Module 2: Supervised learning

Split data into train and test

```
dim(train_x)
```

```
## [1] 106 588
```

```
length(train_y)
```

```
## [1] 106
```

```
dim(test_x)
```

```
## [1]  75 588
```

```
length(test_y)
```

```
## [1] 75
```

LASSO logistic regression

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

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

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

ALASSO logistic regression

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

```
# ALASSO features selected
names(beta.alasso[abs(beta.alasso) > 0])[-1]
```

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

```
# LASSO features selected
names(beta.lasso[abs(beta.lasso) > 0])[-1]
```

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

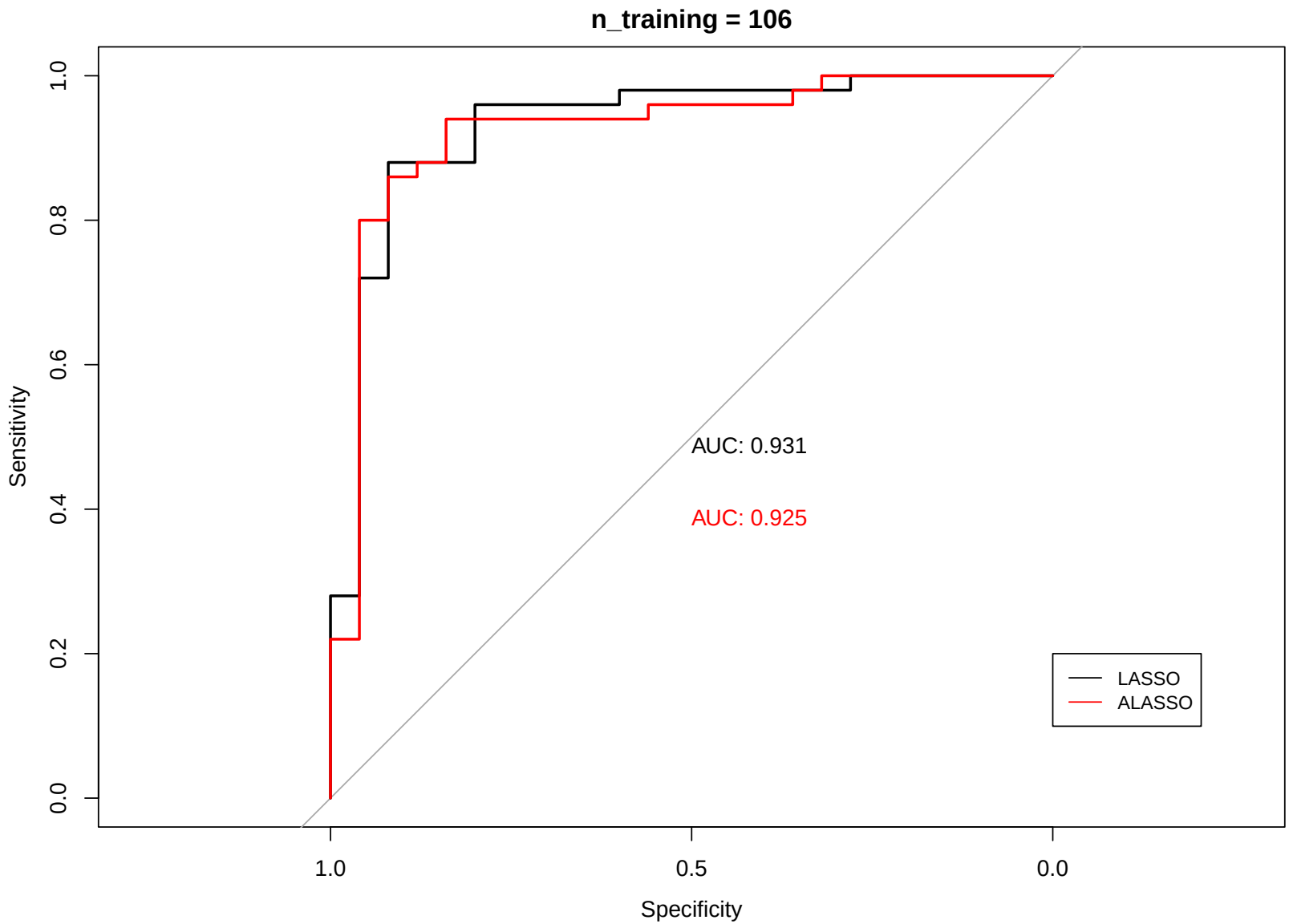
Get model predictions + ROC curve

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

```r
# Prediction on testing set (ALASSO)
y_hat.alasso <- linear_model_predict(
  beta = beta.alasso, x = test_x,
  probability = TRUE
)
```

```r
roc.lasso <- roc(test_y, y_hat.lasso)
roc.alasso <- roc(test_y, y_hat.alasso)
# as expected alasso selects less features
```

LASSO vs. ALASSO



LASSO vs. ALASSO at FPR = 0.10

```
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(0.1, roc_full.alasso)
```

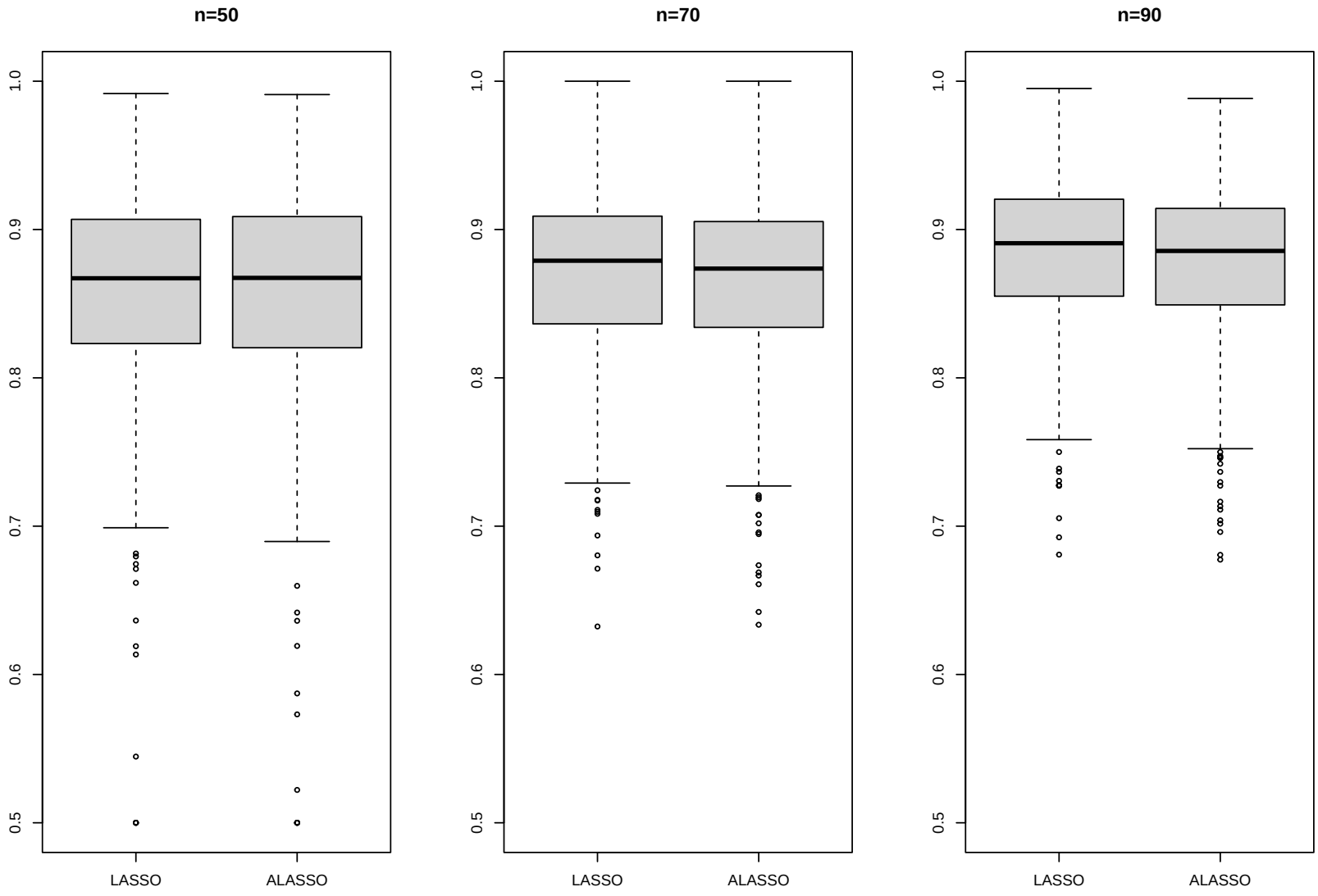
```
##      cutoff  pos.rate FPR  TPR       PPV       NPV        F1
## 1 0.7324685 0.6066667 0.1 0.86 0.9450549 0.7627119 0.9005236
```

LASSO vs. ALASSO with different training set size

- Randomly sample training size $= 50, 70, 90$
- Use the remaining data as the test set
- Repeat 600 times

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

LASSO vs. ALASSO with different training set size

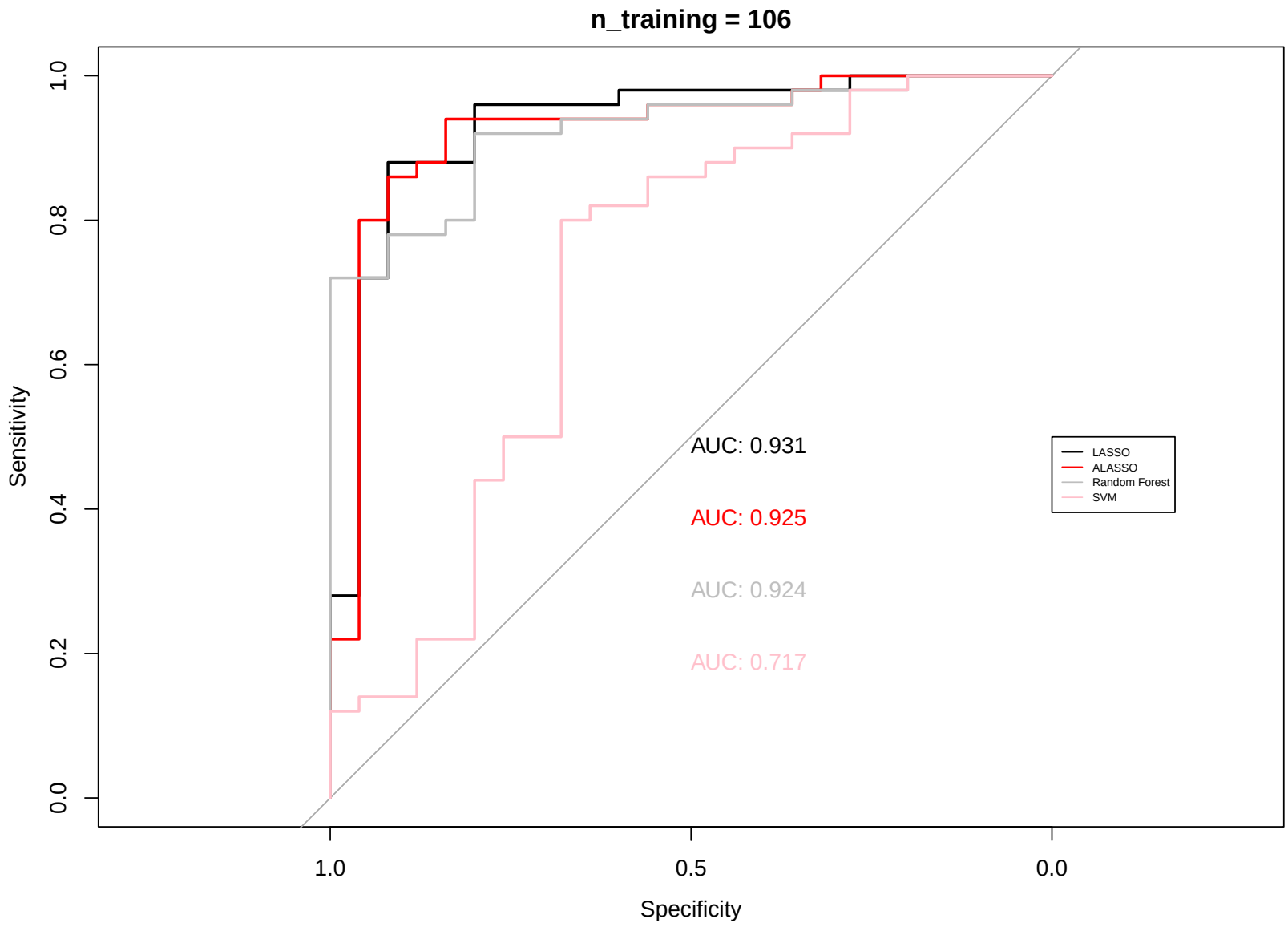


Random Forest and SVM

```
# 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)
# Use the Phecap functions to compute lasso, alssso etc and be consistnet
# clean the helper functions to only have what you need
# same for packages
# Make sure the slides match the markdown - please update
```

```
# 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)
```

ROC curves


n_training = 106
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
1.0
0.5
0.0
Specificity
AUC: 0.931
AUC: 0.925
AUC: 0.924
AUC: 0.717
LASSO
ALASSO
Random Forest
SVM