### Question 1  
library(caret)

# Reading the data  
parkinsons<-read.csv("parkinsons.csv")  
new\_parkinsons <- subset(parkinsons, select = -c(subject., age, sex, test\_time, total\_UPDRS))  
  
# Dividing training and test data(60/40)   
n <- dim(new\_parkinsons)[1]  
set.seed(12345)  
id <- sample(1:n, floor(n\*0.6))  
train <- new\_parkinsons[id,]  
test <- new\_parkinsons[-id,]  
  
#Scaling the data  
scaler <- preProcess(train)  
train\_data <- predict(scaler, train)  
test\_data <- predict(scaler, test)  
  
##Question 2  
  
linear\_modelp <- lm(motor\_UPDRS~0 +., data = train\_data)  
summary(linear\_modelp)

##   
## Call:  
## lm(formula = motor\_UPDRS ~ 0 + ., data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.0255 -0.7363 -0.1087 0.7333 2.1960   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## Jitter... 0.186931 0.149561 1.250 0.211431   
## Jitter.Abs. -0.169609 0.040805 -4.157 3.31e-05 \*\*\*  
## Jitter.RAP -5.269544 18.834160 -0.280 0.779658   
## Jitter.PPQ5 -0.074568 0.087766 -0.850 0.395592   
## Jitter.DDP 5.249558 18.837525 0.279 0.780510   
## Shimmer 0.592436 0.205981 2.876 0.004050 \*\*   
## Shimmer.dB. -0.172655 0.139316 -1.239 0.215315   
## Shimmer.APQ3 32.070932 77.159242 0.416 0.677694   
## Shimmer.APQ5 -0.387507 0.113789 -3.405 0.000668 \*\*\*  
## Shimmer.APQ11 0.305546 0.061236 4.990 6.34e-07 \*\*\*  
## Shimmer.DDA -32.387241 77.158814 -0.420 0.674695   
## NHR -0.185387 0.045567 -4.068 4.84e-05 \*\*\*  
## HNR -0.238543 0.036395 -6.554 6.41e-11 \*\*\*  
## RPDE 0.004068 0.022664 0.179 0.857556   
## DFA -0.280318 0.020136 -13.921 < 2e-16 \*\*\*  
## PPE 0.226467 0.032881 6.887 6.70e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9394 on 3509 degrees of freedom  
## Multiple R-squared: 0.1212, Adjusted R-squared: 0.1172   
## F-statistic: 30.25 on 16 and 3509 DF, p-value: < 2.2e-16

From the summary of linear regression model, Jitter.Abs, Shimmer.APQ11, NHR, HNR, DFA, PPE contribute significantly to motor\_UPDRS. at 0.001 significance level

#MSE Train  
train\_data\_predict <- predict(linear\_modelp, train\_data)  
train\_MSE <- sum((train\_data$motor\_UPDRS - train\_data\_predict)^2) / nrow(train\_data)  
  
#MSE Test  
test\_data\_predict <- predict(linear\_modelp, test\_data)  
test\_MSE <- sum((test\_data$motor\_UPDRS - test\_data\_predict)^2) / nrow(test\_data)  
  
  
### Question 3  
  
##(3.a) Loglikelihood  
  
loglikelihood <- function(parameter)  
{  
 X <- as.matrix(train\_data[,-1])  
 n <- nrow(X)  
 y <- train\_data[,1]  
 sigma <- parameter[17]  
 theta <- parameter[1:16]  
 return( -n/2 \* log(2\*pi) - n/2\*log(sigma^2) - 1/(2\*sigma^2) \* sum((y - X%\*%theta)^2))  
}  
  
##(3.b) Ridge Function  
  
Ridge <- function(parameter, lambda)  
{  
 return( -loglikelihood(parameter) + lambda \* sum(parameter^2))  
}  
  
##(3.c) RidgeOpt Function  
  
RidgeOpt <- function(lambda)  
{  
 return(optim( rep(1,17), fn = Ridge, method = "BFGS", lambda = lambda))  
}  
  
##(3.d) Degree of Freedom  
  
DF <- function(lambda)  
{  
 P <- as.matrix(train\_data[,-1])  
 degree <- P %\*% solve(t(P) %\*% P + lambda \* diag(ncol(P))) %\*% t(P)  
 return(sum(diag(degree)))  
}  
  
## Question 4  
  
train\_P <- as.matrix(train\_data[,-1])  
test\_P <- as.matrix(test\_data[,-1])  
  
# Prediction with lambda=1  
  
ridgeopt\_1 <- RidgeOpt(lambda = 1)  
  
predict\_train\_1 <- train\_P %\*% ridgeopt\_1$par[1:16]  
predict\_test\_1 <- test\_P %\*% ridgeopt\_1$par[1:16]  
  
error\_train\_1 <- mean((predict\_train\_1 - train\_data$motor\_UPDRS)^2)  
error\_test\_1 <- mean((predict\_test\_1 - test\_data$motor\_UPDRS)^2)  
  
# Prediction with lambda=100  
  
ridgeopt\_100 <- RidgeOpt(lambda = 100)  
  
predict\_train\_100 <- train\_P %\*% ridgeopt\_100$par[1:16]  
predict\_test\_100 <- test\_P %\*% ridgeopt\_100$par[1:16]  
  
error\_train\_100 <- mean((predict\_train\_100 - train\_data$motor\_UPDRS)^2)  
error\_test\_100 <- mean((predict\_test\_100 - test\_data$motor\_UPDRS)^2)  
  
# Prediction with lambda=100  
  
ridgeopt\_1000 <- RidgeOpt(lambda = 1000)  
  
predict\_train\_1000 <- train\_P %\*% ridgeopt\_1000$par[1:16]  
predict\_test\_1000 <- test\_P %\*% ridgeopt\_1000$par[1:16]  
  
error\_train\_1000 <- mean((predict\_train\_1000 - train\_data$motor\_UPDRS)^2)  
error\_test\_1000 <- mean((predict\_test\_1000 - test\_data$motor\_UPDRS)^2)  
  
list(error\_train\_1 = error\_train\_1, error\_train\_100 = error\_train\_100, error\_train\_1000 = error\_train\_1000)

## $error\_train\_1  
## [1] 0.8786272  
##   
## $error\_train\_100  
## [1] 0.8841695  
##   
## $error\_train\_1000  
## [1] 0.9128817

list(error\_test\_1 = error\_test\_1, error\_test\_100 = error\_test\_100, error\_test\_1000=error\_test\_1000)

## $error\_test\_1  
## [1] 0.9349982  
##   
## $error\_test\_100  
## [1] 0.9322925  
##   
## $error\_test\_1000  
## [1] 0.9481592

degree\_1 <- DF(1)  
degree\_100 <- DF(100)  
degree\_1000 <- DF(1000)  
  
result <- data.frame( lambda = c(1,100,1000),   
 MSE\_train = c(error\_train\_1, error\_train\_100, error\_train\_1000),   
 MSE\_test = c(error\_test\_1,error\_test\_100,error\_test\_1000),  
 DF = c(degree\_1, degree\_100, degree\_1000))  
  
result

## lambda MSE\_train MSE\_test DF  
## 1 1 0.8786272 0.9349982 13.860736  
## 2 100 0.8841695 0.9322925 9.924887  
## 3 1000 0.9128817 0.9481592 5.643925

**Which penalty parameter is most appropriate among the selected ones?**

The training MSE is lowest for𝜆 = 1 and the test MSE value is lowest for 𝜆 = 100. The training and test MSE value is highest when 𝜆 = 1000. So, the most appropriate penalty parameter among the selected is 𝜆 = 100.

**Compute and compare the degrees of freedom of these models and make appropriate conclusions**

Looking at the result data, we can come to conclusion that when 𝜆value increases, the degree of freedom decreases.