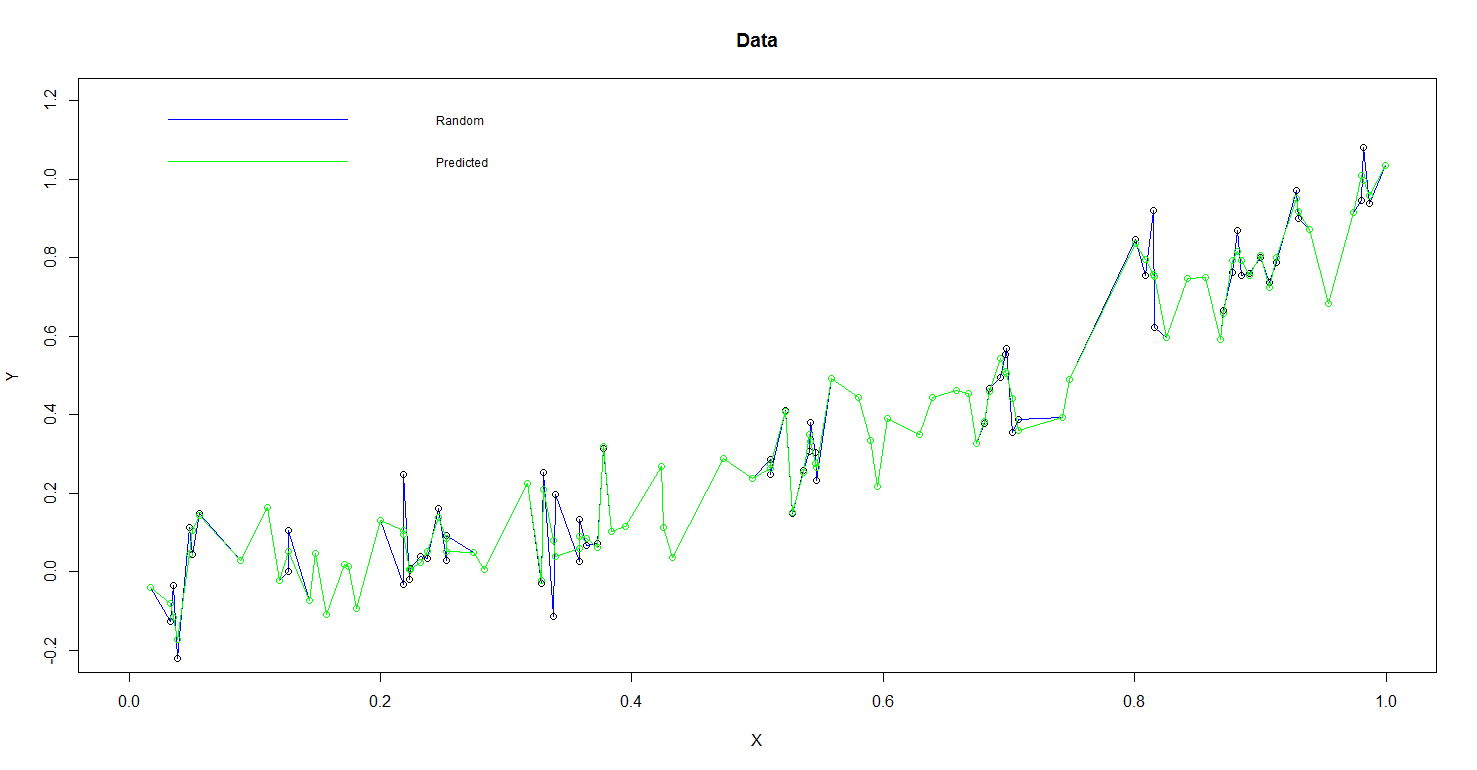
# Statistics 202A

## Anurag Pande – 604749647

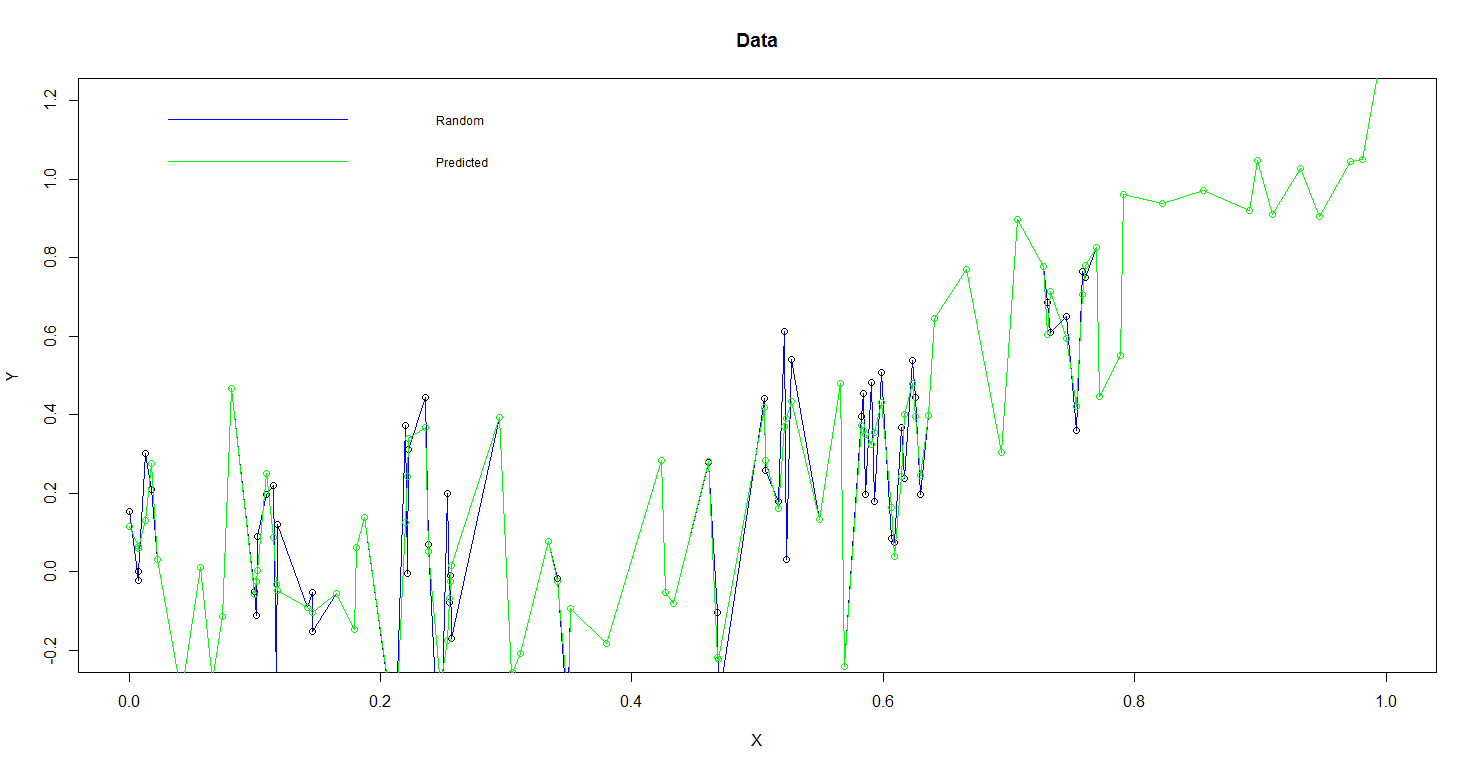
Input data is taken as a randomized set of 100 values in the range of 0 to 1 as the X co-ordinate, and Y co-ordinate as

Regularization can be observed as how close the predicted co-ordinates are to the predicted values. The value of lambda changes the regularization, fitting the predicted values differently to the actual dataset.

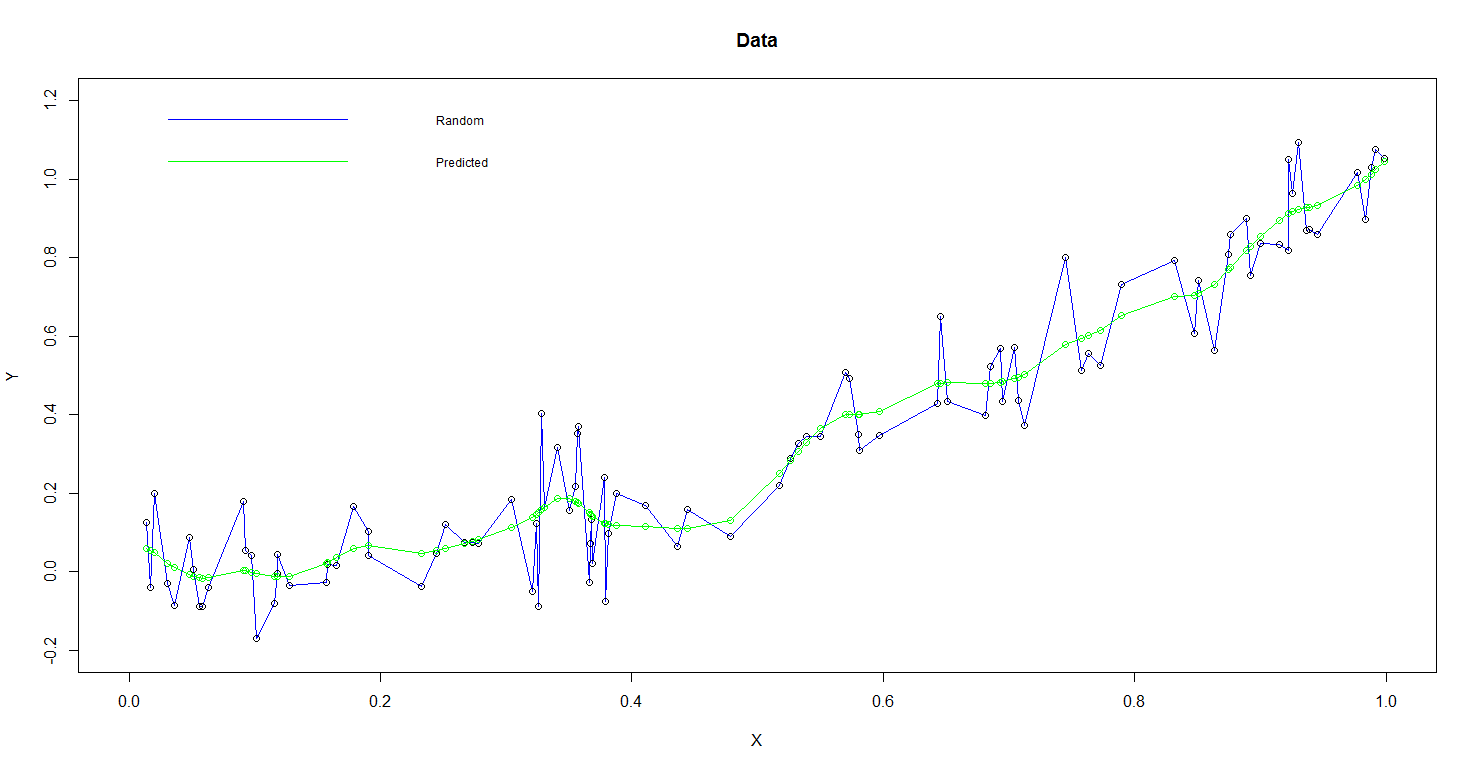
We can see that values of lambda > 1 do not produce quite accurate results, while values of lambda < 0.01 overfit the model too much.



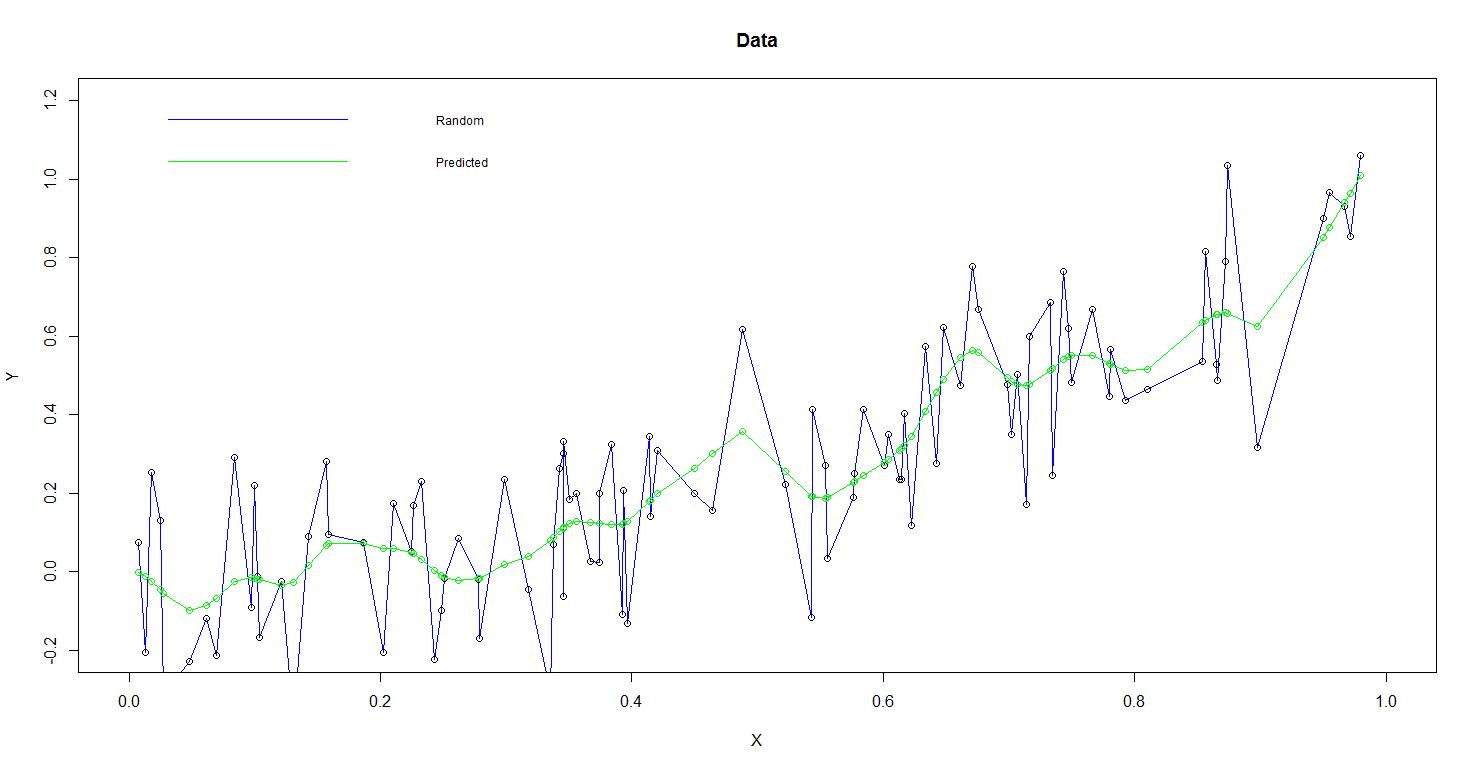
Sigma = 0.1, Lambda = 0.0



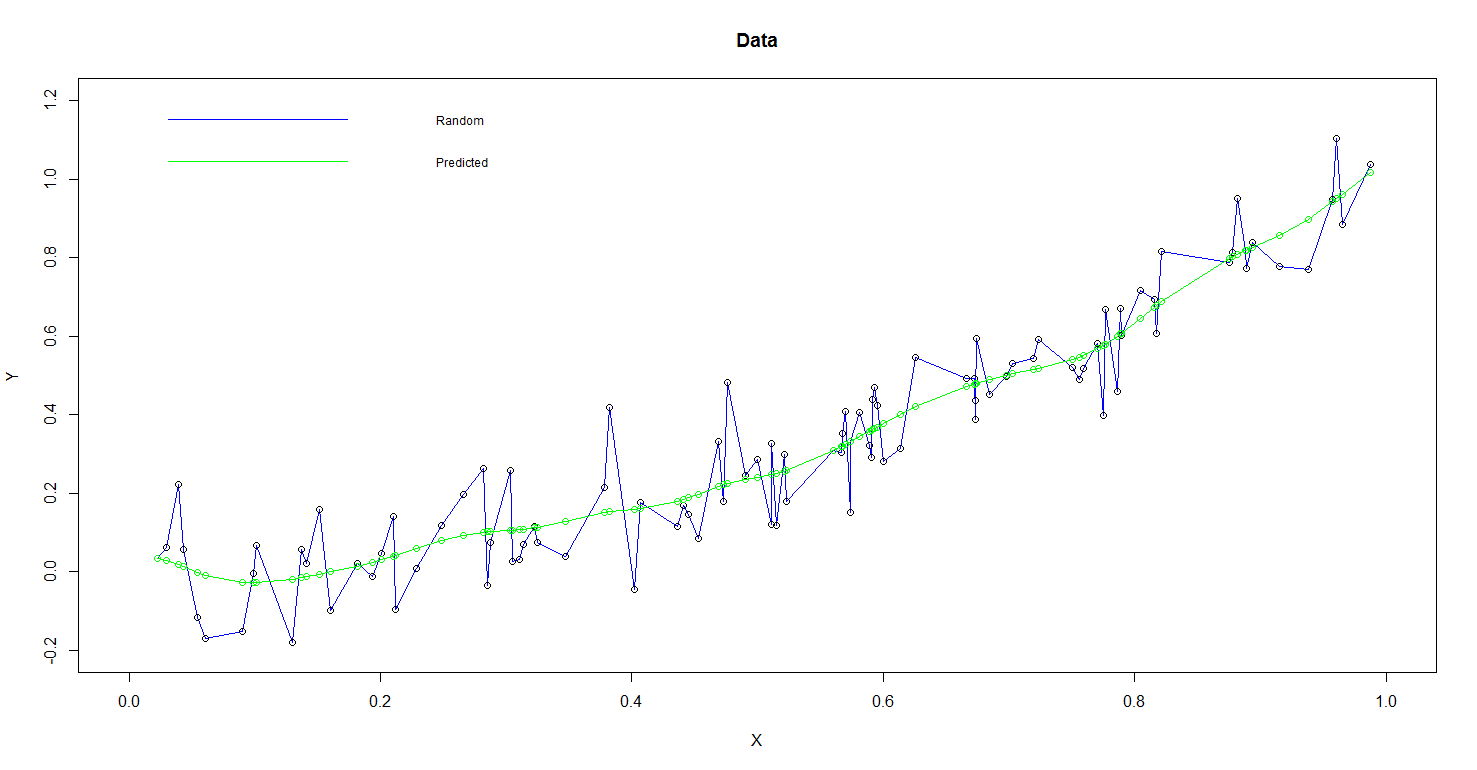
Sigma = 0.2, Lambda = 0.0



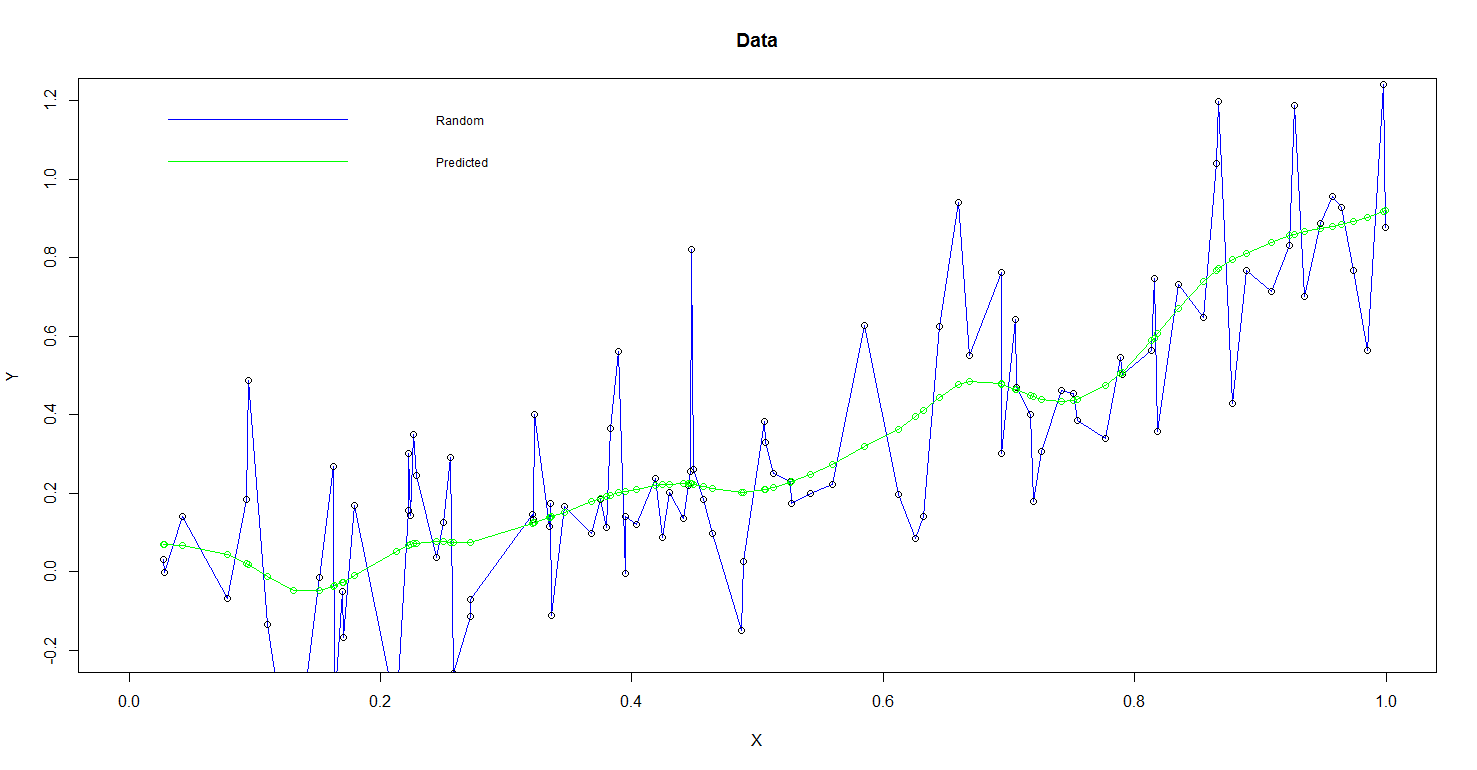
Sigma = 0.1, Lambda = 0.001



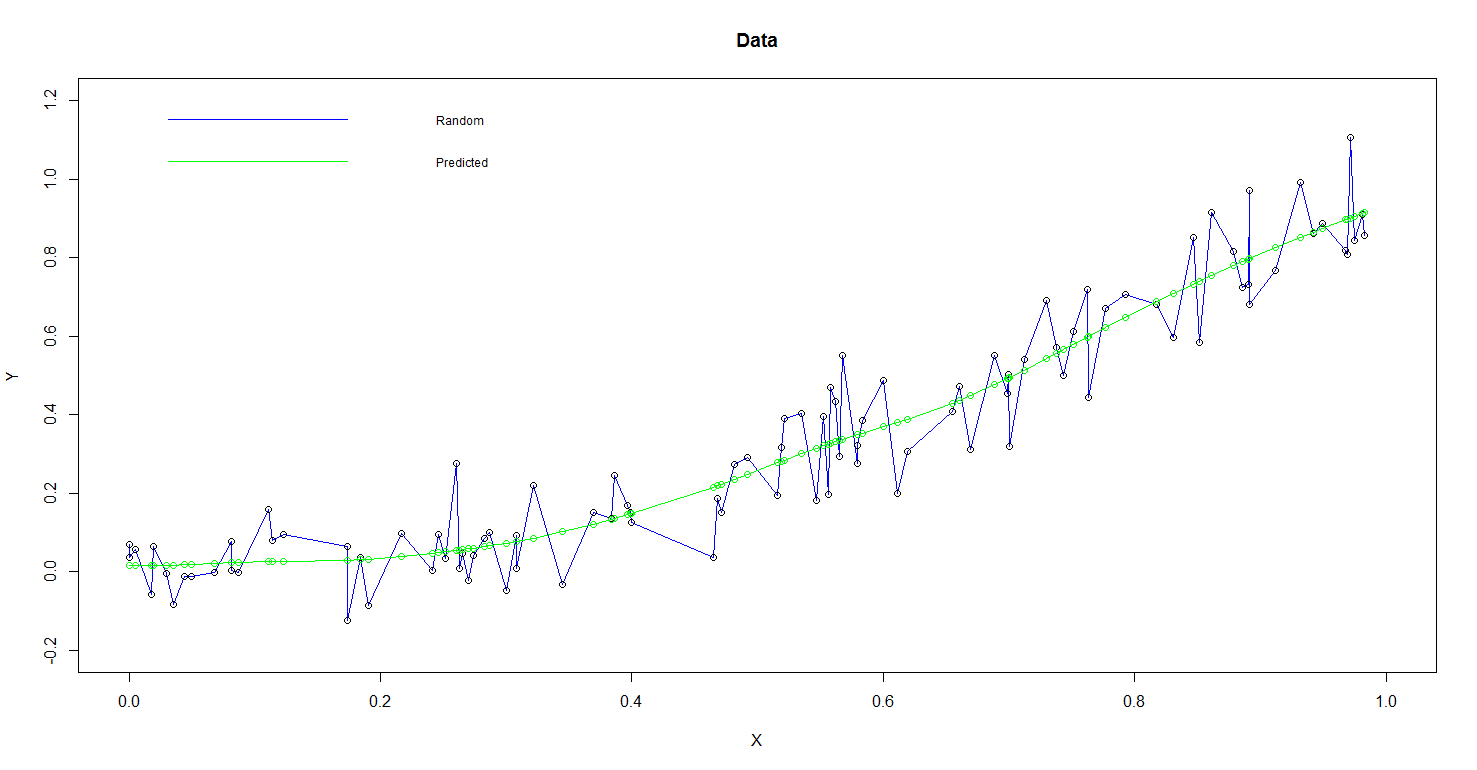
Sigma = 0.2, Lambda = 0.001



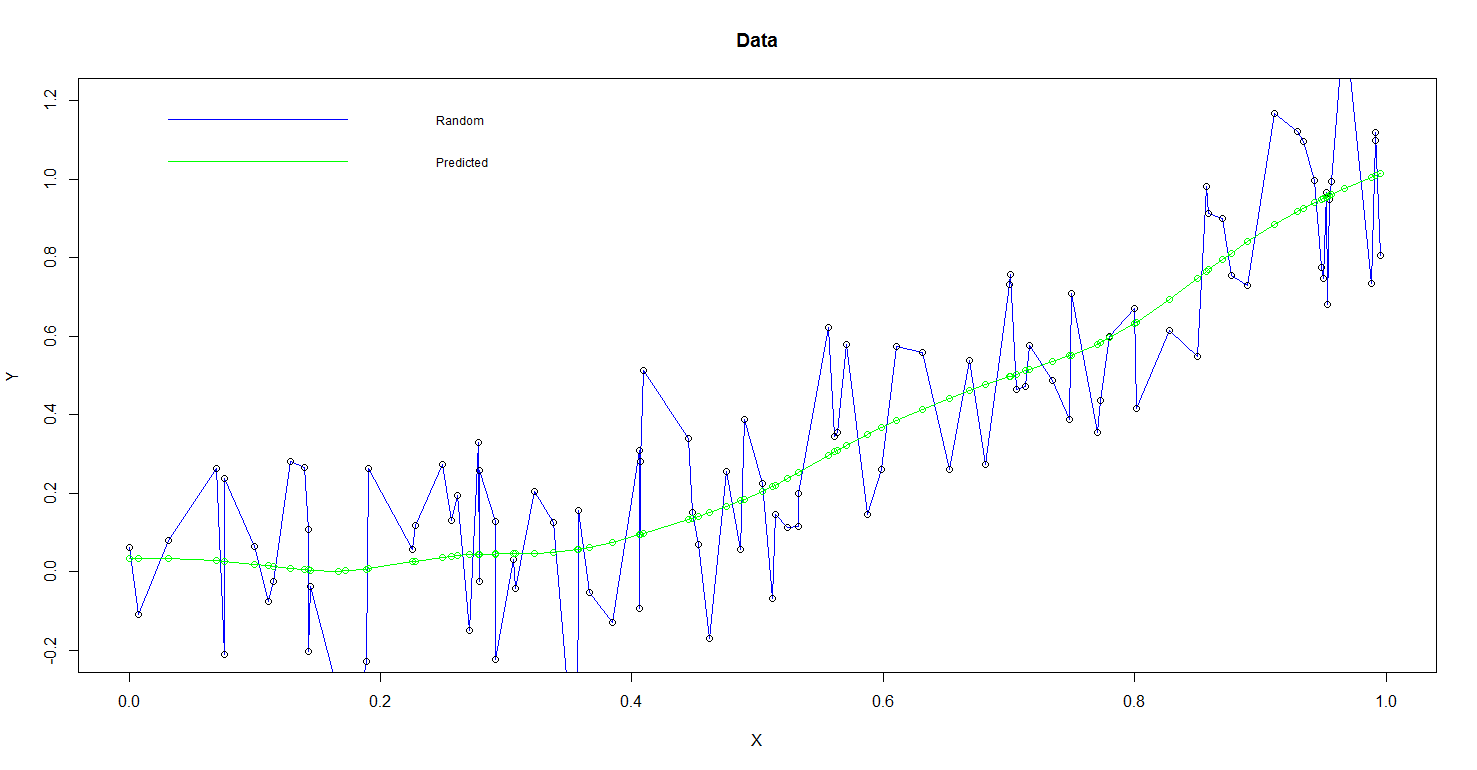
Sigma = 0.1, Lambda = 0.01



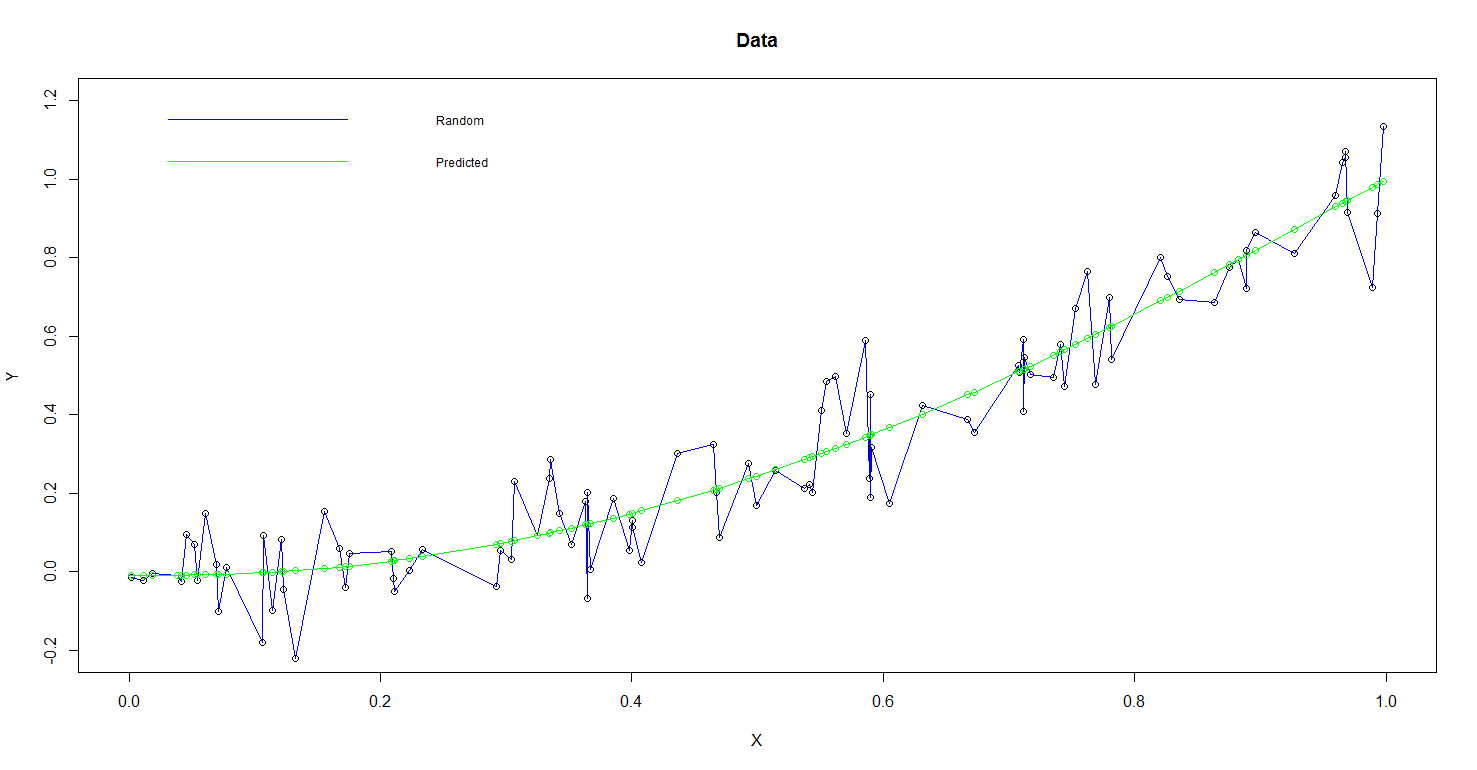
Sigma = 0.2, Lambda = 0.01



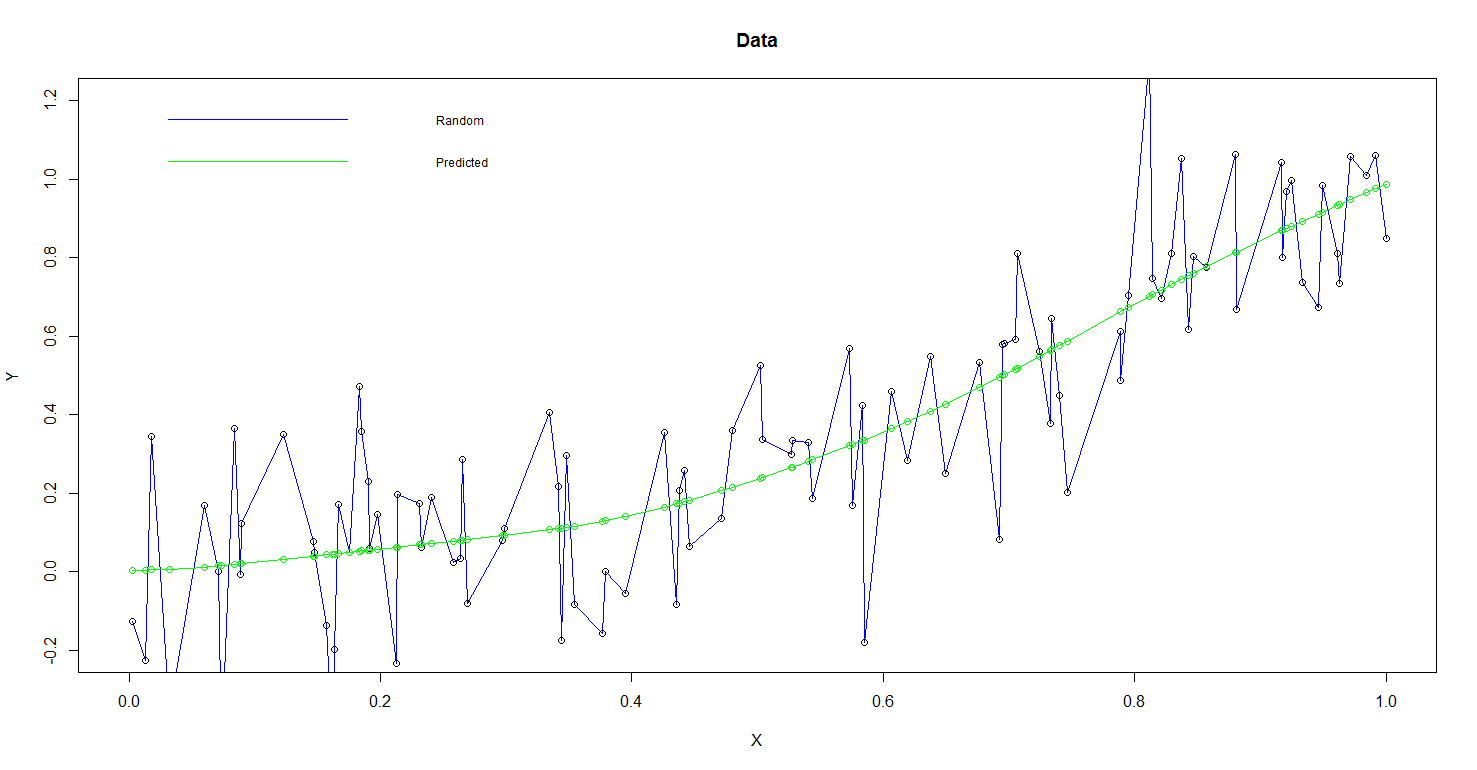
Sigma = 0.1, Lambda = 0.1



Sigma = 0.2, Lambda = 0.1

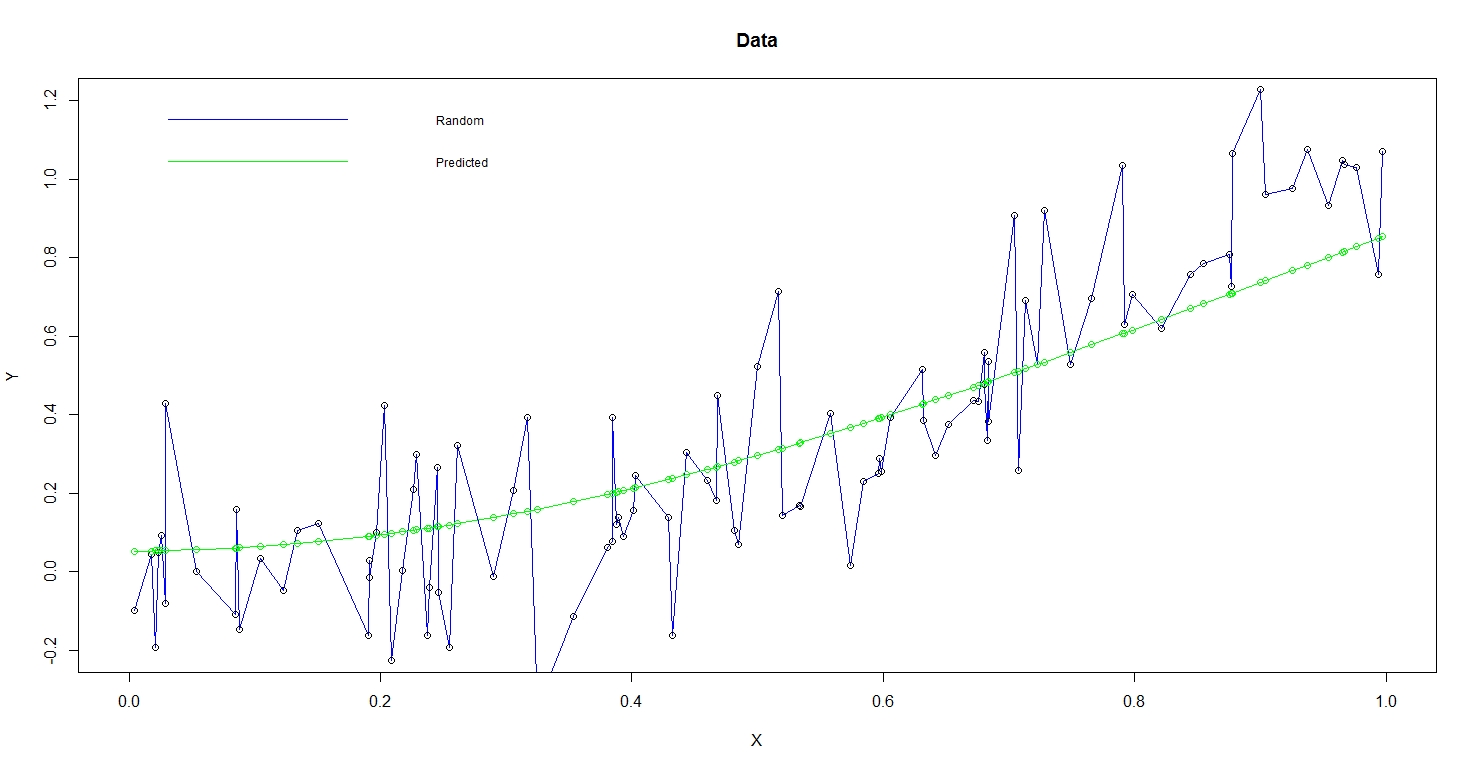
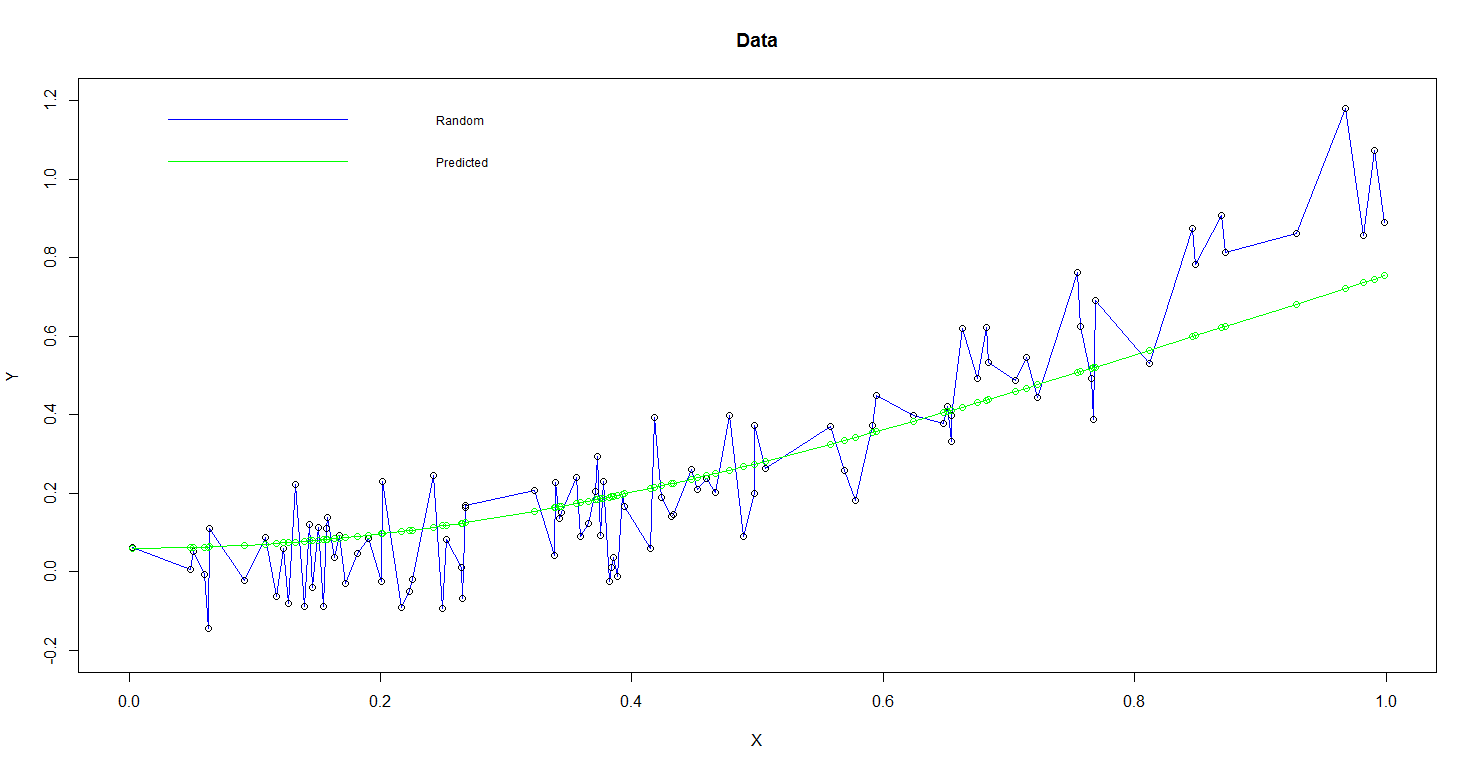


Sigma = 0.1, Lambda = 1

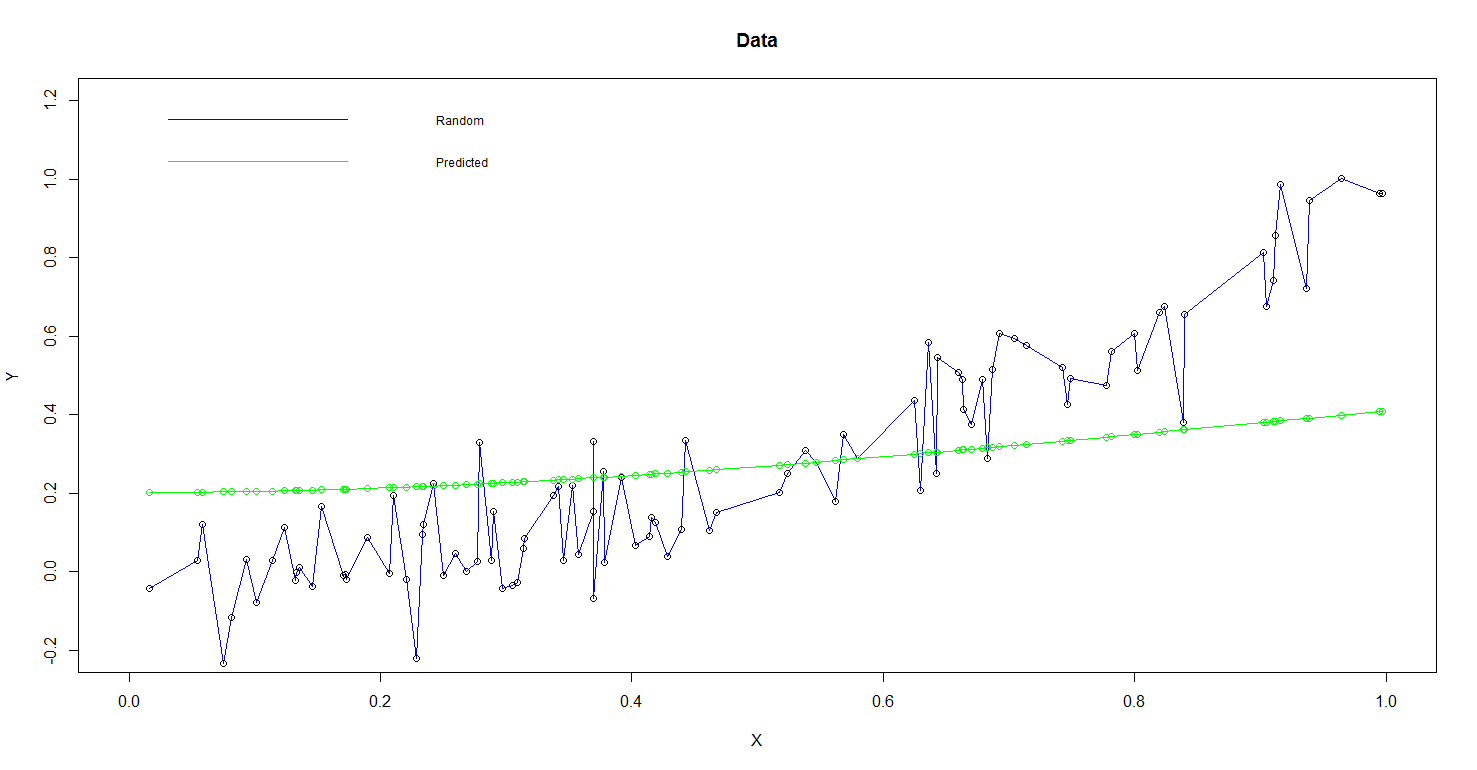


Sigma = 0.2, Lambda = 1

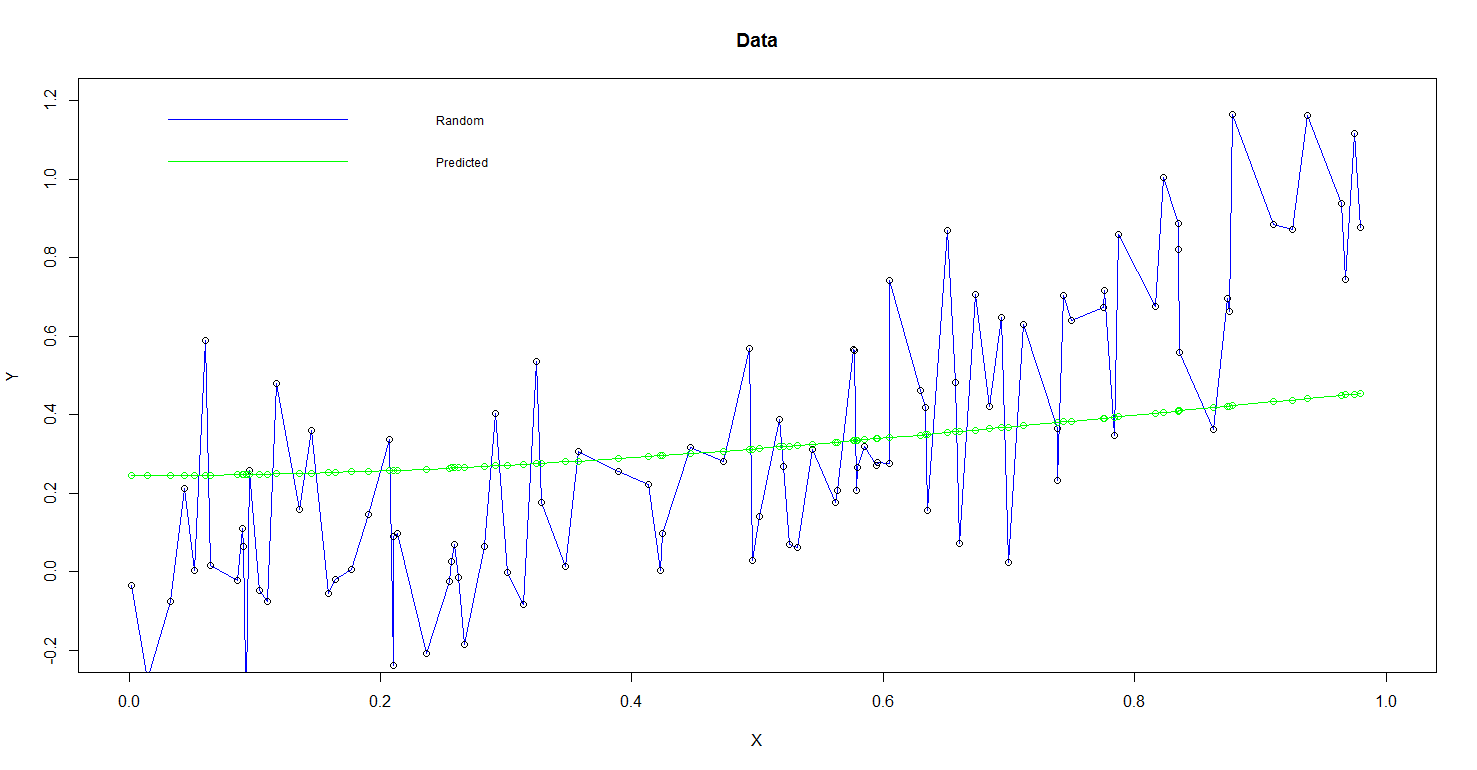
Sigma = 0.1, Lambda = 100



Sigma = 0.2, Lambda = 100



Sigma = 0.1, Lambda = 1000



Sigma = 0.2, Lambda = 1000

**Divide data into 80% training data and 20% testing data as:**

X <- runif(n)

train\_x <- X[1:80]  
test\_x <- X[81:100]

train\_x <- sort (train\_x)  
test\_x <- sort (test\_x)

train\_y <- train\_x^2 + rnorm(80) \* sigma  
test\_y <- test\_x^2 + rnorm(20) \* sigma

**Then, the beta\_spline can be got from the training data as:**

op\_train <- mySpline(train\_x,train\_y,lambda,p)  
beta\_spl <- op$beta\_spline

**The beta spline is used on the test data as:**

op\_test <- mySplineTest(test\_x,test\_y,lambda,p,beta\_spl)

**Training error is calculated as:**

training\_error = sum(abs(train\_y - op\_train$predicted\_y))/length(train\_y)

**Testing error is calculated as:**

testing\_error = sum(abs(test\_y - op\_test$predicted\_y))/length(test\_y)

**The test data is calculated via a Spline Test function as follows:**

mySplineTest <- function(x, Y, lambda, p = 100, beta\_spline){

n = length(x)

X <- matrix(x, nrow=n)

for ( k in (1:(p-1))/p)

X <- cbind(X,(x>k)\*(x-k))

Yhat <- cbind(rep(1,n),X)%\*% beta\_spline

output <- list(beta\_spline = beta\_spline, predicted\_y = Yhat)

return(output)

}

For lambda = 1

training\_error = 0.08270616

testing\_error = 0.103775

For lambda = 0.1

training\_error = 0.07752532

testing\_error = 0.06756782

For lambda = 0.01

training\_error = 0.07071239

testing\_error = 0.0753857

For lambda = 0.001

training\_error = 0.07105358

testing\_error = 0.07870062

For lambda = 100

training\_error = 0.1061344

testing\_error = 0.1051672

For lambda = 1000

training\_error = 0.2157368

testing\_error = 0.2620837

Here, for lambda, which is on the y-axis:

1 -> 0.001  
2 -> 0.01  
3 -> 0.1  
4 -> 1  
5 -> 100  
6 -> 1000

