book_chapter

Team

2024-08-21

```
mydata<-read.csv("C:/Users/rywu/Desktop/HXPC13_DI_v3_11-13-2019.csv")
colnames(mydata)
```

```
##  [1] "course_id"         "userid_DI"         "registered"
##  [4] "viewed"            "explored"          "certified"
##  [7] "final_cc_cname_DI" "LoE_DI"            "YoB"
## [10] "gender"            "grade"             "start_time_DI"
## [13] "last_event_DI"     "nevents"           "ndays_act"
## [16] "nplay_video"       "nchapters"         "nforum_posts"
## [19] "roles"             "incomplete_flag"
```

```
data<-mydata[,c("certified","course_id","explored","gender","nevents","ndays_
act",
                "nplay_video","nchapters","nforum_posts","final_cc_cname_DI")
]
data[data == ""] <- NA
data<-na.omit(data)
```

```
table(data$certified,data$course_id)
```

```
##
##     HarvardX/PH207x/2012_Fall HarvardX/PH278x/2013_Spring
##   0                     17671                       11006
##   1                      1724                         616
```

```
table(data$certified,data$gender)
```

```
##
##         f     m
##   0 13077 15600
##   1  1078  1262
```

```
table(data$certified,data$explored)
```

```
##
##         0     1
##   0 26027  2650
##   1    60  2280
```

```
table(data$certified,data$final_cc_cname_DI)
```

```
##
##     Australia Bangladesh Brazil Canada China Colombia Egypt France Germany
##   0       434         73    923    692    27      361   669    193     436
```

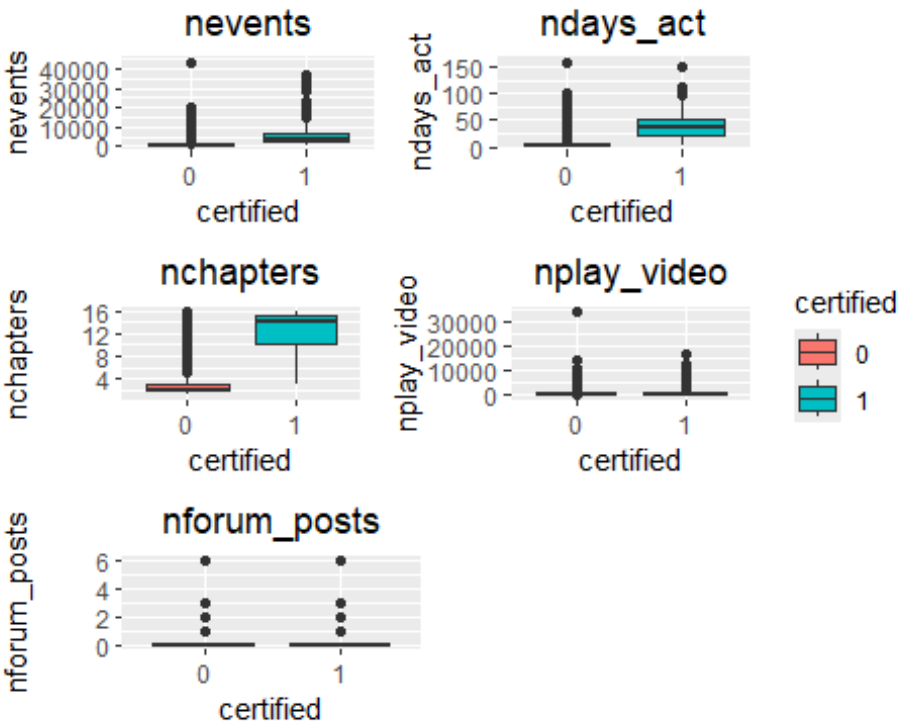
```
##   1        42          2     32     39     5       31    47     22      60
##
##     Greece India Indonesia Japan Mexico Morocco Nigeria Other Africa
##   0    164  4105       289    67    304      32     976         2756
##   1     15   508        20     4     15       1      80          254
##
##     Other East Asia Other Europe Other Middle East/Central Asia
##   0             173         2009                            794
##   1               9          160                             48
##
##     Other North & Central Amer., Caribbean Other Oceania Other South Ameri
ca
##   0                                    239             4                 5
76
##   1                                     14             1
37
##
##     Other South Asia Pakistan Philippines Poland Portugal Russian Federati
on
##   0              918       81         391     82      154                1
80
##   1               65        8          17     11       31
 5
##
##     Spain Ukraine United Kingdom United States Unknown/Other
##   0   730      69           1249          8385           142
##   1   159       2            106           478            12
```

```r
library(ggplot2)
library(patchwork)
# Ensure 'course_id' is a factor
data$course_id <- as.factor(data$course_id)
# Rename specific course names
levels(data$course_id)[levels(data$course_id) == "HarvardX/PH207x/2012_Fall"]
 <- "c_id1"
levels(data$course_id)[levels(data$course_id) == "HarvardX/PH278x/2013_Spring
"] <- "c_id2"

plot1 <- ggplot(data, aes(x = course_id, fill = factor(certified))) +
  geom_bar(position = "fill") +
  labs(y = "proportion", title = "course_id", fill = "certified") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))

# Bar plot for `certified` vs `gender`
plot2 <- ggplot(data, aes(x = gender, fill = factor(certified))) +
  geom_bar(position = "fill") +
  labs(y = "proportion", title = "gender", fill = "certified") +
  theme(plot.title = element_text(hjust = 0.5))
```

```r
# Bar plot for `certified` vs `explored`
plot3 <- ggplot(data, aes(x = explored, fill = factor(certified))) +
  geom_bar(position = "fill") +
  labs(y = "proportion", title = "explored", fill = "certified") +
  theme(plot.title = element_text(hjust = 0.5))

# Combine the plots in a row
combined_plot <- plot1 | plot2 | plot3

# Display the combined plot
combined_plot + plot_layout(guides = 'collect')
```


```r
# Boxplot for `certified` vs `nevents`
# Create individual plots with centered titles
plot1 <- ggplot(data, aes(x = factor(certified), y = nevents, fill = factor(c
ertified))) +
  geom_boxplot() +
  labs(title = "nevents", x = "certified", fill = "certified") +
  theme(plot.title = element_text(hjust = 0.5))

plot2 <- ggplot(data, aes(x = factor(certified), y = ndays_act, fill = factor
(certified))) +
  geom_boxplot() +
  labs(title = "ndays_act", x = "certified", fill = "certified") +
  theme(plot.title = element_text(hjust = 0.5))
```

```r
plot3 <- ggplot(data, aes(x = factor(certified), y = nchapters, fill = factor
(certified))) +
  geom_boxplot() +
  labs(title = "nchapters", x = "certified", fill = "certified") +
  theme(plot.title = element_text(hjust = 0.5))

plot4 <- ggplot(data, aes(x = factor(certified), y = nplay_video, fill = fact
or(certified))) +
  geom_boxplot() +
  labs(title = "nplay_video", x = "certified", fill = "certified") +
  theme(plot.title = element_text(hjust = 0.5))

plot5 <- ggplot(data, aes(x = factor(certified), y = nforum_posts, fill = fac
tor(certified))) +
  geom_boxplot() +
  labs(title = "nforum_posts", x = "certified", fill = "certified") +
  theme(plot.title = element_text(hjust = 0.5))


## Combine the plots with plot5 on the left side of row 3
combined_plot <- (plot1 | plot2) / (plot3 | plot4) / (plot5 | plot_spacer())
combined_plot + plot_layout(guides = 'collect', heights = c(1, 1, 1))
```

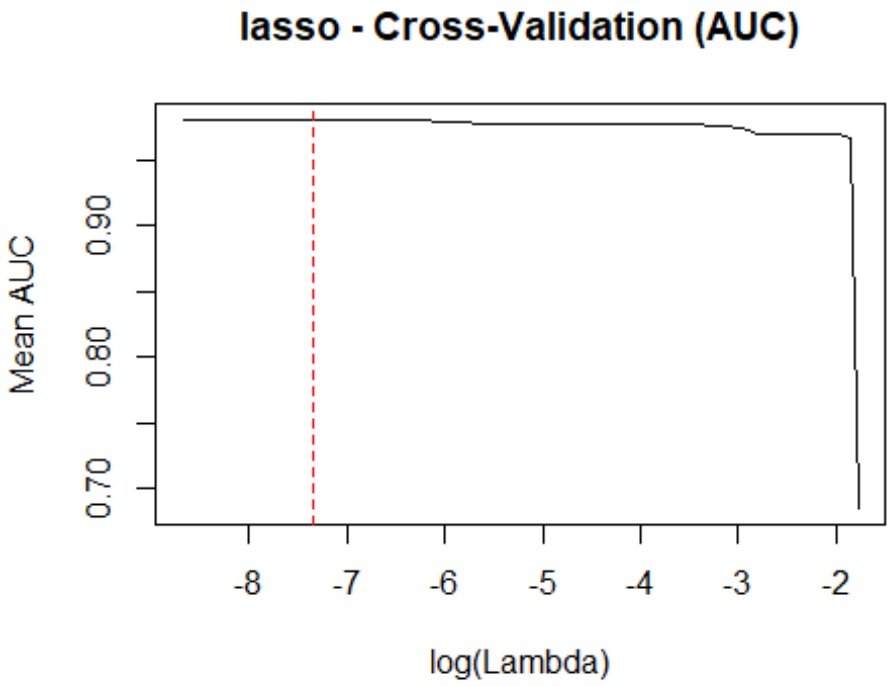



```r
data2<-as.data.frame(scale(data[,c("nevents", "ndays_act", "nplay_video", "nc
hapters", "nforum_posts")]))
data2$certified<-as.factor(data$certified)
```

```r
data2$course_id<-as.factor(data$course_id)
data2$explored<-as.factor(data$explored)
data2$gender<-as.factor(data$gender)
data2$final_cc_cname_DI<-as.factor(data$final_cc_cname_DI)

library(ncvreg)
library(pROC)

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

library(caret)

## Loading required package: lattice

# Assuming data2 is already loaded and prepared

# Split the data into training (80%) and test (20%) sets
set.seed(123)
train_indices <- createDataPartition(data2$certified, p = 0.8, list = FALSE)
train_data <- data2[train_indices, ]
test_data <- data2[-train_indices, ]

# Prepare the design matrices and response vectors
y_train <- train_data$certified
X_train <- model.matrix(certified ~ course_id + gender + final_cc_cname_DI +
nevents + ndays_act + nplay_video + explored + nchapters + nforum_posts + fin
al_cc_cname_DI + (course_id + nevents + ndays_act + nplay_video + explored +n
chapters + nforum_posts) * gender + (nevents + ndays_act + nplay_video + expl
ored + nchapters + nforum_posts) * course_id, data = train_data)[,-1]

y_test <- test_data$certified
X_test <- model.matrix(certified ~ course_id + gender + final_cc_cname_DI + n
events + ndays_act + nplay_video + explored +nchapters + nforum_posts + final
_cc_cname_DI + (course_id + nevents + ndays_act + nplay_video + explored + nc
hapters + nforum_posts) * gender + (nevents + ndays_act + nplay_video + explo
red +nchapters + nforum_posts) * course_id, data = test_data)[,-1]

# Function to perform cross-validation using AUC and fit the model
cv_and_fit_auc <- function(X, y, penalty, nfolds = 5) {
  set.seed(123)  # for reproducibility
  # Fit the model using the entire path
  fit <- ncvreg(X, y, family = "binomial", penalty = penalty, standardize = F
ALSE)
  # Perform k-fold cross-validation
```

```r
  folds <- sample(rep(1:nfolds, length.out = length(y)))

  aucs <- matrix(0, nrow = nfolds, ncol = length(fit$lambda))

  for (i in 1:nfolds) {
    test_indices <- which(folds == i)
    X_train_cv <- X[-test_indices, ]
    y_train_cv <- y[-test_indices]
    X_test_cv <- X[test_indices, ]
    y_test_cv <- y[test_indices]

    # Fit model on training data
    cv_fit <- ncvreg(X_train_cv, y_train_cv, family = "binomial", penalty = p
enalty, lambda = fit$lambda,standardize = FALSE)
    # Calculate AUC for each lambda
    for (j in 1:length(cv_fit$lambda)) {
      beta <- coef(cv_fit)[, j]

      # Ensure beta is a numeric vector
      beta <- as.numeric(beta)

      # Initialize vector to store probabilities
      probs <- numeric(nrow(X_test_cv))

      # Loop over each row in X_test_cv
      for (k in 1:nrow(X_test_cv)) {
        # Extract the predictor vector for the current row
        x_row <- X_test_cv[k, , drop = FALSE]

        # Calculate the linear predictor (including intercept)
        linear_predictor <- cbind(1, x_row) %*% beta

        # Calculate the predicted probability
        probs[k] <- 1 / (1 + exp(-linear_predictor))
      }

      # Calculate AUC for the current lambda
      roc_curve <- roc(y_test_cv, probs, quiet = TRUE)
      aucs[i, j] <- auc(roc_curve)
    }
  }

  # Calculate mean AUCs across folds
  mean_aucs <- colMeans(aucs)

  # Determine the optimal lambda
  optimal_lambda_index <- which.max(mean_aucs)
  optimal_lambda <- fit$lambda[optimal_lambda_index]
```

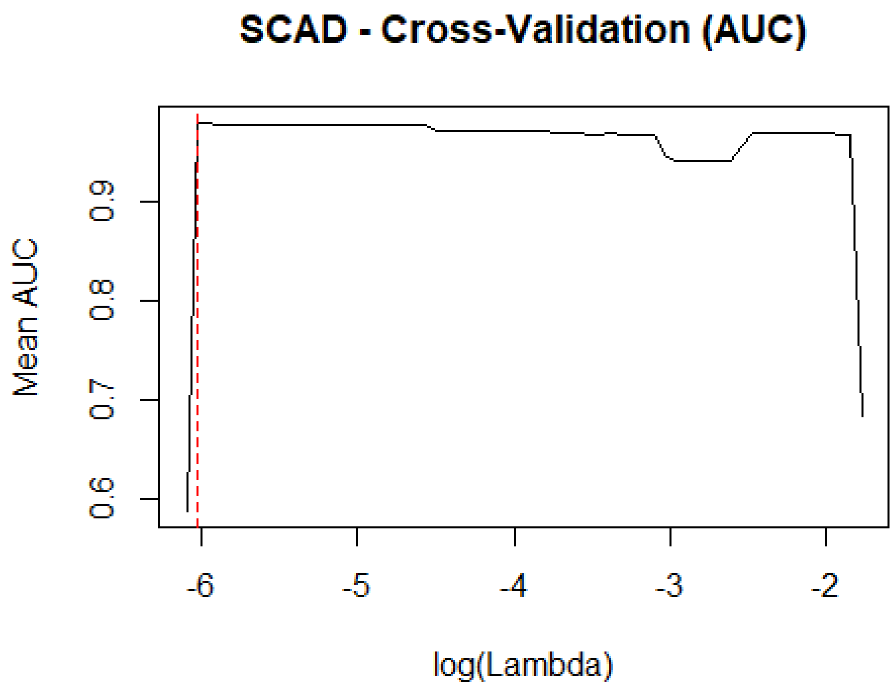
```r
  # Plot AUC vs lambda
  plot(log(fit$lambda), mean_aucs, type = "l",
       xlab = "log(Lambda)", ylab = "Mean AUC",
       main = paste(penalty, "- Cross-Validation (AUC)"))
  abline(v = log(optimal_lambda), col = "red", lty = 2)

  # Extract coefficients at the optimal lambda
  coef <- coef(fit)[, optimal_lambda_index]

  return(list(fit = fit, coef = coef, optimal_lambda = optimal_lambda, mean_a
ucs = mean_aucs))
}

# Fit models for each penalty
lasso_results <- cv_and_fit_auc(X_train, y_train, "lasso")
```



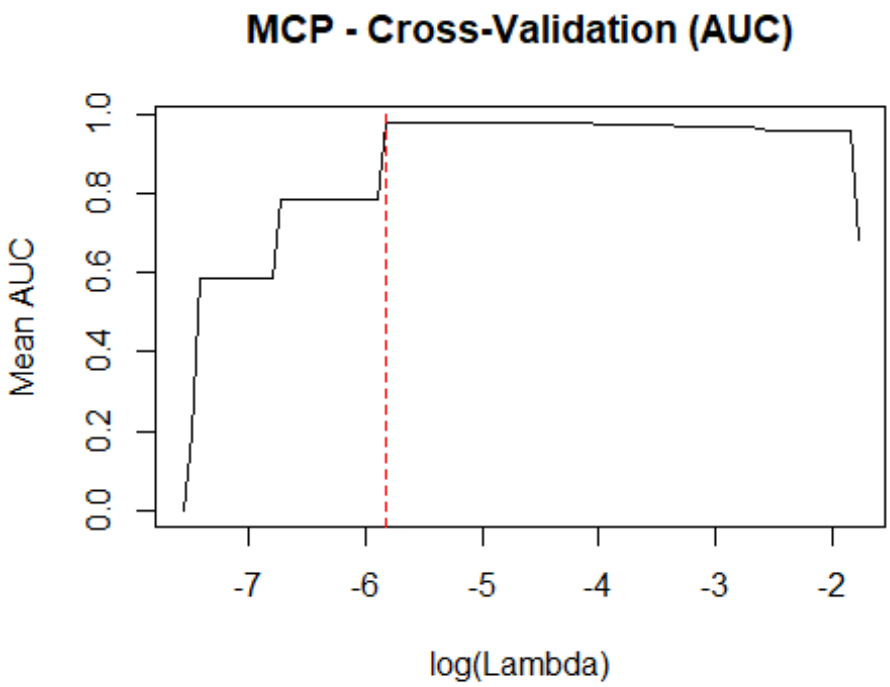

```r
scad_results <- cv_and_fit_auc(X_train, y_train, "SCAD")
```

```
## Warning in ncvreg(X, y, family = "binomial", penalty = penalty, standardiz
e =
## FALSE): Maximum number of iterations reached
```

```
## Warning in ncvreg(X_train_cv, y_train_cv, family = "binomial", penalty =
## penalty, : Maximum number of iterations reached
## Warning in ncvreg(X_train_cv, y_train_cv, family = "binomial", penalty =
## penalty, : Maximum number of iterations reached
```




```
mcp_results <- cv_and_fit_auc(X_train, y_train, "MCP")

## Warning in ncvreg(X, y, family = "binomial", penalty = penalty, standardiz
e =
## FALSE): Maximum number of iterations reached
## Warning in ncvreg(X, y, family = "binomial", penalty = penalty, standardiz
e =
## FALSE): Maximum number of iterations reached
## Warning in ncvreg(X, y, family = "binomial", penalty = penalty, standardiz
e =
## FALSE): Maximum number of iterations reached
## Warning in ncvreg(X, y, family = "binomial", penalty = penalty, standardiz
e =
## FALSE): Maximum number of iterations reached
## Warning in ncvreg(X, y, family = "binomial", penalty = penalty, standardiz
e =
## FALSE): Maximum number of iterations reached
## Warning in ncvreg(X, y, family = "binomial", penalty = penalty, standardiz
e =
## FALSE): Maximum number of iterations reached
```




```r
# Function to calculate test performance metrics
calculate_test_performance <- function(coef, X_test, y_test) {
  # Ensure X_test is a matrix
  X_test <- as.matrix(X_test)

  # Ensure coef is a numeric vector
  coef <- as.numeric(coef)

  # Check that coef length matches number of columns in X_test + 1 (for inter
cept)
  if (length(coef) != (ncol(X_test) + 1)) {
    stop("Length of coef must be equal to the number of predictors plus one f
or the intercept.")
  }

  # Initialize vectors to store linear predictors and probabilities
  linear_predictors <- numeric(nrow(X_test))
  probs <- numeric(nrow(X_test))

  # Loop over each row in X_test
  for (i in 1:nrow(X_test)) {
    # Extract the predictor vector
    x_row <- X_test[i, , drop = FALSE]

    # Calculate the linear predictor (including intercept)
    linear_predictors[i] <- cbind(1, x_row) %*% coef
```

```r
    # Calculate the predicted probability
    probs[i] <- 1 / (1 + exp(-linear_predictors[i]))
  }

  # Convert probabilities to binary predictions
  predictions <- ifelse(probs > 0.5, 1, 0)

  # Calculate performance metrics
  confusion <- confusionMatrix(factor(predictions), factor(y_test))

  accuracy <- confusion$overall['Accuracy']
  precision <- confusion$byClass['Pos Pred Value']
  recall <- confusion$byClass['Sensitivity']
  f1_score <- 2 * (precision * recall) / (precision + recall)

  # Calculate AUC
  roc_curve <- roc(y_test, probs, quiet = TRUE)
  auc <- auc(roc_curve)

  return(list(accuracy = accuracy, precision = precision, recall = recall, f1
_score = f1_score, auc = auc))
}
# Calculate test performance for each method
lasso_performance <- calculate_test_performance(lasso_results$coef, X_test, y
_test)
scad_performance <- calculate_test_performance(scad_results$coef, X_test, y_t
est)
mcp_performance <- calculate_test_performance(mcp_results$coef, X_test, y_tes
t)

# Print results
cat("\nTest Performance Metrics:\n")
```

```
##
## Test Performance Metrics:
```

```r
cat("LASSO:\n")
```

```
## LASSO:
```

```r
cat("Accuracy:", lasso_performance$accuracy, "\n")
```

```
## Accuracy: 0.9571175
```

```r
cat("Precision:", lasso_performance$precision, "\n")
```

```
## Precision: 0.9758135
```

```r
cat("Recall:", lasso_performance$recall, "\n")
```

```
## Recall: 0.9778553
```

```
cat("F1 Score:", lasso_performance$f1_score, "\n")
```

```
## F1 Score: 0.9768333
```

```
cat("AUC:", lasso_performance$auc, "\n")
```

```
## AUC: 0.982536
```

```
cat("\nSCAD:\n")
```

```
##
## SCAD:
```

```
cat("Accuracy:", scad_performance$accuracy, "\n")
```

```
## Accuracy: 0.9574399
```

```
cat("Precision:", scad_performance$precision, "\n")
```

```
## Precision: 0.9769834
```

```
cat("Recall:", scad_performance$recall, "\n")
```

```
## Recall: 0.9769834
```

```
cat("F1 Score:", scad_performance$f1_score, "\n")
```

```
## F1 Score: 0.9769834
```

```
cat("AUC:", scad_performance$auc, "\n")
```

```
## AUC: 0.9823125
```

```
cat("\nMCP:\n")
```

```
##
## MCP:
```

```
cat("Accuracy:", mcp_performance$accuracy, "\n")
```

```
## Accuracy: 0.9579236
```

```
cat("Precision:", mcp_performance$precision, "\n")
```

```
## Precision: 0.9768293
```

```
cat("Recall:", mcp_performance$recall, "\n")
```

```
## Recall: 0.9776809
```

```
cat("F1 Score:", mcp_performance$f1_score, "\n")
```

```
## F1 Score: 0.9772549
```

```r
cat("AUC:", mcp_performance$auc, "\n")
```

```
## AUC: 0.9800762
```

```r
# Compare selected variables
compare_selection <- function(lasso, scad, mcp) {
  all_vars <- unique(c(names(lasso), names(scad), names(mcp)))
  selection <- data.frame(Variable = all_vars,
                          LASSO = ifelse(all_vars %in% names(lasso)[lasso !=
0], "Selected", ""),
                          SCAD = ifelse(all_vars %in% names(scad)[scad != 0],
 "Selected", ""),
                          MCP = ifelse(all_vars %in% names(mcp)[mcp != 0], "S
elected", ""))
  return(selection)
}

variable_selection <- compare_selection(lasso_results$coef, scad_results$coef
, mcp_results$coef)
print(variable_selection)
```

```
##                                                Variable    LASSO     SC
AD
## 1                                           (Intercept) Selected Select
ed
## 2                                        course_idc_id2 Selected Select
ed
## 3                                               genderm

## 4                                final_cc_cname_DIBangladesh

## 5                                   final_cc_cname_DIBrazil Selected Select
ed
## 6                                   final_cc_cname_DICanada Selected

## 7                                    final_cc_cname_DIChina Selected

## 8                                 final_cc_cname_DIColombia Selected

## 9                                    final_cc_cname_DIEgypt Selected

## 10                                  final_cc_cname_DIFrance Selected

## 11                                final_cc_cname_DIGermany Selected

## 12                                  final_cc_cname_DIGreece Selected

## 13                                   final_cc_cname_DIIndia Selected Select
ed
## 14                                final_cc_cname_DIIndonesia
```

```
## 15                                  final_cc_cname_DIJapan
## 16                                 final_cc_cname_DIMexico
## 17                                final_cc_cname_DIMorocco
## 18                                final_cc_cname_DINigeria Selected Select
ed
## 19                            final_cc_cname_DIOther Africa
## 20                          final_cc_cname_DIOther East Asia
## 21                             final_cc_cname_DIOther Europe Selected
## 22          final_cc_cname_DIOther Middle East/Central Asia
## 23 final_cc_cname_DIOther North & Central Amer., Caribbean Selected
## 24                            final_cc_cname_DIOther Oceania Selected
## 25                        final_cc_cname_DIOther South America
## 26                           final_cc_cname_DIOther South Asia
## 27                                final_cc_cname_DIPakistan Selected Select
ed
## 28                              final_cc_cname_DIPhilippines Selected
## 29                                  final_cc_cname_DIPoland
## 30                                final_cc_cname_DIPortugal Selected Select
ed
## 31                         final_cc_cname_DIRussian Federation Selected
## 32                                   final_cc_cname_DISpain Selected Select
ed
## 33                                 final_cc_cname_DIUkraine Selected
## 34                            final_cc_cname_DIUnited Kingdom
## 35                             final_cc_cname_DIUnited States Selected
## 36                             final_cc_cname_DIUnknown/Other Selected
## 37                                                 nevents Selected Select
ed
## 38                                               ndays_act Selected Select
```

```
ed
## 39                                          nplay_video Selected Select
ed
## 40                                            explored1 Selected Select
ed
## 41                                            nchapters Selected Select
ed
## 42                                         nforum_posts Selected

## 43                                 course_idc_id2:genderm

## 44                                        genderm:nevents

## 45                                     genderm:ndays_act Selected

## 46                                    genderm:nplay_video

## 47                                      genderm:explored1

## 48                                      genderm:nchapters Selected

## 49                                    genderm:nforum_posts

## 50                                 course_idc_id2:nevents Selected Select
ed
## 51                               course_idc_id2:ndays_act Selected

## 52                             course_idc_id2:nplay_video Selected Select
ed
## 53                               course_idc_id2:explored1 Selected

## 54                               course_idc_id2:nchapters Selected

## 55                             course_idc_id2:nforum_posts Selected

##         MCP
## 1  Selected
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13 Selected
```

```
## 14
## 15
## 16
## 17
## 18 Selected
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32 Selected
## 33
## 34
## 35
## 36
## 37 Selected
## 38 Selected
## 39 Selected
## 40
## 41 Selected
## 42
## 43
## 44
## 45
## 46
## 47
## 48
## 49
## 50 Selected
## 51
## 52 Selected
## 53 Selected
## 54
## 55
```

```r
# Count number of selected variables for each method
num_vars <- c(sum(lasso_results$coef != 0) - 1,  # Subtract 1 to exclude inte
rcept
              sum(scad_results$coef != 0) - 1,
              sum(mcp_results$coef != 0) - 1)

cat("\nNumber of selected variables:\n")
```

```
##
## Number of selected variables:

cat("LASSO:", num_vars[1], "\n")

## LASSO: 36

cat("SCAD:", num_vars[2], "\n")

## SCAD: 14

cat("MCP:", num_vars[3], "\n")

## MCP: 10

# Plot regularization paths
par(mfrow = c(1, 3))
plot(lasso_results$fit, main = "LASSO Path")
abline(v = log(lasso_results$optimal_lambda), col = "red", lty = 2)
plot(scad_results$fit, main = "SCAD Path")
abline(v = log(scad_results$optimal_lambda), col = "red", lty = 2)
plot(mcp_results$fit, main = "MCP Path")
abline(v = log(mcp_results$optimal_lambda), col = "red", lty = 2)
```