EDA_data_process_Candice

Candice Yu

2024-03-23

Contents

```r
library(caret)
library(tidyverse)
library(gridExtra)
```

Load Data

```r
load("./Data/recovery.RData")
dat <- dat %>%
  mutate(gender = as_factor(gender),
         diabetes = as_factor(diabetes),
         hypertension = as_factor(hypertension),
         vaccine = as_factor(vaccine),
         severity = as_factor(severity)) %>%
  select(-id)
```

EDA

Overview of the Data

```r
# brief summary of the data
skimr::skim(dat)
```

Table 1: Data summary

Name	dat
Number of rows	3000
Number of columns	15
Column type frequency:	
character	1
factor	7
numeric	7
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
study	0	1	1	1	0	2	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	0: 1544, 1: 1456
race	0	1	FALSE	4	1: 1967, 3: 604, 4: 271, 2: 158
smoking	0	1	FALSE	3	0: 1822, 1: 859, 2: 319
hypertension	0	1	FALSE	2	0: 1508, 1: 1492
diabetes	0	1	FALSE	2	0: 2537, 1: 463

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
vaccine	0	1	FALSE	2	1: 1788, 0: 1212
severity	0	1	FALSE	2	0: 2679, 1: 321

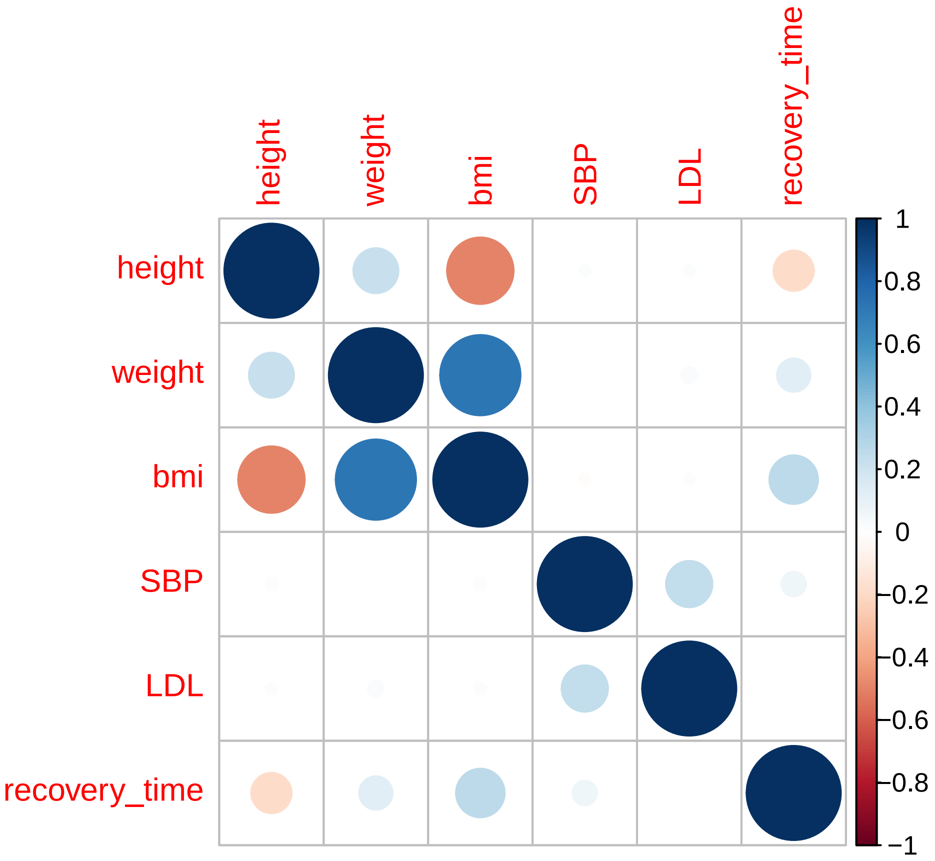
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	0	1	60.20	4.48	42.0	57.0	60.00	63.0	79.0	
height	0	1	169.90	5.97	147.8	166.0	169.90	173.9	188.6	
weight	0	1	79.96	7.14	55.9	75.2	79.80	84.8	103.7	
bmi	0	1	27.76	2.79	18.8	25.8	27.65	29.5	38.9	
SBP	0	1	130.47	7.97	105.0	125.0	130.00	136.0	156.0	
LDL	0	1	110.45	19.76	28.0	97.0	110.00	124.0	178.0	
recovery_time	0	1	42.17	23.15	2.0	31.0	39.00	49.0	365.0	

EDA for Continuous Variables

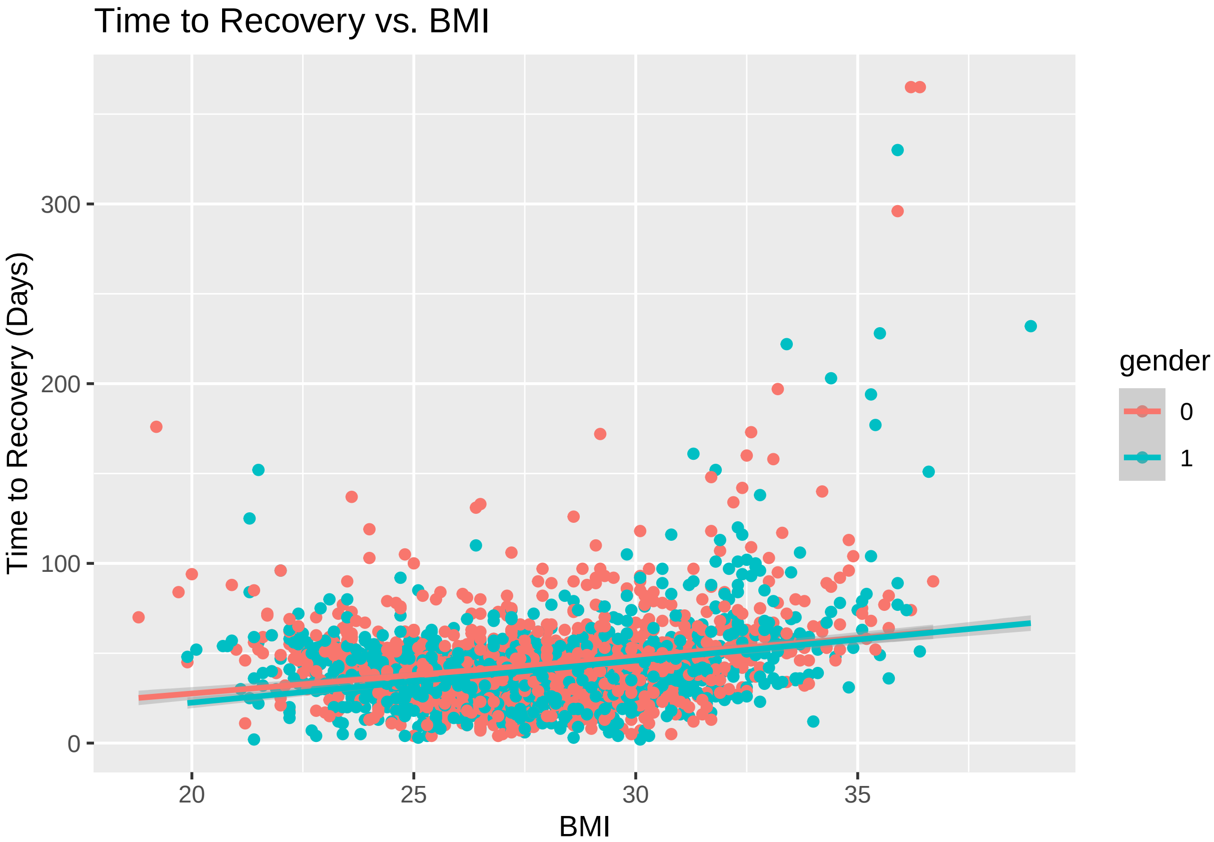
Correlation plot for continuous variables

```r
# correlation plot for continuous variables
continuous_vars <- dat %>%
  select(height, weight, bmi, SBP, LDL, recovery_time)
correlations <- cor(continuous_vars)
corrplot::corrplot(correlations, method = "circle")
```

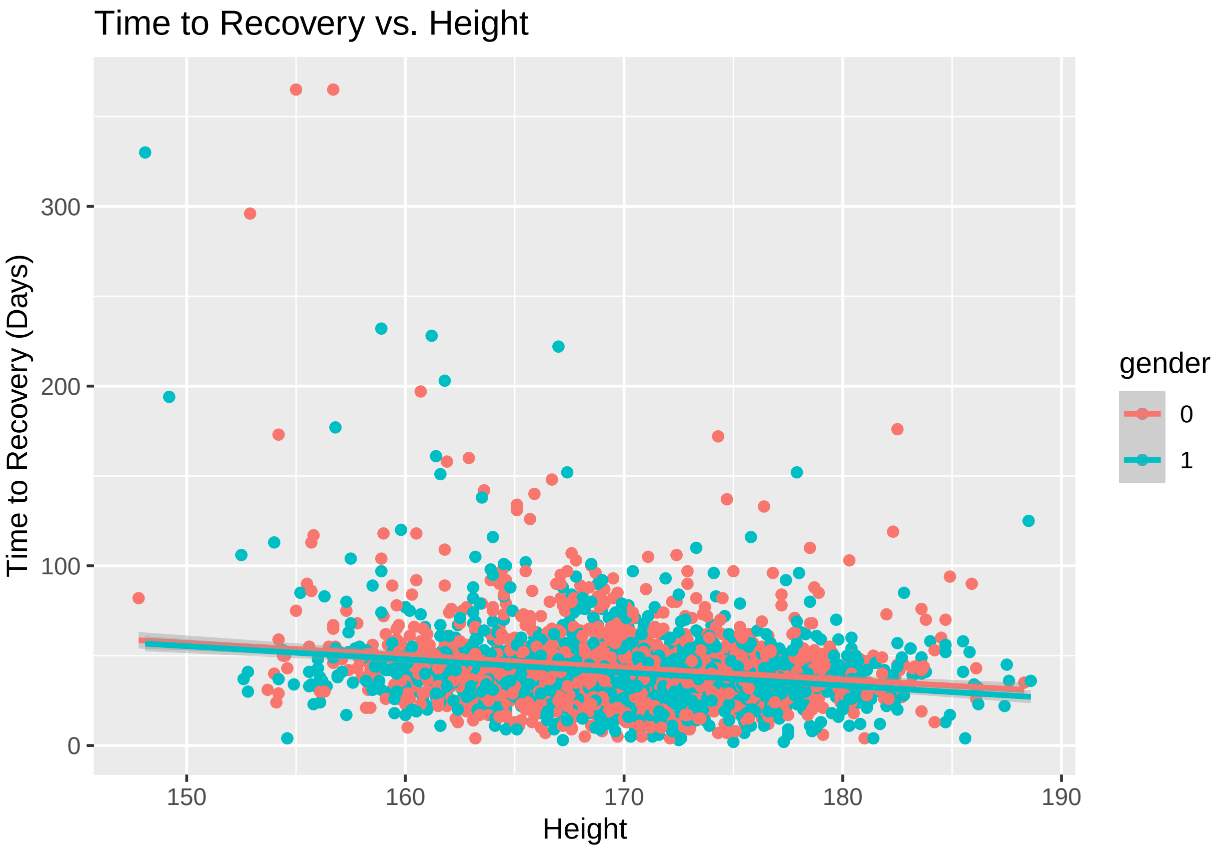


Scatter plots to explore potential relationships

```r
# between time to recovery and bmi
ggplot(dat, aes(x = bmi, y = recovery_time, color = gender)) +
  geom_point() + geom_smooth(method = "lm") +
  labs(title = "Time to Recovery vs. BMI",
       x = "BMI",
       y = "Time to Recovery (Days)")
```



```r
# between time to recovery and height
ggplot(dat, aes(x = height, y = recovery_time, color = gender)) +
  geom_point() + geom_smooth(method = "lm") +
  labs(title = "Time to Recovery vs. Height",
       x = "Height",
       y = "Time to Recovery (Days)")
```

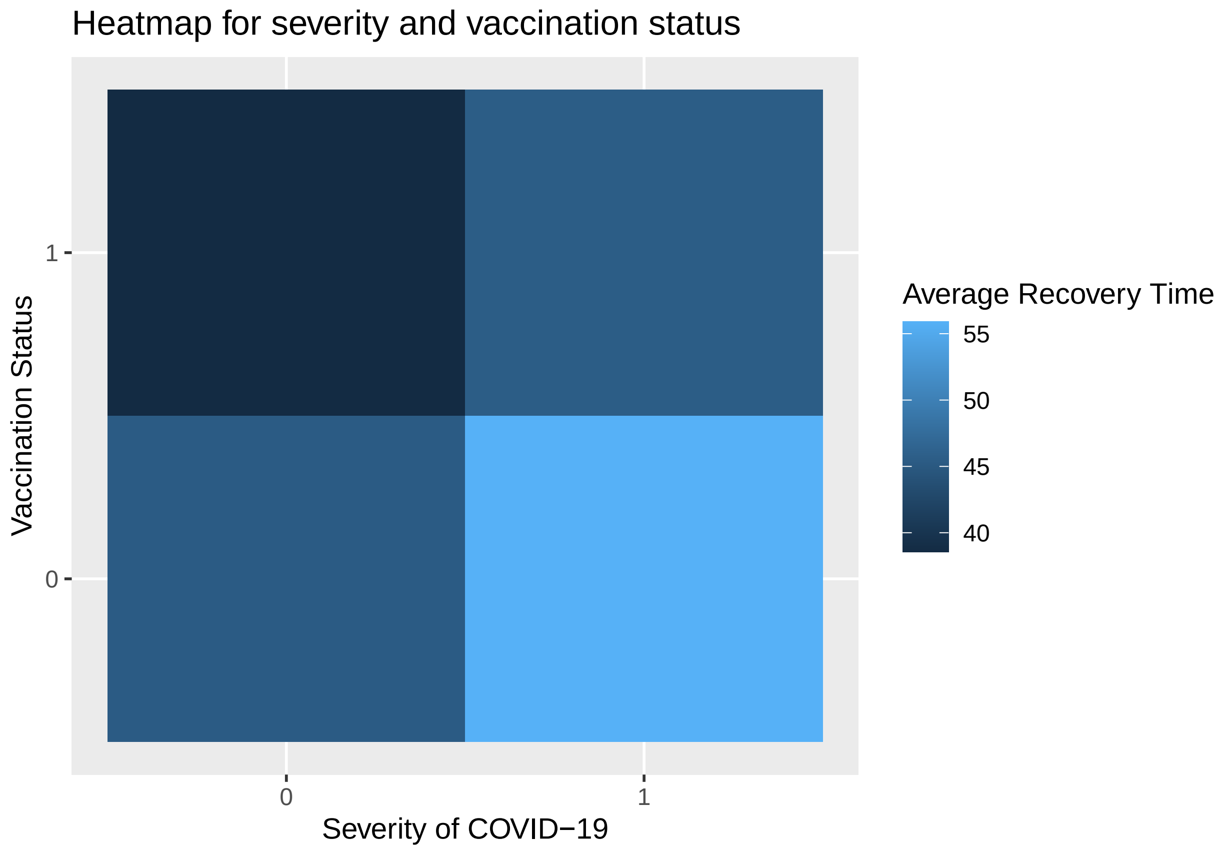


The correlation plot and scatter plots show some relationships between continuous variables, but none of them appear to be strongly correlated with `recovery_time`. This may suggest that linear relationships are not strong, and hence a non-linear model could be more appropriate.

EDA for Discrete Variables

Heatmap for severity and vaccination status

```
# Heatmap for systolic blood pressure across severity and vaccination status
dat %>%
  group_by(severity, vaccine) %>%
  summarise(avg_recovery_time = mean(recovery_time)) %>%
  ggplot(aes(x = factor(severity), y = factor(vaccine), fill = avg_recovery_time)) +
  geom_tile() +
  labs(title = "Heatmap for severity and vaccination status",
       x = "Severity of COVID-19",
       y = "Vaccination Status",
       fill = "Average Recovery Time")
```



The heatmap helps in understanding the bivariate relationship between severity, vaccination status, and recovery time.

Observations from the Heatmap:

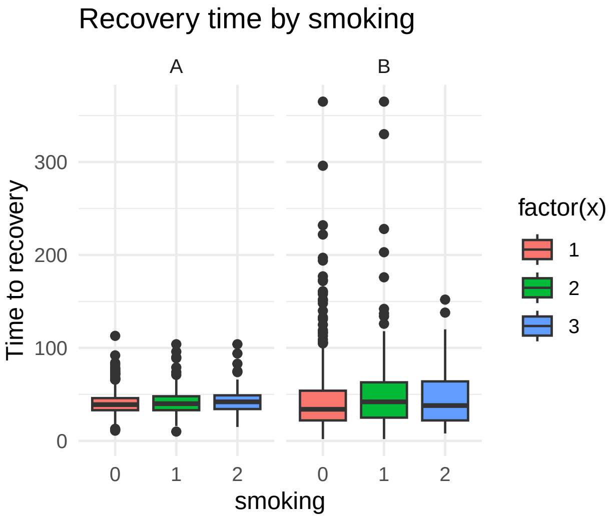
- Individuals with severe COVID-19 infection (1 on the x-axis) have longer average recovery times than those with non-severe infections, regardless of vaccination status.
- Vaccination status seems to have an influence on the recovery time. Those who are vaccinated (1 on the y-axis) tend to have shorter recovery times even when the infection is severe.

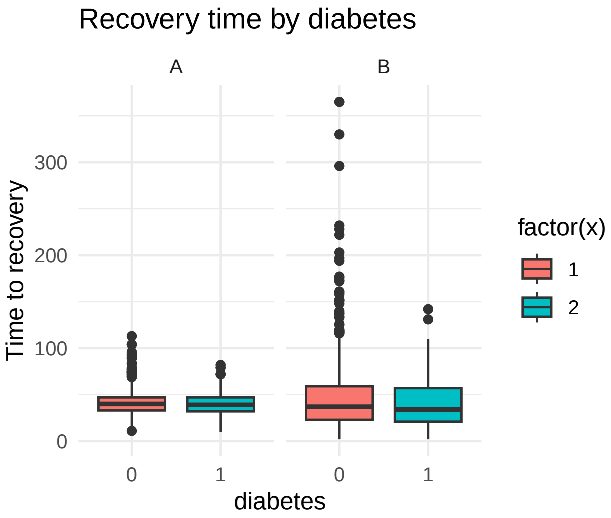
Implications for Modeling:

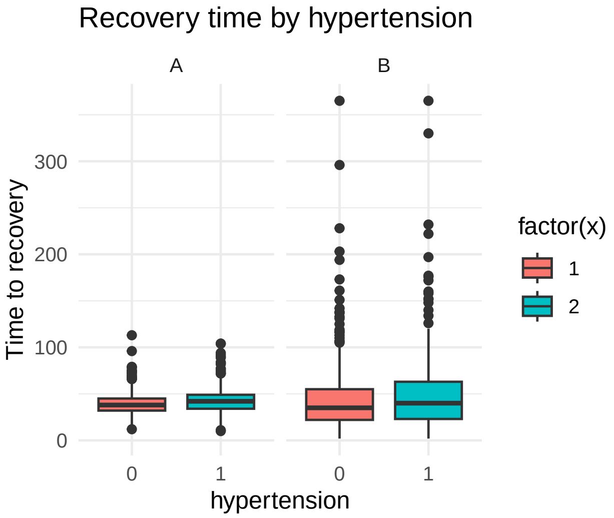
- The heatmap suggests there might be an interaction effect between severity and vaccination status on the recovery time. Therefore, when modeling, consider including an interaction term between these two variables.
- Given the apparent differences in recovery time across the groups, both severity and vaccination status should be included as important predictors in the model.
- If developing separate models for different subgroups is a consideration, you might want to stratify the analysis by severity or vaccination status.

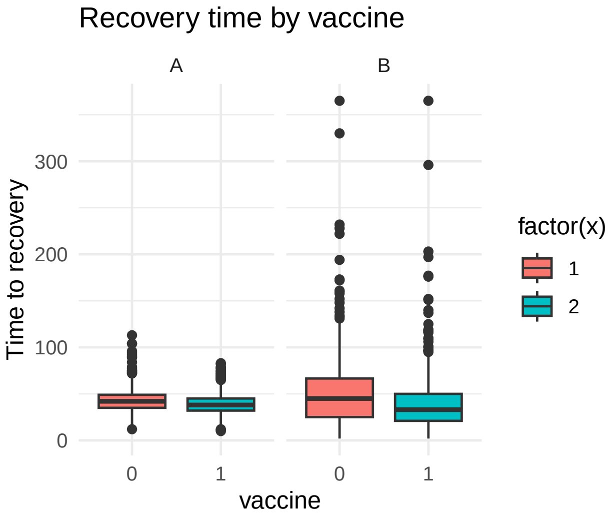
Faceted grid plot for categorical variables

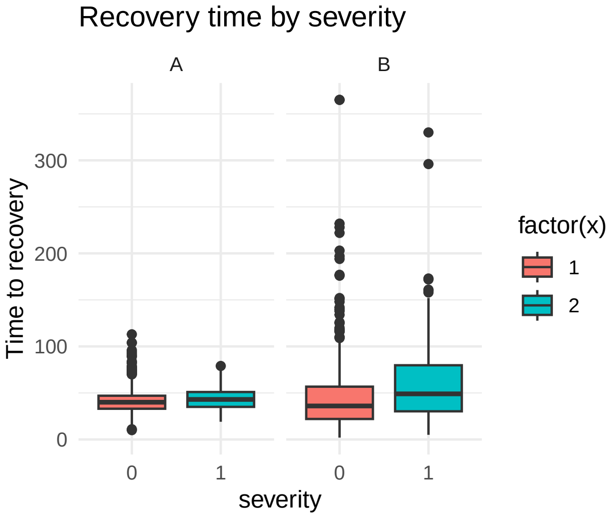
```r
# faceted grid plot for categorical variables
categorical_vars <- c("smoking", "diabetes", "hypertension", "vaccine", "severity")
faceted_plots <- lapply(categorical_vars, function(var) {
  ggplot(dat, aes_string(x = var, y = "recovery_time")) +
    geom_boxplot(aes(fill = factor(..x..))) +
    facet_wrap(~study) +
    labs(title = paste("Recovery time by", var), y = "Time to recovery") +
    theme_minimal()
})

# combine the plots into one grid
grid.arrange(scatter_bmi, scatter_sbp, scatter_ldl, grobs = faceted_plots, ncol = 2)
```


Recovery time by smoking


Recovery time by diabetes


Recovery time by hypertension


Recovery time by vaccine


Recovery time by severity

The boxplots indicate a significant difference in recovery times between study groups A and B across several categorical factors, which suggests that `study` is an important variable to include in the model.

Preprocess of the Data

```
data <- dat

# normalize/standardize numerical variables
#num_vars <- names(data)[sapply(data, is.numeric)][-7]
#preprocess_params <- preProcess(data[, num_vars], method = c("center", "scale"))
#data[num_vars] <- predict(preprocess_params, data[, num_vars])

# log transform 'recovery_time' since it's highly skewed
#data$recovery_time <- log(data$recovery_time)
```

Split data & Define the control method

```
# split data into training and test sets
set.seed(123)
indexes <- createDataPartition(data$recovery_time, p = 0.8, list = FALSE)
train_data <- data[indexes, ]
test_data <- data[-indexes, ]

# matrix of predictors
x <- train_data %>% select(-recovery_time)
y <- train_data$recovery_time

# define the control method for training
ctrl1 <- trainControl(method = "cv", number = 10) # 10-fold cross-validation
```

Model Training Procedure and Final Model:

1. Data Splitting: The dataset was split into training (80%) and test (20%) sets using a stratified random sampling approach based on `recovery_time`.
2. The `train` function from the `caret` package was used to train the MARS model using 10-fold cross-validation. This approach helps to prevent overfitting and gives an estimate of the model performance on new data.
3. The model with the lowest cross-validated Root Mean Squared Error (RMSE) was selected as the final model.

Export the the training/test set & control method

```
# save the training and test sets to CSV files
write.csv(train_data, "./Data/train_data.csv", row.names = FALSE)
write.csv(test_data, "./Data/test_data.csv", row.names = FALSE)

# save the control method using saveRDS
saveRDS(ctrl1, "./Data/train_control.rds")
```

Load the training/test set & control method

```
# Load the training and test sets
train_data <- read.csv("./Data/train_data.csv")
test_data <- read.csv("./Data/test_data.csv")

# Load the control method
```

```r
ctrl1 <- readRDS("./Data/train_control.rds")

# change variables to be factors again
train_data <- train_data %>%
  mutate(gender = as_factor(gender),
         diabetes = as_factor(diabetes),
         hypertension = as_factor(hypertension),
         vaccine = as_factor(vaccine),
         severity = as_factor(severity))

test_data <- test_data %>%
  mutate(gender = as_factor(gender),
         diabetes = as_factor(diabetes),
         hypertension = as_factor(hypertension),
         vaccine = as_factor(vaccine),
         severity = as_factor(severity))
```