# Data Science II Midterm Project

Huanyu Chen

```
library(tidyverse)
library(ggridges)
library(corrplot)
library(ggcorrplot)
library(pheatmap)
library(rsample)
library(lattice)
library(caret)
library(pls)
library(rpart)
library(rpart.plot)

load("recovery.Rdata")
dat = as_tibble(dat) |>
  na.omit() |>
  mutate(gender = factor(gender),
         hypertension = factor(hypertension),
         diabetes = factor(diabetes),
         vaccine = factor(vaccine),
         severity = factor(severity),
         race = factor(race),
         smoking = factor(smoking)) |>
  select(- id) |>
  relocate(recovery_time)
set.seed(11)
dat_split = initial_split(dat, prop = 0.8)
training = training(dat_split)
testing = testing(dat_split)
xtrain = model.matrix(recovery_time ~ ., training)[,-1]
ytrain = training$recovery_time
xtest = model.matrix(recovery_time ~ ., testing)[,-1]
ytest = testing$recovery_time
```

## Exploratory Analysis and Data Visualization

### Exploratory Analysis

In this dataset, `age`, `height`, `weight`, `bmi`, `SBP`, `LDL`, and `recovery_time` are continuous variables.

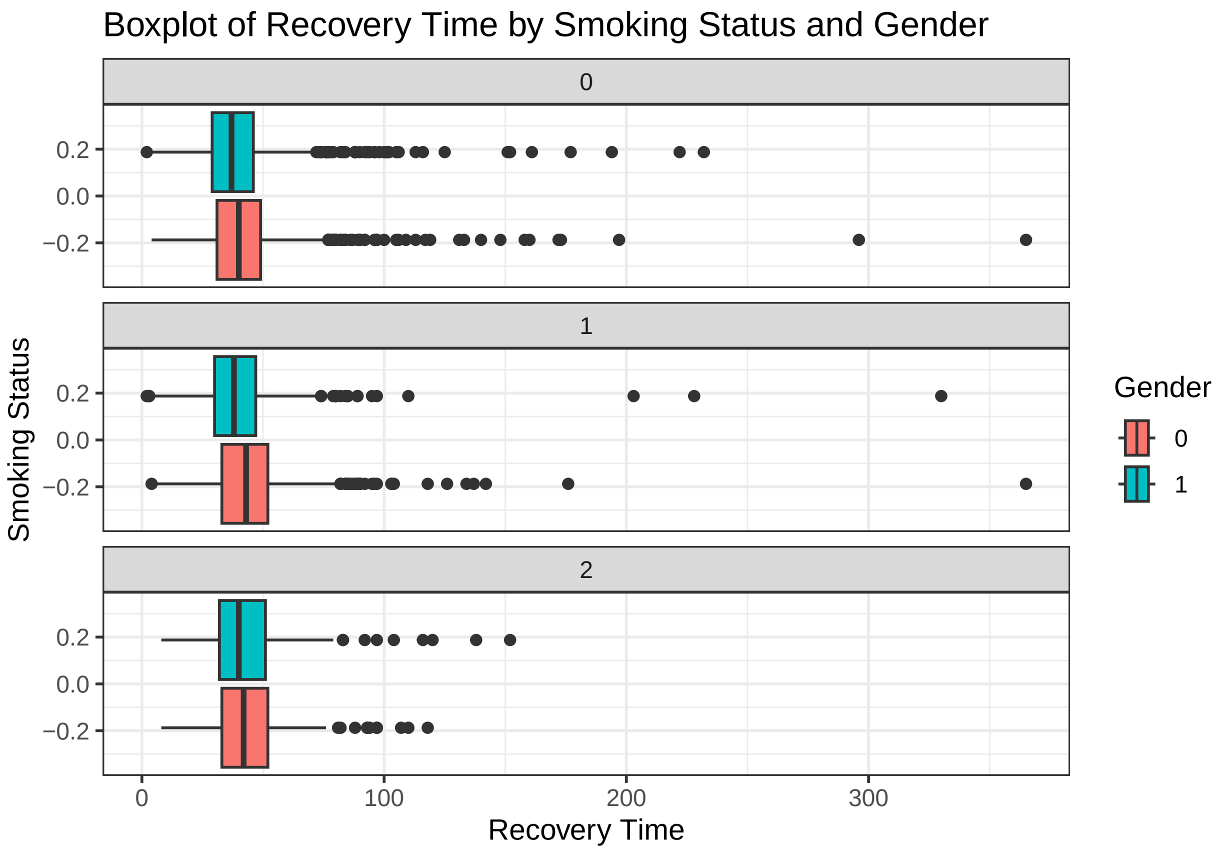
```
continuous_vars <- dat[, c("age", "height", "weight", "bmi",
                           "SBP", "LDL", "recovery_time")]
summary(continuous_vars)
```

```
##       age            height          weight            bmi       
##  Min.   :42.0   Min.   :147.8   Min.   : 55.90   Min.   :18.80  
##  1st Qu.:57.0   1st Qu.:166.0   1st Qu.: 75.20   1st Qu.:25.80  
##  Median :60.0   Median :169.9   Median : 79.80   Median :27.65  
##  Mean   :60.2   Mean   :169.9   Mean   : 79.96   Mean   :27.76  
##  3rd Qu.:63.0   3rd Qu.:173.9   3rd Qu.: 84.80   3rd Qu.:29.50  
##  Max.   :79.0   Max.   :188.6   Max.   :103.70   Max.   :38.90  
##       SBP             LDL        recovery_time   
##  Min.   :105.0   Min.   : 28.0   Min.   :  2.00  
##  1st Qu.:125.0   1st Qu.: 97.0   1st Qu.: 31.00  
##  Median :130.0   Median :110.0   Median : 39.00  
##  Mean   :130.5   Mean   :110.5   Mean   : 42.17  
##  3rd Qu.:136.0   3rd Qu.:124.0   3rd Qu.: 49.00  
##  Max.   :156.0   Max.   :178.0   Max.   :365.00  
```

### Boxplot of Recovery Time by Smoking Status and Gender

Our analysis reveals a notable trend: across all smoking statuses, females (`gender` = 0) consistently exhibit longer recovery times compared to males. Interestingly, individuals who had never smoked had more outliers on the right side of the boxplot, suggesting a longer recovery time. This counter-intuitive finding suggests that individuals with healthier lifestyles, such as non-smokers, paradoxically require more time to recover from COVID-19.

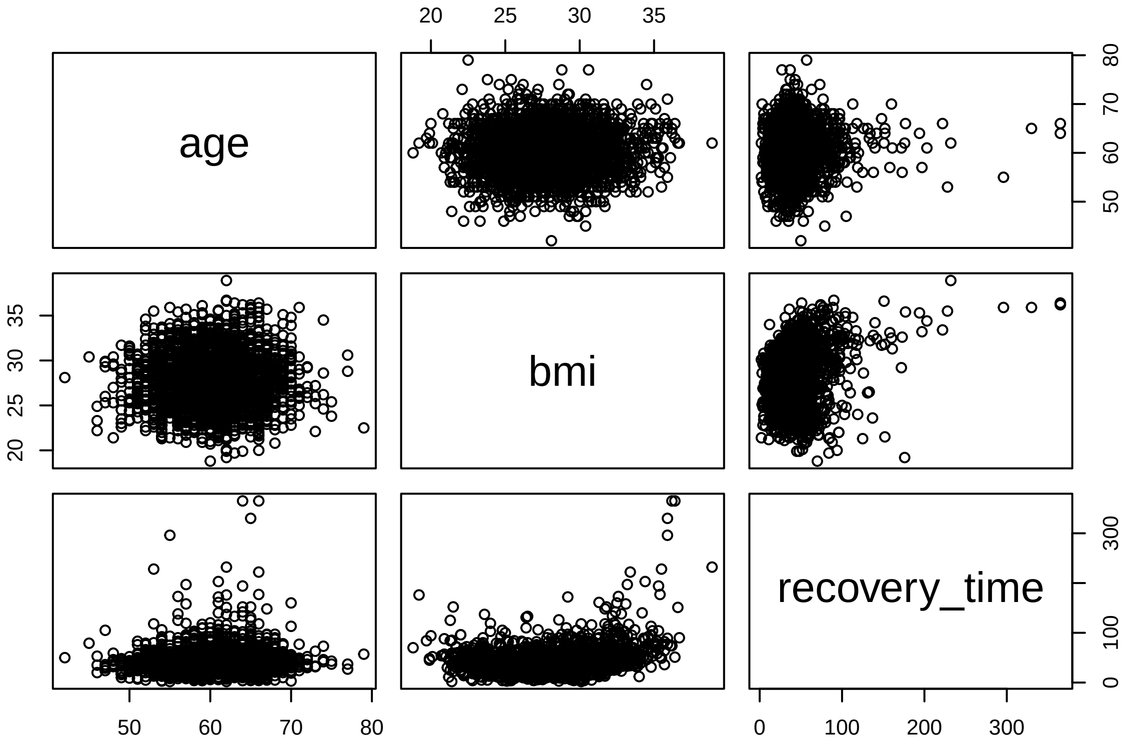
```r
ggplot(dat, aes(x = recovery_time, fill = factor(gender))) +
  geom_boxplot() +
  labs(title = "Boxplot of Recovery Time by Smoking Status and Gender",
       x = "Recovery Time", y = "Smoking Status",
       fill = "Gender") +
  facet_wrap(~factor(smoking), ncol = 1) +
  theme_bw()
```




### Pairs

Our exploration of the variables age, BMI, and recovery time reveals no clear linear relationships among them. It implies that other complex factors beyond these variables might be influencing the recovery time from COVID-19, highlighting the complexity of analysis about recovery time.

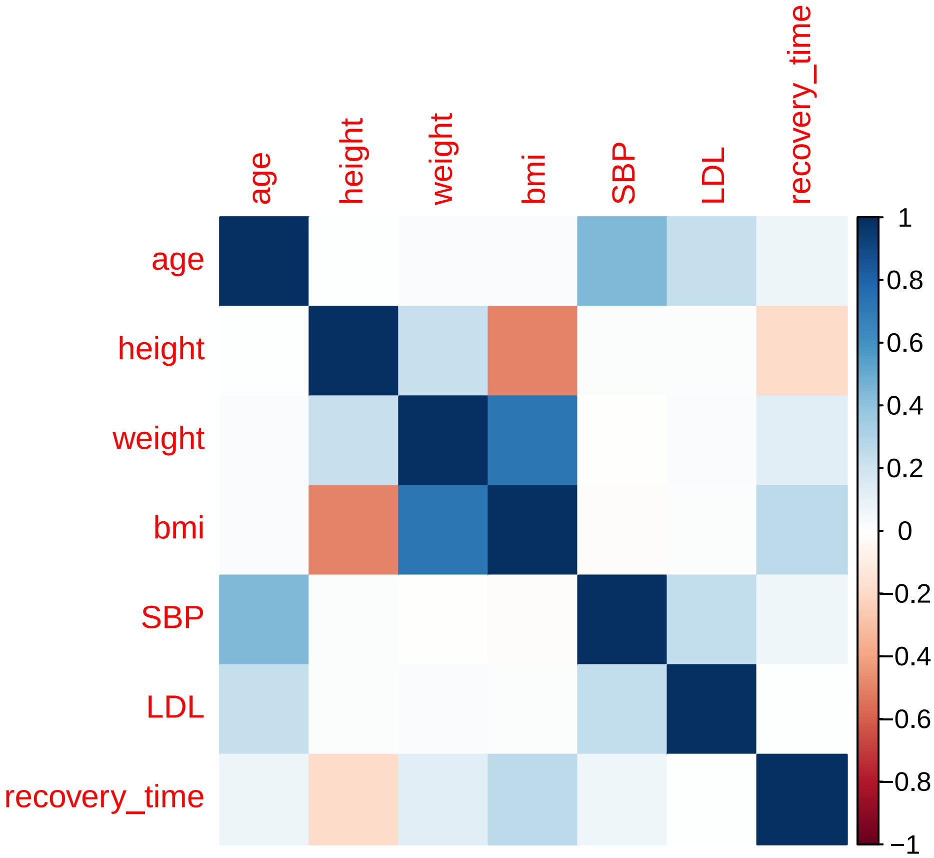
```
pairs(dat[, c("age", "bmi", "recovery_time")])
```




### Correlation Table

The correlation analysis conducted on variables including "height," "weight," and "bmi" suggests a strong positive correlation among these attributes, which aligns with our common understanding. However, no significant correlations were observed between these attributes and other variables in the dataset.

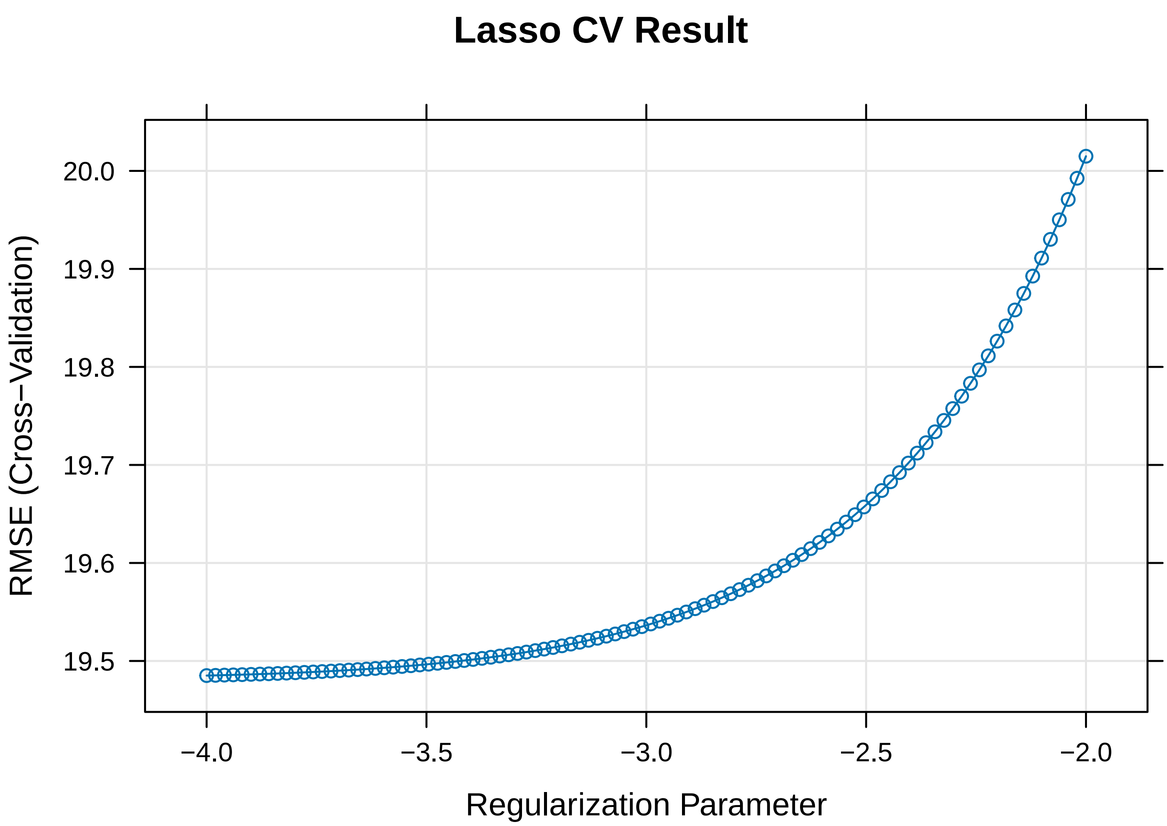
```
correlation_matrix <- cor(dat[, c("age", "height", "weight", "bmi",
                                  "SBP", "LDL", "recovery_time")])
corrplot::corrplot(correlation_matrix, method = "color")
```


age
height
weight
bmi
SBP
LDL
recovery_time
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

# Model Training

## Lasso

```
set.seed(11)

ctrl = trainControl(method = 'cv', number = 10)
ctrl_1se = trainControl(method = 'cv', number = 10, selectionFunction =  'oneSE')

lasso.fit = train(recovery_time ~ ., data = training,
                  method = 'glmnet',
                  tuneGrid = expand.grid(alpha = 1,
                                         lambda = exp(seq(-4, -2, length = 100))),
                  trControl = ctrl)

plot(lasso.fit, xTrans = log, main = "Lasso CV Result")
```




```
# selected lambda
lasso.fit$bestTune$lambda
```

```
## [1] 0.01831564
```

```
# coefficients
coef(lasso.fit$finalModel, s = lasso.fit$bestTune$lambda)
```

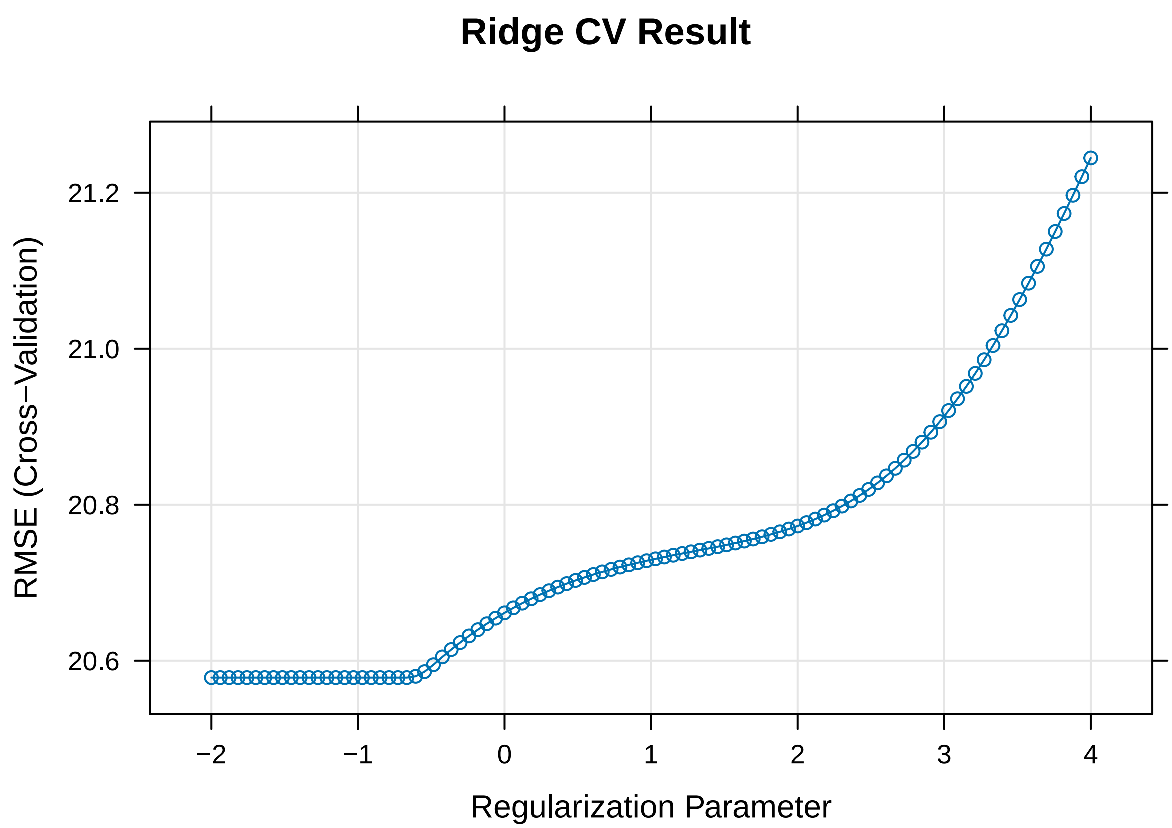
```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)   -1.865676e+03
## age            1.905529e-01
## gender1       -2.272167e+00
## race2          4.069257e+00
## race3         -5.713863e-01
## race4          4.209219e-01
## smoking1       2.232083e+00
## smoking2       4.229749e+00
## height         1.092898e+01
## weight        -1.186704e+01
## bmi            3.571072e+01
## hypertension1  3.389890e+00
## diabetes1     -1.766480e+00
## SBP           -3.790833e-03
## LDL           -2.479261e-02
## vaccine1      -6.318510e+00
## severity1      9.121546e+00
## studyB         4.617454e+00
```

```r
# num of predictors
sum(lasso.fit$coefname != 0)
```

```
## [1] 17
```

## Ridge

```r
set.seed(11)

ctrl = trainControl(method = 'cv', number = 10)
ctrl_1se = trainControl(method = 'cv', number = 10, selectionFunction =  'oneSE')

ridge.fit = train(recovery_time ~ ., data = training,
             method = 'glmnet',
             tuneGrid = expand.grid(alpha = 0,
                                    lambda = exp(seq(-2, 4, length = 100))),
             trControl = ctrl)

plot(ridge.fit, xTrans = log, main = "Ridge CV Result")
```

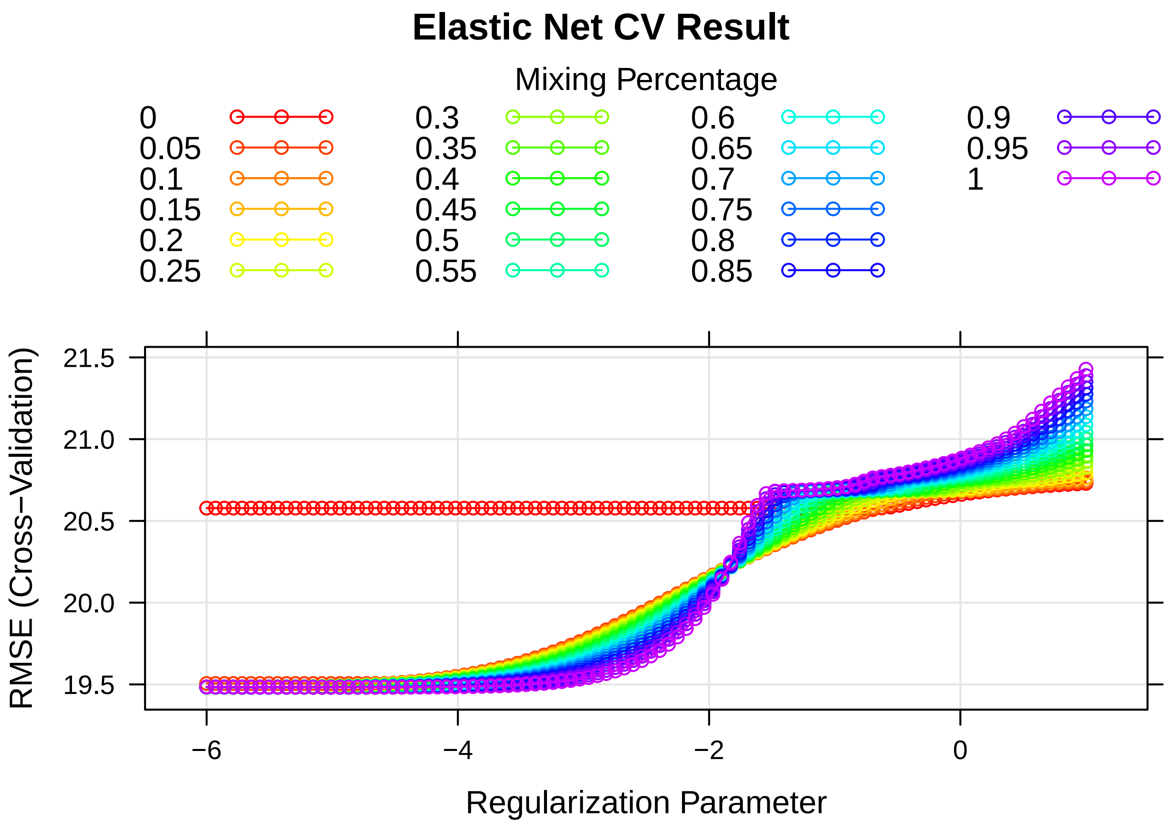



```
# selected lambda
ridge.fit$bestTune$lambda
```

```
## [1] 0.5134171
```

### Elastic Net

```
set.seed(11)
ctrl = trainControl(method = 'cv', number = 10)
ctrl_1se = trainControl(method = 'cv', number = 10, selectionFunction =  'oneSE')

enet.fit = train(recovery_time ~ ., data = training,
             method = 'glmnet',
             tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                    lambda = exp(seq(-6, 1, length = 100))),
             trControl = ctrl)

myCol = rainbow(25)
myPar = list(superpose.symbol = list(col = myCol), superpose.line = list(col = myCol))

plot(enet.fit, par.settings = myPar, xTrans = log, main = "Elastic Net CV Result")
```




```r
# selected alpha and lambda
enet.fit$bestTune
```

```
##     alpha      lambda
## 401   0.2 0.002478752
```

```r
# coefficients
coef(enet.fit$finalModel, s = enet.fit$bestTune$lambda)
```

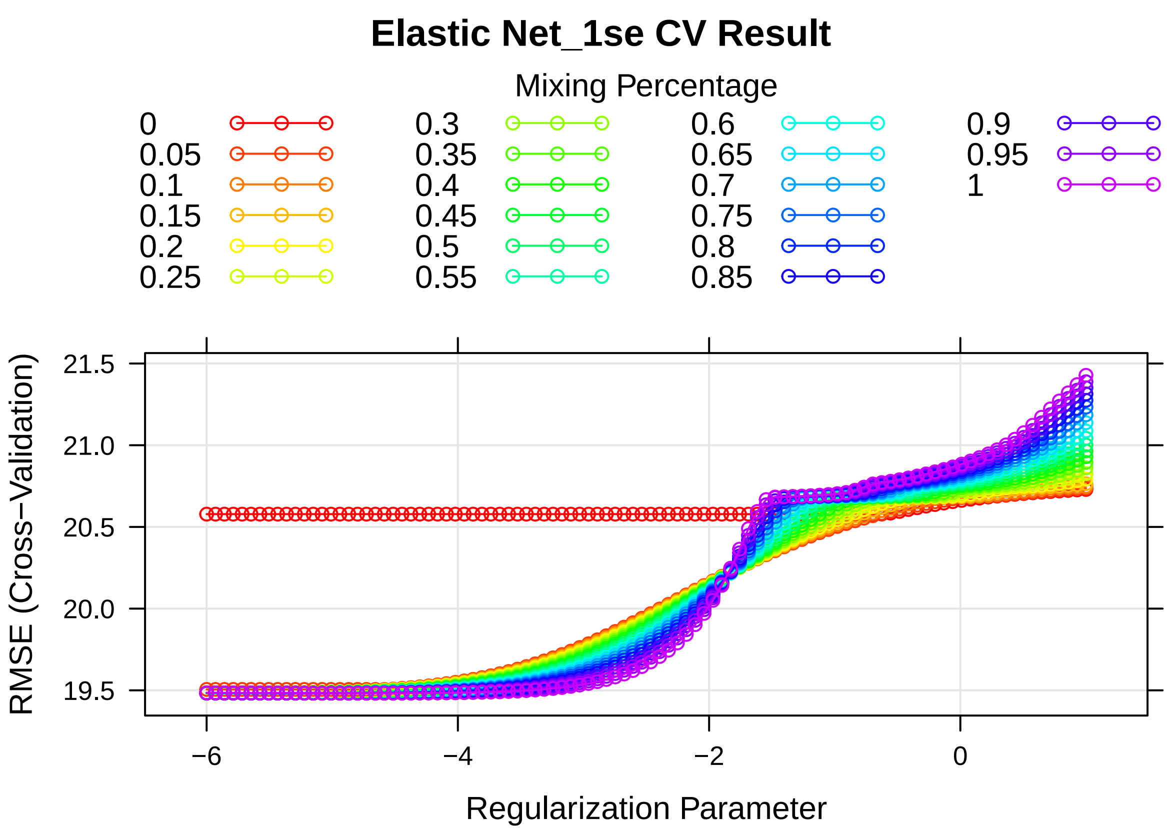
```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)    -1.952199e+03
## age             1.972764e-01
## gender1        -2.306583e+00
## race2           4.139347e+00
## race3          -6.070652e-01
## race4           5.040683e-01
## smoking1        2.287723e+00
## smoking2        4.307013e+00
## height          1.144672e+01
## weight         -1.241459e+01
## bmi             3.728410e+01
## hypertension1  3.568559e+00
## diabetes1      -1.802445e+00
## SBP            -1.688603e-02
## LDL            -2.603621e-02
```

```
## vaccine1       -6.333775e+00
## severity1       9.177936e+00
## studyB          4.637170e+00
```

```r
# num of predictors
sum(enet.fit$coefname != 0)
```

```
## [1] 17
```

```r
# applying 1se rule
set.seed(11)
enet.fit.1se = train(recovery_time ~ ., data = training,
                     method = 'glmnet',
                     tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                            lambda = exp(seq(-6, 1, length = 100))),
                     trControl = ctrl_1se)

plot(enet.fit.1se, par.settings = myPar, xTrans = log, main = "Elastic Net_1se CV Result")
```

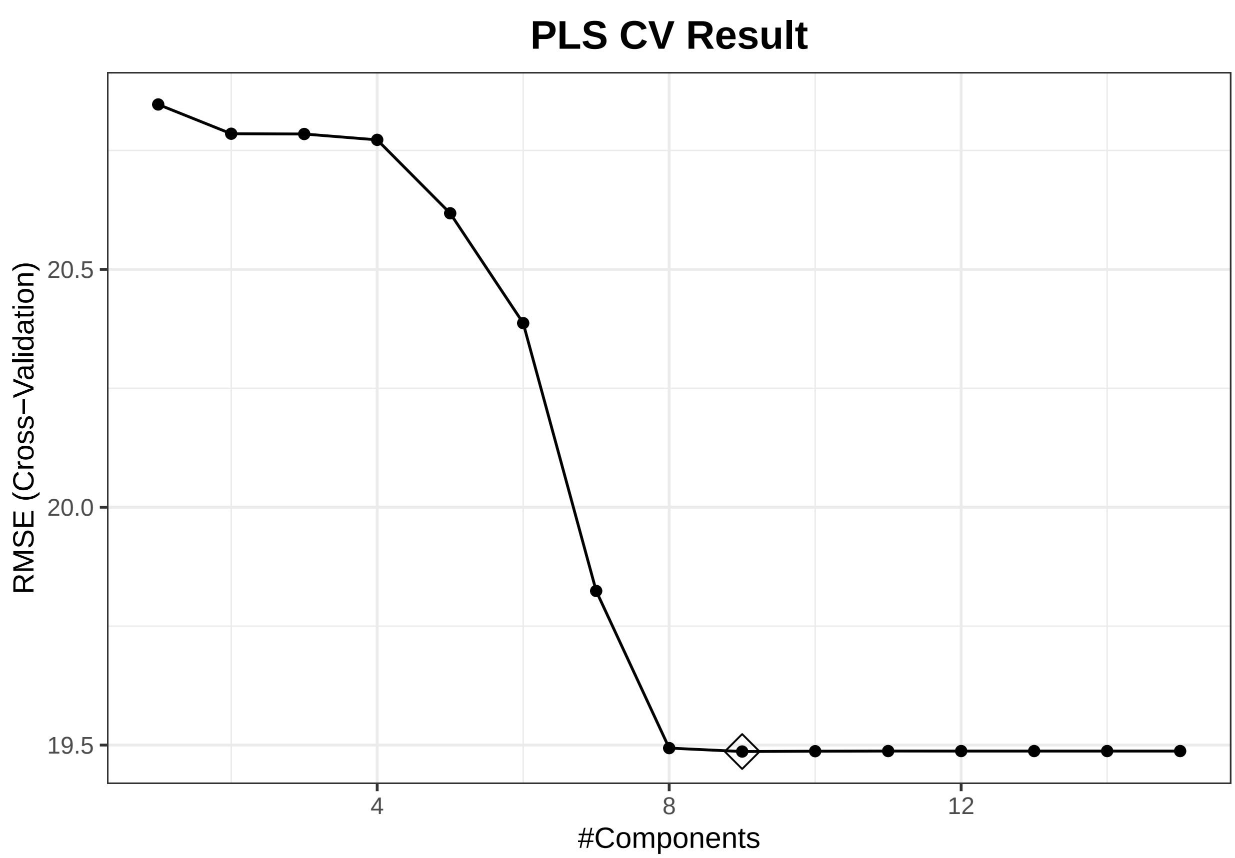



```r
# selected alpha and lambda
enet.fit.1se$bestTune
```

```
##     alpha    lambda
## 170  0.05 0.3258845
```

## PLS

```r
set.seed(11)

#pls = plsr(recovery_time ~ ., data = training, scale = TRUE, validation = 'CV')
#summary(pls)
#validationplot(pls, val.type = 'MSEP', legendpos = 'topright')
#cv.mse = RMSEP(pls)
#ncomp.cv = which.min(cv.mse$val[1,,]) - 1

pls.fit <- train(recovery_time ~ ., data = training,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:15),
                 trControl = ctrl,
                 preProcess = c("center", "scale"))

ggplot(pls.fit, highlight = TRUE) +
  theme_bw() +
  labs(title = "PLS CV Result") +
  theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5))
```




```r
summary(pls.fit)
```

```
## Data:    X dimension: 2400 17
##  Y dimension: 2400 1
```

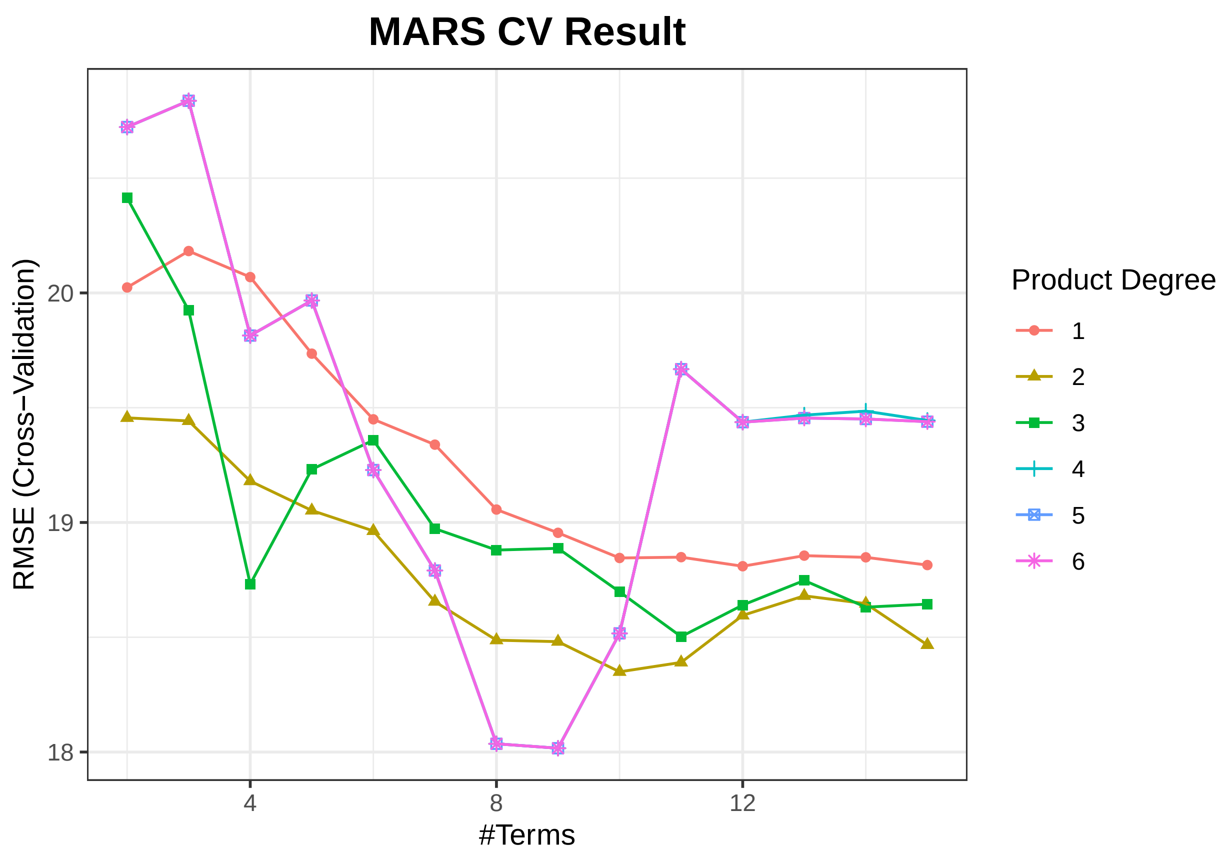
```
## Fit method: oscorespls
## Number of components considered: 9
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           9.704    17.88    28.92    34.88    38.00    42.19    44.12
## .outcome   12.363    13.29    13.38    13.62    14.58    15.86    22.82
##           8 comps  9 comps
## X           48.71    54.05
## .outcome    25.05    25.10
```

```
pls.fit$bestTune
```

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

### MARS

```
set.seed(11)
mars_grid = expand.grid(degree = 1:6, nprune = 2:15)
ctrl = trainControl(method = 'cv', number = 10)

mars.fit = train(xtrain, ytrain,
                 method = "earth",
                 tuneGrid = mars_grid,
                 trControl = ctrl)

ggplot(mars.fit) +
  theme_bw() +
  labs(title = "MARS CV Result") +
  theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5))
```




```r
# fit of the model
mars.fit$bestTune
```

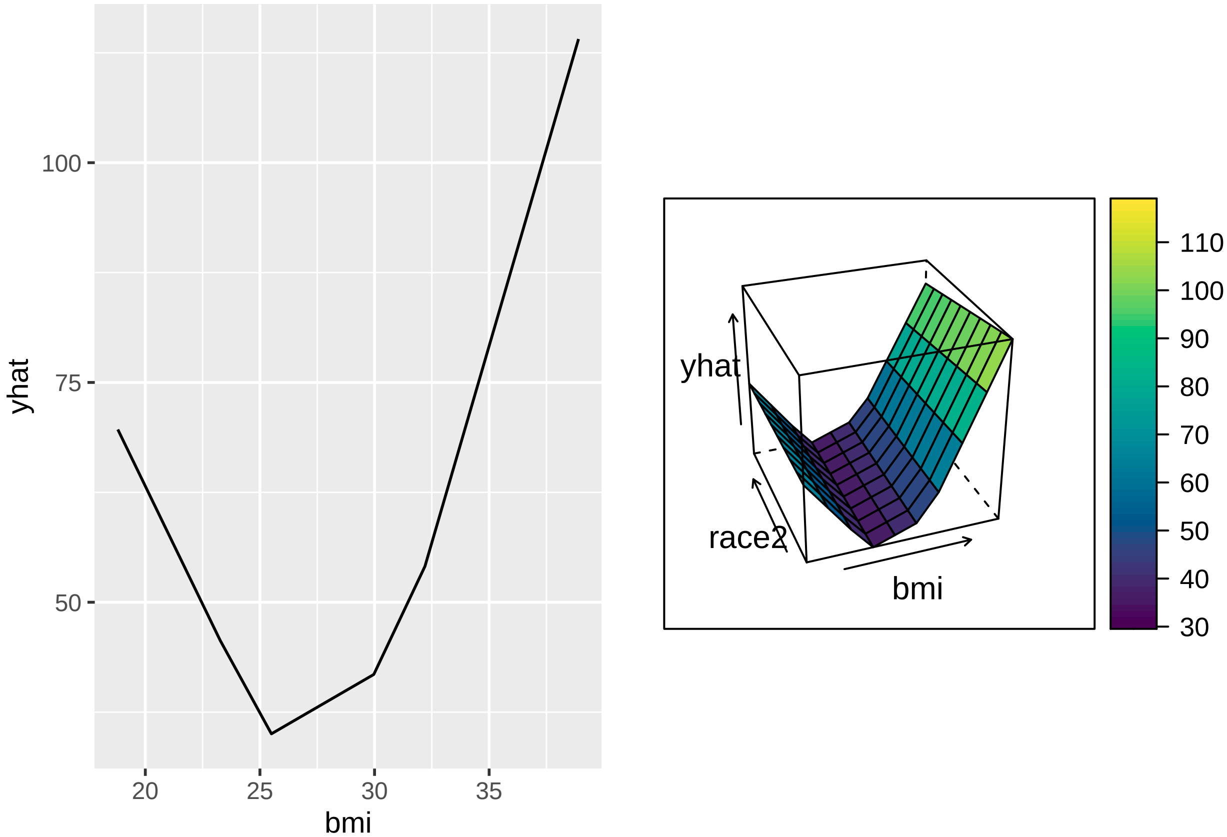
```
##    nprune degree
## 50      9      4
```

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

```
##                            (Intercept)                            h(31-bmi)
##                              6.9166504                            5.3725588
## h(161.6-height) * h(bmi-31) * studyB                           h(bmi-25.3)
##                              2.9896206                            6.8844160
##                               vaccine1           race2 * h(bmi-31) * studyB
##                             -5.7338813                         -523.1860845
##         h(bmi-31) * h(LDL-88) * studyB     age * race2 * h(bmi-31) * studyB
##                              0.2238751                            8.6160130
##                     severity1 * studyB
##                             18.1026072
```

```r
p1 = pdp::partial(mars.fit, pred.var = c("bmi"), grid.resolution = 10) |> autoplot()

p2 = pdp::partial(mars.fit, pred.var = c("bmi", "race2"),
grid.resolution = 10) |>
pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
screen = list(z = 20, x = -60))
```

```r
gridExtra::grid.arrange(p1, p2, ncol = 2)
```




## GAM

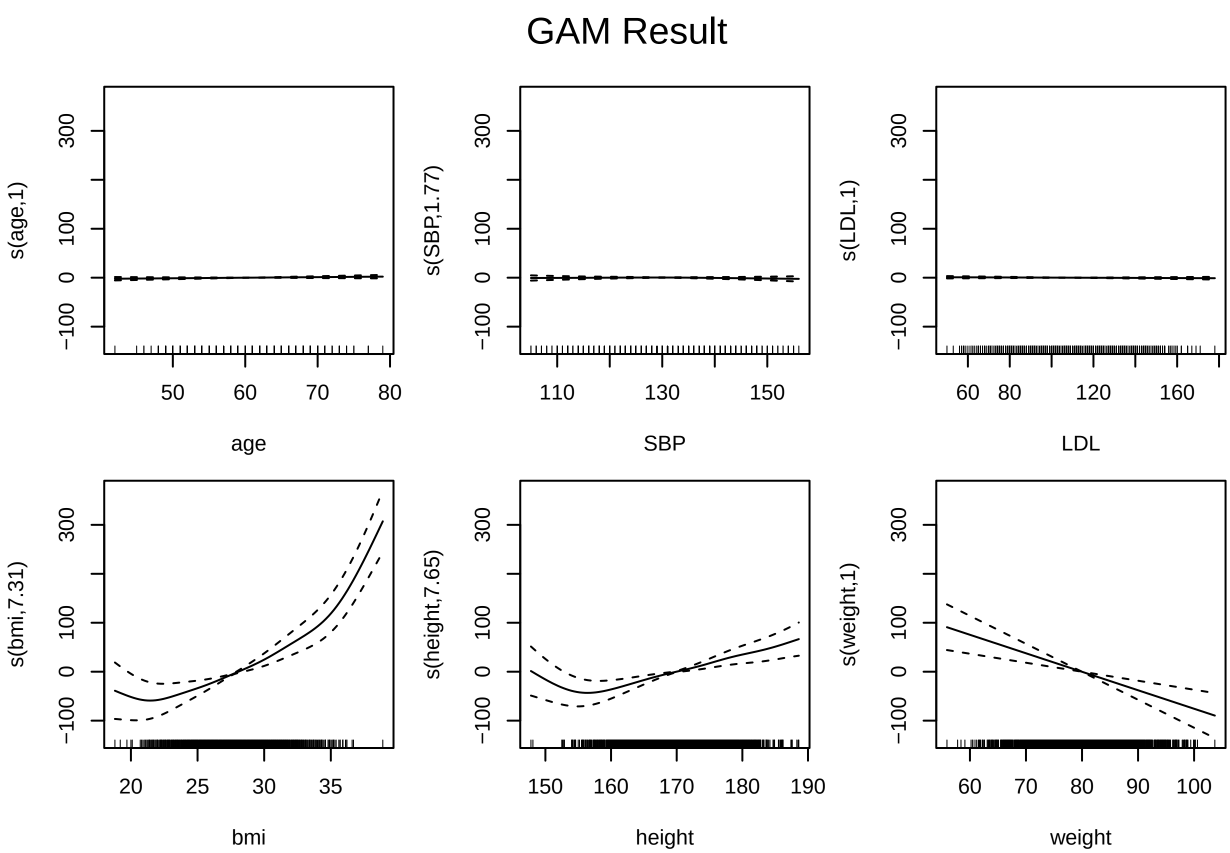
```r
set.seed(11)
gam.fit = train(xtrain, ytrain,
                method = "gam",
                trControl = ctrl)
```

```
## Loading required package: mgcv

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

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

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender1 + race2 + race3 + race4 + smoking1 + smoking2 +
##     hypertension1 + diabetes1 + vaccine1 + severity1 + studyB +
##     s(age) + s(SBP) + s(LDL) + s(bmi) + s(height) + s(weight)
##
## Estimated degrees of freedom:
## 1.00 1.77 1.00 7.31 7.65 1.00  total = 31.73
##
## GCV score: 340.2157
```

```r
par(oma = c(0, 0, 3, 0))
par(mar = c(4, 4, 1, 1), mfrow = c(2, 3))
plot(gam.fit$finalModel)
mtext("GAM Result", side = 3, line = 0.5, outer = TRUE, cex = 1.2)
```




### Model Comparation

```r
library(patchwork)
res = resamples(list(lasso = lasso.fit,
                     ridge = ridge.fit,
                     enet = enet.fit,
                     enet_1se = enet.fit.1se,
                     pls = pls.fit,
                     mars = mars.fit,
                     gam = gam.fit#,
                     ))$value |>
  tibble() |>
  janitor::clean_names() |>
  select(- resample) |>
  pivot_longer(
    everything(),
    names_to = c(".value", "metric"),
    names_pattern = "(.*)_(.*)"
  ) |>
  pivot_longer(c(2:8), names_to = "model", values_to = "result")

plot_rmse = res |>
  filter(metric == "rmse") |>
  ggplot(aes(x = model, y = result, fill = model)) +
  geom_boxplot(alpha = 0.5) +
  labs(y = "RMSE") +
  theme_minimal() +
  guides(fill = guide_legend("Model"))

plot_r_squared = res |>
  filter(metric == "rsquared") |>
  ggplot(aes(x = model, y = result, fill = model)) +
  geom_boxplot(alpha = 0.5) +
  labs(y = "R squared") +
  theme_minimal() +
  guides(fill = guide_legend("Model"))

plot_mae = res |>
  filter(metric == "mae") |>
  ggplot(aes(x = model, y = result, fill = model)) +
  geom_boxplot(alpha = 0.5) +
  labs(y = "MAE") +
  theme_minimal() +
  guides(fill = guide_legend("Model"))

final_plot = plot_rmse + plot_r_squared + plot_mae +
  plot_layout(ncol = 3) +
  plot_annotation(title = "Performance by Models and Metrics",
                  theme = theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5)))
```

```r
final_plot
```

Performance by Models and Metrics
RMSE
25
20
15
enet
enet_1se
gam
lasso
mars
pls
ridge
model
Model
R squared
0.6
0.4
0.2
MAE
14
13
12
11