Computer lab 1 block 1 - Assignment 3

Tim Yuki Washio

11/10/2020

Assignment 3 - Linear regression and LASSO

```
# Load data
data = read.csv("data/tecator.csv")

# Split data into train and test set
n = dim(data)[1]
set.seed(12345)
id = sample(1:n, floor(n*0.5))
train = data[id,]
test = data[-id,]
```

1.

Assume that Fat can be modeled as a linear regression in which absorbance characteristics (Channels) are used as features. Report the underlying probabilistic model, fit the linear regression to the training data and estimate the training and test errors. Comment on the quality of fit and prediction and therefore on the quality of model.

Probabilistic model:

$$\hat{y} = \beta_0 + \sum_{i=1}^{100} \beta_i * x_i + \epsilon \sim \mathsf{N}(\mu, \sigma^2)$$

```
##
## Call:
## lm(formula = fmla, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.201500 -0.041315 -0.001041  0.037636  0.187860
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.815e+01  5.488e+00  -3.306  0.01628 *
## Channel1     2.653e+04  1.126e+04   2.357  0.05649 .
## Channel2    -5.871e+04  3.493e+04  -1.681  0.14385
## Channel3     1.154e+05  7.373e+04   1.565  0.16852
## Channel4    -2.432e+05  1.175e+05  -2.070  0.08387 .
## Channel5     3.026e+05  1.193e+05   2.536  0.04430 *
## Channel6    -2.365e+05  8.160e+04  -2.898  0.02741 *
## Channel7     1.090e+05  3.169e+04   3.440  0.01380 *
## Channel8    -6.054e+04  1.508e+04  -4.015  0.00700 **
## Channel9     7.871e+04  2.160e+04   3.643  0.01079 *
## Channel10   -1.730e+04  1.640e+04  -1.055  0.33215
## Channel11    9.562e+04  3.529e+04   2.710  0.03512 *
## Channel12   -2.114e+05  6.198e+04  -3.410  0.01431 *
## Channel13    9.725e+04  4.424e+04   2.198  0.07026 .
## Channel14    5.296e+04  4.666e+04   1.135  0.29968
## Channel15   -7.855e+04  5.245e+04  -1.498  0.18491
## Channel16   -8.209e+03  1.893e+04  -0.434  0.67969
## Channel17    3.769e+04  1.987e+04   1.897  0.10666
## Channel18    3.306e+04  7.934e+03   4.167  0.00590 **
## Channel19   -8.405e+04  1.929e+04  -4.358  0.00478 **
## Channel20    1.510e+05  3.361e+04   4.492  0.00414 **
## Channel21   -2.069e+05  4.256e+04  -4.862  0.00282 **
## Channel22    1.348e+05  3.824e+04   3.526  0.01243 *
## Channel23   -4.094e+04  3.546e+04  -1.154  0.29222
## Channel24    2.023e+04  2.761e+04   0.733  0.49134
## Channel25    3.269e+03  1.071e+04   0.305  0.77045
## Channel26   -1.297e+04  7.636e+03  -1.699  0.14028
## Channel27    4.131e+03  1.422e+04   0.291  0.78120
## Channel28   -4.548e+03  2.988e+04  -0.152  0.88402
## Channel29    1.089e+04  1.768e+04   0.616  0.56072
## Channel30   -7.985e+04  2.653e+04  -3.010  0.02371 *
## Channel31    1.756e+05  5.279e+04   3.326  0.01589 *
## Channel32   -1.107e+05  2.904e+04  -3.813  0.00883 **
## Channel33   -6.525e+04  5.407e+04  -1.207  0.27294
## Channel34    1.007e+05  6.589e+04   1.528  0.17738
## Channel35   -2.841e+03  1.214e+04  -0.234  0.82266
## Channel36   -2.268e+04  2.295e+04  -0.988  0.36127
## Channel37   -4.479e+04  1.292e+04  -3.468  0.01334 *
## Channel38    3.209e+04  1.843e+04   1.742  0.13221
## Channel39    1.992e+04  2.067e+04   0.964  0.37246
## Channel40   -9.833e+03  2.431e+04  -0.404  0.69988
## Channel41    1.659e+04  3.648e+04   0.455  0.66531
## Channel42   -1.829e+04  3.528e+04  -0.519  0.62260
## Channel43   -2.423e+04  2.427e+04  -0.998  0.35669
## Channel44    3.246e+04  2.013e+04   1.613  0.15793
## Channel45   -8.089e+03  4.023e+04  -0.201  0.84728
## Channel46    7.065e+03  2.810e+04   0.251  0.80990
## Channel47   -4.062e+04  1.007e+04  -4.034  0.00685 **
## Channel48    9.080e+04  2.618e+04   3.469  0.01332 *
## Channel49   -6.647e+04  2.372e+04  -2.803  0.03105 *
## Channel50   -4.196e+04  2.856e+04  -1.469  0.19213
## Channel51    1.097e+05  5.572e+04   1.968  0.09661 .
## Channel52   -1.148e+05  6.376e+04  -1.800  0.12196
## Channel53    9.525e+04  7.450e+04   1.278  0.24830
## Channel54   -4.534e+04  7.363e+04  -0.616  0.56067
## Channel55   -1.535e+03  4.933e+04  -0.031  0.97618
## Channel56   -2.377e+03  2.109e+04  -0.113  0.91394
## Channel57    3.174e+04  1.005e+04   3.158  0.01961 *
## Channel58    2.221e+03  1.048e+04   0.212  0.83915
## Channel59   -8.504e+04  2.574e+04  -3.304  0.01634 *
## Channel60    6.382e+04  1.607e+04   3.972  0.00735 **
## Channel61    2.151e+04  1.234e+04   1.742  0.13211
## Channel62   -2.859e+04  1.065e+04  -2.685  0.03631 *
## Channel63    1.796e+04  9.187e+03   1.955  0.09838 .
## Channel64    5.759e+04  3.526e+04   1.633  0.15354
## Channel65   -1.470e+05  6.911e+04  -2.127  0.07752 .
## Channel66    9.121e+04  4.461e+04   2.045  0.08688 .
## Channel67   -5.733e+03  2.197e+04  -0.261  0.80288
## Channel68   -6.290e+04  2.192e+04  -2.870  0.02843 *
## Channel69    6.421e+04  2.074e+04   3.096  0.02121 *
## Channel70   -1.749e+04  1.581e+04  -1.106  0.31111
## Channel71   -7.248e+03  1.934e+04  -0.375  0.72075
## Channel72    3.406e+04  1.185e+04   2.873  0.02830 *
## Channel73   -2.100e+04  1.132e+04  -1.855  0.11308
## Channel74   -3.314e+04  1.220e+04  -2.717  0.03480 *
## Channel75    7.039e+04  2.054e+04   3.427  0.01402 *
## Channel76   -3.187e+04  1.736e+04  -1.836  0.11597
## Channel77    2.061e+04  1.810e+04   1.138  0.29832
## Channel78   -1.180e+04  2.273e+04  -0.519  0.62225
## Channel79    2.669e+04  2.997e+04   0.890  0.40750
## Channel80   -6.051e+04  1.483e+04  -4.080  0.00650 **
## Channel81    1.386e+03  2.628e+04   0.053  0.95966
## Channel82    1.020e+05  4.694e+04   2.173  0.07275 .
## Channel83   -1.706e+05  4.688e+04  -3.640  0.01083 *
## Channel84    1.097e+05  2.892e+04   3.792  0.00905 **
## Channel85   -1.294e+05  3.600e+04  -3.594  0.01145 *
## Channel86    2.130e+05  4.345e+04   4.903  0.00270 **
## Channel87   -1.198e+05  3.818e+04  -3.139  0.02011 *
## Channel88   -2.199e+04  6.085e+04  -0.361  0.73021
## Channel89    7.974e+04  5.077e+04   1.571  0.16733
## Channel90   -1.711e+05  5.499e+04  -3.112  0.02079 *
## Channel91    2.107e+05  6.406e+04   3.289  0.01663 *
## Channel92   -1.959e+05  7.171e+04  -2.733  0.03407 *
## Channel93    2.874e+05  9.937e+04   2.892  0.02762 *
## Channel94   -3.064e+05  9.601e+04  -3.191  0.01881 *
## Channel95    2.048e+05  6.220e+04   3.292  0.01656 *
## Channel96   -5.600e+04  2.929e+04  -1.912  0.10441
## Channel97   -1.318e+04  3.050e+04  -0.432  0.68065
## Channel98   -2.724e+04  2.107e+04  -1.292  0.24375
## Channel99    3.556e+04  1.382e+04   2.573  0.04218 *
## Channel100  -1.206e+04  4.264e+03  -2.828  0.03006 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3191 on 6 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:  0.9994
## F-statistic:  1651 on 100 and 6 DF,  p-value: 1.058e-09
```

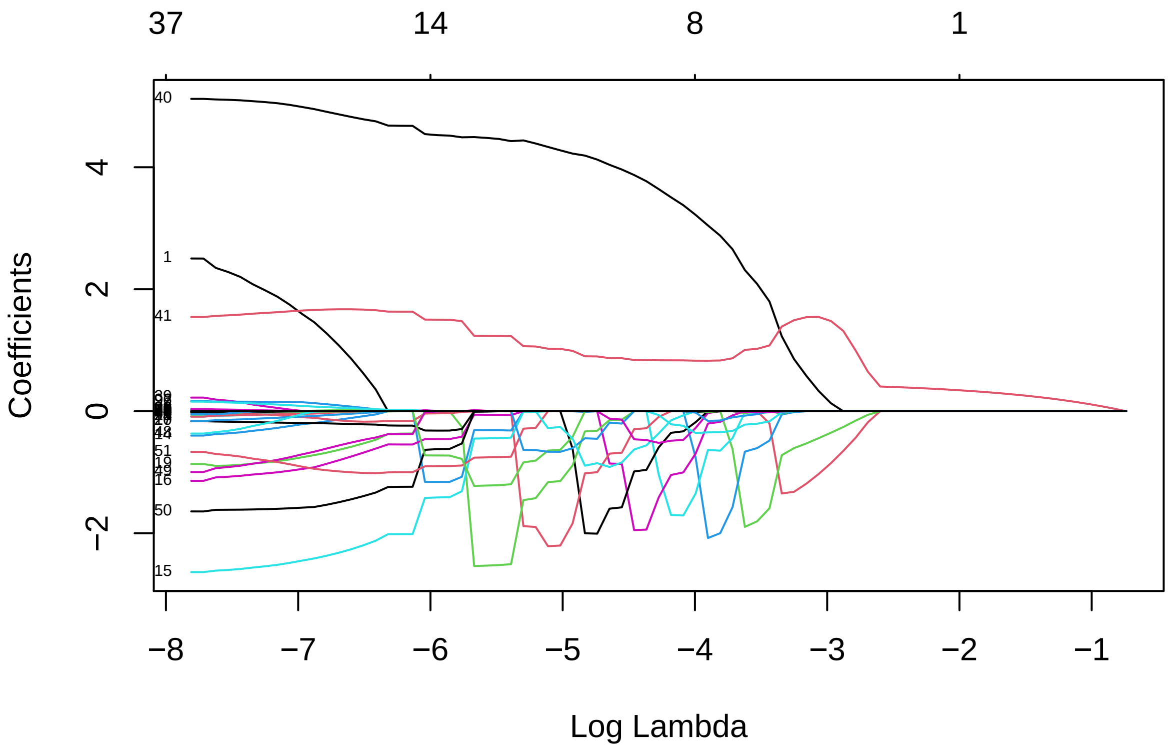
```
## Train error
## MSE:  0.005709117
## RMSE:  0.0755587
##
## Test error
## MSE:  722.4294
## RMSE:  26.87805
```

Our loss-function (MSE) is showing big differences between the prediction of train and test set values. The MSE for the train set is very low while the MSE for the test set returns a very high value. Therefore, our model is overfitting on the training data.

2.

$$\underset{\beta}{\text{argmin}}\left[\sum_{i=1}^{n}\left(Y_i-\beta_0-\sum_{j=1}^{p}\beta_j X_{ji}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|\right]$$

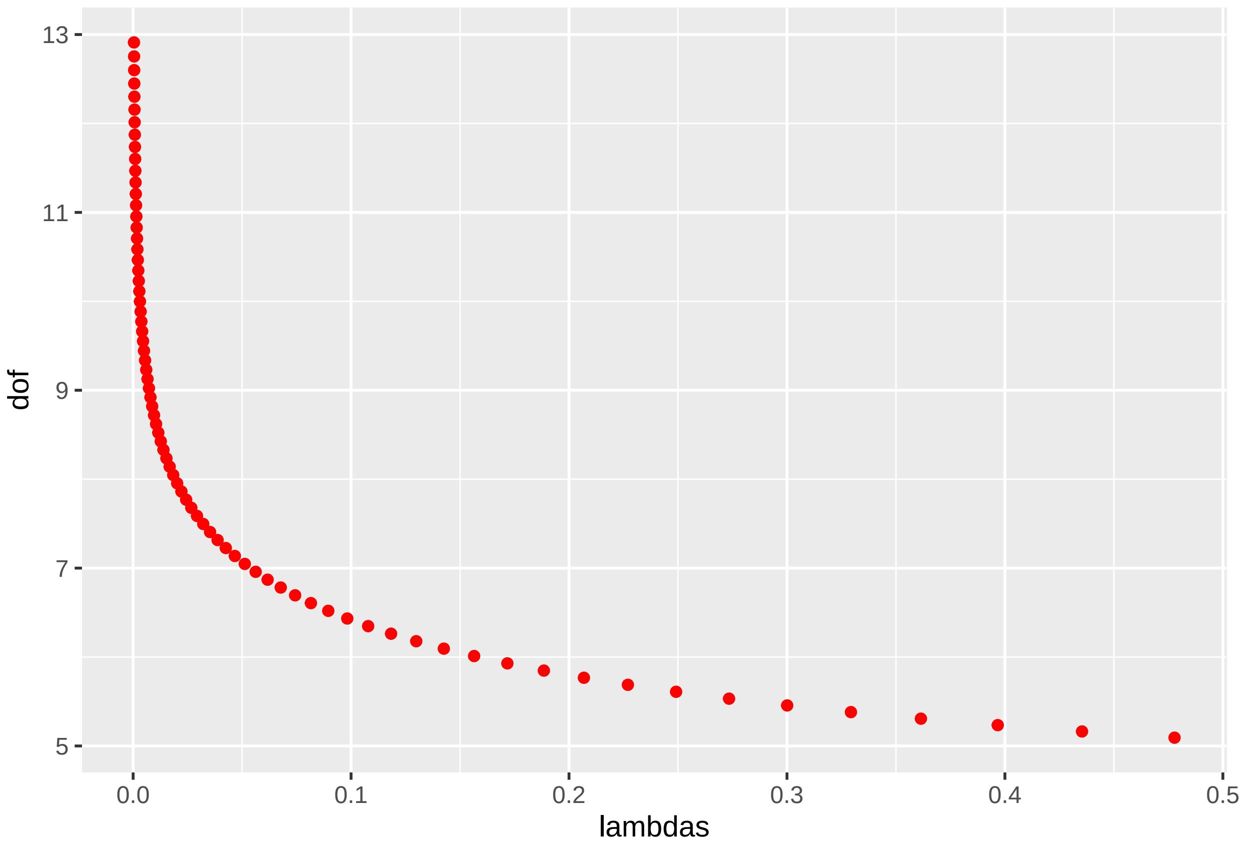
3.



This plot shows the relationship between $log(\lambda)$ and value of coefficients in the *LASSO* model. We can see that a bigger $log(\lambda)$ evaluates to less non-zero coefficients. At $log(\lambda) = -8$ most of the coefficients are non-zero. Choosing $\lambda = 0$ returns the same results as using OLS for the model coefficients since no penalty is given. For each coefficient we can also see the influence (positive or negative) on our prediction depending on different $log(\lambda)$ values.

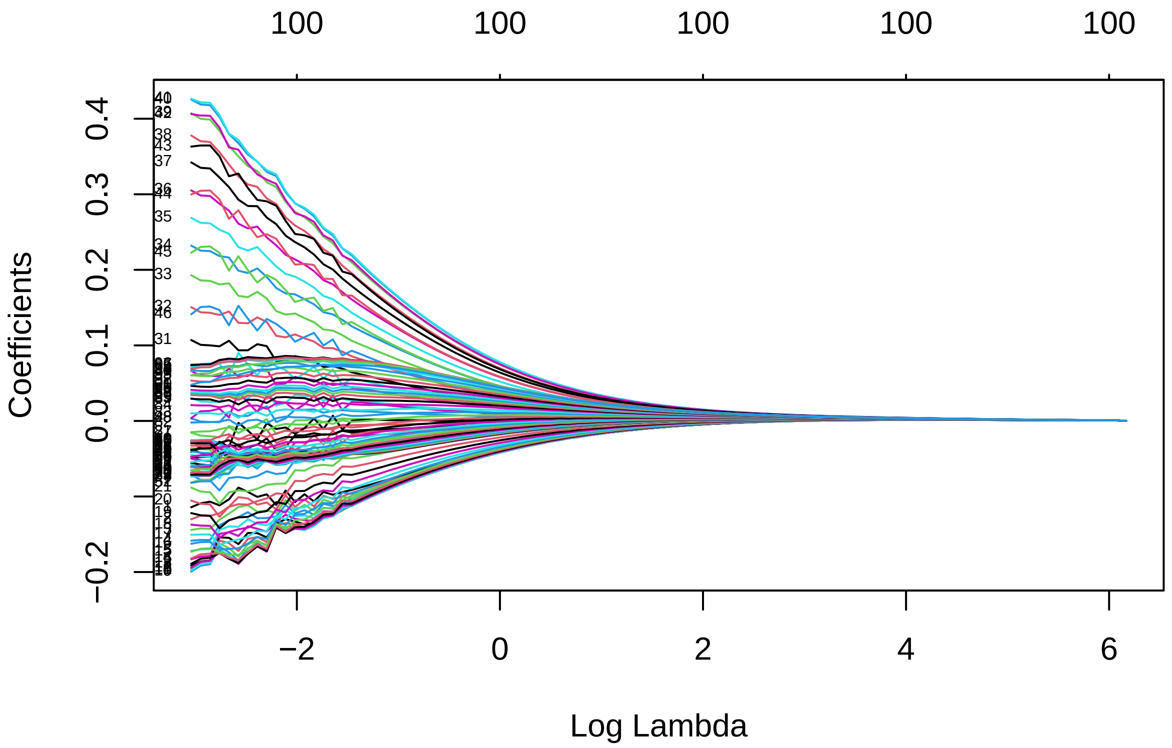
To find the λ value where our model contains only 3 non-zero coefficients we check the plot again. We see that the coefficient with the label '40' goes to 0 at around $log(\lambda) = -2.8$. We calculate $\lambda = e^{-2.8} \approx 0.06$ and get the approximate λ where we have 3 non-zero coefficients left.

4.



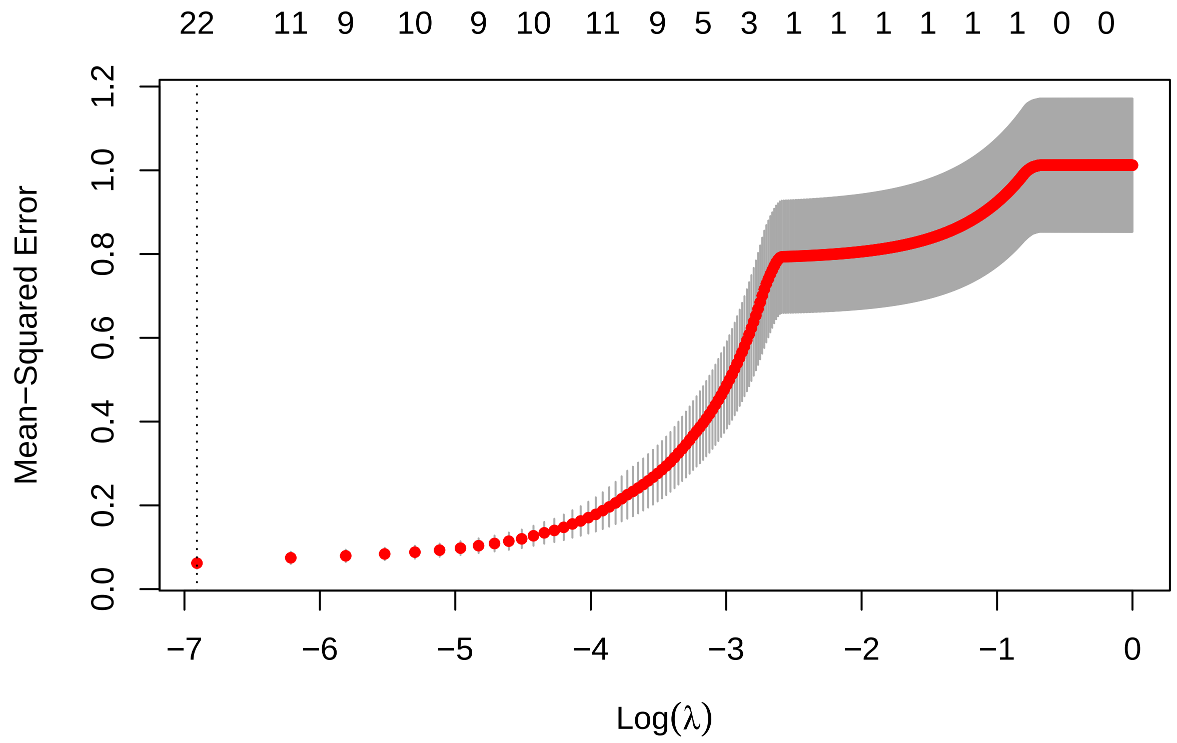
The trend is as expected. If lambda is large, the parameters are heavily constrained and the degrees of freedom will effectively be lower, tending to 0 as λ -> ∞. For λ -> 0, we have p parameters. As in OLS all coefficients stay the same, no penalization to be found.

5.



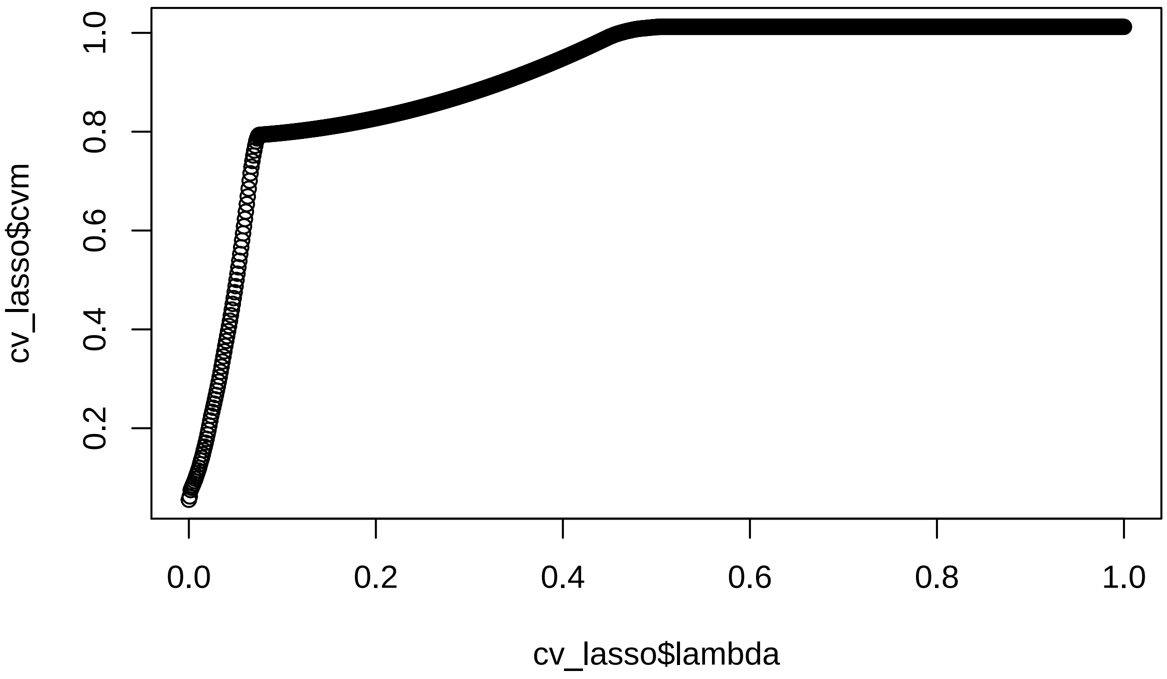
Both Ridge and LASSO regression are shrinkage methods that try to reduce the complexity of a model by shrinking or eliminating (LASSO only) its coefficients. For the Ridge regression the number of coefficients stays the same regardless of how big $log(\lambda)$ gets. For LASSO regression the number of coefficients decreases with increasing value of $log(\lambda)$. Looking at the LASSO plot it is easier to determine which coefficients are the most significant compared to looking at the Ridge plot.

6.



We can see a correlation between increasing $log(\lambda)$ and increasing MSE. Up until $log(\lambda) \approx -4$ we observe a flat rise of MSE. Afterwards the MSE grows very steeply until it reaches a plateau at $log(\lambda) \approx -2.5$. Then we observe a slow rise again until we reach $log(\lambda) \approx -1$. From this point onward the MSE stays at its maximum since there are no non-zero coefficients left.

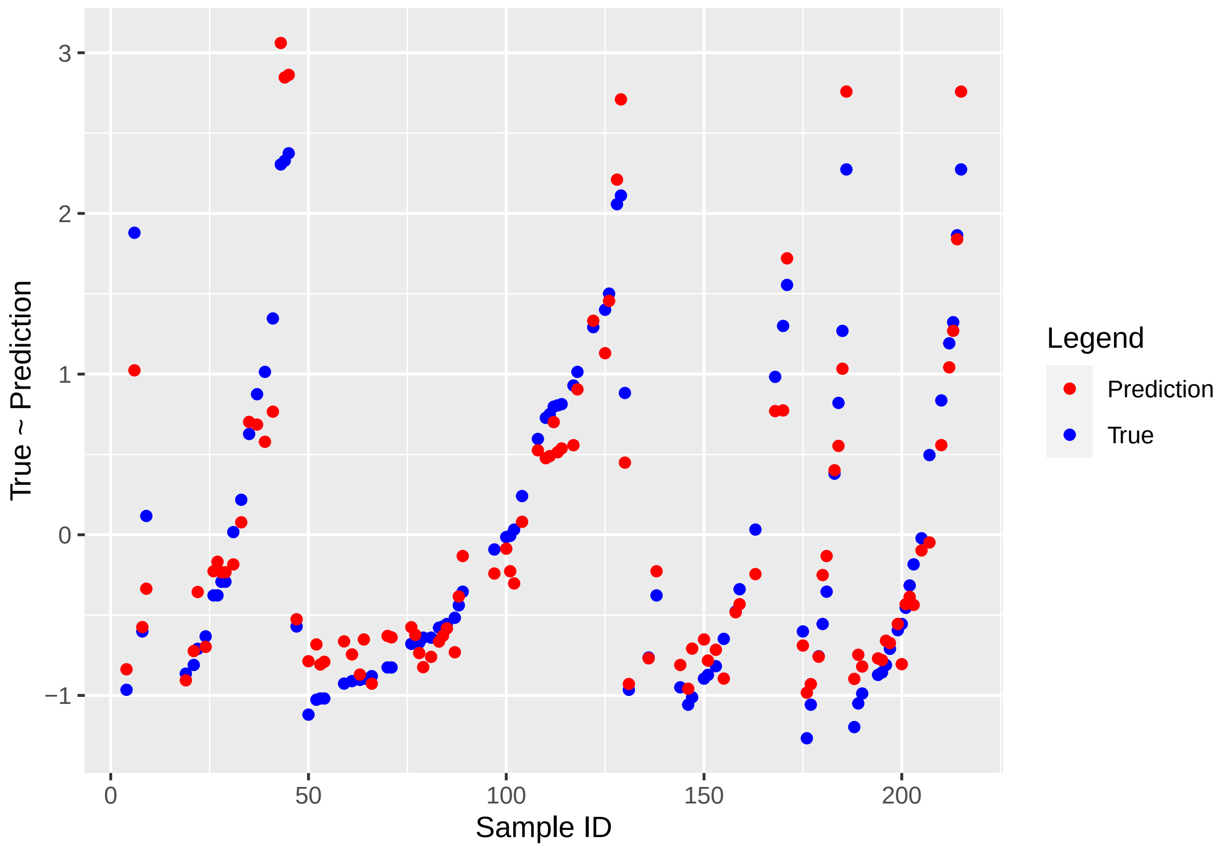
```
## The optimal lambda:  0
## Number of variables choosen:  101
```



```
##    lambda       cvm
## 0.1350000 0.8056608
```

We check *cvm* for $log(\lambda) = -2$ and get a value of 0.7918415. $log(\lambda) = -2$ is significantly worse compared to our optimal λ with cv-score of 0.06060929.

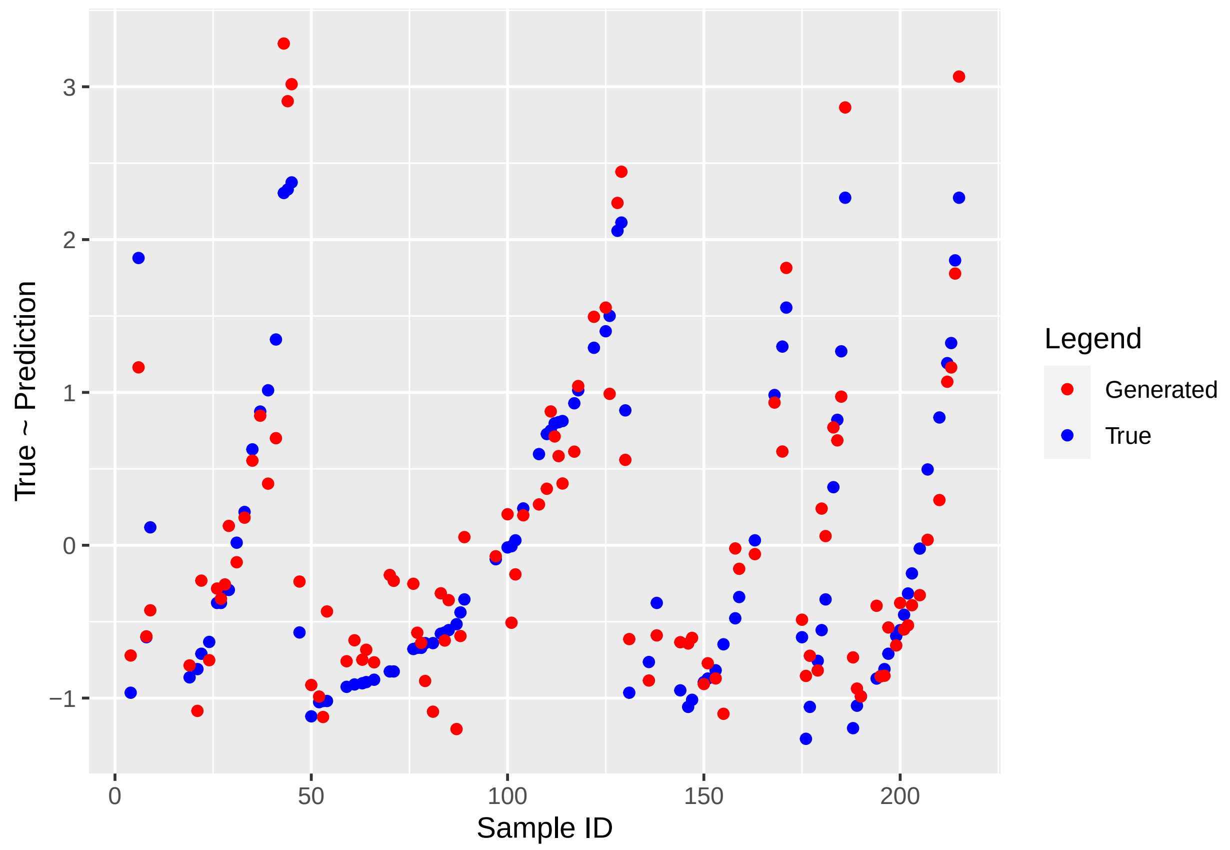
```
## Coefficient of determination:  0.9889883
## MSE:  0.06737877
```



The MSE is better compared to the result of the overfitted linear model from exercise 1. Also, the coefficient of determination is near 1 which proves good predicting ability. Though, it can be seen from the plot that there are still outliers especially for bigger *Fat* values.

7.

```
## Coefficient of determination:  1
## MSE:  0.1231819
```



The generated data seems to be fitting our model. The coefficient of determination is at its maximum. Though we observe that the MSE is worse as the one we have seen in exercise 6.

Appendix: Assignment 3

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
library(glmnet)
library(psych)
library(tidyr)
# Load data
data = read.csv("data/tecator.csv")

# Split data into train and test set
n = dim(data)[1]
set.seed(12345)
id = sample(1:n, floor(n*0.5))
train = data[id,]
test = data[-id,]
# Fitting the linear regression model
predictors <- paste("Channel", 1:100, sep="")
fmla <- as.formula(paste("Fat ~ ", paste(predictors, collapse= "+")))

fit = lm(fmla, data=train)
summary(fit)

# Estimating training and test errors
train_predictions_lm = predict(fit, train)
train_mse = mean((train$Fat-train_predictions_lm)^2)
train_rmse = sqrt(mean((train$Fat-train_predictions_lm)^2))

test_predictions_lm = predict(fit, test)
test_mse = mean((test$Fat-test_predictions_lm)^2)
test_rmse = sqrt(mean((test$Fat-test_predictions_lm)^2))

cat("Train error\nMSE: ", train_mse, "\nRMSE: ", train_rmse, "\n\nTest error\nMSE: ", test_mse, "\nRMSE
covariates = scale(train[,2:101])
response = scale(train$Fat)

# alpha=1 to choose 'Lasso'
set.seed(12345)
model_lasso = glmnet(as.matrix(covariates), response, alpha=1, family="gaussian")
plot(model_lasso, xvar="lambda", label=T)
calc_dof <- function(lambda) {
  ld <- lambda * diag(ncol(covariates))
  H <- covariates %*% solve(t(covariates) %*% covariates + ld) %*% t(covariates)
  dof <- tr(H)
}

dof = c()
lambdas = model_lasso$lambda
for (i in 1:length(lambdas)) {
  dof[i] = calc_dof(lambdas[i])
}

ggplot(data=NULL, aes(lambdas, dof)) + geom_point(color="red")
```

```
# alpha=0 to choose 'Ridge'
set.seed(12345)
model_ridge = glmnet(as.matrix(covariates), response, alpha=0, family="gaussian")
plot(model_ridge, xvar="lambda", label=T)

set.seed(12345)
cv_lasso = cv.glmnet(as.matrix(covariates), response, alpha=1, family="gaussian", lambda=seq(0,1,0.001)
plot(cv_lasso)
optimal_lambda = cv_lasso$lambda.min
cat("The optimal lambda: ", optimal_lambda, "\nNumber of variables choosen: ", length(coef(cv_lasso, s=
lambda_to_compare = round(exp(-2),3)
cvm = cv_lasso$cvm
lambda = cv_lasso$lambda
plot(cv_lasso$lambda, cv_lasso$cvm)

lambda_cvm = cbind(lambda, cvm)
ind_lambda_to_compare = which(lambda_cvm[,1]==lambda_to_compare, arr.ind = TRUE)
cvm_lambda_to_compare = lambda_cvm[ind_lambda_to_compare,]
print(cvm_lambda_to_compare)
test_x = scale(test[, 2:101])
test_y = scale(test$Fat)

model_lasso_optimal = glmnet(as.matrix(covariates), response, alpha = 1, lambda=optimal_lambda)
test_predictions_lasso = predict(model_lasso_optimal, newx=test_x, type="response")

determination_coefficient = sum((test_predictions_lasso-mean(test_y))^2)/sum((test_y-mean(test_y))^2)
test_mse_lasso_optimal = mean((test_predictions_lasso-test_y)^2)

cat("Coefficient of determination: ", determination_coefficient, "\nMSE: ", test_mse_lasso_optimal)

true_pred_df = cbind(test[c(1)], scale(test[c(102)]), test_predictions_lasso)
colnames(true_pred_df)[2] = "True"
colnames(true_pred_df)[3] = "Prediction"

true_pred_df %>%
  gather(key, value, -Sample) %>%
  ggplot(aes(Sample, value)) + geom_point(aes(color = key)) +
  ylab("True ~ Prediction") +
  xlab("Sample ID") +
  scale_colour_manual(name="Legend", values=c("red", "blue"))
calc_sigma <- function(model, covariates, response) {
  betas = as.vector((coef(model))[-1, ])
  residuals = response - (covariates %*% betas)
  sigma = sd(residuals)
  return(sigma)
}
sigma = calc_sigma(model_lasso_optimal, covariates, response)

set.seed(12345)
generated_data = rnorm(length(test_y), test_predictions_lasso, sigma)

determination_coefficient = sum((generated_data-mean(test_y))^2)/sum((generated_data-mean(test_y))^2)
test_mse_lasso_optimal = mean((generated_data-test_y)^2)
```

```r
cat("Coefficient of determination: ", determination_coefficient, "\nMSE: ", test_mse_lasso_optimal)

true_pred_df = cbind(test[c(1)], scale(test[c(102)]), generated_data)
colnames(true_pred_df)[2] = "True"
colnames(true_pred_df)[3] = "Generated"

true_pred_df %>%
  gather(key, value, -Sample) %>%
  ggplot(aes(Sample, value)) + geom_point(aes(color = key)) +
  ylab("True ~ Prediction") +
  xlab("Sample ID") +
  scale_colour_manual(name="Legend", values=c("red", "blue"))
```