P8106_group2recovery_primaryanalysis

Yimin Chen (yc4195), Yang Yi (yy3307), Qingyue Zhuo (qz2493)

Contents

Import and data manipulation

```r
# Load recovery.RData environment
load("./recovery.Rdata")

dat %>% na.omit()

# dat1 draw a random sample of 2000 participants Uni:3307
set.seed(3307)

dat1 = dat[sample(1:10000, 2000),]

dat1 =
  dat1[, -1] %>%
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    smoking = as.factor(smoking),
    hypertension = as.factor(hypertension),
    diabetes = as.factor(diabetes),
    vaccine = as.factor(vaccine),
    severity = as.factor(severity),
    study = as.factor(
      case_when(study == "A" ~ 1, study == "B" ~ 2, study == "C" ~ 3)
      )
    )

# dat2 draw a random sample of 2000 participants Uni:2493
set.seed(2493)

dat2 = dat[sample(1:10000, 2000),]
```

```
dat2 =
  dat2[, -1] %>%
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    smoking = as.factor(smoking),
    hypertension = as.factor(hypertension),
    diabetes = as.factor(diabetes),
    vaccine = as.factor(vaccine),
    severity = as.factor(severity),
    study = as.factor(
      case_when(study == "A" ~ 1, study == "B" ~ 2, study == "C" ~ 3)
      )
    )

# Merged dataset with unique observation
covid_dat = rbind(dat1, dat2) %>%
  unique()

covid_dat2 = model.matrix(recovery_time ~ ., covid_dat)[, -1]

# Partition dataset into two parts: training data (70%) and test data (30%)
rowTrain = createDataPartition(y = covid_dat$recovery_time, p = 0.7, list = FALSE)

trainData = covid_dat[rowTrain, ]
testData = covid_dat[-rowTrain, ]

ctrl1 = trainControl(method = "repeatedcv", number = 10, repeats = 5)
```
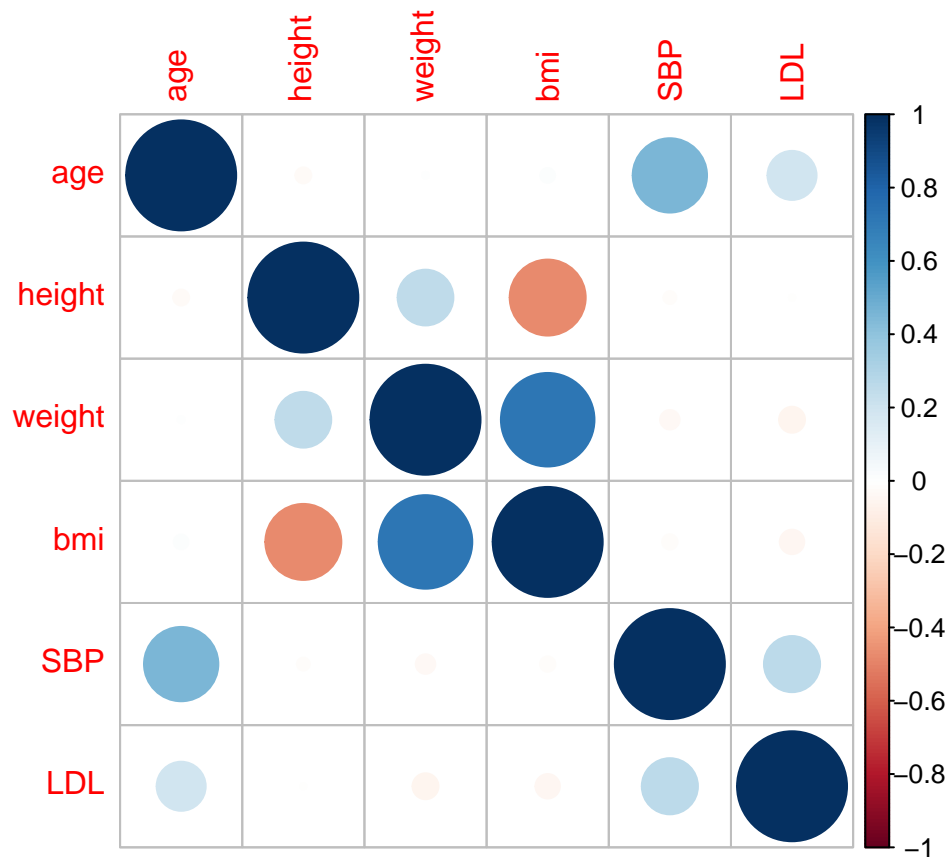
Data visualization

Correlation plot

```
corr_dat = covid_dat[rowTrain,] %>%
  select('age', 'height', 'weight', 'bmi', 'SBP', 'LDL')
corrplot(cor(corr_dat), method = "circle", type = "full")
```

Feature plot

```r
vis_trdat = trainData %>%
  select('age', 'height', 'weight', 'bmi', 'SBP', 'LDL', '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 = vis_trdat[, 1:6],
            y = vis_trdat[, 7],
            plot = "scatter",
            span = 0.5,
            labels = c("Predictors", "Recovery Time"),
            type = c("p", "smooth"))
```
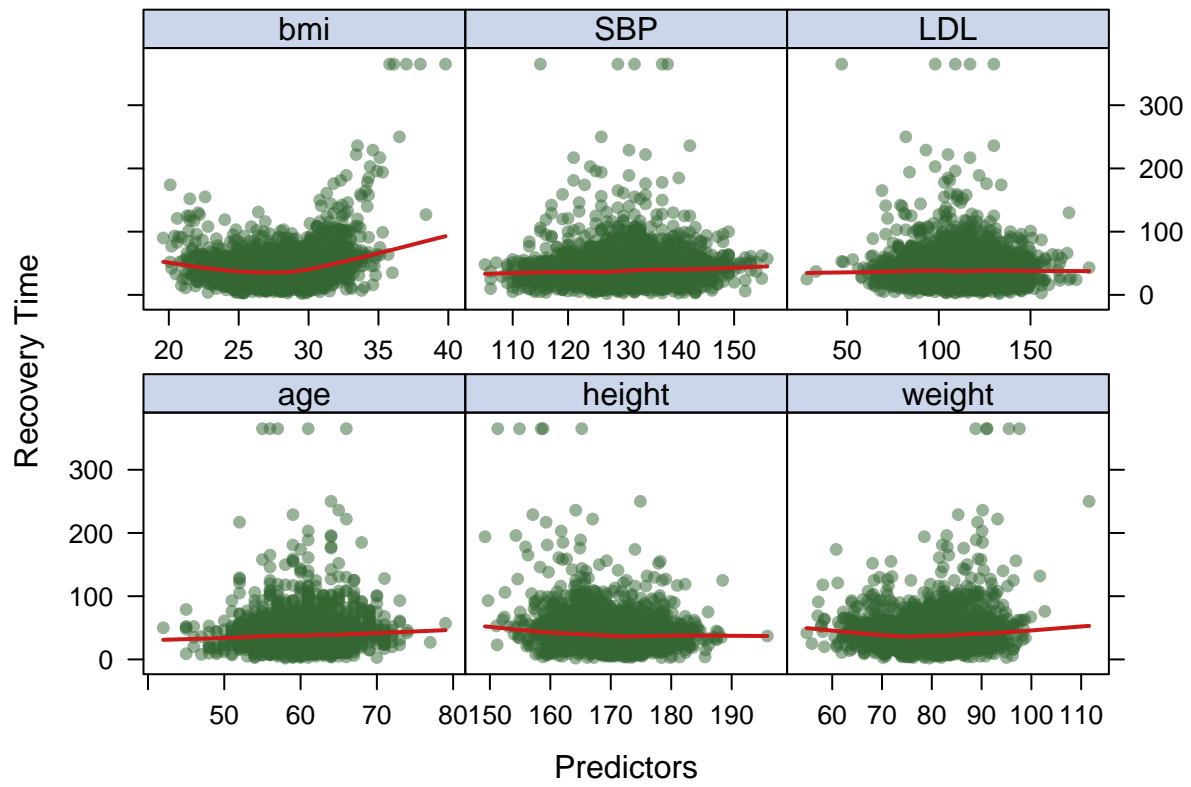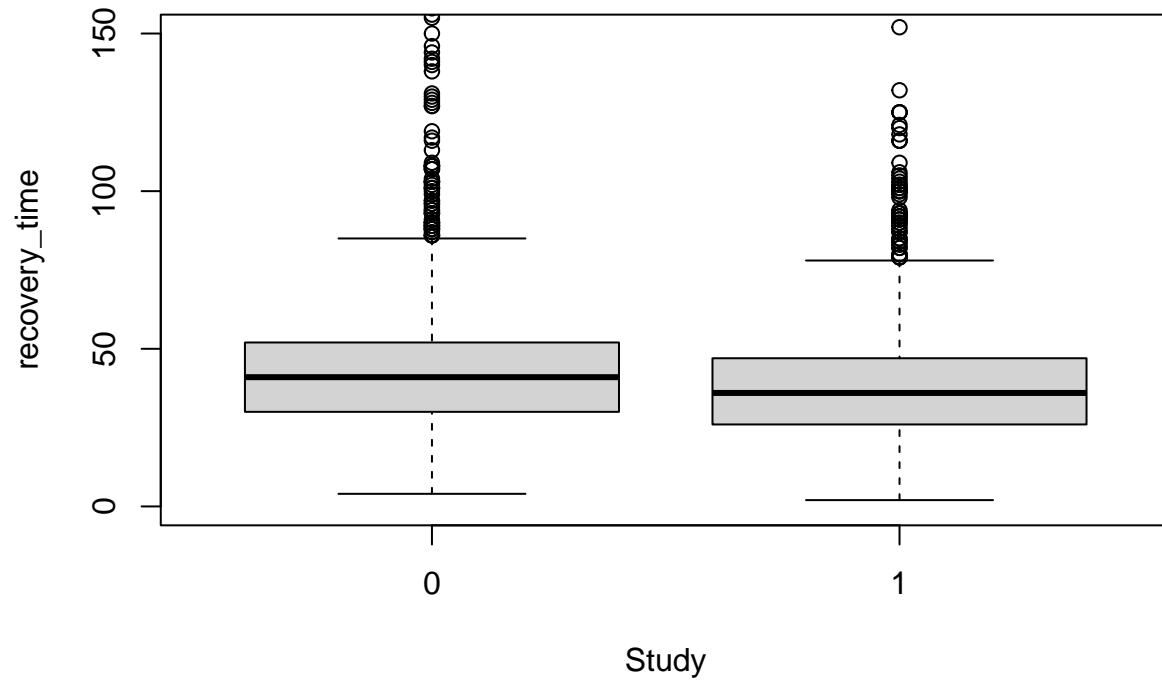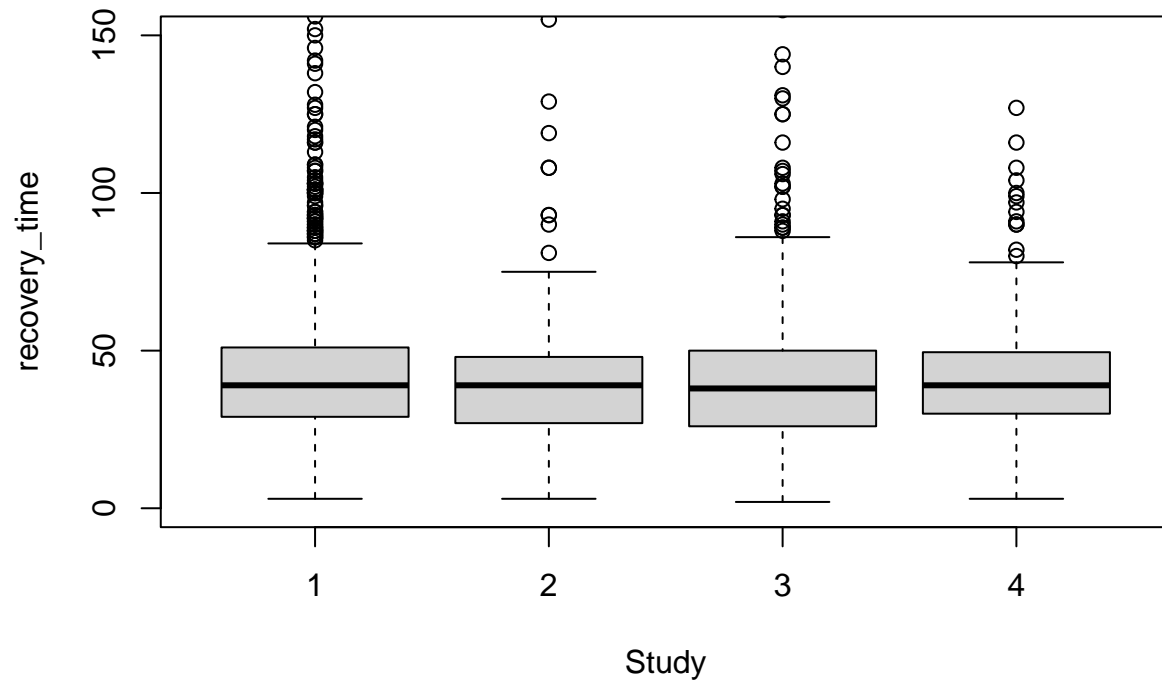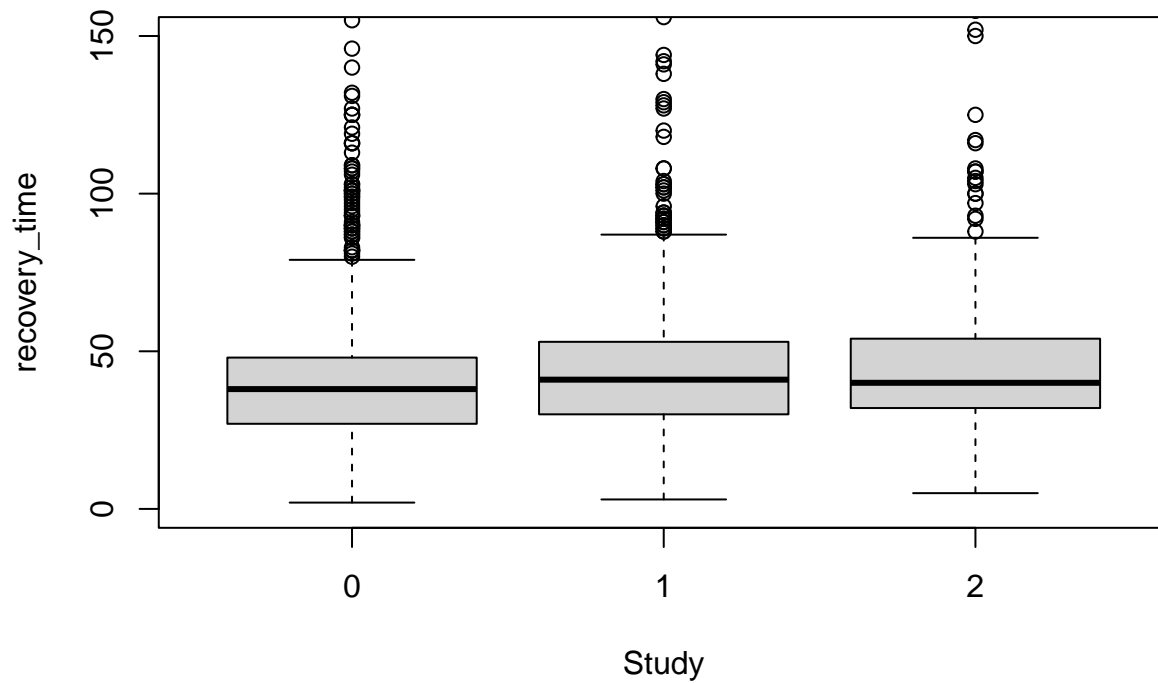
Boxplot
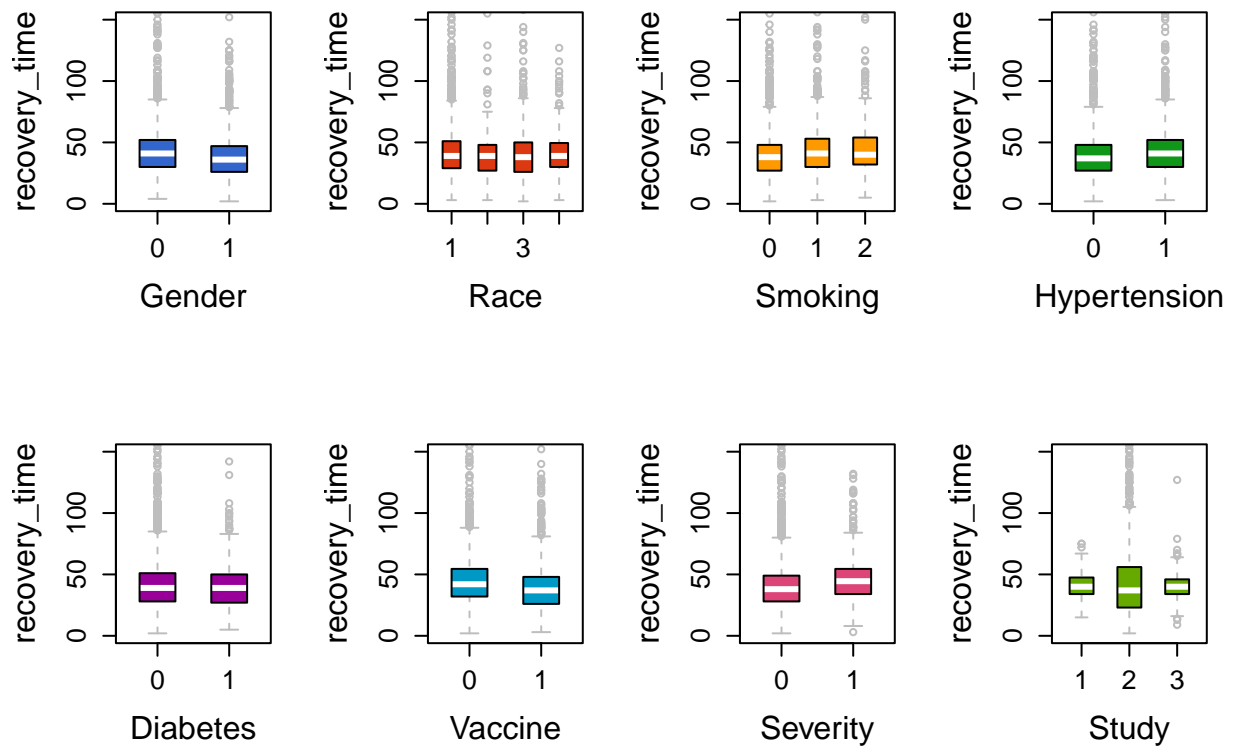
```
bp_gender = boxplot(recovery_time ~ gender, data = trainData, xlab = "Study", ylim = c(0, 150))
```
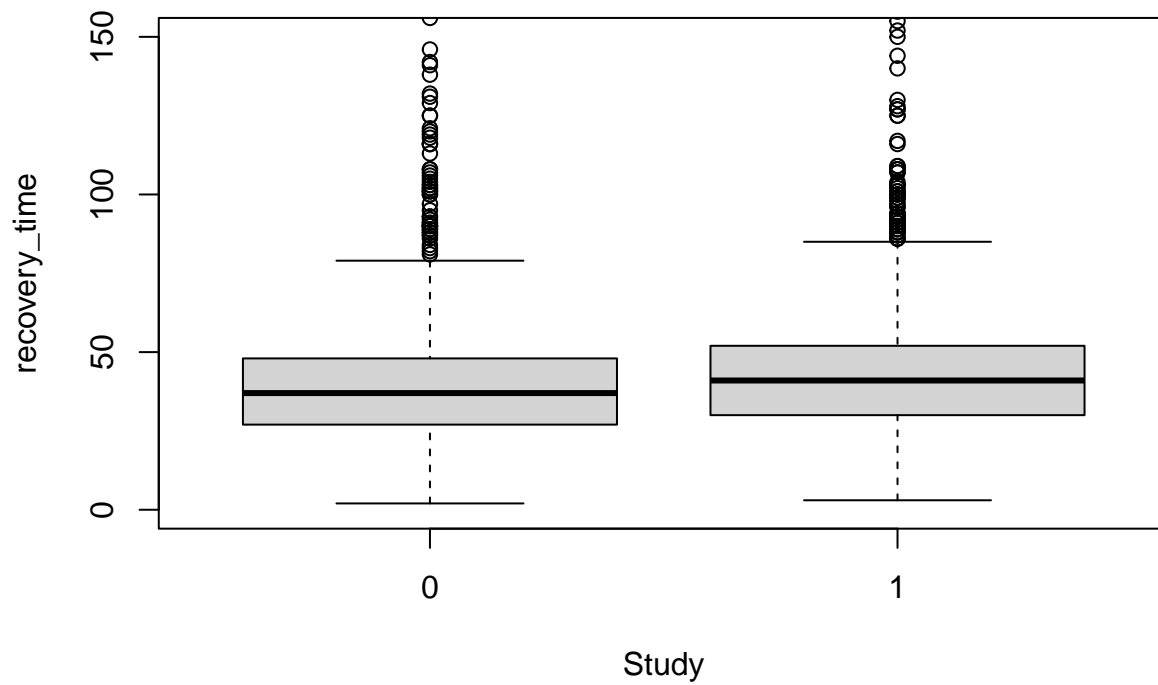


```
bp_race = boxplot(recovery_time ~ race, data = trainData, xlab = "Study", ylim = c(0, 150))
```
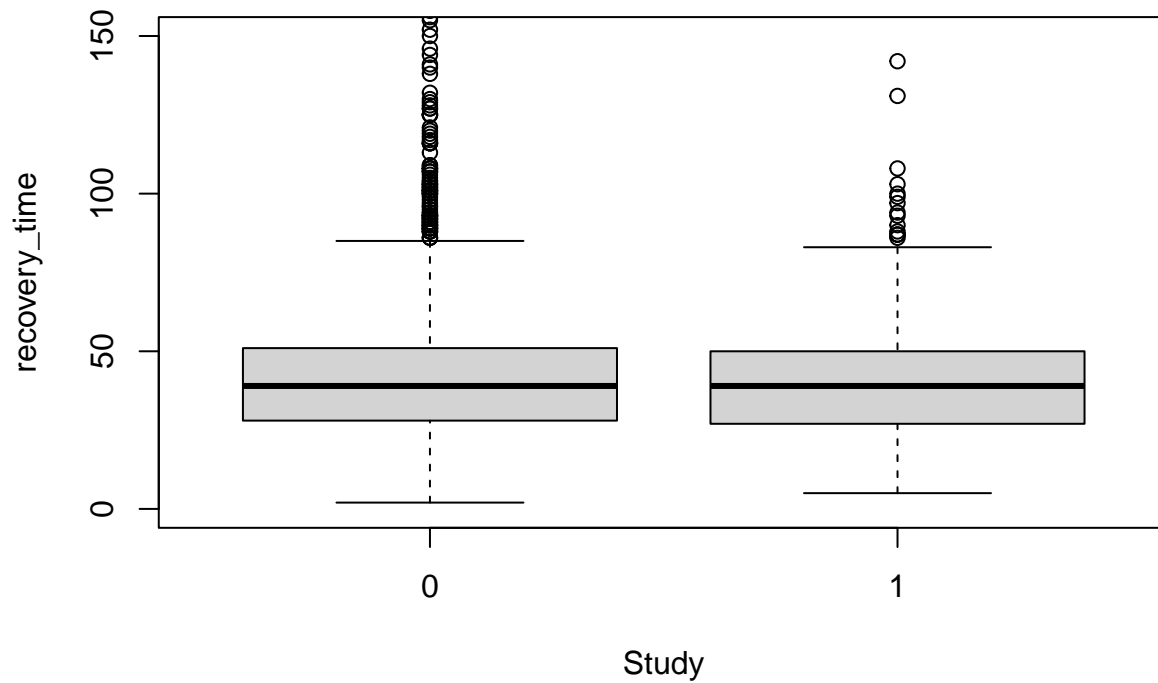


```
bp_smoking = boxplot(recovery_time ~ smoking, data = trainData, xlab = "Study", ylim = c(0, 150))
```
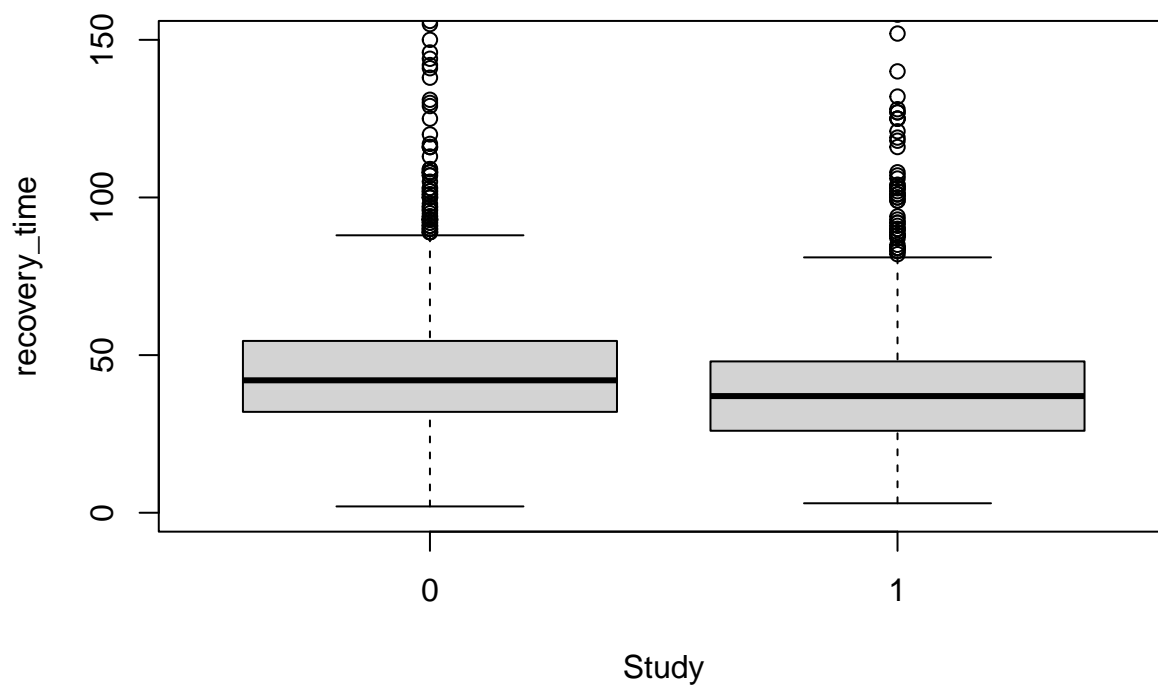
```
bp_hypertension = boxplot(recovery_time ~ hypertension, data = trainData, xlab = "Study", ylim = c(0, 15
```
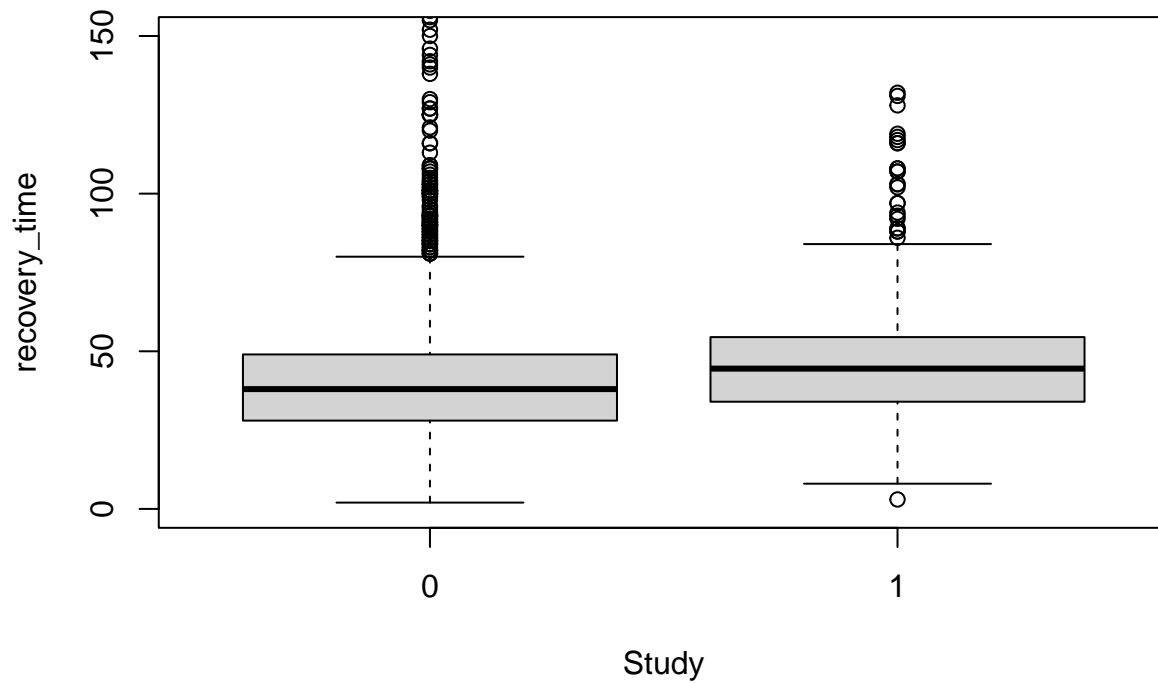


```
bp_diabetes = boxplot(recovery_time ~ diabetes, data = trainData, xlab = "Study", ylim = c(0, 150))
```
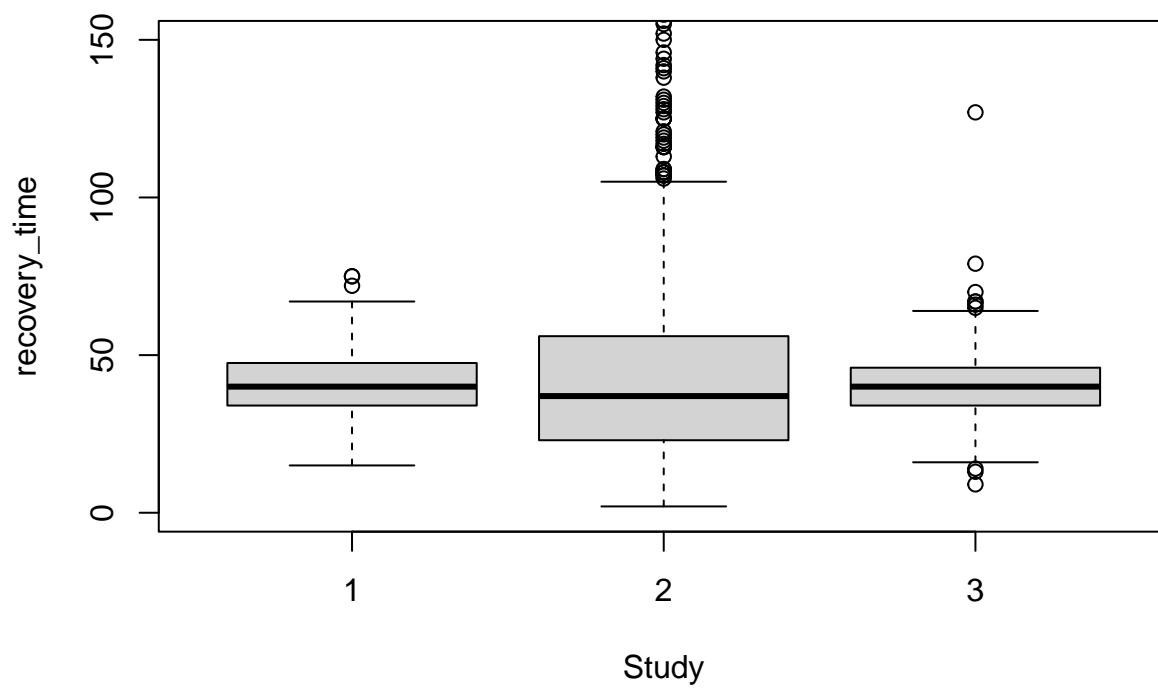
```
bp_vaccine = boxplot(recovery_time ~ vaccine, data = trainData, xlab = "Study", ylim = c(0, 150))
```



```
bp_severity = boxplot(recovery_time ~ severity, data = trainData, xlab = "Study", ylim = c(0, 150))
```

```
bp_study = boxplot(recovery_time ~ study, data = trainData, xlab = "Study", ylim = c(0, 150))
```


Model training