P8106 Final Project: Predicting COVID-19 Recovery Time and Identifying Significant Risk Factors

Runze Cui (rc3521), Yuchen Hua (yh3555), Hongpu Min (hm2946)

2023-05-01

Contents

Background

[Check the report]

Data:

[Description check the report]

```
# For primary analysis:
# Dataset Loading:
load("data/recovery.Rdata")

set.seed(3521) # Runze Cui's uni(2183):
# Create a first random sample of 2000 participants:
dat1 <- dat[sample(1:10000, 2000),]

set.seed(3555) # Yuchen Hua's uni(3555)
# Create a second random sample of 2000 participants:
dat2 <- dat[sample(1:10000, 2000),]

# Merged the two datasets and remove repeated observations:
dat <- unique(rbind(dat1, dat2))

# Get rid of the id variable from the merged dataset and do the data cleaning:
dat <- dat %>%
  select(-id) %>%
  mutate(gender = as.factor(gender)) %>%
  mutate(race = as.factor(race)) %>%
  mutate(smoking = as.factor(smoking)) %>%
  mutate(hypertension = as.factor(hypertension)) %>%
  mutate(diabetes = as.factor(diabetes)) %>%
  mutate(vaccine = as.factor(vaccine)) %>%
  mutate(severity = as.factor(severity)) %>%
  mutate(study = as.factor(study)) %>%
  na.omit() %>%
  relocate(recovery_time)

head(dat)
##      recovery_time age gender race smoking height weight  bmi hypertension
## 8158            52  61      0    1       1  169.9   87.6 30.4            0
## 3387            24  60      1    1       2  173.4   70.6 23.5            0
## 1709            36  60      1    1       1  178.2   79.9 25.1            0
## 4051            23  70      1    4       0  167.4   77.7 27.7            1
## 954             24  63      1    4       0  175.4   88.7 28.8            1
## 531             36  65      0    1       0  160.4   74.4 28.9            1
##      diabetes SBP LDL vaccine severity study
## 8158        0 118 103       0        0     C
## 3387        0 129 101       1        0     B
## 1709        0 130 107       1        0     A
## 4051        0 145 128       1        0     B
## 954         0 131 100       0        0     A
## 531         0 137 153       1        0     A
```

```r
# Separate the data as training and test data:
set.seed(3521)
# Specify rows of training data:
trRows <- createDataPartition(dat$recovery_time, p = 0.7, list = FALSE)


# Training data:
training <- dat[trRows, ]
## Covariates' matrix:
x <- model.matrix(recovery_time ~ ., dat)[trRows, -1]
## Response's vector:
y <- dat$recovery_time[trRows]

# Test data:
test <- dat[-trRows, ]
## Covariates' matrix:
x2 <- model.matrix(recovery_time ~ ., dat)[-trRows, -1]
## Response's vector:
y2 <- dat$recovery_time[-trRows]




# For secondary analysis:
dat_2 <- dat %>%
  mutate(recovery_time = ifelse(recovery_time > 30, "great", "less")) %>%
  mutate(recovery_time = as.factor(recovery_time))

# Training data:
training_sec <- dat_2[trRows, ]
## Covariates' matrix:
x_sec <- model.matrix(recovery_time ~ ., dat_2)[trRows, -1]
## Response's vector:
y_sec <- dat_2$recovery_time[trRows]

# Test data:
test_sec <- dat_2[-trRows, ]
## Covariates' matrix:
x2_sec <- model.matrix(recovery_time ~ ., dat_2)[-trRows, -1]
## Response's vector:
y2_sec <- dat_2$recovery_time[-trRows]
```

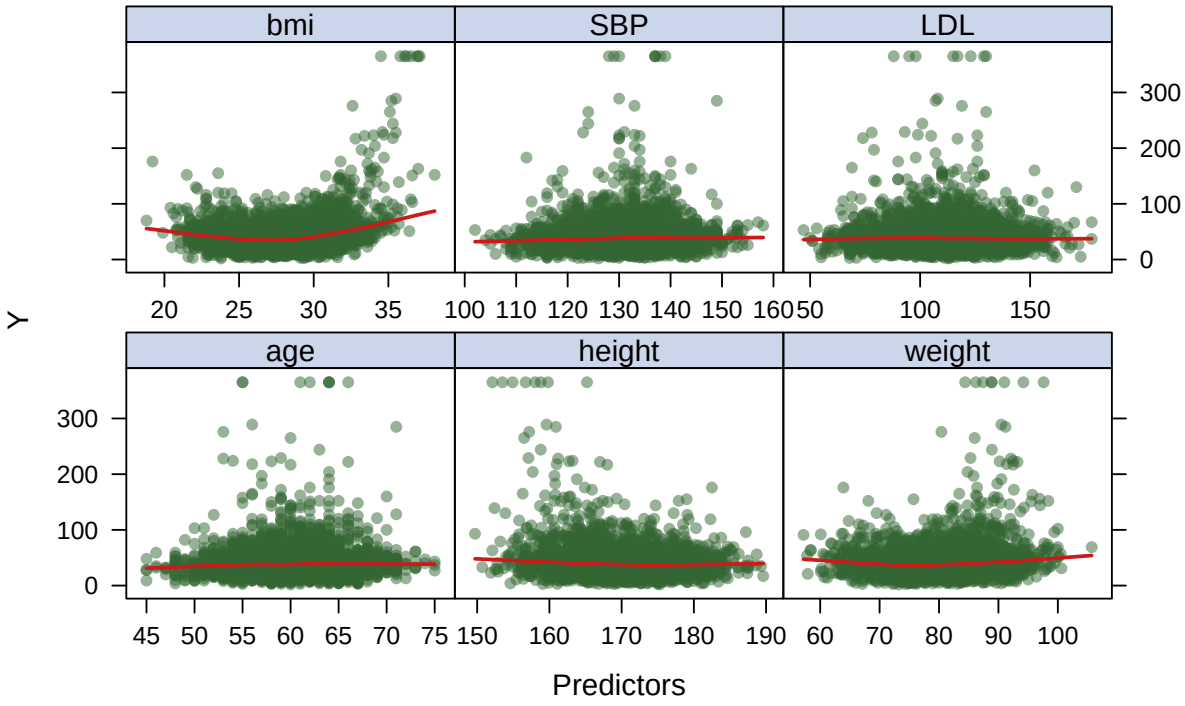
Exploratory Analysis and Data Visualization

[Description check the report]

```r
# For primary analysis:
## For continuous variables:
theme = trellis.par.get()
theme$plot.symbol$col = rgb(.2, .4, .2, .5)
theme$plot.symbol$pch = 16
```

```
theme$plot.line$col = rgb(.8, .1, .1, 1)
theme$plot.line$lwd = 2
theme$strip.background$col = rgb(.0, .2, .6, .2)
trellis.par.set(theme)

featurePlot(x = dat %>% dplyr::select(age, height, weight, bmi, SBP, LDL),
            y = dat$recovery_time,
            plot = "scatter",
            span = .5,
            labels = c("Predictors","Y"),
            main = "Figure 1.1. Lattice Plots for Continuous Variables in Primary Analysis",
            type = c("p", "smooth"))
```

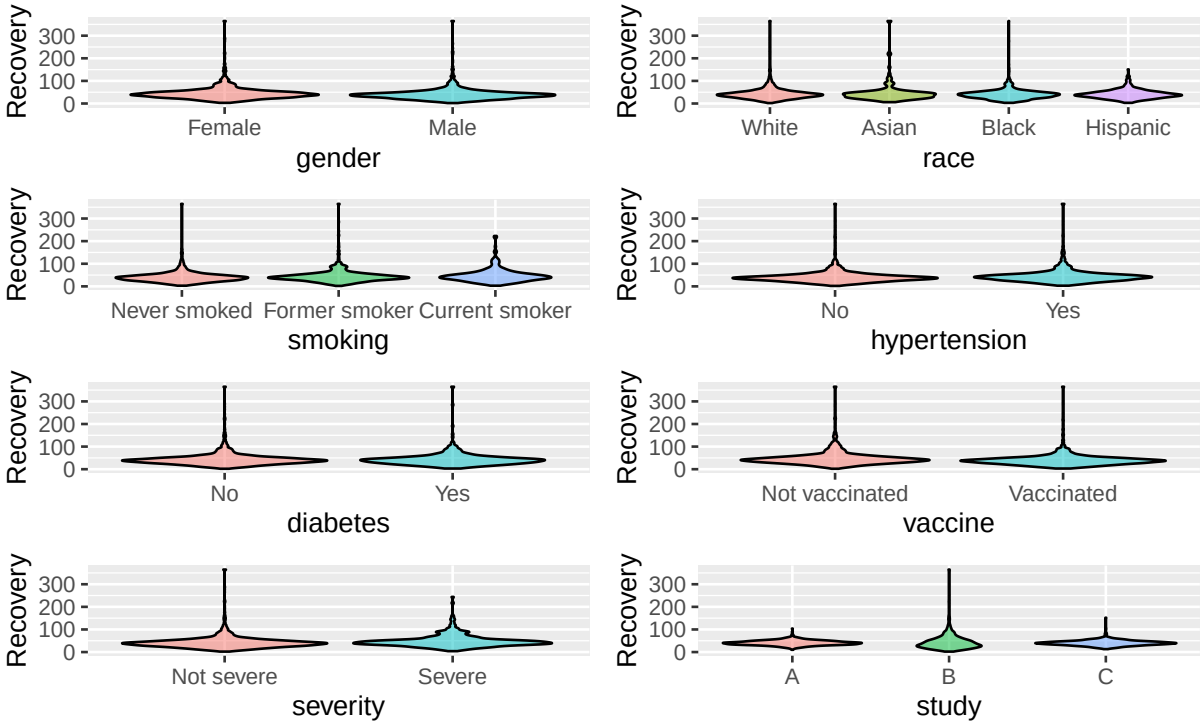
Figure 1.1. Lattice Plots for Continuous Variables in Primary Analysis




```
## For categorical variables:
gender_plot = dat %>%
  ggplot(aes(x = gender, y = recovery_time, fill = gender)) +
  geom_violin(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Female','Male')) +
  ylab("Recovery") +
  theme(legend.position = "none")

race_plot = dat %>%
  ggplot(aes(x = race, y = recovery_time, fill = race)) +
  geom_violin(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('White','Asian','Black', 'Hispanic')) +
  ylab("Recovery") +
  theme(legend.position = "none")
```

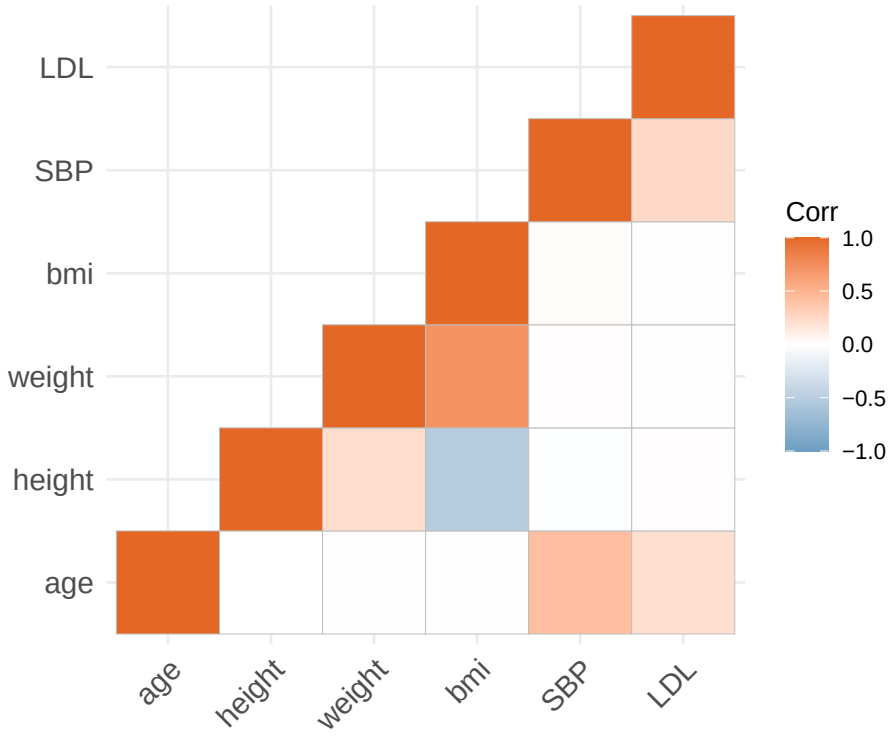
```r
smoking_plot = dat %>%
  ggplot(aes(x = smoking, y = recovery_time, fill = smoking)) +
  geom_violin(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Never smoked','Former smoker','Current smoker')) +
  ylab("Recovery") +
  theme(legend.position = "none")

hyper_plot = dat %>%
  ggplot(aes(x = hypertension, y = recovery_time, fill = hypertension)) +
  geom_violin(color = "black", alpha = .5) +
  ylab("Recovery") +
  scale_x_discrete(labels = c('No','Yes')) +
  theme(legend.position = "none")

diabetes_plot = dat %>%
  ggplot(aes(x = diabetes, y = recovery_time, fill = diabetes)) +
  geom_violin(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('No','Yes')) +
  ylab("Recovery") +
  theme(legend.position = "none")

vac_plot = dat %>%
  ggplot(aes(x = vaccine, y = recovery_time, fill = vaccine)) +
  geom_violin(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Not vaccinated','Vaccinated')) +
  ylab("Recovery") +
  theme(legend.position = "none")

severity_plot = dat %>%
  ggplot(aes(x = severity, y = recovery_time, fill = severity)) +
  geom_violin(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Not severe','Severe')) +
  ylab("Recovery") +
  theme(legend.position = "none")

study_plot = dat %>%
  ggplot(aes(x = study, y = recovery_time, fill = study)) +
  geom_violin(color = "black", alpha = .5) +
  ylab("Recovery") +
  theme(legend.position = "none")

(gender_plot + race_plot + smoking_plot + hyper_plot) / (diabetes_plot + vac_plot + severity_plot + stu
  plot_layout(guides = "collect") +
  plot_annotation(title = "Figure 1.2. Violin Plots for Categorical Variables in Primary Analysis")
```

Figure 1.2. Violin Plots for Categorical Variables in Primary Analysis



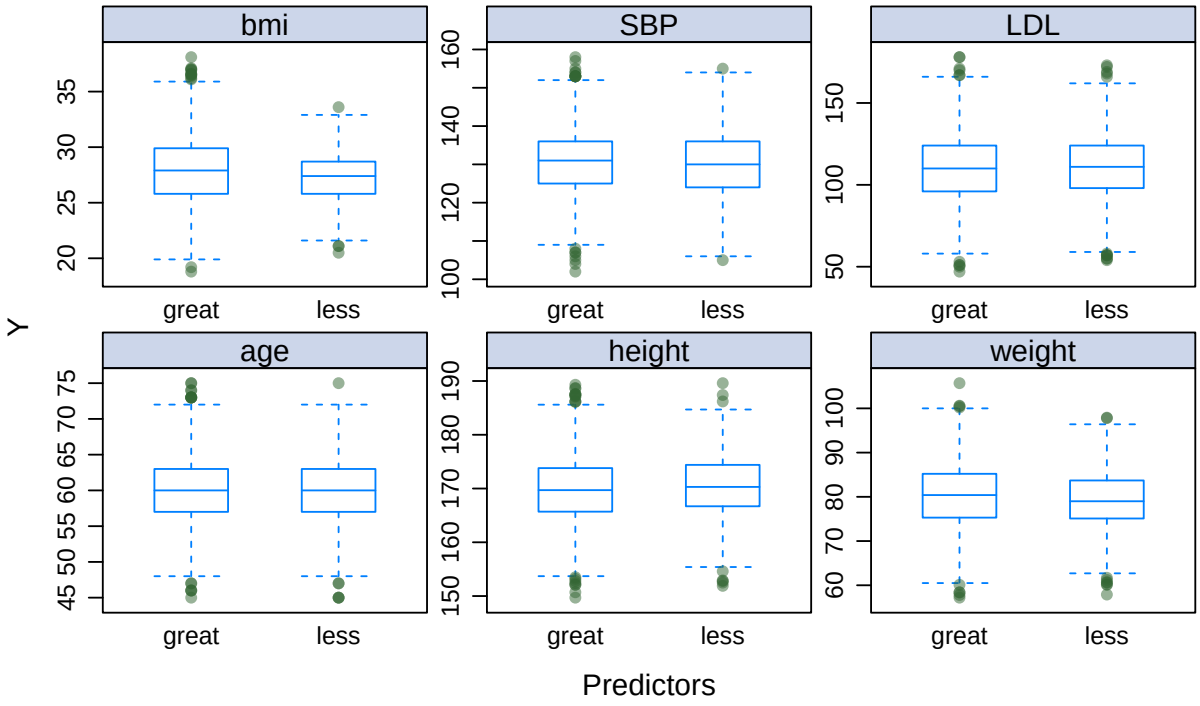

```
## Correlation matrix for continuous variables ONLY:
model.matrix( ~ 0+., data = dat %>% dplyr::select(age, height, weight, bmi, SBP, LDL)) %>%
  cor(use = "pairwise.complete.obs") %>%
  ggcorrplot::ggcorrplot(show.diag = T,
                         type = "lower",
                         lab = F,
                         colors = c("#6D9EC1", "white", "#E46726")) +
  ggtitle("Figure 1.3. Correlation matrix for continuous datasat")
```

Figure 1.3. Correlation matrix for continuous datasat




```
# For secondary analysis:
## For continuous variables:
theme = trellis.par.get()
theme$plot.symbol$col = rgb(.2, .4, .2, .5)
theme$plot.symbol$pch = 16
theme$plot.line$col = rgb(.8, .1, .1, 1)
theme$plot.line$lwd = 2
theme$strip.background$col = rgb(.0, .2, .6, .2)
trellis.par.set(theme)

featurePlot(x = dat_2 %>% dplyr::select(age, height, weight, bmi, SBP, LDL),
            y = dat_2$recovery_time,
            plot = "box", pch = "|",
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            labels = c("Predictors","Y"),
            main = "Figure 1.4. Lattice Plots for Continuous Variables in Secondary Analysis",
            auto.key = list(columns = 2))
```

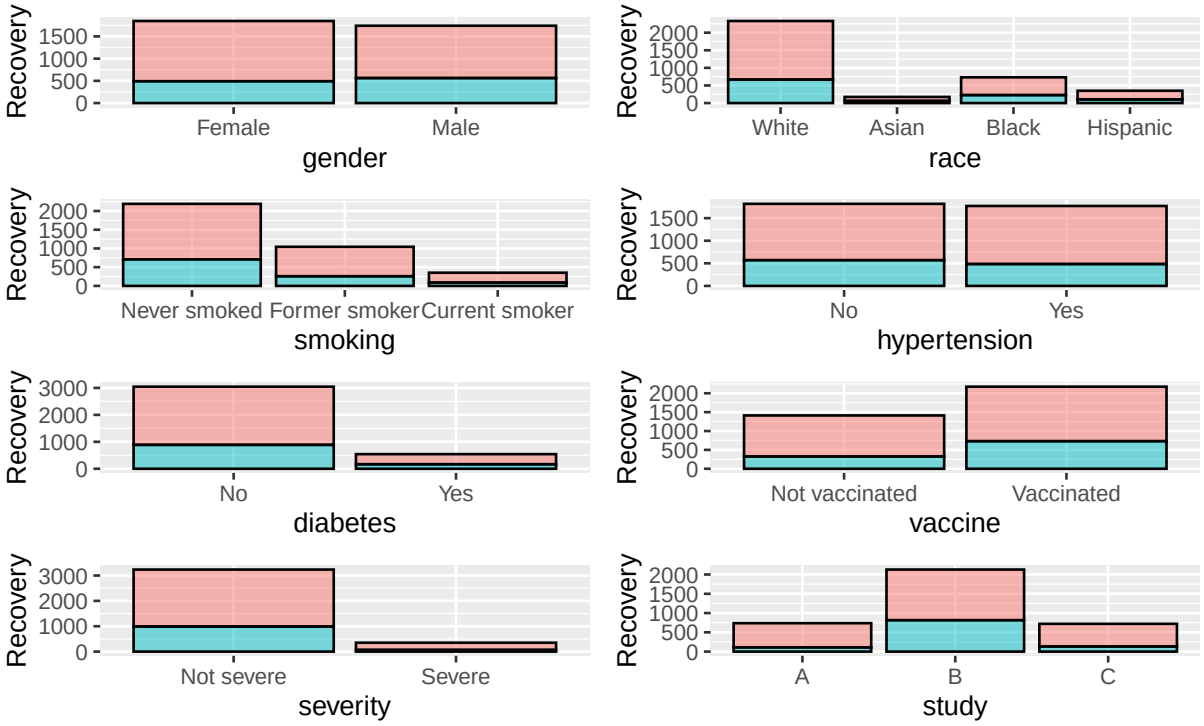
Figure 1.4. Lattice Plots for Continuous Variables in Secondary Analysis




```
## For categorical variables:
gender_plot_sec = dat_2 %>%
  ggplot(aes(x = gender, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Female','Male')) +
  ylab("Recovery") +
  theme(legend.position = "none")

race_plot_sec = dat_2 %>%
  ggplot(aes(x = race, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('White','Asian','Black', 'Hispanic')) +
  ylab("Recovery") +
  theme(legend.position = "none")

smoking_plot_sec = dat_2 %>%
  ggplot(aes(x = smoking, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Never smoked','Former smoker','Current smoker')) +
  ylab("Recovery") +
  theme(legend.position = "none")

hyper_plot_sec = dat_2 %>%
  ggplot(aes(x = hypertension, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  ylab("Recovery") +
  scale_x_discrete(labels = c('No','Yes')) +
```

```r
  theme(legend.position = "none")

diabetes_plot_sec = dat_2 %>%
  ggplot(aes(x = diabetes, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('No','Yes')) +
  ylab("Recovery") +
  theme(legend.position = "none")

vac_plot_sec = dat_2 %>%
  ggplot(aes(x = vaccine, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Not vaccinated','Vaccinated')) +
  ylab("Recovery") +
  theme(legend.position = "none")

severity_plot_sec = dat_2 %>%
  ggplot(aes(x = severity, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  scale_x_discrete(labels = c('Not severe','Severe')) +
  ylab("Recovery") +
  theme(legend.position = "none")

study_plot_sec = dat_2 %>%
  ggplot(aes(x = study, fill = recovery_time)) +
  geom_bar(color = "black", alpha = .5) +
  ylab("Recovery") +
  theme(legend.position = "none")

(gender_plot_sec + race_plot_sec + smoking_plot_sec + hyper_plot_sec) / (diabetes_plot_sec + vac_plot_s
  plot_layout(guides = "collect") +
  plot_annotation(title = "Figure 1.5. Bar Plots for Categorical Variables in Secondary Analysis")
```

Figure 1.5. Bar Plots for Categorical Variables in Secondary Analysis



Note: Red is recovery time less than and equal to 30. Blue is greater than 30.

Primary Analysis

Recovery time as continuous variable.

Secondary Analysis

Create a new dataset with unify variables' name:

```
ctrl1 = trainControl(method = "repeatedcv",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

ctrl = trainControl(method = "repeatedcv", number = 10, repeats = 5)

set.seed(3521)
# GLM fit
glm = train(x = x_sec,
            y = y_sec,
            method = "glm", metric = "ROC", trControl = ctrl1)
```