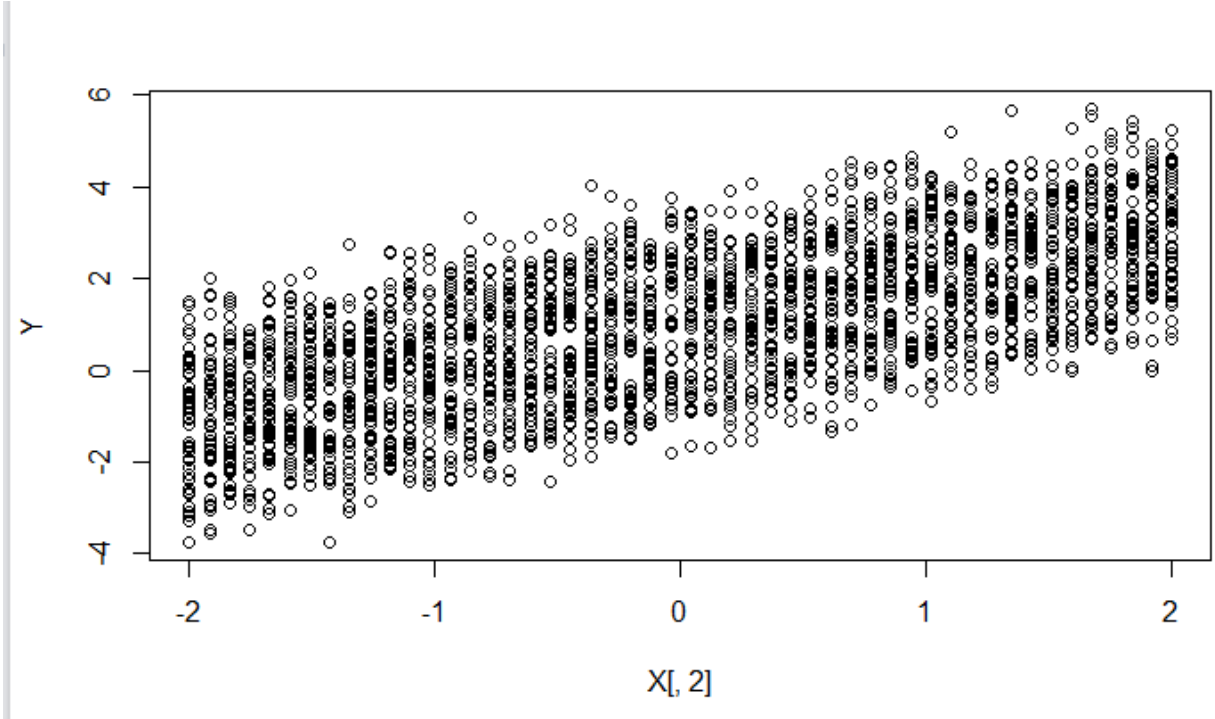
Name: Nirmal kumar Ravi

Id: A20320832

Dataset used: mvar-set1.dat

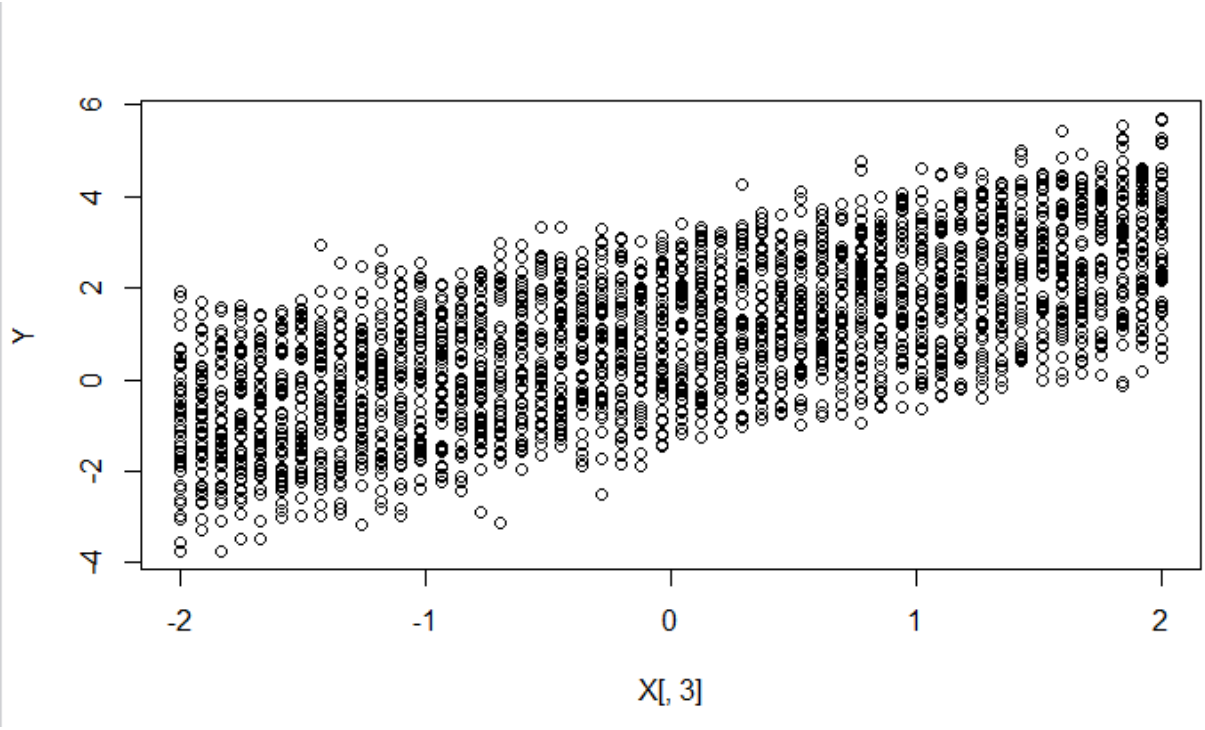
Let's plot the graph and see how data looks like

Feature 1 with target variable



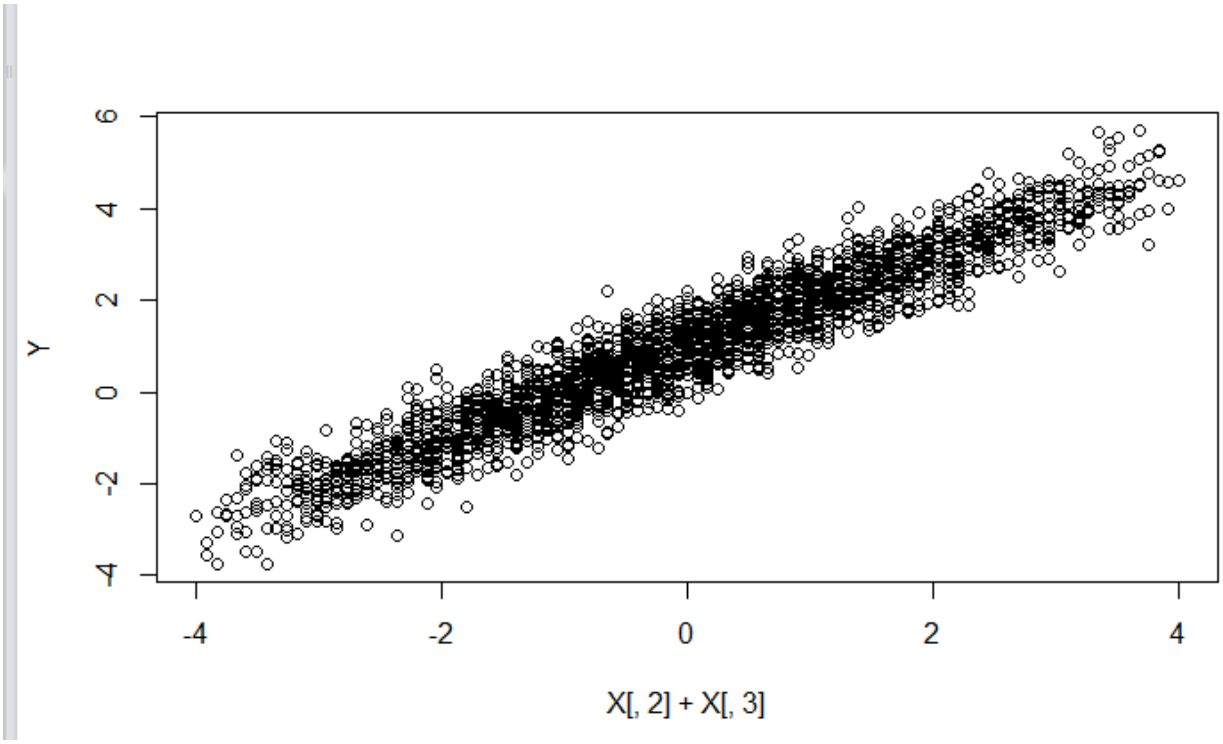
As we can see the data appears somewhat linear but has lot of noise

Feature 2 with target variable



Neither this be a good predictor but let's train our model without mapping to higher dimensions

Let's try **adding feature 1 and feature 2**



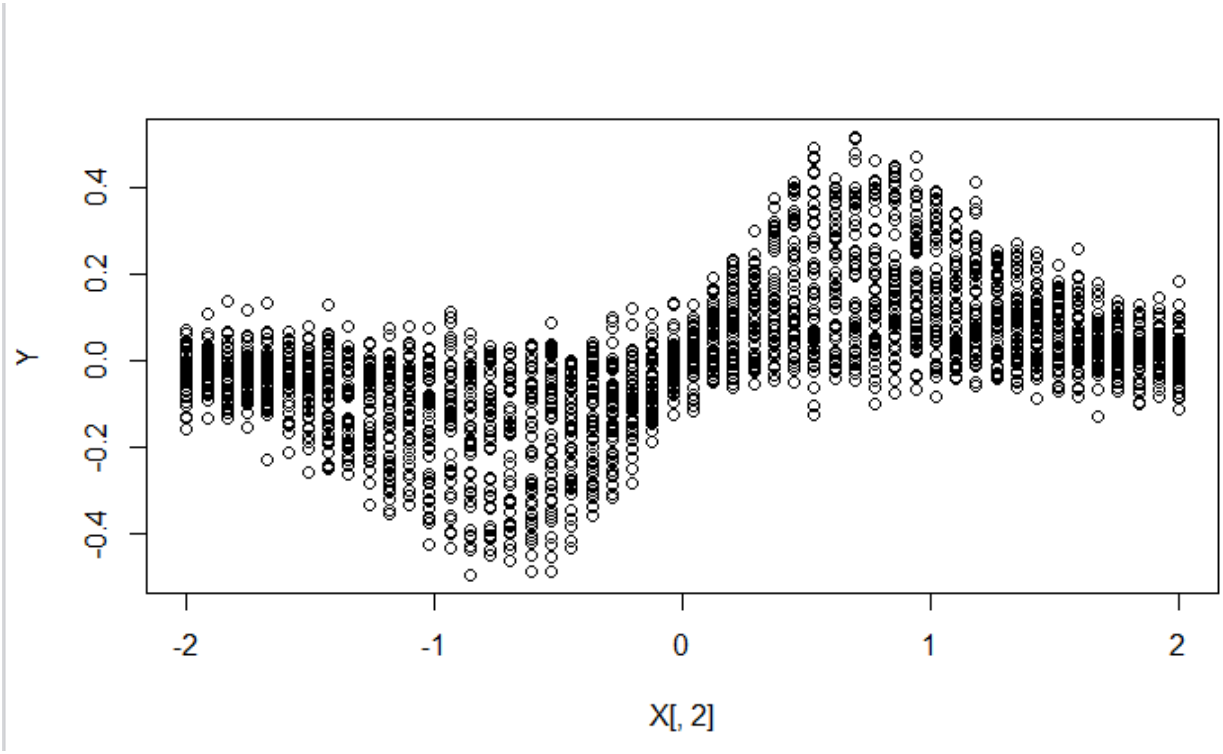
Here we got good **linear spread of data**. We can fit linear model in it.

Mean Squared Error	
Test Set	0.2637052
Train Set	0.2573266

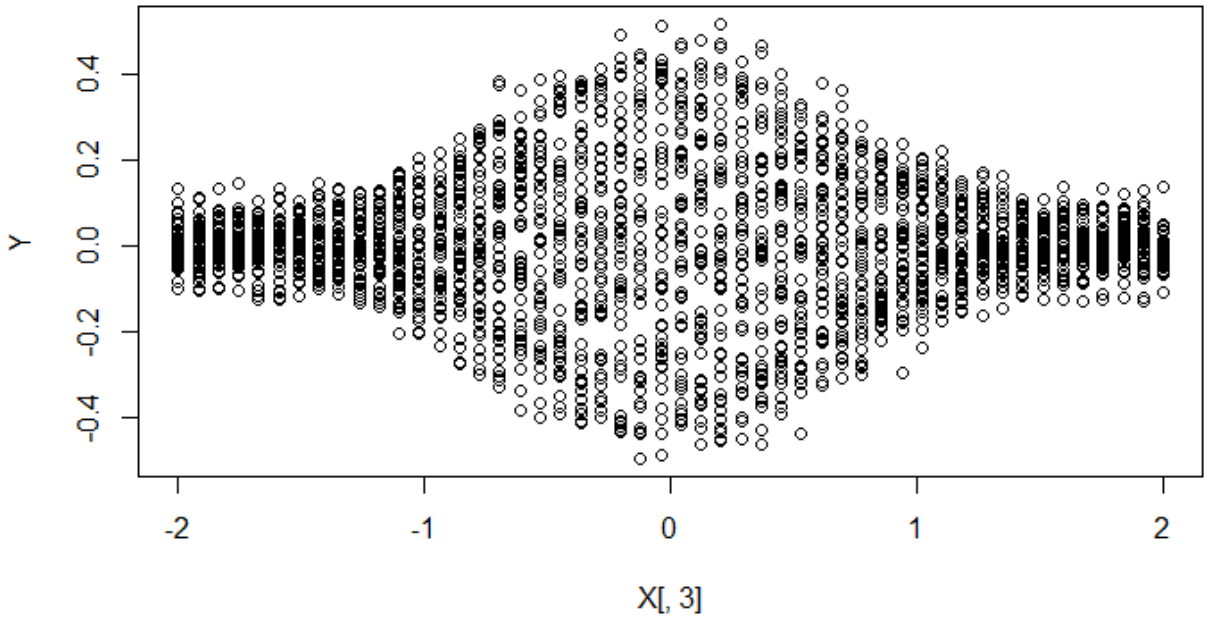
Dataset used: mvar-set2.dat

Let us plot the data and see how it looks like

Plot of feature 1 with target variable

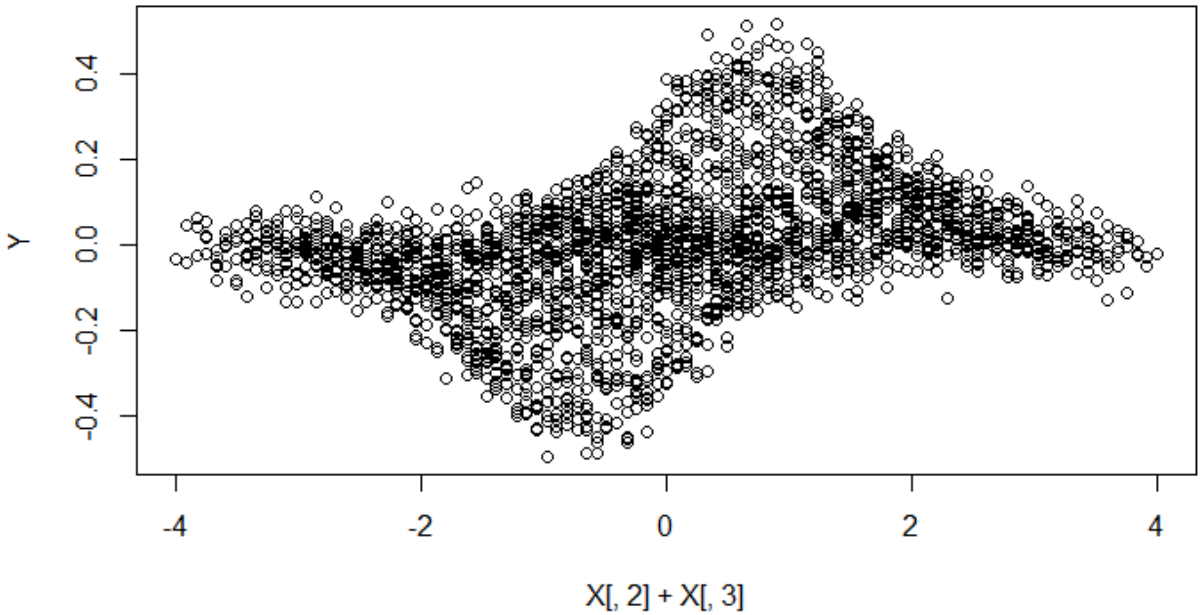


Plot of feature 2 to target variable



From the above plot we can see some sort of bell curve. Probably a Gaussian function

Let's combine feature 1 and feature 2



Clearly we can see a bell shaped curve.

Let us fix a model without any mapping to higher dimensions and see MSE error

By doing this we get MSE error on test set as 0.01933999 and train set as 0.0201085. Let us see we can map do any better by mapping the features to higher dimensions and do any better

One usual approach is to **add features** let's add two features X1+X2 and check MSE error. The MSE error is 0.02253033. The MSE error increased from 0.01 to 0.02, so this is isn't better model

Let's try with **multivariate Gaussian function and** fit a model to it. **The MSE error is 0.00187** which is much lesser that all our previous models. So this seems to be a nice fit for the data.

Dataset used: mvar-set3.dat

In this data set we have four features to predict with. One problem of having more features is, it is hard to visualize how data looks like.

Let us first fit our model **without doing any feature mapping to higher dimension**.

As we expect the MSE is very high 0.2509218. Let's try mapping features to higher

We shall add features and see that we can do any better. Here we are adding two new features V7=V1+V2 and V8=V3+V4

Let's check MSE error and do 10 fold cross validation to see our model is any better than previous. After model fit the MSE error has reduced from 0.25 to 0.23 and mean MSE error of cross validation is also reduced.

Here **we have six features we have 6! Way of combining them**. Here we tried one and made MSE error lesser

Dataset used: mvar-set4.dat

This data set is bigger than previous one. We have 100000 rows here.

We add two features V7 = V1 + V2 +V4 and V8= V3+ V5

We trained the model with this and we see a considerable amount of reduction in MSE after 10 fold cross validation. Again as mentioned above there are 6! Ways of combining the features and we haven't tried all of those. We get 0.2509218 MSE without increasing the dimension and it has reduced 20% after adding two features.

In this we compare time **and accuracy of explicit and recursive solutions**.

- The iterative solution is slower than explicit solution
- As our explicit solution requires matrix inversion and as all the matrices are not invertible , we can go for iterative solution
- Before doing the iterative solution we have to normalize our features.

Gaussian Kernel – linear regression

- We have performed linear regression using Gaussian kernel.
- The accuracy of Gaussian kernel is better than explicit solution
- But Performance is bit slow as we have to transform the values to higher dimension using Gaussian kernel function.