Data Science II Midterm Project

Huanyu Chen

```
library(ggplot2)
library(tidyverse)
library(corrplot)

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)
```

Exploratory Analysis and Data Visualization

Exploratory Analysis

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

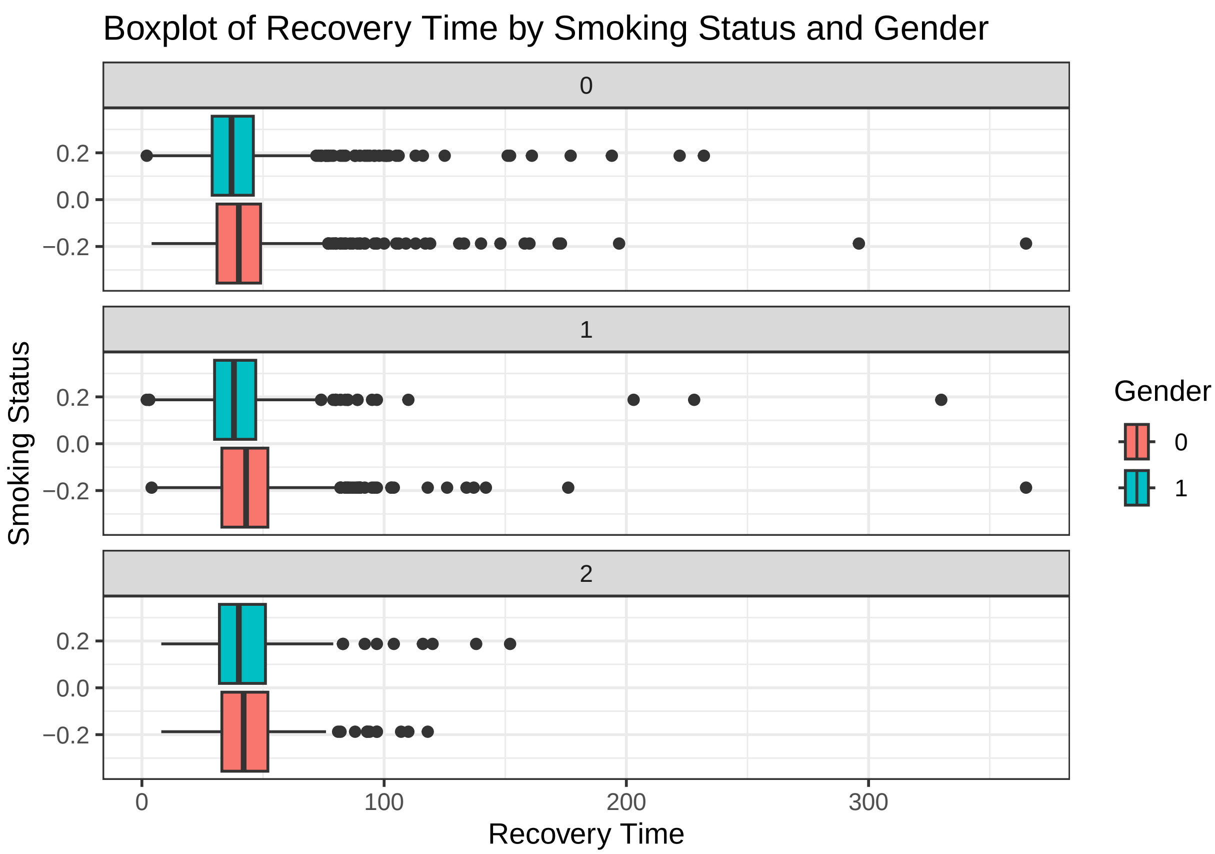
```
continuous_vars <- dat[, c("age", "height", "weight", "bmi",
                           "SBP", "LDL", "recovery_time")]
summary(continuous_vars)
```

```
##       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.

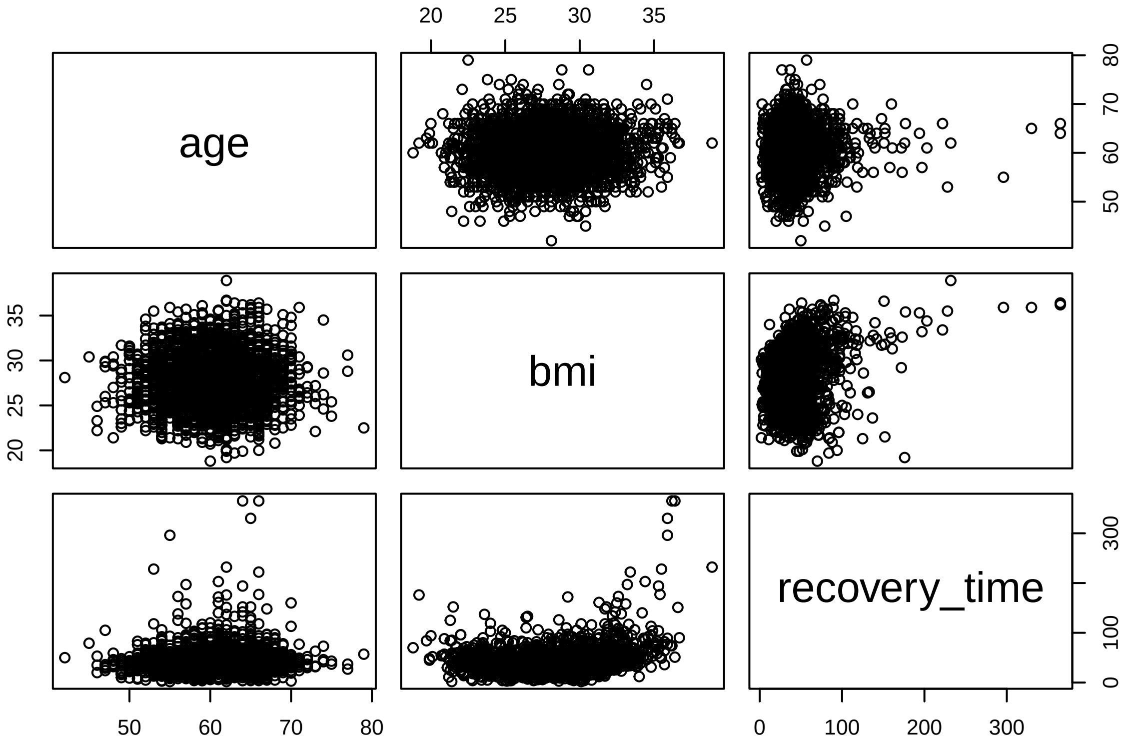
```
ggplot(dat, aes(x = recovery_time, fill = factor(gender))) +
  geom_boxplot() +
  labs(title = "Boxplot of Recovery Time by Smoking Status and Gender",
       x = "Recovery Time", y = "Smoking Status",
       fill = "Gender") +
  facet_wrap(~factor(smoking), ncol = 1) +
  theme_bw()
```



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.

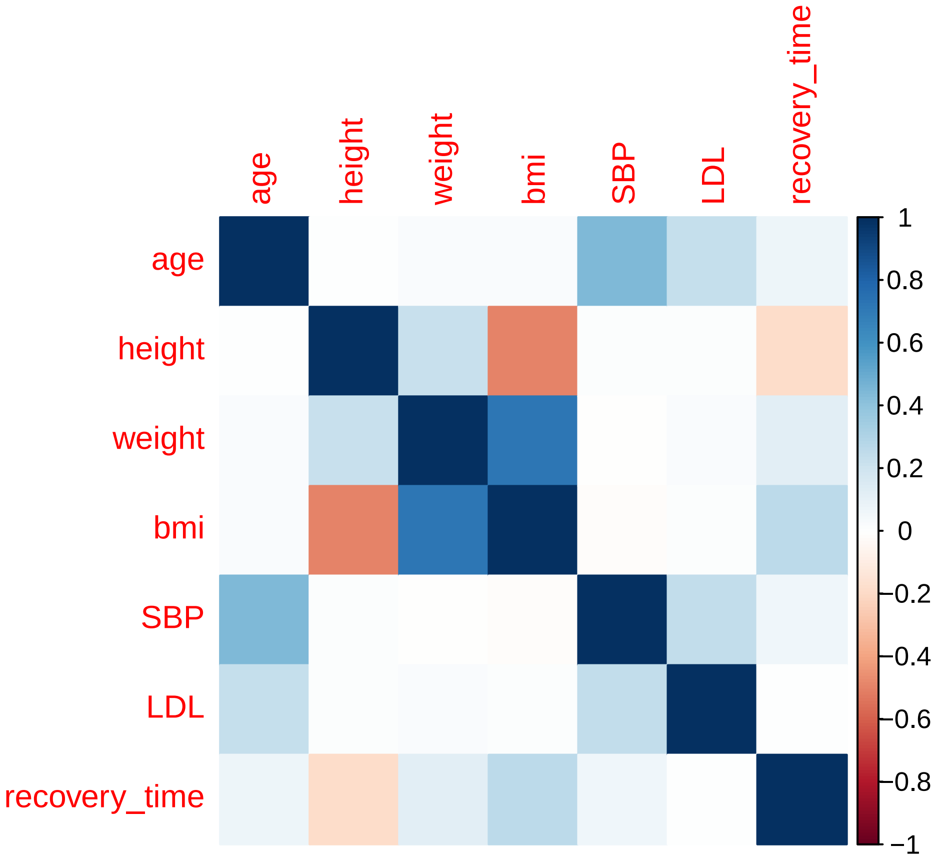
```
pairs(dat[, c("age", "bmi", "recovery_time")])
```



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

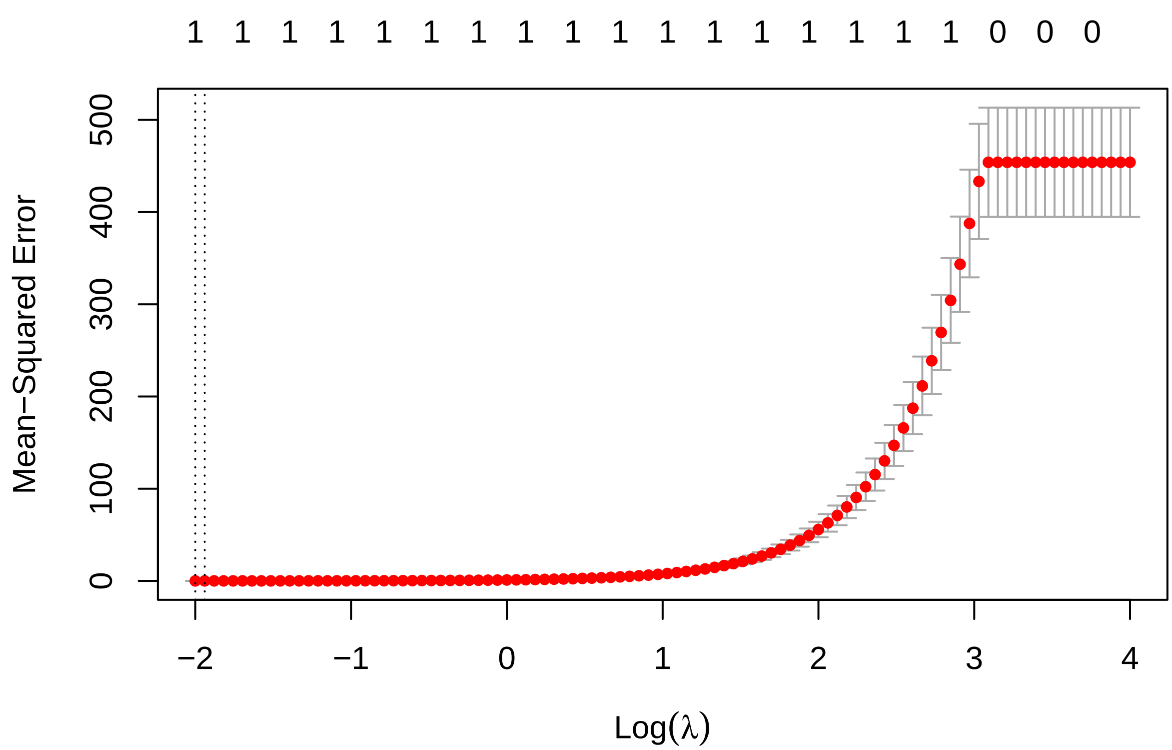
Model Training

Lasso

```r
library(caret)
library(glmnet)
library(pls)

set.seed(11)
indexTrain <- createDataPartition(y = dat$recovery_time, p = 0.8, list = FALSE)
trainData <- dat[indexTrain, ]
testData <- dat[-indexTrain, ]

cv.lasso <- cv.glmnet(as.matrix(trainData[, -ncol(trainData)]),
                      trainData$recovery_time,
                      alpha = 1,
                      lambda = exp(seq(4, -2, length = 100)))
plot(cv.lasso)
```




```r
selected_lambda <- cv.lasso$lambda.min
selected_lambda
```

```
## [1] 0.1353353
```

```r
coefficients_min <- coef(cv.lasso$finalModel, s = selected_lambda)
num_predictors_min <- sum(coefficients_min != 0)
```

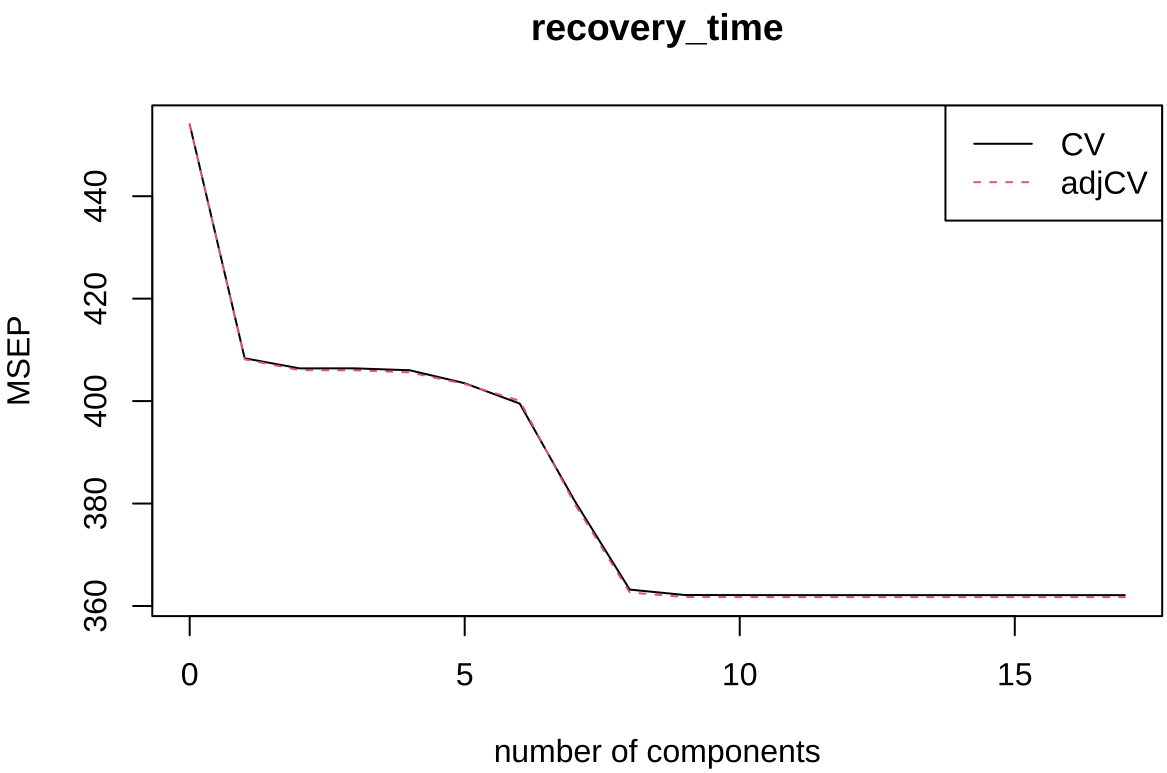
```r
test_predictions <- predict(cv.lasso, newx = as.matrix(testData[, -ncol(testData)]),
                            s = selected_lambda, type = "response")
test_error <- sqrt(mean((test_predictions - testData$recovery_time)^2))
```

PLS Model

```r
set.seed(11)
pls_model <- plsr(recovery_time ~ ., data = trainData,
                  scale = TRUE, validation = "CV")

test_error_pls <- RMSEP(pls_model)
n_comp <- which.min(test_error_pls$val[1,,]) - 1

validationplot(pls_model, val.type = "MSEP", legendpos = "topright")
```




```r
pred_pls_model <- predict(pls_model, newdata = testData, ncomp = n_comp)
test_error <- sqrt(mean((pred_pls_model - testData$recovery_time)^2))
print(test_error)
```

```
## [1] 24.74328
```

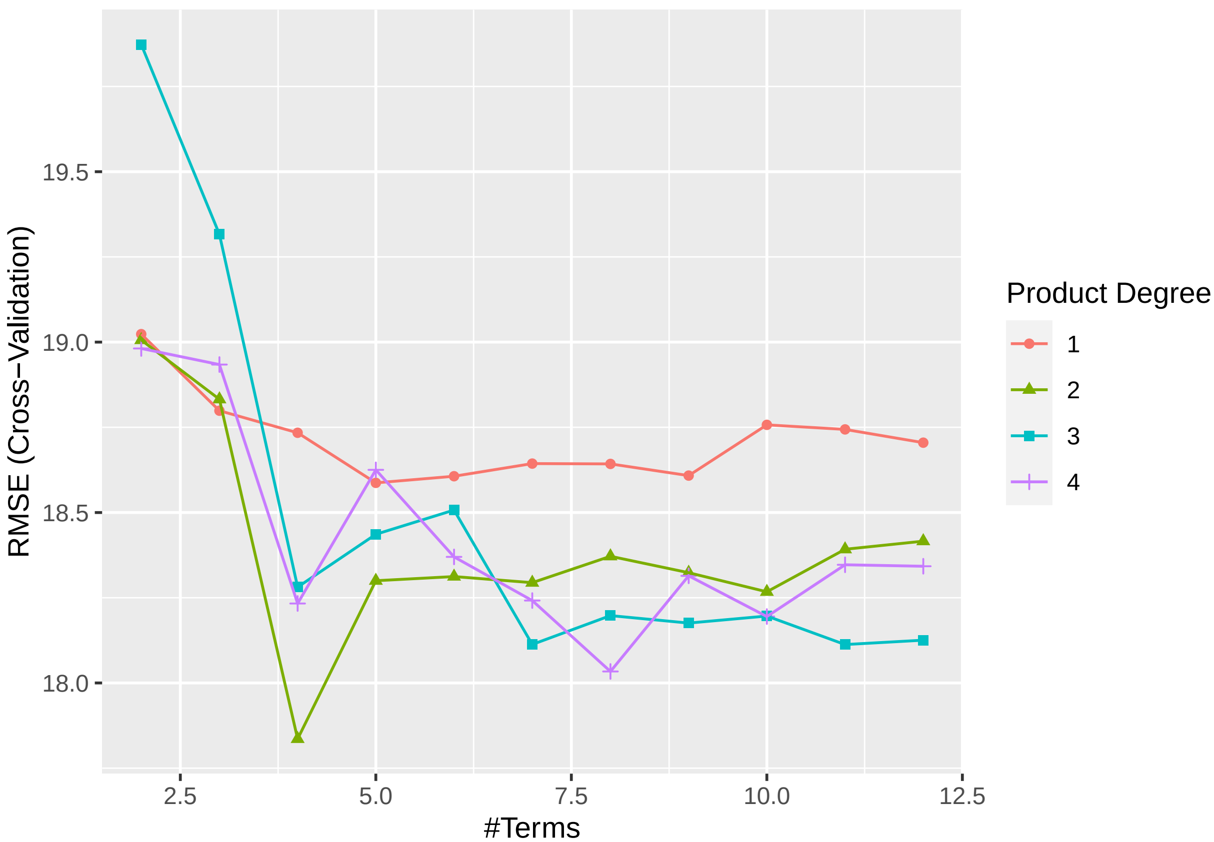
```r
library(mgcv)
library(earth)
```

```
## Warning: package 'earth' was built under R version 4.3.2

## Warning: package 'TeachingDemos' was built under R version 4.3.2
```

MARS

```
ctrl1 <- trainControl(method = "cv", number = 5)
set.seed(11)
mars_grid <- expand.grid(degree = 1:4, nprune = 2:12)
mars.fit <- train(recovery_time ~ .,
                  data = trainData,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)
ggplot(mars.fit)
```



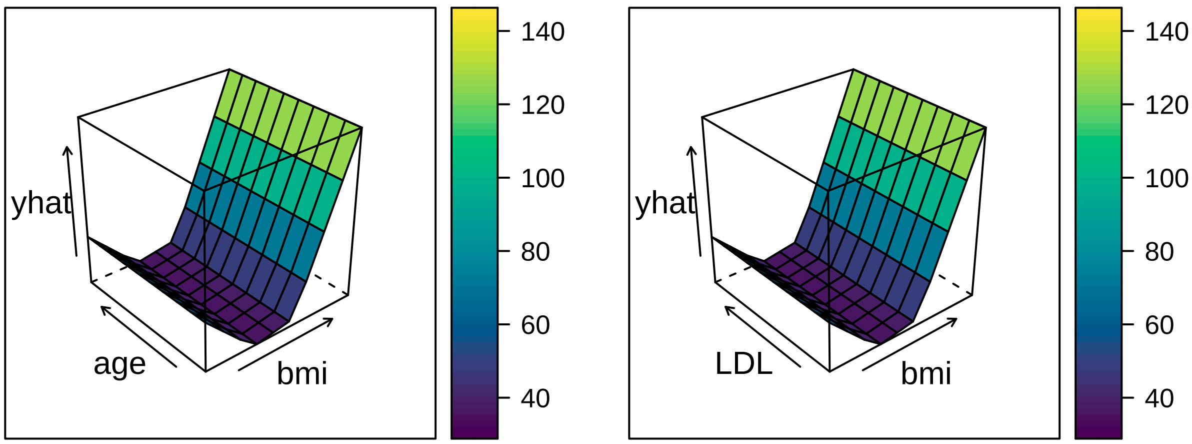

```
mars.fit$bestTune
```

```
##    nprune degree
## 14      4      2
```

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

```
##           (Intercept)          h(30.5-bmi) h(bmi-30.5) * studyB
##              9.413153             4.872689            18.461802
##           h(bmi-25.1)
##              5.678935
```

```r
p1 = pdp::partial(mars.fit, pred.var = c("bmi", "age"), grid.resolution = 10) %>%
  pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE, screen = list(z = 40, x = -60))
p2 = pdp::partial(mars.fit, pred.var = c("bmi", "LDL"), grid.resolution = 10) %>%
  pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE, screen = list(z = 40, x = -60))
gridExtra::grid.arrange(p1, p2, ncol = 2)
```




```r
mars_pred <- predict(mars.fit, newdata = testData)
y_test <- testData$recovery_time
squared_errors <- (mars_pred - y_test)^2
rmse <- sqrt(mean(squared_errors))
print(rmse)
```

```
## [1] 22.97017
```

GAM

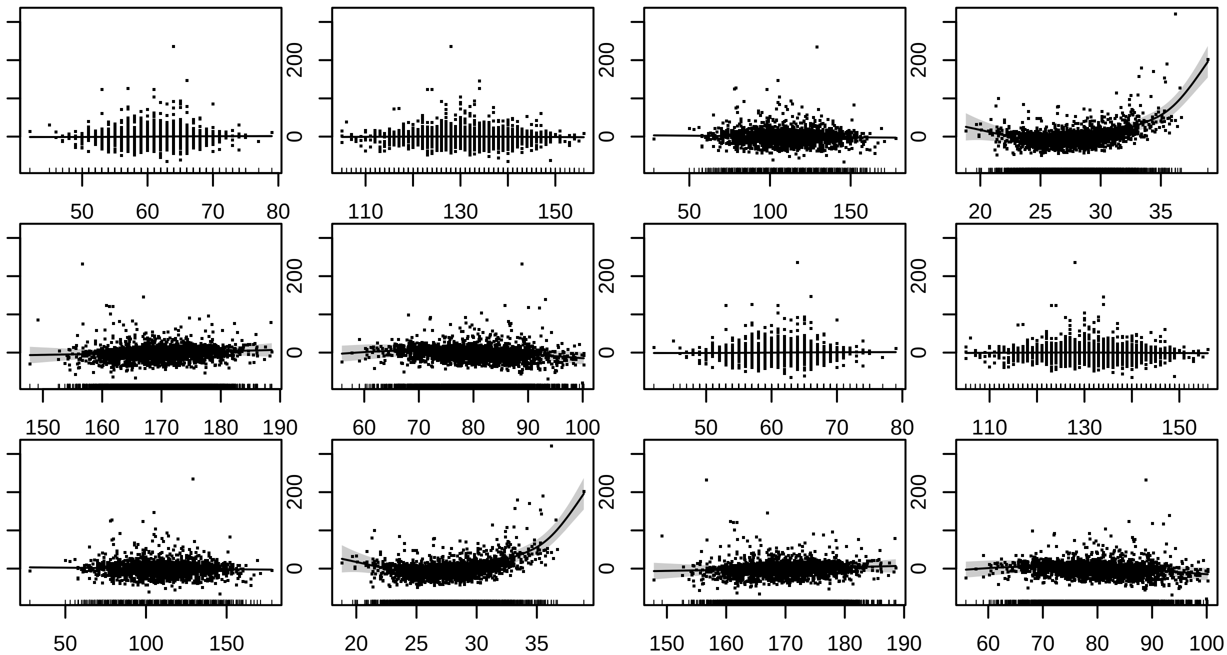
For the variables `height` and `bmi`, the residuals in the plots suggest that there appears to be some curvature or non-linearity in the relationship to `recovery_time`. Therefore, when modeling these variables, it may be necessary to consider more flexible approaches, such as including polynomial terms or using non-linear transformations to better capture the underlying relationship with the outcome variable.

```r
set.seed(11)
gam.fit <- train(recovery_time ~ .,
                 data = trainData,
                 method = "gam",
                 tuneGrid = data.frame(method = "GCV.Cp", select = TRUE),
                 trControl = ctrl1)
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:
## 0.887 0.919 0.864 8.520 1.363 3.570  total = 28.12
##
## GCV score: 326.091
```

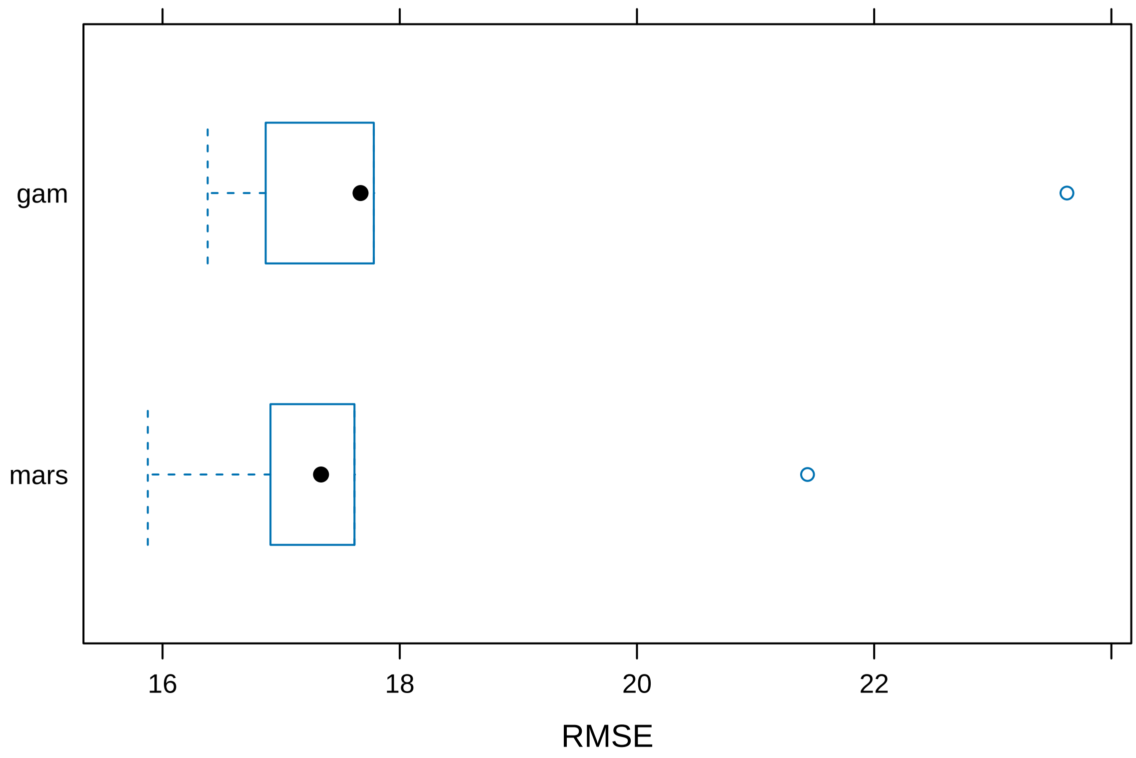
```r
par(mar = c(1, 1, 1, 1), mfrow=c(4,4))
for (i in 1:length(gam.fit$finalModel$term.labels)) {
  plot(gam.fit$finalModel, residuals = TRUE, shade = TRUE,
       xlab = gam.fit$finalModel$term.labels[i], ylab = "Residuals")
  }
```



```r
gam_pred <- predict(gam.fit, newdata = testData)
y_test <- testData$recovery_time
squared_errors <- (gam_pred - y_test)^2
rmse <- sqrt(mean(squared_errors))
print(rmse)
```

```
## [1] 23.76602
```

```r
bwplot(resamples(list(mars = mars.fit, gam = gam.fit)),
       metric = "RMSE")
```



gam
mars
16
18
20
22
RMSE

Results

The RMSE values obtained from Lasso and PLS models were comparable, suggesting that both models performed similarly in predicting the target variable `recovery_time`. This implies that both regularization techniques, despite their differences in approach, yielded comparable predictive performance in this scenario.

The RMSE results indicate that the MARS model achieves a smaller error compared to the GAM model, suggesting superior predictive accuracy. MARS utilizes a piecewise linear approach, allowing for both linear and nonlinear relationships between predictors and the response, while GAM assumes smooth, nonlinear relationships using smoothing functions like splines. Despite MARS potentially offering less interpretability due to its segmented nature, its ability to capture intricate relationships in the data appears to contribute to its better performance in this scenario.

Conclusions