Applied Predictive Modeling 1/23/2018

Chapter 6: Linear Regression and Its Cousins

Name: Kamaru-Deen Lawal

Email: NA

1

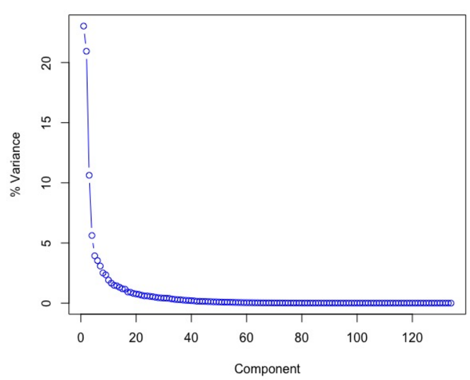
(a)

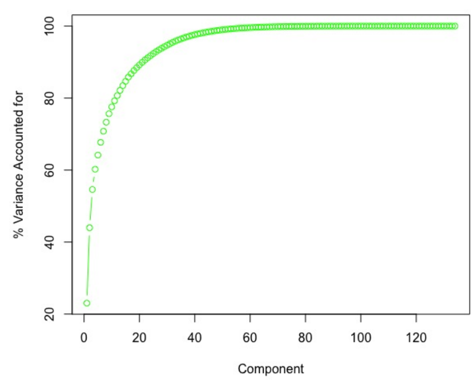
```
library(caret)
data(tecator)
?tecator
dim(absorp)
dim(endpoints)
```

(b)

```
# Find Principal Components
pcaObject <- prcomp(absorp, center = TRUE, scale = TRUE)
percentvariance <- pcaobject$sd^2 / sum(pcaobject$sd^2) * 100
plot(percentvariance, type = "b", col = "blue", xlab = "Component", ylab = "% Variance")
names(pcaobject)

# Cumulative Sum of Percent Variance
cpv <- cumsum(percentvariance) # 31 components accounts for 95% of variance
plot(cpv, type = "b", col = "green", xlab = "Component", ylab = "% Variance Accounted for
    ")
```

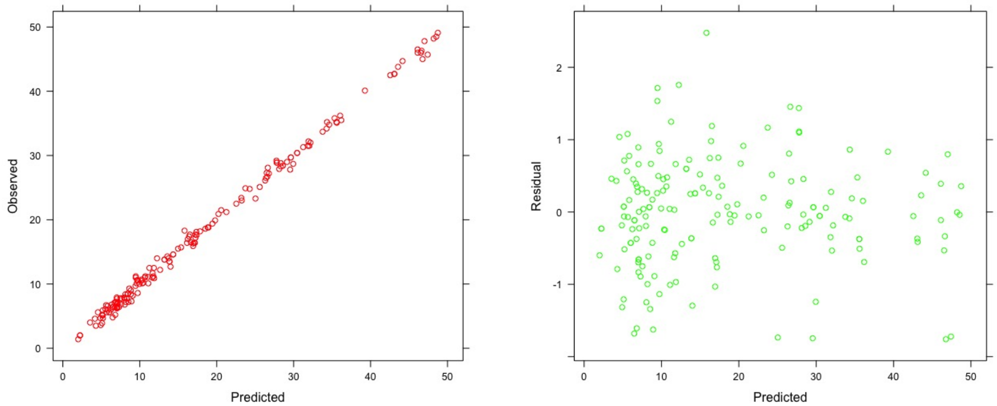




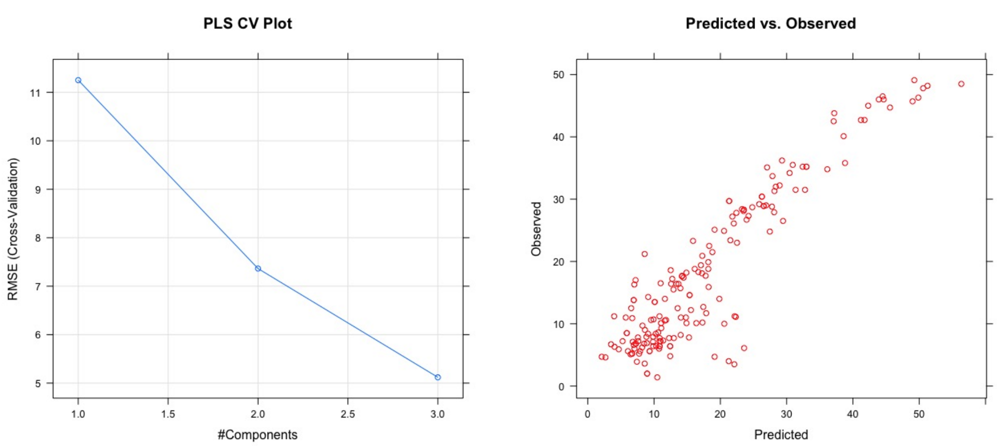
(c)

```
# Split Data
set.seed(777)
trainlabels <- createDataPartition(endpoints[, 2], p = .8, list = FALSE)
trainab <- absorp[trainlabels, ]
testab <- absorp[-trainlabels, ]
ctrl <- trainControl(method = "cv", number = 10)
trainfat <- endpoints[trainlabels, 2]
testfat <- endpoints[-trainlabels, 2]
```

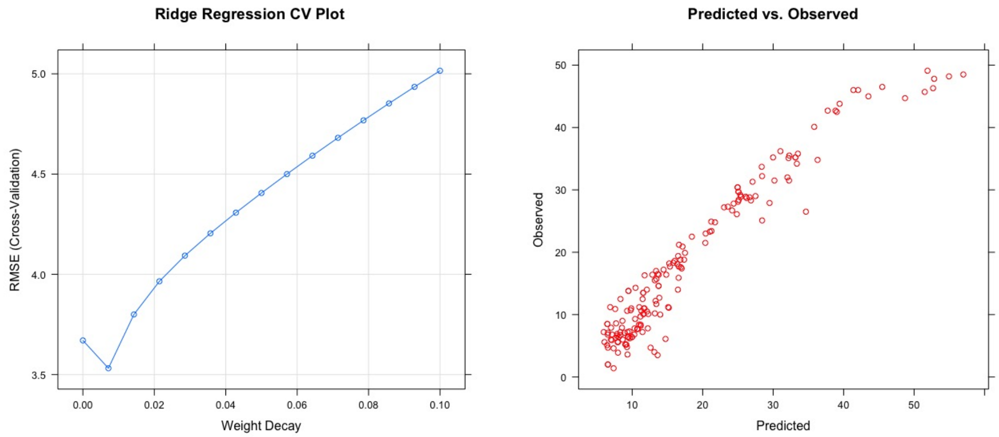
```
# Linear Model
lmfit1 <- train(x = data.frame(trainab), y = trainfat, method = "lm", trControl = ctrl)
xyplot(trainfat ~ predict(lmfit1), col = "red", xlab = "Predicted", ylab = "Observed")
xyplot(resid(lmfit1) ~ predict(lmfit1), col = "green", xlab = "Predicted", ylab = "
    Residual")
```



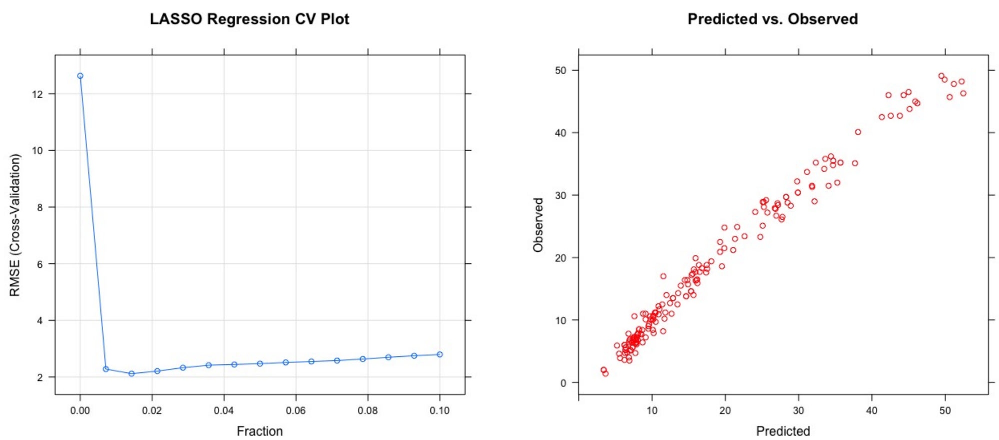

```
# PLS
plsregfit <- train(data.frame(trainab), trainfat, method = "pls", tunelength = 20,
                trControl = ctrl, preProc = c("center", "scale"))
plot(plsregfit, main = "PLS CV Plot") # 3 components
xyplot(trainfat ~ predict(plsregfit), col = "red", xlab = "Predicted", ylab = "Observed",
     main = "Predicted vs. Observed")
```




```
# Ridge Regression
ridgegrid <- data.frame(.lambda = seq(0, .1, length = 15))
ridgeregfit <- train(x = data.frame(trainab), trainfat, method = "ridge", tuneGrid =
    ridgegrid,
                   trControl = ctrl, preProc = c("center", "scale"))
plot(ridgeregfit, main = "Ridge Regression CV Plot") # lambda = .007
xyplot(trainfat ~ predict(ridgeregfit), col = "red", xlab = "Predicted", ylab = "Observed
    ", main = "Predicted vs. Observed")
```

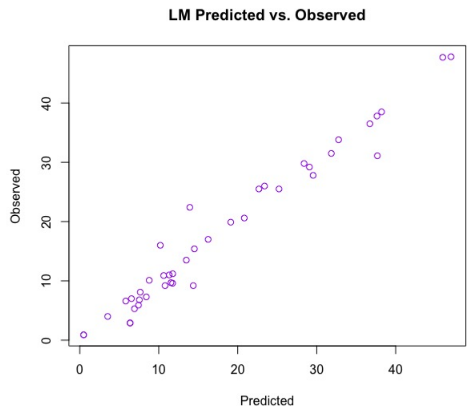


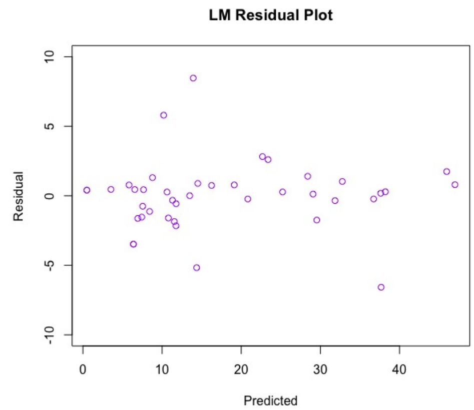

```
# LASSO Regression
lassogrid <- data.frame(.fraction = seq(0, .1, length = 15))
lassoregfit <- train(x = data.frame(trainab), trainfat, method = "lars", tuneGrid =
    lassogrid,
                trControl = ctrl, preProc = c("center", "scale"))
plot(lassoregfit, main = "LASSO Regression CV Plot") # fraction = 0.0142
xyplot(trainfat ~ predict(lassoregfit), col = "red", xlab = "Predicted", ylab = "Observed
    ", main = "Predicted vs. Observed")
```



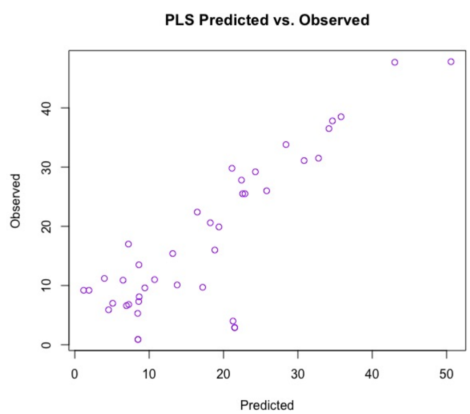
(d)

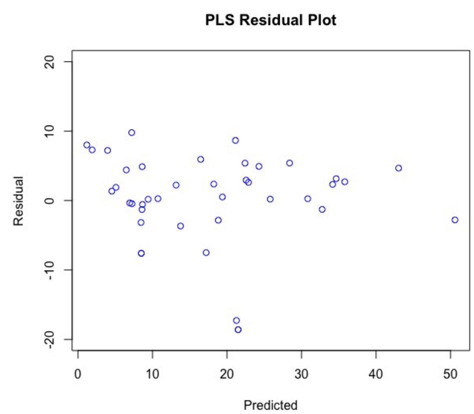
```
# Linear Model Predictions
lmpred <- predict(lmfit1, newdata = data.frame(testab))
par(mfrow = c(1, 2))
plot(lmpred, testfat, col = "purple", main = "LM Predicted vs. Observed", xlab = "
    Predicted", ylab = "Observed")
summary(testfat - lmpred)
plot(lmpred, testfat - lmpred, ylim = c(-10,10), main = "LM Residual Plot", xlab = "
    Predicted", ylab = "Residual", col = "purple")
lm_mse <- mean((lmpred - testfat) ^ 2) # 6.037773
```



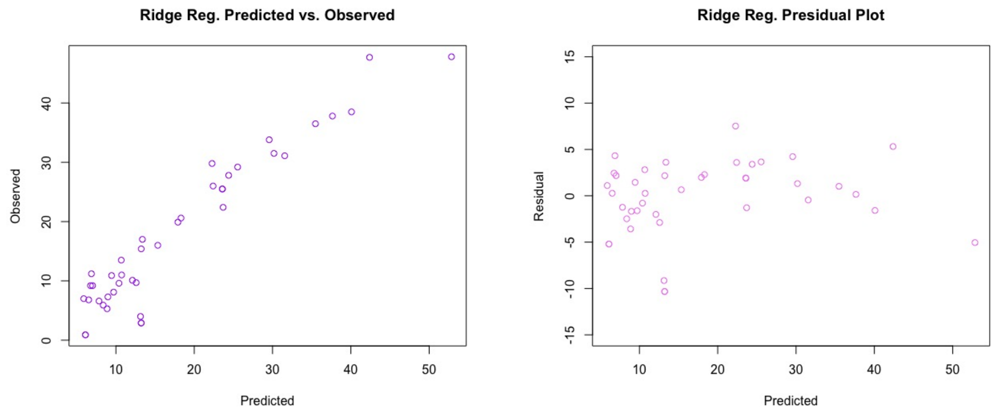



```
# PLS Predictions
plspred <- predict(plsregfit, newdata = data.frame(testab))
plot(plspred, testfat, col = "purple", main = "PLS Predicted vs. Observed", xlab = "
    Predicted", ylab = "Observed")
summary(testfat - plspred)
plot(plspred, testfat - plspred, ylim = c(-20,20), main = "PLS Residual Plot", xlab = "
    Predicted", ylab = "Residual", col = "blue")
pls_mse <- mean((plspred - testfat) ^ 2) # 43.46856
```

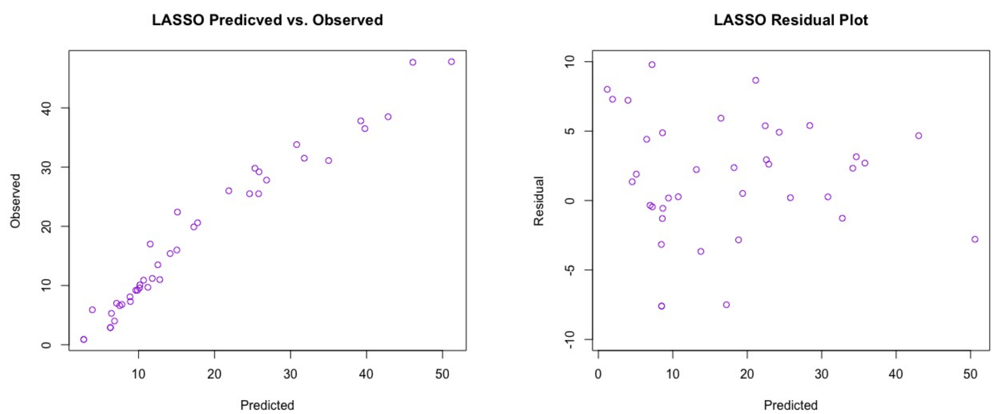





```
# Ridge Regression
ridgepred <- predict(ridgeregfit, newdata = data.frame(testab))
plot(ridgepred, testfat, col = "purple", main = "Ridge Reg. Predicted vs. Observed", xlab
     = "Predicted", ylab = "Observed")
summary(testfat - ridgepred)
plot(ridgepred, testfat - ridgepred, ylim = c(-15, 15), main = "Ridge Reg. Presidual Plot
    ", xlab = "Predicted", ylab = "Residual", col = "violet")
ridge_mse <- mean((ridgepred - testfat) ^ 2) # 15.57342
```




```
# LASSO
lassopred <- predict(lassoregfit, newdata = data.frame(testab))
plot(lassopred, testfat, col = "purple", main = "LASSO Predicved vs. Observed", xlab = "
    Predicted", ylab = "Observed")
summary(testfat - lassopred)
plot(plspred, testfat - plspred, ylim = c(-10,10), main = "LASSO Residual Plot", xlab = "
    Predicted", ylab = "Residual", col = "purple")
lasso_mse <- mean((lassopred - testfat) ^ 2) # 6.752217
```



(e) The Linear model has the best predictive ability, and the LASSO model is a close runner up. The Ridge Regression and the Partial Least Squares model were both the worst performing models, with the PLS model performing significantly worse than the other models. In this case I would use the Linear model because it has the lowest MSE, and is the most interpretable of all the models used. The lack of accuracy for the PLS model highlights the point made in the chapter about variance amongst predictors not necessarily being an indicator for final model accuracy.

2

(a)

```
library(AppliedPredictiveModeling)
data(permeability)
```

(b)

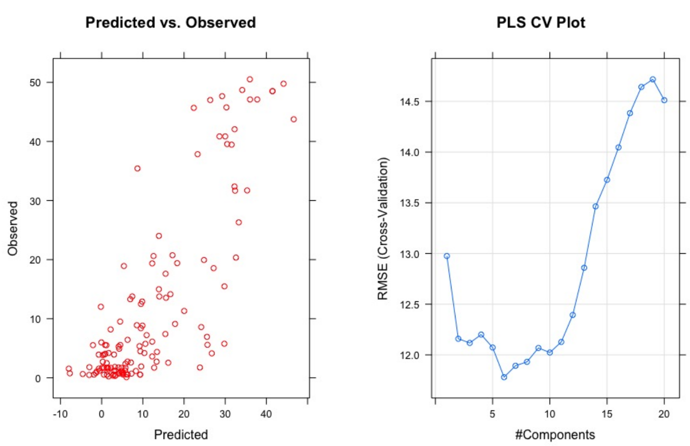
```
nzv <- nearZeroVar(fingerprints)
fpdata <- fingerprints[, -nzv] # 388 Predictors left for modeling
```

(c)

```
set.seed(777)
trainlabels <- createDataPartition(permeability[, 1], p = .8, list = FALSE)
trainfp <- fpdata[trainlabels, ]
testfp <- fpdata[-trainlabels, ]
trainperm <- permeability[trainlabels, ]
testperm <- permeability[-trainlabels, ]
ctrl <- trainControl(method = "cv", number = 10)

# PLS Model
set.seed(777)
plsfit <- train(data.frame(trainfp), trainperm, method = "pls", tuneLength = 20,
                trControl = ctrl, preProc = c("center", "scale"), metric = "Rsquared")
plot(plsfit, main = "PLS CV Plot") # 6 components

# PLS Prediction
set.seed(777)
plspred <- predict(plsfit, newdata = data.frame(testfp))
plot(plspred, testperm, col = "purple", main = "PLS Predicted vs. Observed", xlab = "
    Predicted", ylab = "Observed", xlim = c(-10, 50), ylim = c(-10, 50))
summary(testperm - plspred)
plot(plspred, testperm - plspred, ylim = c(-30,30), main = "PLS Residual Plot", xlab = "
    Predicted", ylab = "Residual", col = "purple")
pls_mse <- mean((plspred - testfat) ^ 2) # 104.21
```



(d)

```
# LASSO Model + Prediction
set.seed(777)
lassogrid <- data.frame(.fraction = seq(0, 1, length = 5))
```

```r
lassoregfit <- train(x = data.frame(trainfp), trainperm, method = "lars", tuneGrid =
    lassogrid,
                   trControl = ctrl, preProc = c("center", "scale"))
plot(lassoregfit, main = "LASSO Regression CV Plot") # fraction = 0.0142
xyplot(trainfat ~ predict(lassoregfit), col = "red", xlab = "Predicted", ylab = "Observed
    ", main = "Predicted vs. Observed")
```

The number of predictors is greater than the number of samples, so building a linear model in this case does not make sense.