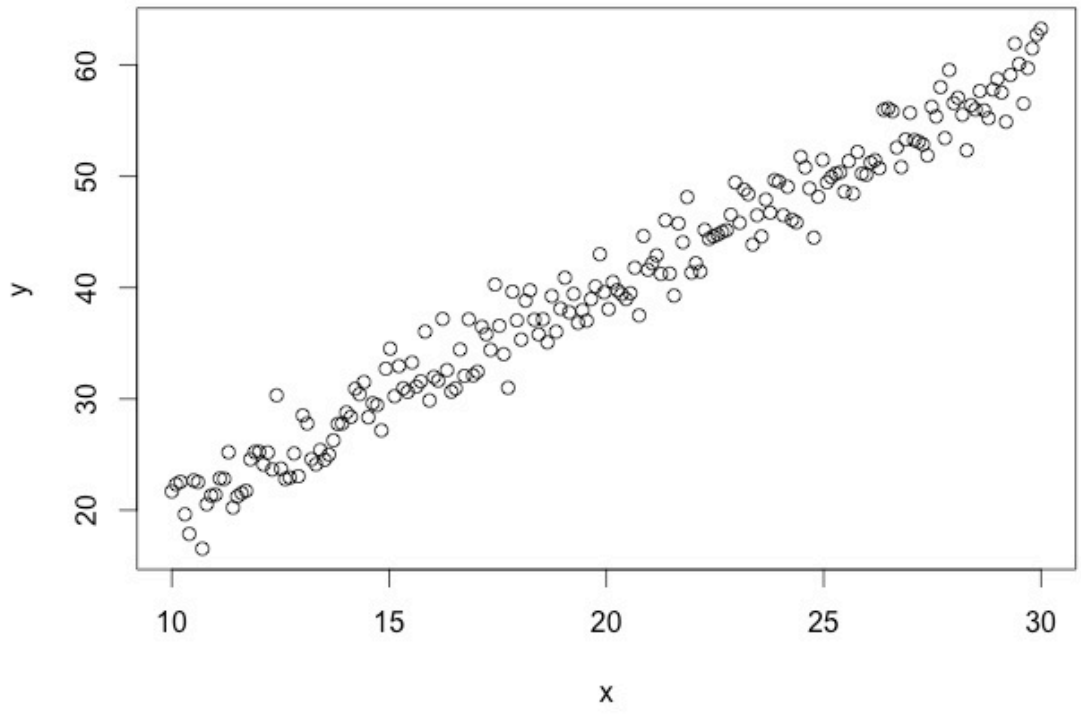
The last step with the multivariate data sets was to solve the dual linear regression problem via a Gaussian kernel function. Therefore, I constructed my Gaussian kernel function (which can measure the similarity between vectors), and plugged in the data to get the final result. The codes gave more implementation details.

4. Results and discussion

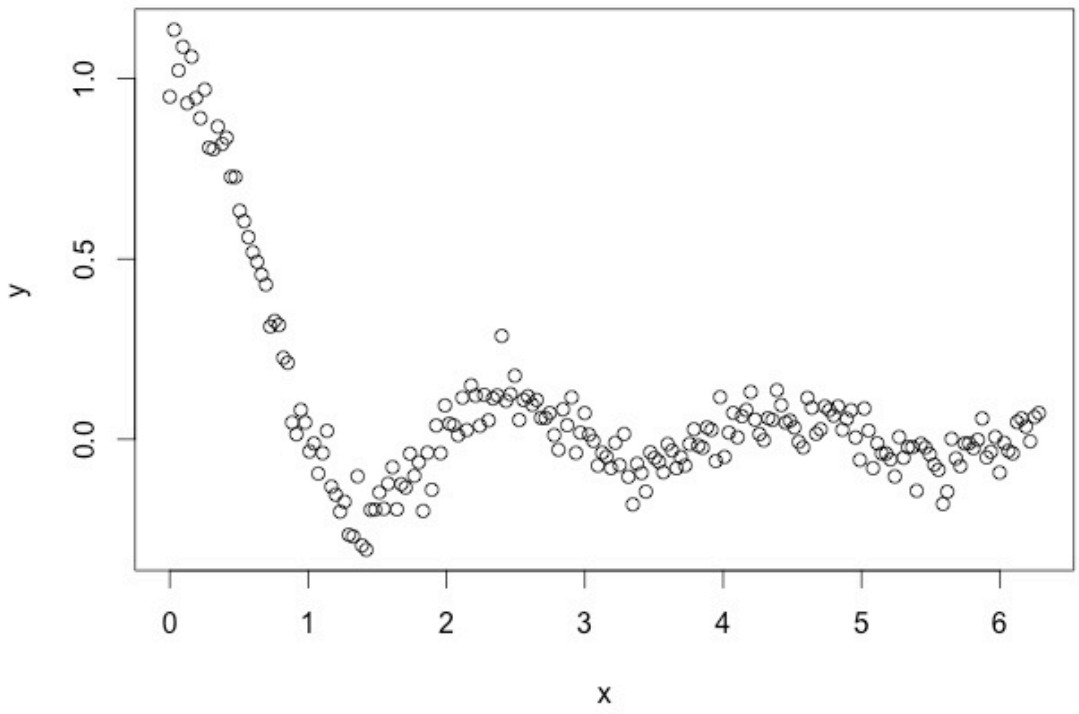
Single variable regression:

1.Plot the data.

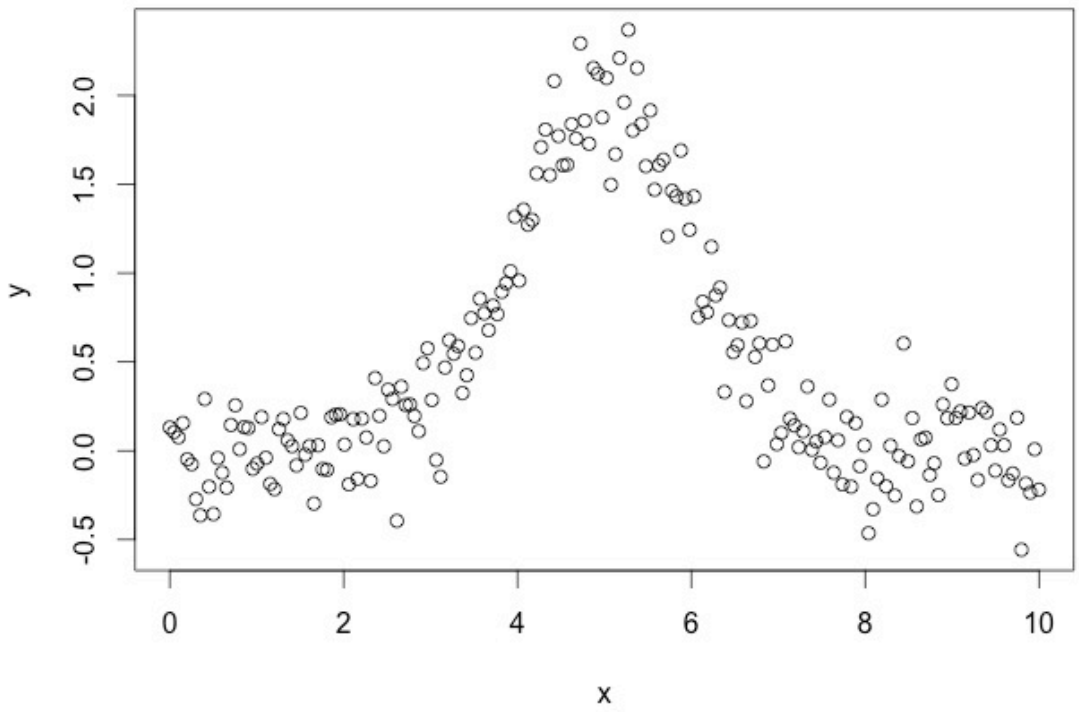
Data svar-set 1:



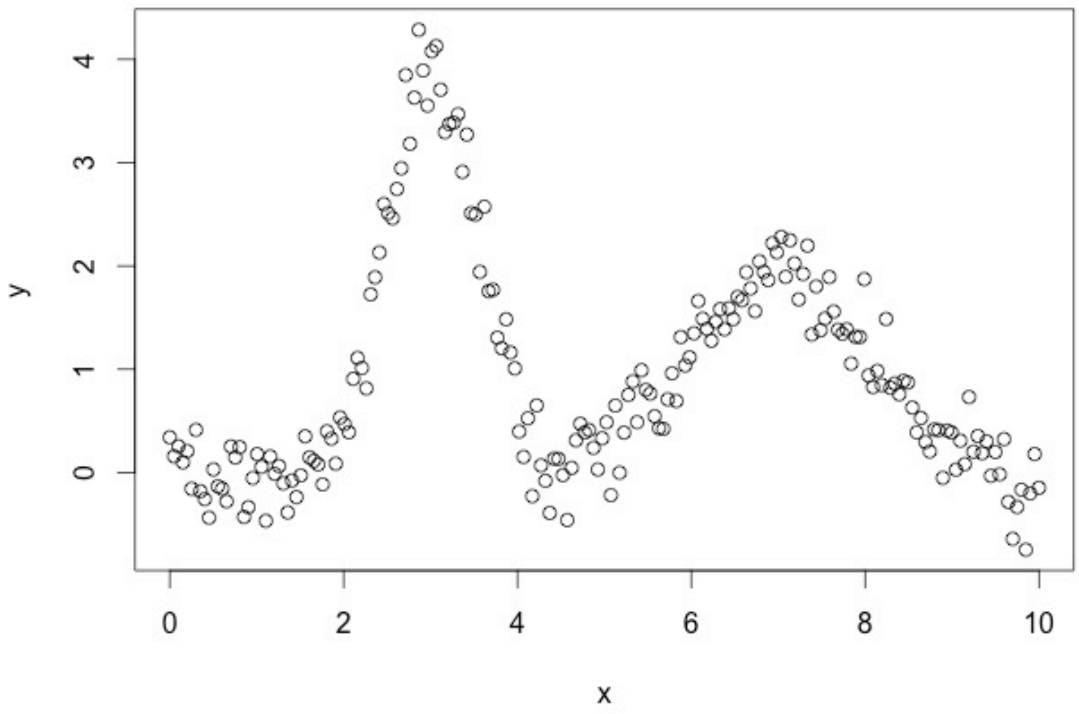
Data svar-set 2:



Data svar-set 3:



Data set 4:



As we can see from the graphs above, it turns out that data set 1 should be fitted into a linear model, whereas the polynomial model might be proper for the other three data sets.

2. Fit the linear model for these data sets.

Since the 10 fold cross validation is applied, all the graphs show a result of ten groups.

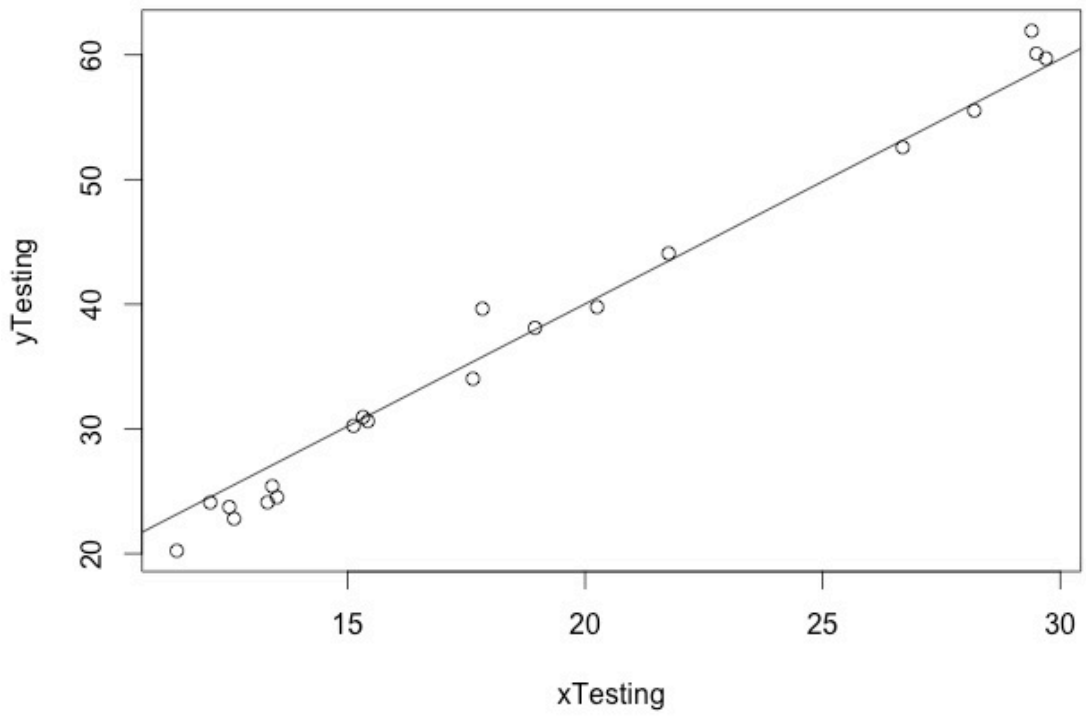
Attention: The four columns from the right are computed by the built-in lm() function.

Attention: After executing the programs, the calculated matrix of various errors (i.e. the training MSE RSE and the testing MSE RSE) are stored in the variable "finalMatrix".

Data svar1:

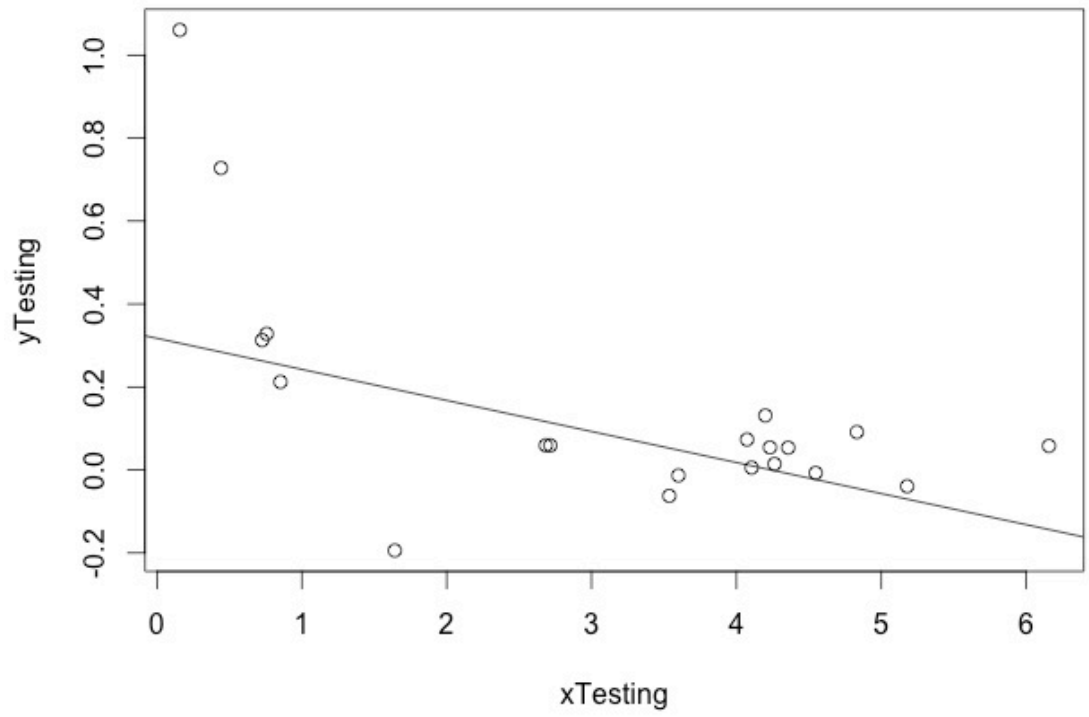
10 observations of 8 variables

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	4.335186	0.003744185	3.494482	0.004317968	4.335186	0.003744185	3.494482	0.004317968
2	4.277045	0.003348520	3.887770	0.007344515	4.277045	0.003348520	3.887770	0.007344515
3	4.091733	0.003650942	5.549540	0.003774046	4.091733	0.003650942	5.549540	0.003774046
4	4.180044	0.003789476	4.725401	0.002641218	4.180044	0.003789476	4.725401	0.002641218
5	4.267194	0.003720976	3.953828	0.003225347	4.267194	0.003720976	3.953828	0.003225347
6	4.288279	0.003810050	3.769533	0.002303222	4.288279	0.003810050	3.769533	0.002303222
7	4.120873	0.003501331	5.420900	0.004700293	4.120873	0.003501331	5.420900	0.004700293
8	4.323401	0.003870991	3.498455	0.002290633	4.323401	0.003870991	3.498455	0.002290633
9	4.137732	0.003558992	5.135499	0.004446763	4.137732	0.003558992	5.135499	0.004446763
10	4.233439	0.003739340	4.252117	0.003410972	4.233439	0.003739340	4.252117	0.003410972



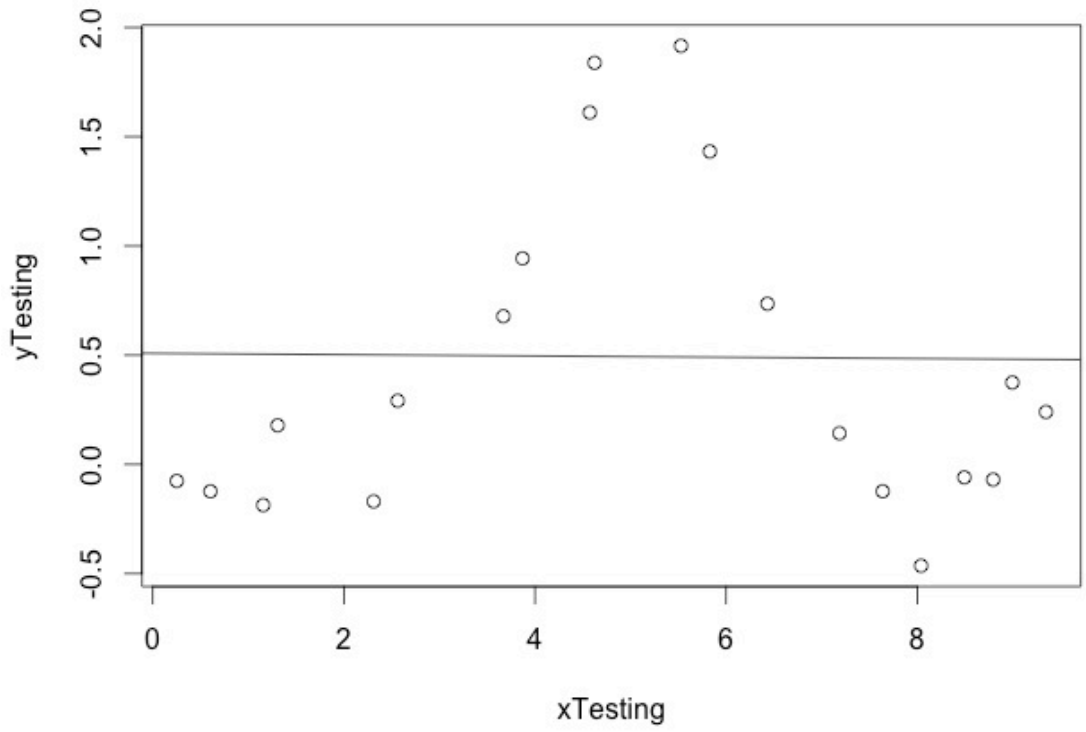
Data svar2:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.06050155	109.25281	0.05188392	2.873233	0.06050155	109.25281	0.05188392	2.873233
2	0.06011406	113.55133	0.05562345	17.777016	0.06011406	113.55133	0.05562345	17.777016
3	0.06135907	122.32438	0.04391637	10.472282	0.06135907	122.32438	0.04391637	10.472282
4	0.05998325	112.43108	0.05591346	6.923642	0.05998325	112.43108	0.05591346	6.923642
5	0.05647698	95.72254	0.09011061	10.281299	0.05647698	95.72254	0.09011061	10.281299
6	0.06047064	99.27515	0.05167362	38.343779	0.06047064	99.27515	0.05167362	38.343779
7	0.05926223	75.21299	0.06461504	376.809704	0.05926223	75.21299	0.06461504	376.809704
8	0.05754051	57.38414	0.07795091	571.464205	0.05754051	57.38414	0.07795091	571.464205
9	0.05720344	99.06150	0.08206406	5.225814	0.05720344	99.06150	0.08206406	5.225814
10	0.06196014	119.22060	0.03825547	37.893190	0.06196014	119.22060	0.03825547	37.893190



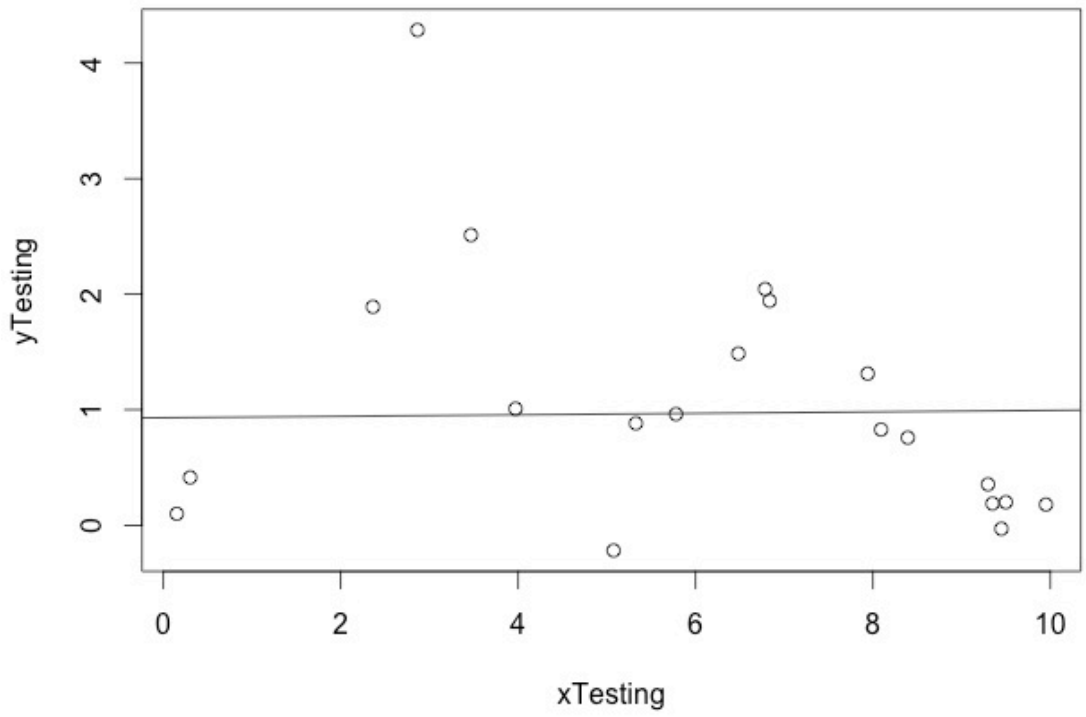
Data svar3:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.4978847	108.09488	0.5063583	15.09066	0.4978847	108.09488	0.5063583	15.09066
2	0.5101433	107.71608	0.3968387	41.95110	0.5101433	107.71608	0.3968387	41.95110
3	0.4809184	112.98245	0.6612739	16.86955	0.4809184	112.98245	0.6612739	16.86955
4	0.5153103	55.18142	0.3506255	524.82345	0.5153103	55.18142	0.3506255	524.82345
5	0.5102343	99.91972	0.3983225	20.26865	0.5102343	99.91972	0.3983225	20.26865
6	0.5071622	104.59489	0.4250998	69.37147	0.5071622	104.59489	0.4250998	69.37147
7	0.4906359	85.07500	0.5719139	200.42023	0.4906359	85.07500	0.5719139	200.42023
8	0.4982870	102.25508	0.5062072	22.34489	0.4982870	102.25508	0.5062072	22.34489
9	0.4552930	86.31250	0.9007749	62.10720	0.4552930	86.31250	0.9007749	62.10720
10	0.5183025	113.04419	0.3259777	27.29271	0.5183025	113.04419	0.3259777	27.29271



Data svar4:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	1.209249	1864.7001	1.1373348	63.02765	1.209249	1864.7001	1.1373348	63.02765
2	1.108694	1945.0266	2.0361988	94.07296	1.108694	1945.0266	2.0361988	94.07296
3	1.264240	1938.1454	0.6417086	25.25932	1.264240	1938.1454	0.6417086	25.25932
4	1.216254	1878.7561	1.0637961	381.57959	1.216254	1878.7561	1.0637961	381.57959
5	1.245234	1926.7927	0.8033368	263.39939	1.245234	1926.7927	0.8033368	263.39939
6	1.200808	1965.6000	1.2050841	80.93370	1.200808	1965.6000	1.2050841	80.93370
7	1.211565	1812.3515	1.1122321	36.79575	1.211565	1812.3515	1.1122321	36.79575
8	1.199433	1921.3058	1.2150857	21.48668	1.199433	1921.3058	1.2150857	21.48668
9	1.203398	1833.3157	1.1836388	75.49866	1.203398	1833.3157	1.1836388	75.49866
10	1.143150	109.2519	1.7244547	15402.82172	1.143150	109.2519	1.7244547	15402.82172



To conclude from above, MSE is not a good criterion to judge our regression model's performance sometimes (as is shown from "svar2"). By contrast, typically RSE is a more powerful measurement and can give us better intuition about how far the data is fitted into the model.

3. Fit different polynomial models for these data sets.

Data svar1 quadratic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	4.335057	0.003738617	3.492569	0.004297838	4.335057	0.003738617	3.492569	0.004297838
2	4.277024	0.003347397	3.886196	0.007329826	4.277024	0.003347397	3.886196	0.007329826
3	4.079226	0.003599873	5.734364	0.003920211	4.079226	0.003599873	5.734364	0.003920211
4	4.178315	0.003807187	4.775689	0.002674118	4.178315	0.003807187	4.775689	0.002674118
5	4.264347	0.003688803	3.982094	0.003274573	4.264347	0.003688803	3.982094	0.003274573
6	4.285221	0.003785323	3.806018	0.002309800	4.285221	0.003785323	3.806018	0.002309800
7	4.109874	0.003434465	5.603980	0.005059486	4.109874	0.003434465	5.603980	0.005059486
8	4.323304	0.003875619	3.506194	0.002296473	4.323304	0.003875619	3.506194	0.002296473
9	4.132343	0.003587270	5.271057	0.004569263	4.132343	0.003587270	5.271057	0.004569263
10	4.233111	0.003732409	4.249345	0.003379520	4.233111	0.003732409	4.249345	0.003379520

Data svar2 quadratic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.03932099	45.61272	0.03355561	38.800889	0.03932099	45.61272	0.03355561	38.800889
2	0.03815394	61.41816	0.04423368	11.379989	0.03815394	61.41816	0.04423368	11.379989
3	0.03899411	55.57426	0.03597805	7.241984	0.03899411	55.57426	0.03597805	7.241984
4	0.03898935	49.98374	0.03601029	10.044591	0.03898935	49.98374	0.03601029	10.044591
5	0.03845144	45.56514	0.04274088	8.293182	0.03845144	45.56514	0.04274088	8.293182
6	0.03822802	48.80753	0.04321095	54.081089	0.03822802	48.80753	0.04321095	54.081089
7	0.03797684	36.75465	0.04632114	113.316902	0.03797684	36.75465	0.04632114	113.316902
8	0.03646637	40.31816	0.05897048	153.560930	0.03646637	40.31816	0.05897048	153.560930
9	0.03829878	45.35752	0.04330380	5.939707	0.03829878	45.35752	0.04330380	5.939707
10	0.04116200	51.85303	0.01675475	89.112598	0.04116200	51.85303	0.01675475	89.112598

Data svar3 quadratic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.2533747	127.50446	0.2527063	6.727343	0.2533747	127.50446	0.2527063	6.727343
2	0.2645076	128.57348	0.1527399	25.062886	0.2645076	128.57348	0.1527399	25.062886
3	0.2463464	123.03607	0.3184495	24.537178	0.2463464	123.03607	0.3184495	24.537178
4	0.2514906	58.00774	0.2742641	713.587811	0.2514906	58.00774	0.2742641	713.587811
5	0.2619816	125.13471	0.1799095	5.392916	0.2619816	125.13471	0.1799095	5.392916
6	0.2541872	123.84999	0.2465288	86.793430	0.2541872	123.84999	0.2465288	86.793430
7	0.2475771	94.02120	0.3056611	314.106584	0.2475771	94.02120	0.3056611	314.106584
8	0.2578809	125.84683	0.2125813	11.725520	0.2578809	125.84683	0.2125813	11.725520
9	0.2364931	113.44615	0.4167000	21.625733	0.2364931	113.44615	0.4167000	21.625733
10	0.2551243	134.96348	0.2463147	22.091167	0.2551243	134.96348	0.2463147	22.091167

Data svar4 quadratic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.9385611	4557.46404	0.8459894	5.473091	0.9385611	4557.46404	0.8459894	5.473091
2	0.8520740	4545.26369	1.6175590	49.417300	0.8520740	4545.26369	1.6175590	49.417300
3	0.9453360	5041.67486	0.8070249	37.923890	0.9453360	5041.67486	0.8070249	37.923890
4	0.9563461	4612.05093	0.6797805	142.711563	0.9563461	4612.05093	0.6797805	142.711563
5	0.9592929	4802.24825	0.6568567	222.657238	0.9592929	4802.24825	0.6568567	222.657238
6	0.9365191	4664.44506	0.8584897	21.647567	0.9365191	4664.44506	0.8584897	21.647567
7	0.9319079	4584.96886	0.8988553	14.934074	0.9319079	4584.96886	0.8988553	14.934074
8	0.9453780	4496.43446	0.7804820	25.584377	0.9453780	4496.43446	0.7804820	25.584377
9	0.9380318	4388.96551	0.8497544	55.063379	0.9380318	4388.96551	0.8497544	55.063379
10	0.8749809	59.36377	1.4135124	39892.676884	0.8749809	59.36377	1.4135124	39892.676884

Data svar1 cubic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	4.270310	0.003669648	3.396933	0.004467152	4.270310	0.003669648	3.396933	0.004467152
2	4.223039	0.003370999	3.699203	0.006610357	4.223039	0.003370999	3.699203	0.006610357
3	4.041561	0.003593693	5.422481	0.003774997	4.041561	0.003593693	5.422481	0.003774997
4	4.112768	0.003755428	4.666048	0.002613189	4.112768	0.003755428	4.666048	0.002613189
5	4.131681	0.003578299	4.641088	0.004207299	4.131681	0.003578299	4.641088	0.004207299
6	4.243201	0.003763652	3.500011	0.002201780	4.243201	0.003763652	3.500011	0.002201780
7	3.988493	0.003336344	6.198879	0.006333791	3.988493	0.003336344	6.198879	0.006333791
8	4.229390	0.003817927	3.740550	0.002468917	4.229390	0.003817927	3.740550	0.002468917
9	4.073950	0.003536372	5.113187	0.004454482	4.073950	0.003536372	5.113187	0.004454482
10	4.172333	0.003729495	4.119279	0.002885757	4.172333	0.003729495	4.119279	0.002885757

Data svar2 cubic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.02125813	23.72711	0.01224481	9.961797	0.02125813	23.72711	0.01224481	9.961797
2	0.02002471	23.25074	0.02364672	12.226918	0.02002471	23.25074	0.02364672	12.226918
3	0.01948947	23.99300	0.02910702	7.563324	0.01948947	23.99300	0.02910702	7.563324
4	0.02034287	25.69987	0.02054823	5.025499	0.02034287	25.69987	0.02054823	5.025499
5	0.02095230	24.00323	0.01555265	6.178099	0.02095230	24.00323	0.01555265	6.178099
6	0.01948935	20.89310	0.02874397	27.992605	0.01948935	20.89310	0.02874397	27.992605
7	0.02011949	19.25745	0.02286465	33.318363	0.02011949	19.25745	0.02286465	33.318363
8	0.01982152	16.71469	0.02587386	61.075620	0.01982152	16.71469	0.02587386	61.075620
9	0.02015293	24.96468	0.02243762	6.354434	0.02015293	24.96468	0.02243762	6.354434
10	0.02126954	21.67581	0.01268473	61.534358	0.02126954	21.67581	0.01268473	61.534358

Data svar3 cubic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.2533716	128.34563	0.2528288	6.727415	0.2533716	128.34563	0.2528288	6.727415
2	0.2645048	129.38004	0.1528505	25.285012	0.2645048	129.38004	0.1528505	25.285012
3	0.2460586	131.32110	0.3245226	25.838104	0.2460586	131.32110	0.3245226	25.838104
4	0.2514790	57.10460	0.2743430	705.479231	0.2514790	57.10460	0.2743430	705.479231
5	0.2618990	120.70900	0.1812069	5.563901	0.2618990	120.70900	0.1812069	5.563901
6	0.2541631	121.55116	0.2468463	85.954359	0.2541631	121.55116	0.2468463	85.954359
7	0.2475690	94.82001	0.3059494	320.392146	0.2475690	94.82001	0.3059494	320.392146
8	0.2578014	121.52930	0.2137854	11.871095	0.2578014	121.52930	0.2137854	11.871095
9	0.2361658	105.36369	0.4219721	22.292077	0.2361658	105.36369	0.4219721	22.292077
10	0.2550828	138.30349	0.2475221	22.002891	0.2550828	138.30349	0.2475221	22.002891

Data svar4 cubic model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.9306341	4476.04559	0.8252199	8.538462	0.9306341	4476.04559	0.8252199	8.538462
2	0.8474275	4531.51076	1.5731561	53.071168	0.8474275	4531.51076	1.5731561	53.071168
3	0.9276743	4902.60625	0.8845715	42.624053	0.9276743	4902.60625	0.8845715	42.624053
4	0.9446416	4532.18005	0.6936165	150.839617	0.9446416	4532.18005	0.6936165	150.839617
5	0.9460646	4728.05334	0.6866626	243.967933	0.9460646	4728.05334	0.6866626	243.967933
6	0.9274613	4578.90432	0.8465041	24.823672	0.9274613	4578.90432	0.8465041	24.823672
7	0.9211508	4483.88959	0.9025555	15.337139	0.9211508	4483.88959	0.9025555	15.337139
8	0.9374763	4426.87671	0.7590386	27.255454	0.9374763	4426.87671	0.7590386	27.255454
9	0.9286686	4334.22629	0.8404128	56.352186	0.9286686	4334.22629	0.8404128	56.352186
10	0.8702402	62.63017	1.3702961	39303.470641	0.8702402	62.63017	1.3702961	39303.470641

For the data set three and four, the errors are remarkably improved while using polynomial models.

4. Reduce data and observe how the performance will change.

I reduce the four training data sets by half.

Data svar1 linear model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	4.427272	0.004067292	1.678542	0.002473867	4.427272	0.004067292	1.678542	0.002473867
2	4.148367	0.003839852	4.169394	0.004032987	4.148367	0.003839852	4.169394	0.004032987
3	3.993164	0.003056434	5.601510	0.011660303	3.993164	0.003056434	5.601510	0.011660303
4	4.402573	0.004026449	1.777036	0.001482598	4.402573	0.004026449	1.777036	0.001482598
5	4.082272	0.003905863	4.754983	0.002747129	4.082272	0.003905863	4.754983	0.002747129
6	3.835106	0.003541127	6.960692	0.005534486	3.835106	0.003541127	6.960692	0.005534486
7	4.035796	0.003911365	5.099815	0.002597701	4.035796	0.003911365	5.099815	0.002597701
8	4.065618	0.003891349	5.058077	0.002791670	4.065618	0.003891349	5.058077	0.002791670
9	4.061582	0.003705695	4.976767	0.003991719	4.061582	0.003705695	4.976767	0.003991719
10	4.234884	0.003884130	3.294980	0.003006521	4.234884	0.003884130	3.294980	0.003006521

Data svar2 linear model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.06114549	12.243821	0.02588202	5.771790	0.06114549	12.243821	0.02588202	5.771790
2	0.05534002	8.910910	0.08217694	1.021912	0.05534002	8.910910	0.08217694	1.021912
3	0.05829130	9.766713	0.05107390	14.229271	0.05829130	9.766713	0.05107390	14.229271
4	0.05644248	10.919658	0.07428817	29.955432	0.05644248	10.919658	0.07428817	29.955432
5	0.05924266	11.228718	0.04495742	21.823695	0.05924266	11.228718	0.04495742	21.823695
6	0.05848936	11.319530	0.04919545	2.707636	0.05848936	11.319530	0.04919545	2.707636
7	0.06122983	11.766545	0.02579368	13.908537	0.06122983	11.766545	0.02579368	13.908537
8	0.05394160	10.320387	0.09050835	2.467881	0.05394160	10.320387	0.09050835	2.467881
9	0.05368904	7.719222	0.09498930	15.736590	0.05368904	7.719222	0.09498930	15.736590
10	0.05555355	9.815526	0.07758566	4.484509	0.05555355	9.815526	0.07758566	4.484509

Data svar3 linear model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	0.4633249	116.59748	0.4137518	13.151629	0.4633249	116.59748	0.4137518	13.151629
2	0.4421738	103.16440	0.6050561	12.470957	0.4421738	103.16440	0.6050561	12.470957
3	0.4586416	108.97271	0.4572010	48.833624	0.4586416	108.97271	0.4572010	48.833624
4	0.4720335	109.36816	0.3336586	21.131071	0.4720335	109.36816	0.3336586	21.131071
5	0.4103972	97.90087	0.9006855	16.383021	0.4103972	97.90087	0.9006855	16.383021
6	0.4602555	130.74730	0.4546629	10.650961	0.4602555	130.74730	0.4546629	10.650961
7	0.4962489	18.88510	0.1197591	908.870352	0.4962489	18.88510	0.1197591	908.870352
8	0.4454350	109.57218	0.5736835	7.951342	0.4454350	109.57218	0.5736835	7.951342
9	0.4508348	95.83259	0.5346078	1.229209	0.4508348	95.83259	0.5346078	1.229209
10	0.4769507	108.56157	0.2934969	36.147225	0.4769507	108.56157	0.2934969	36.147225

Data svar4 linear model:

	trainingMSE	trainingRSE	testingMSE	testingRSE	lmTrainingMSE	lmTrainingRSE	lmTestingMSE	lmTestingRSE
1	1.1822316	152.81059	0.5534880	7.958363	1.1822316	152.81059	0.5534880	7.958363
2	1.0476951	133.32794	1.8062567	117.762399	1.0476951	133.32794	1.8062567	117.762399
3	1.0392333	183.47859	1.9077896	24.208698	1.0392333	183.47859	1.9077896	24.208698
4	0.9929642	130.16289	2.2786013	110.663501	0.9929642	130.16289	2.2786013	110.663501
5	1.1842099	150.56469	0.5444580	0.505386	1.1842099	150.56469	0.5444580	0.505386
6	1.1647006	172.47607	0.7456200	46.072268	1.1647006	172.47607	0.7456200	46.072268
7	1.0499704	146.87302	1.7745808	5.290430	1.0499704	146.87302	1.7745808	5.290430
8	1.1971853	85.77198	0.4241572	714.956582	1.1971853	85.77198	0.4241572	714.956582
9	1.1906913	125.18466	0.4886600	329.941669	1.1906913	125.18466	0.4886600	329.941669
10	1.1193879	141.41194	1.1288368	184.147240	1.1193879	141.41194	1.1288368	184.147240

After reducing the input data, we can see most of recorded errors are greater than before.

Multivariate regression

1. Map the multivariate data sets to high dimensions.

Data mvar1 (15 out of 2500 observations):

	1	x1	x1^2	x2	x2^2	x1*x2
1	1	1.67346939	2.800499792	0.44897959	0.201582674	0.751353603
2	1	-0.04081633	0.001665973	0.53061224	0.281549354	-0.021657643
3	1	-0.77551020	0.601416077	-1.59183673	2.533944190	1.234485631
4	1	-2.00000000	4.000000000	-1.34693878	1.814244065	2.693877551
5	1	0.77551020	0.601416077	-0.44897959	0.201582674	-0.348188255
6	1	-1.26530612	1.600999584	1.51020408	2.280716368	-1.910870471
7	1	-1.83673469	3.373594336	-0.61224490	0.374843815	1.124531445
8	1	-1.42857143	2.040816327	1.75510204	3.080383174	-2.507288630
9	1	-0.53061224	0.281549354	2.00000000	4.000000000	-1.061224490
10	1	1.59183673	2.533944190	-0.04081633	0.001665973	-0.064972928
11	1	0.93877551	0.881299459	1.59183673	2.533944190	1.494377343
12	1	-1.75510204	3.080383174	0.61224490	0.374843815	-1.074552270
13	1	0.44897959	0.201582674	-0.61224490	0.374843815	-0.274885464
14	1	1.91836735	3.680133278	-0.61224490	0.374843815	-1.174510621
15	1	-0.69387755	0.481466056	-0.77551020	0.601416077	0.538109121

Data mvar2 (15 out of 2500 observations):

	1	x1	x1^2	x2	x2^2	x1*x2
1	1	-0.69387755	0.481466056	1.34693878	1.814244065	-0.934610579
2	1	-0.93877551	0.881299459	1.75510204	3.080383174	-1.647646814
3	1	-1.42857143	2.040816327	-1.18367347	1.401082882	1.690962099
4	1	0.12244898	0.014993753	1.59183673	2.533944190	0.194918784
5	1	2.00000000	4.000000000	0.36734694	0.134943773	0.734693878
6	1	0.20408163	0.041649313	1.51020408	2.280716368	0.308204915
7	1	-1.02040816	1.041232820	-0.20408163	0.041649313	0.208246564
8	1	0.44897959	0.201582674	0.36734694	0.134943773	0.164931279
9	1	-0.69387755	0.481466056	1.26530612	1.600999584	-0.877967514
10	1	-2.00000000	4.000000000	0.36734694	0.134943773	-0.734693878
11	1	-0.85714286	0.734693878	-0.04081633	0.001665973	0.034985423
12	1	1.10204082	1.214493961	0.28571429	0.081632653	0.314868805
13	1	-1.34693878	1.814244065	0.53061224	0.281549354	-0.714702207
14	1	-1.91836735	3.680133278	-1.10204082	1.214493961	2.114119117
15	1	1.42857143	2.040816327	-0.93877551	0.881299459	-1.341107872
16	1	-1.42857143	2.040816327	-1.67346939	2.800499792	2.390670554

Data mvar3 (15 out of 100000 observations)

	1	x1	x2	x3	x4	x5	x1*x2	x3*x4	x1*x5
1	1	1.4587176926	8.490879e-01	1.070332831	1.678542238	-1.3274005963	1.2385795190	1.7965988655	-1.9363027351
2	1	-0.4414873987	-2.896533e-02	-0.890986529	0.980678189	-2.0022396995	0.0127878263	-0.8737710557	0.8839635964
3	1	-0.9852445238	-5.361219e-01	-0.175666910	-1.728023133	-0.7093127840	0.5282111553	0.3035564848	0.6988465360
4	1	-1.7098580462	1.390245e+00	-0.028098452	0.912877538	-0.4554935502	-2.3771216810	-0.0256504456	0.7788293117
5	1	-1.2958784083	1.661739e+00	-1.181145302	-0.867208617	-0.1497359789	-2.1534120803	1.0242993836	0.1940396221
6	1	-1.5902872542	1.724966e-01	2.053764372	-1.703158652	1.9211598646	-0.2743191086	-3.4978865611	-3.0551960460
7	1	0.0999814400	1.867485e+00	-0.792276777	1.715577029	1.4711394503	0.1867138363	-1.3592118391	0.1470866406
8	1	-1.9127889713	-2.045926e+00	1.280442251	0.327067969	-1.9526831937	3.9134253157	0.4187916472	3.7350708774
9	1	-1.9899357210	6.688937e-01	-0.947050249	0.770355137	0.7643749993	-1.3310554120	-0.7295650242	-1.5210571152
10	1	-0.4788208354	-2.829533e-02	1.608914266	-0.576222779	0.8336169767	0.0135483914	-0.9270930495	-0.3991531772
11	1	1.7141944844	-6.153980e-01	-1.126845857	-0.849477041	-0.8566538716	-1.0549118156	0.9572296841	-1.4684713418
12	1	-0.5696008304	8.060623e-01	-1.738626263	-0.450661216	-1.8844485711	-0.4591337493	0.7835314263	1.0733834709
13	1	-1.4151465217	-1.176230e+00	2.245301711	1.824341578	-1.7182520420	1.6645371228	4.0961972662	2.4315784006
14	1	-2.0708418169	2.200608e+00	-0.761365051	0.893707177	-0.5286457996	-4.5571116234	-0.6804374109	1.0947418282
15	1	1.5543147596	-1.409907e+00	-2.187856857	0.820634456	-1.3746621794	-2.1914398880	-1.7954307209	-2.1366577149

Data mvar4 (15 out of 100000 observations)

	1	x1	x2	x3	x4	x5	x1*x2	x3*x4	x1*x5
1	1	-3.910563e-01	0.692569193	0.6318149811	0.1930049879	-0.609675625	-2.708336e-01	1.219434e-01	2.384175e-01
2	1	9.952647e-02	-1.733582655	-1.1890098092	-1.2895160135	-2.001739015	-1.725374e-01	1.533247e+00	-1.992260e-01
3	1	-3.290092e-01	1.186568731	2.1386001544	-0.5758584463	-2.245578841	-3.903920e-01	-1.231531e+00	7.388161e-01
4	1	-1.829362e+00	-0.293887876	-1.1949141007	-0.4312863581	1.720009964	5.376274e-01	5.153502e-01	-3.146521e+00
5	1	-6.178816e-01	0.511744803	-2.0162515650	1.8364944534	-0.763072887	-3.161977e-01	-3.702835e+00	4.714887e-01
6	1	-7.884587e-01	-1.808621312	-1.6853950277	-0.4725349195	0.795075693	1.426023e+00	7.964080e-01	-6.268844e-01
7	1	1.788307e+00	1.964275271	-1.4630737305	1.1938314964	0.888925993	3.512728e+00	-1.746664e+00	1.589673e+00
8	1	-6.912903e-01	1.247682184	-0.7243615630	1.5021907394	-1.581625456	-8.625105e-01	-1.088129e+00	1.093362e+00
9	1	8.590709e-01	-1.409288247	1.5291626042	-0.2957805624	-0.582520882	-1.210679e+00	-4.522966e-01	-5.004268e-01
10	1	6.236646e-01	-0.310730507	0.5920623629	-1.2479845613	0.089252274	-1.937916e-01	-7.388847e-01	5.566349e-02
11	1	-2.015315e+00	-1.422433002	-1.5780991952	1.8122616028	-2.079108814	2.866651e+00	-2.859929e+00	4.190059e+00
12	1	-6.513013e-01	-0.390073004	1.6773414752	-0.1750014882	-0.027388041	2.540551e-01	-2.935373e-01	1.783787e-02
13	1	2.015623e+00	1.091154736	0.2451327687	-0.4333579433	1.369423826	2.199356e+00	-1.062302e-01	2.760242e+00
14	1	-7.994222e-01	-2.299705269	0.3685436250	-0.6644049802	0.238196073	1.838436e+00	-2.448622e-01	-1.904192e-01
15	1	-1.889990e+00	-1.080530912	0.5235315333	0.5538991916	-0.536658874	2.042193e+00	2.899837e-01	1.014280e+00

2. Perform linear regressions on these data sets.

I perform linear regression for each data set based on the high dimensional matrices I got (which are shown above). The training MSE and testing MSE are as follow.

Data mvar1:

	trainingMSE	testingMSE
1	0.2575996	0.2647846
2	0.2581962	0.2591663
3	0.2609733	0.2353852
4	0.2593653	0.2484765
5	0.2569973	0.2699720
6	0.2553064	0.2868107
7	0.2571894	0.2681744
8	0.2629283	0.2165531
9	0.2558428	0.2802554
10	0.2574470	0.2665882

Data mvar2:

	trainingMSE	testingMSE
1	0.01995401	0.01955221
2	0.01949582	0.02373106
3	0.01976644	0.02131389
4	0.01974891	0.02159134
5	0.02004716	0.01870460
6	0.02006092	0.01858346
7	0.01998180	0.01928381
8	0.01987563	0.02022553
9	0.01995556	0.01955953
10	0.02014142	0.01783052

Data mvar3:

	trainingMSE	testingMSE
1	0.2512059	0.2465174
2	0.2505806	0.2521235
3	0.2505492	0.2524511
4	0.2506433	0.2515661
5	0.2503637	0.2540955
6	0.2506605	0.2514050
7	0.2507773	0.2503540
8	0.2510034	0.2483233
9	0.2509177	0.2491164
10	0.2505966	0.2519994

Data mvar4:

	trainingMSE	testingMSE
1	0.004222307	0.003890356
2	0.004163933	0.004415828
3	0.004189558	0.004185143
4	0.004190329	0.004178483
5	0.004193873	0.004146326
6	0.004183085	0.004243283
7	0.004175888	0.004308300
8	0.004201186	0.004080431
9	0.004194578	0.004140181
10	0.004176044	0.004306987

3. Implement an iterative solution and compare it with the explicit one.

I implemented the iterative solution by using the Gradient Descent Algorithm.

The results are as follow.

Data mvar1:

	trainingMSE	testingMSE
1	0.2575996	0.2647845
2	0.2581962	0.2591663
3	0.2609733	0.2353853
4	0.2593653	0.2484766
5	0.2569973	0.2699720
6	0.2553064	0.2868107
7	0.2571894	0.2681744
8	0.2629283	0.2165530
9	0.2558428	0.2802554
10	0.2574470	0.2665882

Data mvar2:

	trainingMSE	testingMSE
1	0.01995401	0.01955220
2	0.01949582	0.02373104
3	0.01976644	0.02131394
4	0.01974891	0.02159128
5	0.02004716	0.01870461
6	0.02006092	0.01858346
7	0.01998180	0.01928382
8	0.01987563	0.02022553
9	0.01995556	0.01955957
10	0.02014142	0.01783051

Data mvar3:

	trainingMSE	testingMSE
1	0.2512059	0.2465174
2	0.2505806	0.2521235
3	0.2505492	0.2524511
4	0.2506433	0.2515661
5	0.2503637	0.2540955
6	0.2506605	0.2514050
7	0.2507773	0.2503540
8	0.2510034	0.2483233
9	0.2509177	0.2491164
10	0.2505966	0.2519994

Data mvar4:

	trainingMSE	testingMSE
1	0.004222307	0.003890356
2	0.004163933	0.004415828
3	0.004189558	0.004185143
4	0.004190329	0.004178483
5	0.004193873	0.004146326
6	0.004183085	0.004243283
7	0.004175888	0.004308300
8	0.004201186	0.004080431
9	0.004194578	0.004140181
10	0.004176044	0.004306987

Compared with the explicit solution, the MSE of the two solutions are similar. However, when I was executing the gradient descent algorithm on the data sets “mvar3” and “mvar4”, I found it took much more time than I spent in running the explicit algorithm. Therefore, for sufficiently large data sets, we can’t ignore the running time if we choose to use the Gradient Descent Algorithm.