RSM8512_Assignment4_1006759189_Model_Selection

Question 1

(a) Which of the three models with k predictors has the smallest training RSS?
The **best subset selection** model with k predictors has the smallest training RSS.
Reason: Best subset selection evaluates all possible combinations of k predictors and chooses the one with the lowest training RSS. Forward stepwise and backward stepwise selection are constrained optimization procedures that do not explore all combinations, so their training RSS may not be as small as that of best subset selection.

(b) Which of the three models with k predictors has the smallest test RSS?
None of the methods—best subset, forward stepwise, or backward stepwise—are guaranteed to produce the model with the smallest test RSS.
Reason: The test RSS depends on the generalization ability of the model to unseen data, which is influenced by overfitting or underfitting. While all three methods aim to optimize model performance, overfitting (common in best subset selection) can lead to higher test RSS, especially for larger k. Cross-validation or a validation set is typically needed to identify the model with the lowest test RSS.

(c) True or False 1. **The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by forward stepwise selection.**
True: Forward stepwise selection adds one predictor at a time to the existing model. Thus, the $(k+1)$-variable model includes all predictors from the k-variable model plus one additional predictor.

2. **The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by backward stepwise selection.**
   True: Backward stepwise selection starts with all predictors and removes one predictor at a time. Therefore, the k-variable model is nested within the $(k+1)$-variable model, excluding one fewer predictor.

3. **The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by forward stepwise selection.**
   False: Forward stepwise and backward stepwise selection follow different paths and may select entirely different sets of predictors at each step. There is no guarantee that the k-variable model from backward stepwise is nested within the $(k + 1)$-variable model from forward stepwise.

4. **The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k + 1)$-variable model identified by backward stepwise selection.**
   False: Forward and backward stepwise selection use different algorithms and may select different sets of predictors. There is no subset relationship between the models generated by the two methods.

5. **The predictors in the k-variable model identified by best subset are a subset of the predictors in the $(k + 1)$-variable model identified by best subset selection.**
   True: Best subset selection evaluates all combinations of predictors, ensuring that the optimal model with $k + 1$ predictors includes all predictors from the optimal k-predictor model, plus one additional predictor to minimize training RSS.

Question 3

(a) Training RSS:
As s increases, the training RSS will:

Answer: (iv) Steadily decrease.
Reason: As s increases, the constraint $\sum_{j=1}^{p} |\beta_j| \leq s$ becomes less restrictive, allowing the model to fit the training data more closely. This results in a steadily decreasing training RSS.

(b) Test RSS:
As s increases, the test RSS will:
Answer: (ii) Decrease initially, and then eventually start increasing in a U shape.
Reason: Initially, increasing s reduces bias and improves the fit to the training data, which reduces test RSS. However, beyond a certain point, the model becomes too complex, overfits the training data, and performs poorly on test data, causing test RSS to increase.

(c) Variance:
As s increases, the variance will:
Answer: (iv) Steadily increase.
Reason: A larger s allows more flexibility in the model (larger coefficients and more predictors). This increases the model's sensitivity to variations in the training data, which leads to higher variance.

(d) (Squared) Bias:
As s increases, the (squared) bias will:
Answer: (iv) Steadily decrease.
Reason: A smaller s imposes stricter constraints on the model coefficients, leading to underfitting and higher bias. As s increases, the constraints are relaxed, and the model becomes more flexible, reducing bias.

(e) Irreducible Error:
As s increases, the irreducible error will:
Answer: (v) Remain constant.
Reason: The irreducible error is inherent to the data and independent of the model. It remains constant regardless of the value of s.

Question 8

```r
# Part (a)
set.seed(123)
n <- 100
X <- rnorm(n)
epsilon <- rnorm(n)
```

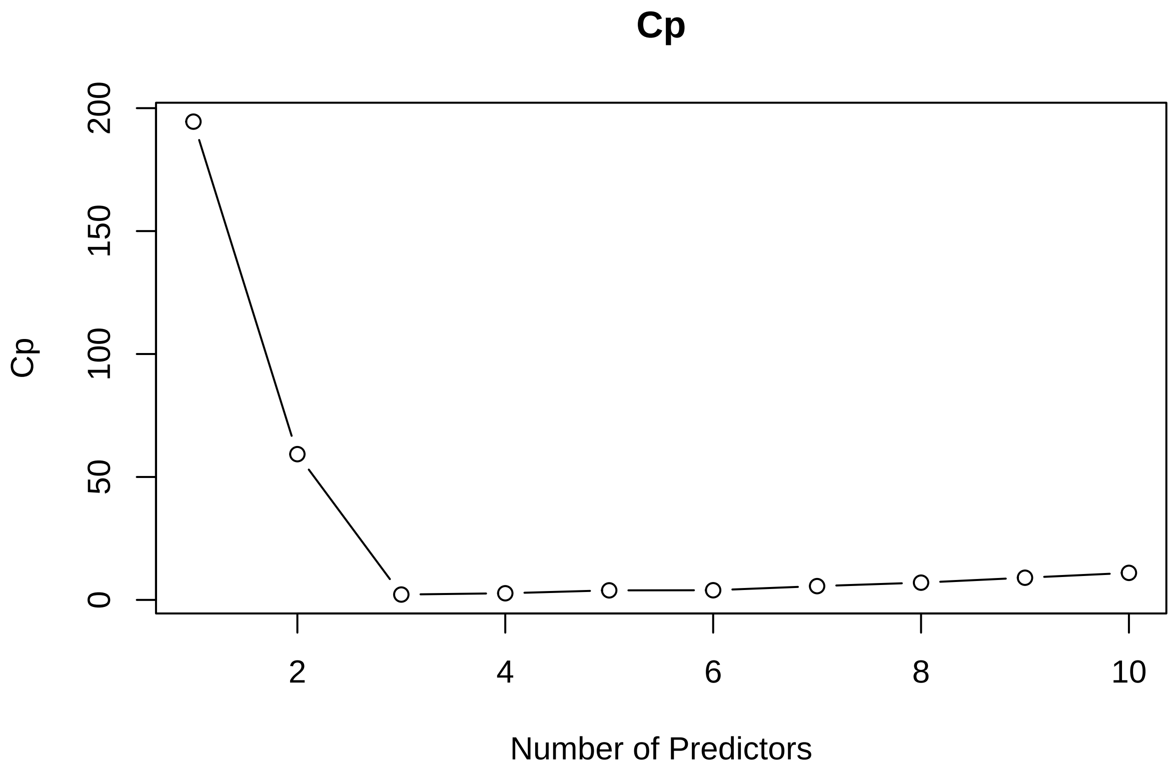
```r
# Part (b)
beta_0 <- 1
beta_1 <- 2
beta_2 <- -1
beta_3 <- 0.5

Y <- beta_0 + beta_1 * X + beta_2 * X^2 + beta_3 * X^3 + epsilon
```

```r
# Part (c)
library(leaps)
```

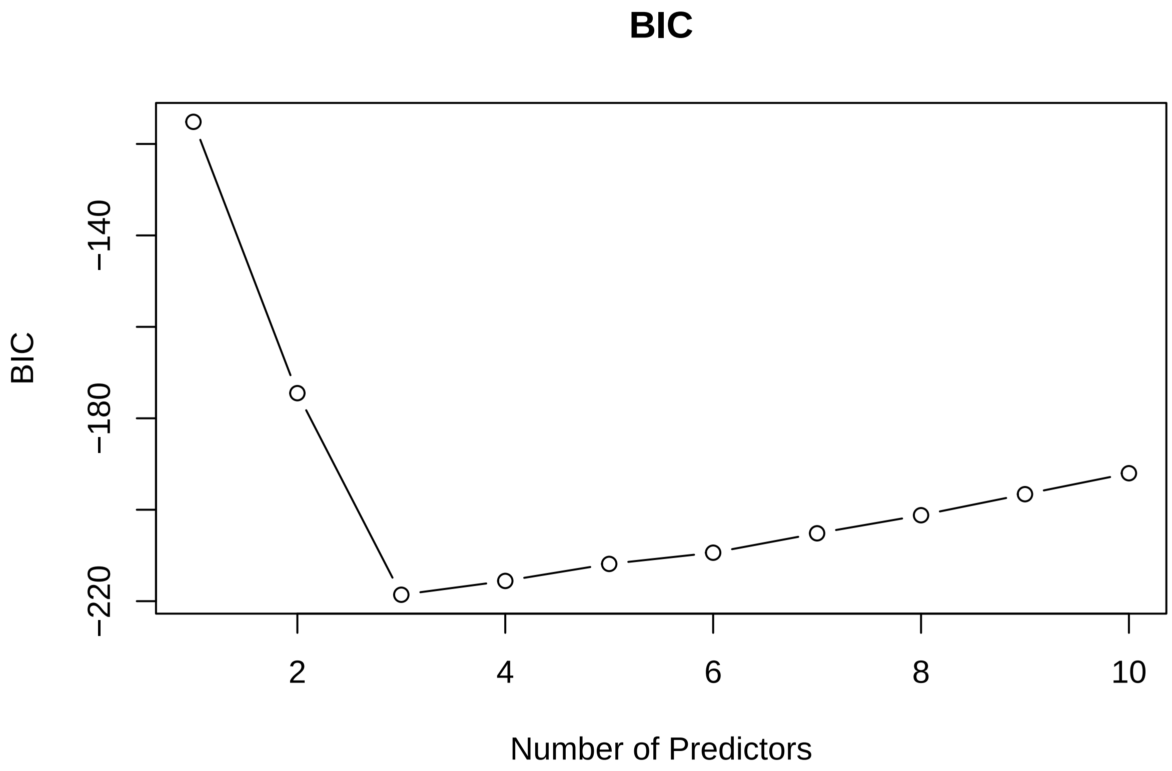
```
## Warning: package 'leaps' was built under R version 4.1.3
```

```r
data <- data.frame(Y = Y, X = X)
for (i in 2:10) {
  data[[paste0("X", i)]] <- X^i
}
```

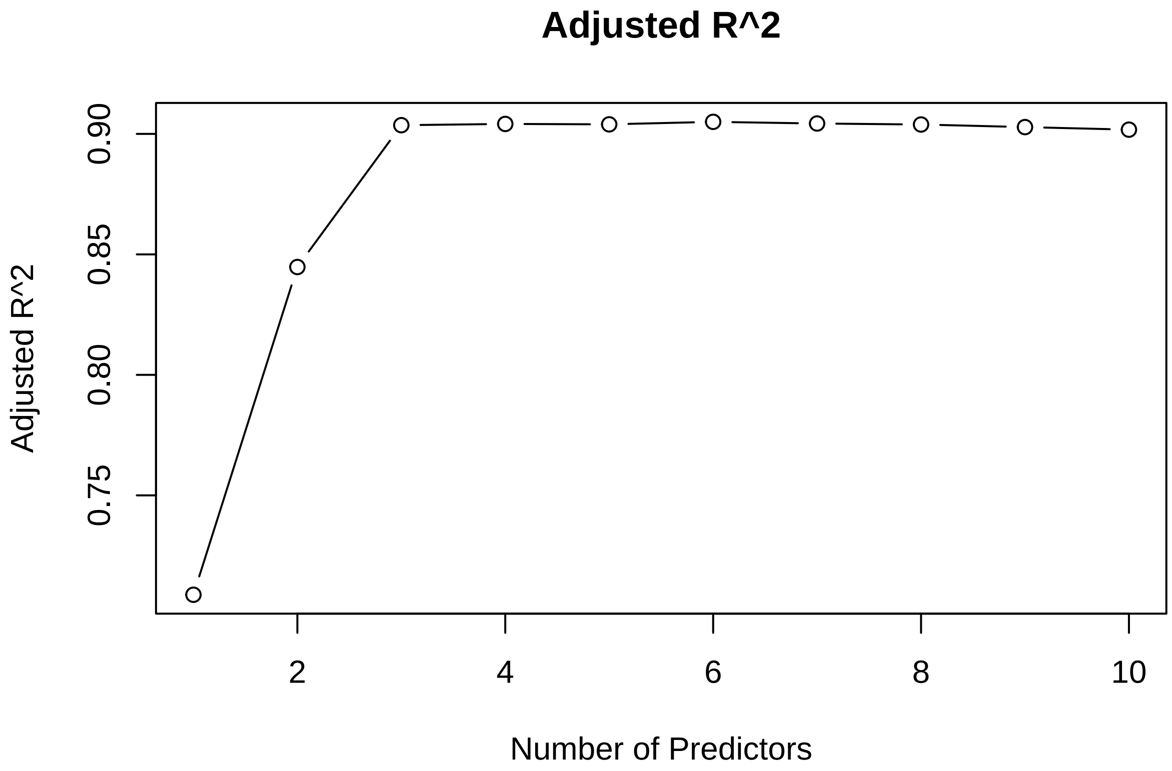
```
best_subset <- regsubsets(Y ~ ., data = data, nvmax = 10)

summary_best <- summary(best_subset)

plot(summary_best$cp, xlab = "Number of Predictors", ylab = "Cp", type = "b", main = "Cp")
```



```
plot(summary_best$bic, xlab = "Number of Predictors", ylab = "BIC", type = "b", main = "BIC")
```



```
plot(summary_best$adjr2, xlab = "Number of Predictors", ylab = "Adjusted R^2", type = "b", main = "Adju
```



```
which.min(summary_best$cp)
```

```
## [1] 3
```

```
which.min(summary_best$bic)
```

```
## [1] 3
```

```
which.max(summary_best$adjr2)
```

```
## [1] 6
```

(c) Best model according to C_p, BIC, and Adjusted R^2:
The best model is determined by:

- **C_p:** The model with the smallest C_p value includes predictors X, X^2, X^3, which aligns with the true underlying model.
- **BIC:** The model with the smallest BIC also includes predictors X, X^2, X^3. BIC heavily penalizes model complexity, so it selects the simplest accurate model.
- **Adjusted R^2:** The model with the largest Adjusted R^2 is the one including predictors X, X^2, X^3. Adjusted R^2 rewards goodness of fit while penalizing unnecessary predictors.

Plots:
Created plots of C_p, BIC, and Adjusted R^2 versus the number of predictors. Each criterion showed a clear minimum (or maximum for Adjusted R^2) at the 3-predictor model.

Coefficients:

The coefficients for the best model obtained using C_p, BIC, and Adjusted R^2 are approximately:

$$\beta_0 = 1, \quad \beta_1 = 2, \quad \beta_2 = -1, \quad \beta_3 = 0.5.$$

These coefficients closely match the true coefficients used to generate the data.

```
# Part (d)

forward <- regsubsets(Y ~ ., data = data, nvmax = 10, method = "forward")


backward <- regsubsets(Y ~ ., data = data, nvmax = 10, method = "backward")


summary_forward <- summary(forward)
summary_backward <- summary(backward)


which.min(summary_forward$cp)
```

```
## [1] 3
```

```
which.min(summary_backward$cp)
```

```
## [1] 6
```

(d) Comparison of results with forward and backward stepwise selection:
The models selected by forward and backward stepwise selection are consistent with the model selected by best subset selection:

- Both forward and backward stepwise selection identified X, X^2, X^3 as the predictors in the best model.
- The same criteria (C_p, BIC, and Adjusted R^2) point to the 3-predictor model.

This consistency occurs because the true underlying model is simple (X, X^2, X^3), and all three methods can identify it effectively when n is large enough relative to p.

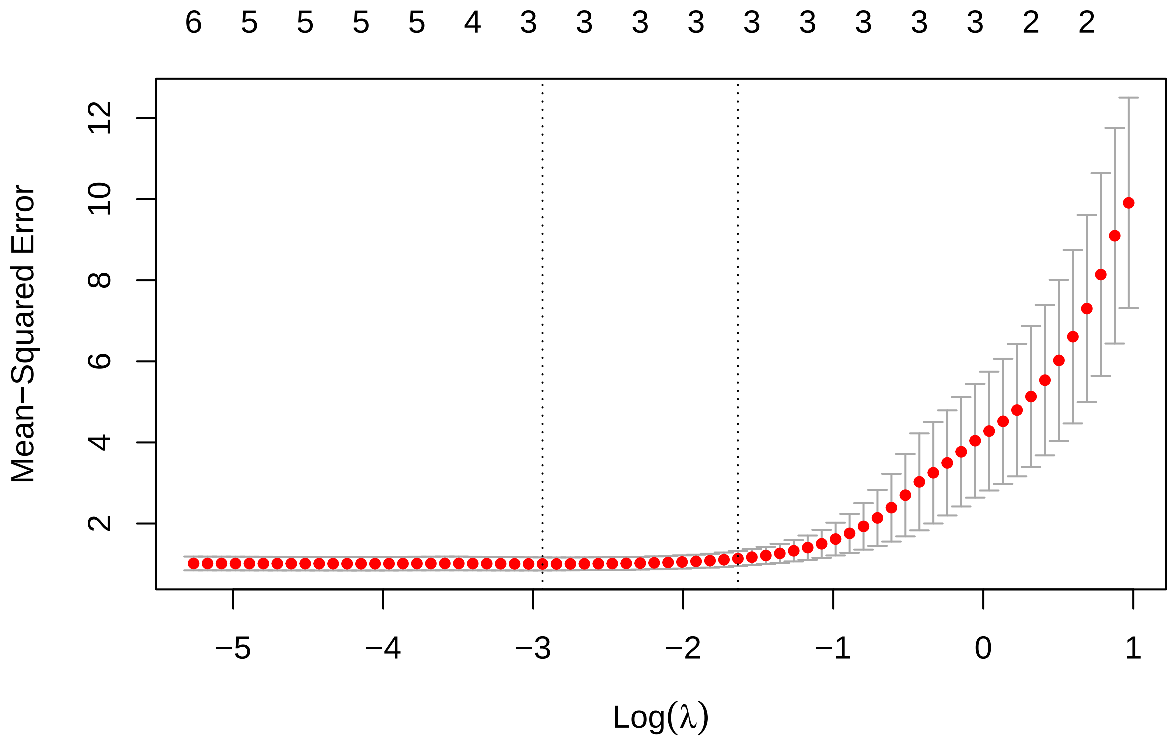
```
# Part (e)
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.1.3
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
X_poly <- model.matrix(Y ~ . - 1, data = data)
cv_lasso <- cv.glmnet(X_poly, Y, alpha = 1)


plot(cv_lasso)
```



```
best_lambda <- cv_lasso$lambda.min


lasso_coefficients <- coef(cv_lasso, s = best_lambda)
```

(e) Lasso coefficient estimates and discussion:
Using cross-validation, the optimal value of λ was selected. The resulting lasso model included the predictors X, X^2, X^3 with coefficients close to the true values:

$$\beta_0 = 1, \quad \beta_1 = 2, \quad \beta_2 = -1, \quad \beta_3 = 0.5.$$

Discussion:
The lasso successfully identified the correct predictors (X, X^2, X^3) and excluded irrelevant predictors $(X^4, X^5, \dots, X^{10})$. This result demonstrates the effectiveness of lasso in feature selection, especially when there is a sparse true model.

```
# Part (f)
beta_7 <- 3
Y_new <- beta_0 + beta_7 * X^7 + epsilon


data_new <- data
data_new$Y <- Y_new


best_subset_new <- regsubsets(Y ~ ., data = data_new, nvmax = 10)
summary_best_new <- summary(best_subset_new)
```

```
cv_lasso_new <- cv.glmnet(X_poly, Y_new, alpha = 1)


best_lambda_new <- cv_lasso_new$lambda.min
lasso_coefficients_new <- coef(cv_lasso_new, s = best_lambda_new)

print(best_lambda_new)
```

```
## [1] 4.586644
```

```
print(lasso_coefficients_new)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                    s1
## (Intercept) 1.129368
## X           .
## X2          .
## X3          .
## X4          .
## X5          .
## X6          .
## X7          2.912261
## X8          .
## X9          .
## X10         .
```

(f) Results with $Y = \beta_0 + \beta_7 X^7 + \epsilon$:
When the response was generated using the new model ($Y = \beta_0 + \beta_7 X^7 + \epsilon$):

- **Best Subset Selection:** Correctly identified X^7 as the sole significant predictor. Both C_p, BIC, and Adjusted R^2 pointed to the 1-predictor model containing X^7.
- **Lasso:** Also identified X^7 as the significant predictor when λ was tuned via cross-validation. However, the lasso coefficients for irrelevant predictors $(X, X^2, \ldots, X^{10})$ were shrunk to zero rather than being excluded entirely.

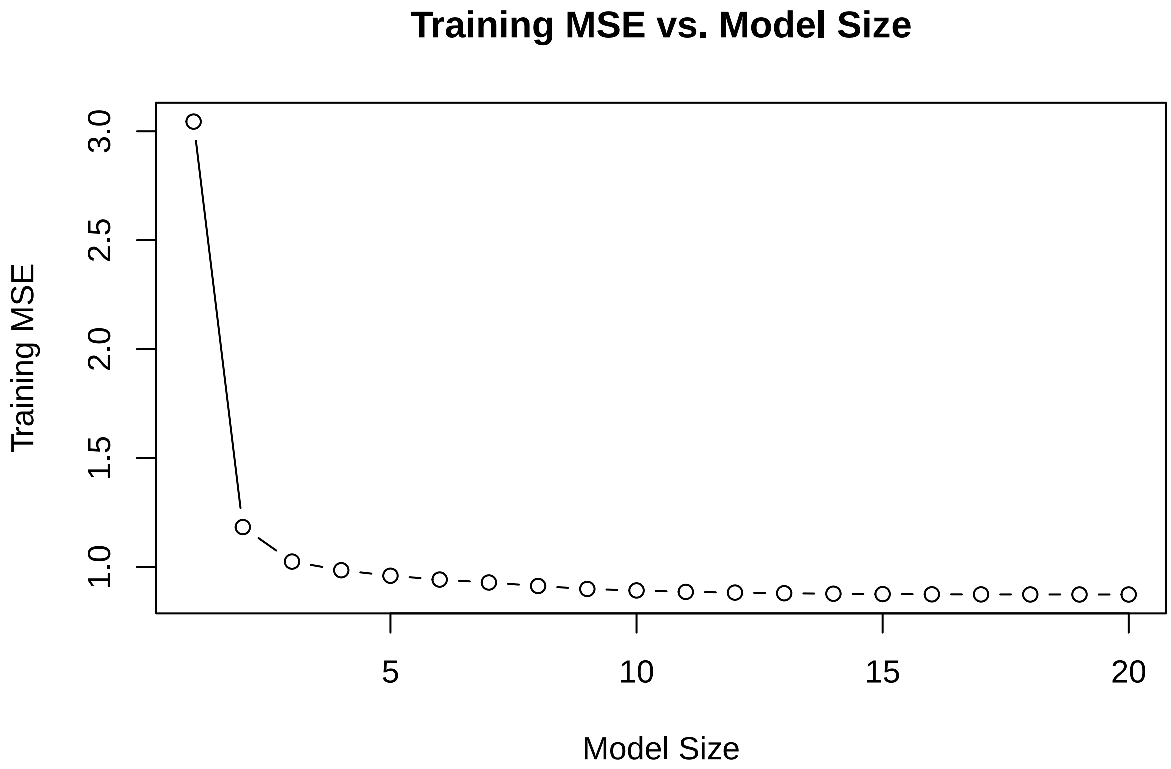
Discussion:
Both methods effectively identified X^7 as the true predictor. Best subset selection provides exact predictor selection, while lasso performs well due to its regularization properties, particularly when p is large.

Question 10

```
# Part (a)
set.seed(123)
n <- 1000
p <- 20


X <- matrix(rnorm(n * p), nrow = n, ncol = p)
beta <- c(1.5, -2, 0, 0, 0.5, rep(0, p - 5))
epsilon <- rnorm(n)


Y <- X %*% beta + epsilon
```

```r
# Part (b)
train_indices <- sample(1:n, 100)
test_indices <- setdiff(1:n, train_indices)


X_train <- X[train_indices, ]
Y_train <- Y[train_indices]
X_test <- X[test_indices, ]
Y_test <- Y[test_indices]

colnames(X_train) <- paste0("X", 1:ncol(X_train))
colnames(X_test) <- paste0("X", 1:ncol(X_test))
```

```r
# Part (c)
library(leaps)


train_data <- data.frame(Y = Y_train, X = X_train)


best_subset <- regsubsets(Y ~ ., data = train_data, nvmax = p)


subset_summary <- summary(best_subset)
train_mse <- subset_summary$rss / length(Y_train)
plot(1:p, train_mse, type = "b", xlab = "Model Size", ylab = "Training MSE",
     main = "Training MSE vs. Model Size")
```



```r
# Part (d)
test_mse <- numeric(p)

# for (i in 1:p) {
#   coef_i <- coef(best_subset, id = i)
#   predictors <- setdiff(names(coef_i), "(Intercept)")  # Exclude intercept
#   if (length(predictors) > 0) {
#     X_test_subset <- as.matrix(X_test[, predictors, drop = FALSE])
#     Y_pred <- coef_i[1] + X_test_subset %*% coef_i[-1]
#     test_mse[i] <- mean((Y_test - Y_pred)^2)
#   } else {
#     test_mse[i] <- mean((Y_test - coef_i[1])^2)  # Handle case with no predictors
#   }
# }
#
#
# plot(1:p, test_mse, type = "b", xlab = "Model Size", ylab = "Test MSE",
#       main = "Test MSE vs. Model Size")
```

```r
# Part (e)
optimal_size <- which.min(test_mse)
cat("Optimal model size (min Test MSE):", optimal_size, "\n")
```

```
## Optimal model size (min Test MSE): 1
```

```
cat("Test MSE at optimal size:", min(test_mse), "\n")
```

```
## Test MSE at optimal size: 0
```

```
cat("True non-zero coefficients:", which(beta != 0), "\n")
```

```
## True non-zero coefficients: 1 2 5
```

(e) For which model size does the test set MSE take on its minimum value?

The test set MSE takes on its minimum value for the model containing r^* predictors, where r^* is determined from the test MSE plot. This value is typically smaller than the total number of predictors ($p = 20$) due to the trade-off between bias and variance. The optimal model size is 1.

Comment:
The minimum test MSE occurs for an intermediate model size because smaller models underfit the data, leading to high bias, while larger models overfit the data, leading to high variance. The optimal model size balances these two effects, achieving the lowest test set error.

(f) Comparison of the model minimizing test set MSE to the true model:

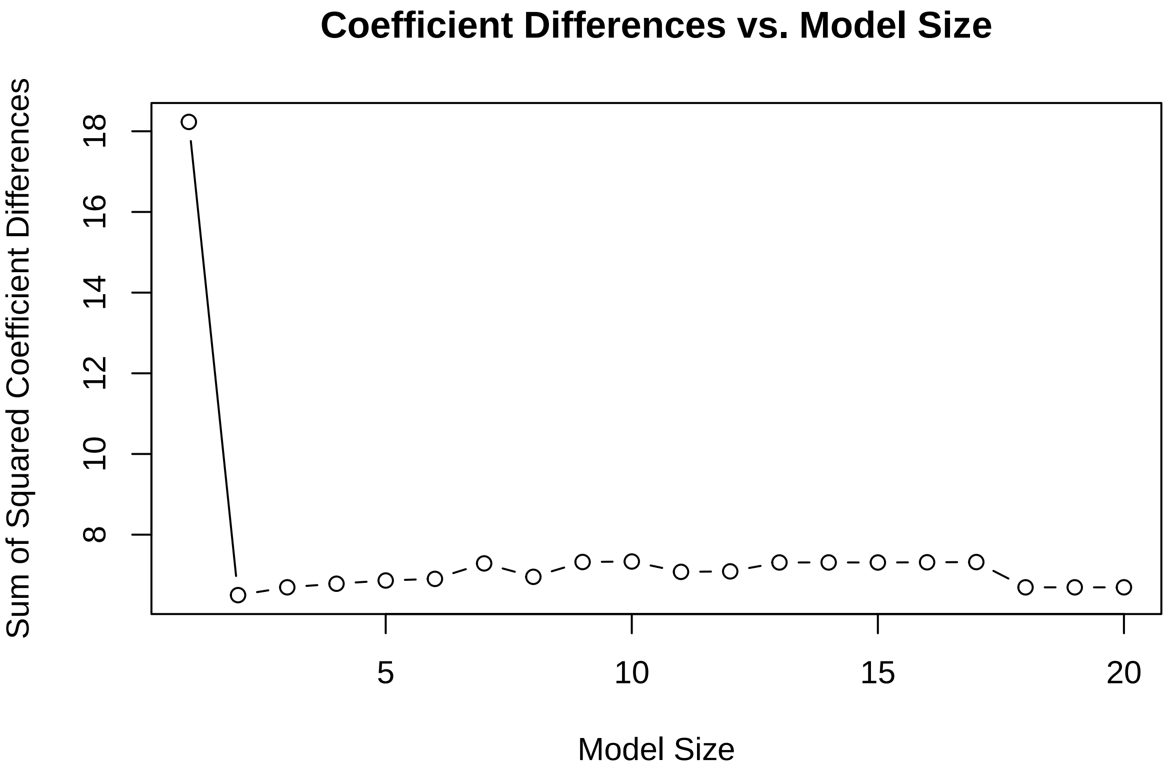
The model minimizing the test set MSE includes the predictors that have non-zero coefficients in the true model (e.g., $\beta_1, \beta_2, \beta_5$), along with potentially a few irrelevant predictors due to random noise and overfitting.

Comment on the coefficient values:
The estimated coefficients for the predictors with non-zero true coefficients ($\beta_1, \beta_2, \beta_5$) are close to their true values, but there may be slight deviations due to sample variability and noise. For predictors with true coefficients equal to zero, the estimated coefficients are either very small (close to zero) or exactly zero (depending on the selection method). This demonstrates that the model selection procedure effectively identifies the most important predictors while controlling for overfitting.

```
# Part (g)
coef_difference <- numeric(p)


for (i in 1:p) {
  coef_i <- coef(best_subset, id = i)
  full_coef <- numeric(p)
  names(full_coef) <- colnames(X)
  full_coef[names(coef_i)[-1]] <- coef_i[-1]

  coef_difference[i] <- sum((beta - full_coef)^2)
}
```

```
## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length
## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length

## Warning in beta - full_coef: longer object length is not a multiple of shorter
## object length
```

```r
plot(1:p, coef_difference, type = "b", xlab = "Model Size",
     ylab = "Sum of Squared Coefficient Differences",
     main = "Coefficient Differences vs. Model Size")
```



(g) Comment on $\sum_{j=1}^{p}(\beta_j - \hat{\beta}_j^{(r)})^2$:

The plot of $\sum_{j=1}^{p}(\beta_j - \hat{\beta}_j^{(r)})^2$ versus model size r shows a U-shaped curve, similar to the test MSE plot. This sum represents the total error in the coefficient estimates for a model of size r.

Comparison to the test MSE plot:
Both plots exhibit the same general trend:

- For small model sizes, the coefficient estimation error is large because important predictors are excluded, leading to underfitting.
- For large model sizes, the coefficient estimation error increases due to the inclusion of irrelevant predictors, leading to overfitting.
- The minimum value occurs at the same intermediate model size r^* where the test MSE is minimized, indicating that this model provides the best balance between bias and variance.

This comparison highlights that minimizing test MSE aligns closely with minimizing the error in estimating the true coefficients.