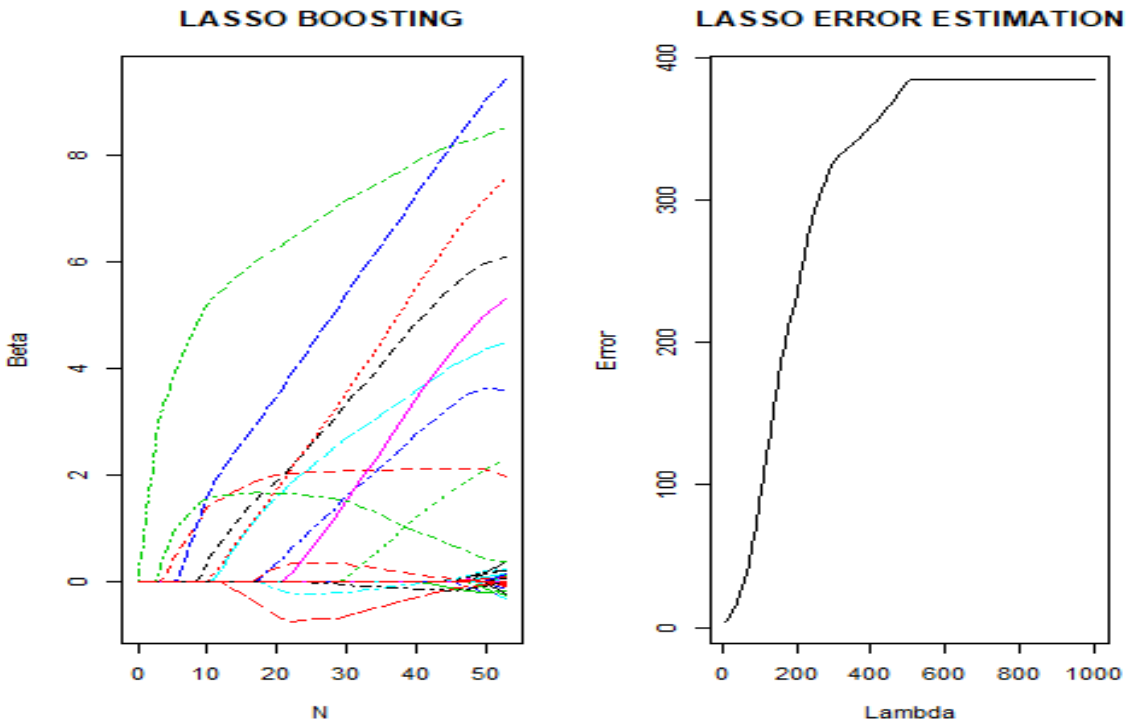
Q2: Plot of the estimation error over the different values of lambda (Lasso).



Q3: Analysis of package functions over real datasets available in R

PCA

Dataset used: iris

It gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

```
> iris = datasets::iris
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa

> a = as.matrix(iris[, 1:4])
> p = PCA(a)
> p
$D
[1] 9208.30507  315.45432   11.97804    3.55257

$V
           [,1]       [,2]        [,3]       [,4]
[1,] -0.7511082 -0.2841749  0.50215472  0.3208143
[2,] -0.3800862 -0.5467445 -0.67524332 -0.3172561
[3,] -0.5130089  0.7086646 -0.05916621 -0.4807451
[4,] -0.1679075  0.3436708 -0.53701625  0.7518717
```

```
> e = eigen(t(a)%*%a)
> e
eigen() decomposition
$values
[1] 9208.30507  315.45432   11.97804    3.55257

$vectors
           [,1]       [,2]        [,3]       [,4]
[1,] -0.7511082  0.2841749 -0.50215472  0.3208143
[2,] -0.3800862  0.5467445  0.67524332 -0.3172561
[3,] -0.5130089 -0.7086646  0.05916621 -0.4807451
[4,] -0.1679075 -0.3436708  0.53701625  0.7518717
```

Logistic Regression

Dataset used: binary.csv

It calculates if it's an admit or not using gre, gpa and rank data.

```
> mydata <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")

> head(mydata)
  admit gre  gpa rank
1     0 380 3.61    3
2     1 660 3.67    3
3     1 800 4.00    1
4     1 640 3.19    4
5     0 520 2.93    4
6     1 760 3.00    2
> x = as.matrix(mydata[,2:4])
> y = as.matrix(mydata[,1])
> x[,1] = (x[,1] - mean(x[,1]))/sd(x[,1])
> x[,2] = (x[,2] - mean(x[,2]))/sd(x[,2])
> x[,3] = (x[,3] - mean(x[,3]))/sd(x[,3])
> LogisticRegression(x, y)

$coefficients
       gre        gpa       rank
 0.2233584  0.2510192 -0.4472078

$standard_error
      gre       gpa      rank
0.1147555 0.1140612 0.1082179

> print(glm(formula = y ~ x + 0,  family="binomial"))

Call:  glm(formula = y ~ x + 0, family = "binomial")

Coefficients:
   xgre     xgpa    xrank
 0.2217   0.2500  -0.4453

Degrees of Freedom: 400 Total (i.e. Null);  397 Residual
Null Deviance:       554.5
Residual Deviance: 519.9  AIC: 525.9
```

Linear Regression

Dataset used: swiss

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

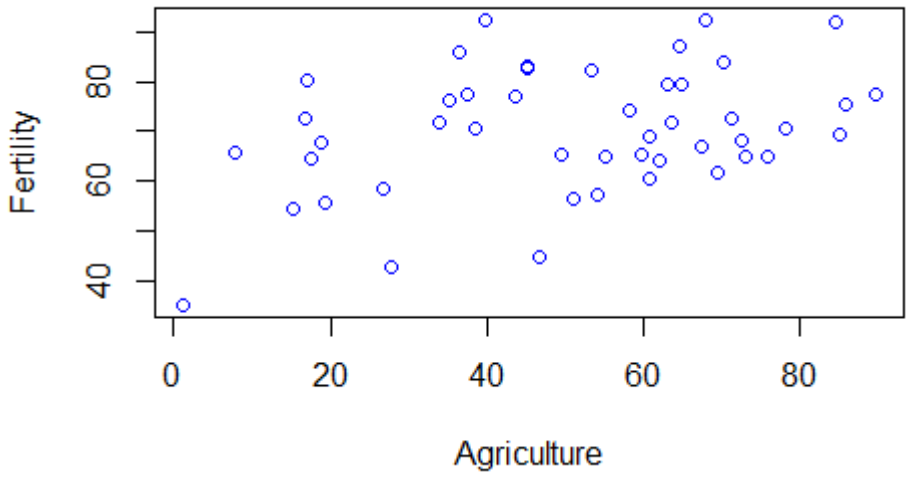
```
> swiss = datasets::swiss
> head(swiss)
             Fertility Agriculture Examination Education Catholic Infant.Mortality
Courtelary        80.2        17.0          15        12     9.96             22.2
Delemont          83.1        45.1           6         9    84.84             22.2
Franches-Mnt      92.5        39.7           5         5    93.40             20.2
Moutier           85.8        36.5          12         7    33.77             20.3
Neuveville        76.9        43.5          17        15     5.16             20.6
Porrentruy        76.1        35.3           9         7    90.57             26.6
```

```
> x = as.matrix(swiss[, 2:6])
> y = as.matrix(swiss[, 1])
> LinearRegression(x, y)
$coefficients
                    Agriculture      Examination        Education         Catholic Infa
nt.Mortality
      66.9151817       -0.1721140       -0.2580082       -0.8709401        0.1041153
1.0770481

$standard_error
                    Agriculture      Examination        Education         Catholic Infa
nt.Mortality
     10.70603759        0.07030392       0.25387820       0.18302860       0.03525785
0.38171965
```

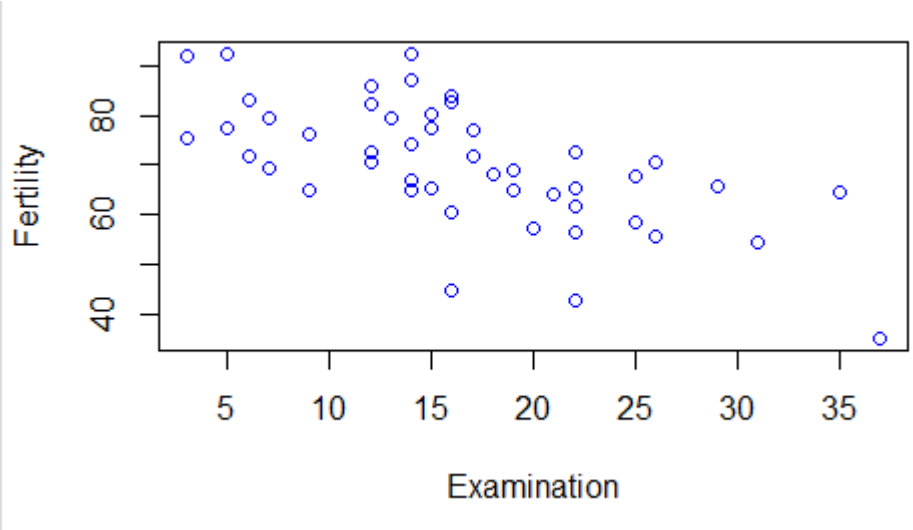
```
> coef(lm(y ~ x))
      (Intercept)      xAgriculture      xExamination        xEducation         xCatholic
xInfant.Mortality
       66.9151817        -0.1721140        -0.2580082        -0.8709401         0.1041153
1.0770481
```

```
> plot(x[, 1], y, xlab='Agriculture', ylab='Fertility', col='blue')
```



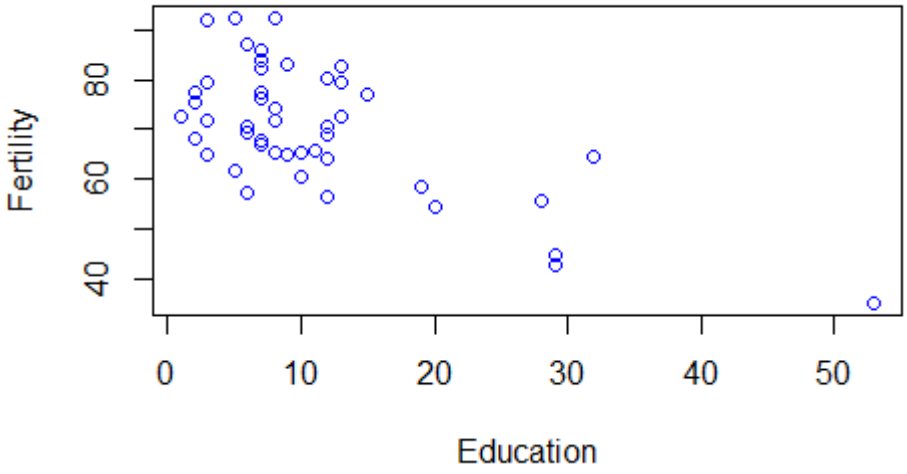
We can observe a positive correlation between Agriculture and Fertility

```
> plot(x[, 2], y, xlab='Examination', ylab='Fertility', col='blue')
```

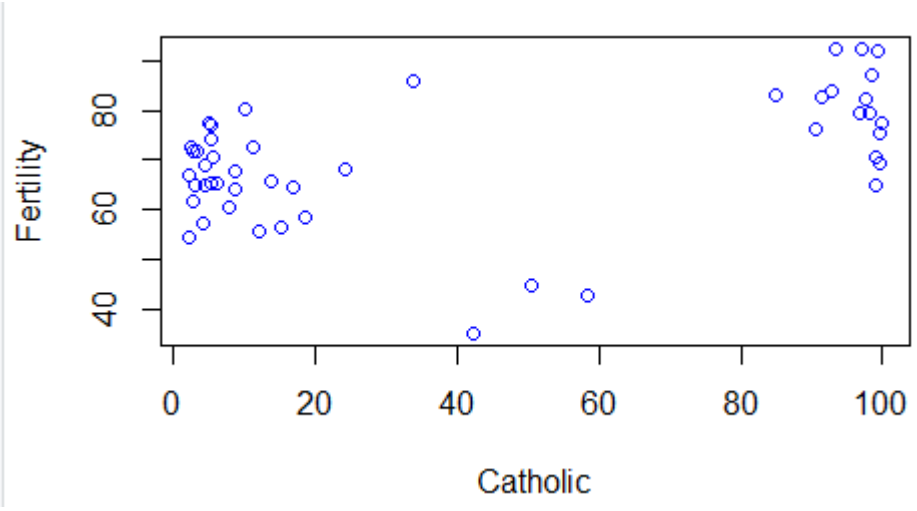


We can observe a negative correlation between Examination and Fertility

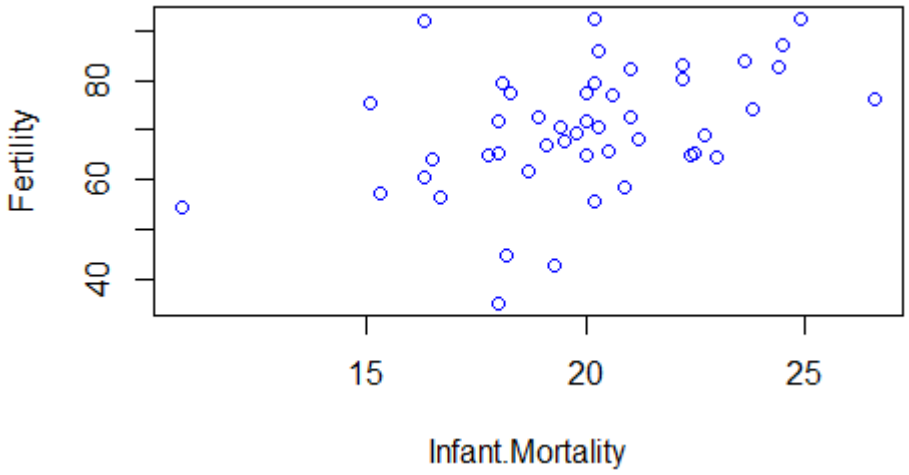
```
> plot(x[, 3], y, xlab='Agriculture', ylab='Fertility', col='blue')
```




```
> plot(x[, 4], y, xlab='Catholic', ylab='Fertility', col='blue')
```




```
> plot(x[, 5], y, xlab='Infant.Mortality', ylab='Fertility', col='blue')
```



Ridge Regression

Dataset used: swiss

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

```
> swiss <- datasets::swiss
> head(swiss)
             Fertility Agriculture Examination Education Catholic Infant.Mortality
Courtelary        80.2        17.0          15        12     9.96             22.2
Delemont          83.1        45.1           6         9    84.84             22.2
Franches-Mnt      92.5        39.7           5         5    93.40             20.2
Moutier           85.8        36.5          12         7    33.77             20.3
Neuveville        76.9        43.5          17        15     5.16             20.6
Porrentruy        76.1        35.3           9         7    90.57             26.6

> x = model.matrix(Fertility~., swiss)[,-1]
> y = swiss$Fertility
> lambda = 10^seq(10, -2, length = 100)
> library(glmnet)
> set.seed(489)
```

```
> train = sample(1:nrow(x), nrow(x)/2)
> test = (-train)
> ytest = y[test]
> swisslm = lm(Fertility~., data = swiss)
> coef(swisslm)
     (Intercept)      Agriculture      Examination        Education         Cathol
ic
      66.9151817       -0.1721140       -0.2580082       -0.8709401        0.10411
53
Infant.Mortality
       1.0770481
> lambda = 0.1
> ridge_R = glmnet(x[train,], y[train], alpha = 0, lambda= lambda)
> ridge_P = myRidge(x[train,],y[train],lambda)
> source('C:/Users/shraddha_m26/Desktop/Stats Programming/Assignments/6/Ridge_Spli
ne.R')
> ridge_P = myRidge(x[train,],y[train],lambda)
> ridge_R

Call:  glmnet(x = x[train, ], y = y[train], alpha = 0, lambda = lambda)

     Df   %Dev Lambda
[1,]  5 0.8002    0.1
> ridge_P
                     Agriculture      Examination        Education         Cathol
ic
     74.64436146      -0.27807670      -0.93900466      -0.35978119       0.065001
47
Infant.Mortality
      1.37552338
> coef(ridge_R)
6 x 1 sparse Matrix of class "dgCMatrix"
                           s0
(Intercept)      73.36350615
Agriculture      -0.26542433
Examination      -0.89519263
Education        -0.36435849
Catholic          0.06570399
Infant.Mortality  1.37394755
```

Observation: Ridge regression performs better than Linear Regression because of the regularization.

Lasso

Dataset used: swiss

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

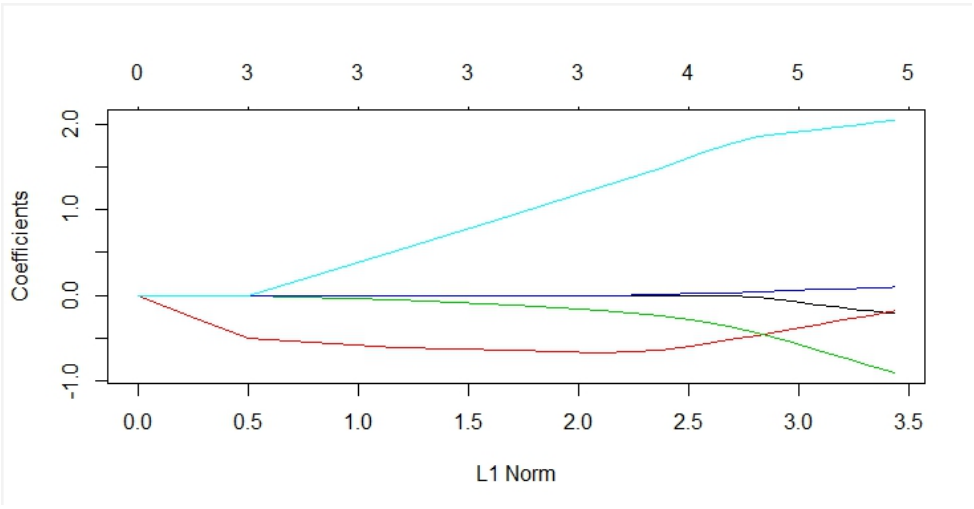
```
> swiss <- datasets::swiss
> x <- model.matrix(Fertility~., swiss)[,-1]
> y <- swiss$Fertility
> lambda <- 10^seq(10, -2, length = 100)
> library(Stats202A)
> cv.out <- cv.glmnet(x[train,], y[train], alpha = 0)
> bestlam <- cv.out$lambda.min
> lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = lambda)
```

```
> lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[test,])
> mean((lasso.pred-ytest)^2)
[1] 113.7041

> lasso.coef <- predict(lasso.mod, type = 'coefficients', s = bestlam)[1:6,]
> lasso.coe
      (Intercept)       Agriculture       Examination
      57.85476722       -0.06225277       -0.50145205

        Education          Catholic Infant.Mortality
      -0.12425311        0.04456320        1.25231604

> res <- Lasso(x[train,], y[train], lambda)
> plot(lasso.mod)
```




```
> matplot(t(matrix(rep(1, p), nrow = 1)%*%abs(beta_all)), t(beta_all), type =
'l',main='LASSO BOOSTING',xlab='N',ylab='Beta')
```