Lab2

Tristen Tooming

1/18/2021

1

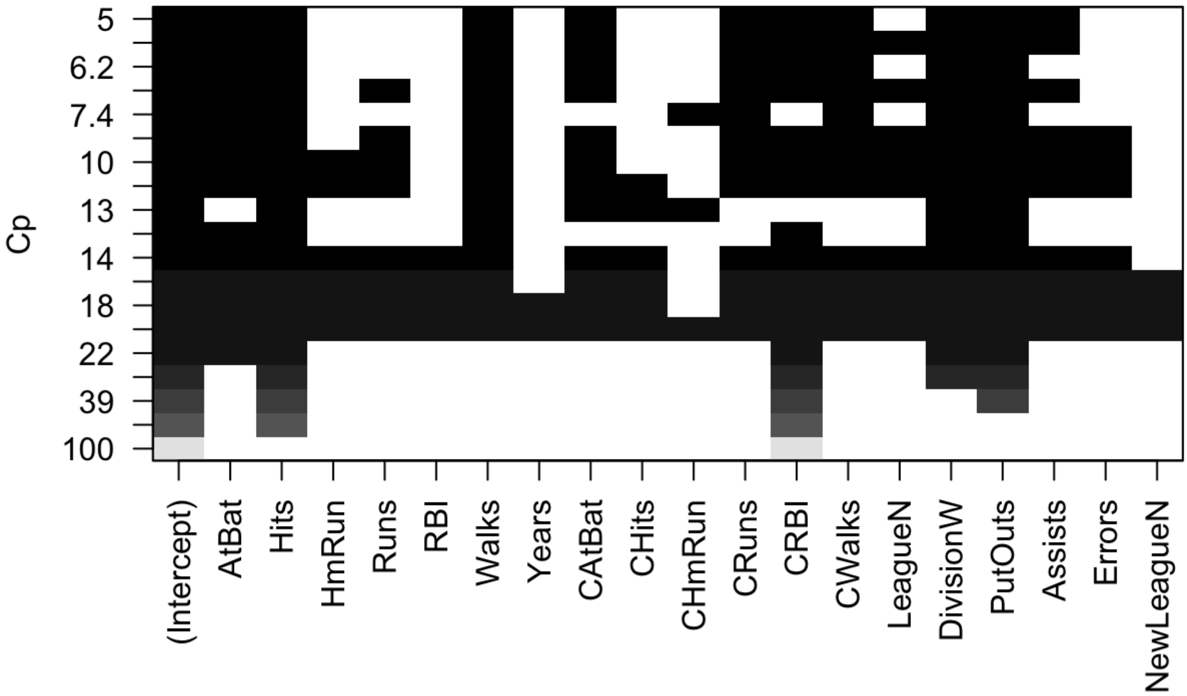
Pre

```
library(ISLR)
library(leaps)

summary(Hitters)
```

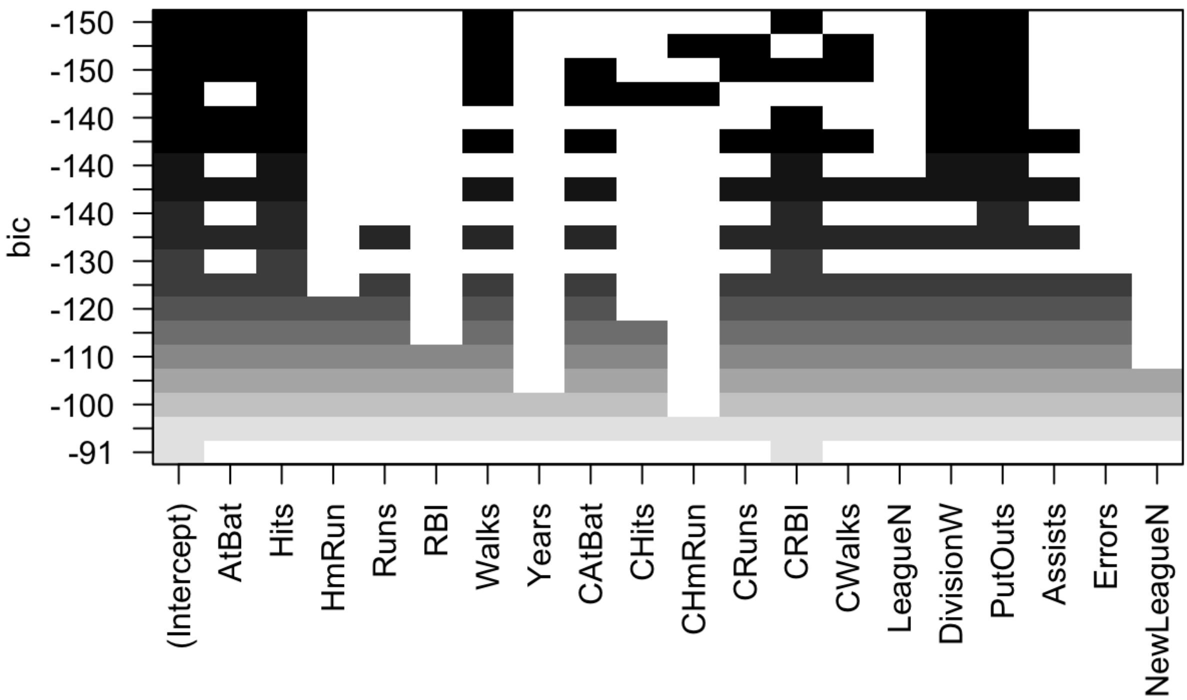
```
     AtBat            Hits          HmRun            Runs       
 Min.   : 16.0   Min.   :  1   Min.   : 0.00   Min.   :  0.00  
 1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25  
 Median :379.5   Median : 96   Median : 8.00   Median : 48.00  
 Mean   :380.9   Mean   :101   Mean   :10.77   Mean   : 50.91  
 3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00  
 Max.   :687.0   Max.   :238   Max.   :40.00   Max.   :130.00  
                                                               
      RBI             Walks            Years            CAtBat       
 Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0  
 1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.:  816.8  
 Median : 44.00   Median : 35.00   Median : 6.000   Median : 1928.0  
 Mean   : 48.03   Mean   : 38.74   Mean   : 7.444   Mean   : 2648.7  
 3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.: 3924.2  
 Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0  
                                                                     
     CHits            CHmRun           CRuns             CRBI        
 Min.   :   4.0   Min.   :  0.00   Min.   :   1.0   Min.   :   0.00  
 1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.:  88.75  
 Median : 508.0   Median : 37.50   Median : 247.0   Median : 220.50  
 Mean   : 717.6   Mean   : 69.49   Mean   : 358.8   Mean   : 330.12  
 3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.: 426.25  
 Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.00  
                                                                     
     CWalks          League  Division    PutOuts          Assists     
 Min.   :   0.00   A:175   E:157    Min.   :   0.0   Min.   :  0.0  
 1st Qu.:  67.25   N:147   W:165    1st Qu.: 109.2   1st Qu.:  7.0  
 Median : 170.50                    Median : 212.0   Median : 39.5  
 Mean   : 260.24                    Mean   : 288.9   Mean   :106.9  
 3rd Qu.: 339.25                    3rd Qu.: 325.0   3rd Qu.:166.0  
 Max.   :1566.00                    Max.   :1378.0   Max.   :492.0  
                                                                    
     Errors          Salary       NewLeague
 Min.   : 0.00   Min.   :  67.5   A:176    
 1st Qu.: 3.00   1st Qu.: 190.0   N:146    
 Median : 6.00   Median : 425.0            
 Mean   : 8.04   Mean   : 535.9            
 3rd Qu.:11.00   3rd Qu.: 750.0            
 Max.   :32.00   Max.   :2460.0            
                 NA's   :59                
```

Best Subset Selection

```r
# Fit
# Showing only best (nbest = 1)
regfit.models <- regsubsets(Salary~., data = Hitters, nbest = 1, nvmax = ncol(Hitters))

# Summary, Cp, BIC
res.sum <- summary(regfit.models)
# as.data.frame(res.sum$outmat)

plot(regfit.models, scale='Cp')
```




```r
plot(regfit.models, scale='bic')
```




```
Best Cp (5.01) model: 10
Model: Salary ~ (Intercept) + AtBat + Hits + Walks + CAtBat + CRuns + CRBI + CWalks + DivisionW
+ PutOuts + Assists

Best BIC (-147.92) model: 6
Model: Salary ~ (Intercept) + AtBat + Hits + Walks + CRBI + DivisionW + PutOuts
```

Forward Stepwise Selection

```
regfit.fwd = regsubsets(Salary~., data=Hitters, nvmax=ncol(Hitters), method ="forward")

regfit.fwd.sum = summary(regfit.fwd)

regfit.fwd.sum.min.bic = which.min(regfit.fwd.sum$bic)
regfit.fwd.sum.min.bic.value = min(regfit.fwd.sum$bic)

regfit.fwd.sum.min.cp = which.min(regfit.fwd.sum$cp)
regfit.fwd.sum.min.cp.value = min(regfit.fwd.sum$cp)
```

```
Best Cp (5.01) model: 10
Model: Salary ~ (Intercept) + AtBat + Hits + Walks + CAtBat + CRuns + CRBI + CWalks + DivisionW
+ PutOuts + Assists

Best BIC (-147.92) model: 6
Model: Salary ~ (Intercept) + AtBat + Hits + Walks + CRBI + DivisionW + PutOuts
```

Backward Stepwise Selection

```
regfit.bfd = regsubsets(Salary~., data=Hitters, nvmax=ncol(Hitters), method ="backward")
regfit.bfd.sum = summary(regfit.bfd)

regfit.bfd.sum.min.bic = which.min(regfit.bfd.sum$bic)
regfit.bfd.sum.min.bic.value = min(regfit.bfd.sum$bic)

regfit.bfd.sum.min.cp = which.min(regfit.bfd.sum$cp)
regfit.bfd.sum.min.cp.value = min(regfit.bfd.sum$cp)
```

```
Best Cp (5.01) model: 10
Model: Salary ~ (Intercept) + AtBat + Hits + Walks + CAtBat + CRuns + CRBI + CWalks + DivisionW
+ PutOuts + Assists

Best BIC (-147.38) model: 8
Model: Salary ~ (Intercept) + AtBat + Hits + Walks + CRuns + CRBI + CWalks + DivisionW + PutOut
s
```

Exercise 1. Conclusion

Coeffs and their Significance (Sorted by P-value)

term	estimate	std.error	statistic	p.value
PutOuts	0.282	0.077	3.640	0.000
Walks	6.231	1.829	3.408	0.001
Hits	7.501	2.378	3.155	0.002
AtBat	-1.980	0.634	-3.123	0.002
DivisionW	-116.849	40.367	-2.895	0.004
CWalks	-0.812	0.328	-2.474	0.014
CRuns	1.454	0.750	1.938	0.054
(Intercept)	163.104	90.779	1.797	0.074
Assists	0.371	0.221	1.678	0.095
CAtBat	-0.171	0.135	-1.267	0.206
CRBI	0.808	0.693	1.166	0.245
Runs	-2.376	2.981	-0.797	0.426
LeagueN	62.599	79.261	0.790	0.430
Errors	-3.361	4.392	-0.765	0.445
HmRun	4.331	6.201	0.698	0.486
RBI	-1.045	2.601	-0.402	0.688
NewLeagueN	-24.762	79.003	-0.313	0.754
Years	-3.489	12.412	-0.281	0.779
CHits	0.134	0.675	0.199	0.843
CHmRun	-0.173	1.617	-0.107	0.915

Best Subset Selection and Forward Stepwise Selection perform simillary and they give identical results. However selecting model based on BIC criteria Backward Stepwise Selection performs best with BIC value of -147.38 compared -147.92 when using Best Subset and Stepwise Forward methods. When using Cp-value as a comparison all methods gave identical models: Salary ~ (Intercept) + AtBat + Hits + Walks + CAtBat + CRuns + CRBI + CWalks + DivisionW + PutOuts + Assists with Cp-value of 5.01.

Using p-values of the full model and just simply counting and concluding subset, forward and backwards results the most important variables are: PutOuts, Walks, Hits, AtBat, DivisionW, Cwalks, CRuns and AtBat. For improving model we could add interaction terms and perform model selection again.

2

Pre

```
Loading required package: lattice
```

```
Loading required package: ggplot2
```

```


Attaching package: 'pls'
```

```
The following object is masked from 'package:caret':

    R2
```

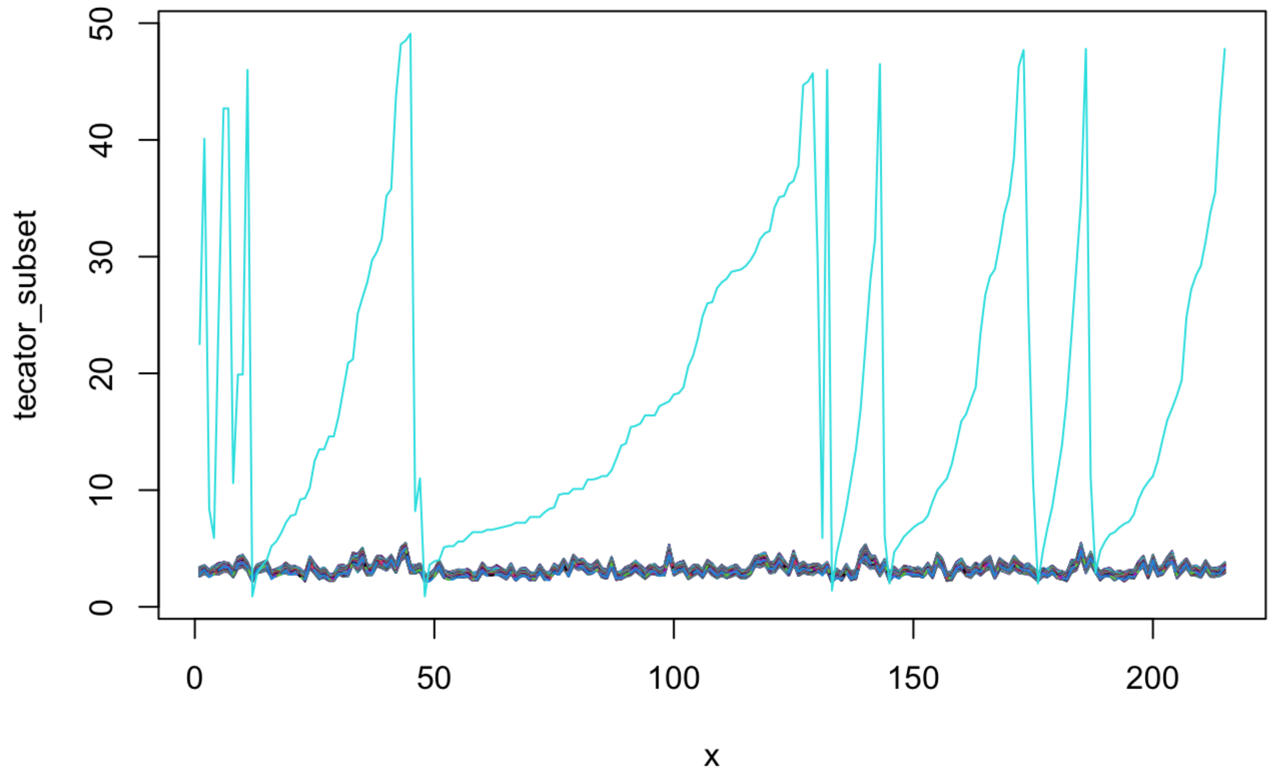
```
The following object is masked from 'package:stats':

    loadings
```

```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
✓ tibble  3.0.5     ✓ dplyr   1.0.3
✓ tidyr   1.1.2     ✓ stringr 1.4.0
✓ readr   1.4.0     ✓ forcats 0.5.1
✓ purrr   0.3.4
```

```
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
x purrr::lift()   masks caret::lift()
x dplyr::select() masks MASS::select()
```

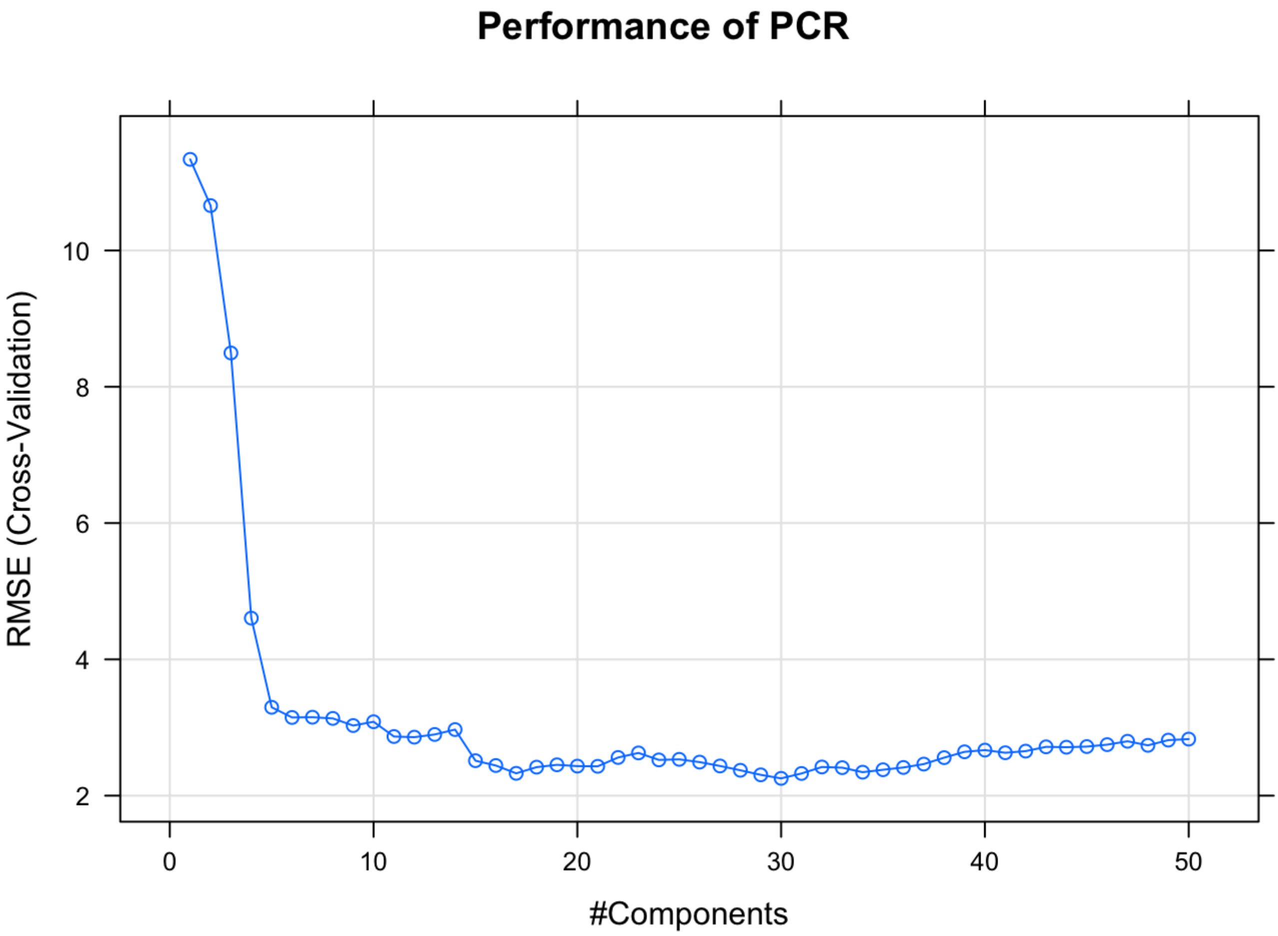
```
tecator = read.csv("/Users/tuuba/code/ISRL/Lab2/tecator.csv")
tecator_subset = tecator[ , !names(tecator) %in% c('Sample', 'Protein', 'Moisture')]

set.seed(2707)
smp_size <- floor(0.75 * nrow(tecator_subset))
train_ind <- sample(seq_len(nrow(tecator_subset)), size = smp_size)
tecator_train = tecator_subset[train_ind, ]
tecator_test = tecator_subset[-train_ind, ]
x = 1:215
matplot(x = x, y=tecator_subset, type = 'l')
```



> Conclusion:

The straight lines indicate good linear relationships so on PCR, PLS, Lasso and Ridge ression could be suitable to model the data.

```
model_pcr <- train(
  Fat~., data = tecator_train, method = "pcr",
  scale = TRUE, # scale = TRUE for standardizing the variables to make them comparable.
  trControl = trainControl("cv", number = 25),
  tuneLength = 50
  )
# Plot model RMSE vs different values of components
plot(model_pcr, main = 'Performance of PCR')
```

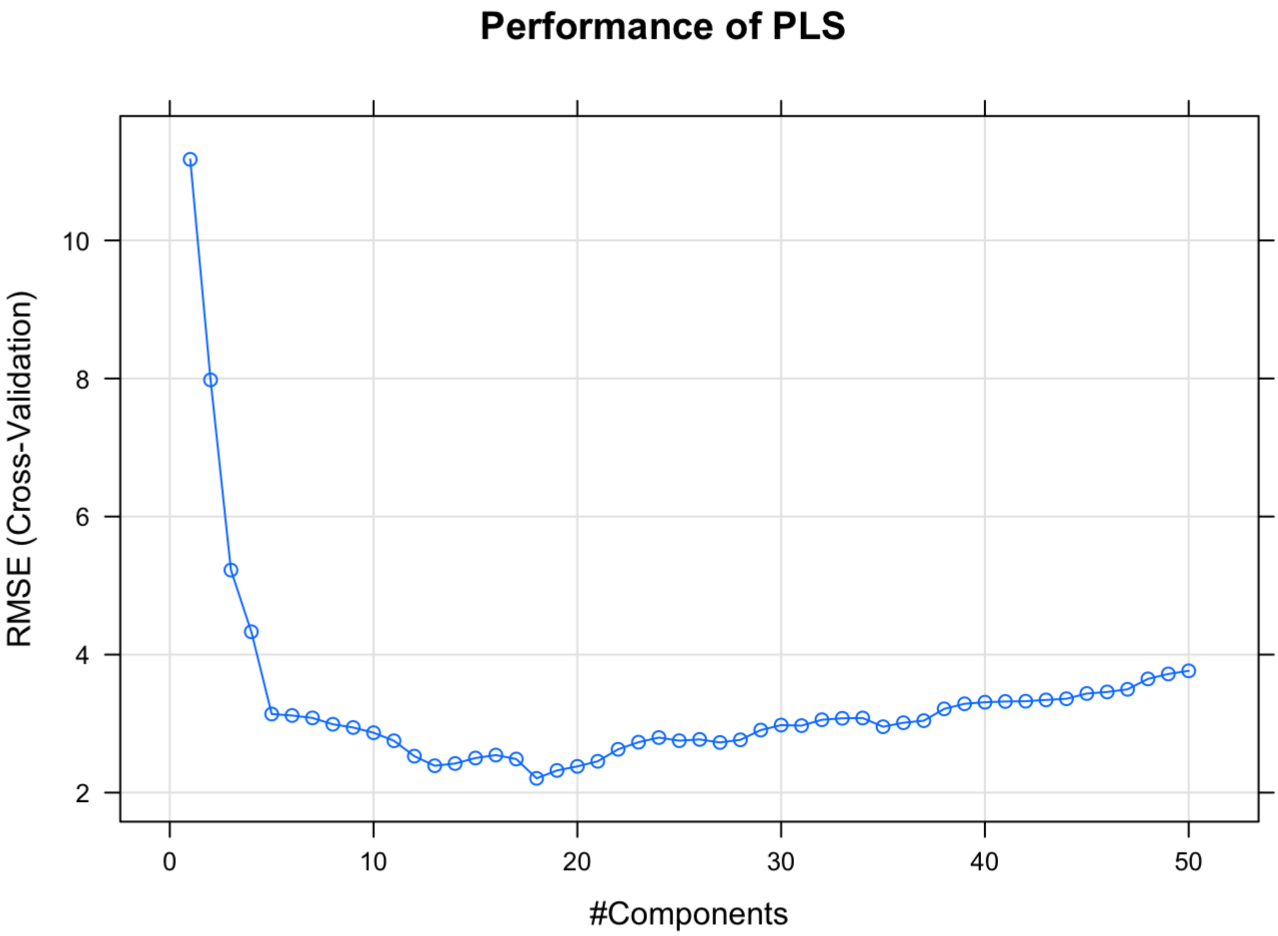



```r
# Print the best tuning parameter ncomp that
# minimize the cross-validation error, RMSE
model_pcr$bestTune
```

	ncomp <dbl>
30	30

1 row

```r
model_pls <- train(
  Fat~., data = tecator_train, method = "pls",
  scale = TRUE, # scale = TRUE for standardizing the variables to make them comparable.
  trControl = trainControl("cv", number = 25),
  tuneLength = 50
  )
# Plot model RMSE vs different values of components
plot(model_pls, main = 'Performance of PLS')
```



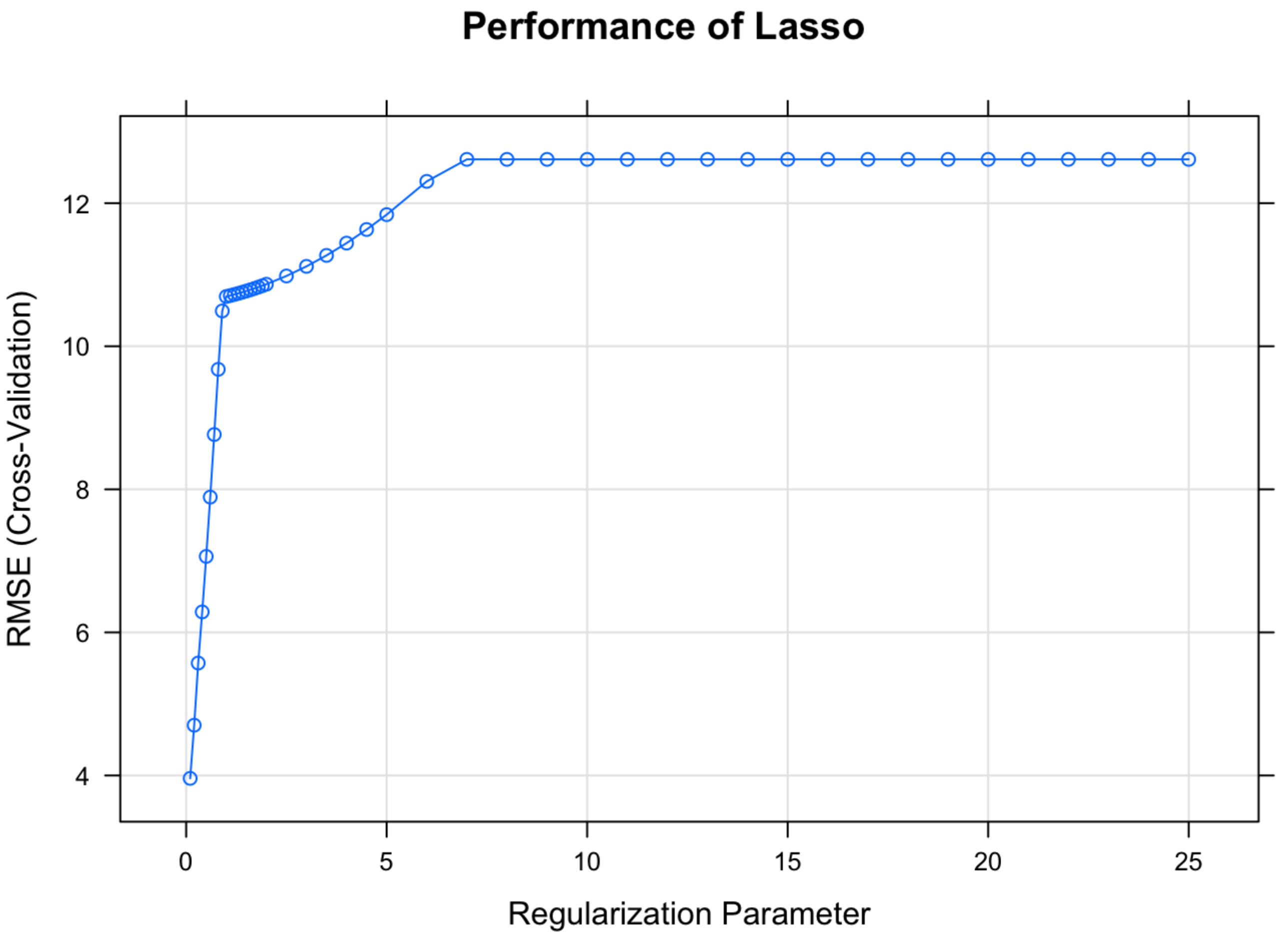

```
# Print the best tuning parameter ncomp that
# minimize the cross-validation error, RMSE
model_pls$bestTune
```

	ncomp <dbl>
18	18

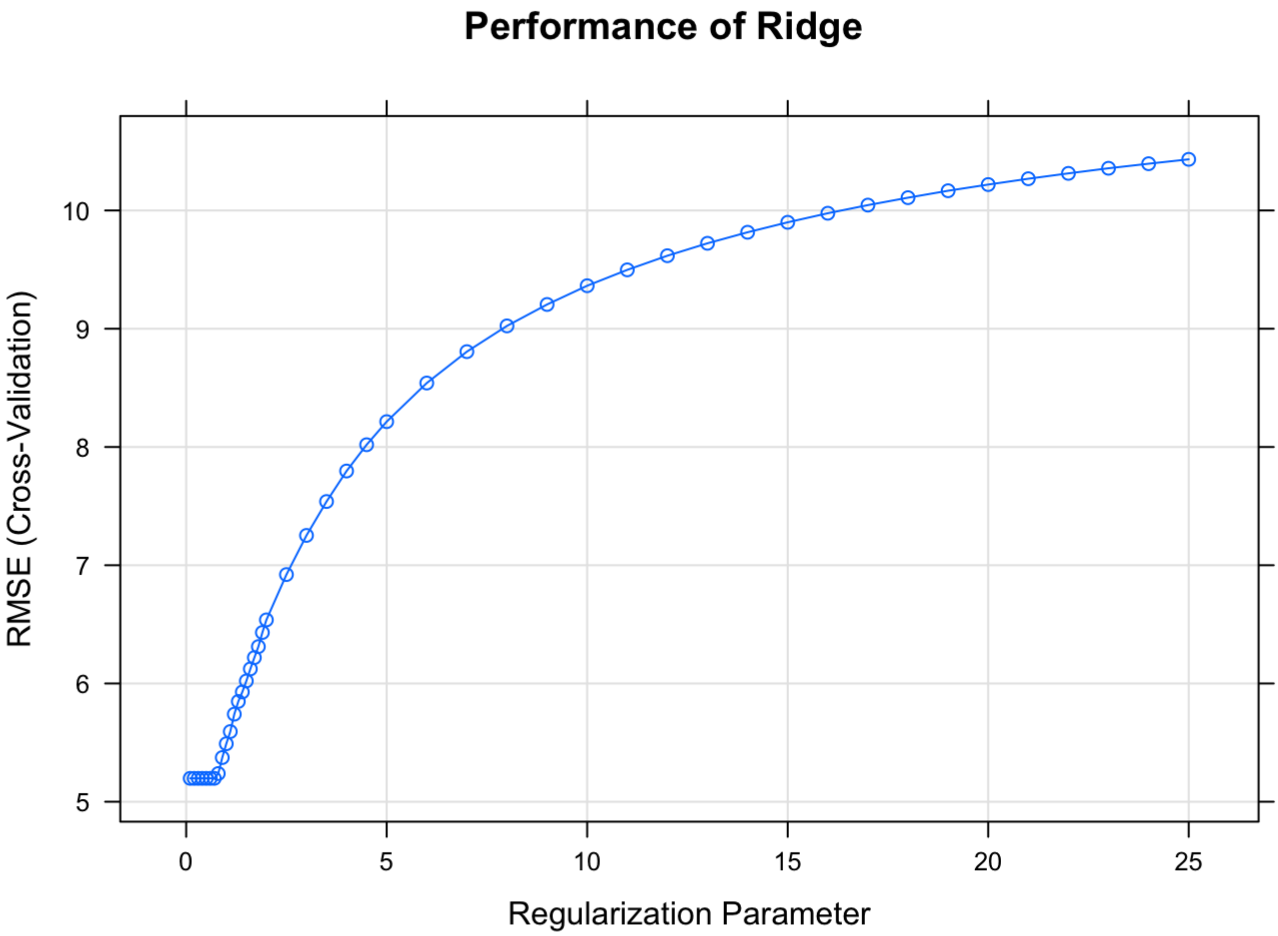
1 row

```
# Predictions

# Make predictions
pcr_predictions <- model_pcr %>% predict(tecator_test)
pls_predictions <- model_pls %>% predict(tecator_test)
# Model performance metrics
pcr_mse = caret::RMSE(pcr_predictions, tecator_test$Fat)**2
pls_mse = caret::RMSE(pls_predictions, tecator_test$Fat)**2
```

```r
# Regularization parameters
parameters <- c(seq(0.1, 2, by =0.1) ,  seq(2, 5, 0.5) , seq(5, 25, 1))

model_lasso <- train(
  Fat~., data = tecator_train, method = "glmnet",
  scale = TRUE, # scale = TRUE for standardizing the variables to make them comparable.
  trControl = trainControl("cv", number = 25),
  tuneGrid = expand.grid(alpha = 1,  lambda = parameters),
  metric = 'RMSE'
  )

model_ridge <- train(
  Fat~., data = tecator_train, method = "glmnet",
  scale = TRUE, # scale = TRUE for standardizing the variables to make them comparable.
  trControl = trainControl("cv", number = 25),
  tuneGrid = expand.grid(alpha = 0,  lambda = parameters),
  metric = 'RMSE'
  )

plot(model_lasso, main = 'Performance of Lasso')
```




```r
plot(model_ridge, main = 'Performance of Ridge')
```




```
# Predictions
lasso_predictions <- model_lasso %>% predict(tecator_test)
ridge_predictions <- model_ridge %>% predict(tecator_test)

# Model performance metrics
lasso_mse = caret::RMSE(lasso_predictions, tecator_test$Fat)**2
ridge_mse = caret::RMSE(ridge_predictions, tecator_test$Fat)**2
```

```
PCR MSE: 4.70
PLS MSE: 3.92
Lasso MSE: 12.42
Ridge MSE: 17.30
```

PLS has the best performance with MSE of 3.92, then comes PCR (MSE = 4.70), Lasso (MSE = 12.42) and lastly Ridge (MSE = 17.30).