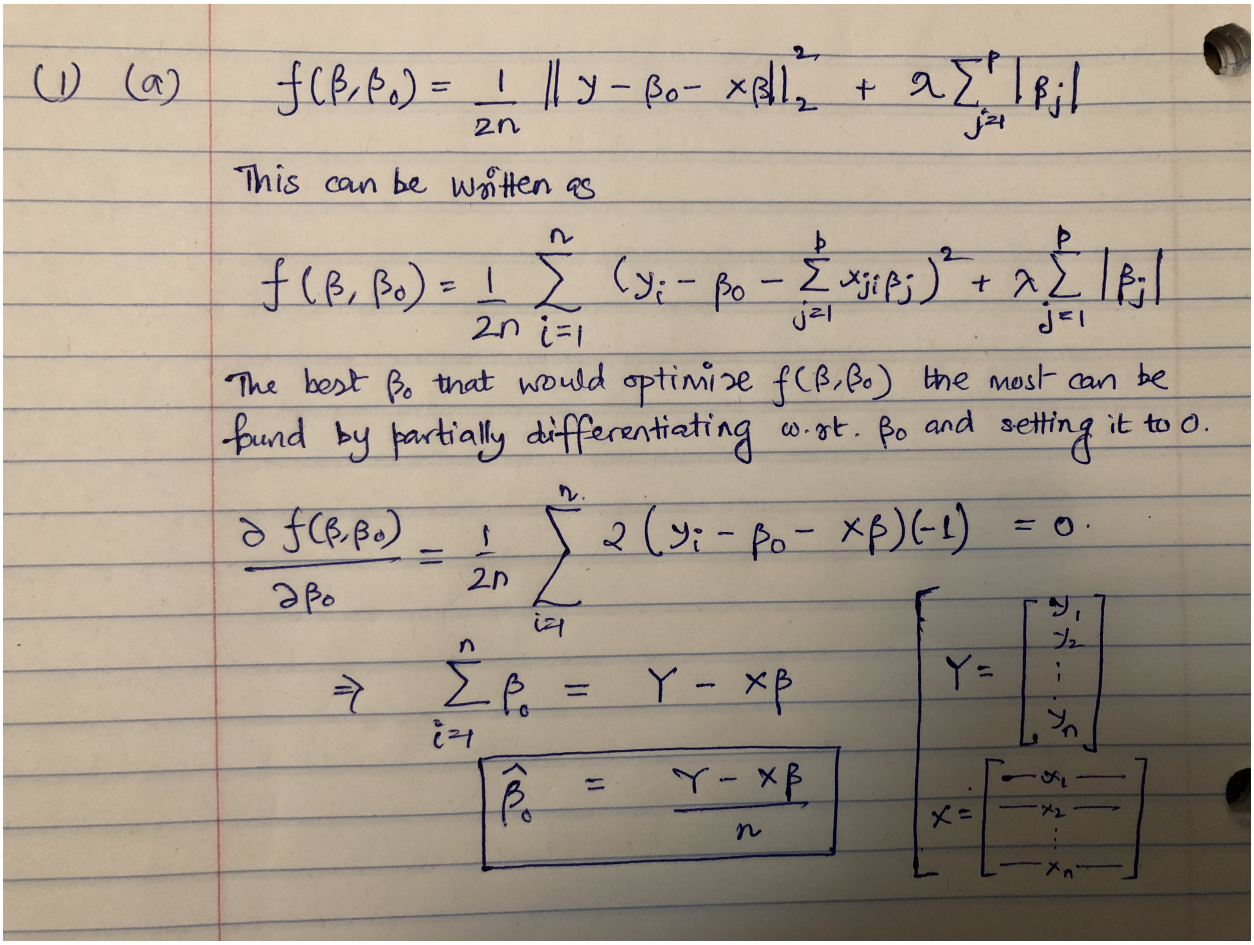
Homework 2

Subhankar Ghosh

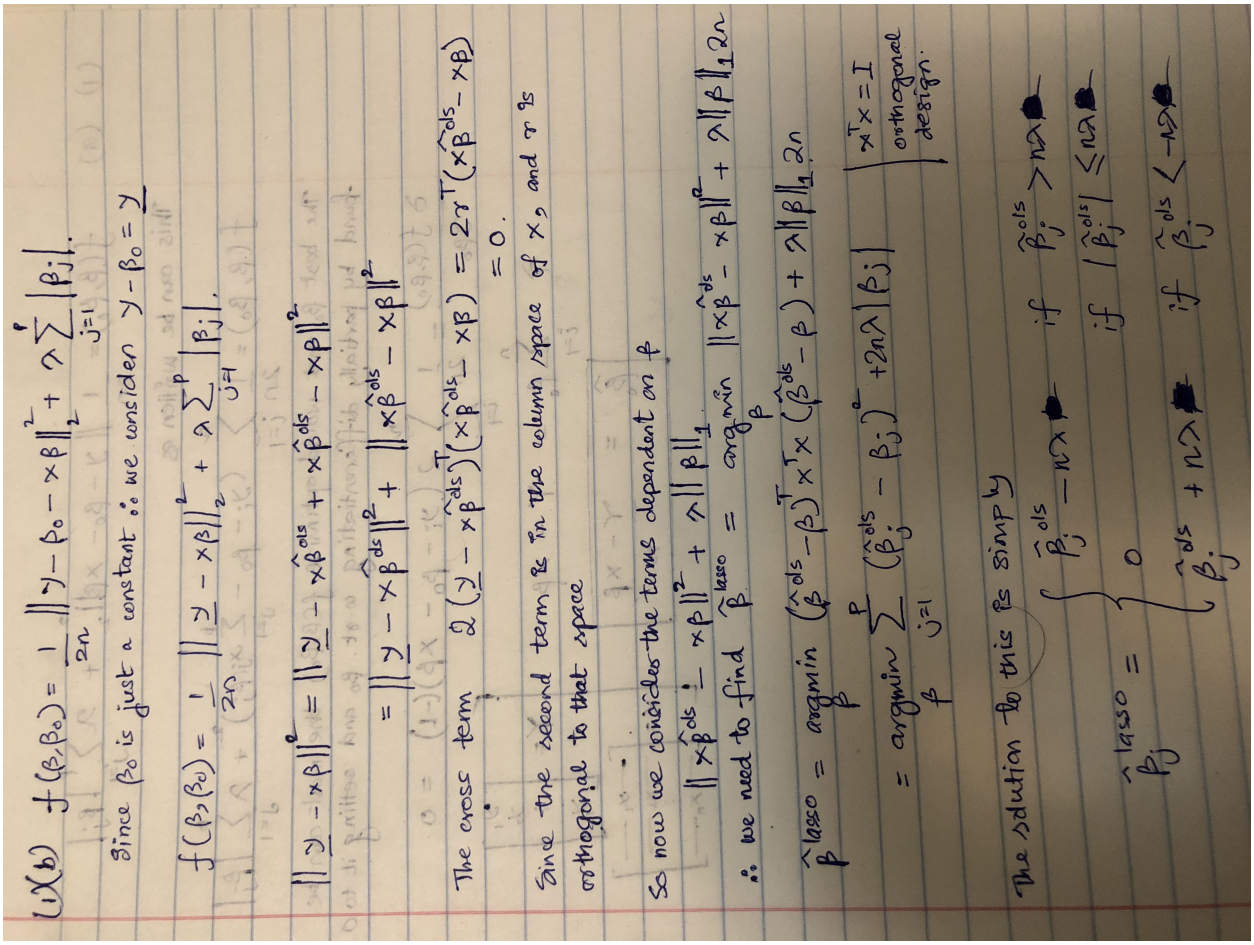
Question 1

(a)



The one-variable optimization problem of Beta_0 based on this objective function.

(b)



The one-variable optimization problem of Beta_j based on this objective function.

For parts (c) and (d) we need to first implement lasso. Following is the implementation in R of lasso

```
# now start to write functions to fit the lasso
# prepare the soft thresholding function for updating beta_j (part b)
soft_th <- function(b, lambda)
{
  if(b < 0 & abs(b) > lambda)
    b = b + lambda
  else if(b > 0 & abs(b) > lambda)
    b = b - lambda
  else
    b = 0
  b
}
# initiate lambda as the lambda_max value in part c)
lambda_max = 20
# produce a sequence of lambda values
lambda = exp(seq(log(lambda_max), log(0.01), length.out = 100))
# if you use this formula, you will need to calculate this for the real data too.
LassoFit <- function(X, y, lambda, tol = 1e-5, maxiter = 100)
{
    # initiate objects to record the values
    mybeta = matrix(NA, ncol(X), length(lambda))
```

```
    mybeta0 = rep(NA, length(lambda))
    mylambda = rep(NA, length(lambda))
    nlambda = length(lambda)
    z = colSums(X^2)

    current_beta = matrix(0, P, 1)
    # current_beta0 = mean(y)

    for (l in 1:nlambda)
    {
        # reduce the current lambda value to a smaller one
        current_lambda = lambda[l]

        for (k in 1:maxiter)
        {
          old_beta = current_beta
            # update the intercept term based on the current beta values.
            current_beta0 = mean(y - X %*% current_beta)
            # start to update each beta_j
            for (j in 1:ncol(X))
            {
                # remove the effect of variable j from model,
              # and compute the residual
              current_beta_copy = current_beta
              current_beta_copy[j, ] = 0
                r = y - current_beta0 - X %*% current_beta_copy

                # update beta_j using the results in part b)
                current_beta[j, ] = soft_th(sum(r * X[, j]), nrow(X)*current_lambda)/z[j]
            }

            # check if beta changed more than the tolerance level
            # in this iteration (use tol as the threshold)
            # if not, break out of this loop k
            if(max(abs(old_beta-current_beta)) < tol)
              break
        }

        mylambda[l] = current_lambda
        mybeta[, l] = current_beta
        mybeta0[l] = current_beta0
    }

    return(list("beta" = mybeta, "b0" = mybeta0, "lambda" = mylambda))
}
```

(c) the smallest λ value such that none of the β_j's, j = 1, . . . , p can be updated out of zero in the next iteration. Denote this value as λ_{max}.

Let us look at a plot of the number of β values equal to 0 vs the λ values.

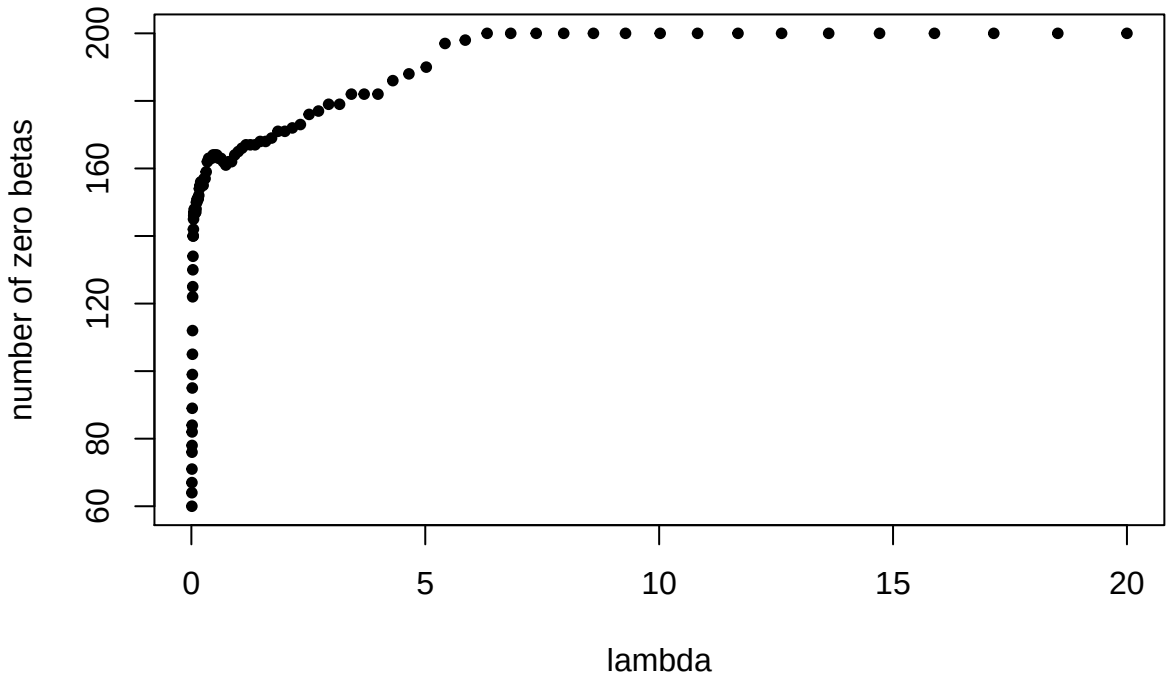
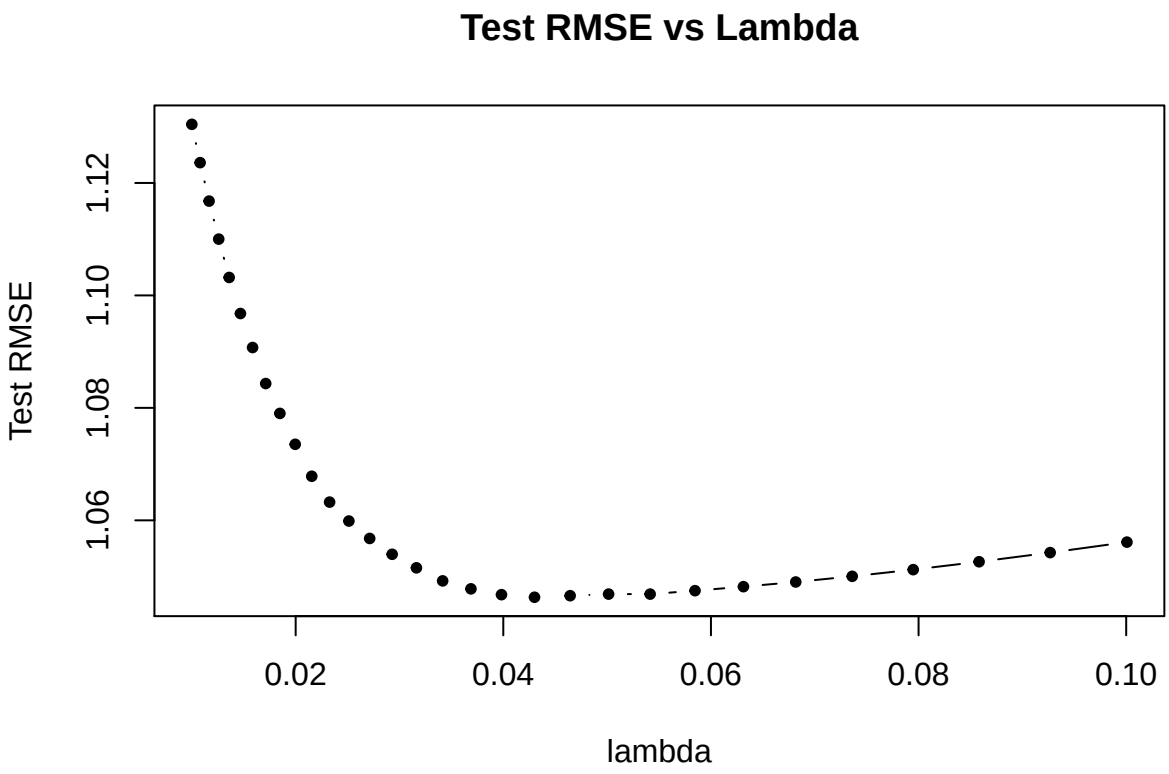


Table 1: Number of 0 valued beta vs lambda

lambda	zeros
10.022	200
9.281	200
8.595	200
7.960	200
7.372	200
6.827	200
6.322	200
5.855	198
5.422	197

From the plot as well as from the data we can see that $\lambda_{max} = 6.322$

(d) For testing I have generated an independent set of 1000 observations under the same model as the test dataset. We will look at the plot of test RMSE vs λ to see how it varies.



λ value that yields the minimum RMSE is

```
## [1] 0.04300619
```

Question 2

(a) After running the lasso implementation in glmnet package and a 10 fold cross-validation, **the best lambda we got was:**

```
## [1] 0.05026652
```

The number of nonzero parameters using this lambda:

```
## [1] 55
```

(b) The estimated degree of freedom of the lasso fit given by $\sum_{i=1}^{n} Cov(\hat{y}_i, y_i)/\sigma^2$ is the following:

```
## [1] 53.077
```

```
## [1] "The number of non-zero parameters from part (a) we got 55"
```

```
## [1] "The estimated degree of freedom we get is equal to 53.077"
```

I think both are quite close and the estimated value agrees pretty well with the theoretical value.

(c) We repeat the process of part (a) and part (b) here with ridge regression.

The best λ value we get for ridge regression is:

```
## [1] 0.6342915
```

The theoretical value of the degree of freedom of ridge regression is calculated by the formula:

$$df(\lambda) = Trace(X(X^T X + \lambda I)^{-1} X^T)$$

Using this formula we get the value:

```
## [1] 199.1702
```

The estimated value of the degree of freedom of the ridge regression is equal to:

```
## [1] 161.426
```

The estimation for lasso was closer as compared to ridge regression.