# Loading the library

library(glmnet)

# Loading the data

data(swiss)

x\_vars <- model.matrix(Fertility~. , swiss)[,-1]

y\_var <- swiss$Fertility

lambda\_seq <- 10^seq(2, -2, by = -.1)

# Splitting the data into test and train

set.seed(86)

train = sample(1:nrow(x\_var), nrow(x\_var)/2)

x\_test = (-train)

y\_test = y\_var[test]

cv\_output <- cv.glmnet(x\_vars[train,], y\_var[train],

alpha = 1, lambda = lambda\_seq)

# identifying best lamda

best\_lam <- cv\_output$lambda.min

Using this value, let us train the lasso model again.

# Rebuilding the model with best lamda value identified

lasso\_best <- glmnet(x\_vars[train,], y\_var[train], alpha = 1, lambda = best\_lam)

pred <- predict(lasso\_best, s = best\_lam, newx = x\_vars[test,])

Finally, we combine the predicted values and actual values to see the two values side by side and then you can use the R-Squared formula to check the model performance. Note - you must calculate the R-Squared values for both train and test dataset.

final <- cbind(y\_var[test], pred)

# Checking the first six obs

head(final)

Sharing the R Squared formula

The function provided below is just indicative and you must provide the actual and predicted values based upon your dataset.

actual <- test$actual

preds <- test$predicted

rss <- sum((preds - actual) ^ 2)

tss <- sum((actual - mean(actual)) ^ 2)

rsq <- 1 - rss/tss

rsq

Getting the list of important variables

To get the list of important variables we just need to investigate the beta coefficients of final best model.

# Inspecting beta coefficients

coef(lasso\_best)

# Output

6 x 1 sparse Matrix of class "dgCMatrix"

s0

(Intercept) 55.16706057

Agriculture .

Examination -0.30124968

Education .

Catholic 0.04700893

Infant.Mortality 0.84730322

The model indicates that the coefficients of Agriculture and Education have been shrinked to zero. Thus we are left with three variables namely; Examination, Catholic, and Infant.Mortality