```r
# pcr.r

# install.packages("pls")
library(pls)  # needed?

library(dplyr)
library(caret)   # createDataPartition(), train()
library(MASS)    # Boston dataset
dim(Boston)
```

```
## [1] 506  14
```

```r
#
# 20% test sets
set.seed(123)
training.samples <- Boston$medv %>% createDataPartition(p = 0.8, list = FALSE)
head(training.samples)
```

```
##      Resample1
## [1,]         1
## [2,]         2
## [3,]         4
## [4,]         5
## [5,]         7
## [6,]         8
```

```r
train.data  <- Boston[training.samples, ]
test.data <- Boston[-training.samples, ]
dim(train.data)
```

```
## [1] 407  14
```

```r
dim(test.data)
```

```
## [1] 99 14
```

```r
head(test.data)
```

```
##       crim   zn indus chas   nox    rm   age    dis rad tax ptratio  black
## 3  0.02729  0.0  7.07    0 0.469 7.185  61.1 4.9671   2 242    17.8 392.83
## 6  0.02985  0.0  2.18    0 0.458 6.430  58.7 6.0622   3 222    18.7 394.12
## 9  0.21124 12.5  7.87    0 0.524 5.631 100.0 6.0821   5 311    15.2 386.63
## 11 0.22489 12.5  7.87    0 0.524 6.377  94.3 6.3467   5 311    15.2 392.52
## 14 0.62976  0.0  8.14    0 0.538 5.949  61.8 4.7075   4 307    21.0 396.90
## 15 0.63796  0.0  8.14    0 0.538 6.096  84.5 4.4619   4 307    21.0 380.02
##    lstat medv
## 3   4.03 34.7
## 6   5.21 28.7
## 9  29.93 16.5
## 11 20.45 15.0
## 14  8.26 20.4
## 15 10.26 18.2
```

```r
# pcr
#
# fit 10 models
#
set.seed(1)
```

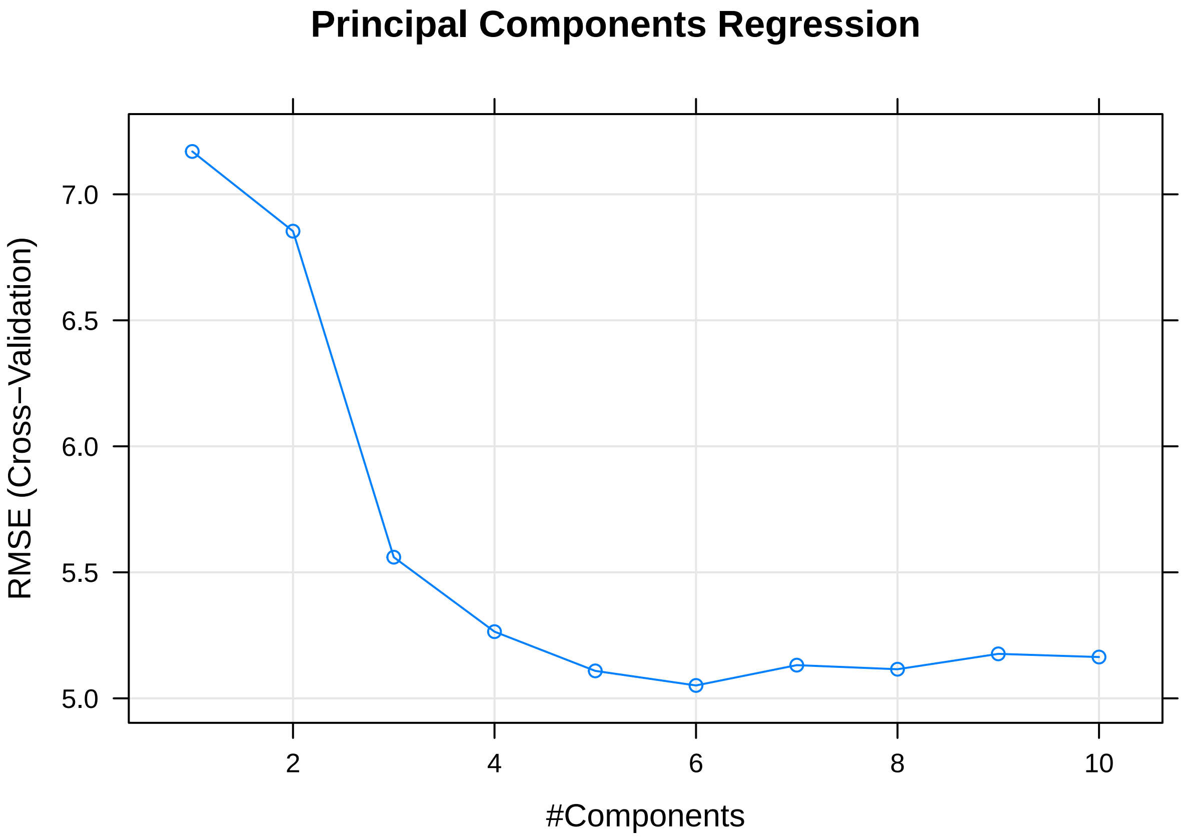
```r
model <- train(medv~., data = train.data,
               method = "pcr",
               scale = TRUE,
               trControl = trainControl("cv", number = 10),
               tuneLength = 10)
str(model)
```

```
## List of 23
##  $ method      : chr "pcr"
##  $ modelInfo   :List of 12
##   ..$ label     : chr "Principal Component Analysis"
##   ..$ library   : chr "pls"
##   ..$ type      : chr "Regression"
##   ..$ parameters:'data.frame':   1 obs. of  3 variables:
##   .. ..$ parameter: Factor w/ 1 level "ncomp": 1
##   .. ..$ class    : Factor w/ 1 level "numeric": 1
##   .. ..$ label    : Factor w/ 1 level "#Components": 1
##   ..$ grid      :function (x, y, len = NULL, search = "grid")
##   .. ..- attr(*, "srcref")= 'srcref' int [1:8] 7 26 14 19 26 19 7 14
##   .. .. ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x7ff6c5384a40>
##   ..$ loop      :function (grid)
##   .. ..- attr(*, "srcref")= 'srcref' int [1:8] 15 26 20 19 26 19 15 20
##   .. .. ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x7ff6c5384a40>
##   ..$ fit       :function (x, y, wts, param, lev, last, classProbs, ...)
##   .. ..- attr(*, "srcref")= 'srcref' int [1:8] 21 25 26 19 25 19 21 26
##   .. .. ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x7ff6c5384a40>
##   ..$ predict   :function (modelFit, newdata, submodels = NULL)
##   .. ..- attr(*, "srcref")= 'srcref' int [1:8] 27 29 37 19 29 19 27 37
##   .. .. ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x7ff6c5384a40>
##   ..$ predictors:function (x, ...)
##   .. ..- attr(*, "srcref")= 'srcref' int [1:8] 38 32 38 70 32 70 38 38
##   .. .. ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x7ff6c5384a40>
##   ..$ tags      : chr [1:2] "Linear Regression" "Feature Extraction"
##   ..$ prob      : NULL
##   ..$ sort      :function (x)
##   .. ..- attr(*, "srcref")= 'srcref' int [1:8] 41 26 41 53 26 53 41 41
##   .. .. ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environment: 0x7ff6c5384a40>
##  $ modelType   : chr "Regression"
##  $ results     :'data.frame':    10 obs. of  7 variables:
##   ..$ ncomp     : num [1:10] 1 2 3 4 5 6 7 8 9 10
##   ..$ RMSE      : num [1:10] 7.17 6.85 5.56 5.26 5.11 ...
##   ..$ Rsquared  : num [1:10] 0.389 0.449 0.622 0.661 0.689 ...
##   ..$ MAE       : num [1:10] 4.99 4.98 3.84 3.54 3.42 ...
##   ..$ RMSESD    : num [1:10] 1.15 1.11 1.14 1.19 1.13 ...
##   ..$ RsquaredSD: num [1:10] 0.135 0.14 0.163 0.169 0.153 ...
##   ..$ MAESD     : num [1:10] 0.59 0.602 0.489 0.448 0.449 ...
##  $ pred        : NULL
##  $ bestTune    :'data.frame':    1 obs. of  1 variable:
##   ..$ ncomp: num 6
##  $ call        : language train.formula(form = medv ~ ., data = train.data, method = "pcr", scale = T
##  $ dots        :List of 1
##   ..$ scale: logi TRUE
##  $ metric      : chr "RMSE"
##  $ control     :List of 28
##   ..$ method           : chr "cv"
##   ..$ number           : num 10
##   ..$ repeats          : logi NA
##   ..$ search           : chr "grid"
##   ..$ p                : num 0.75
##   ..$ initialWindow    : NULL
##   ..$ horizon          : num 1
##   ..$ fixedWindow      : logi TRUE
##   ..$ skip             : num 0
##   ..$ verboseIter      : logi FALSE
##   ..$ returnData       : logi TRUE
##   ..$ returnResamp     : chr "final"
##   ..$ savePredictions  : chr "none"
##   ..$ classProbs       : logi FALSE
##   ..$ summaryFunction  :function (data, lev = NULL, model = NULL)
##   ..$ selectionFunction: chr "best"
##   ..$ preProcOptions   :List of 6
##   .. ..$ thresh   : num 0.95
##   .. ..$ ICAcomp  : num 3
##   .. ..$ k        : num 5
##   .. ..$ freqCut  : num 19
##   .. ..$ uniqueCut: num 10
##   .. ..$ cutoff   : num 0.9
##   ..$ sampling         : NULL
##   ..$ index            :List of 10
##   .. ..$ Fold01: int [1:366] 1 4 5 6 7 8 9 10 11 12 ...
##   .. ..$ Fold02: int [1:366] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ Fold03: int [1:366] 2 3 4 5 6 8 9 10 13 14 ...
##   .. ..$ Fold04: int [1:366] 1 2 3 5 6 7 8 9 11 12 ...
##   .. ..$ Fold05: int [1:367] 1 2 3 4 6 7 9 10 11 12 ...
##   .. ..$ Fold06: int [1:365] 1 2 3 4 5 7 8 9 10 11 ...
##   .. ..$ Fold07: int [1:367] 1 2 3 4 5 6 7 8 10 11 ...
##   .. ..$ Fold08: int [1:366] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ Fold09: int [1:367] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ Fold10: int [1:367] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ indexOut         :List of 10
##   .. ..$ Resample01: int [1:41] 2 3 22 27 39 44 57 61 72 86 ...
##   .. ..$ Resample02: int [1:41] 54 64 98 110 111 131 139 145 156 167 ...
##   .. ..$ Resample03: int [1:41] 1 7 11 12 36 38 45 52 85 99 ...
##   .. ..$ Resample04: int [1:41] 4 10 31 41 42 47 51 55 68 69 ...
##   .. ..$ Resample05: int [1:40] 5 8 16 50 63 70 71 73 77 79 ...
##   .. ..$ Resample06: int [1:42] 6 13 28 37 53 65 78 83 92 93 ...
##   .. ..$ Resample07: int [1:40] 9 19 21 24 25 30 75 81 88 101 ...
##   .. ..$ Resample08: int [1:41] 14 17 18 20 23 29 32 35 46 58 ...
##   .. ..$ Resample09: int [1:40] 15 33 49 76 82 84 90 95 102 120 ...
##   .. ..$ Resample10: int [1:40] 26 34 40 43 48 56 59 60 62 66 ...
##   ..$ indexFinal       : NULL
##   ..$ timingSamps      : num 0
##   ..$ predictionBounds : logi [1:2] FALSE FALSE
##   ..$ seeds            :List of 11
##   .. ..$ : int 452737
##   .. ..$ : int 124413
##   .. ..$ : int 436523
##   .. ..$ : int 856018
##   .. ..$ : int 666931
##   .. ..$ : int 25173
##   .. ..$ : int 294762
##   .. ..$ : int 392686
##   .. ..$ : int 640775
##   .. ..$ : int 538191
##   .. ..$ : int 270373
##   ..$ adaptive         :List of 4
##   .. ..$ min     : num 5
##   .. ..$ alpha   : num 0.05
##   .. ..$ method  : chr "gls"
##   .. ..$ complete: logi TRUE
##   ..$ trim             : logi FALSE
##   ..$ allowParallel    : logi TRUE
##   ..$ yLimits          : num [1:2] 2.75 52.25
##  $ finalModel  :List of 23
##   ..$ coefficients : num [1:13, 1, 1:6] -0.5735 0.567 -0.7786 0.0146 -0.7678 ...
##   .. ..- attr(*, "dimnames")=List of 3
##   .. .. ..$ : chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. .. ..$ : chr ".outcome"
##   .. .. ..$ : chr [1:6] "1 comps" "2 comps" "3 comps" "4 comps" ...
##   ..$ scores       : 'scores' num [1:407, 1:6] -2.13 -1.47 -2.63 -2.47 -1.39 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:407] "X1" "X2" "X4" "X5" ...
##   .. .. ..$ : chr [1:6] "Comp 1" "Comp 2" "Comp 3" "Comp 4" ...
##   ..$ loadings     : 'loadings' num [1:13, 1:6] 0.25424 -0.25133 0.34515 -0.00649 0.34034 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. .. ..$ : chr [1:6] "Comp 1" "Comp 2" "Comp 3" "Comp 4" ...
##   ..$ Yloadings    : 'loadings' num [1, 1:6] -2.26 1.89 3.65 2.11 -1.21 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr ".outcome"
##   .. .. ..$ : chr [1:6] "Comp 1" "Comp 2" "Comp 3" "Comp 4" ...
##   ..$ projection   : num [1:13, 1:6] 0.25424 -0.25133 0.34515 -0.00649 0.34034 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. .. ..$ : chr [1:6] "Comp 1" "Comp 2" "Comp 3" "Comp 4" ...
##   ..$ Xmeans       : Named num [1:13] 0.435 0.488 1.623 0.282 4.806 ...
##   .. ..- attr(*, "names")= chr [1:13] "crim" "zn" "indus" "chas" ...
##   ..$ Ymeans       : Named num 22.5
##   .. ..- attr(*, "names")= chr ".outcome"
##   ..$ fitted.values: num [1:407, 1, 1:6] 27.3 25.8 28.4 28.1 25.7 ...
##   .. ..- attr(*, "dimnames")=List of 3
##   .. .. ..$ : chr [1:407] "X1" "X2" "X4" "X5" ...
##   .. .. ..$ : chr ".outcome"
##   .. .. ..$ : chr [1:6] "1 comps" "2 comps" "3 comps" "4 comps" ...
##   ..$ residuals    : num [1:407, 1, 1:6] -3.31 -4.24 4.96 8.11 -2.75 ...
##   .. ..- attr(*, "dimnames")=List of 3
##   .. .. ..$ : chr [1:407] "X1" "X2" "X4" "X5" ...
##   .. .. ..$ : chr ".outcome"
##   .. .. ..$ : chr [1:6] "1 comps" "2 comps" "3 comps" "4 comps" ...
##   ..$ Xvar         : Named num [1:6] 2535 570 483 360 331 ...
##   .. ..- attr(*, "names")= chr [1:6] "Comp 1" "Comp 2" "Comp 3" "Comp 4" ...
##   ..$ Xtotvar      : num 5278
##   ..$ fit.time     : Named num 0.001
##   .. ..- attr(*, "names")= chr "elapsed"
##   ..$ ncomp        : num 6
##   ..$ method       : chr "svdpc"
##   ..$ scale        : Named num [1:13] 8.264 22.715 6.809 0.262 0.116 ...
##   .. ..- attr(*, "names")= chr [1:13] "crim" "zn" "indus" "chas" ...
##   ..$ call         : language pcr(formula = .outcome ~ ., ncomp = ncomp, data = dat, scale = TRUE)
##   ..$ terms        :Classes 'terms', 'formula'  language .outcome ~ crim + zn + indus + chas + nox +
##   .. .. ..- attr(*, "variables")= language list(.outcome, crim, zn, indus, chas, nox, rm, age, dis, r
##   .. .. ..- attr(*, "factors")= int [1:14, 1:13] 0 1 0 0 0 0 0 0 0 0 ...
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:14] ".outcome" "crim" "zn" "indus" ...
##   .. .. .. .. ..$ : chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. .. ..- attr(*, "term.labels")= chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. .. ..- attr(*, "order")= int [1:13] 1 1 1 1 1 1 1 1 1 1 ...
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: 0x7ff6b0a1b440>
##   .. .. ..- attr(*, "predvars")= language list(.outcome, crim, zn, indus, chas, nox, rm, age, dis, ra
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:14] "numeric" "numeric" "numeric" "numeric" ...
##   .. .. .. ..- attr(*, "names")= chr [1:14] ".outcome" "crim" "zn" "indus" ...
##   ..$ model        :'data.frame':    407 obs. of  14 variables:
##   .. ..$ .outcome: num [1:407] 24 21.6 33.4 36.2 22.9 27.1 18.9 18.9 21.7 19.9 ...
##   .. ..$ crim    : num [1:407] 0.00632 0.02731 0.03237 0.06905 0.08829 ...
##   .. ..$ zn      : num [1:407] 18 0 0 0 12.5 12.5 12.5 12.5 12.5 0 ...
##   .. ..$ indus   : num [1:407] 2.31 7.07 2.18 2.18 7.87 7.87 7.87 7.87 7.87 8.14 ...
##   .. ..$ chas    : num [1:407] 0 0 0 0 0 0 0 0 0 0 ...
##   .. ..$ nox     : num [1:407] 0.538 0.469 0.458 0.458 0.524 0.524 0.524 0.524 0.524 0.538 ...
##   .. ..$ rm      : num [1:407] 6.58 6.42 7 7.15 6.01 ...
##   .. ..$ age     : num [1:407] 65.2 78.9 45.8 54.2 66.6 96.1 85.9 82.9 39 56.5 ...
##   .. ..$ dis     : num [1:407] 4.09 4.97 6.06 6.06 5.56 ...
##   .. ..$ rad     : num [1:407] 1 2 3 3 5 5 5 5 5 4 ...
##   .. ..$ tax     : num [1:407] 296 242 222 222 311 311 311 311 311 307 ...
##   .. ..$ ptratio : num [1:407] 15.3 17.8 18.7 18.7 15.2 15.2 15.2 15.2 15.2 21 ...
##   .. ..$ black   : num [1:407] 397 397 395 397 396 ...
##   .. ..$ lstat   : num [1:407] 4.98 9.14 2.94 5.33 12.43 ...
##   .. ..- attr(*, "terms")=Classes 'terms', 'formula'  language .outcome ~ crim + zn + indus + chas +
##   .. .. .. ..- attr(*, "variables")= language list(.outcome, crim, zn, indus, chas, nox, rm, age, dis
##   .. .. .. ..- attr(*, "factors")= int [1:14, 1:13] 0 1 0 0 0 0 0 0 0 0 ...
##   .. .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. .. ..$ : chr [1:14] ".outcome" "crim" "zn" "indus" ...
##   .. .. .. .. .. ..$ : chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. .. .. ..- attr(*, "term.labels")= chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. .. .. ..- attr(*, "order")= int [1:13] 1 1 1 1 1 1 1 1 1 1 ...
##   .. .. .. ..- attr(*, "intercept")= int 1
##   .. .. .. ..- attr(*, "response")= int 1
##   .. .. .. ..- attr(*, ".Environment")=<environment: 0x7ff6b0a1b440>
##   .. .. .. ..- attr(*, "predvars")= language list(.outcome, crim, zn, indus, chas, nox, rm, age, dis
##   .. .. .. ..- attr(*, "dataClasses")= Named chr [1:14] "numeric" "numeric" "numeric" "numeric" ...
##   .. .. .. .. ..- attr(*, "names")= chr [1:14] ".outcome" "crim" "zn" "indus" ...
##   ..$ xNames       : chr [1:13] "crim" "zn" "indus" "chas" ...
##   ..$ problemType  : chr "Regression"
##   ..$ tuneValue    :'data.frame':    1 obs. of  1 variable:
##   .. ..$ ncomp: num 6
##   ..$ obsLevels    : logi NA
##   ..$ param        :List of 1
##   .. ..$ scale: logi TRUE
##   ..- attr(*, "class")= chr "mvr"
##  $ preProcess  : NULL
##  $ trainingData:'data.frame':    407 obs. of  14 variables:
##   ..$ .outcome: num [1:407] 24 21.6 33.4 36.2 22.9 27.1 18.9 18.9 21.7 19.9 ...
##   ..$ crim    : num [1:407] 0.00632 0.02731 0.03237 0.06905 0.08829 ...
##   ..$ zn      : num [1:407] 18 0 0 0 12.5 12.5 12.5 12.5 12.5 0 ...
##   ..$ indus   : num [1:407] 2.31 7.07 2.18 2.18 7.87 7.87 7.87 7.87 7.87 8.14 ...
##   ..$ chas    : int [1:407] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ nox     : num [1:407] 0.538 0.469 0.458 0.458 0.524 0.524 0.524 0.524 0.524 0.538 ...
##   ..$ rm      : num [1:407] 6.58 6.42 7 7.15 6.01 ...
##   ..$ age     : num [1:407] 65.2 78.9 45.8 54.2 66.6 96.1 85.9 82.9 39 56.5 ...
##   ..$ dis     : num [1:407] 4.09 4.97 6.06 6.06 5.56 ...
##   ..$ rad     : int [1:407] 1 2 3 3 5 5 5 5 5 4 ...
##   ..$ tax     : num [1:407] 296 242 222 222 311 311 311 311 311 307 ...
##   ..$ ptratio : num [1:407] 15.3 17.8 18.7 18.7 15.2 15.2 15.2 15.2 15.2 21 ...
##   ..$ black   : num [1:407] 397 397 395 397 396 ...
##   ..$ lstat   : num [1:407] 4.98 9.14 2.94 5.33 12.43 ...
##  $ resample    :'data.frame':    10 obs. of  4 variables:
##   ..$ RMSE    : num [1:10] 4.17 4.36 4.8 5.15 6.01 ...
##   ..$ Rsquared: num [1:10] 0.807 0.698 0.79 0.662 0.58 ...
##   ..$ MAE     : num [1:10] 3.25 2.86 3.23 3.51 3.65 ...
##   ..$ Resample: chr [1:10] "Fold03" "Fold08" "Fold02" "Fold06" ...
##  $ resampledCM : NULL
##  $ perfNames   : chr [1:3] "RMSE" "Rsquared" "MAE"
##  $ maximize    : logi FALSE
##  $ yLimits     : num [1:2] 2.75 52.25
##  $ times       :List of 3
##   ..$ everything: 'proc_time' Named num [1:5] 1.218 0.045 1.273 0 0
##   .. ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...
##   ..$ final     : 'proc_time' Named num [1:5] 0.006 0.001 0.006 0 0
##   .. ..- attr(*, "names")= chr [1:5] "user.self" "sys.self" "elapsed" "user.child" ...
##   ..$ prediction: logi [1:3] NA NA NA
##  $ levels      : logi NA
##  $ terms       :Classes 'terms', 'formula'  language medv ~ crim + zn + indus + chas + nox + rm + age
##   .. ..- attr(*, "variables")= language list(medv, crim, zn, indus, chas, nox, rm, age, dis, rad, ta
##   .. ..- attr(*, "factors")= int [1:14, 1:13] 0 1 0 0 0 0 0 0 0 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:14] "medv" "crim" "zn" "indus" ...
##   .. .. .. ..$ : chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. ..- attr(*, "term.labels")= chr [1:13] "crim" "zn" "indus" "chas" ...
##   .. ..- attr(*, "order")= int [1:13] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(medv, crim, zn, indus, chas, nox, rm, age, dis, rad, tax
##   .. ..- attr(*, "dataClasses")= Named chr [1:14] "numeric" "numeric" "numeric" "numeric" ...
##   .. .. ..- attr(*, "names")= chr [1:14] "medv" "crim" "zn" "indus" ...
##  $ coefnames   : chr [1:13] "crim" "zn" "indus" "chas" ...
##  $ xlevels     : Named list()
##  - attr(*, "class")= chr [1:2] "train" "train.formula"
```

```
#
model$results
```

```
##    ncomp     RMSE  Rsquared      MAE    RMSESD RsquaredSD      MAESD
## 1      1 7.170043 0.3889250 4.987823 1.1458409  0.1352420 0.5901698
## 2      2 6.853749 0.4488643 4.984816 1.1099369  0.1401927 0.6023489
## 3      3 5.560047 0.6219351 3.837986 1.1413884  0.1632687 0.4889031
## 4      4 5.264562 0.6612251 3.536613 1.1938471  0.1686282 0.4484231
## 5      5 5.108941 0.6885868 3.415854 1.1334370  0.1534321 0.4491089
## 6      6 5.051134 0.6963419 3.366435 1.1530716  0.1531997 0.4614229
## 7      7 5.131647 0.6891018 3.399233 1.1319485  0.1548154 0.4460782
## 8      8 5.115450 0.6908804 3.404935 1.0950841  0.1491827 0.4304041
## 9      9 5.176459 0.6827258 3.448911 1.0937752  0.1521372 0.4133588
## 10    10 5.163884 0.6845580 3.512858 0.9806345  0.1350371 0.3864128
```

```
#
# plot train RMSE vs number of components
plot(model,main="Principal Components Regression")
```



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

```
##   ncomp
## 6     6
```

```
#
# variance per model
summary(model$finalModel)
```

```
## Data:    X dimension: 407 13
##  Y dimension: 407 1
## Fit method: svdpc
## Number of components considered: 6
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X           48.02    58.82    67.97    74.78    81.04    86.16
## .outcome    37.80    43.76    62.57    67.24    68.66    69.30
```

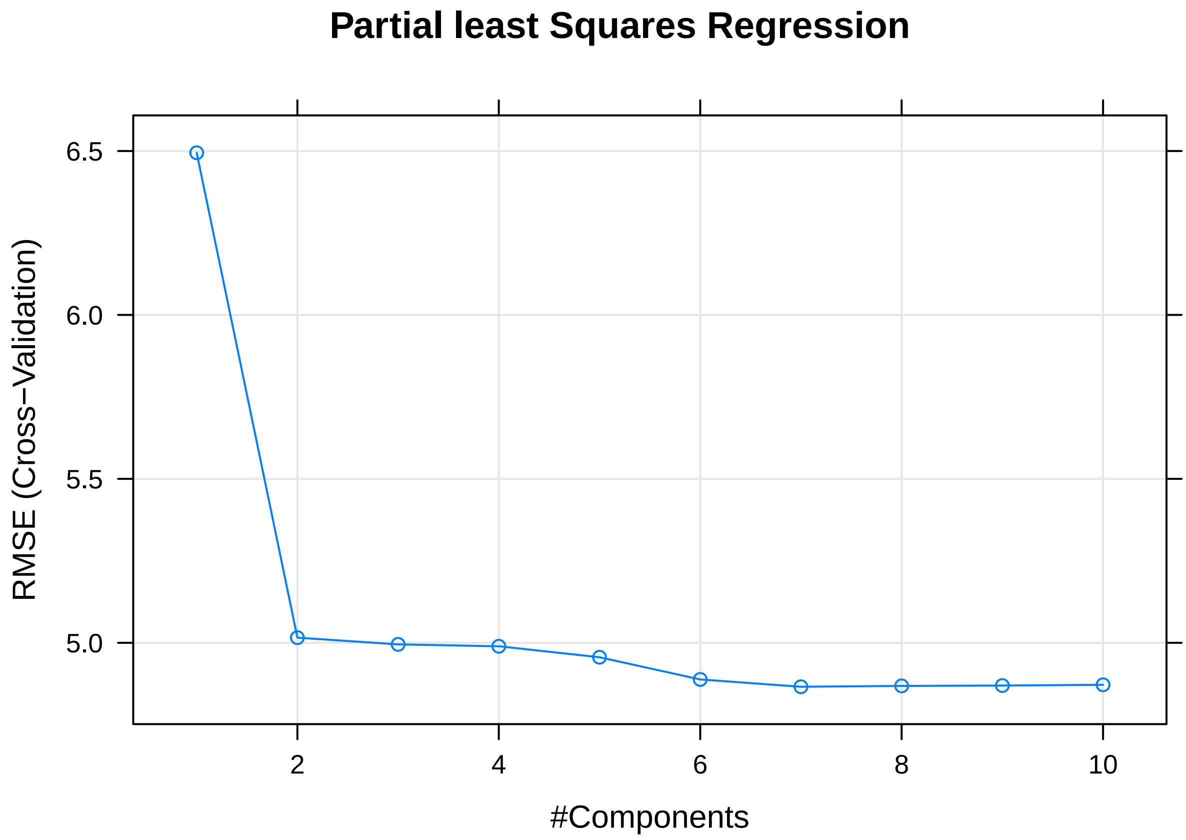
```
#
# variance explained in predictors (1st row) and in the response (2nd row)
#
# test performance
predictions = model %>% predict(test.data)
RMSE = caret::RMSE(predictions, test.data$medv)
Rsquare = caret::R2(predictions, test.data$medv)
d1 = data.frame(RMSE,Rsquare)
rownames(d1) = c('pcr')
d1
```

```
##         RMSE   Rsquare
## pcr 4.847172 0.7488739
```

```
#
# pls
#
set.seed(1)
model <- train(medv~., data = train.data,
               method = "pls",
               scale = TRUE,
               trControl = trainControl("cv", number = 10),
               tuneLength = 10)
model$results
```

```
##    ncomp     RMSE  Rsquared      MAE    RMSESD RsquaredSD     MAESD
## 1      1 6.494611 0.4955165 4.447078 1.1137875  0.1443515 0.5343725
## 2      2 5.015789 0.6980533 3.392457 1.1443205  0.1531122 0.4772334
## 3      3 4.995150 0.7025177 3.368314 1.0116818  0.1375136 0.3761320
## 4      4 4.989396 0.7059699 3.430505 0.9674060  0.1306264 0.4482073
## 5      5 4.955958 0.7070120 3.453991 0.8937295  0.1227171 0.4176809
## 6      6 4.888249 0.7145149 3.423969 0.9125166  0.1201442 0.4126241
## 7      7 4.865945 0.7167157 3.402248 0.9238277  0.1212114 0.4289319
## 8      8 4.868566 0.7165412 3.409289 0.9209121  0.1208916 0.4320559
## 9      9 4.869615 0.7164567 3.411307 0.9196800  0.1208792 0.4312978
## 10    10 4.871954 0.7161181 3.415144 0.9176257  0.1206587 0.4296808
```

```
#
# plot train RMSE vs number of components
plot(model,main="Partial least Squares Regression")
```



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

```
##   ncomp
## 7     7
```

```r
#
# variance per model
summary(model$finalModel)
```

```
## Data:    X dimension: 407 13
##  Y dimension: 407 1
## Fit method: oscorespls
## Number of components considered: 7
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           46.91    57.21    64.47    69.89    75.76    79.59    83.17
## .outcome    49.21    70.10    71.68    72.72    73.13    73.28    73.38
```

```r
#
# variance explained in predictors (1st row) and in the response (2nd row)
#
# test performance
predictions = model %>% predict(test.data)
RMSE = caret::RMSE(predictions, test.data$medv)
Rsquare = caret::R2(predictions, test.data$medv)
d2 = data.frame(RMSE,Rsquare)
rownames(d2) = c('pls')
```

```
#
# compare models
d3 = rbind(d1,d2)
d3
```

```
##         RMSE   Rsquare
## pcr 4.847172 0.7488739
## pls 4.581767 0.7620960
```