```r
# Load necessary libraries
library(readr)
library(glmnet)
library(Matrix)

# Read the data
column_names <- c("y", "x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10")
data <- readr::read_delim("/Users/kissshot894/Documents/MAE/Econometrics ll/
pset7.txt", delim = " ", col_names = column_names)

# Create a formula that includes all 2-way and 3-way interactions
interaction_formula <- reformulate(
  termlabels = paste(column_names[-1], collapse = "+"),
  response = "y"
)
# Extend the formula to include 2-way and 3-way interactions
extended_formula <- update(interaction_formula, . ~ . + (.)^2 + (.)^3)

# Generate the model matrix excluding the intercept
x <- model.matrix(extended_formula, data = data)[,-1]  # '-1' to exclude the intercept
column

# Extract the response variable
y <- data$y

# Fit Lasso model using glmnet with cross-validation
cvfit <- cv.glmnet(x, y, type.measure = "mse", nfolds = 5, alpha = 1)

# Display the CV plot (optional, if you are using an interactive R session)
plot(cvfit)

# Extract the lambda that gives the minimum mean cross-validated error
lambda.min <- cvfit$lambda.min

# Print the optimal lambda value
print(paste("The optimal lambda value is:", lambda.min))

# Fit final Lasso model using the selected lambda
finalfit <- glmnet(x, y, alpha = 1, lambda = lambda.min)

# Determine the number of non-zero coefficients
non_zero_coeff <- sum(coef(finalfit, s = lambda.min) != 0)

# Print active set of covariates
activeset <- coef(finalfit, s = lambda.min)
print(activeset)
print(paste("Number of covariates with non-zero coefficients:", non_zero_coeff))


# Adaptive Lasso
# Fit initial Lasso model to obtain coefficients for penalty factors

# Extract coefficients at the best lambda and calculate penalty factors
initial_coefficients <- coef(finalfit, s = lambda.min)[-1]

penalty_factors <- 1 / abs(initial_coefficients)

# Fit Adaptive Lasso using calculated penalty factors
adaptive_cv_fit <- cv.glmnet(x, y, alpha = 1, penalty.factor = penalty_factors,
type.measure = "mse")
lambda_adaptive <- adaptive_cv_fit$lambda.min
```

```
# Fit final Adaptive Lasso model using the selected lambda
adaptive_fit <- glmnet(x, y, alpha = 1, lambda = lambda_adaptive, penalty.factor =
penalty_factors)

# Determine the number of non-zero coefficients in the Adaptive Lasso model
non_zero_adaptive_coeff <- sum(coef(adaptive_fit, s = lambda_adaptive) != 0)

# Optionally, print the coefficients to see which are non-zero
active_set_adaptive <- coef(adaptive_fit, s = lambda_adaptive)
print(active_set_adaptive)

# Print results
print(paste("Optimal lambda for Adaptive Lasso:", lambda_adaptive))
print(paste("Number of covariates with non-zero coefficients in Adaptive Lasso:",
non_zero_adaptive_coeff))


# 3) Post-Lasso
# Extract non-zero coefficients from adaptive Lasso, excluding the intercept if
included
non_zero_indices <- which(coef(adaptive_fit, s = lambda_adaptive)[-1] != 0)  #
Assuming the first element is the intercept
non_zero_names <- colnames(x)[non_zero_indices]

# Subset the x matrix to include only columns with non-zero coefficients from adaptive
Lasso
x_selected <- x[, non_zero_names, drop = FALSE]

# Convert the subset x matrix to a data frame
x_selected_df <- as.data.frame(x_selected)

# Combine 'y' with 'x_selected_df' into a new data frame
ols_data <- cbind(y = y, x_selected_df)

# Fit the OLS model using only the selected covariates
post_lasso_model <- lm(y ~ . - 1, data = ols_data)  # '-1' to exclude the intercept

# Summarize the OLS model results
summary_post_lasso <- summary(post_lasso_model)

# Print the summary of the OLS model
print(summary_post_lasso)
```

```
1)
[1] "The optimal lambda value is: 0.00603300588258399″
[1] "Number of covariates with non-zero coefficients: 12″

176 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept) -6.452080e-03
x1           1.893285e-01
x2           .
x3          -1.975927e-01
x4           .
x5           .
x6           .
x7           .
x8           .
x9           .
x10          1.934630e-01
x1:x2        .
x1:x3        .
x1:x4        .
x1:x5        .
x1:x6        .
x1:x7        .
x1:x8        .
x1:x9        .
x1:x10       .
x2:x3        .
x2:x4        .
x2:x5        .
x2:x6        .
x2:x7        .
x2:x8        1.938849e-01
x2:x9        .
x2:x10       .
x3:x4        .
x3:x5        .
x3:x6        .
x3:x7        .
x3:x8        .
x3:x9        .
x3:x10       .
x4:x5        .
x4:x6        .
x4:x7        .
x4:x8        .
x4:x9        .
x4:x10       .
x5:x6        .
x5:x7        .
x5:x8        .
x5:x9        .
x5:x10       .
x6:x7        .
x6:x8        .
x6:x9        .
x6:x10       .
x7:x8        .
x7:x9        .
x7:x10       .
x8:x9        .
x8:x10       .
x9:x10       .
x1:x2:x3     .
```

```
x1:x2:x4     .
x1:x2:x5     .
x1:x2:x6     .
x1:x2:x7     .
x1:x2:x8     .
x1:x2:x9     .
x1:x2:x10    .
x1:x3:x4     .
x1:x3:x5     .
x1:x3:x6     .
x1:x3:x7     .
x1:x3:x8     .
x1:x3:x9     .
x1:x3:x10    .
x1:x4:x5    -7.549036e-05
x1:x4:x6     .
x1:x4:x7     .
x1:x4:x8     .
x1:x4:x9     .
x1:x4:x10    .
x1:x5:x6     .
x1:x5:x7    -2.832037e-03
x1:x5:x8     .
x1:x5:x9     .
x1:x5:x10    .
x1:x6:x7     .
x1:x6:x8     1.086008e-03
x1:x6:x9     .
x1:x6:x10    .
x1:x7:x8     .
x1:x7:x9     3.939691e-01
x1:x7:x10    4.521294e-06
x1:x8:x9     .
x1:x8:x10    .
x1:x9:x10    .
x2:x3:x4     .
x2:x3:x5     .
x2:x3:x6     .
x2:x3:x7     .
x2:x3:x8     .
x2:x3:x9     .
x2:x3:x10    .
x2:x4:x5     .
x2:x4:x6     .
x2:x4:x7     .
x2:x4:x8     .
x2:x4:x9     .
x2:x4:x10    .
x2:x5:x6     .
x2:x5:x7     .
x2:x5:x8     .
x2:x5:x9     .
x2:x5:x10    .
x2:x6:x7     .
x2:x6:x8    -9.304756e-04
x2:x6:x9     3.214970e-04
x2:x6:x10    .
x2:x7:x8     .
x2:x7:x9     .
x2:x7:x10    .
x2:x8:x9     .
x2:x8:x10    .
x2:x9:x10    .
x3:x4:x5     .
```

```
x3:x4:x6     .
x3:x4:x7     .
x3:x4:x8     .
x3:x4:x9     .
x3:x4:x10    .
x3:x5:x6     .
x3:x5:x7     .
x3:x5:x8     .
x3:x5:x9     .
x3:x5:x10    .
x3:x6:x7     .
x3:x6:x8     .
x3:x6:x9     .
x3:x6:x10    .
x3:x7:x8     .
x3:x7:x9     .
x3:x7:x10    .
x3:x8:x9     .
x3:x8:x10    .
x3:x9:x10    .
x4:x5:x6     .
x4:x5:x7     .
x4:x5:x8     .
x4:x5:x9     .
x4:x5:x10    .
x4:x6:x7     .
x4:x6:x8     .
x4:x6:x9     .
x4:x6:x10    .
x4:x7:x8     .
x4:x7:x9     .
x4:x7:x10    .
x4:x8:x9     .
x4:x8:x10    .
x4:x9:x10    .
x5:x6:x7     .
x5:x6:x8     .
x5:x6:x9     .
x5:x6:x10    .
x5:x7:x8     .
x5:x7:x9     .
x5:x7:x10    .
x5:x8:x9     .
x5:x8:x10    .
x5:x9:x10    .
x6:x7:x8     .
x6:x7:x9     .
x6:x7:x10    .
x6:x8:x9     .
x6:x8:x10    .
x6:x9:x10    .
x7:x8:x9     .
x7:x8:x10    .
x7:x9:x10    .
x8:x9:x10    .
```

```
2)
[1] "Optimal lambda for Adaptive Lasso: 0.462177128798116"
[1] "Number of covariates with non-zero coefficients in Adaptive Lasso: 6"
176 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) -0.006499725
x1           0.193548347
x2           .
x3          -0.201923475
x4           .
x5           .
x6           .
x7           .
x8           .
x9           .
x10          0.197796666
x1:x2        .
x1:x3        .
x1:x4        .
x1:x5        .
x1:x6        .
x1:x7        .
x1:x8        .
x1:x9        .
x1:x10       .
x2:x3        .
x2:x4        .
x2:x5        .
x2:x6        .
x2:x7        .
x2:x8        0.198194340
x2:x9        .
x2:x10       .
x3:x4        .
x3:x5        .
x3:x6        .
x3:x7        .
x3:x8        .
x3:x9        .
x3:x10       .
x4:x5        .
x4:x6        .
x4:x7        .
x4:x8        .
x4:x9        .
x4:x10       .
x5:x6        .
x5:x7        .
x5:x8        .
x5:x9        .
x5:x10       .
x6:x7        .
x6:x8        .
x6:x9        .
x6:x10       .
x7:x8        .
x7:x9        .
x7:x10       .
x8:x9        .
x8:x10       .
x9:x10       .
x1:x2:x3     .
x1:x2:x4     .
```

```
x1:x2:x5     .
x1:x2:x6     .
x1:x2:x7     .
x1:x2:x8     .
x1:x2:x9     .
x1:x2:x10    .
x1:x3:x4     .
x1:x3:x5     .
x1:x3:x6     .
x1:x3:x7     .
x1:x3:x8     .
x1:x3:x9     .
x1:x3:x10    .
x1:x4:x5     .
x1:x4:x6     .
x1:x4:x7     .
x1:x4:x8     .
x1:x4:x9     .
x1:x4:x10    .
x1:x5:x6     .
x1:x5:x7     .
x1:x5:x8     .
x1:x5:x9     .
x1:x5:x10    .
x1:x6:x7     .
x1:x6:x8     .
x1:x6:x9     .
x1:x6:x10    .
x1:x7:x8     .
x1:x7:x9     0.399160959
x1:x7:x10    .
x1:x8:x9     .
x1:x8:x10    .
x1:x9:x10    .
x2:x3:x4     .
x2:x3:x5     .
x2:x3:x6     .
x2:x3:x7     .
x2:x3:x8     .
x2:x3:x9     .
x2:x3:x10    .
x2:x4:x5     .
x2:x4:x6     .
x2:x4:x7     .
x2:x4:x8     .
x2:x4:x9     .
x2:x4:x10    .
x2:x5:x6     .
x2:x5:x7     .
x2:x5:x8     .
x2:x5:x9     .
x2:x5:x10    .
x2:x6:x7     .
x2:x6:x8     .
x2:x6:x9     .
x2:x6:x10    .
x2:x7:x8     .
x2:x7:x9     .
x2:x7:x10    .
x2:x8:x9     .
x2:x8:x10    .
x2:x9:x10    .
x3:x4:x5     .
x3:x4:x6     .
```

```
x3:x4:x7     .
x3:x4:x8     .
x3:x4:x9     .
x3:x4:x10    .
x3:x5:x6     .
x3:x5:x7     .
x3:x5:x8     .
x3:x5:x9     .
x3:x5:x10    .
x3:x6:x7     .
x3:x6:x8     .
x3:x6:x9     .
x3:x6:x10    .
x3:x7:x8     .
x3:x7:x9     .
x3:x7:x10    .
x3:x8:x9     .
x3:x8:x10    .
x3:x9:x10    .
x4:x5:x6     .
x4:x5:x7     .
x4:x5:x8     .
x4:x5:x9     .
x4:x5:x10    .
x4:x6:x7     .
x4:x6:x8     .
x4:x6:x9     .
x4:x6:x10    .
x4:x7:x8     .
x4:x7:x9     .
x4:x7:x10    .
x4:x8:x9     .
x4:x8:x10    .
x4:x9:x10    .
x5:x6:x7     .
x5:x6:x8     .
x5:x6:x9     .
x5:x6:x10    .
x5:x7:x8     .
x5:x7:x9     .
x5:x7:x10    .
x5:x8:x9     .
x5:x8:x10    .
x5:x9:x10    .
x6:x7:x8     .
x6:x7:x9     .
x6:x7:x10    .
x6:x8:x9     .
x6:x8:x10    .
x6:x9:x10    .
x7:x8:x9     .
x7:x8:x10    .
x7:x9:x10    .
x8:x9:x10    .
```

```
Call:
lm(formula = y ~ . - 1, data = ols_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3127 -0.6783 -0.0048  0.6637  4.0979 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
x1          0.195276   0.003157   61.85   <2e-16 ***
x3         -0.203636   0.003153  -64.59   <2e-16 ***
x10         0.199537   0.003152   63.31   <2e-16 ***
`x2:x8`     0.199941   0.003150   63.48   <2e-16 ***
`x1:x7:x9`  0.400009   0.003164  126.41   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9968 on 99995 degrees of freedom
Multiple R-squared:  0.2417,	Adjusted R-squared:  0.2416 
F-statistic:  6373 on 5 and 99995 DF,  p-value: < 2.2e-16
```