Predicting Recovery Time of Patients with Covid-19

Data Science II Final Project

Yi Huang, Yuchen Zhang, Shun Xie

May 8, 2023

Abstract

This project applies varies regression and classfication methods to predict the recovery time of participants with Covid-19 to estimate the recovery time of these participants. The ultimate goal is to develop a prediction model for recovery time and identify important risk factors for long recovery time. The dataset is from a study that combines three existing cohort studies that have been tracking participants for several years.

1 Introduction

1.1 Background

To gain a better understanding of the factors that predict recovery time from COVID-19 illness, a study was designed to combine three existing cohort studies that have been tracking participants for several years. The study collects recovery information through questionnaires and medical records, and leverages existing data on personal characteristics prior to the pandemic. The ultimate goal is to develop a prediction model for recovery time and identify important risk factors for long recovery time.

1.2 Data Description

The dataset is a random sample of 3593 participants draw from `"recovery.RData` includes 16 variables and 3593 observations.

Description of each variable:

- Gender (gender): 1 = Male, 0 = Female;
- Race/ethnicity (race): 1 = White, 2 = Asian, 3 = Black, 4 = Hispanic;
- Smoking (smoking): Smoking status; 0 = Never smoked, 1 = Former smoker, 2 = Current smoker;
- Height (height): Height (in centimeters);
- Weight (weight): Weight (in kilograms);
- BMI (bmi): Body Mass Index; BMI = weight (in kilograms) / height (in meters) squared;
- Hypertension (hypertension): 0 = No, 1 = Yes;
- Diabetes (diabetes): 0 = No, 1 = Yes;
- Systolic blood pressure (SBP): Systolic blood pressure (in mm/Hg);
- (10)LDL cholesterol (LDL): LDL (low-density lipoprotein) cholesterol (in mg/dL);
- Vaccination status at the time of infection (vaccine): 0 = Not vaccinated, 1 = Vaccinated;
- Severity of COVID-19 infection (severity): 0 = Not severe, 1= Severe;
- Study (study): The study (A/B/C) that the participant belongs to;
- Age: age of participants;
- Time to recovery (tt_recovery_time): Time from COVID-19 infection to recovery in days;
- ID: unique id of each participant.

1.3 Data Cleaning

Package `tidyverse` and `dplyr` are used for data cleaning, wrangling, and manipulation. Variable `ID` is omitted before data wrangling. Table 1 includes the summary of all variables including 6 continuous and 9 categorical predictors and 1 continuous response. Based on the summary, there is no missing data. Then, factor all the categorical data before data partition. Apply `caret` package to split the data into 70% train data and 30% test data. Based on the Table 1, the distribution of predictors are quite similar across different studies.

Table 1: **Summary of Dataset by Study Group**

Characteristic	**A**, N = 729	**B**, N = 2,179	**C**, N = 685
age	60.0 (57.0, 63.0)	60.0 (57.0, 63.0)	60.0 (57.0, 63.0)
height	170 (166, 174)	170 (166, 174)	170 (166, 174)
weight	79 (75, 84)	80 (75, 85)	80 (76, 85)
bmi	27.50 (25.80, 29.30)	27.60 (25.80, 29.50)	27.80 (25.90, 29.50)
SBP	130 (125, 136)	130 (125, 136)	129 (124, 135)
LDL	111 (98, 125)	110 (97, 124)	110 (97, 123)
recovery_time	40 (33, 47)	38 (24, 55)	39 (33, 45)
gender			
0	383 (53%)	1,099 (50%)	358 (52%)
1	346 (47%)	1,080 (50%)	327 (48%)

Characteristic	**A**, N = 729	**B**, N = 2,179	**C**, N = 685
race			
1	491 (67%)	1,408 (65%)	447 (65%)
2	35 (4.8%)	107 (4.9%)	33 (4.8%)
3	146 (20%)	454 (21%)	142 (21%)
4	57 (7.8%)	210 (9.6%)	63 (9.2%)
smoking			
0	445 (61%)	1,316 (60%)	424 (62%)
1	210 (29%)	632 (29%)	201 (29%)
2	74 (10%)	231 (11%)	60 (8.8%)
hypertension			
0	378 (52%)	1,112 (51%)	399 (58%)
1	351 (48%)	1,067 (49%)	286 (42%)
diabetes			
0	606 (83%)	1,833 (84%)	573 (84%)
1	123 (17%)	346 (16%)	112 (16%)
vaccine			
0	299 (41%)	854 (39%)	281 (41%)
1	430 (59%)	1,325 (61%)	404 (59%)
severity			
0	640 (88%)	1,951 (90%)	623 (91%)
1	89 (12%)	228 (10%)	62 (9.1%)

2 Exploratory analysis and data visualization:

We consider the EDA for the training data, which we will use later on to train our models. Recovery data contains 6 continuous predictors, namely age, height, weight, bmi, Systolic blood pressure (SBP) and LDL cholesterol (LDL), as well as 8 discrete/factor predictors, namely gender, race, smoking, hypertension, diabetes, vaccine status, Severity of COVID-19 infection and study which the data belongs to. The response value is the time for recovery. It is a continuous variable.

Distribution of data is included in appendix 1. In accordance to distribution of continuous variables, only the recovery time, which is the response, has a right skewed shape. All predictors are normally distributed with symmetric density plot.

The correlation plot (only considers continuous variable), calculated from Pearson Correlation in appendix 2, gives an insight of linear correlation between continuous predictors and transformed recovery time. It can be seen that bmi and weight have a relatively weak positive relation with log recovery time whereas height may have a weak negative linear correlation with log recovery time. While linear trend may be sufficient in some cases, some data may show a non-linear correlation with log recovery time. As shown in appendix 3, weight and bmi is likely to have a non-linear correlation with the response, suggesting a motivation to consider non-linear method. Box plot in appendix 4 illustrates that most data have a small difference in median. People with smoking is likely to have a higher recovery time as it has a higher median and people that are vaccinated has a lower median recovery time than the people without vaccination. Thus, we include these factors in our study of predicting recovery time.

3 Model training for regression

This section describes the models used for predicting time to recovery from COVID-19. State the assumptions made by using the models and detailed description of the model training procedure and how to obtained the final model.

This project performs varies regression models on the dataset, including linear regression, K-Nearest Neighbors (KNN), elastic net, partial least squares (pls), generalized additive model (gam), Multivariate

Adaptive Regression (mars), random forest, and boosting using 10-fold cross-validation results on train data. The model with the minimum mean of train RMSE from cross-validation results will be the optimal model. All models are implemented in `train()` function from `caret` package. To improve computation efficiency, parallel computing function from `doParallel` package is utilized.

1. Linear Model: Linear model assumes a linear relationship between the predictor and response variables (linearity), and assumes that the errors are normally distributed (normality) and have constant variance (homoscedasticity), and that the observations are independent of each other. The linear model is the most basic and assumes a linear relationship between the variables, while the other models allow for more flexible relationships. A method = "lm" argument was used within the `train()` function to specify a linear model fit.

2. KNN (K-Nearest Neighbors): KNN is a non-parametric machine learning algorithm that can be used for classification or regression tasks. The algorithm works by finding the K closest data points to a new input, and using their output values to predict the output value for the new input. It assume data points which exist in close proximity to each other are highly similar, while if a data point is far away from another group it's dissimilar to those data points. With `train()` function to specify the KNN model and use `expand.grid()` with sequence from 1 to 40 by 1.

3. Elastic Net Regression: Elastic Net combines the L1 and L2 regularization methods to prevent overfitting in the model by adding L1 and L2 penalties. Elastic net regression models have the same assumption as linear model. A method = "glmnet" argument was used within the `train()`function to specify model fit. Additionally, a grid of tuning parameters for these models were set using the "tuneGrid" argument within the `train()` function. The grid contains 100 values of the lambda parameter, ranging from exp(-10) to exp(-5) with equal intervals on the log scale for elastic net.

4. Partial Least Squares (PLS): PLS is a multivariate regression method that is used to model the relationship between two sets of variables. The method works by finding the linear combination of the input variables that best explains the variation in the output variables. A method = "pls" argument was used within the `train()` function to specify a PLS model fit. Additionally, a tuning parameter for the model were set using the tuneGrid argument within the `train()` function. The tuneGrid object includes a data frame with a single column ncomp that ranges from 1 to 16, representing the number of components used in the model. It has the same assumptions as the linear model. The preProcess argument is set to "center" and "scale", which means that the training data will be centered and scaled prior to model fitting.

5. Generalized Additive Model (GAM): GAM is a type of regression model that can capture non-linear relationships between the input and output variables. The method works by modeling the input variables as a sum of smooth functions. It has the same assumptions as the linear model. A method = "gam" argument was used within the `train()` function to specify a GAM fit.

6. Multivariate Adaptive Regression Splines (MARS): MARS can model non-linear relationships between the input and output variables. The method works by fitting piecewise linear or nonlinear functions to the data. It has the same assumptions as the linear model. A method = "earth" argument was used within the `train()` function to specify a MARS model fit. Additionally, the `expand.grid()` function is used to generate a grid of tuning parameters. The mars_grid object includes two arguments: degree is set to 1, 2, and 3, representing the number of possible product hinge functions in a single term, and nprune is set to integers between 2 and 20, representing the upper bound on the number of terms in the model. The tuneGrid argument in the `train()` function uses the mars_grid object to specify the parameters for model tuning.

7. Random Forest, and Boosting Regression: Random Forest used for regression and classification tasks. The algorithm works by training an ensemble of decision trees on different subsets of the training data and then combining their predictions. Boosting can be used to improve the accuracy of a model by combining weak learners into a strong learner. The method works by training a sequence of models

on the training data, with each model focusing on the instances that were misclassified by the previous model. Use`randomForest()` function to fit Bagging and random forest models, and `train()` function for the boosting regression with `expand.grid()` function for generating a grid of tuning parameters.

4 Results for regression

After resampling, 8 models share a common set of resampled datasets. The boosting model has least mean value of RMSE. The RMSE of all four model are showed below.

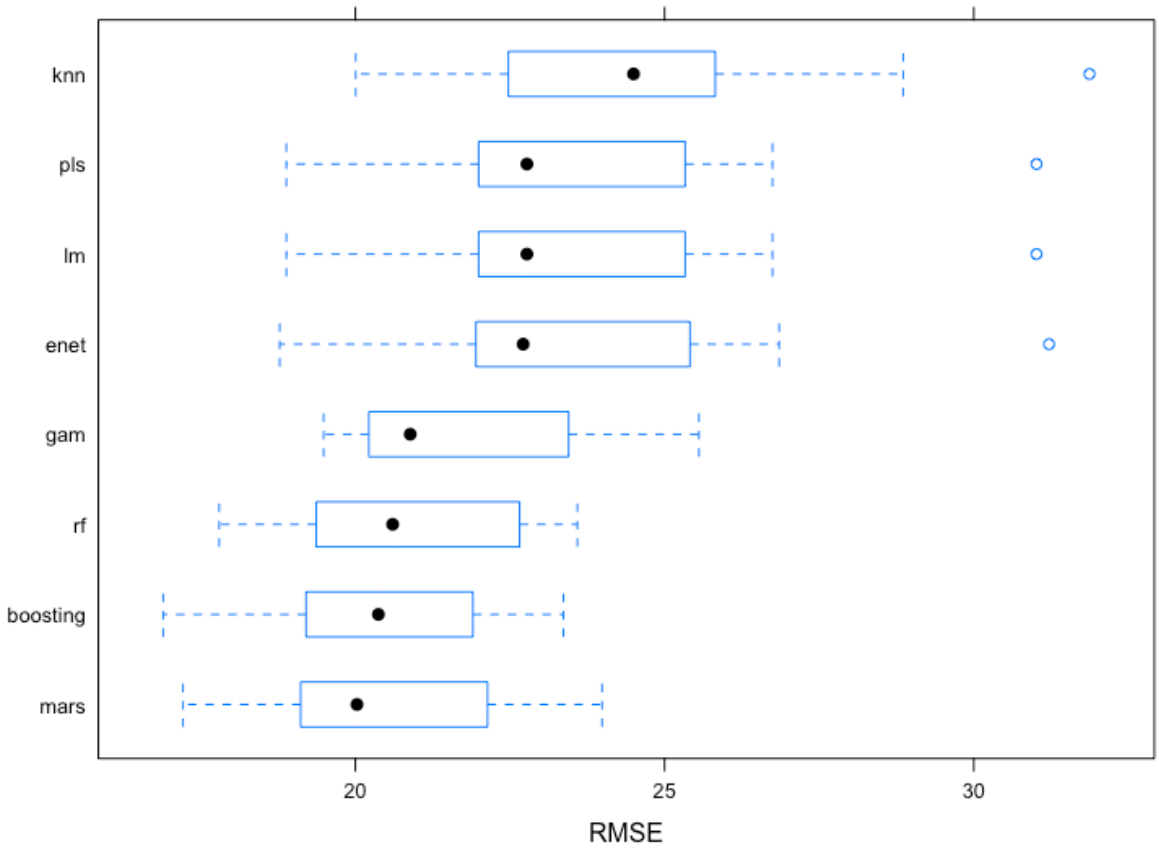


Figure 1: A caption

variable importance and partial dependence plots for interpretation.

Variable importance plot indicates that the bmi, study and height are most influential in the model among all the predictors.

Interpretions of some significant coefficients: Hold other condition same, people have higher value of bmi, people who are in Study B and people have higher height are expected to have more recovery time.

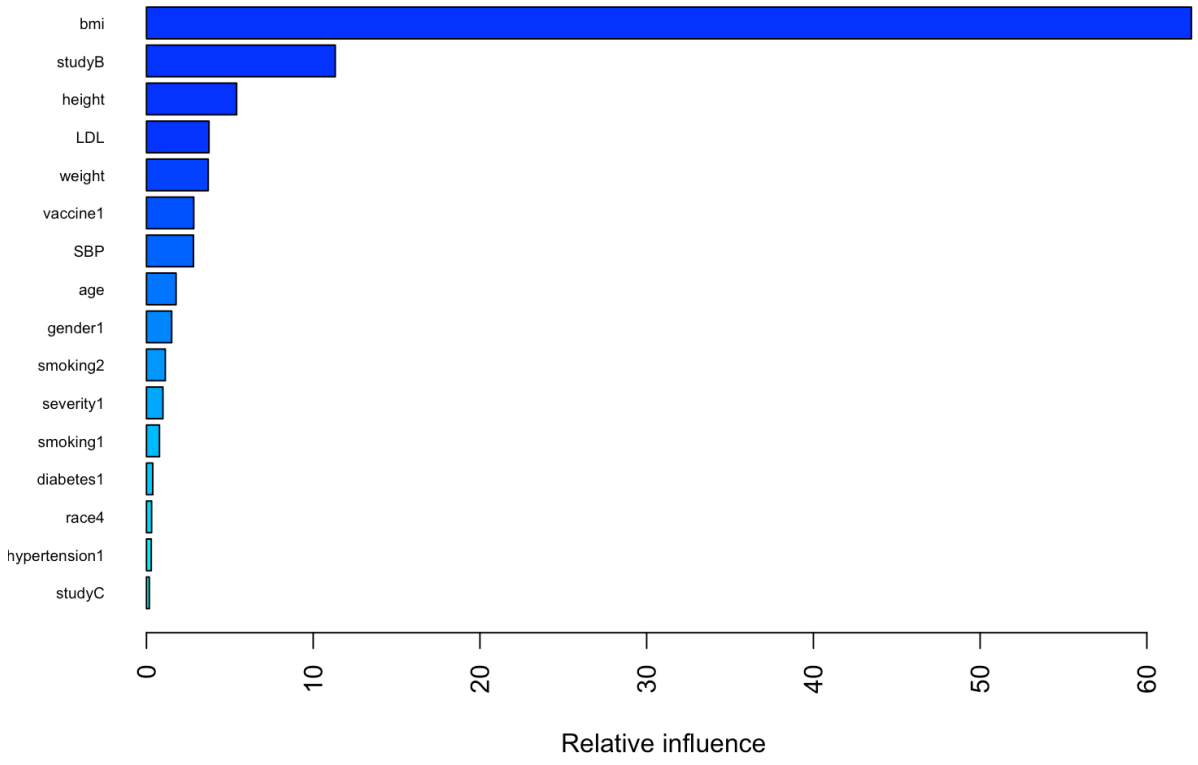


Figure 2: A caption

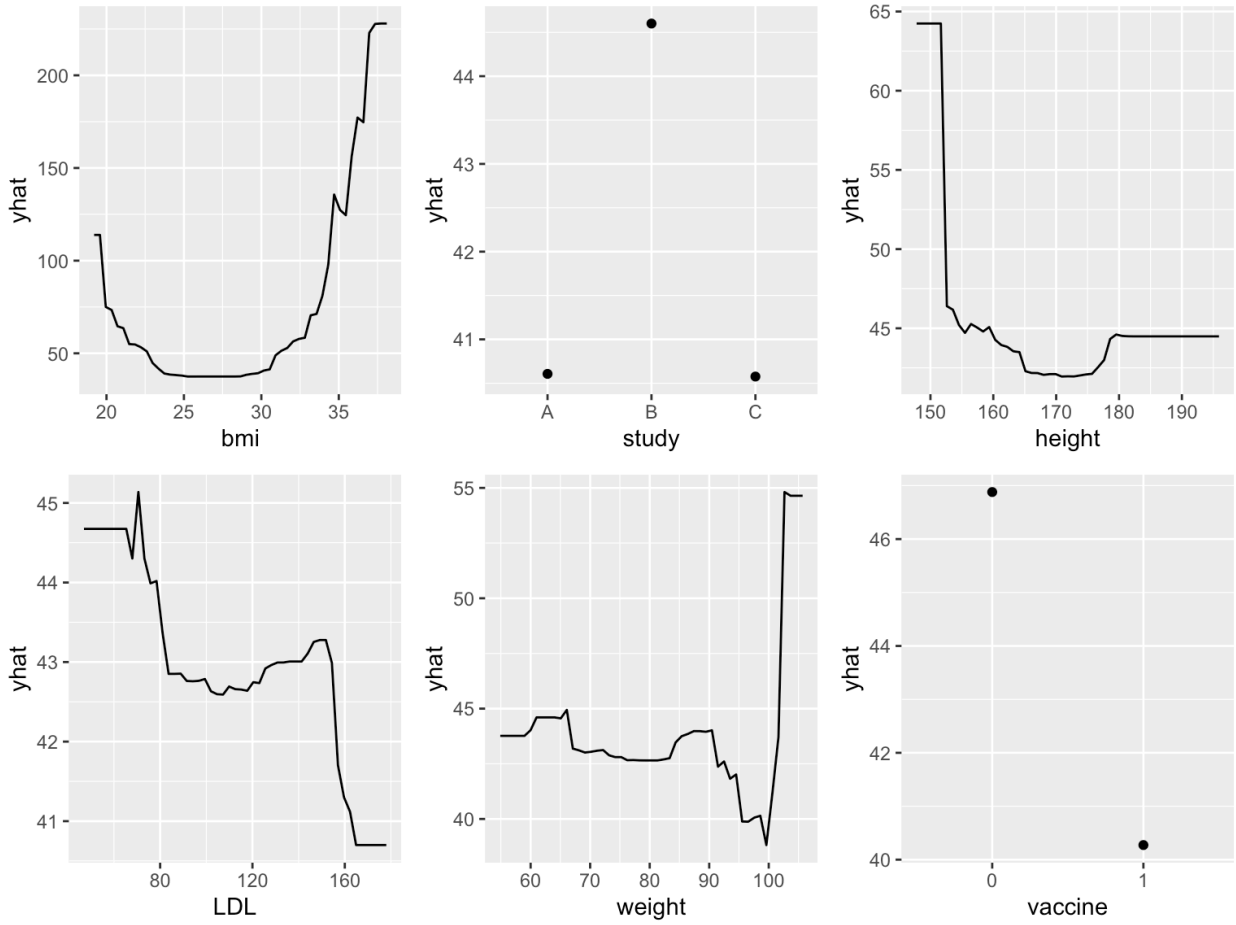


Figure 3: A caption

5 Model training for classification

This section describes the models used for predicting time to recovery from COVID-19. State the assumptions made by using the models and detailed description of the model training procedure and how to obtained the final model.

6 Results for classification

In this section, report the final model that you built for predicting time to recovery from COVID-19. Interpret the results. Assess the model's training/test performance.

7 Conclusions and discussion

References

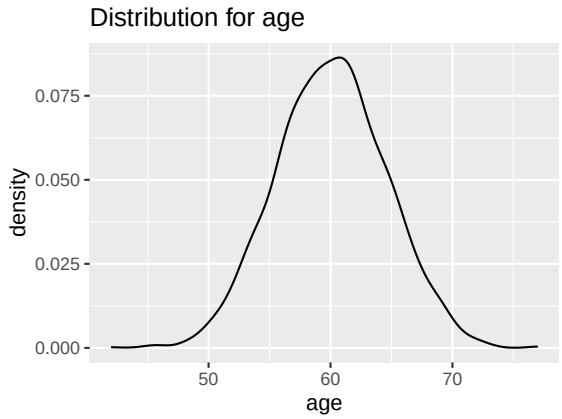
CDC Covid Data Tracker. (2021), Centers for Disease Control and Prevention.

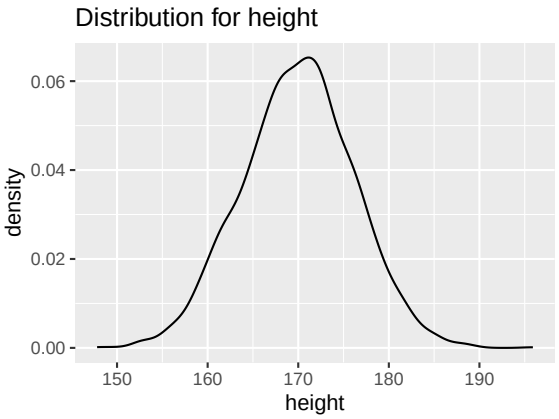
James, Gareth, e. a. (2021), An Introduction to Statistical Learning: With Applications in R., Springer.

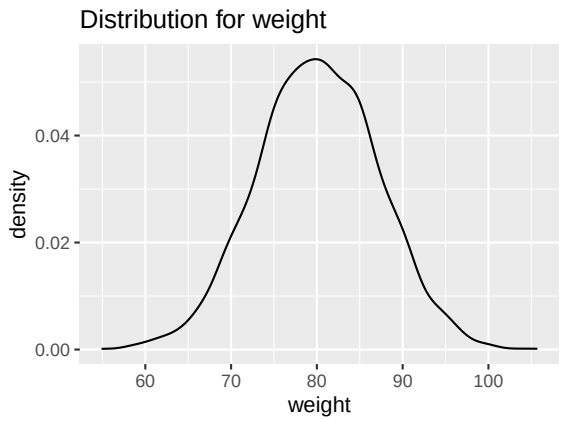
Pradhan, A. and Olsson., P.-E. (2021), 'Sex differences in severity and mortality from covid- 19: are males more vulnerable?', Biology of sex differences 11(1), 53.

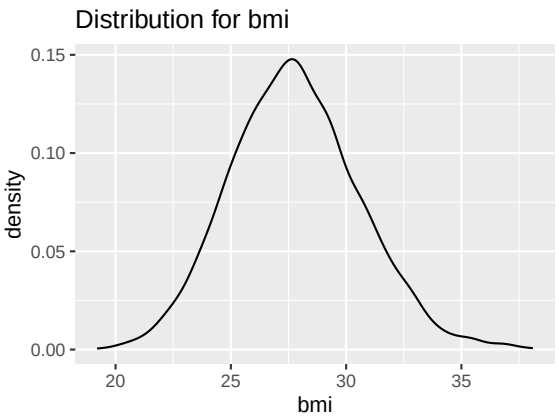
Appendix

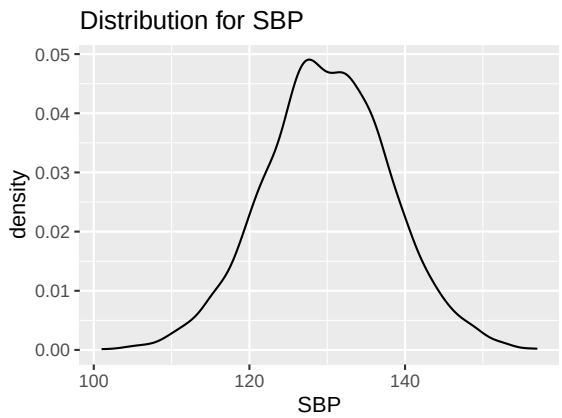
Appendix 1

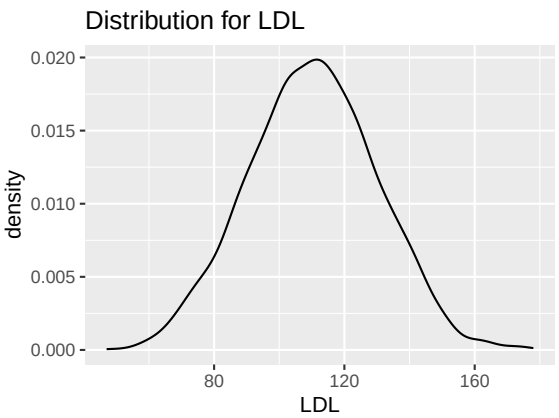

Distribution for age
density
0.075
0.050
0.025
0.000
50
60
70
age

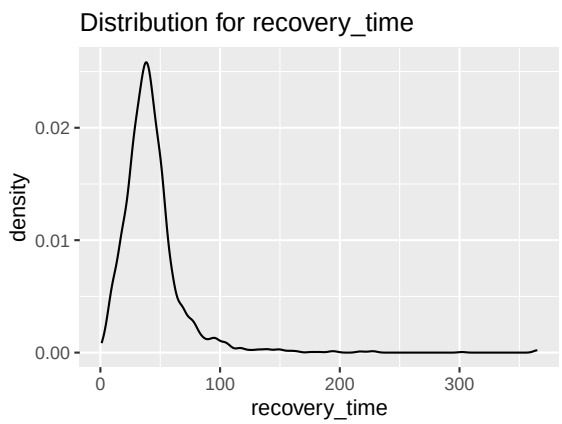

Distribution for height
density
0.06
0.04
0.02
0.00
150
160
170
180
190
height

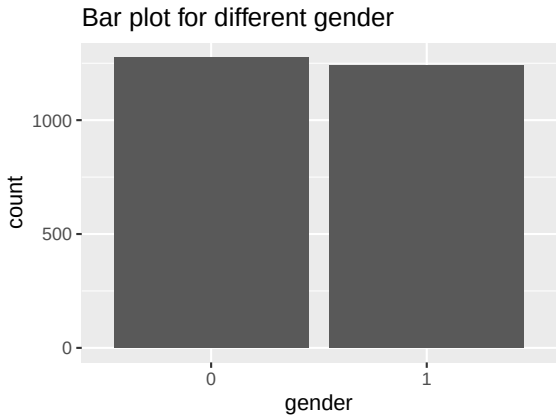

Distribution for weight
density
0.04
0.02
0.00
60
70
80
90
100
weight

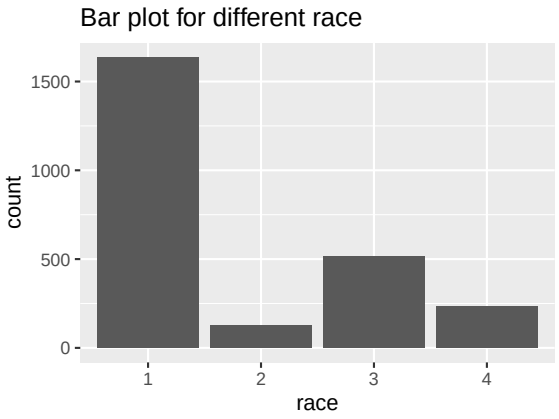

Distribution for bmi
density
0.15
0.10
0.05
0.00
20
25
30
35
bmi

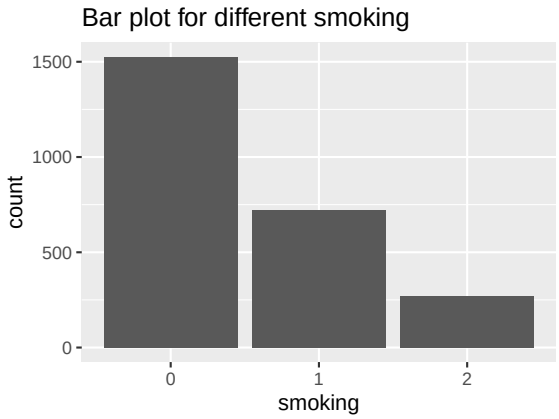

Distribution for SBP
density
0.05
0.04
0.03
0.02
0.01
0.00
100
120
140
SBP

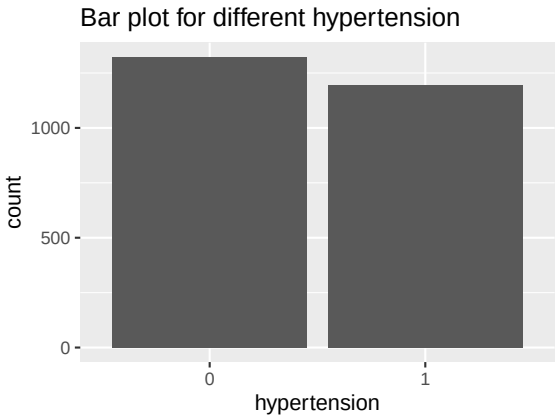

Distribution for LDL
density
0.020
0.015
0.010
0.005
0.000
80
120
160
LDL

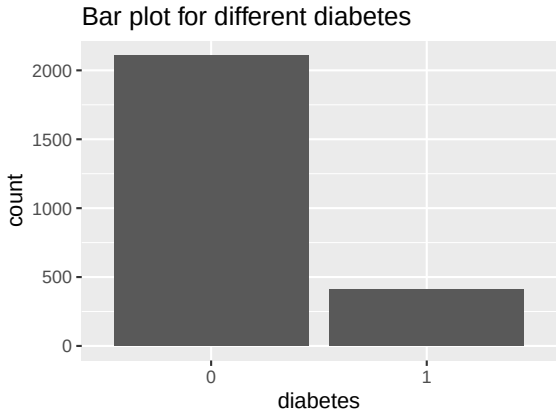

Distribution for recovery_time
density
0.02
0.01
0.00
0
100
200
300
recovery_time

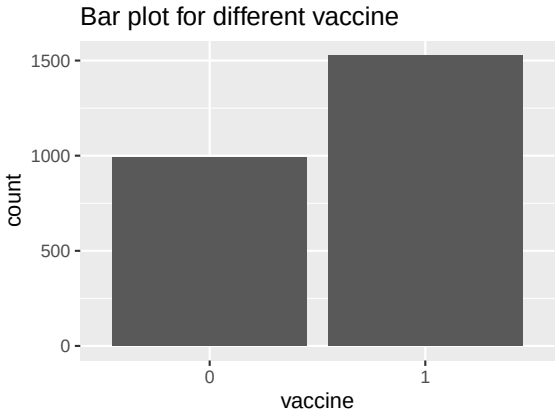

Bar plot for different gender
count
1000
500
0
0
1
gender

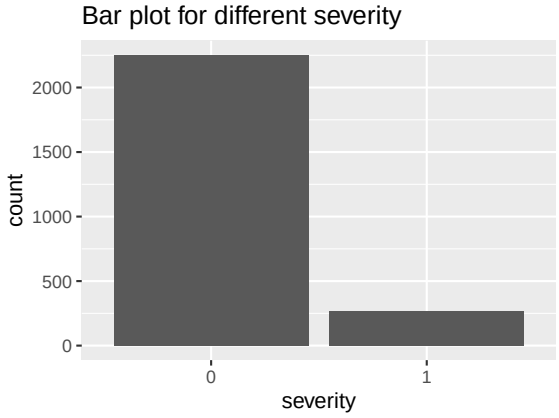

Bar plot for different race
count
1500
1000
500
0
1
2
3
4
race

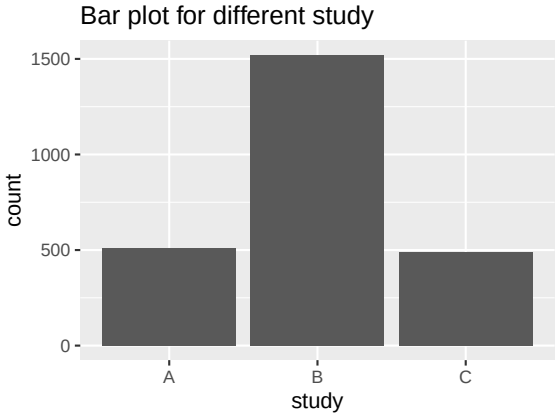

Bar plot for different smoking
count
1500
1000
500
0
0
1
2
smoking


Bar plot for different hypertension
count
1000
500
0
0
1
hypertension

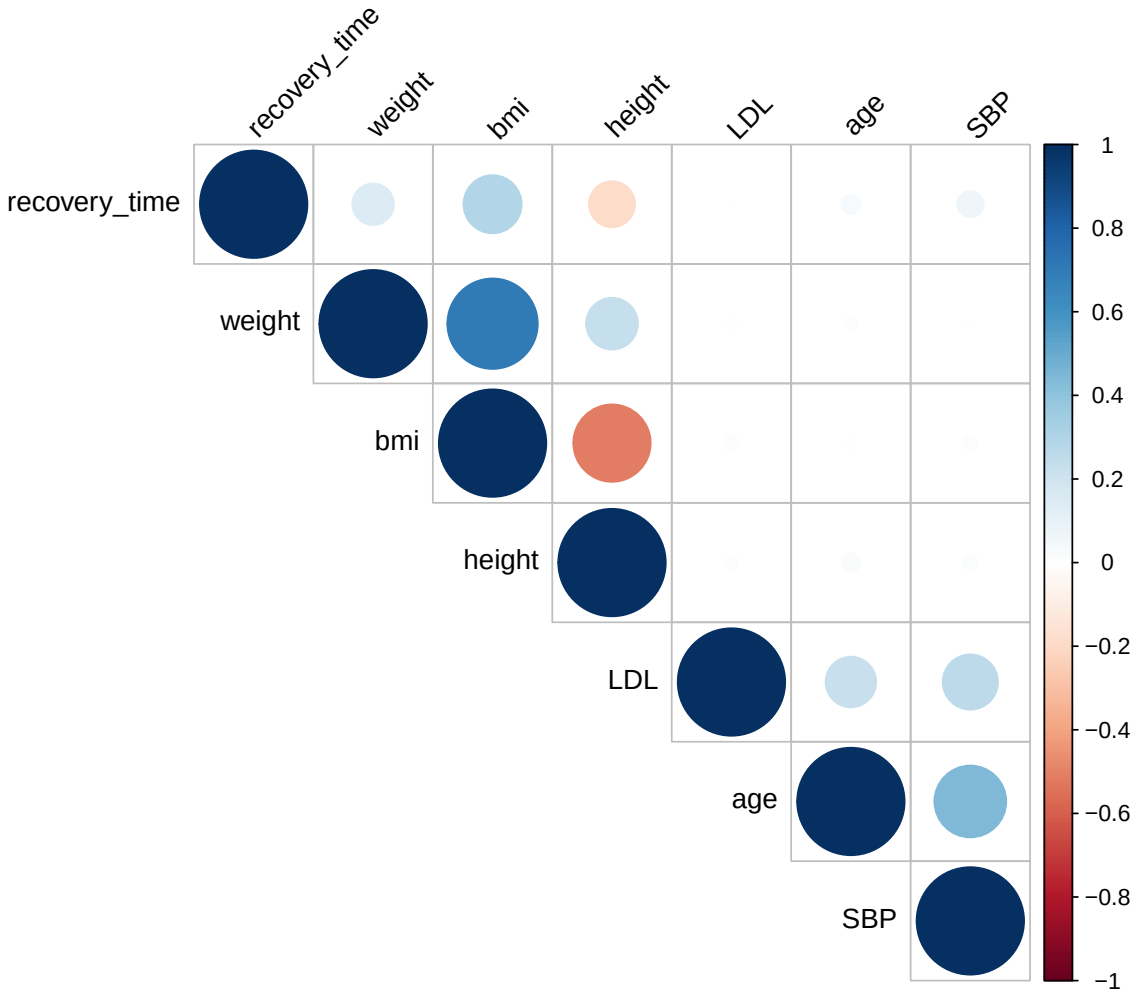

Bar plot for different diabetes
count
2000
1500
1000
500
0
0
1
diabetes


Bar plot for different vaccine
count
1500
1000
500
0
0
1
vaccine

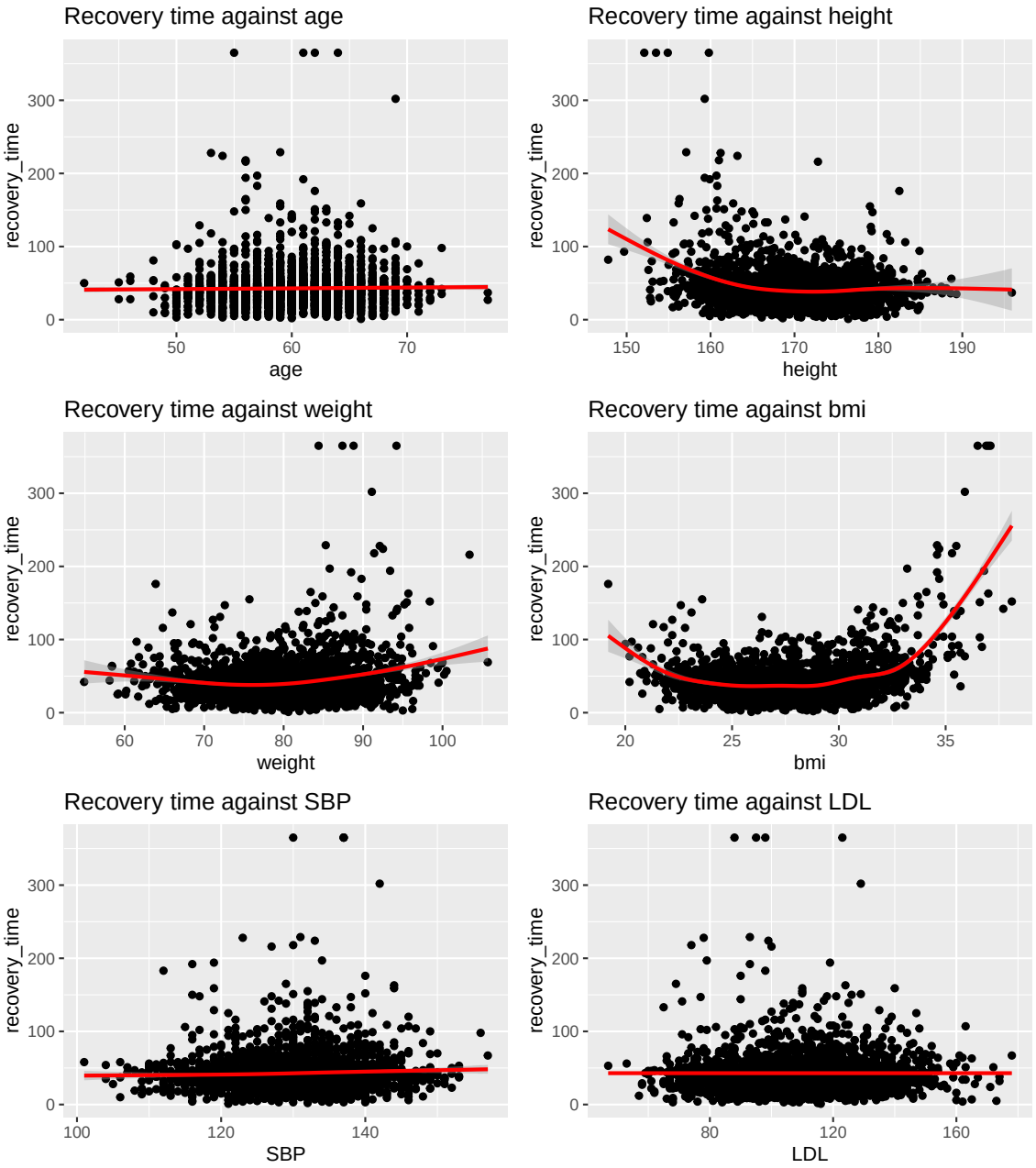

Bar plot for different severity
count
2000
1500
1000
500
0
0
1
severity


Bar plot for different study
count
1500
1000
500
0
A
B
C
study

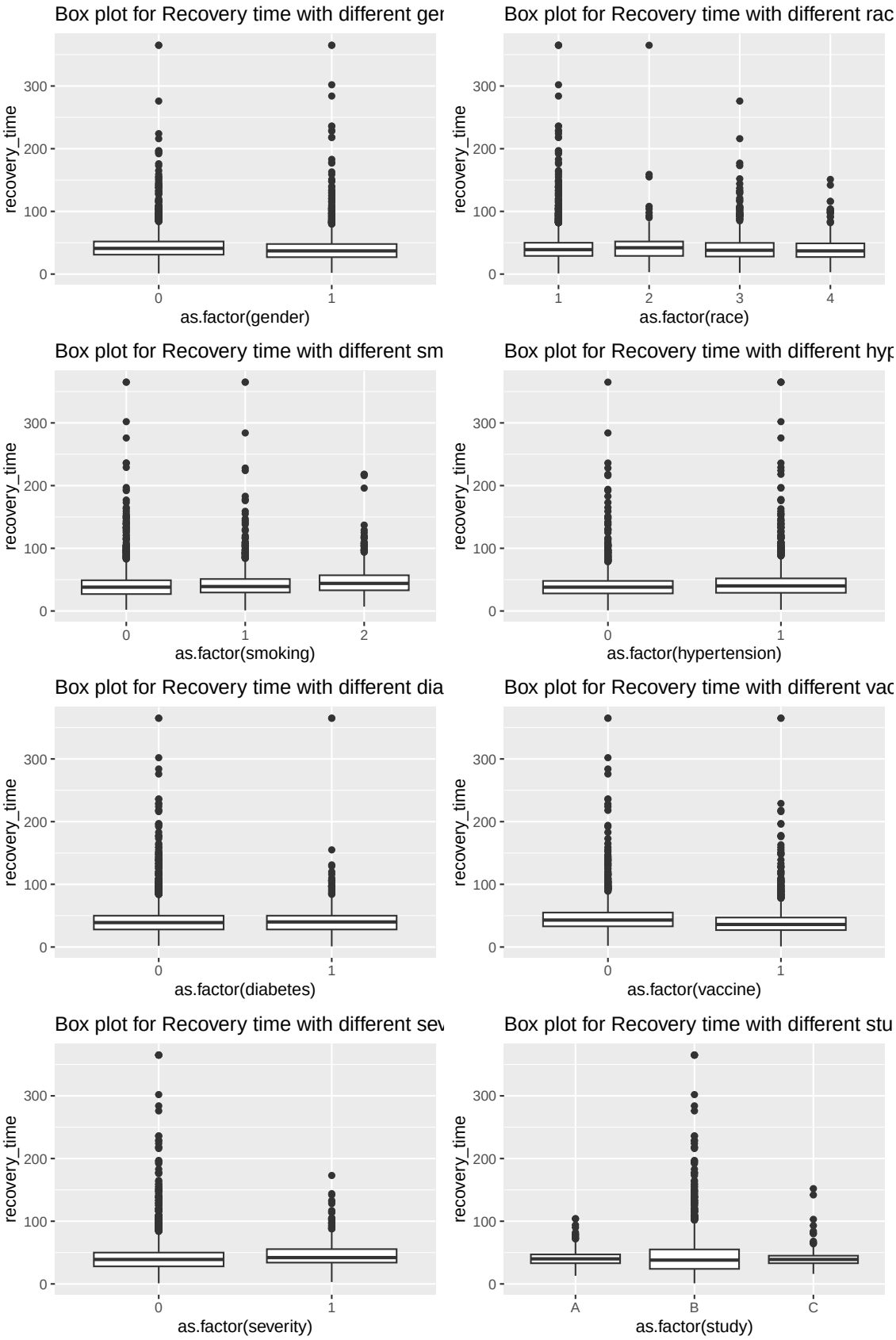
Appendix 2


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

Appendix 3


Recovery time against age
recovery_time
age
Recovery time against height
recovery_time
height
Recovery time against weight
recovery_time
weight
Recovery time against bmi
recovery_time
bmi
Recovery time against SBP
recovery_time
SBP
Recovery time against LDL
recovery_time
LDL

Appendix4


Box plot for Recovery time with different ger
recovery_time
as.factor(gender)
Box plot for Recovery time with different rac
as.factor(race)
Box plot for Recovery time with different sm
as.factor(smoking)
Box plot for Recovery time with different hyp
as.factor(hypertension)
Box plot for Recovery time with different dia
as.factor(diabetes)
Box plot for Recovery time with different vac
as.factor(vaccine)
Box plot for Recovery time with different sev
as.factor(severity)
Box plot for Recovery time with different stu
as.factor(study)

A Appendix: Figure

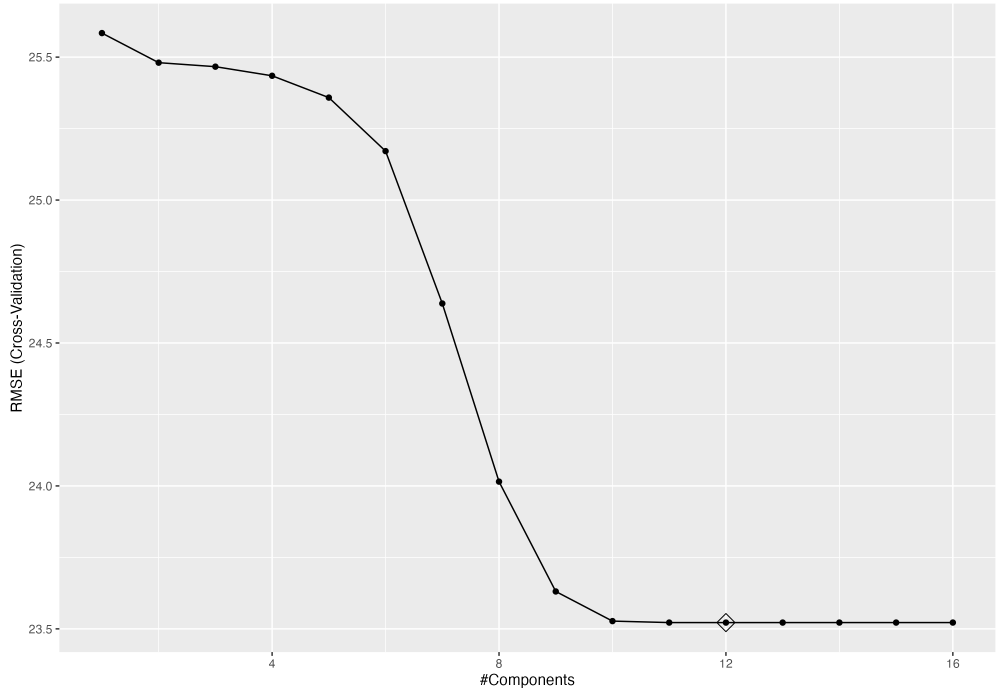


Figure 4: example

B Appendix: Code