Module 2: Supervised Learning

31/05/2022

```
# If a package is installed, it will be loaded. If any
## are not, the missing package(s) will be installed
## from CRAN and then loaded.

## First specify the packages of interest
packages <- c(
  "dplyr", "PheCAP", "glmnet", "randomForestSRC", "PheNorm",
  "MAP", "pROC", "mltools", "data.table", "ggplot2", "parallel"
)

## Now load or install&load all
package.check <- lapply(
  packages,
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE)
      library(x, character.only = TRUE)
    }
  }
)

# load environment from example 1
load("environment.RData")
```

Prepare data for algorithm development

Split data into training and testing set.

```
data <- PhecapData(PheCAP::ehr_data, "healthcare_utilization", "label", 75,
  patient_id = "patient_id", seed = 123
)

# Data with non-missing labels
labeled_data <- ehr_data %>% dplyr::filter(!is.na(label))


# All Features
all_x <- ehr_data %>% dplyr::select(
  starts_with("COD"), starts_with("NLP"),
  starts_with("main"), healthcare_utilization
)
health_count <- ehr_data$healthcare_utilization
```

```
# Training Set
train_data <- ehr_data %>% dplyr::filter(patient_id %in% data$training_set)
train_x <- train_data %>%
  dplyr::select(
    starts_with("COD"), starts_with("NLP"),
    starts_with("main"), healthcare_utilization
  ) %>%
  as.matrix()
train_y <- train_data %>%
  dplyr::select(label) %>%
  pull()

# Testing Set
test_data <- ehr_data %>% dplyr::filter(patient_id %in% data$validation_set)
test_x <- test_data %>%
  dplyr::select(
    starts_with("COD"), starts_with("NLP"),
    starts_with("main"), healthcare_utilization
  ) %>%
  as.matrix()
test_y <- test_data %>%
  dplyr::select(label) %>%
  pull()
```

Penalized logistic regression

- Fit LASSO and Adaptive LASSO(ALASSO)

```
# Choose best lambda using CV
beta.lasso <- lasso_fit(
  x = train_x, y = train_y,
  tuning = "cv", family = "binomial"
)
```

```
# Features Selected
names(beta.lasso[abs(beta.lasso) > 0])[-1]
```

```
## [1] "NLP93"                    "NLP104"                   "NLP304"
## [4] "main_NLP"                 "main_ICDNLP"              "healthcare_utilization"
```

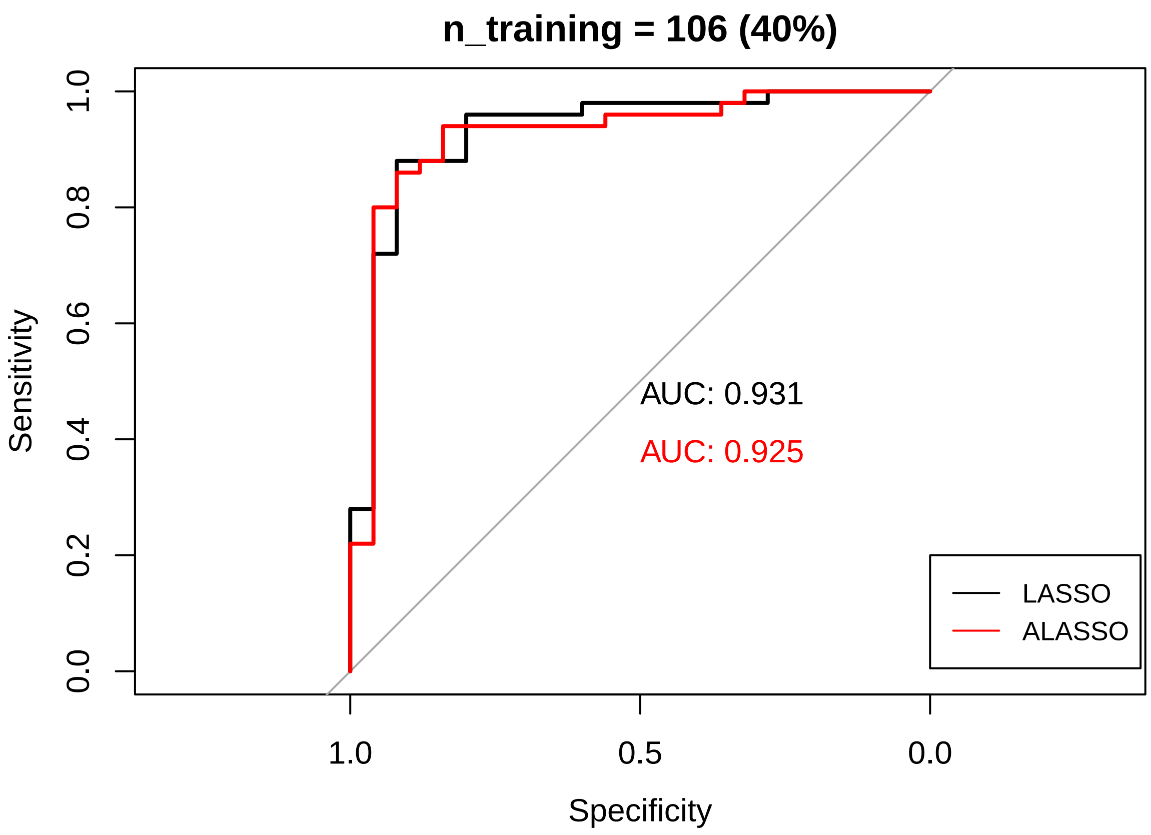
```
# prediction on testing set
y_hat.lasso <- linear_model_predict(
  beta = beta.lasso, x = test_x,
  probability = TRUE
)
```

```
# Fit Adaptive LASSO
beta.alasso <- adaptive_lasso_fit(
  x = train_x, y = train_y,
  tuning = "cv", family = "binomial"
```

```r
)
y_hat.alasso <- linear_model_predict(
  beta = beta.alasso, x = test_x,
  probability = TRUE
)


# Features Selected
names(beta.alasso[abs(beta.alasso) > 0])[-1]
```

```
## [1] "NLP304"                "main_NLP"                "healthcare_utilization"
```

```r
roc.lasso <- roc(test_y, y_hat.lasso)
roc.alasso <- roc(test_y, y_hat.alasso)

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 106 (40%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = "red", add = TRUE, print.auc.y = 0.4
)
legend(0, 0.2,
  legend = c("LASSO", "ALASSO"), col = c("black", "red"),
  lty = 1, cex = 0.8
)
```



- ROC parameter at FPR = 5% and 10% cut-off

```
roc_full.lasso <- get_roc(y_true = test_y, y_score = y_hat.lasso) %>% data.frame()
get_roc_parameter(0.05, roc_full.lasso)
```

```
##      cutoff  pos.rate  FPR  TPR       PPV       NPV        F1
## 1 0.8781599 0.2000000 0.04 0.39 0.9512195 0.4403670 0.5531915
## 2 0.8749928 0.2066667 0.04 0.50 0.9615385 0.4897959 0.6578947
## 3 0.8130579 0.4133333 0.04 0.61 0.9682540 0.5517241 0.7484663
```

```
roc_full.lasso <- get_roc(y_true = test_y, y_score = y_hat.lasso) %>% data.frame()
get_roc_parameter(0.1, roc_full.lasso)
```

```
##      cutoff pos.rate FPR  TPR       PPV       NPV        F1
## 1 0.6431842     0.62 0.1 0.88 0.9462366 0.7894737 0.9119171
```

```
roc_full.alasso <- get_roc(y_true = test_y, y_score = y_hat.alasso) %>% data.frame()
get_roc_parameter(0.05, roc_full.alasso)
```

```
##      cutoff  pos.rate  FPR   TPR       PPV       NPV        F1
## 1 0.9631509 0.1600000 0.04 0.365 0.9480519 0.4304933 0.5270758
## 2 0.9631337 0.1666667 0.04 0.510 0.9622642 0.4948454 0.6666667
## 3 0.8722516 0.4800000 0.04 0.655 0.9703704 0.5818182 0.7820896
```

```
roc_full.alasso <- get_roc(y_true = test_y, y_score = y_hat.alasso) %>% data.frame()
get_roc_parameter(0.1, roc_full.alasso)
```

```
##      cutoff  pos.rate FPR  TPR       PPV       NPV        F1
## 1 0.7324685 0.6066667 0.1 0.86 0.9450549 0.7627119 0.9005236
```

Different train size

- randomly sample training size = 50, 70, 90
- rest as testing set
- repeat 600 times

```
start <- Sys.time()
auc_supervised <- validate_supervised(
  dat = labeled_data, nsim = 600,
  n.train = c(50, 70, 90)
)
end <- Sys.time()

end - start
```

```
## Time difference of 4.812445 mins
```

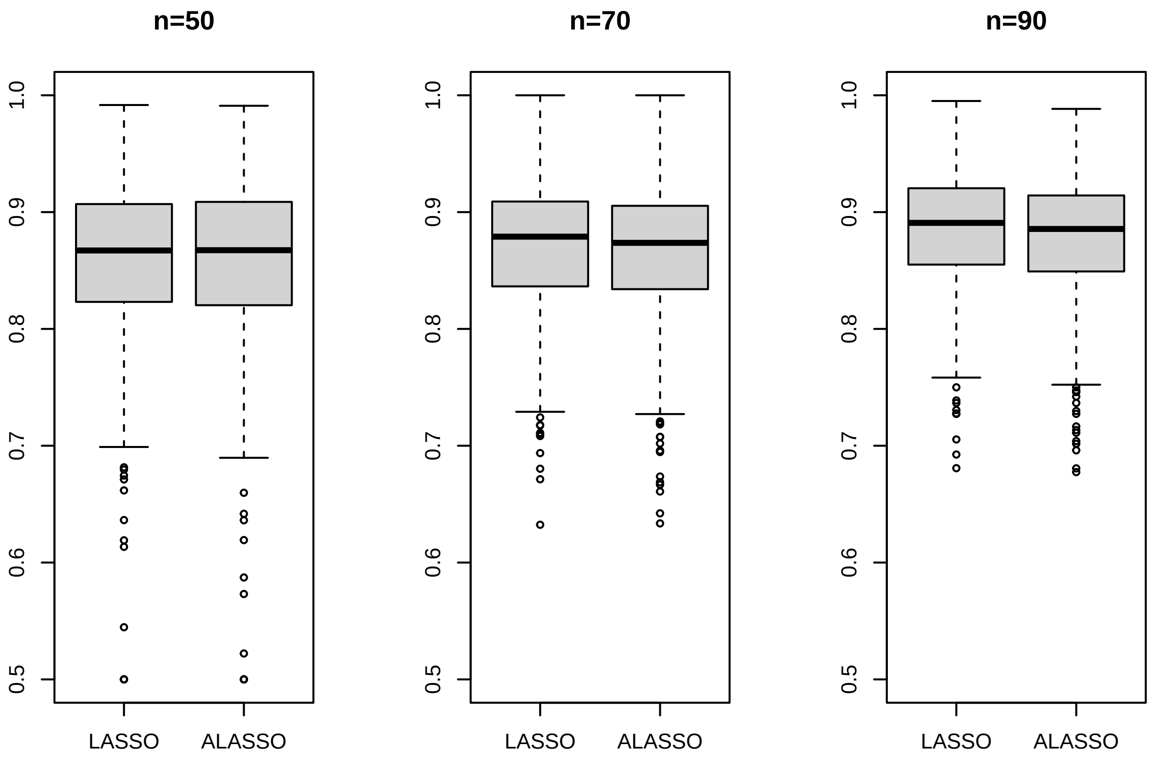
```r
# median AUC
apply(auc_supervised, 2, median)
```

```
##  n=50,LASSO  n=70,LASSO  n=90,LASSO n=50,ALASSO n=70,ALASSO n=90,ALASSO
##   0.8671329   0.8789683   0.8907670   0.8673935   0.8736602   0.8855655
```

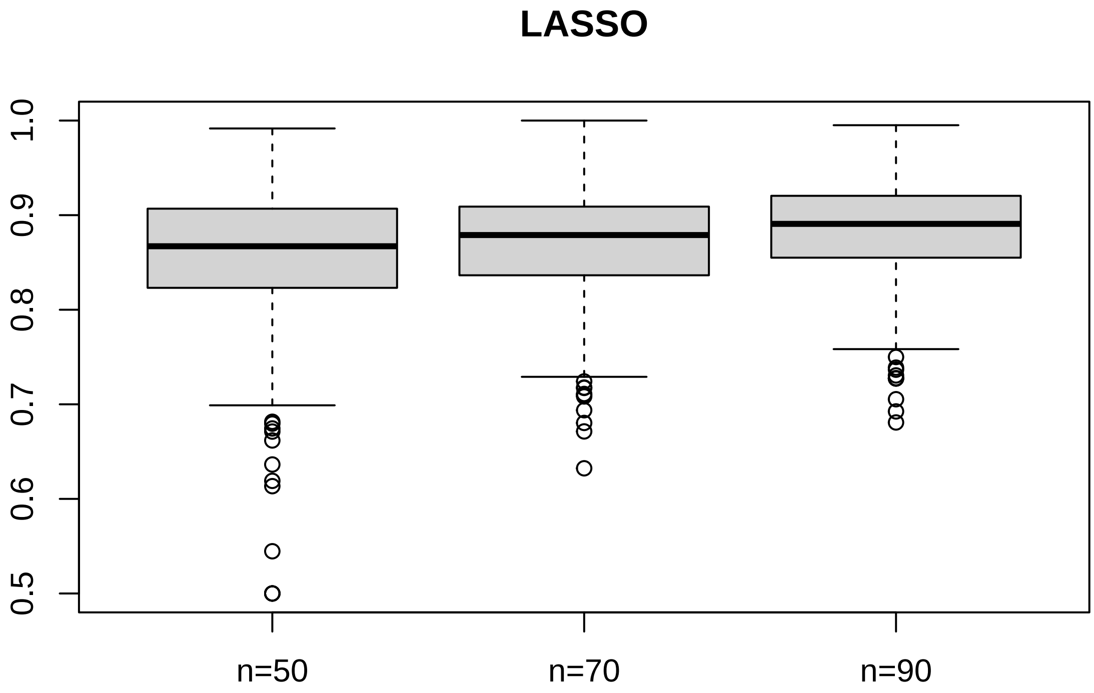
```r
# SE
apply(auc_supervised, 2, sd)
```

```
##  n=50,LASSO  n=70,LASSO  n=90,LASSO n=50,ALASSO n=70,ALASSO n=90,ALASSO
##  0.07198110  0.05590032  0.05184181  0.07300070  0.05871888  0.05415953
```

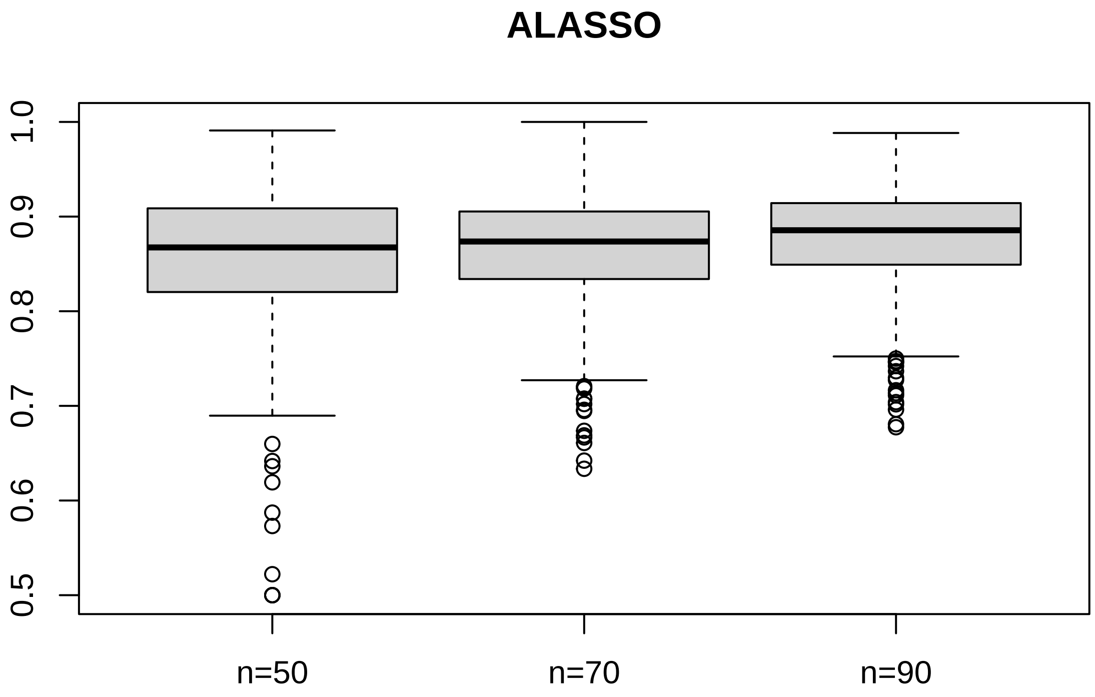
```r
par(mfrow = c(1, 3))
boxplot(auc_supervised %>% select(starts_with("n=50")),
  ylim = c(0.5, 1),
  names = c("LASSO", "ALASSO"), main = "n=50"
)
boxplot(auc_supervised %>% select(starts_with("n=70")),
  ylim = c(0.5, 1),
  names = c("LASSO", "ALASSO"), main = "n=70"
)
boxplot(auc_supervised %>% select(starts_with("n=90")),
  ylim = c(0.5, 1),
  names = c("LASSO", "ALASSO"), main = "n=90"
)
```



```
boxplot(auc_supervised[, 1:3],
  ylim = c(0.5, 1),
  names = c("n=50", "n=70", "n=90"), main = "LASSO"
)
```



```
boxplot(auc_supervised[, 4:6],
  ylim = c(0.5, 1),
  names = c("n=50", "n=70", "n=90"), main = "ALASSO"
)
```

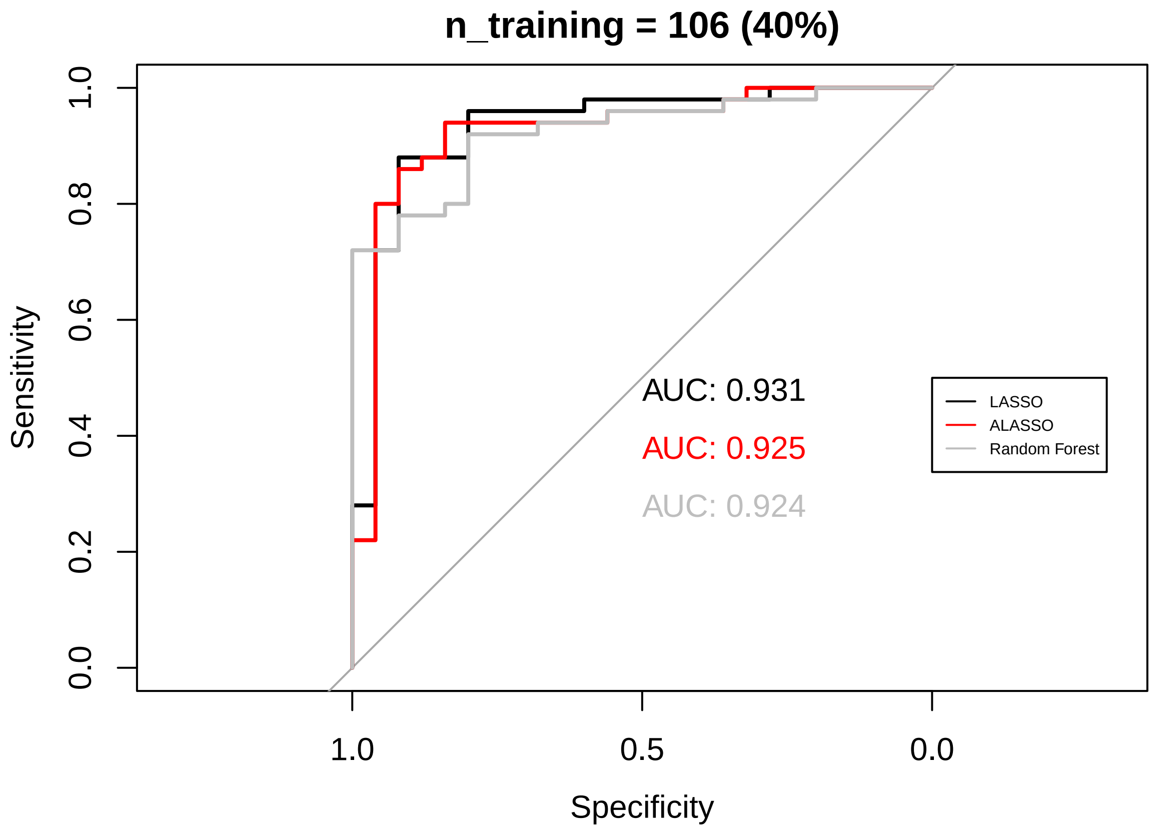


Appendix

Random Forest

```
model_rf <- rfsrc(y ~ ., data = data.frame(y = train_y, x = train_x))
y_hat.rf <- predict(model_rf, newdata = data.frame(x = test_x))$predicted
```

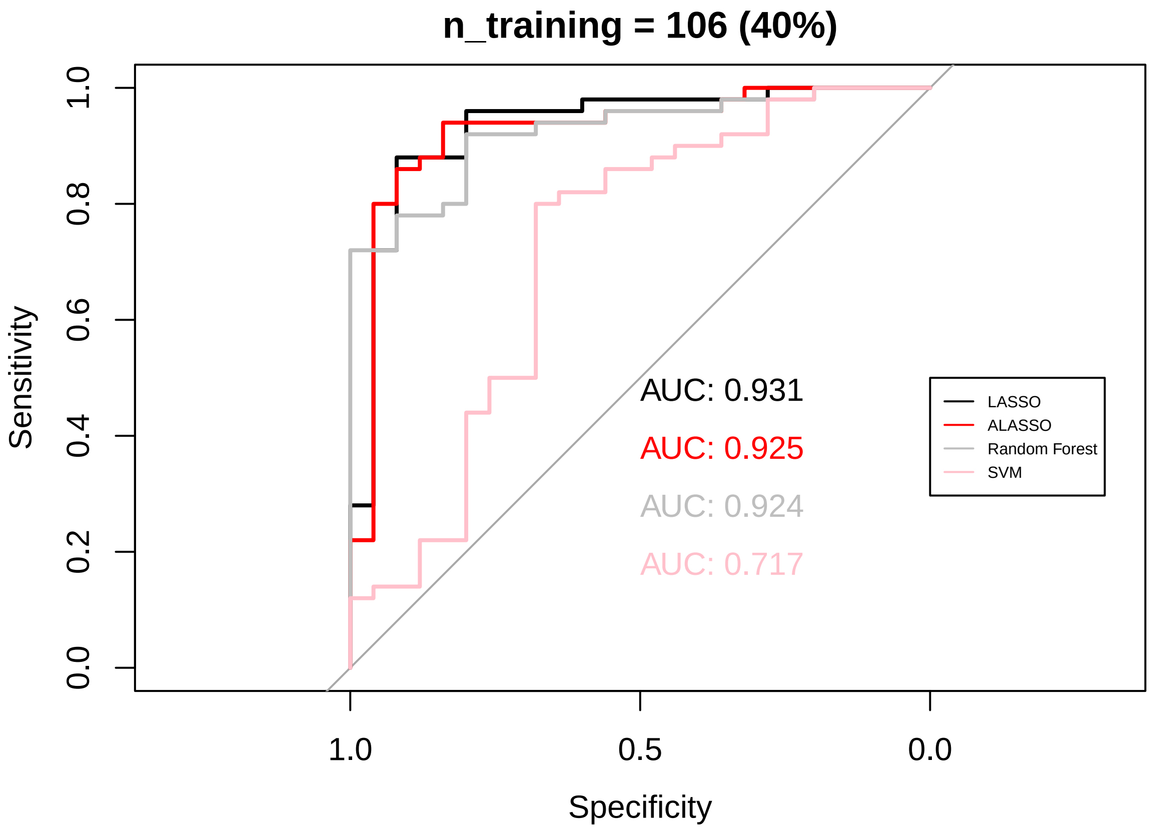
```
roc.rf <- roc(test_y, y_hat.rf)

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 106 (40%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = "red", add = TRUE, print.auc.y = 0.4
)
plot(roc.rf,
  print.auc = TRUE, col = "grey", add = TRUE, print.auc.y = 0.3
)
legend(0, 0.5,
  legend = c("LASSO", "ALASSO", "Random Forest"), col = c("black", "red", "grey"),
  lty = 1, cex = 0.5
)
```



SVM

```
model_svm <- SVMMaj::svmmaj(X = train_x, y = train_y)
y_hat.svm <- predict(model_svm, test_x)
```

```
roc.svm <- roc(test_y, y_hat.svm)

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 106 (40%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = "red", add = TRUE, print.auc.y = 0.4
)
plot(roc.rf,
  print.auc = TRUE, col = "grey", add = TRUE, print.auc.y = 0.3
)
plot(roc.svm,
  print.auc = TRUE, col = "pink", add = TRUE, print.auc.y = 0.2
)
legend(0, 0.5,
  legend = c("LASSO", "ALASSO", "Random Forest", "SVM"),
  col = c("black", "red", "grey", "pink"),
  lty = 1, cex = 0.5
)
```



Validation

```
start <- Sys.time()
auc_rfandsvm <- validate_svmandrf(dat = labeled_data, nsim = 600)
end <- Sys.time()
end - start
saveRDS(auc_rfandsvm, "appendix.rds")
```

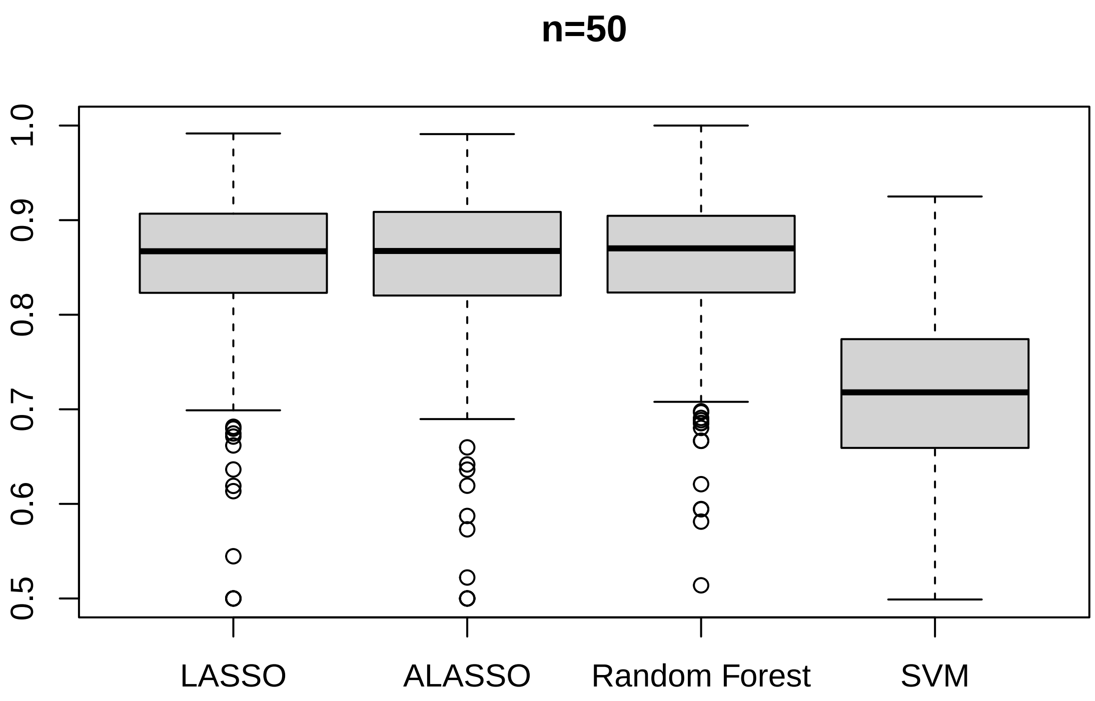
```
auc_rfandsvm <- readRDS("appendix.rds")

# median AUC
apply(auc_rfandsvm, 2, median)
```

```
##   n=50,rf   n=70,rf   n=90,rf  n=50,svm  n=70,svm  n=90,svm
## 0.8701826 0.8873775 0.9008419 0.7179279 0.7540064 0.7848541
```

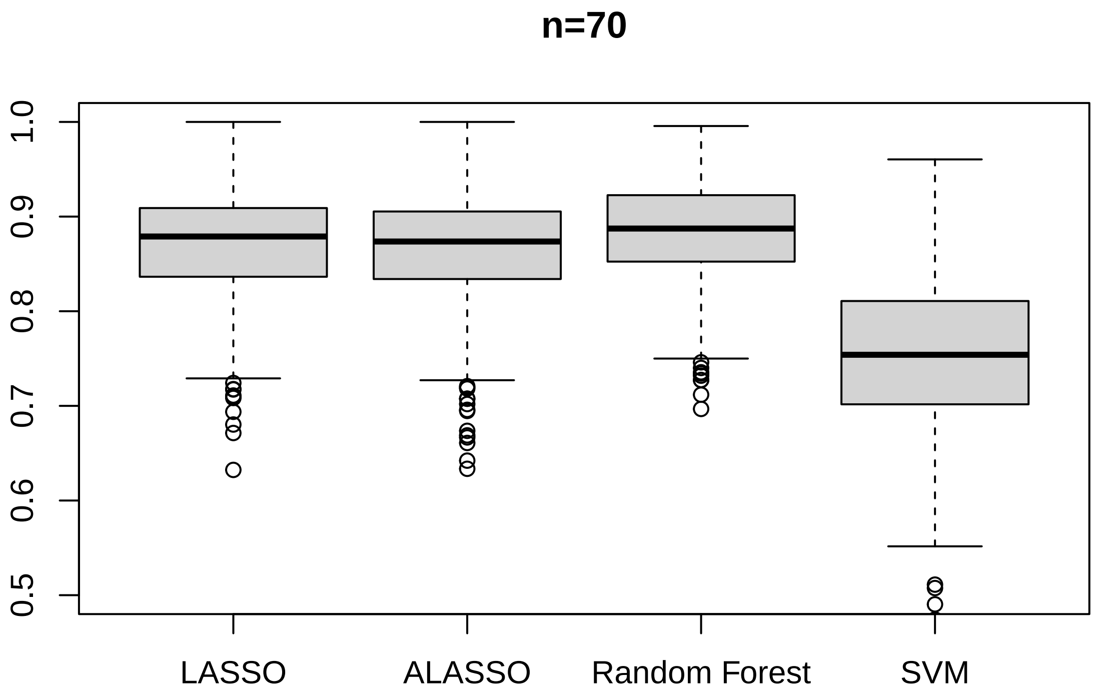
```
# SE
apply(auc_rfandsvm, 2, sd)
```

```
##    n=50,rf    n=70,rf    n=90,rf   n=50,svm   n=70,svm   n=90,svm
## 0.06771061 0.05256562 0.04799935 0.08221553 0.08092279 0.07128458
```

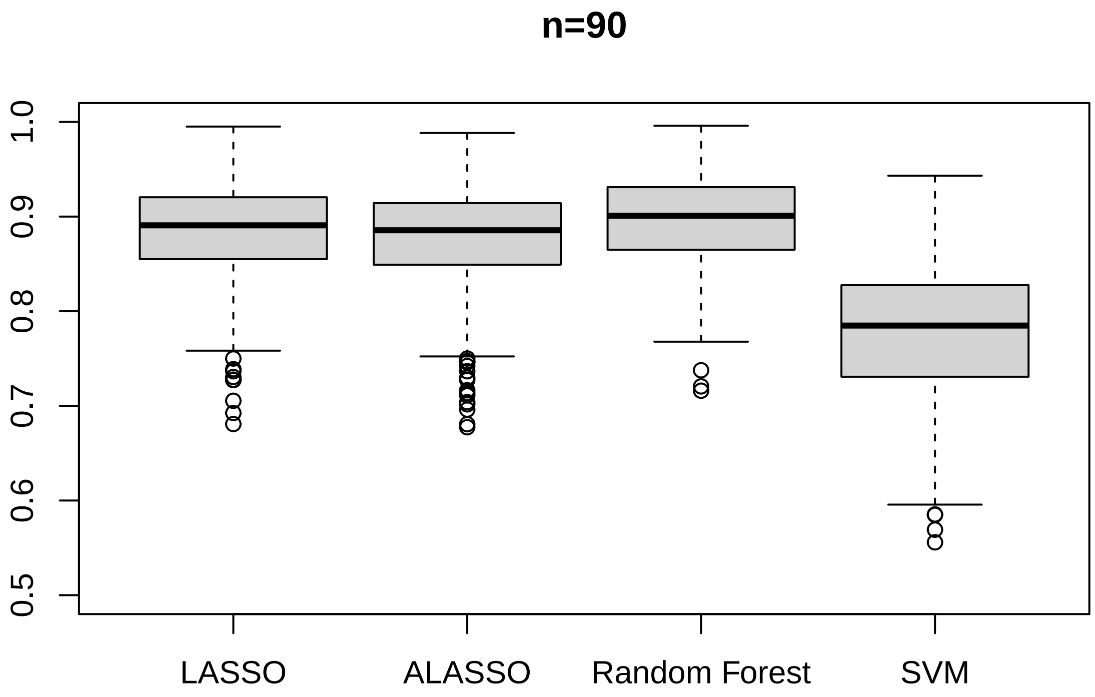
```
boxplot(cbind(auc_supervised, auc_rfandsvm) %>% select(starts_with("n=50")),
  ylim = c(0.5, 1),
  names = c("LASSO", "ALASSO", "Random Forest", "SVM"), main = "n=50"
)
```



```
boxplot(cbind(auc_supervised, auc_rfandsvm) %>% select(starts_with("n=70")),
  ylim = c(0.5, 1),
  names = c("LASSO", "ALASSO", "Random Forest", "SVM"), main = "n=70"
)
```



```
boxplot(cbind(auc_supervised, auc_rfandsvm) %>% select(starts_with("n=90")),
  ylim = c(0.5, 1),
  names = c("LASSO", "ALASSO", "Random Forest", "SVM"), main = "n=90"
)
```


n=90
1.0
0.9
0.8
0.7
0.6
0.5
LASSO
ALASSO
Random Forest
SVM