8. Appendix for R Code

Table of Contents

1. Preprocessing - 1
2. Exploratory Data Analysis - 3
3. Data Modelling - 10
4. Model Selection using ROC Analysis - 17
5. Analyze the Best Performing Model - Logistic Regression - 19

Binary Response: Death_Event (0 – Alive, 1 – Death)

1. Preprocessing

```
# load the csv data
heart_data <- read.csv("heart_failure_clinical_records_dataset.csv")

# Check if there is any missing value (number of missing values = 0)
sum(is.na(heart_data))
```

```
## [1] 0
```

```
# Check nrows and ncols
dim(heart_data)
```

```
## [1] 299  13
```

1.1 Drop time

```
heart_data <- subset(heart_data, select = -time)
dim(heart_data)
```

```
## [1] 299  12
```

1.2 Define Baseline for Categorical Variables -> Factor Type

```
heart_data$anaemia <- factor(heart_data$anaemia, levels=c("0","1"), labels=c("No","Yes"))
heart_data$diabetes <- factor(heart_data$diabetes, levels=c("0","1"), labels=c("No","Yes"))
heart_data$high_blood_pressure <- factor(heart_data$high_blood_pressure,
                                         levels=c("0","1"), labels=c("No","Yes"))
```

```
heart_data$sex <- factor(heart_data$sex, levels=c("0","1"), labels=c("F","M"))
heart_data$smoking <- factor(heart_data$smoking, levels=c("0","1"), labels=c("No","Yes"))
heart_data$DEATH_EVENT <- factor(heart_data$DEATH_EVENT, levels=c("0","1"),
                                 labels=c("Alive","Death"))

# Check the data type to see the changes
str(heart_data)
```

```
## ’data.frame':    299 obs. of  12 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 2 2 1 2 ...
##  $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes                : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 2 1 1 ...
##  $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 2 1 1 1 2 ...
##  $ platelets               : num  265000 263358 162000 210000 327000 ...
##  $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
##  $ sex                     : Factor w/ 2 levels "F","M": 2 2 2 2 1 2 2 2 1 2 ...
##  $ smoking                 : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 2 1 2 ...
##  $ DEATH_EVENT             : Factor w/ 2 levels "Alive","Death": 2 2 2 2 2 2 2 2 2 2 ...
```

2. Exploratory Data Analysis

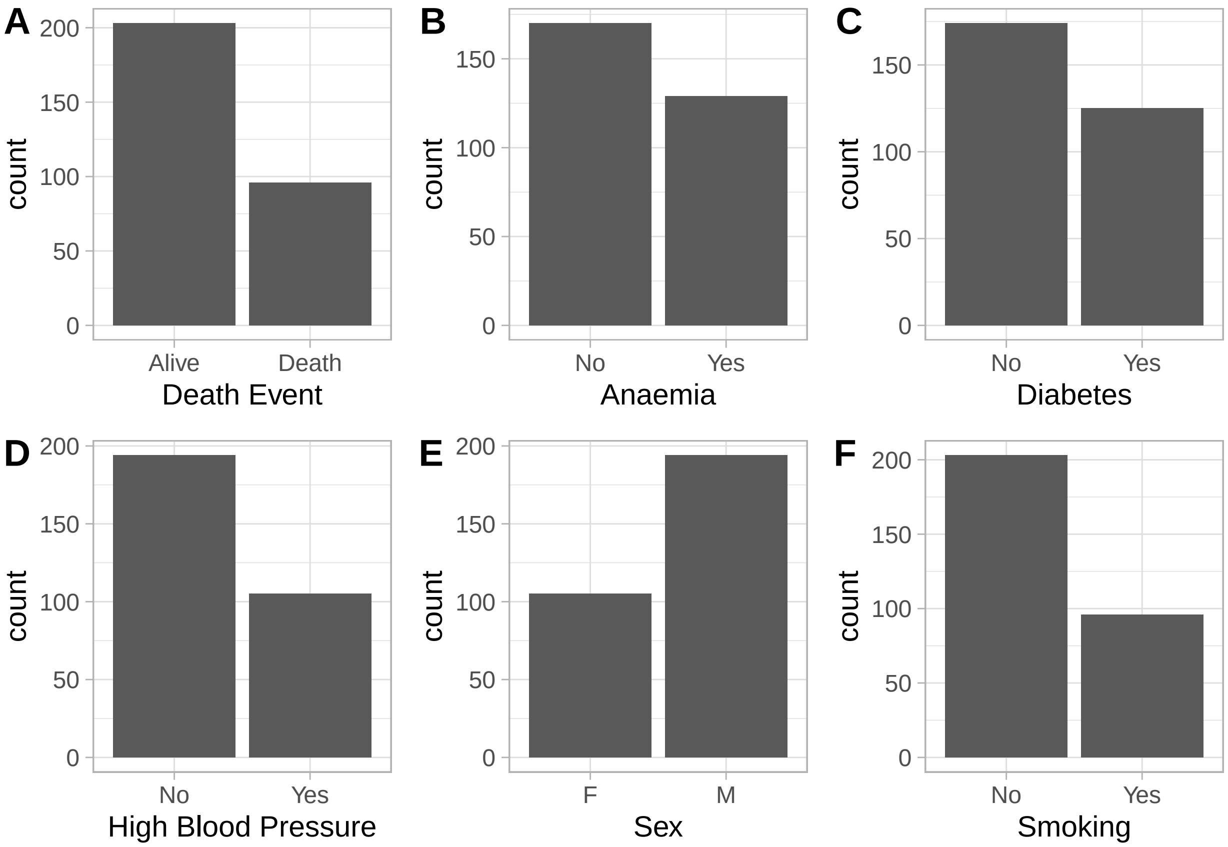
```
# Generate summary statistics
summary(heart_data)
```

```
##       age        anaemia   creatinine_phosphokinase diabetes  ejection_fraction
##  Min.   :40.00   No :170   Min.   :  23.0           No :174   Min.   :14.00
##  1st Qu.:51.00   Yes:129   1st Qu.: 116.5           Yes:125   1st Qu.:30.00
##  Median :60.00             Median : 250.0                     Median :38.00
##  Mean   :60.83             Mean   : 581.8                     Mean   :38.08
##  3rd Qu.:70.00             3rd Qu.: 582.0                     3rd Qu.:45.00
##  Max.   :95.00             Max.   :7861.0                     Max.   :80.00
##  high_blood_pressure   platelets      serum_creatinine  serum_sodium   sex
##  No :194             Min.   : 25100   Min.   :0.500    Min.   :113.0   F:105
##  Yes:105             1st Qu.:212500   1st Qu.:0.900    1st Qu.:134.0   M:194
##                      Median :262000   Median :1.100    Median :137.0
##                      Mean   :263358   Mean   :1.394    Mean   :136.6
##                      3rd Qu.:303500   3rd Qu.:1.400    3rd Qu.:140.0
##                      Max.   :850000   Max.   :9.400    Max.   :148.0
##  smoking   DEATH_EVENT
##  No :203   Alive:203
##  Yes: 96   Death: 96
##
##
##
##
```

2.1 Categorical Variable Distributions

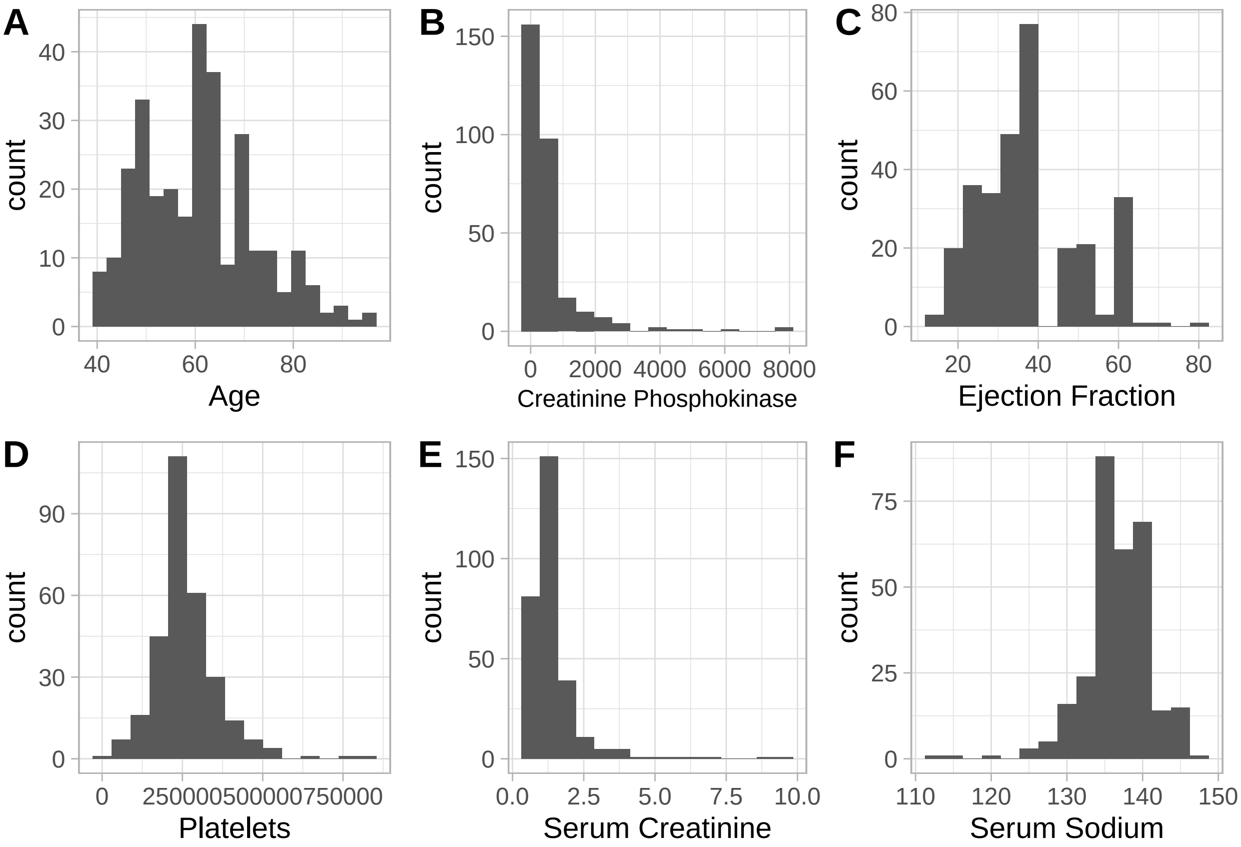
```
suppressMessages(library(tidyverse))

# response
response <- heart_data %>%
  ggplot(aes(x = DEATH_EVENT)) + geom_bar() + labs(x = "Death Event") + theme_light()

# anaemia
anaemia <- heart_data %>%
  ggplot(aes(x = anaemia)) + geom_bar() + labs(x = "Anaemia") + theme_light()

# diabetes
diabetes <- heart_data %>%
  ggplot(aes(x = diabetes)) + geom_bar() + labs(x = "Diabetes") + theme_light()

# high_blood_pressure
high_blood_pressure <- heart_data %>%
  ggplot(aes(x = high_blood_pressure)) + geom_bar() + labs(x = "High Blood Pressure") +
  theme_light()

# sex
sex <- heart_data %>%
  ggplot(aes(x = sex)) + geom_bar() + labs(x = "Sex") + theme_light()
```

```
# smoking
smoking <- heart_data %>%
  ggplot(aes(x = smoking)) + geom_bar() + labs(x = "Smoking") + theme_light()

# library allows clear labels and save plot as pdf
suppressMessages(library(cowplot))
(p_cat <- plot_grid(response, anaemia, diabetes, high_blood_pressure, sex, smoking,
                    labels = "AUTO"))
```



2.2 Continuous Variable Distributions

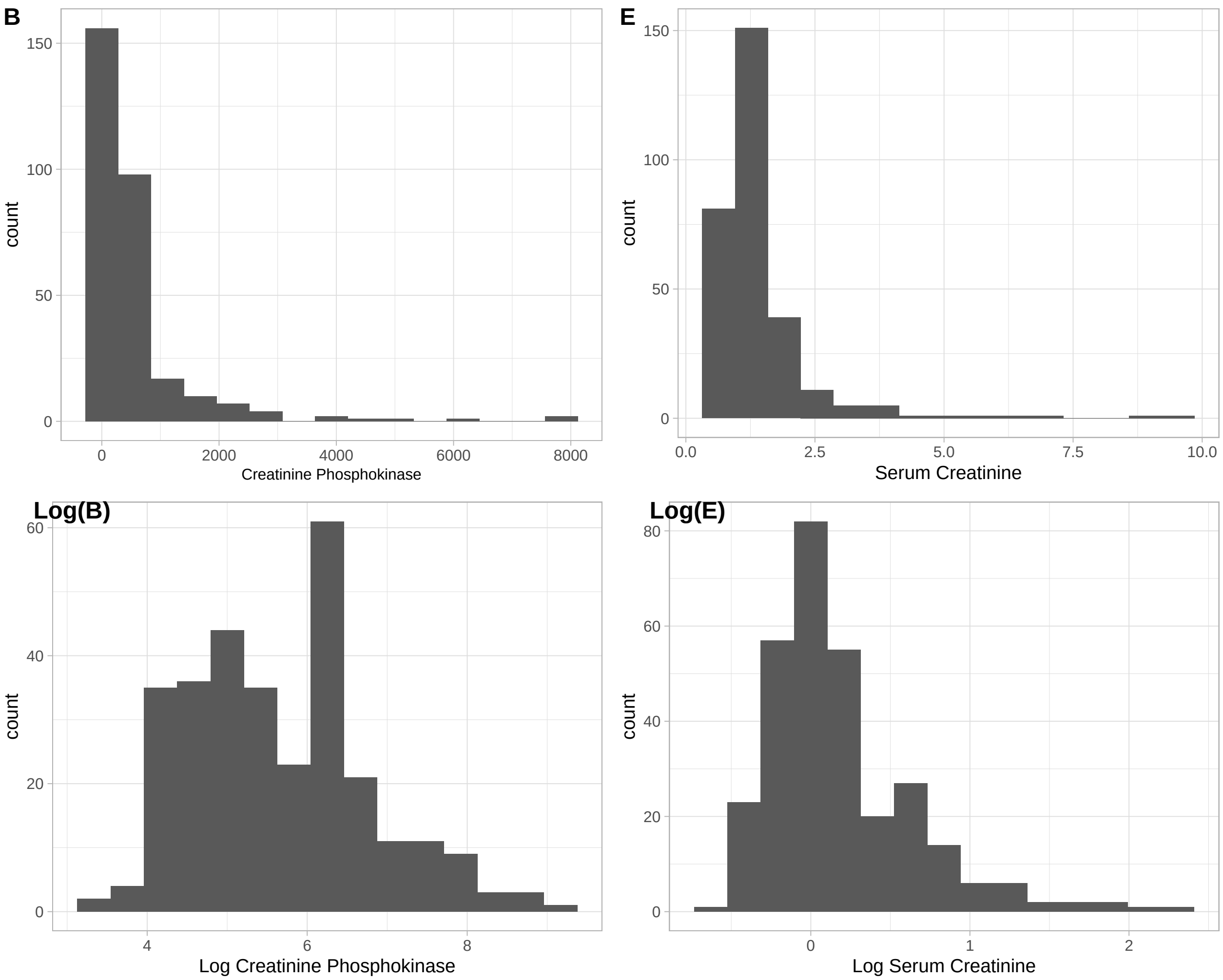
```
# age
age <- heart_data %>%
  ggplot(aes(x = age)) + geom_histogram(bins=20) + labs(x = "Age") + theme_light()

# creatinine_phosphokinase
creatinine_phosphokinase <- heart_data %>%
  ggplot(aes(x = creatinine_phosphokinase)) + geom_histogram(bins=15) +
  labs(x = "Creatinine Phosphokinase") + theme_light() +
  theme(axis.title.x = element_text(size = 9))

# ejection_fraction
ejection_fraction <- heart_data %>%
  ggplot(aes(x = ejection_fraction)) + geom_histogram(bins=15) +
  labs(x = "Ejection Fraction") + theme_light()

# platelets
platelets <- heart_data %>%
  ggplot(aes(x = platelets)) + geom_histogram(bins=15) + labs(x = "Platelets") +
  theme_light()

# serum_creatinine
serum_creatinine <- heart_data %>%
  ggplot(aes(x = serum_creatinine)) + geom_histogram(bins=15) +
  labs(x = "Serum Creatinine") + theme_light()

# serum_sodium
serum_sodium <- heart_data %>%
  ggplot(aes(x = serum_sodium)) + geom_histogram(bins=15) + labs(x = "Serum Sodium") +
  theme_light()

# plot
(p_con <- plot_grid(age, creatinine_phosphokinase, ejection_fraction, platelets,
                    serum_creatinine, serum_sodium, labels = "AUTO"))
```



- B, E are heavily right-skewed, may consider using log transformation

Log Transform Continuous Variable

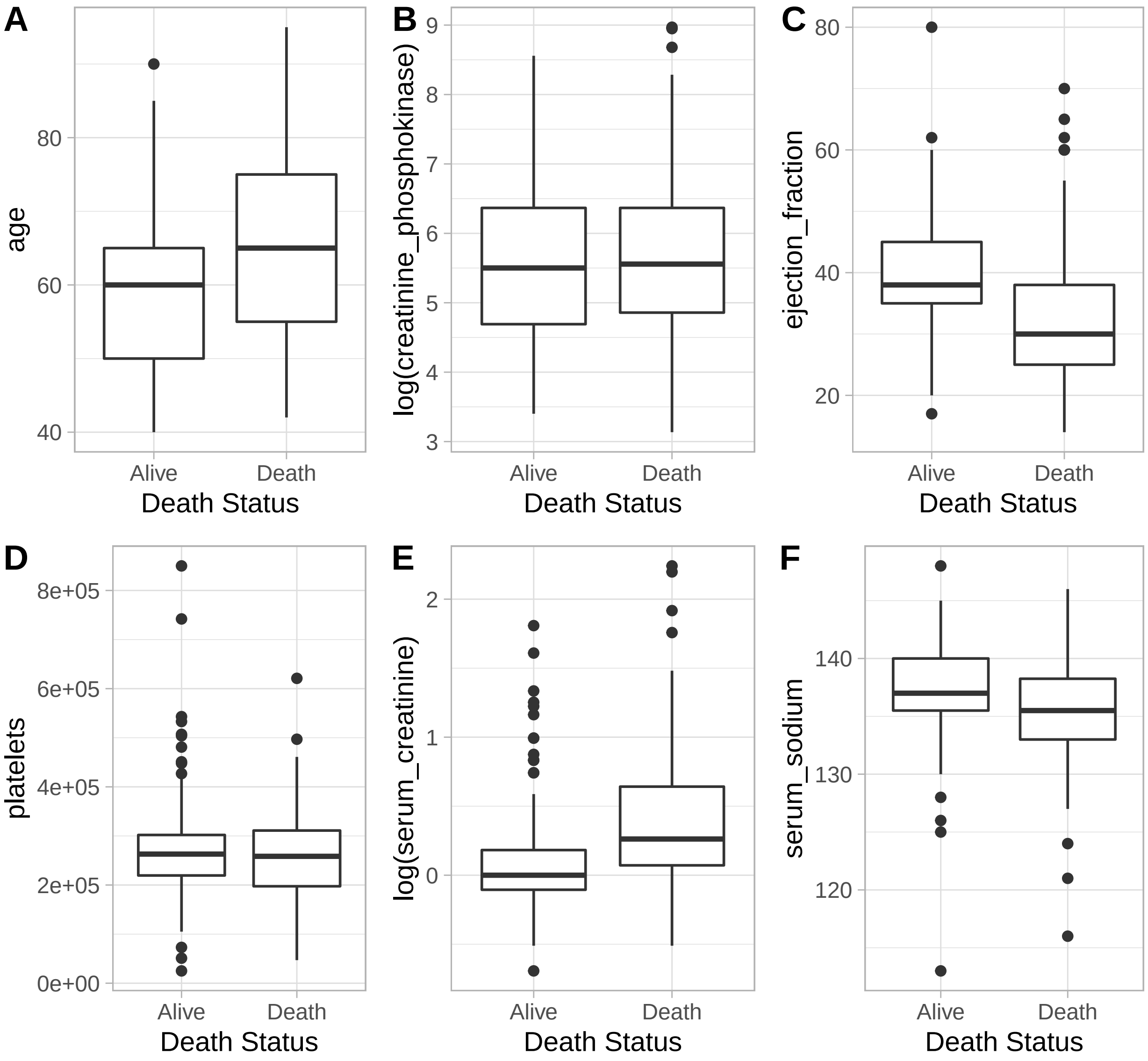
```r
# visualize the result
# creatinine_phosphokinase
log_cp <- heart_data %>%
  ggplot(aes(x = log(creatinine_phosphokinase))) + geom_histogram(bins=15) +
  labs(x = "Log Creatinine Phosphokinase") + theme_light()

# serum_creatinine
log_sc <- heart_data %>%
  ggplot(aes(x = log(serum_creatinine))) + geom_histogram(bins=15) +
  labs(x = "Log Serum Creatinine") + theme_light()

# plot
(p_log <- plot_grid(creatinine_phosphokinase, serum_creatinine,
                    log_cp, log_sc, ncol=2,
                    labels = c("B", "E", "Log(B)", "Log(E)")))
```



- Log-transform decreases skewness in the distributions.

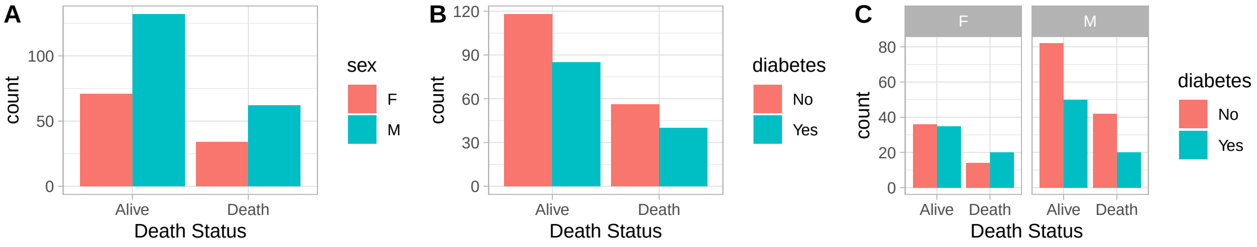
2.3 2D plot

```r
# Age and Response
Box1 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, y=age)) + geom_boxplot() + labs(x = "Death Status") +
  theme_light()

# log(creatinine_phosphokinase) and Response
Box2 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, y=log(creatinine_phosphokinase))) + geom_boxplot() +
  labs(x = "Death Status") + theme_light()

# Ejection fraction and Response
Box3 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, y=ejection_fraction)) + geom_boxplot() +
  labs(x = "Death Status") + theme_light()

# platelets and Response
Box4 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, y=platelets)) + geom_boxplot() +
  labs(x = "Death Status") + theme_light()

# log(serum_creatinine) and Response
Box5 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, y=log(serum_creatinine))) + geom_boxplot() +
  labs(x = "Death Status") + theme_light()

# serum_sodium and Response
Box6 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, y=serum_sodium)) + geom_boxplot() +
  labs(x = "Death Status") + theme_light()

plot_grid(Box1, Box2, Box3, Box4, Box5, Box6, ncol=3, labels = "AUTO")
```



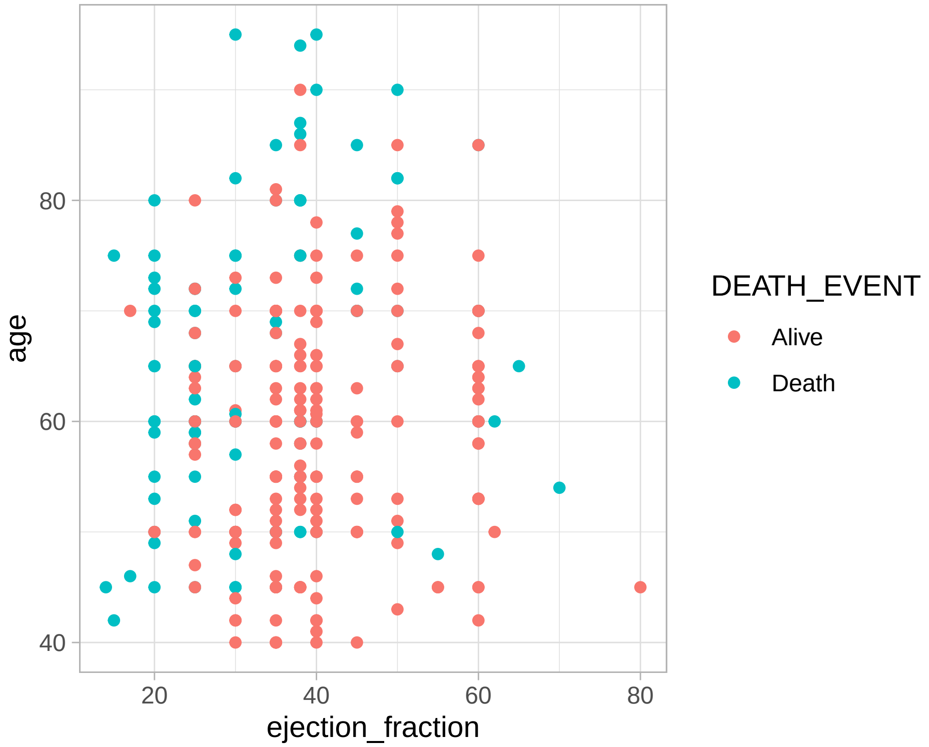
- The median age of deaths is roughly 5 years older than survivors (65 vs 60 yrs)
- It seems lower ejection fraction seems to be associated with greater chance of heart failure according to our sample data.

```
# Sex and Response
Bar1 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, fill=sex)) + geom_bar(position="dodge") +
  labs(x = "Death Status") + theme_light()

# Diabetes and Response
Bar2 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, fill=diabetes)) + geom_bar(position="dodge") +
  labs(x = "Death Status") + theme_light()

# Diabetes, Sex and Response
Bar3 <- heart_data %>%
  ggplot(aes(x=DEATH_EVENT, fill=diabetes)) + geom_bar(position="dodge") +
  labs(x = "Death Status") + theme_light() + facet_wrap(~sex)

plot_grid(Bar1, Bar2, Bar3, ncol = 3, labels = "AUTO")
```



- Roughly the same proportion of males and females die of heart failure.
- Roughly the same proportion of heart failure for people with and without diabetes.
- For males, the death ratio is similar bewteen people with and without diabetes.
- For females, the death ratio is higher in people with diabetes.

```
# Ejection fraction and Age for different Response
heart_data %>%
  ggplot(aes(x=ejection_fraction, y=age, color=DEATH_EVENT)) + geom_point() +
  theme_light()
```



- No clear trends here, though it seems there is a greater concentration of deaths at lower ejection fraction levels at all ages.

3. Data Modelling

3.1 Data Spliting

```
# Let's first split the data into training and test data (70/30)
set.seed(414)
n = nrow(heart_data)
idx_tr <- sample(n, round(0.7*n), replace=FALSE)

# Define training and test data
train = heart_data[idx_tr,]
test = heart_data[-idx_tr,]

dim(train)
```

```
## [1] 209  12
```

```
dim(test)
```

```
## [1] 90 12
```

3.2 Model1: Logistic Regression

Using all the 11 features

```
model1_LR <- glm(DEATH_EVENT ~ ., family=binomial, data=train)

# GOF (Hosmer-Lemeshow) -> p-value > 0.05, no lack of fit
suppressMessages(library(ResourceSelection))
hoslem.test(model1_LR$y,fitted(model1_LR),g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model1_LR$y, fitted(model1_LR)
## X-squared = 15.385, df = 8, p-value = 0.05207
```

Feature selection using AIC

```
# Feature selection using AIC: left with 4 features
model1_LR_small <- step(model1_LR, trace=0)
library(faraway)
```

```
##
## Attaching package: 'faraway'

## The following object is masked _by_ '.GlobalEnv':
##
##     diabetes
```

```r
sumary(model1_LR_small)
```

```
##                         Estimate Std. Error z value  Pr(>|z|)
## (Intercept)            -2.458993   1.079707 -2.2775 0.0227586
## age                     0.042042   0.015002  2.8023 0.0050735
## ejection_fraction      -0.065281   0.017665 -3.6956 0.0002194
## high_blood_pressureYes  0.704623   0.349857  2.0140 0.0440059
## serum_creatinine        0.842412   0.246316  3.4200 0.0006261
##
## n = 209 p = 5
## Deviance = 212.38153 Null Deviance = 262.21202 (Difference = 49.83049)
```

```r
# GOF: p-value > 0.05, no lack of fit
hoslem.test(model1_LR_small$y,fitted(model1_LR_small),g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model1_LR_small$y, fitted(model1_LR_small)
## X-squared = 7.6892, df = 8, p-value = 0.4644
```

- **ejection_fraction** is the most significant predictor.

Add quadratic term for ejection_fraction

```r
model1_LR_small_quadra <- glm(DEATH_EVENT ~ age + poly(ejection_fraction, 2)
                              + high_blood_pressure + serum_creatinine,
                              family=binomial, data=train)
sumary(model1_LR_small_quadra)
```

```
##                              Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                 -5.091122   1.047635 -4.8596 1.176e-06
## age                          0.045397   0.015598  2.9103 0.0036105
## poly(ejection_fraction, 2)1 -8.668854   2.630956 -3.2949 0.0009844
## poly(ejection_fraction, 2)2  6.522273   2.726830  2.3919 0.0167620
## high_blood_pressureYes       0.653746   0.357416  1.8291 0.0673863
## serum_creatinine             0.853430   0.257579  3.3133 0.0009221
##
## n = 209 p = 6
## Deviance = 206.78081 Null Deviance = 262.21202 (Difference = 55.43121)
```

```r
# GOF (Hosmer-Lemeshow) -> p-value > 0.05, no lack of fit
hoslem.test(model1_LR_small_quadra$y,fitted(model1_LR_small_quadra),g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model1_LR_small_quadra$y, fitted(model1_LR_small_quadra)
## X-squared = 9.3783, df = 8, p-value = 0.3114
```

Test the significance of quadratic term

$$H_0: \text{Smaller model selected by AIC is adequate} \;\; vs. \;\; H_a: \text{Larger model with quadratic term is adequate}$$

```
# Compare with backward selection model
anova(model1_LR_small, model1_LR_small_quadra, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: DEATH_EVENT ~ age + ejection_fraction + high_blood_pressure +
##     serum_creatinine
## Model 2: DEATH_EVENT ~ age + poly(ejection_fraction, 2) + high_blood_pressure +
##     serum_creatinine
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       204     212.38
## 2       203     206.78  1   5.6007  0.01795 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- LRT: Deviance $= 5.6007$, follow χ^2_1
- The p-value is 0.01795, which is smaller than 0.05, we have evidence to reject the null hypothesis. Thus, we prefer the larger model with quadratic term.

Drop high_blood_pressure since it is not significant

```
model1_LR_small_quadra_drop <- glm(DEATH_EVENT ~ age + poly(ejection_fraction, 2)
                                   + serum_creatinine, family=binomial, data=train)
sumary(model1_LR_small_quadra_drop)
```

```
##                              Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                 -4.857947   1.012128 -4.7997 1.589e-06
## age                          0.047169   0.015300  3.0829  0.002050
## poly(ejection_fraction, 2)1 -8.244672   2.590283 -3.1829  0.001458
## poly(ejection_fraction, 2)2  6.810303   2.689282  2.5324  0.011329
## serum_creatinine             0.790146   0.249656  3.1649  0.001551
##
## n = 209 p = 5
## Deviance = 210.14863 Null Deviance = 262.21202 (Difference = 52.06339)
```

```
# GOF (Hosmer-Lemeshow) -> p-value > 0.05, no lack of fit
hoslem.test(model1_LR_small_quadra_drop$y,fitted(model1_LR_small_quadra_drop),g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model1_LR_small_quadra_drop$y, fitted(model1_LR_small_quadra_drop)
## X-squared = 7.3221, df = 8, p-value = 0.5023
```

Compare model with and without drop high_blood_pressure

$$H_0 : \text{Smaller model without high blood pressure is adequate}$$
$$H_a : \text{larger model with high blood pressure is adequate}$$

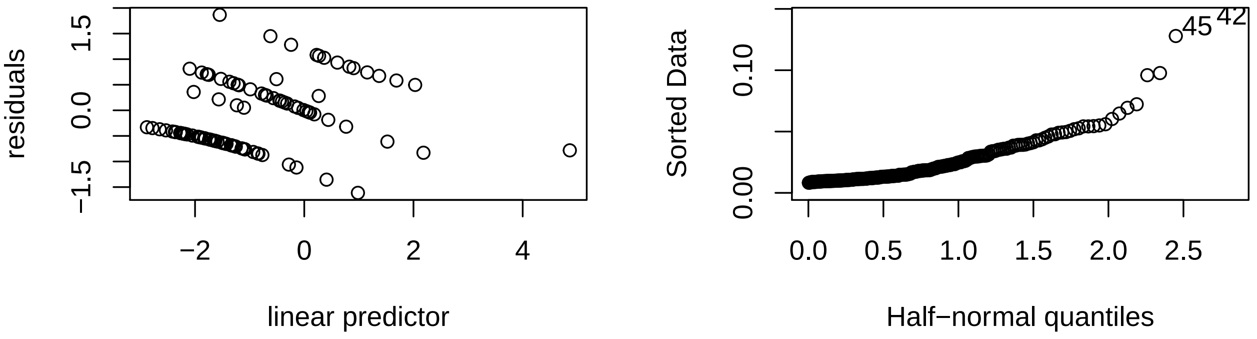
```r
# Compare with backward selection model
anova(model1_LR_small_quadra_drop, model1_LR_small_quadra, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: DEATH_EVENT ~ age + poly(ejection_fraction, 2) + serum_creatinine
## Model 2: DEATH_EVENT ~ age + poly(ejection_fraction, 2) + high_blood_pressure +
##     serum_creatinine
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       204     210.15
## 2       203     206.78  1   3.3678  0.06648 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- LRT: Deviance = 3.3678, follow χ^2_1
- The p-value is 0.06648, which is larger than 0.05, we fail to reject the null hypothesis. Thus, we prefer the smaller model without high_blood_pressure.

Model Diagnostics

```r
# Deviance Residuals plot
par(mfrow = c(1, 2))
train_assumption <- mutate(train, residuals=residuals(model1_LR_small_quadra_drop),
                           linpred=predict(model1_LR_small_quadra_drop))
gdf <- group_by(train_assumption,
                cut(linpred, breaks=c(min(linpred),
                                      unique(quantile(linpred,(1:100)/101)),max(linpred)),
                    include.lowest = TRUE))
diagdf <- summarise(gdf, residuals=mean(residuals), linpred=mean(linpred), .groups = 'drop')

plot(residuals ~ linpred, diagdf, xlab="linear predictor",cex.lab=1)

# half-normal plot
suppressMessages(library(faraway))
halfnorm(hatvalues(model1_LR_small_quadra_drop))
```



3.3 Model2: LDA

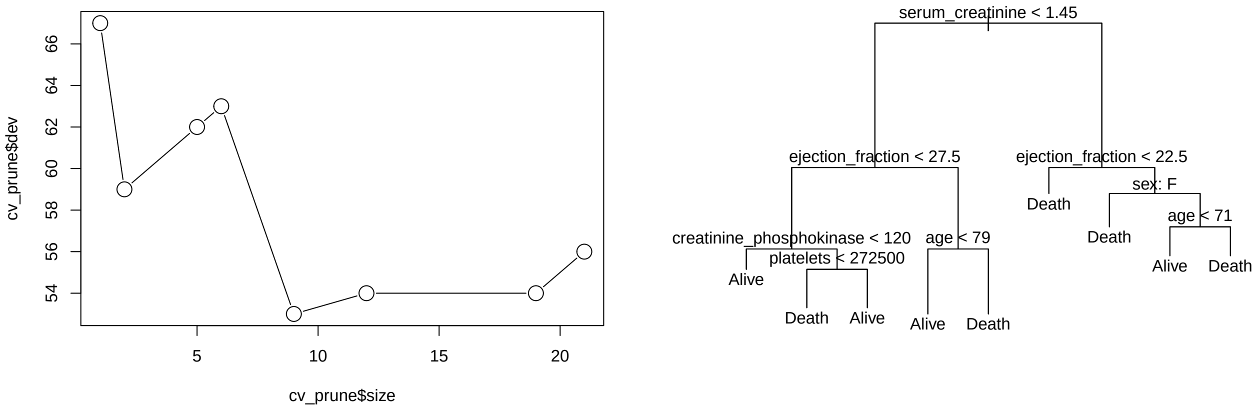
```
suppressMessages(library(MASS))
model2_LDA = lda(DEATH_EVENT ~ age + poly(ejection_fraction, 2) +
                   log(serum_creatinine), data=train)
```

3.4 Model3: QDA

```
model3_QDA = qda(DEATH_EVENT ~ age + poly(ejection_fraction, 2) +
                   log(serum_creatinine), data=train)
```

3.5 Model4: Classification Tree

```
suppressMessages(library(tree))
set.seed(108)
model4_tree <- tree(DEATH_EVENT ~ ., data=train)

# pruned tree -> 9 terminal nodes
par(mfrow = c(1, 2))
cv_prune <- cv.tree(model4_tree, FUN=prune.misclass)
plot(cv_prune$size, cv_prune$dev, type='b', cex=2)
model4_tree_prune <- prune.misclass(model4_tree, best=9)
plot(model4_tree_prune); text(model4_tree_prune, pretty=0)
```



3.6 Model5: Bagging

```
suppressMessages(library(randomForest))
set.seed(108)
model5_bagging <- randomForest(DEATH_EVENT ~ ., data=train, ntree=200,
                        mtry=11, importance=TRUE)
model5_bagging
```

```
##
## Call:
##  randomForest(formula = DEATH_EVENT ~ ., data = train, ntree = 200,      mtry = 11, importance = TRUE
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 11
##
##         OOB estimate of  error rate: 28.71%
## Confusion matrix:
##       Alive Death class.error
## Alive   118    24   0.1690141
## Death    36    31   0.5373134
```

3.7 Model6: Random Forest

```
set.seed(108)
model6_rf <- randomForest(DEATH_EVENT ~ ., data=train, ntree=200,
                        mtry=4, importance=TRUE)
model6_rf
```

```
##
## Call:
##  randomForest(formula = DEATH_EVENT ~ ., data = train, ntree = 200,      mtry = 4, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 28.23%
## Confusion matrix:
##       Alive Death class.error
## Alive   119    23   0.1619718
## Death    36    31   0.5373134
```

3.8 Model7: Boosting

```
suppressMessages(library(gbm))
set.seed(108)

# gbm need character type response instead of factor
train_boost <- train
train_boost$DEATH_EVENT <- as.character(train_boost$DEATH_EVENT)
train_boost <- train_boost %>%
  mutate(DEATH_EVENT = case_when(DEATH_EVENT=="Alive" ~ "0",
                                DEATH_EVENT=="Death" ~ "1"))
test_boost <- test
test_boost$DEATH_EVENT <- as.character(test_boost$DEATH_EVENT)
test_boost <- test_boost %>%
  mutate(DEATH_EVENT = case_when(DEATH_EVENT=="Alive" ~ "0",
                                DEATH_EVENT=="Death" ~ "1"))
```

```
# build the model
model7_boosting <- gbm(DEATH_EVENT ~ ., data=train_boost, distribution="bernoulli",
                n.trees=200, shrinkage=0.1)
model7_boosting
```

```
## gbm(formula = DEATH_EVENT ~ ., distribution = "bernoulli", data = train_boost,
##     n.trees = 200, shrinkage = 0.1)
## A gradient boosted model with bernoulli loss function.
## 200 iterations were performed.
## There were 11 predictors of which 11 had non-zero influence.
```

4. Model Selection using ROC Analysis

```r
suppressMessages(library(pROC))

# Create matrix to store the evaluation metrics for each model
eva_metrics = matrix(0, nrow=7, ncol=5)

# phat
phat1 <- predict(model1_LR_small_quadra_drop, newdata=test, type="response")
phat2 <- predict(model2_LDA, newdata=test)$posterior[,2]
phat3 <- predict(model3_QDA, newdata=test)$posterior[,2]
phat4 <- predict(model4_tree_prune, newdata=test)[,2]
phat5 <- predict(model5_bagging, newdata=test, type="prob")[,2]
phat6 <- predict(model6_rf, newdata=test, type="prob")[,2]
phat7 <- predict(model7_boosting, newdata=test_boost, type="response")

# create roc object
roc_obj1 <- roc(response=test$DEATH_EVENT, predictor=phat1)
roc_obj2 <- roc(response=test$DEATH_EVENT, predictor=phat2)
roc_obj3 <- roc(response=test$DEATH_EVENT, predictor=phat3)
roc_obj4 <- roc(response=test$DEATH_EVENT, predictor=phat4)
roc_obj5 <- roc(response=test$DEATH_EVENT, predictor=phat5)
roc_obj6 <- roc(response=test$DEATH_EVENT, predictor=phat6)
roc_obj7 <- roc(response=test$DEATH_EVENT, predictor=phat7)

# calculate AUC
AUC1 <- auc(roc_obj1)
AUC2 <- auc(roc_obj2)
AUC3 <- auc(roc_obj3)
AUC4 <- auc(roc_obj4)
AUC5 <- auc(roc_obj5)
AUC6 <- auc(roc_obj6)
AUC7 <- auc(roc_obj7)

# show the performance matric
roc_1 <- c(coords(roc_obj1, "b", ret=c("threshold","se","sp","accuracy"),
                         best.method="youden", transpose=TRUE), AUC1)
eva_metrics[1,] <- t(roc_1)

roc_2 <- c(coords(roc_obj2, "b", ret=c("threshold","se","sp","accuracy"),
                         best.method="youden", transpose=TRUE), AUC2)
eva_metrics[2,] <- t(roc_2)

roc_3 <- c(coords(roc_obj3, "b", ret=c("threshold","se","sp","accuracy"),
                         best.method="youden", transpose=TRUE), AUC3)
eva_metrics[3,] <- t(roc_3)

roc_4 <- c(coords(roc_obj4, "b", ret=c("threshold","se","sp","accuracy"),
                         best.method="youden", transpose=TRUE), AUC4)
eva_metrics[4,] <- t(roc_4)

roc_5 <- c(coords(roc_obj5, "b", ret=c("threshold","se","sp","accuracy"),
                         best.method="youden", transpose=TRUE), AUC5)
```

```
eva_metrics[5,] <- t(roc_5)

roc_6 <- c(coords(roc_obj6, "b", ret=c("threshold","se","sp","accuracy"),
                         best.method="youden", transpose=TRUE), AUC6)
eva_metrics[6,] <- t(roc_6)

roc_7 <- c(coords(roc_obj7, "b", ret=c("threshold","se","sp","accuracy"),
                         best.method="youden", transpose=TRUE), AUC7)
eva_metrics[7,] <- t(roc_7)

# Create metrics df
metrics <- as.data.frame(eva_metrics)
colnames(metrics) = c("Threshold","Sensitivity","Specificity","Accuracy","AUC")
rownames(metrics) = c("Logistic Regression","LDA","QDA","Tree","Bagging","RF","Boosting")
metrics
```

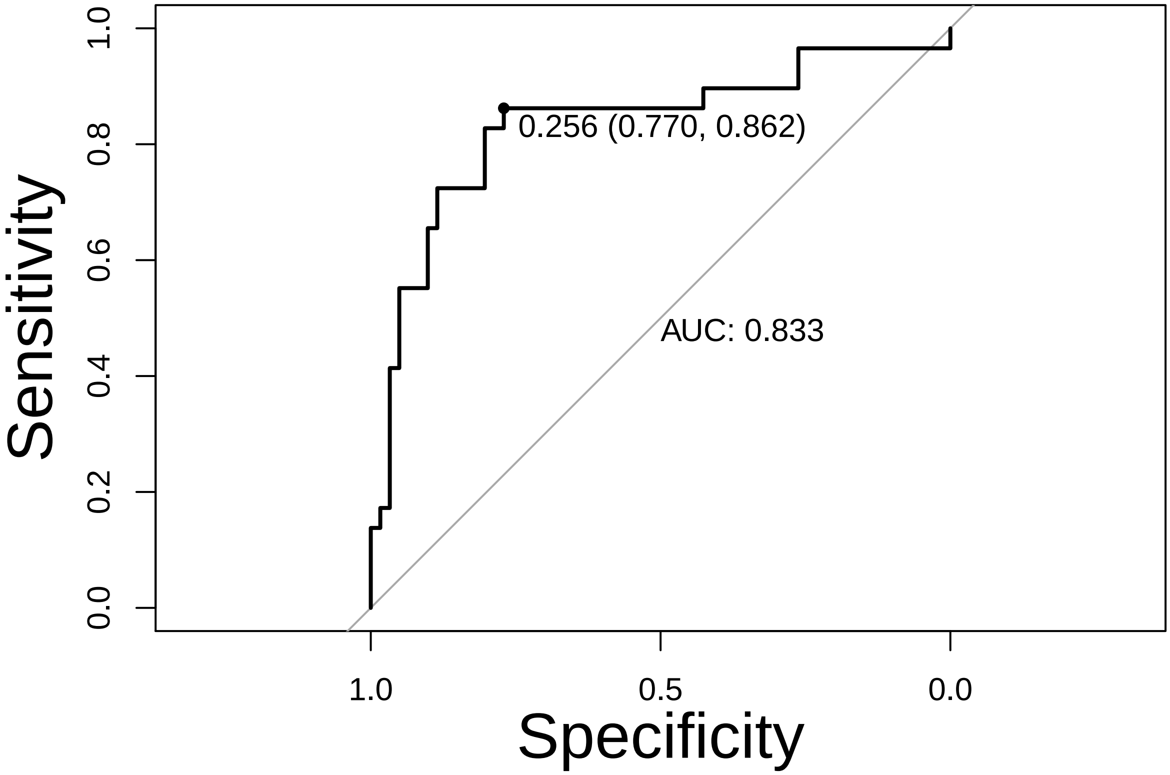
```
##                     Threshold Sensitivity Specificity  Accuracy       AUC
## Logistic Regression 0.2563136   0.8620690   0.7704918 0.8000000 0.8332391
## LDA                 0.2384947   0.7931034   0.7868852 0.7888889 0.8128886
## QDA                 0.3358923   0.7241379   0.8524590 0.8111111 0.7908423
## Tree                0.1370523   0.7931034   0.8032787 0.8000000 0.8114754
## Bagging             0.1725000   0.9310345   0.6393443 0.7333333 0.8174110
## RF                  0.2375000   0.8965517   0.6721311 0.7444444 0.8383267
## Boosting            0.3113842   0.7586207   0.8360656 0.8111111 0.7970605
```

- The best model based on the highest test AUC is Random Forest, the AUC = 0.8383267.
- But Logistic Regression performs (0.8332391) very close to Random Forest, we decide to select Logistic Regression as our final model.

5. Analyze the Best Performing Model - Logistic Regression

5.1 Logistic Regression ROC Analysis

```
# produce ROC Curve
plot(roc_obj1,legacy.axes=FALSE,print.auc=TRUE,print.thres=TRUE,cex.lab=2)
```



5.2 Logistic Regression Confusion Matrix using Best Threshold

```
# Obtain Y_hat values for the data observation (cutoff=0.2563136)
proba_hat <- predict(model1_LR_small_quadra_drop, newdata=test, type="response")

n = nrow(test); y_hat = rep(0,n)
cutoff = 0.2563136; idx = which(proba_hat > cutoff)
y_hat[idx] = 1

# confusion matrix at cutoff=0.2563136
(conf_mat = table(predicted = y_hat, actual = test$DEATH_EVENT))
```

```
##          actual
## predicted Alive Death
##         0    47     4
##         1    14    25
```

```r
# sensitivity/recall
conf_mat[2, 2] / sum(conf_mat[, 2])
```

```
## [1] 0.862069
```

```r
# precision/positive predictive value
conf_mat[2, 2] / sum(conf_mat[2, ])
```

```
## [1] 0.6410256
```

```r
# specificity
conf_mat[1, 1] / sum(conf_mat[, 1])
```

```
## [1] 0.7704918
```

	Summary Metrics for Logistic Regression
Sensitivity/Recall	0.8620690
Precision/Positive Predictive Value	0.6410256
Specificity	0.7704918
Accuracy	0.8000000
AUC	0.8332391

5.3 Logistic Regression Coefficient Interpretation

```r
sumary(model1_LR_small_quadra_drop)
```

```
##                              Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                 -4.857947   1.012128 -4.7997 1.589e-06
## age                          0.047169   0.015300  3.0829  0.002050
## poly(ejection_fraction, 2)1 -8.244672   2.590283 -3.1829  0.001458
## poly(ejection_fraction, 2)2  6.810303   2.689282  2.5324  0.011329
## serum_creatinine             0.790146   0.249656  3.1649  0.001551
##
## n = 209 p = 5
## Deviance = 210.14863 Null Deviance = 262.21202 (Difference = 52.06339)
```

```r
exp(coefficients(model1_LR_small_quadra_drop))
```

```
##                 (Intercept)                         age
##                7.766413e-03                1.048299e+00
## poly(ejection_fraction, 2)1 poly(ejection_fraction, 2)2
##                2.626543e-04                9.071458e+02
##            serum_creatinine
##                2.203718e+00
```