ex3

chris

2023-07-03

```r
library(caret)
data(tecator)
str(absorp)
```

```
##  num [1:215, 1:100] 2.62 2.83 2.58 2.82 2.79 ...
```

```r
str(endpoints)
```

```
##  num [1:215, 1:3] 60.5 46 71 72.8 58.3 44 44 69.3 61.4 61.4 ...
```

b) In this example the predictors are the measurements at the individual frequencies. Because the frequencies lie in a systematic order (850–1,050nm), the predictors have a high degree of correlation. Hence, the data lie in a smaller dimension than the total number of predictors (215). Use PCA to determine the effective dimension of these data. What is the effective dimension?

```r
pc <- prcomp(absorp, center=T,scale=T)

summary(pc)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     9.9311 0.9847 0.52851 0.33827 0.08038 0.05123 0.02681
## Proportion of Variance 0.9863 0.0097 0.00279 0.00114 0.00006 0.00003 0.00001
## Cumulative Proportion  0.9863 0.9960 0.99875 0.99990 0.99996 0.99999 0.99999
##                            PC8      PC9     PC10     PC11     PC12     PC13
## Standard deviation     0.01961 0.008564 0.006739 0.004442 0.003361 0.001867
## Proportion of Variance 0.00000 0.000000 0.000000 0.000000 0.000000 0.000000
## Cumulative Proportion  1.00000 1.000000 1.000000 1.000000 1.000000 1.000000
##                            PC14      PC15      PC16      PC17      PC18
## Standard deviation     0.001377 0.0009449 0.0008641 0.0007558 0.0006977
## Proportion of Variance 0.000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.000000 1.0000000 1.0000000 1.0000000 1.0000000
##                             PC19      PC20      PC21      PC22      PC23
## Standard deviation     0.0005884 0.0004628 0.0003897 0.0003341 0.0003123
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##                             PC24      PC25     PC26      PC27      PC28
## Standard deviation     0.0002721 0.0002616 0.000211 0.0001954 0.0001857
## Proportion of Variance 0.0000000 0.0000000 0.000000 0.0000000 0.0000000
```

```
## Cumulative Proportion  1.0000000 1.0000000 1.000000 1.0000000 1.0000000
##                             PC29      PC30      PC31      PC32      PC33
## Standard deviation     0.0001729 0.0001656 0.0001539 0.0001473 0.0001392
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##                             PC34      PC35      PC36     PC37     PC38
## Standard deviation     0.0001339 0.0001269 0.0001082 0.000104 9.98e-05
## Proportion of Variance 0.0000000 0.0000000 0.0000000 0.000000 0.00e+00
## Cumulative Proportion  1.0000000 1.0000000 1.0000000 1.000000 1.00e+00
##                             PC39      PC40      PC41      PC42     PC43
## Standard deviation     9.081e-05 8.668e-05 8.026e-05 7.762e-05 7.36e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.00e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.00e+00
##                             PC44      PC45     PC46      PC47      PC48
## Standard deviation     6.808e-05 6.541e-05 6.44e-05 5.897e-05 5.422e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.00e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.00e+00 1.000e+00 1.000e+00
##                             PC49      PC50      PC51      PC52      PC53
## Standard deviation     5.027e-05 4.893e-05 4.608e-05 4.419e-05 4.037e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC54    PC55     PC56      PC57      PC58      PC59
## Standard deviation     3.854e-05 3.8e-05 3.64e-05 3.497e-05 3.443e-05 3.264e-05
## Proportion of Variance 0.000e+00 0.0e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.0e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC60     PC61      PC62      PC63      PC64
## Standard deviation     3.104e-05 3.04e-05 2.959e-05 2.844e-05 2.699e-05
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC65      PC66      PC67      PC68      PC69
## Standard deviation     2.586e-05 2.388e-05 2.364e-05 2.284e-05 2.173e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC70      PC71     PC72      PC73      PC74
## Standard deviation     2.058e-05 1.997e-05 1.93e-05 1.854e-05 1.807e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.00e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.00e+00 1.000e+00 1.000e+00
##                             PC75      PC76      PC77      PC78      PC79
## Standard deviation     1.728e-05 1.693e-05 1.612e-05 1.569e-05 1.516e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC80      PC81      PC82      PC83      PC84
## Standard deviation     1.445e-05 1.408e-05 1.356e-05 1.275e-05 1.224e-05
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC85     PC86      PC87      PC88      PC89
## Standard deviation     1.178e-05 1.09e-05 1.045e-05 1.009e-05 9.396e-06
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.000e+00 1.000e+00
##                             PC90     PC91      PC92     PC93      PC94
## Standard deviation     8.728e-06 8.27e-06 7.613e-06 6.83e-06 6.383e-06
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.00e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.00e+00 1.000e+00
##                             PC95      PC96      PC97      PC98      PC99
```

```
## Standard deviation     5.946e-06 5.478e-06 4.826e-06 4.521e-06 4.164e-06
## Proportion of Variance 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.000e+00 1.000e+00
##                            PC100
## Standard deviation     4.122e-06
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

The first principal explains 98.63% of the variance, thus the data is effectively unidimensional.

c) Split the data into a training and a test set the response of the percentage of moisture, pre-process the data, and build each variety of models described in this chapter. For those models with tuning parameters, what are the optimal values of the tuning parameter(s)?

```r
set.seed(111)
train <- createDataPartition(endpoints[,1], p=.80, list=F)

predicttrain <- as.data.frame(absorp[train,])
predicttest <- as.data.frame(absorp[-train,])
outcometrain <- endpoints[train, 1]
outcometest <- endpoints[-train, 1]
```

Train models on 80% of the data. Outcome we are predicting is percentage of moisture.

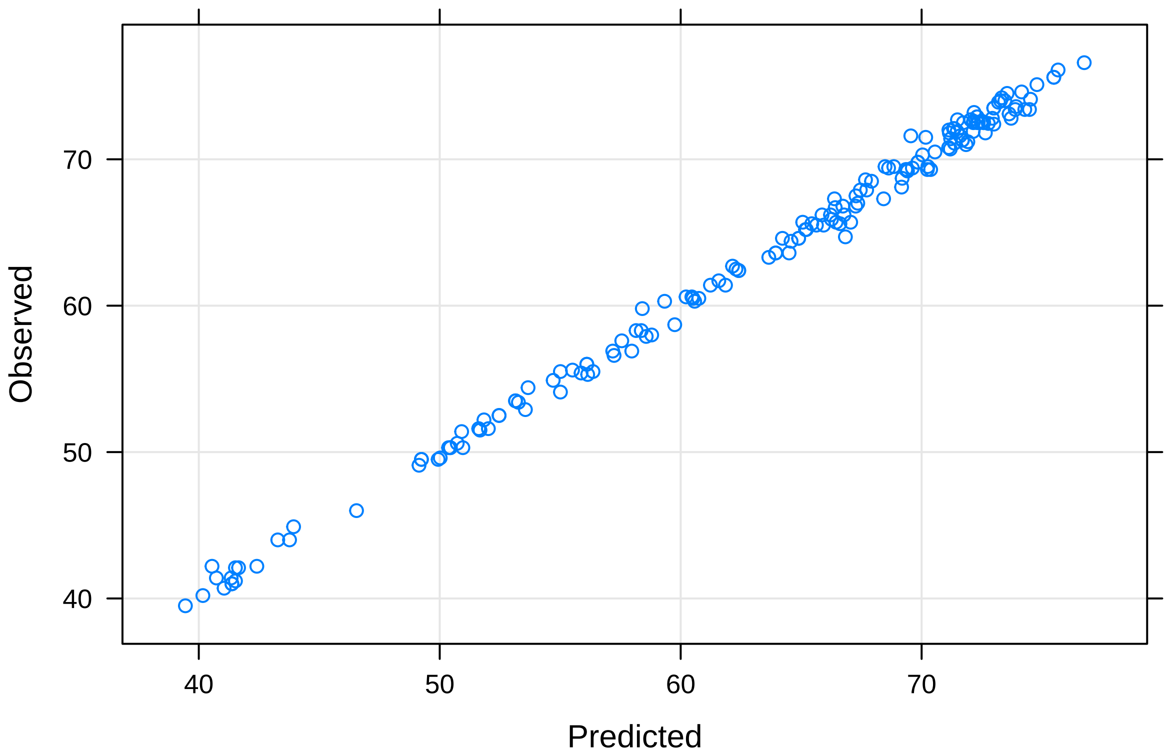
Train a linear regression model using 10-fold cross-validation

```r
set.seed(111)
lm <- train(x=predicttrain,
            y=outcometrain,
            method='lm',
            trControl=trainControl(method="cv", number=10))

lm
```

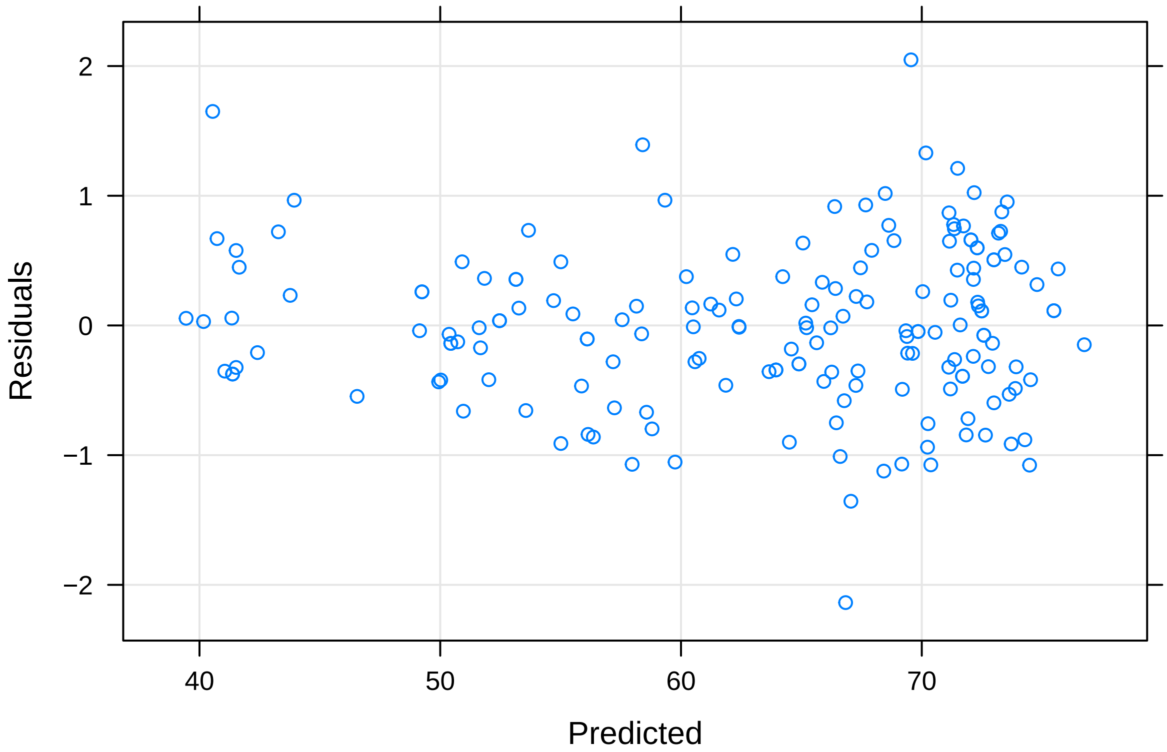
```
## Linear Regression
##
## 175 samples
## 100 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 158, 158, 156, 159, 157, 156, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2.469334  0.9442206  1.515848
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Plot the observed (from testing set) vs predicted values (from training set). Plot residuals of the training model vs the predicted values. This should show no pattern.

```r
lmpred <- predict(lm, predicttest)

xyplot(outcometrain ~ predict(lm),
  type = c("p", "g"),
  xlab = "Predicted", ylab = "Observed")
```




```r
xyplot(resid(lm) ~ predict(lm),
  type = c("p", "g"),
  xlab = "Predicted", ylab = "Residuals")
```



Looks pretty good.

Train a principal components regression using 10-fold cross-validation

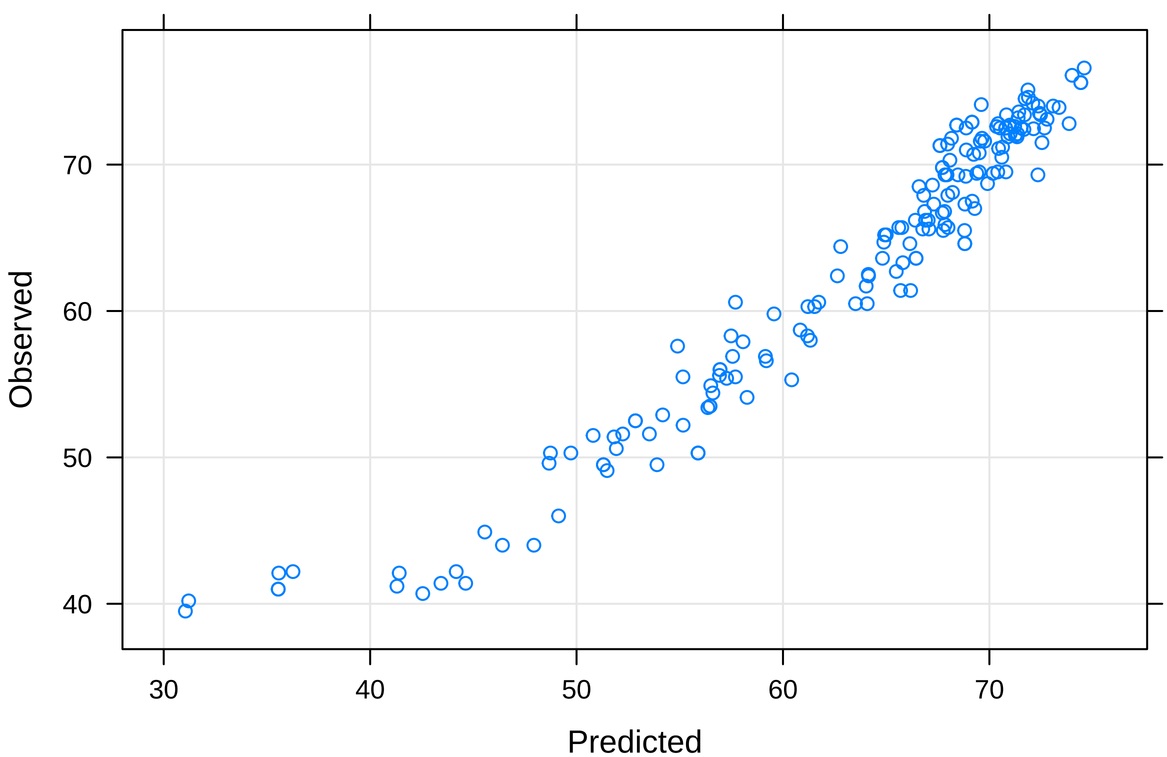
```r
set.seed(111)
pcr <- train(x=predicttrain,
             y=outcometrain,
             method='pcr',
             trControl=trainControl(method="cv", number=10),
             tuneLength=10)

pcr
```

```
## Principal Component Analysis
##
## 175 samples
## 100 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 158, 158, 156, 159, 157, 156, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE      Rsquared   MAE
##    1     8.732817  0.2454458  7.278465
##    2     8.702855  0.2587405  7.158763
```

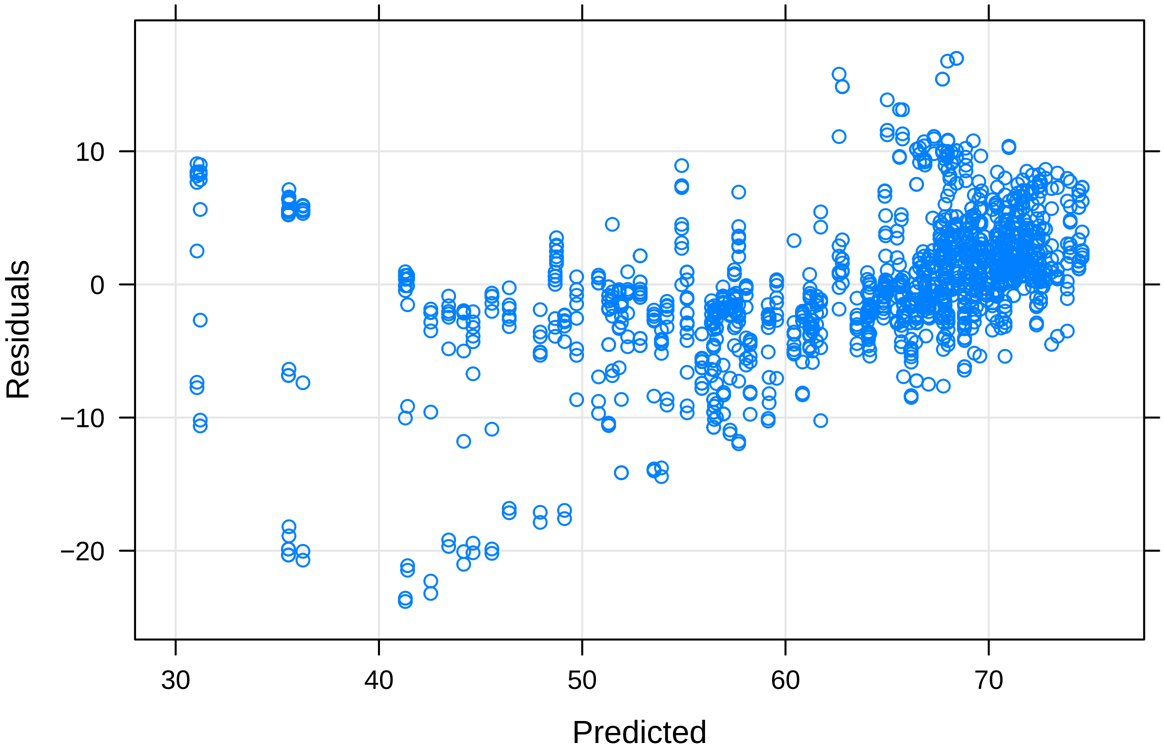
```
##    3     6.166037  0.6248224  5.072502
##    4     3.341998  0.8945293  2.695482
##    5     2.846718  0.9283716  2.269699
##    6     2.757673  0.9334208  2.193450
##    7     2.717331  0.9345690  2.157157
##    8     2.668456  0.9389073  2.112536
##    9     2.704526  0.9370858  2.136534
##   10     2.718902  0.9371467  2.105594
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 8.
```

Optimal principal components = 8

```
pcrpred <- predict(pcr, predicttest)

xyplot(outcometrain ~ predict(pcr),
  type = c("p", "g"),
  xlab = "Predicted", ylab = "Observed")
```




```
xyplot(resid(pcr) ~ predict(pcr),
  type = c("p", "g"),
  xlab = "Predicted", ylab = "Residuals")
```



Some outliers at lower predicted values.

Train a lasso regression model using 10-fold cross-validation

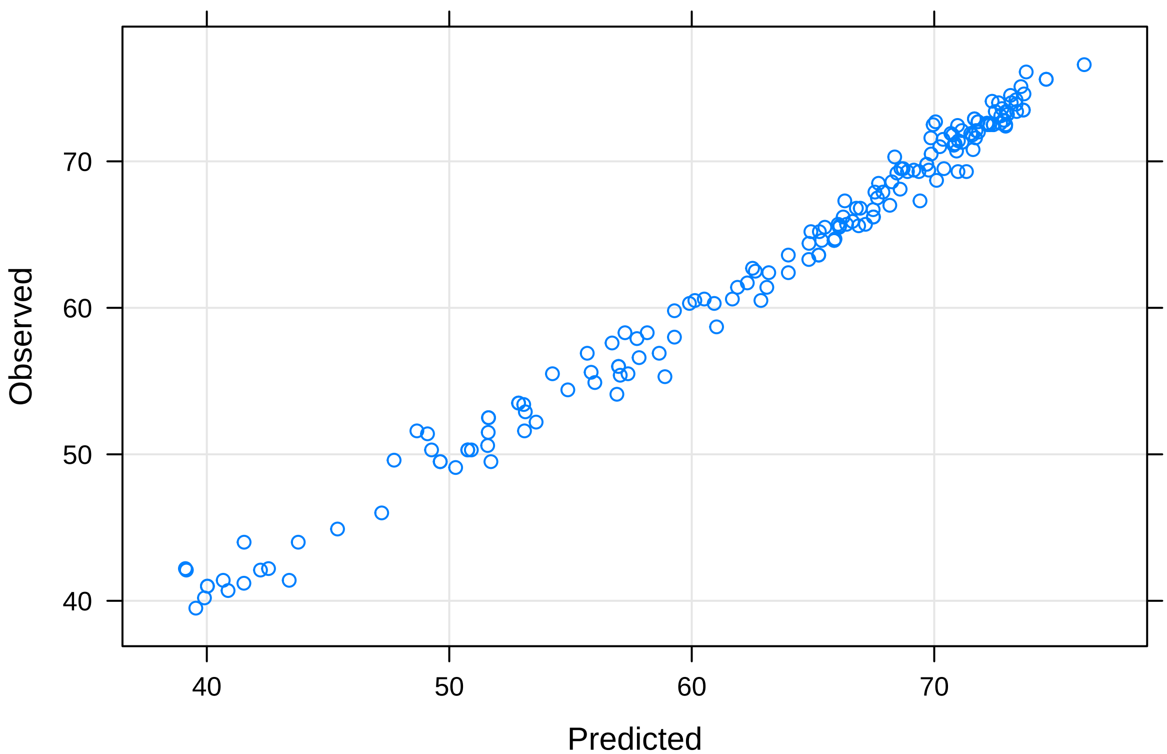
```
set.seed(111)
lasso <- train(x=predicttrain,
               y=outcometrain,
               method='lasso',
               trControl=trainControl(method="cv", number=10),
               tuneLength=10)

lasso
```

```
## The lasso
##
## 175 samples
## 100 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 158, 158, 156, 159, 157, 156, ...
## Resampling results across tuning parameters:
##
##   fraction   RMSE      Rsquared   MAE
##   0.1000000  2.288559  0.9498322  1.520358
##   0.1888889  2.498725  0.9372794  1.532576
##   0.2777778  2.465400  0.9393976  1.532947
```

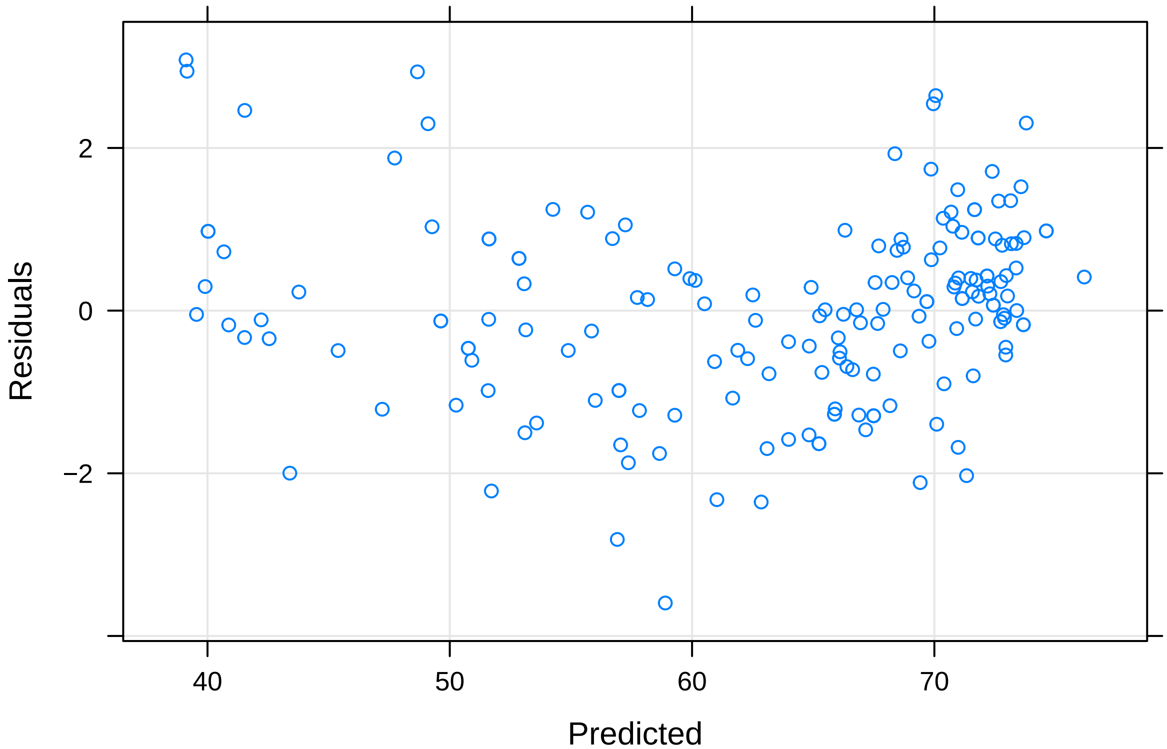
```
##   0.3666667  2.414537  0.9415405  1.492477
##   0.4555556  2.356738  0.9442840  1.460027
##   0.5444444  2.329111  0.9465609  1.449341
##   0.6333333  2.377737  0.9453798  1.473832
##   0.7222222  2.357705  0.9474424  1.476261
##   0.8111111  2.344851  0.9488697  1.471363
##   0.9000000  2.375282  0.9480120  1.481852
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.1.
```

Optimal value of fraction of full solution = 0.1

```
lassopred <- predict(lasso, predicttest)

xyplot(outcometrain ~ predict(lasso),
  type = c("p", "g"),
  xlab = "Predicted", ylab = "Observed")
```




```
xyplot(resid(lasso) ~ predict(lasso),
  type = c("p", "g"),
  xlab = "Predicted", ylab = "Residuals")
```



Looks good

d) Which model has the best predictive ability? Is any model significantly better or worse than the others?

Lasso regression had the lowest cross-validation error (RMSE = 2.29), followed by linear regression (RMSE = 2.47), and PCR (RMSE = 2.67)

e) Explain which model you would use for predicting the percentage of moisture of a sample.

I would use a lasso regression as this model had the best performance in predicting the values of the test set, had consistent performance across the range of values, and had the lowest RMSE of the prediction models.