midterm project

Yixiao Sun

2024-03-19

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(summarytools)
```

```
##
## Attaching package: 'summarytools'
##
## The following object is masked from 'package:tibble':
##
##     view
```

```
library(leaps)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(dplyr)
library(ggplot2)
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.1.1 --
## v broom        1.0.5     v rsample      1.2.0
## v dials        1.2.0     v tune         1.1.2
## v infer        1.0.5     v workflows    1.1.3
## v modeldata    1.2.0     v workflowsets 1.0.1
## v parsnip      1.1.1     v yardstick    1.2.0
## v recipes      1.0.8
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard()        masks purrr::discard()
## x Matrix::expand()         masks tidyr::expand()
## x dplyr::filter()          masks stats::filter()
## x recipes::fixed()         masks stringr::fixed()
## x dplyr::lag()             masks stats::lag()
## x caret::lift()            masks purrr::lift()
## x Matrix::pack()           masks tidyr::pack()
## x yardstick::precision()   masks caret::precision()
## x yardstick::recall()      masks caret::recall()
## x yardstick::sensitivity() masks caret::sensitivity()
## x yardstick::spec()        masks readr::spec()
## x yardstick::specificity() masks caret::specificity()
## x recipes::step()          masks stats::step()
## x Matrix::unpack()         masks tidyr::unpack()
## x recipes::update()        masks Matrix::update(), stats::update()
## x summarytools::view()     masks tibble::view()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```r
library(plotmo)
```

```
## Loading required package: Formula
## Loading required package: plotrix
##
## Attaching package: 'plotrix'
##
## The following object is masked from 'package:scales':
##
##     rescale
##
## Loading required package: TeachingDemos
```

```r
library(caret)
library(tidymodels)
library(splines)
library(mgcv)
```

```
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
## This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.
```

```r
library(pdp)
```

```
##
## Attaching package: 'pdp'
##
## The following object is masked from 'package:purrr':
##
##     partial
```

```r
library(earth)
library(tidyverse)
library(ggplot2)
library(bayesQR)
```

Exploratory analysis and data visualization

```r
load("recovery.RData")
st_options(plain.ascii = TRUE,
           style = "rmarkdown",
           dfSummary.silent = TRUE,
           footnote = NA,
           subtitle.emphasis = FALSE)
dfSummary(dat[,-1])
```

```
## Data Frame Summary
## dat
## Dimensions: 3000 x 15
## Duplicates: 0
##
## ----------------------------------------------------------------------------------------------------
## No   Variable         Stats / Values             Freqs (% of Valid)    Graph                 Valid
## ---- ---------------- -------------------------- --------------------- --------------------- --------
## 1    age              Mean (sd) : 60.2 (4.5)     34 distinct values          : .             3000
##      [numeric]        min < med < max:                                       : :             (100.0%)
##                       42 < 60 < 79                                           : :
##                       IQR (CV) : 6 (0.1)                                   . : : .
##                                                                           : : : :
##
## 2    gender          Min  : 0                   0 : 1544 (51.5%)      IIIIIIIIII            3000
##      [integer]       Mean : 0.5                 1 : 1456 (48.5%)      IIIIIIIII             (100.0%)
##                      Max  : 1
##
## 3    race            1. 1                       1967 (65.6%)          IIIIIIIIIIIII         3000
##      [factor]        2. 2                        158 ( 5.3%)          I                     (100.0%)
##                      3. 3                        604 (20.1%)          IIII
##                      4. 4                        271 ( 9.0%)          I
##
## 4    smoking         1. 0                       1822 (60.7%)          IIIIIIIIIIII          3000
##      [factor]        2. 1                        859 (28.6%)          IIIII                 (100.0%)
##                      3. 2                        319 (10.6%)          II
##
## 5    height          Mean (sd) : 169.9 (6)      313 distinct values          : :           3000
##      [numeric]       min < med < max:                                        : :           (100.0%)
##                      147.8 < 169.9 < 188.6                                 . : : : .
##                      IQR (CV) : 7.9 (0)                                    : : : :
##                                                                          . : : : : : .
##
## 6    weight          Mean (sd) : 80 (7.1)       364 distinct values          : .           3000
##      [numeric]       min < med < max:                                        : :           (100.0%)
##                      55.9 < 79.8 < 103.7                                   : : : :
##                      IQR (CV) : 9.6 (0.1)                                . : : : : : .
##                                                                        . : : : : : : : .
##
## 7    bmi             Mean (sd) : 27.8 (2.8)     163 distinct values        . :             3000
##      [numeric]       min < med < max:                                      : : :           (100.0%)
##                      18.8 < 27.6 < 38.9                                    : : :
##                      IQR (CV) : 3.7 (0.1)                                : : : : :
##                                                                        . : : : : : : .
##
## 8    hypertension    Min  : 0                   0 : 1508 (50.3%)      IIIIIIIIII            3000
##      [numeric]       Mean : 0.5                 1 : 1492 (49.7%)      IIIIIIIII             (100.0%)
##                      Max  : 1
##
## 9    diabetes        Min  : 0                   0 : 2537 (84.6%)      IIIIIIIIIIIIIIII      3000
##      [integer]       Mean : 0.2                 1 :  463 (15.4%)      III                   (100.0%)
##                      Max  : 1
##
## 10   SBP             Mean (sd) : 130.5 (8)      52 distinct values           : .           3000
##      [numeric]       min < med < max:                                        : : .         (100.0%)
##                      105 < 130 < 156                                       : : : :
##                      IQR (CV) : 11 (0.1)                                 . : : : : .
##                                                                        . : : : : : : : .
##
## 11   LDL             Mean (sd) : 110.5 (19.8)   114 distinct values            :           3000
##      [numeric]       min < med < max:                                        : : .         (100.0%)
##                      28 < 110 < 178                                          : : :
##                      IQR (CV) : 27 (0.2)                                   . : : : .
##                                                                          . : : : : : .
##
## 12   vaccine         Min  : 0                   0 : 1212 (40.4%)      IIIIIIII              3000
##      [integer]       Mean : 0.6                 1 : 1788 (59.6%)      IIIIIIIIIII           (100.0%)
##                      Max  : 1
##
## 13   severity        Min  : 0                   0 : 2679 (89.3%)      IIIIIIIIIIIIIIIII     3000
##      [integer]       Mean : 0.1                 1 :  321 (10.7%)      II                    (100.0%)
##                      Max  : 1
##
## 14   study           1. A                       2000 (66.7%)          IIIIIIIIIIIII         3000
##      [character]     2. B                       1000 (33.3%)          IIIIII                (100.0%)
##
## 15   recovery_time   Mean (sd) : 42.2 (23.2)    140 distinct values   : :                   3000
##      [numeric]       min < med < max:                                 : :                   (100.0%)
##                      2 < 39 < 365                                     : :
##                      IQR (CV) : 18 (0.5)                              : :
##                                                                       : : .
## ----------------------------------------------------------------------------------------------------
```

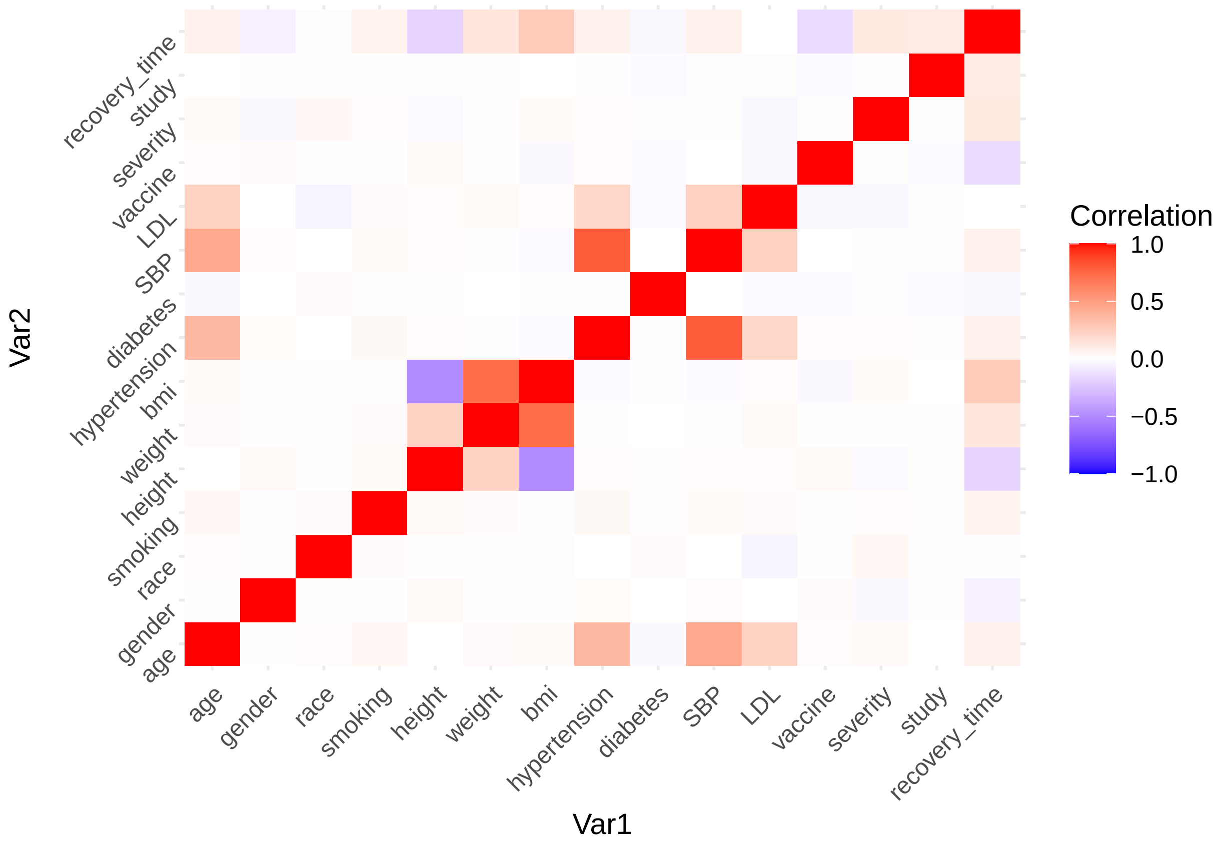
```
summary(dat)
```

```
##        id              age            gender        race     smoking
##  Min.   :   1.0   Min.   :42.0   Min.   :0.0000   1:1967   0:1822
##  1st Qu.: 750.8   1st Qu.:57.0   1st Qu.:0.0000   2: 158   1: 859
##  Median :1500.5   Median :60.0   Median :0.0000   3: 604   2: 319
##  Mean   :1500.5   Mean   :60.2   Mean   :0.4853   4: 271
##  3rd Qu.:2250.2   3rd Qu.:63.0   3rd Qu.:1.0000
##  Max.   :3000.0   Max.   :79.0   Max.   :1.0000
##      height          weight            bmi         hypertension
##  Min.   :147.8   Min.   : 55.90   Min.   :18.80   Min.   :0.0000
##  1st Qu.:166.0   1st Qu.: 75.20   1st Qu.:25.80   1st Qu.:0.0000
##  Median :169.9   Median : 79.80   Median :27.65   Median :0.0000
##  Mean   :169.9   Mean   : 79.96   Mean   :27.76   Mean   :0.4973
##  3rd Qu.:173.9   3rd Qu.: 84.80   3rd Qu.:29.50   3rd Qu.:1.0000
##  Max.   :188.6   Max.   :103.70   Max.   :38.90   Max.   :1.0000
##     diabetes           SBP             LDL           vaccine
##  Min.   :0.0000   Min.   :105.0   Min.   : 28.0   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:125.0   1st Qu.: 97.0   1st Qu.:0.000
##  Median :0.0000   Median :130.0   Median :110.0   Median :1.000
##  Mean   :0.1543   Mean   :130.5   Mean   :110.5   Mean   :0.596
##  3rd Qu.:0.0000   3rd Qu.:136.0   3rd Qu.:124.0   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :156.0   Max.   :178.0   Max.   :1.000
##     severity        study           recovery_time
##  Min.   :0.000   Length:3000        Min.   :  2.00
##  1st Qu.:0.000   Class :character   1st Qu.: 31.00
##  Median :0.000   Mode  :character   Median : 39.00
##  Mean   :0.107                      Mean   : 42.17
##  3rd Qu.:0.000                      3rd Qu.: 49.00
##  Max.   :1.000                      Max.   :365.00
```

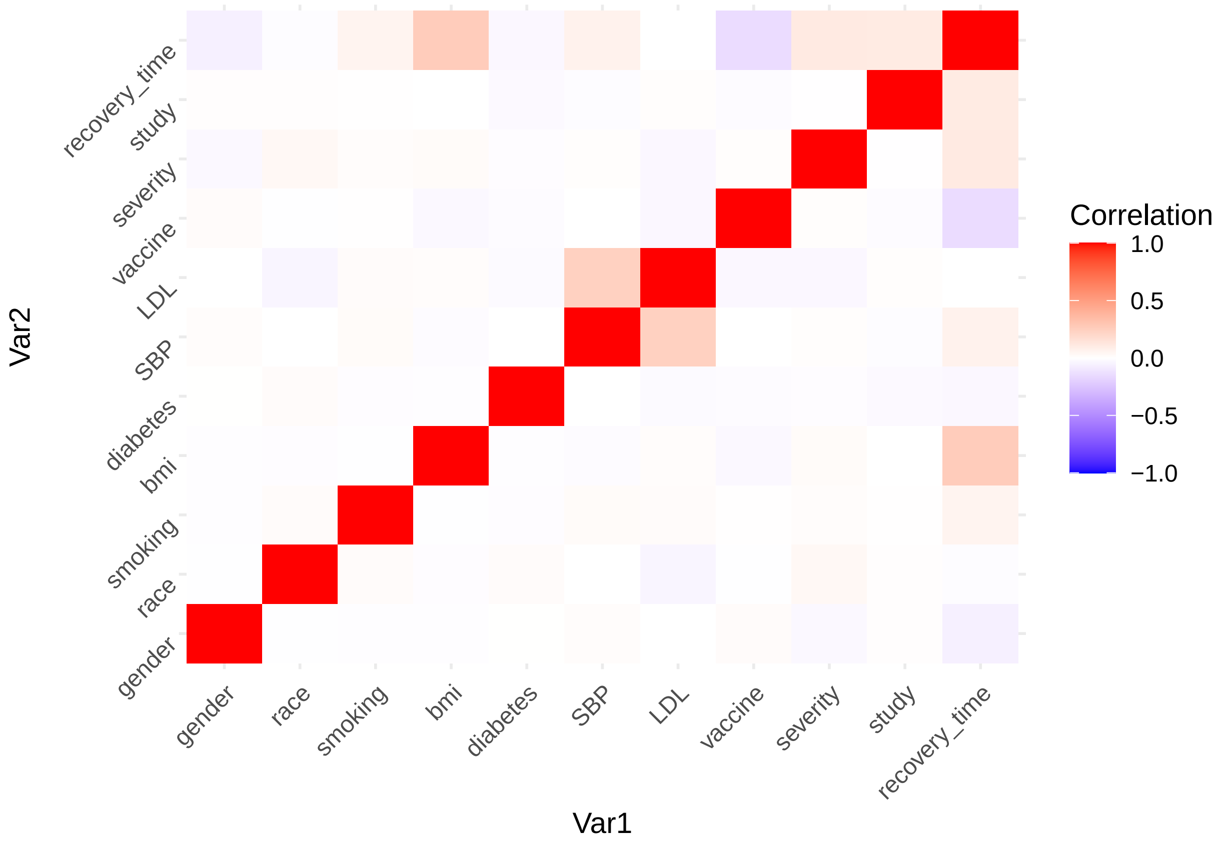
```
columns_to_convert <- c("gender", "race", "smoking", "hypertension", "diabetes", "vaccine", "severity")

dat$study <- as.character(dat$study)
unique(dat$study)
```

```
## [1] "A" "B"
```

```r
# Convert selected factor variables to numeric using mutate
dat <- dat %>%
  mutate(across(all_of(columns_to_convert), as.numeric)) %>%
  mutate(study = case_when(
    study == "A" ~ 1,
    study == "B" ~ 2
  ))

numeric_data <- dat[, c("age", "gender", "race", "smoking", "height", "weight", "bmi", "hypertension",

# Compute correlation matrix
correlation_matrix <- cor(numeric_data)
correlation_df <- as.data.frame(as.table(correlation_matrix))
names(correlation_df) <- c("Var1", "Var2", "Correlation")

ggplot(correlation_df, aes(x = Var1, y = Var2, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limits = c(-1, 1),
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        axis.text.y = element_text(angle = 45, vjust = 1, hjust = 1))
```



```
dat_subset <- dat %>%
  select(-id, -height, -weight, -hypertension, -age)
numeric_data <- dat_subset[, c("gender", "race", "smoking", "bmi", "diabetes", "SBP", "LDL", "vaccine",

# Compute correlation matrix
correlation_matrix <- cor(numeric_data)
correlation_df <- as.data.frame(as.table(correlation_matrix))
names(correlation_df) <- c("Var1", "Var2", "Correlation")

ggplot(correlation_df, aes(x = Var1, y = Var2, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, limits = c(-1, 1),
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        axis.text.y = element_text(angle = 45, vjust = 1, hjust = 1))
```



Model training

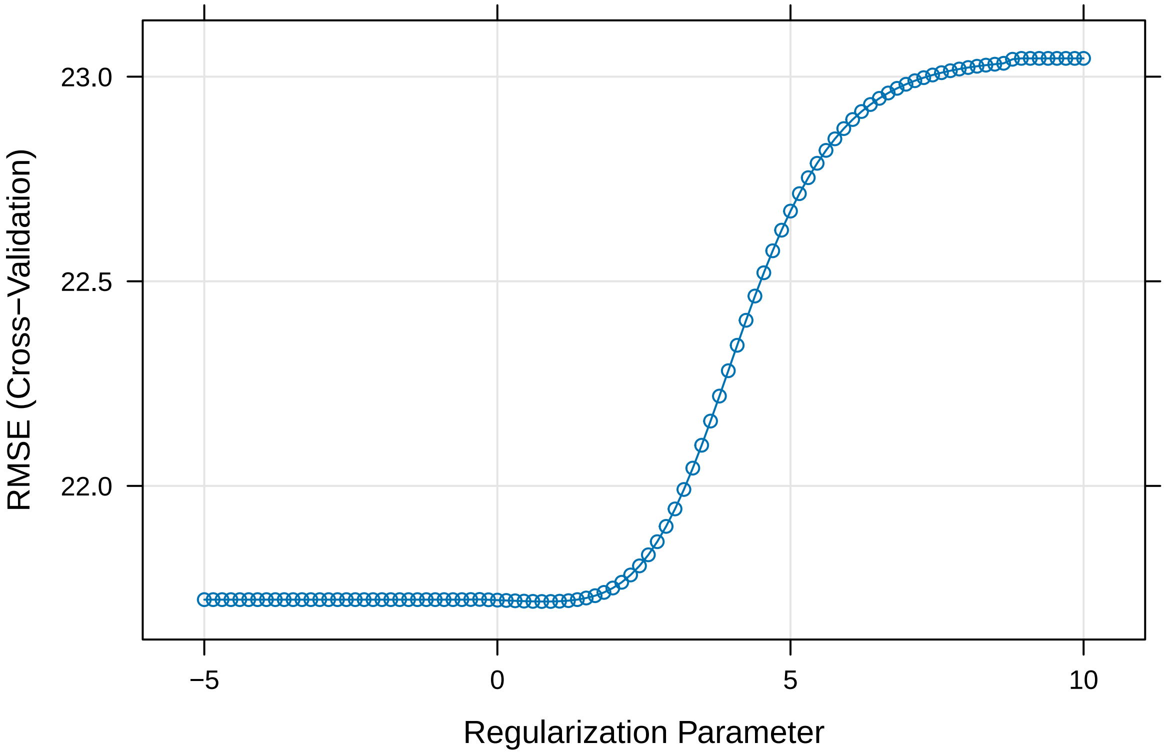
```
ctrl1 <- trainControl(method = "cv", number = 10)
set.seed(111)
data <-
  dat_subset %>%
  mutate(gender = as.factor(gender),
         race = as.factor(race),
         smoking = as.factor(smoking),
         diabetes = as.factor(diabetes),
         vaccine = as.factor(vaccine),
         severity = as.factor(severity),
         study = as.factor(study))
data_split <- initial_split(data, prop = 0.8)
# Extract the training and test data
training_data <- training(data_split)
testing_data <- testing(data_split)
```

Ridge regression

```
set.seed(111)
ridge.fit <- train(recovery_time ~ . ,
                   data = training_data,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                       lambda = exp(seq(10, -5, length=100))),
                   trControl = ctrl1)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
plot(ridge.fit, xTrans = log)
```



```
ridge.fit$bestTune
```

```
##    alpha   lambda
## 39     0 2.133099
```

```
coef(ridge.fit$finalModel, s = ridge.fit$bestTune$lambda)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) -39.99066788
## gender1      -2.03150346
## race2         3.76316239
## race3         0.27703667
## race4        -0.49507418
## smoking2      1.24572928
## smoking3      2.10917123
## bmi           2.10138727
## diabetes1    -1.43993196
## SBP           0.20603710
## LDL          -0.01277054
## vaccine1     -5.77510955
## severity1     6.79253148
## study2        4.71375040
```

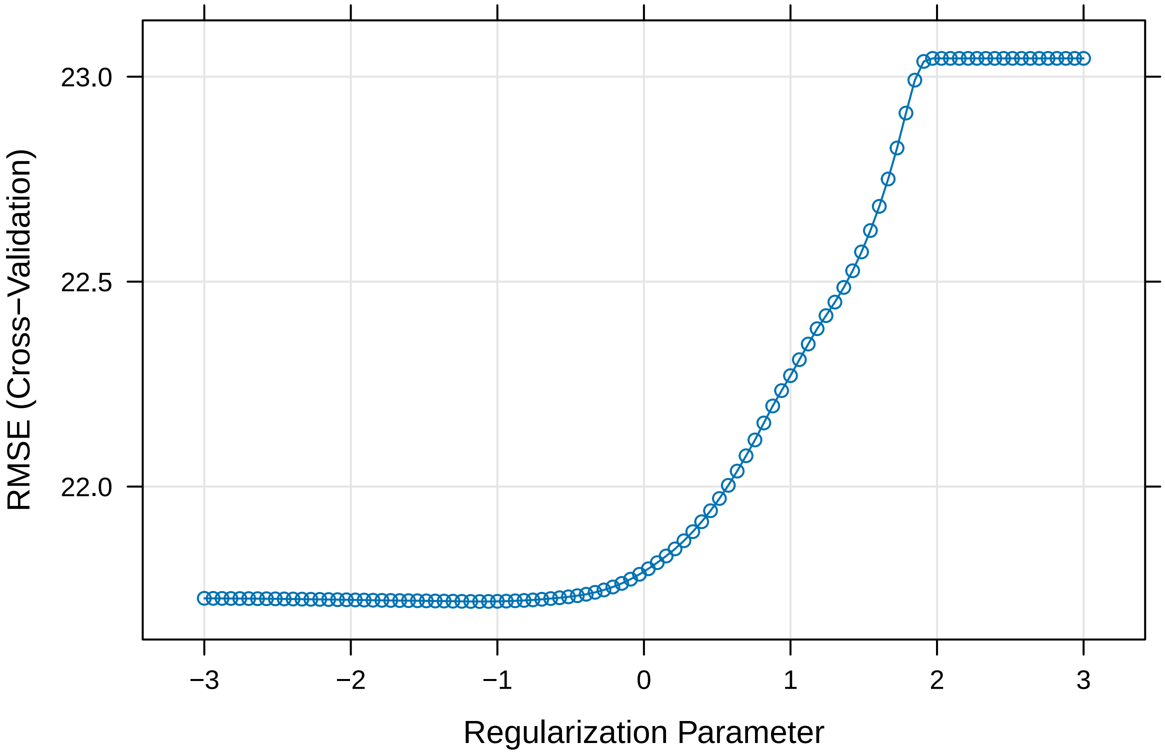
```
ridge.pred <- predict(ridge.fit, newdata = testing_data)
# test error
mean((ridge.pred - testing_data[, "recovery_time"])^2)
```

```
## [1] 427.0682
```

```
set.seed(111)
lasso.fit <- train(recovery_time ~ .,
                   data = training_data,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(3, -3, length = 100))),
                   trControl = ctrl1)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
plot(lasso.fit, xTrans = log)
```



```
lasso.fit$bestTune
```

```
##    alpha    lambda
## 32     1 0.3258845
```

```
coef(lasso.fit$finalModel, lasso.fit$bestTune$lambda)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) -39.6053968
## gender1      -1.5578746
## race2         2.5493572
## race3         .
## race4         .
## smoking2      0.5190367
## smoking3      1.0052440
## bmi           2.1775220
## diabetes1    -0.6394963
## SBP           0.1768472
## LDL           .
## vaccine1     -5.6533751
## severity1     6.4249923
## study2        4.4427281
```

```r
lasso.pred <- predict(lasso.fit, newdata = testing_data)
# test error
mean((lasso.pred - testing_data[, "recovery_time"])^2)
```

```
## [1] 431.4991
```

Elastic Net

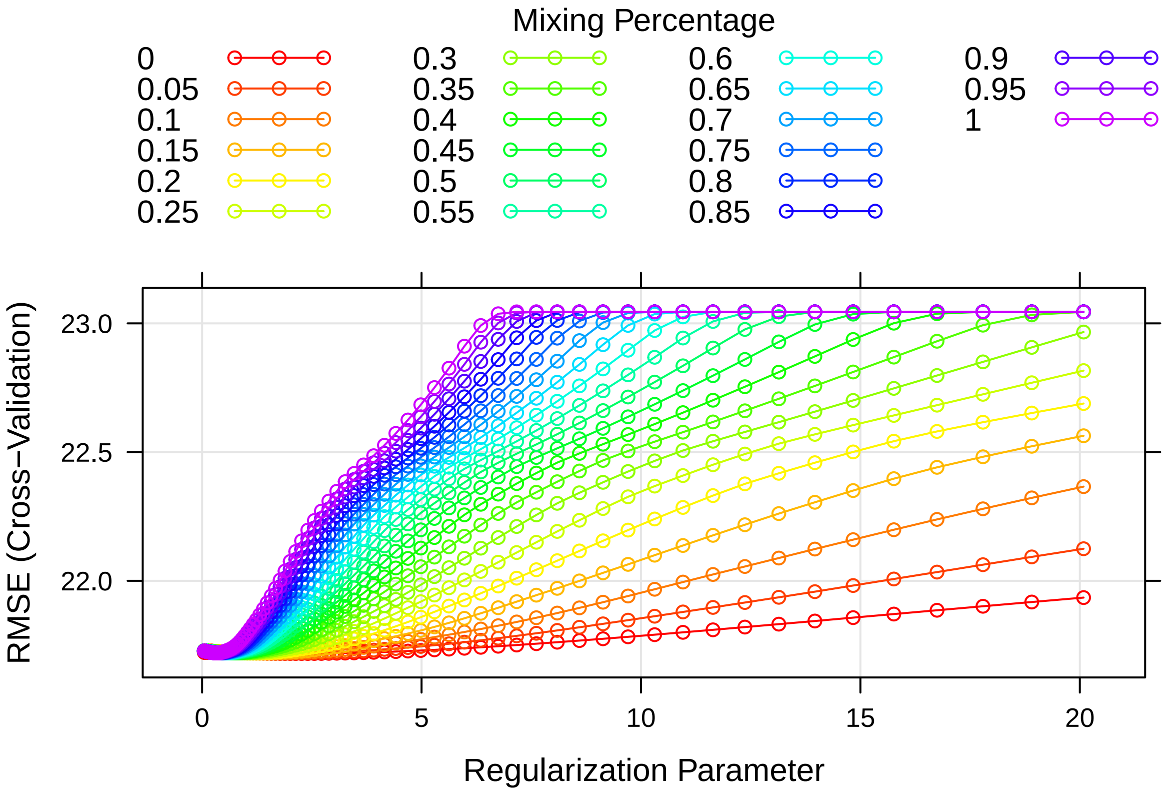
```r
set.seed(111)
enet.fit <- train(recovery_time ~ .,
                  data = training_data,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                         lambda = exp(seq(3, -3, length = 100))),
                  trControl = ctrl1)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```r
enet.fit$bestTune
```

```
##    alpha   lambda
## 64     0 2.266375
```

```r
myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
plot(enet.fit, par.settings = myPar)
```



```
coef(enet.fit$finalModel, enet.fit$bestTune$lambda)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) -39.56757379
## gender1      -2.02212306
## race2         3.74890700
## race3         0.27474258
## race4        -0.49037543
## smoking2      1.23986327
## smoking3      2.09517232
## bmi           2.09066734
## diabetes1    -1.43365274
## SBP           0.20480789
## LDL          -0.01252015
## vaccine1     -5.74718186
## severity1     6.76046127
## study2        4.69005538
```

```
enet.pred <- predict(enet.fit, newdata = testing_data)
# test error
mean((enet.pred - testing_data[, "recovery_time"])^2)
```

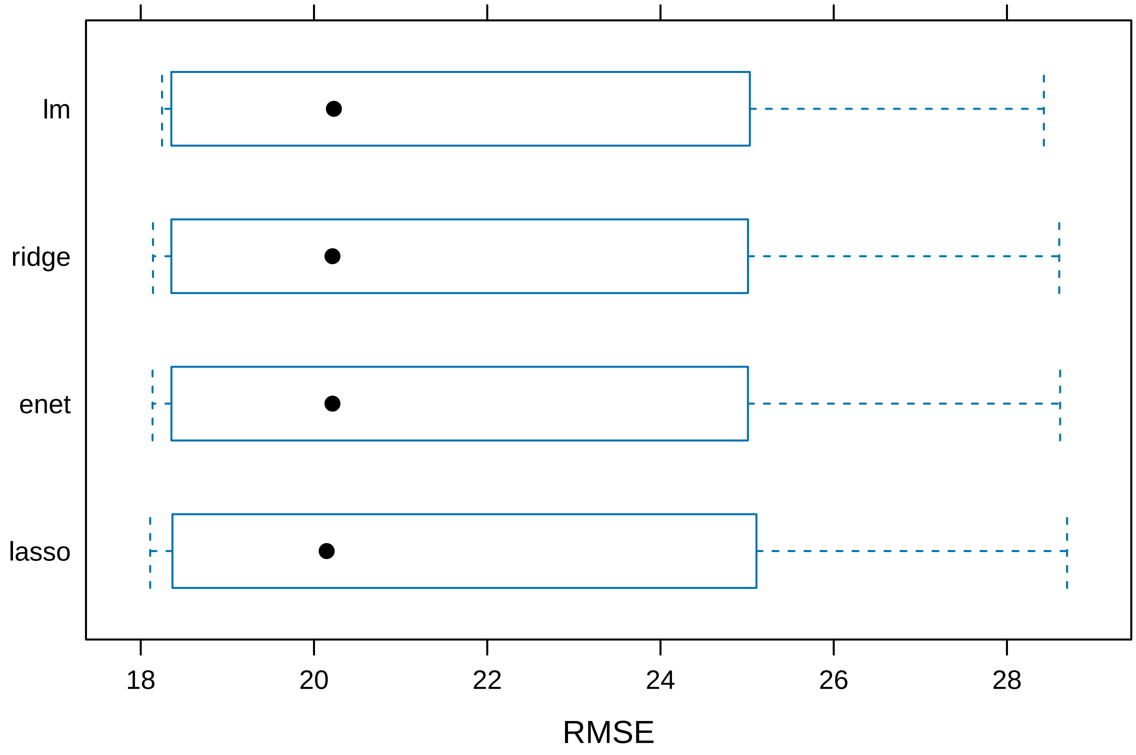
```
## [1] 427.0791
```

Comparison

```r
set.seed(111)
lm.fit <- train(recovery_time ~ .,
                data = training_data,
                method = "lm",
                trControl = ctrl1)
resamp <- resamples(list(enet = enet.fit, lasso = lasso.fit, ridge = ridge.fit, lm = lm.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: enet, lasso, ridge, lm
## Number of resamples: 10
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA’s
## enet  12.06815 12.71472 13.17008 13.36525 13.60493 15.44596    0
## lasso 12.07222 12.72041 13.15336 13.37658 13.64259 15.54233    0
## ridge 12.07265 12.71805 13.17603 13.37100 13.60944 15.44725    0
## lm    12.15894 12.78772 13.28894 13.48031 13.69656 15.47492    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA’s
## enet  18.13603 18.44825 20.21325 21.71749 24.32500 28.61377    0
## lasso 18.10858 18.47894 20.14645 21.71970 24.39625 28.69381    0
## ridge 18.14126 18.44868 20.21356 21.71752 24.32451 28.60322    0
## lm    18.24549 18.46878 20.22984 21.72940 24.32765 28.42577    0
##
## Rsquared
##             Min.    1st Qu.     Median      Mean   3rd Qu.      Max. NA’s
## enet  0.07305218 0.08515099 0.09979822 0.1217721 0.1523059 0.2392763    0
## lasso 0.07338295 0.08085135 0.10533213 0.1216336 0.1532726 0.2310643    0
## ridge 0.07303745 0.08516798 0.09977657 0.1217754 0.1523387 0.2392584    0
## lm    0.07276847 0.08527662 0.09973873 0.1218198 0.1528988 0.2389192    0
```

```r
bwplot(resamp, metric = "RMSE")
```



PCR

```
# show information about the model
modelLookup("pcr")
```

```
##   model parameter       label forReg forClass probModel
## 1   pcr     ncomp #Components   TRUE    FALSE     FALSE
```

```
modelLookup("pls")
```

```
##   model parameter       label forReg forClass probModel
## 1   pls     ncomp #Components   TRUE     TRUE      TRUE
```

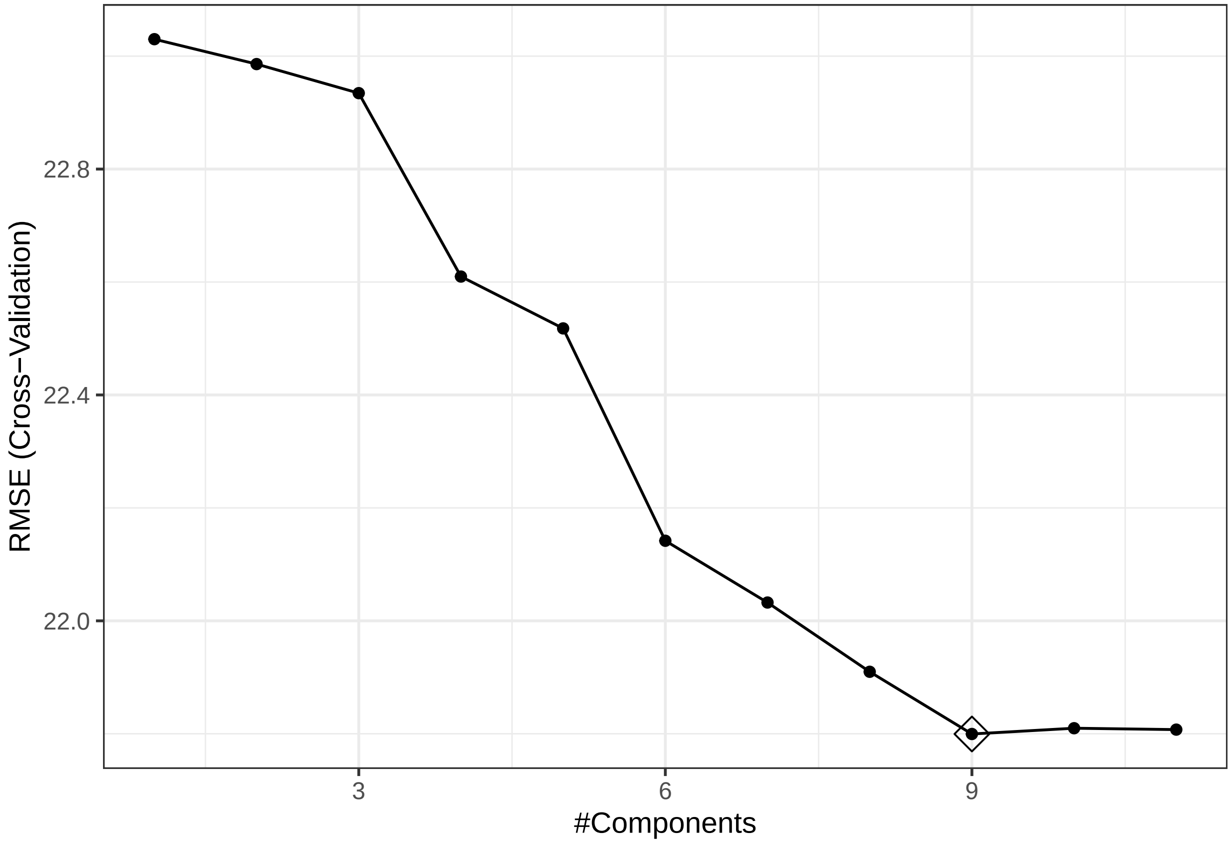
```
x <- model.matrix(recovery_time ~ ., training_data)[, -1]
y <- training_data$recovery_time
# test data
x2 <- model.matrix(recovery_time ~ .,testing_data)[, -1]
y2 <- testing_data$recovery_time

set.seed(111)
pcr.fit <- train(recovery_time ~ .,
                 data = training_data,
                 method = "pcr",
                 tuneGrid = data.frame(ncomp = 1:11),
                 trControl = ctrl1,
                  preProcess = c("center", "scale"))
pcr.fit$bestTune
```

```
##   ncomp
## 9     9
```

```
predy2.pcr2 <- predict(pcr.fit, newdata = testing_data)
mean((y2 - predy2.pcr2)^2)
```

```
## [1] 437.2806
```

```
ggplot(pcr.fit, highlight = TRUE) + theme_bw()
```



PLS

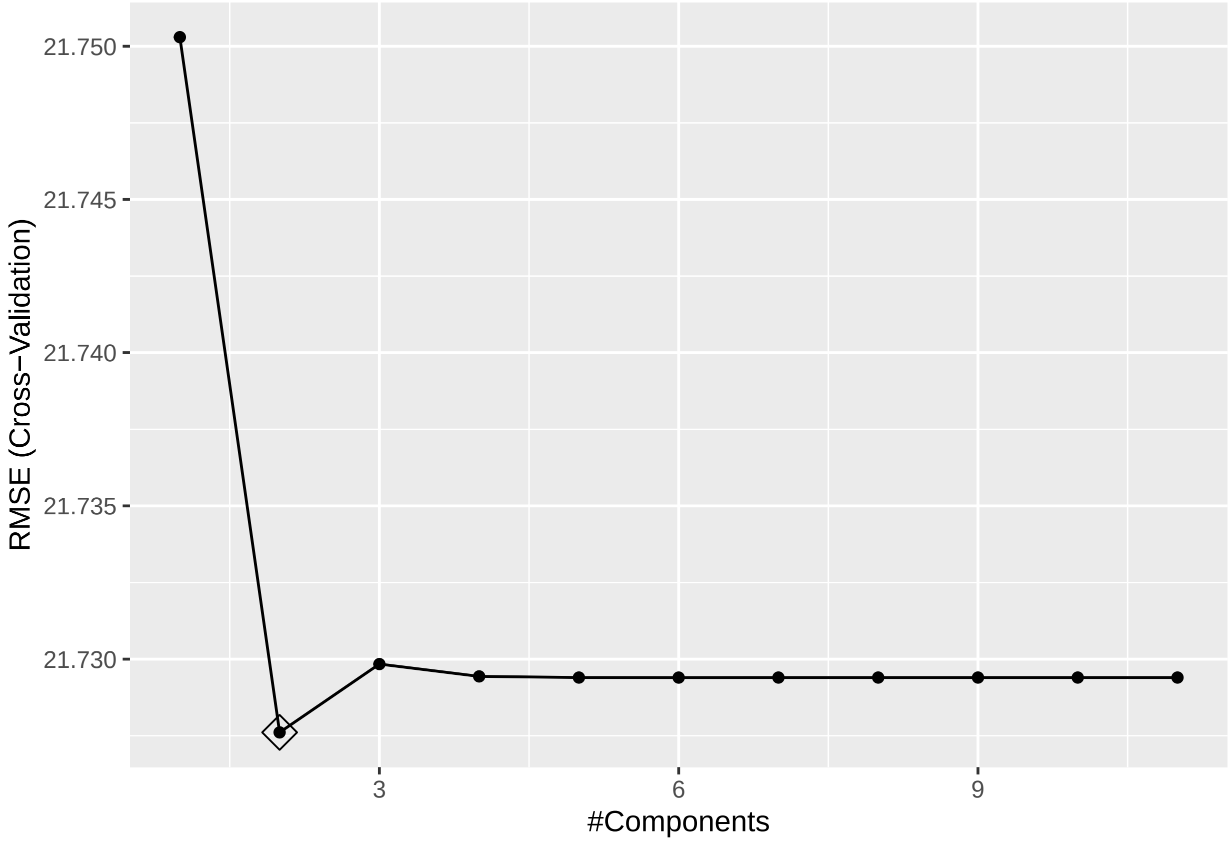
```
set.seed(111)
pls.fit <- train(recovery_time ~ .,
                  data = training_data,
                  method = "pls",
                  tuneGrid = data.frame(ncomp = 1:11),
                  trControl = ctrl1,
                  preProcess = c("center", "scale"))
predy2.pls2 <- predict(pls.fit, newdata = testing_data)
mean((y2 - predy2.pls2)^2)
```

```
## [1] 427.7757
```

```
pls.fit$bestTune
```

```
##   ncomp
## 2     2
```

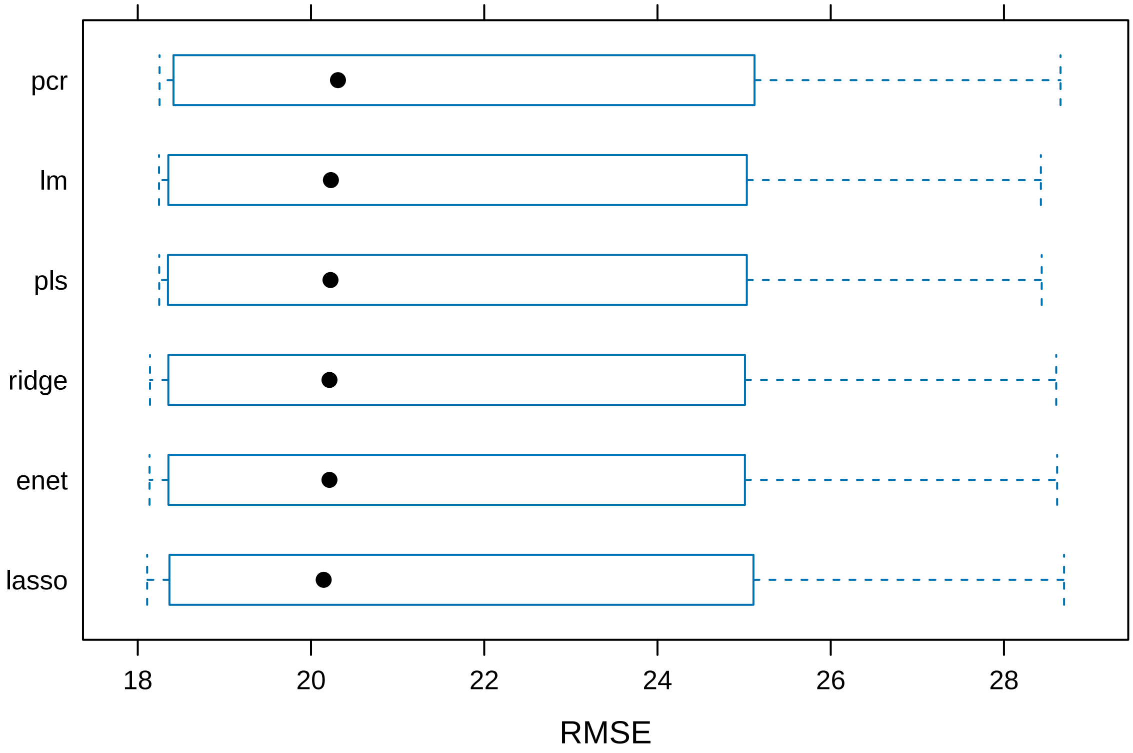
```
ggplot(pls.fit, highlight = TRUE)
```



```
resamp <- resamples(list(enet = enet.fit, lasso = lasso.fit, ridge = ridge.fit, lm = lm.fit,pls = pls.fi
summary(resamp)
```

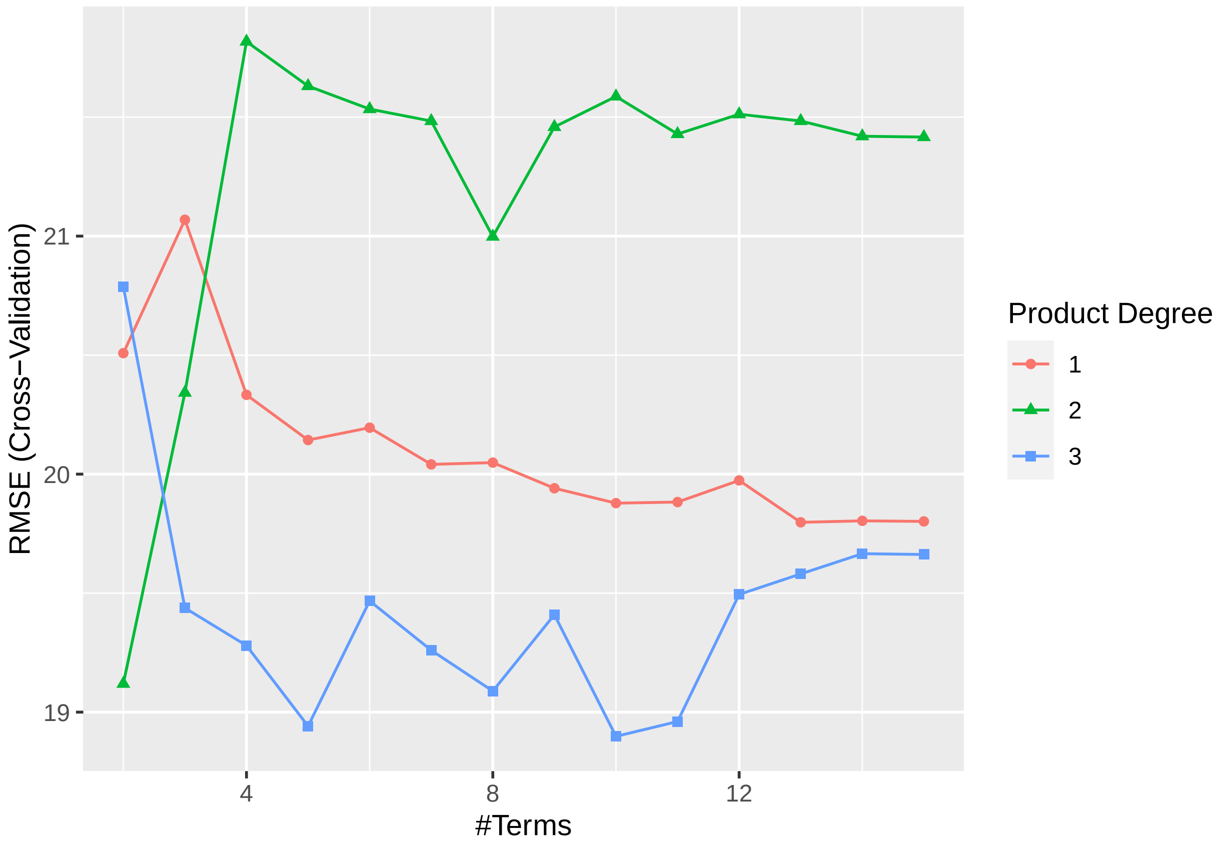
```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: enet, lasso, ridge, lm, pls, pcr
## Number of resamples: 10
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## enet  12.06815 12.71472 13.17008 13.36525 13.60493 15.44596    0
## lasso 12.07222 12.72041 13.15336 13.37658 13.64259 15.54233    0
## ridge 12.07265 12.71805 13.17603 13.37100 13.60944 15.44725    0
## lm    12.15894 12.78772 13.28894 13.48031 13.69656 15.47492    0
## pls   12.16257 12.79255 13.28120 13.48183 13.72053 15.49690    0
## pcr   12.28527 12.85206 13.27795 13.52402 13.89703 15.70430    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## enet  18.13603 18.44825 20.21325 21.71749 24.32500 28.61377    0
## lasso 18.10858 18.47894 20.14645 21.71970 24.39625 28.69381    0
## ridge 18.14126 18.44868 20.21356 21.71752 24.32451 28.60322    0
## lm    18.24549 18.46878 20.22984 21.72940 24.32765 28.42577    0
## pls   18.24773 18.46280 20.22510 21.72761 24.33183 28.43546    0
## pcr   18.25151 18.53173 20.31123 21.79963 24.45173 28.65251    0
##
## Rsquared
##             Min.    1st Qu.     Median      Mean   3rd Qu.      Max. NA's
## enet  0.07305218 0.08515099 0.09979822 0.1217721 0.1523059 0.2392763    0
## lasso 0.07338295 0.08085135 0.10533213 0.1216336 0.1532726 0.2310643    0
## ridge 0.07303745 0.08516798 0.09977657 0.1217754 0.1523387 0.2392584    0
## lm    0.07276847 0.08527662 0.09973873 0.1218198 0.1528988 0.2389192    0
## pls   0.07319969 0.08570529 0.10014792 0.1219070 0.1534540 0.2383485    0
## pcr   0.07403719 0.07755086 0.09968917 0.1155529 0.1387400 0.2257989    0
```

```r
bwplot(resamp, metric = "RMSE")
```



MARS

```
mars_grid <- expand.grid(degree = 1:3, nprune = 2:15)
set.seed(111)

x3<-model.matrix(recovery_time ~ ., data)[,-1]
y3<-data$recovery_time
mars.fit <- train(recovery_time ~ .,data = training_data,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)
ggplot(mars.fit)
```



```
mars.fit$bestTune
```

```
##    nprune degree
## 37     10      3
```

```
mars.pred <- predict(mars.fit, newdata = testing_data)
mean((mars.pred - testing_data[, "recovery_time"])^2)
```

```
## [1] 371.0523
```

```r
set.seed(111)
gam.fit <- train(recovery_time ~ .,
                 data = training_data,
                 method = "gam",
                 trControl = ctrl1)
gam.fit$bestTune
```

```
##   select method
## 1  FALSE GCV.Cp
```

```r
gam.fit$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender1 + race3 + race4 + smoking2 + smoking3 + diabetes1 +
##     vaccine1 + severity1 + study2 + s(SBP) + s(LDL) + s(bmi)
##
## Estimated degrees of freedom:
## 2.59 3.10 8.24  total = 23.92
##
## GCV score: 383.102
```

```r
gam_pred <- predict(gam.fit, newdata = testing_data)
mean((gam_pred - testing_data$recovery_time)^2)
```

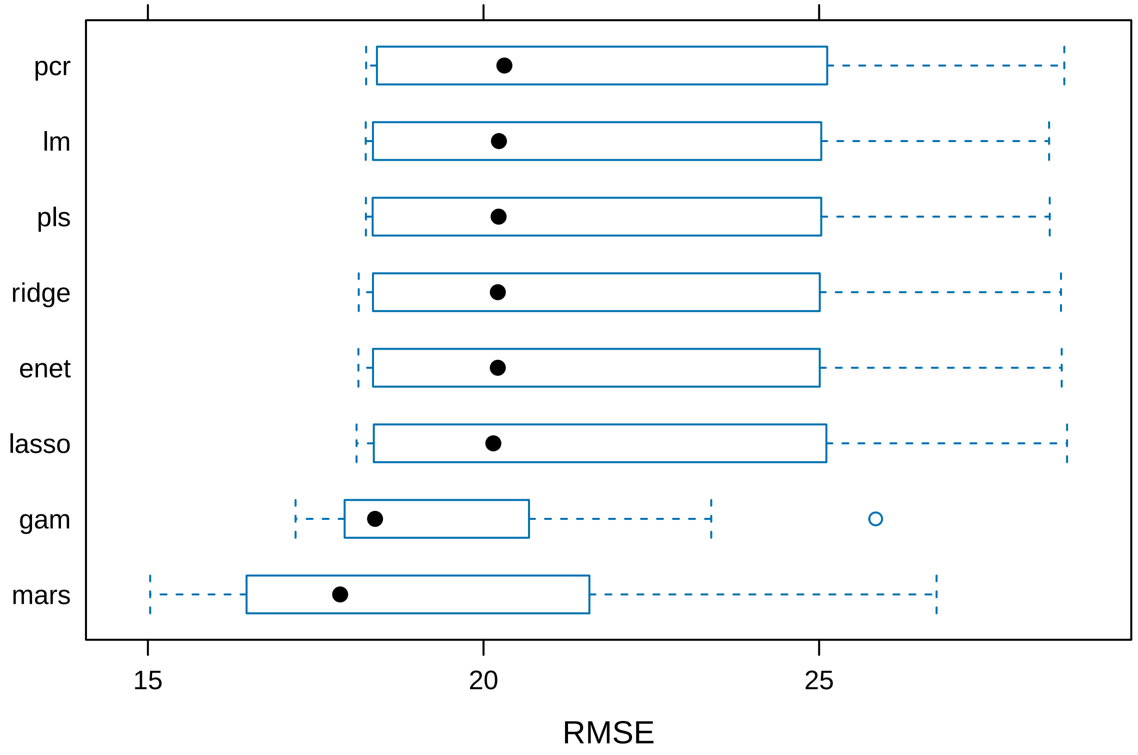
```
## [1] 330.1853
```

```r
resamp <- resamples(list(enet = enet.fit, lasso = lasso.fit, ridge = ridge.fit, lm = lm.fit,pls = pls.f
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: enet, lasso, ridge, lm, pls, pcr, mars, gam
## Number of resamples: 10
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## enet  12.06815 12.71472 13.17008 13.36525 13.60493 15.44596    0
## lasso 12.07222 12.72041 13.15336 13.37658 13.64259 15.54233    0
## ridge 12.07265 12.71805 13.17603 13.37100 13.60944 15.44725    0
## lm    12.15894 12.78772 13.28894 13.48031 13.69656 15.47492    0
## pls   12.16257 12.79255 13.28120 13.48183 13.72053 15.49690    0
## pcr   12.28527 12.85206 13.27795 13.52402 13.89703 15.70430    0
## mars  10.10553 11.59179 11.95667 12.06618 12.75610 13.64774    0
## gam   10.76759 12.17114 12.53876 12.71612 13.03311 15.03980    0
##
```

```
## RMSE
##            Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA’s
## enet  18.13603 18.44825 20.21325 21.71749 24.32500 28.61377    0
## lasso 18.10858 18.47894 20.14645 21.71970 24.39625 28.69381    0
## ridge 18.14126 18.44868 20.21356 21.71752 24.32451 28.60322    0
## lm    18.24549 18.46878 20.22984 21.72940 24.32765 28.42577    0
## pls   18.24773 18.46280 20.22510 21.72761 24.33183 28.43546    0
## pcr   18.25151 18.53173 20.31123 21.79963 24.45173 28.65251    0
## mars  15.03378 16.57506 17.86446 18.89814 20.85112 26.74932    0
## gam   17.19998 17.95971 18.38445 19.81000 20.59671 25.84242    0
##
## Rsquared
##              Min.    1st Qu.     Median      Mean   3rd Qu.      Max. NA’s
## enet  0.07305218 0.08515099 0.09979822 0.1217721 0.1523059 0.2392763    0
## lasso 0.07338295 0.08085135 0.10533213 0.1216336 0.1532726 0.2310643    0
## ridge 0.07303745 0.08516798 0.09977657 0.1217754 0.1523387 0.2392584    0
## lm    0.07276847 0.08527662 0.09973873 0.1218198 0.1528988 0.2389192    0
## pls   0.07319969 0.08570529 0.10014792 0.1219070 0.1534540 0.2383485    0
## pcr   0.07403719 0.07755086 0.09968917 0.1155529 0.1387400 0.2257989    0
## mars  0.15268114 0.27369692 0.41155049 0.3713612 0.4865602 0.5606916    0
## gam   0.13200573 0.18638380 0.31944309 0.2885874 0.3767280 0.4161776    0
```

```r
bwplot(resamp, metric = "RMSE")
```


pcr
lm
pls
ridge
enet
lasso
gam
mars
RMSE