8106mid

Ze Li

```
library(ggplot2)
library(MASS)
library(glmnet)
library(rsample)
library(corrplot)
library(caret)
library(mgcv)
library(tidyverse)
library(earth)
library(Formula)
library(plotmo)
library(plotrix)
library(TeachingDemos)
library(gridExtra)
library(patchwork)
```

```
load("/Users/zeze/Library/Mobile Documents/com~apple~CloudDocs/2024/24S BIST P8106 DS II/midtermproject/
dat <- as.data.frame(dat)
head(dat)
```

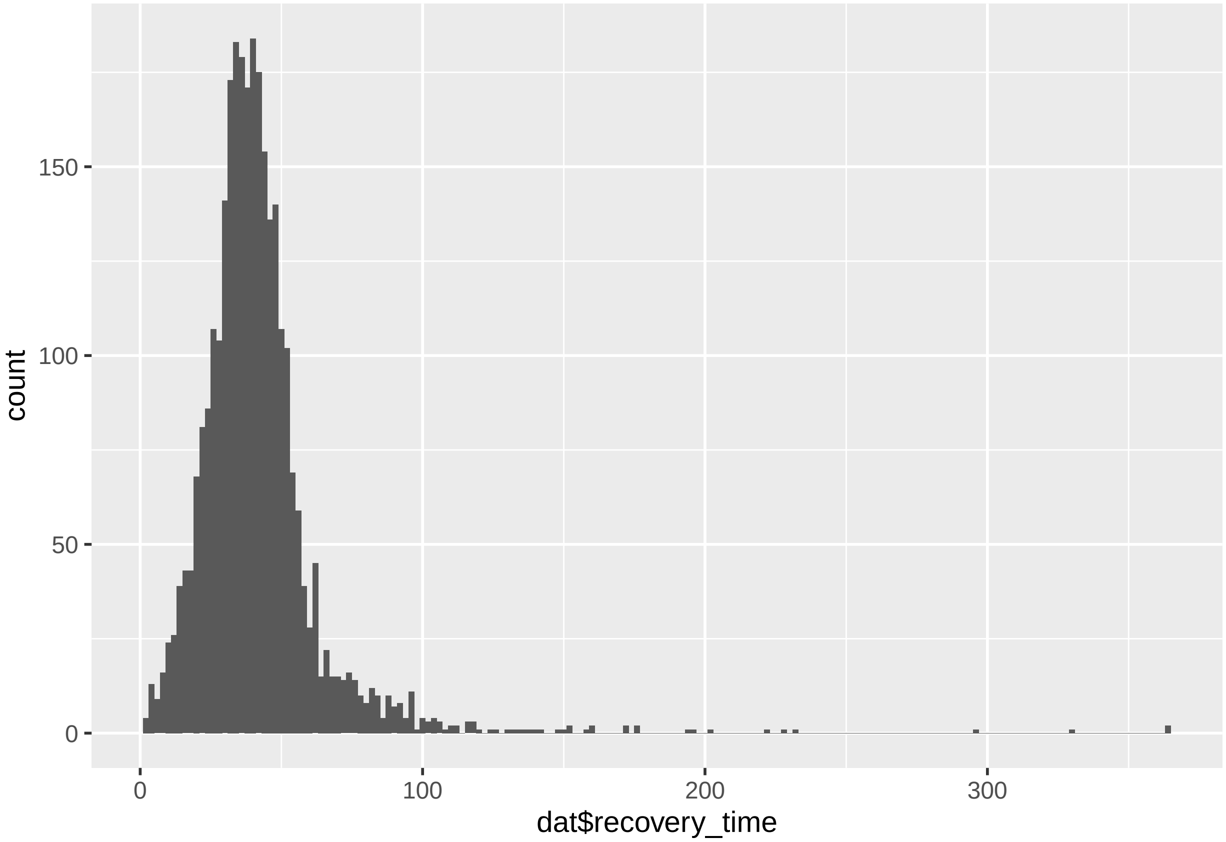
```
##   id age gender race smoking height weight  bmi hypertension diabetes SBP LDL
## 1  1  56      0    1       2  170.2   78.7 27.2            0        0 120  97
## 2  2  70      1    1       1  169.6   73.1 25.4            1        0 134 112
## 3  3  57      1    1       0  168.4   77.4 27.3            1        0 131  88
## 4  4  53      0    1       0  166.7   76.1 27.4            0        0 115  87
## 5  5  59      1    1       2  173.6   70.2 23.3            0        0 127 118
## 6  6  60      1    3       1  162.8   75.1 28.4            0        0 129 104
##   vaccine severity study recovery_time
## 1       0        0     A            31
## 2       0        0     A            44
## 3       1        0     A            29
## 4       0        1     A            47
## 5       1        0     A            40
## 6       0        0     A            34
```

```
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
```

```r
ggplot(dat, aes(x = dat$recovery_time)) + geom_histogram(binwidth = 2)
```

```
## Warning: Use of ‘dat$recovery_time‘ is discouraged.
## i Use ‘recovery_time‘ instead.
```

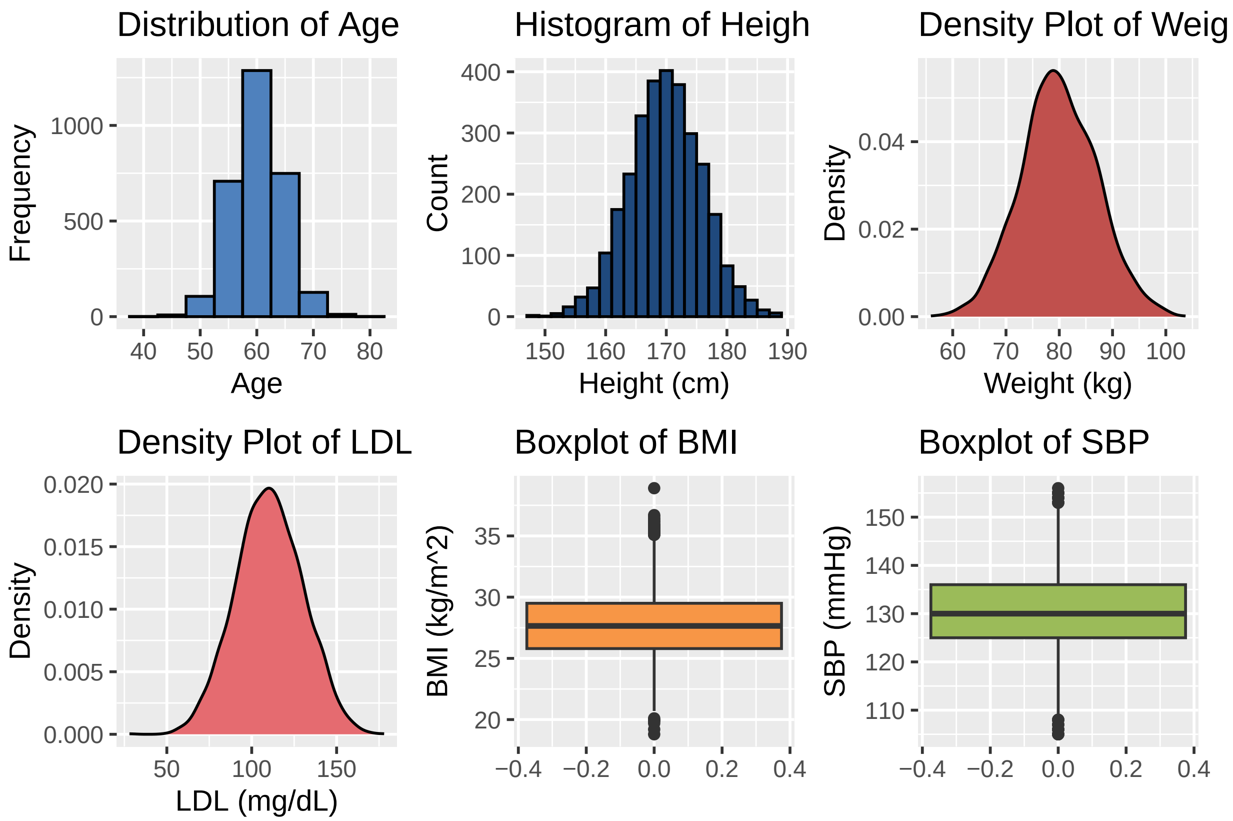


Exploratary Data Analysis

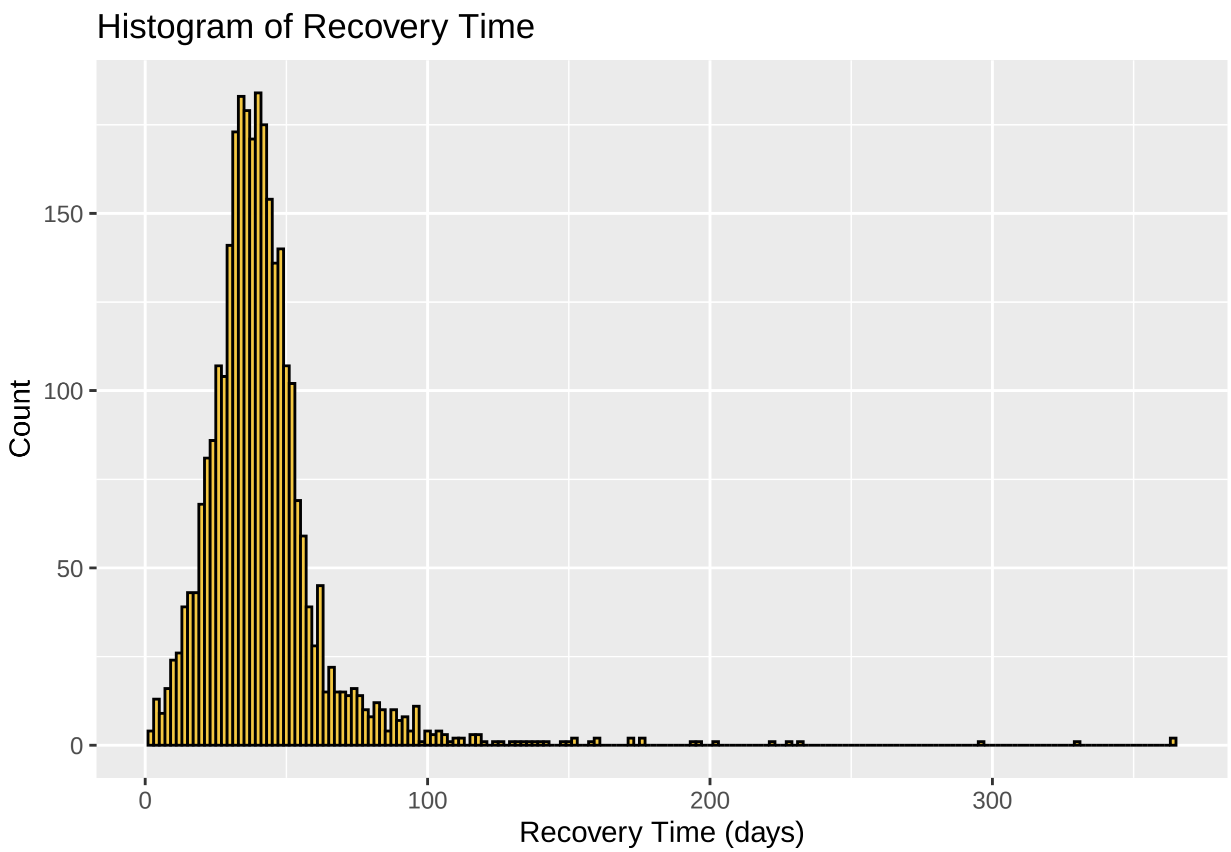
Univariate Analysis

```
# Histogram for Age
p1 <- ggplot(dat, aes(x = age)) +
  geom_histogram(binwidth = 5, fill = "#4F81BD", color = "black") +
  labs(title = "Distribution of Age", x = "Age", y = "Frequency")

# Histogram for Height
p2 <- ggplot(dat, aes(x = height)) +
  geom_histogram(binwidth = 2, fill = "#1F497D", color = "black") +
  labs(title = "Histogram of Height", x = "Height (cm)", y = "Count")

# Density Plot for Weight
p3 <- ggplot(dat, aes(x = weight)) +
  geom_density(fill = "#C0504D") +
  labs(title = "Density Plot of Weight", x = "Weight (kg)", y = "Density")

# Density Plot for LDL
p4 <- ggplot(dat, aes(x = LDL)) +
  geom_density(fill = "#E56B70") +
  labs(title = "Density Plot of LDL", x = "LDL (mg/dL)", y = "Density")
```

```r
# Boxplot for BMI
p5 <- ggplot(dat, aes(y = bmi)) +
  geom_boxplot(fill = "#F79646") +
  labs(title = "Boxplot of BMI", x = "", y = "BMI (kg/m^2)")

# Boxplot for SBP
p6 <- ggplot(dat, aes(y = SBP)) +  # Corrected to display SBP instead of BMI again
  geom_boxplot(fill = "#9BBB59") +
  labs(title = "Boxplot of SBP", x = "", y = "SBP (mmHg)")

# Histogram for Recovery Time
p7 <- ggplot(dat, aes(x = recovery_time)) +
  geom_histogram(binwidth = 2, fill = "#F4C842", color = "black") +
  labs(title = "Histogram of Recovery Time", x = "Recovery Time (days)", y = "Count")

# Arranging the plots in a 2x3 grid
plot_grid <- p1 + p2 + p3 + p4 + p5 + p6 +
  plot_layout(ncol = 3, byrow = TRUE)

# Display the combined plot
plot_grid
```

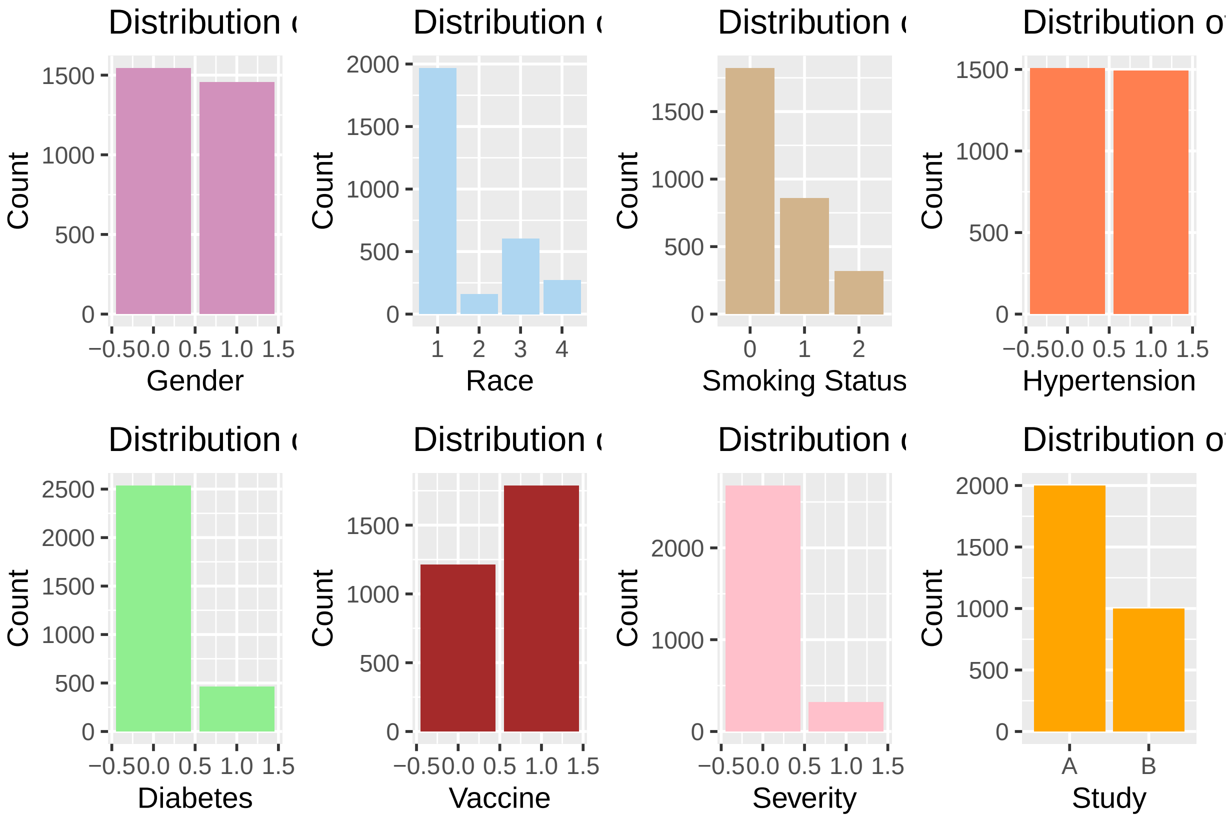


```r
p7
```



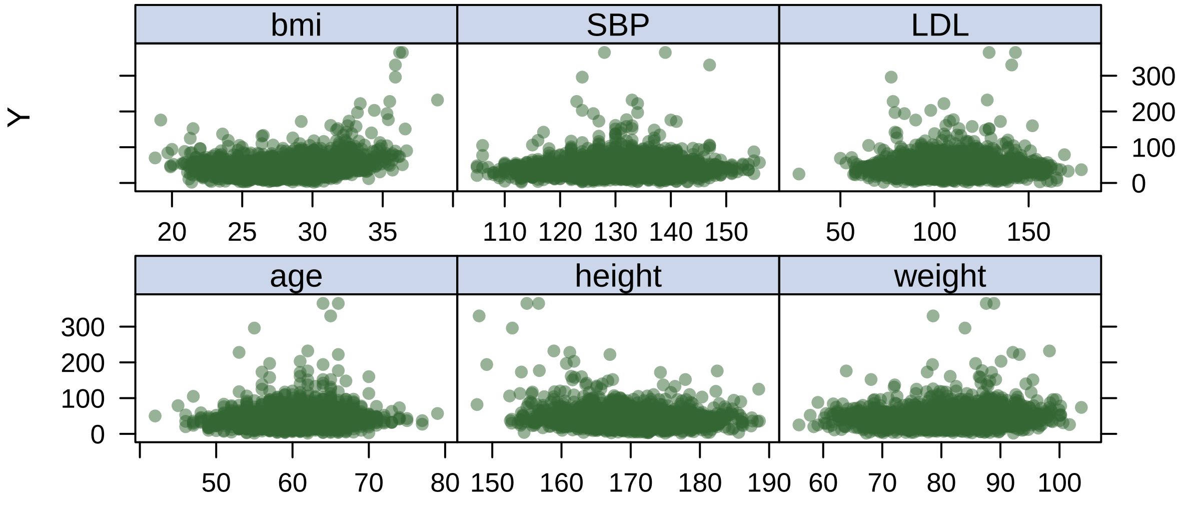
```
# Bar Plot for Gender
p1 <- ggplot(dat, aes(x = gender)) +
  geom_bar(fill = "#D291BC") +
  labs(title = "Distribution of Gender", x = "Gender", y = "Count")

# Bar Plot for Race
p2 <- ggplot(dat, aes(x = race)) +
  geom_bar(fill = "#AED6F1") +
  labs(title = "Distribution of Race", x = "Race", y = "Count")

# Bar Plot for Smoking Status
p3 <- ggplot(dat, aes(x = smoking)) +
  geom_bar(fill = "#D2B48C") +
  labs(title = "Distribution of Smoking Status", x = "Smoking Status", y = "Count")

# Bar Plot for Hypertension
p4 <- ggplot(dat, aes(x = hypertension)) +
  geom_bar(fill = "#FF7F50") +
  labs(title = "Distribution of Hypertension", x = "Hypertension", y = "Count")

# Bar Plot for Diabetes
p5 <- ggplot(dat, aes(x = diabetes)) +
  geom_bar(fill = "#90EE90") +
  labs(title = "Distribution of Diabetes", x = "Diabetes", y = "Count")

# Bar plot for Vaccine
```

```
p6 <- ggplot(dat, aes(x = vaccine)) +
  geom_bar(fill = "#A52A2A") +
  labs(title = "Distribution of Vaccine", x = "Vaccine", y = "Count")

# Bar plot for Severity
p7 <- ggplot(dat, aes(x = severity)) +
  geom_bar(fill = "#FFC0CB") +
  labs(title = "Distribution of Severity", x = "Severity", y = "Count")

# Bar plot for Study
p8 <- ggplot(dat, aes(x = study)) +
  geom_bar(fill = "#FFA500") +
  labs(title = "Distribution of Study", x = "Study", y = "Count")

# Combine the plots into a 2x4 grid
plot_grid <- p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 +
  plot_layout(ncol = 4, byrow = TRUE)

# Display the combined plot
plot_grid
```

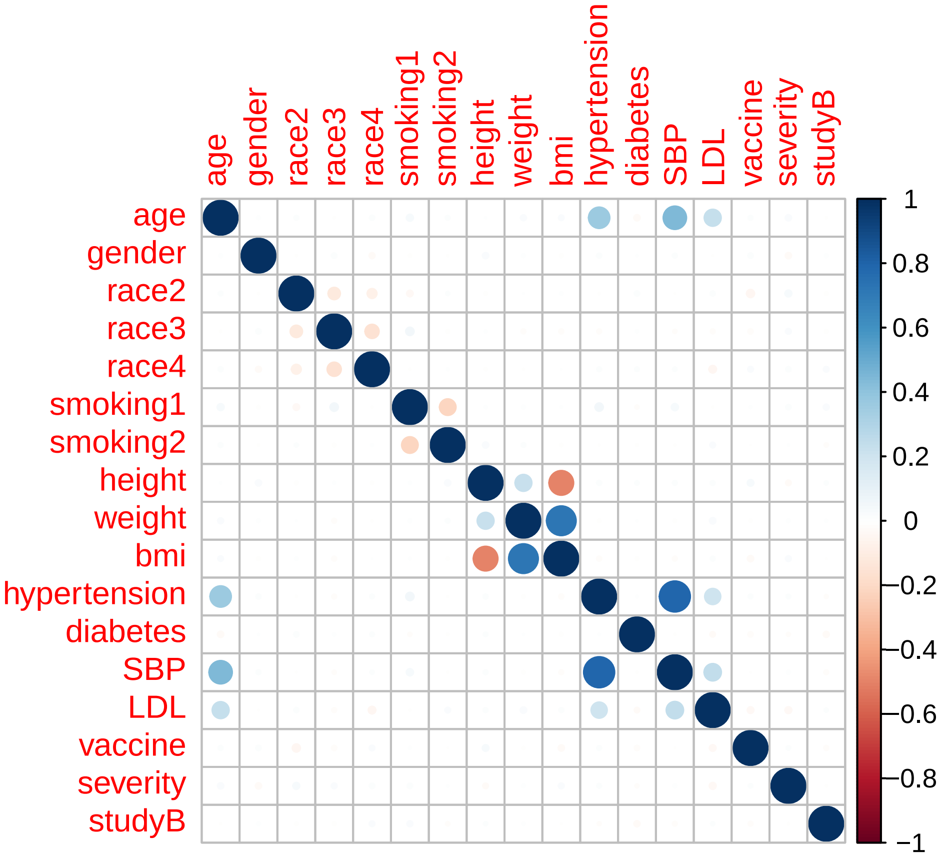


bivariate visualization

```
# matrix of predictors
x.orig <- model.matrix(recovery_time ~ ., dat[,-1])[, -1]
# vector of response
y.orig <- dat$recovery_time

theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)

featurePlot(x.orig[, -c(2, 3, 4, 5, 6, 7, 11, 12, 15, 16, 17)], y.orig, plot = "scatter", labels = c(""
            type = c("p"), layout = c(3, 3))
```



```
corrplot(cor(x.orig), method = "circle", type = "full")
```



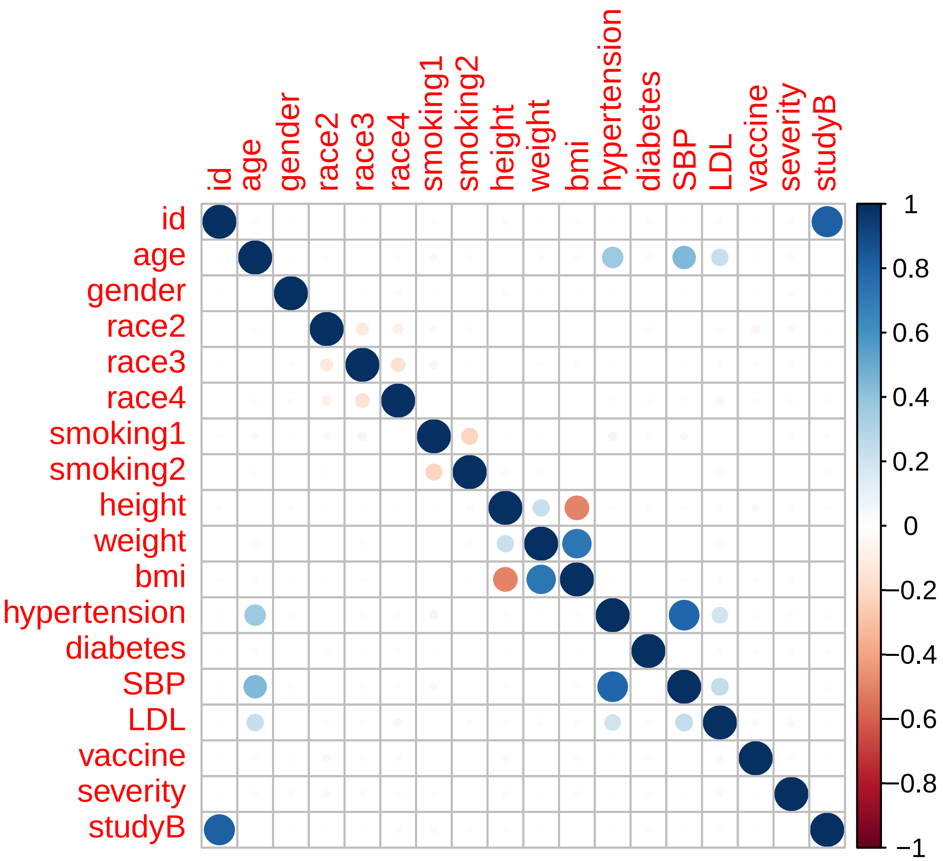
linear regression

```
# Fit a multiple linear regression model
model1 <- lm(recovery_time ~ ., data = dat)

# Summarize the model
summary(model1)
```

```
##
## Call:
## lm(formula = recovery_time ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.168 -10.997  -0.272   8.664 258.278
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.188e+03  1.044e+02 -20.964  < 2e-16 ***
## id              2.504e-04  7.363e-04   0.340 0.733854
## age             2.170e-01  9.279e-02   2.339 0.019407 *
## gender         -2.976e+00  7.368e-01  -4.039 5.49e-05 ***
## race2           2.036e+00  1.670e+00   1.219 0.222908
## race3        -7.797e-01  9.390e-01  -0.830 0.406410
## race4        -7.574e-01  1.309e+00  -0.579 0.562752
## smoking1      2.433e+00  8.366e-01   2.908 0.003665 **
## smoking2      3.442e+00  1.223e+00   2.814 0.004928 **
## height        1.277e+01  6.123e-01  20.851  < 2e-16 ***
## weight       -1.385e+01  6.468e-01 -21.408  < 2e-16 ***
## bmi           4.150e+01  1.857e+00  22.351  < 2e-16 ***
## hypertension  2.123e+00  1.214e+00   1.750 0.080267 .
## diabetes     -1.484e+00  1.019e+00  -1.456 0.145571
## SBP           5.932e-02  7.917e-02   0.749 0.453776
## LDL          -3.887e-02  1.945e-02  -1.998 0.045759 *
## vaccine      -6.387e+00  7.521e-01  -8.493  < 2e-16 ***
## severity      7.512e+00  1.194e+00   6.294 3.55e-10 ***
## studyB        4.535e+00  1.353e+00   3.351 0.000816 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.13 on 2981 degrees of freedom
## Multiple R-squared:  0.2485, Adjusted R-squared:  0.244
## F-statistic: 54.77 on 18 and 2981 DF,  p-value: < 2.2e-16
```

cross validation

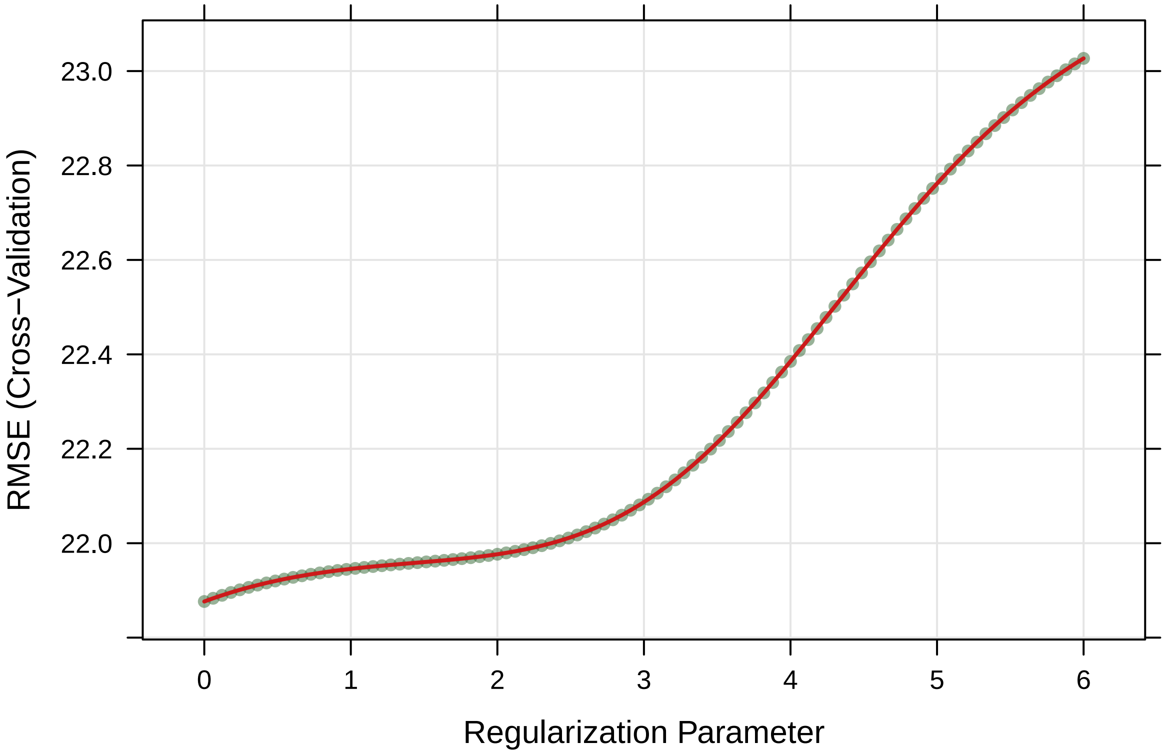
```r
set.seed(7890)
data_split <- initial_split(dat, prop = 0.8)

# Extract the training and test data
train <- training(data_split)
test <- testing(data_split)
# matrix of predictors (glmnet uses input matrix)
x <- model.matrix(recovery_time ~ ., dat)[,-1]
# vector of response
y <- dat$recovery_time
corrplot(cor(x), method = "circle", type = "full")
```



```
x_train <- model.matrix(recovery_time ~ ., train[,-1])[,-1]
y_train <- train$recovery_time
x_test <- model.matrix(recovery_time ~ ., test[,-1])[,-1]
y_test <- test$recovery_time
```

ridge regression

```
ctrl1 <- trainControl(method ="cv",number=10)
set.seed(7890)
ridge.fit <- train(recovery_time ~ .,
                   data = train[,-1],
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(6, 0, length = 100))),
                   trControl = ctrl1)

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



```
ridge.fit$bestTune
```

```
##   alpha lambda
## 1     0      1
```

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

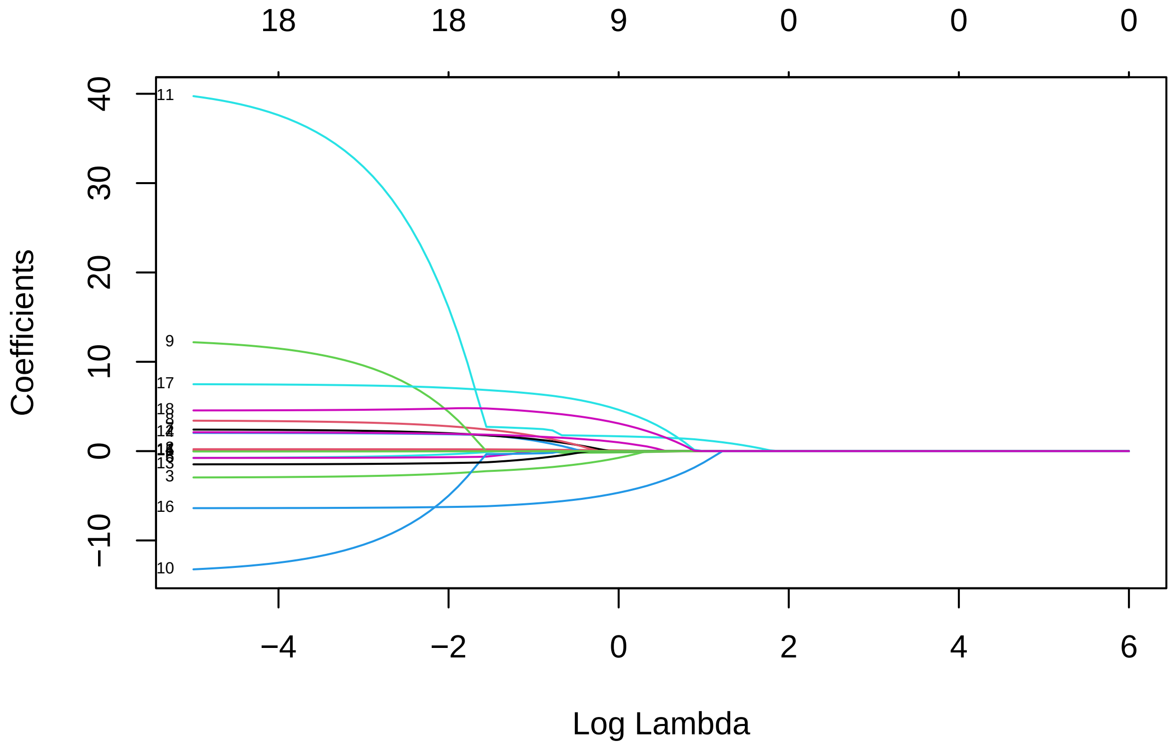
```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  -49.83377815
## age            0.16533422
## gender        -2.56824288
## race2          2.55401865
## race3         -1.14838615
## race4         -0.35066541
## smoking1       2.54406485
## smoking2       2.74257488
## height         0.16469270
## weight        -0.49964996
## bmi            3.18812674
## hypertension   2.22965489
## diabetes      -1.73185880
## SBP            0.08389363
## LDL           -0.04084677
## vaccine       -6.18711894
## severity       7.09546547
## studyB         4.98970576
```

LASSO regression

```
# LASSO regression
## lasso alpha = 1
cv.lasso <- cv.glmnet(x, y,
                      alpha = 1,
                      lambda = exp(seq(6, -5, length = 100)))

cv.lasso$lambda.min
```

```
## [1] 0.006737947
```

```
# trace plot
plot(cv.lasso$glmnet.fit, xvar = "lambda", label=TRUE)
```



```
predict(cv.lasso, s = "lambda.min", type = "coefficients")
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                   lambda.min
## (Intercept)  -2.089325e+03
## id            2.373267e-04
## age           2.166036e-01
## gender       -2.948733e+00
## race2         2.033937e+00
## race3        -7.580487e-01
## race4        -7.635900e-01
## smoking1      2.410317e+00
## smoking2      3.407395e+00
## height        1.218796e+01
## weight       -1.323432e+01
## bmi           3.973616e+01
## hypertension  2.113575e+00
## diabetes     -1.480874e+00
## SBP           5.945777e-02
## LDL          -3.829888e-02
## vaccine      -6.383491e+00
## severity      7.489264e+00
## studyB        4.554325e+00
```

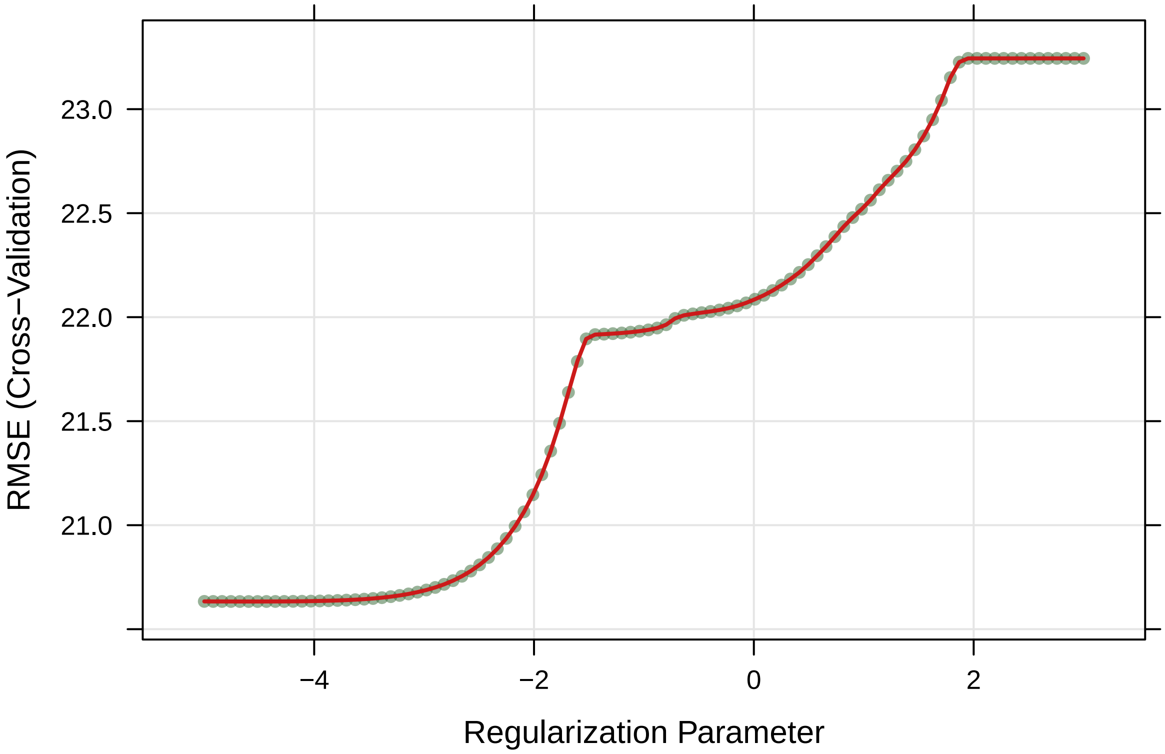
```
head(predict(cv.lasso, newx = model.matrix(recovery_time ~ .,dat)[,-1],
             s = "lambda.min", type = "response"))
```

```
##   lambda.min
## 1   43.30510
## 2   40.03784
## 3   33.13477
## 4   46.52219
## 5   33.19537
## 6   44.87192
```

```
ctrl1 <- trainControl(method = "cv", number = 10)

set.seed(7890)
lasso.fit <- train(recovery_time ~ .,
                   data = train[,-1],
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(3, -5, length = 100))),
                   trControl = ctrl1)
```

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

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



```
# tuning parameter
lasso.fit$bestTune
```

```
##   alpha     lambda
## 6     1 0.01009253
```

elastic net model

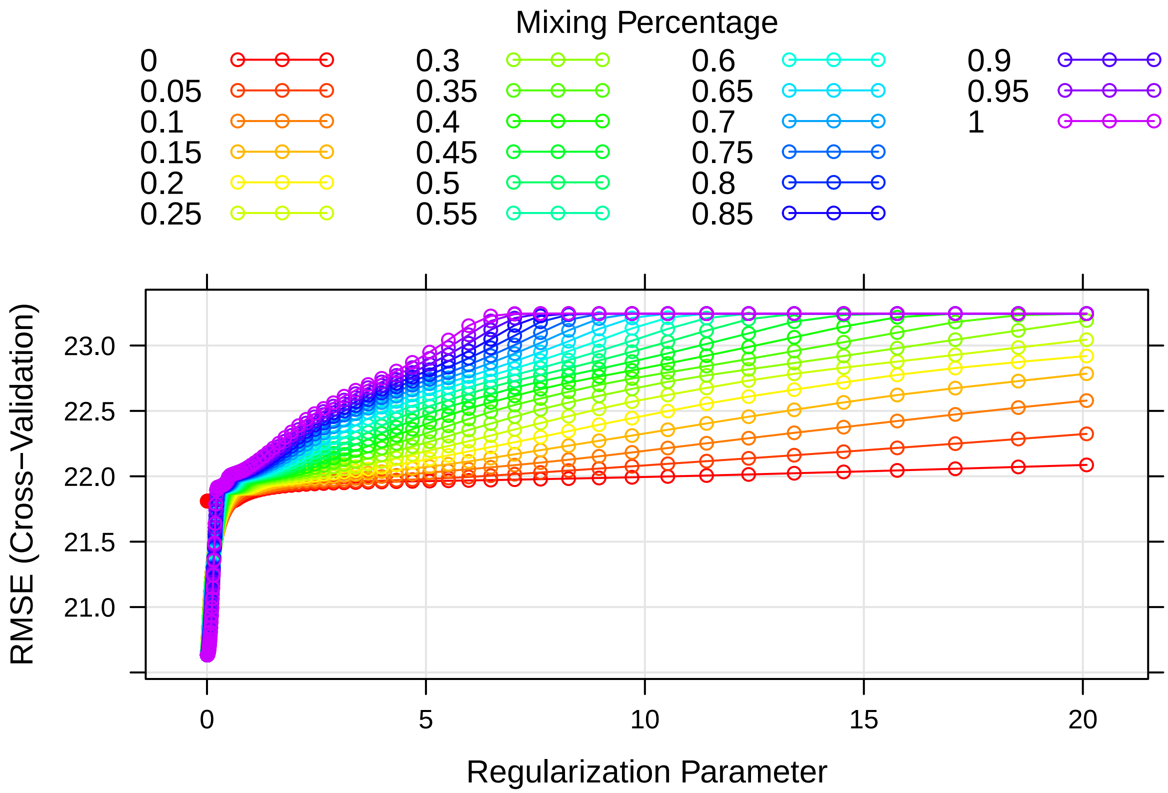
```
set.seed(7890)
ctrl1 <- trainControl(method = "cv", number = 10)
enet.caret.fit <- train(recovery_time ~ .,
                   data = train[,-1],
                   method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                         lambda = exp(seq(3, -5, length = 100))),
                  trControl = ctrl1)
```

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

```
enet.caret.fit$bestTune
```

```
##      alpha      lambda
## 1601   0.8 0.006737947
```

```r
myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))

plot(enet.caret.fit, par.settings = myPar)
```

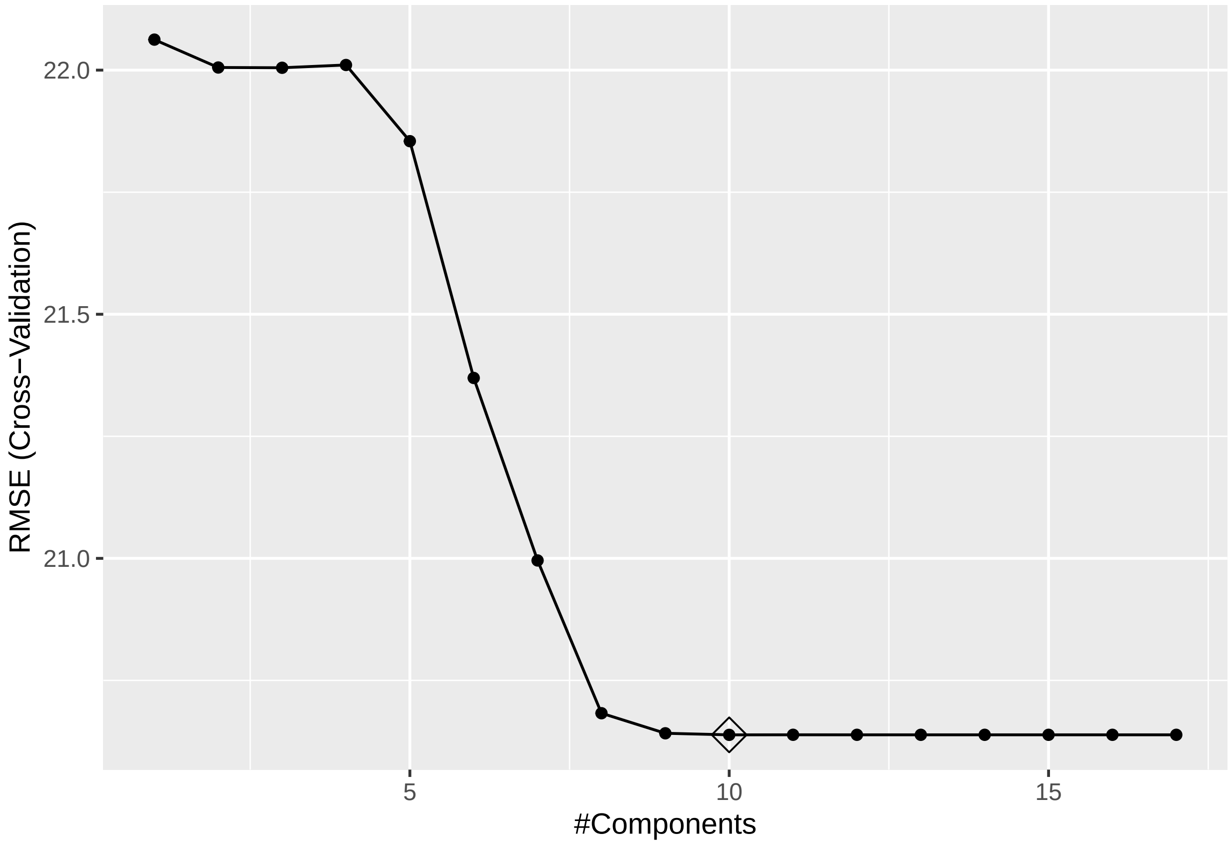


```r
# coefficients in the final model
coef(enet.caret.fit$finalModel, enet.caret.fit$bestTune$lambda)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)  -2.095563e+03
## age           1.580349e-01
## gender       -2.953746e+00
## race2         1.982764e+00
## race3        -1.187346e+00
## race4        -5.096701e-02
## smoking1      2.749906e+00
## smoking2      3.132109e+00
## height        1.222064e+01
## weight       -1.324918e+01
## bmi           3.988775e+01
## hypertension  2.238743e+00
## diabetes     -1.090213e+00
## SBP           7.491860e-02
## LDL           -4.495426e-02
## vaccine       -6.134406e+00
## severity       7.244184e+00
## studyB         4.767330e+00
```

partial least squares

```
set.seed(7890)
pls.fit <- train(x_train, y_train,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:17),
                 trControl = ctrl1,
                 preProcess = c("center", "scale"))

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



```
pls.fit$bestTune
```

```
##    ncomp
## 10    10
```

principal component regression

```
set.seed(7890)
pcr.fit <- train(x_train, y_train, method = "pcr",
                 tuneGrid = data.frame(ncomp = 1:18),
                 trControl = ctrl1,
                 preProcess = c("center", "scale"))
pcr.fit$bestTune
```

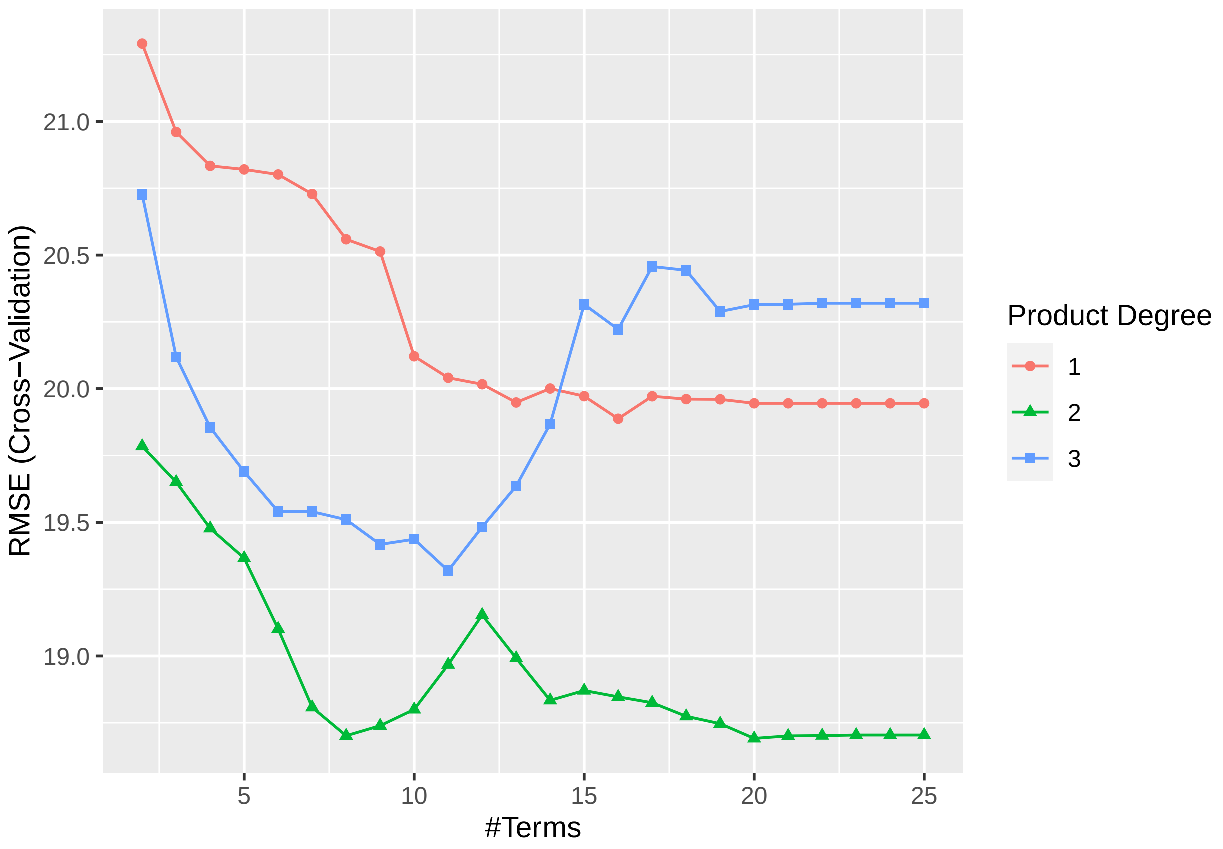
```
##    ncomp
## 17    17
```

MARS

```
ctrl1 <- trainControl(method = "cv", number = 10)
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 2:25)

set.seed(7890)
mars.fit <- train(x_train, y_train,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)
```

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

```
ggplot(mars.fit)
```



```
mars.fit$bestTune
```

```
##    nprune degree
## 43     20      2
```

```
coef(mars.fit$finalModel)
```

```
##                  (Intercept)                    h(bmi-31) 
##                   17.7932181                   -7.9070283 
##                    h(31-bmi)          h(bmi-31) * studyB 
##                    3.8322535                   14.3098948 
## h(height-158.8) * h(bmi-31) h(158.8-height) * h(bmi-31) 
##                    2.1950298                    1.1577839 
##                  h(bmi-25.7)                      vaccine 
##                    5.0549251                   -5.7519036 
##  h(weight-86.6) * h(bmi-31)                    h(bmi-34) 
##                   -2.5308124                   66.4133858 
##      h(bmi-31) * h(LDL-112)      h(bmi-31) * h(112-LDL) 
##                    0.1998164                    0.1866774 
##                       gender          h(bmi-34) * studyB 
##                   -3.1718582                   31.1590606 
##           race4 * h(bmi-34)     h(bmi-34) * hypertension 
##                  -54.8504806                  -33.4758125 
##           severity * studyB     h(bmi-22) * hypertension 
##                   12.2021025                    0.6530952
##    h(22-bmi) * hypertension  h(168.6-height) * severity
##                  11.2719170                   1.2404095
```

```
# partial dependence plot
#p1 <- pdp::partial(mars.fit, pred.var = c("recovery_time"), grid.resolution = 10) %>% autoplot()
#p1
```

GAM

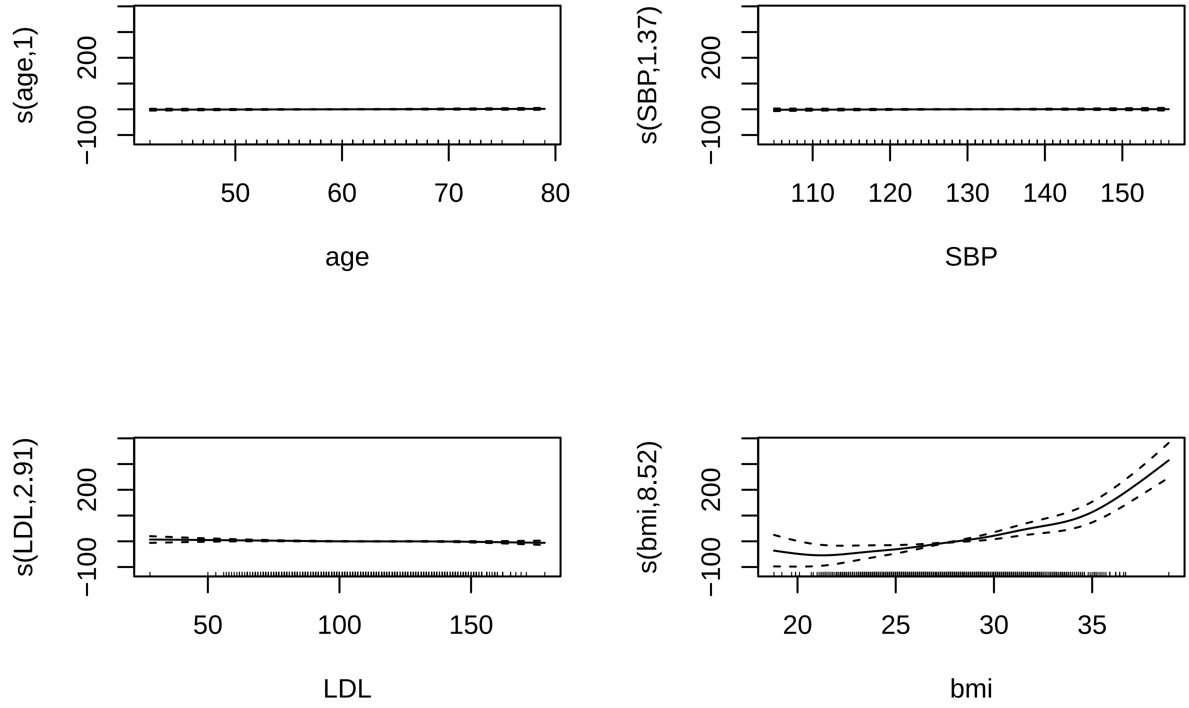
```
ctrl1 <- trainControl(method = "cv", number = 10)

set.seed(7890)
gam.fit <- train(x_train, y_train,
                 method = "gam",
                 tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE,FALSE)),
                 trControl = ctrl1)

gam.fit$bestTune
```

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

```
gam.fit$finalModel
```

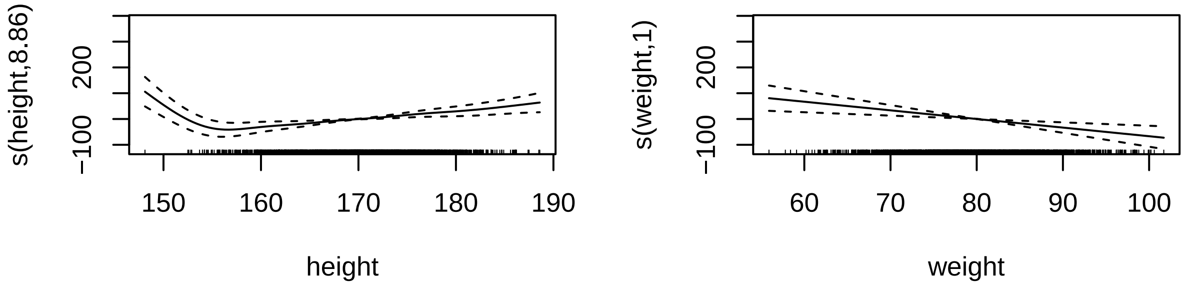
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + race3 + race4 + smoking1 + smoking2 + hypertension +
##     diabetes + vaccine + severity + studyB + s(age) + s(SBP) +
##     s(LDL) + s(bmi) + s(height) + s(weight)
##
## Estimated degrees of freedom:
## 1.00 1.37 2.91 8.52 8.86 1.00  total = 34.67
##
## GCV score: 368.0526
```

```
par(mfrow = c(2,2))
plot(gam.fit$finalModel)
```



```r
gam.pred <- predict(gam.fit, newdata = x_test)
sqrt(mean((y_test - gam.pred)^2))
```

```
## [1] 19.48927
```



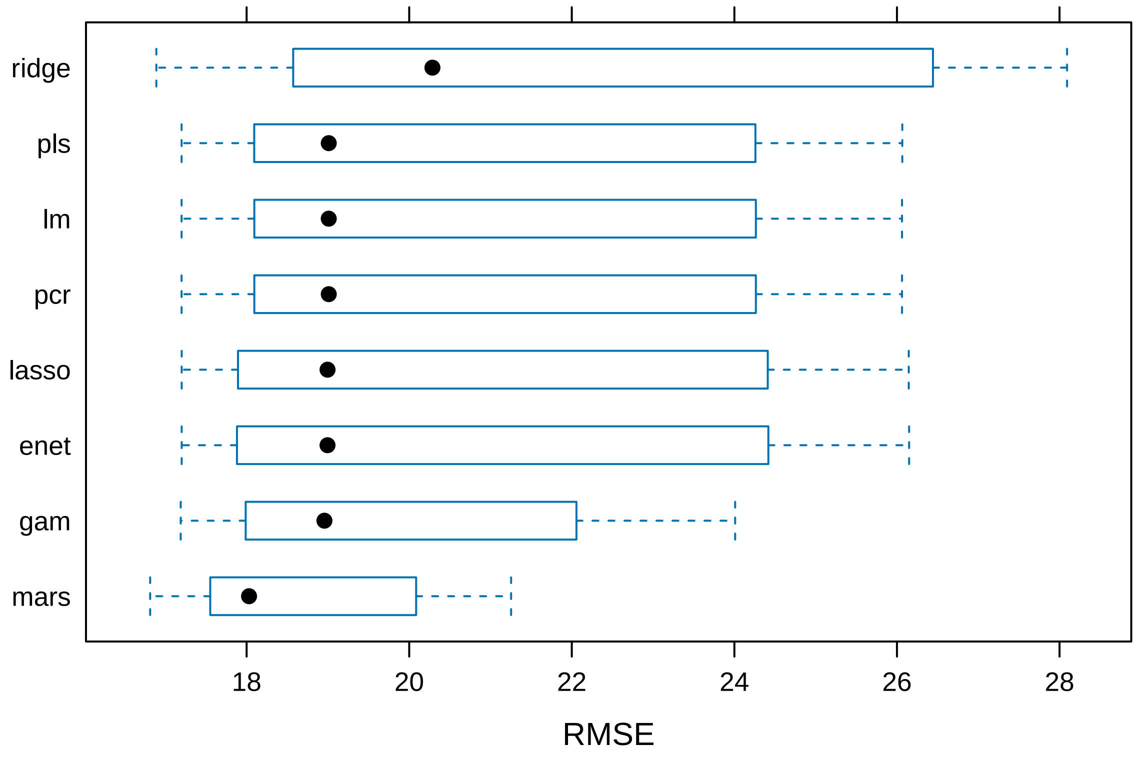
model comparison

```r
set.seed(7890)
lm.fit = train(x_train, y_train,
                method = "lm",
               trControl = ctrl1)
rs <- resamples(list(lasso = lasso.fit,
                     enet = enet.caret.fit,
                     pls = pls.fit,
                     mars = mars.fit,
                     ridge = ridge.fit,
                     lm = lm.fit,
                     pcr = pcr.fit,
                     gam = gam.fit))
summary(rs)
```

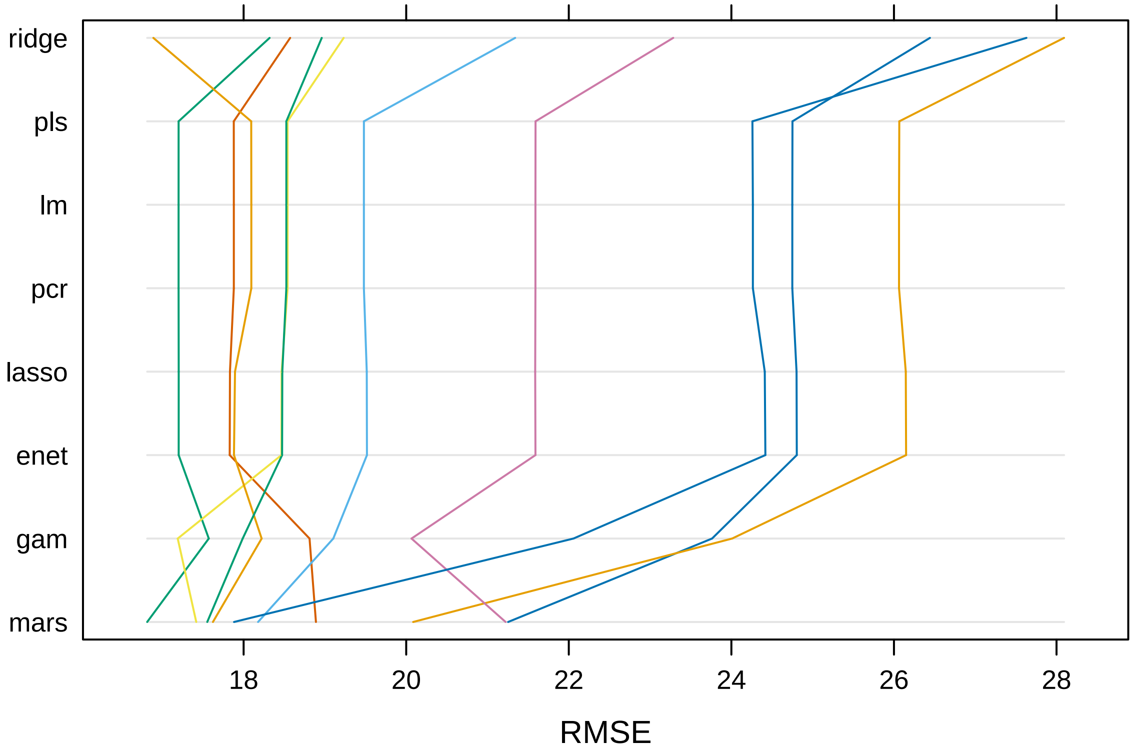
```
##
## Call:
## summary.resamples(object = rs)
##
## Models: lasso, enet, pls, mars, ridge, lm, pcr, gam
## Number of resamples: 10
##
```

```
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA’s
## lasso 12.38265 13.04949 13.61688 13.40714 13.87794 13.96867    0
## enet  12.37918 13.04710 13.61067 13.40453 13.87703 13.97025    0
## pls   12.46919 13.11531 13.73032 13.48824 13.93408 14.04729    0
## mars  11.65700 11.85062 12.47896 12.41716 12.95216 13.11735    0
## ridge 12.30361 12.79509 13.26380 13.40508 13.92406 14.48969    0
## lm    12.46861 13.11564 13.73120 13.48835 13.93216 14.04509    0
## pcr   12.46861 13.11564 13.73120 13.48835 13.93216 14.04509    0
## gam   12.19940 12.53137 12.84414 12.82412 13.12049 13.43754    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA’s
## lasso 17.20133 18.03735 18.99539 20.63304 23.70473 26.14554    0
## enet  17.20110 18.02735 18.99509 20.63289 23.71111 26.14939    0
## pls   17.20000 18.20147 19.01027 20.63853 23.59184 26.06573    0
## mars  16.81327 17.56997 18.02966 18.69160 19.78626 21.25342    0
## ridge 16.88956 18.66977 20.28569 21.87652 25.65353 28.09281    0
## lm    17.19980 18.20244 19.01026 20.63860 23.59574 26.06235    0
## pcr   17.19980 18.20244 19.01026 20.63860 23.59574 26.06235    0
## gam   17.18932 18.04665 18.95698 19.87799 21.55939 24.00970    0
##
## Rsquared
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA’s
## lasso 0.10419042 0.1629245 0.2628202 0.2345929 0.2767353 0.3910205    0
## enet  0.10429512 0.1630491 0.2625538 0.2344972 0.2766795 0.3905869    0
## pls   0.10406232 0.1617768 0.2644877 0.2352180 0.2770459 0.3937048    0
## mars  0.15638929 0.1962165 0.2747224 0.3417529 0.4887273 0.6601054    0
## ridge 0.02027266 0.1054438 0.1290328 0.1316712 0.1816491 0.2176850    0
## lm    0.10410951 0.1617287 0.2646628 0.2352123 0.2770389 0.3932775    0
## pcr   0.10410951 0.1617287 0.2646628 0.2352123 0.2770389 0.3932775    0
## gam   0.12924449 0.1899599 0.2679329 0.3019027 0.4226746 0.4886730    0
```

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



```
parallelplot(rs, metric = "RMSE")
```



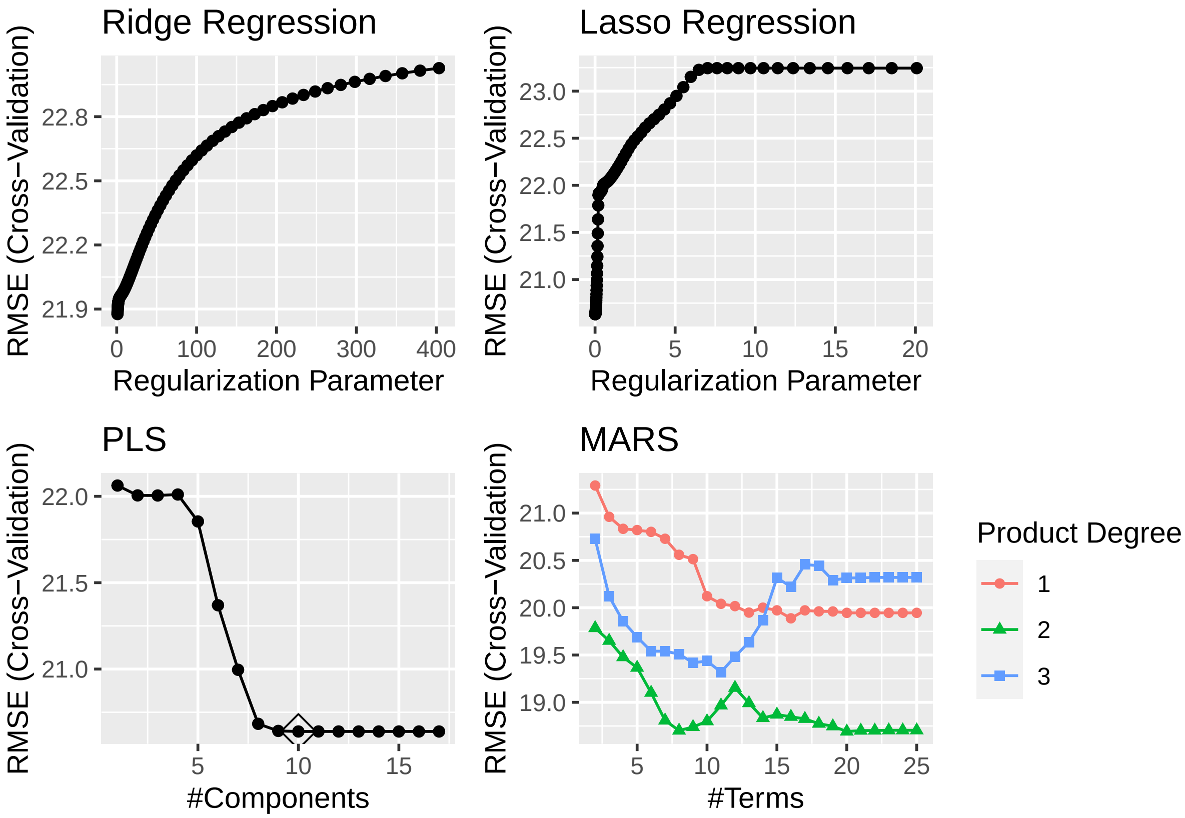
Final model test error

```
#Prediction on test data
mars_pred <- predict(mars.fit, newdata = x_test)
# test error
mars_test.error <- mean((mars_pred - y_test)^2)
mars_test.error
```

```
## [1] 273.1345
```

tunning parameter plots

```
p11 <- ggplot(ridge.fit, trans = "log") + ggtitle("Ridge Regression")
p12 <- ggplot(lasso.fit, trans = "log") + ggtitle("Lasso Regression")
p13 <- ggplot(pls.fit, highlight = TRUE) + ggtitle("PLS")
p14 <- ggplot(mars.fit) + ggtitle("MARS")
plot_grid2 <- p11 + p12 + p13 + p14 +
  plot_layout(ncol = 2, nrow = 2)

plot_grid2
```


Ridge Regression
RMSE (Cross−Validation)
22.8
22.5
22.2
21.9
0
100
200
300
400
Regularization Parameter
Lasso Regression
RMSE (Cross−Validation)
23.0
22.5
22.0
21.5
21.0
0
5
10
15
20
Regularization Parameter
PLS
RMSE (Cross−Validation)
22.0
21.5
21.0
5
10
15
#Components
MARS
RMSE (Cross−Validation)
21.0
20.5
20.0
19.5
19.0
5
10
15
20
25
#Terms
Product Degree
1
2
3