8.

# (a) 使用 rnorm() 函数生成长度为 n=100 的预测变量 X 和长度为 n=100 的噪声向量 ε

set.seed(1) # 为了结果可重现性设置随机种子

n <- 100

X <- rnorm(n)

epsilon <- rnorm(n)

# (b) 依据以下模型产生长度为 n=100 的响应变量 Y：

# 定义beta常数

beta0 <- 1

beta1 <- 2

beta2 <- 0.5

beta3 <- -1

# 生成响应变量 Y

Y <- beta0 + beta1 \* X + beta2 \* X^2 + beta3 \* X^3 + epsilon

# (c) 利用 regsubsets() 函数对数据集使用最优子集选择法，从包含预测变量X,X2,...,X10X,X2,...,X10的模型中选出最优的模型

# 创建X的多项式项

X\_poly <- model.matrix(Y ~ poly(X, 10, raw = TRUE))[,-1] # 移除截距项

colnames(X\_poly) <- paste0("X", 1:10) # 给列命名

# 将X的多项式项和Y组合成数据框

data\_df <- data.frame(Y, X\_poly)

# 使用regsubsets进行最优子集选择

library(leaps)

regfit.full <- regsubsets(Y ~ ., data = data\_df, nvmax = 10) # nvmax指定最大变量数

# 查看选择结果的统计量

summary\_regfit <- summary(regfit.full)

# 根据Cp、BIC和调整R^2选择最优模型

# 1. Cp准则 (Mallows' Cp)

# Cp越小越好，通常接近p的模型是好的

which.min(summary\_regfit$cp)

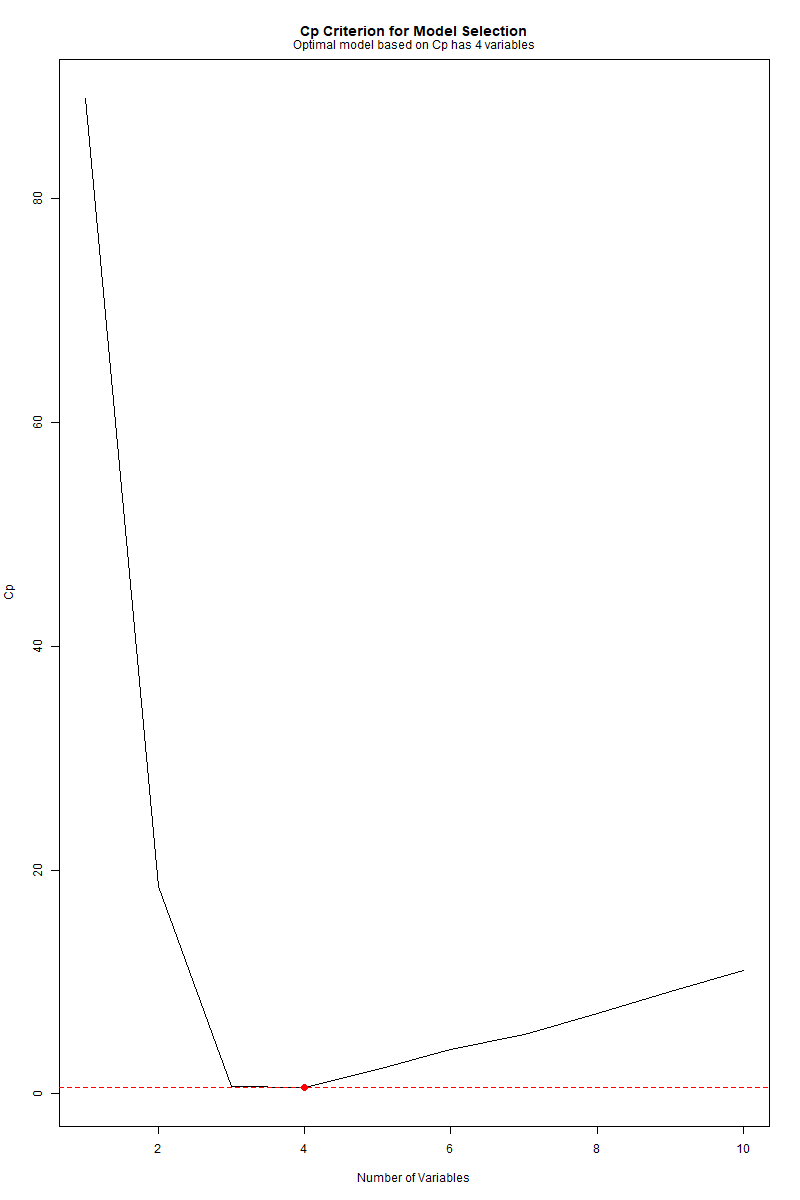
# 绘制Cp图

plot(summary\_regfit$cp, type = "l", xlab = "Number of Variables", ylab = "Cp", main = "Cp Criterion for Model Selection")

points(which.min(summary\_regfit$cp), summary\_regfit$cp[which.min(summary\_regfit$cp)], col = "red", cex = 2, pch = 20)

abline(h = min(summary\_regfit$cp), col = "red", lty = 2)

mtext(paste("Optimal model based on Cp has", which.min(summary\_regfit$cp), "variables"), side = 3, line = 0.5)



# 2. BIC准则 (Bayesian Information Criterion)

# BIC越小越好，惩罚更复杂的模型

which.min(summary\_regfit$bic)

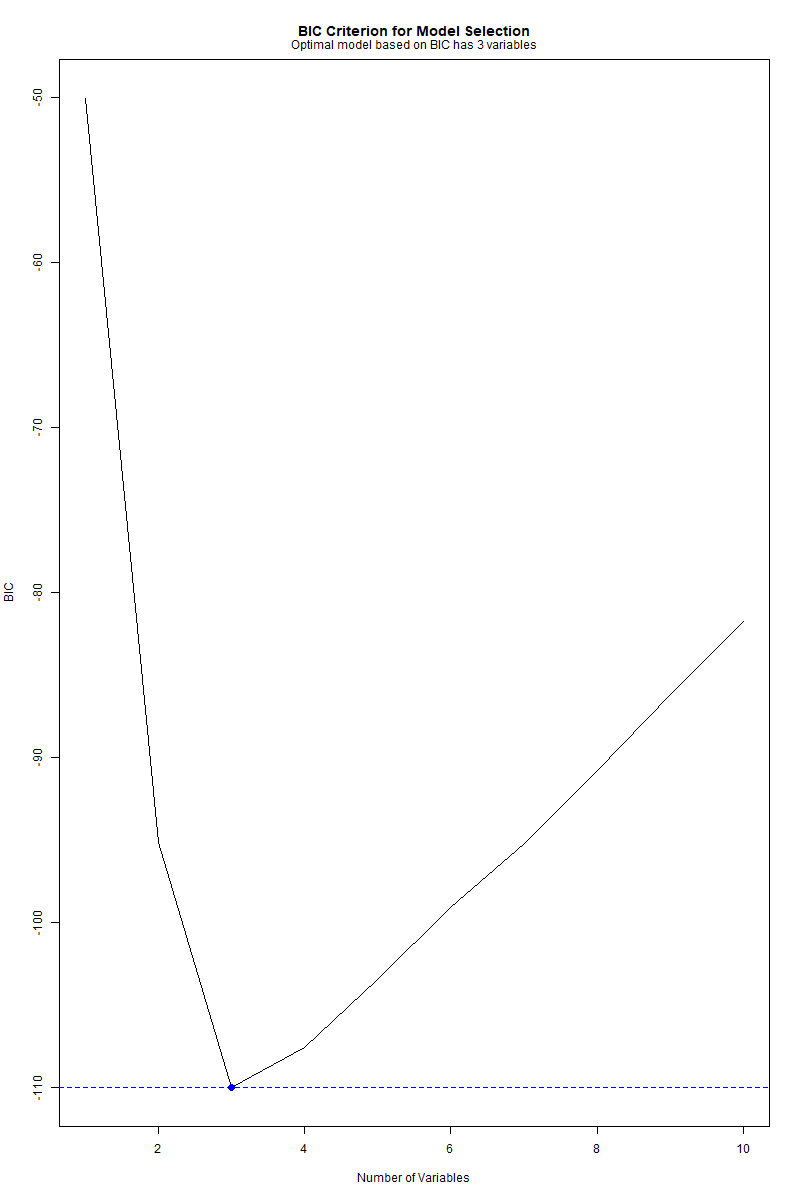
# 绘制BIC图

plot(summary\_regfit$bic, type = "l", xlab = "Number of Variables", ylab = "BIC", main = "BIC Criterion for Model Selection")

points(which.min(summary\_regfit$bic), summary\_regfit$bic[which.min(summary\_regfit$bic)], col = "blue", cex = 2, pch = 20)

abline(h = min(summary\_regfit$bic), col = "blue", lty = 2)

mtext(paste("Optimal model based on BIC has", which.min(summary\_regfit$bic), "variables"), side = 3, line = 0.5)



# 3. 调整R^2 (Adjusted R-squared)

# 调整R^2越大越好

which.max(summary\_regfit$adjr2)

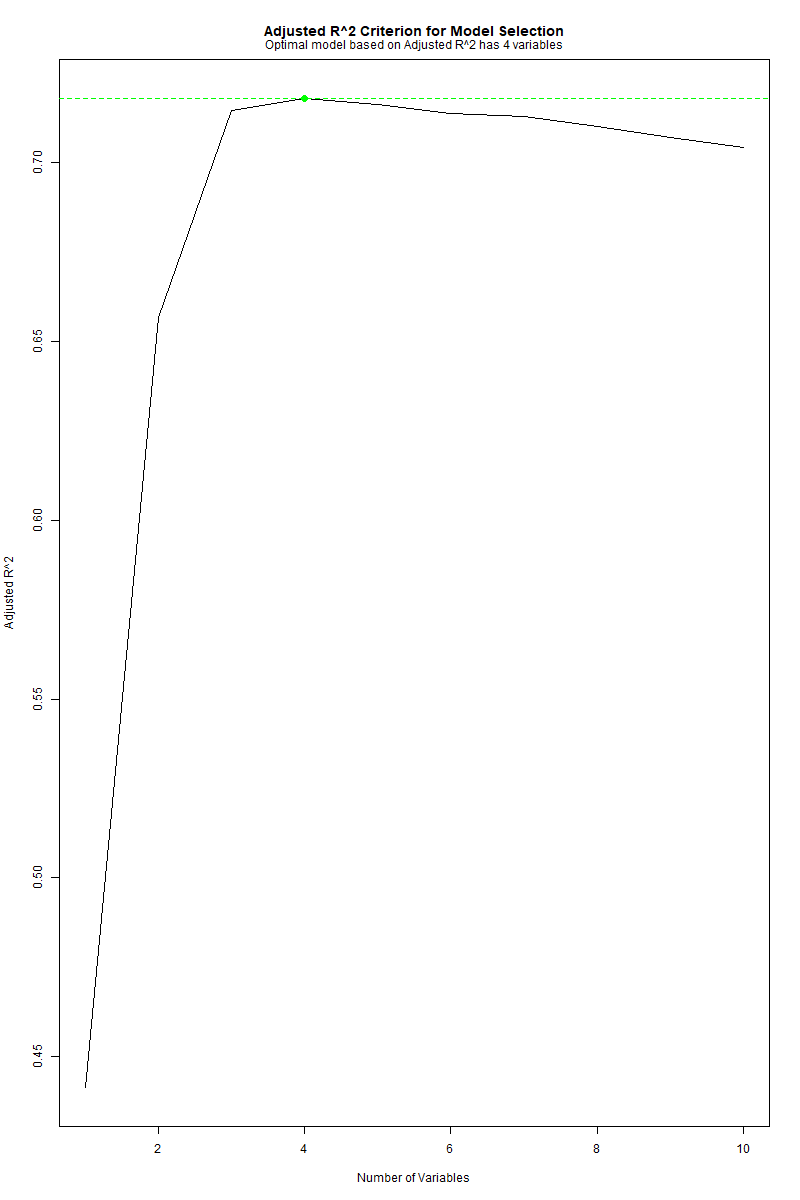
# 绘制调整R^2图

plot(summary\_regfit$adjr2, type = "l", xlab = "Number of Variables", ylab = "Adjusted R^2", main = "Adjusted R^2 Criterion for Model Selection")

points(which.max(summary\_regfit$adjr2), summary\_regfit$adjr2[which.max(summary\_regfit$adjr2)], col = "green", cex = 2, pch = 20)

abline(h = max(summary\_regfit$adjr2), col = "green", lty = 2)

mtext(paste("Optimal model based on Adjusted R^2 has", which.max(summary\_regfit$adjr2), "variables"), side = 3, line = 0.5)



# 获取基于Cp、BIC和Adjusted R^2选择的最优模型（假设都是3个变量）

# 例如，选择3个变量的模型

print(coef(regfit.full, 3))



# (d) 使用向前逐步选择法和向后逐步选择法重复 (c) 中的步骤。

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

summary\_regfit\_fwd <- summary(regfit.fwd)

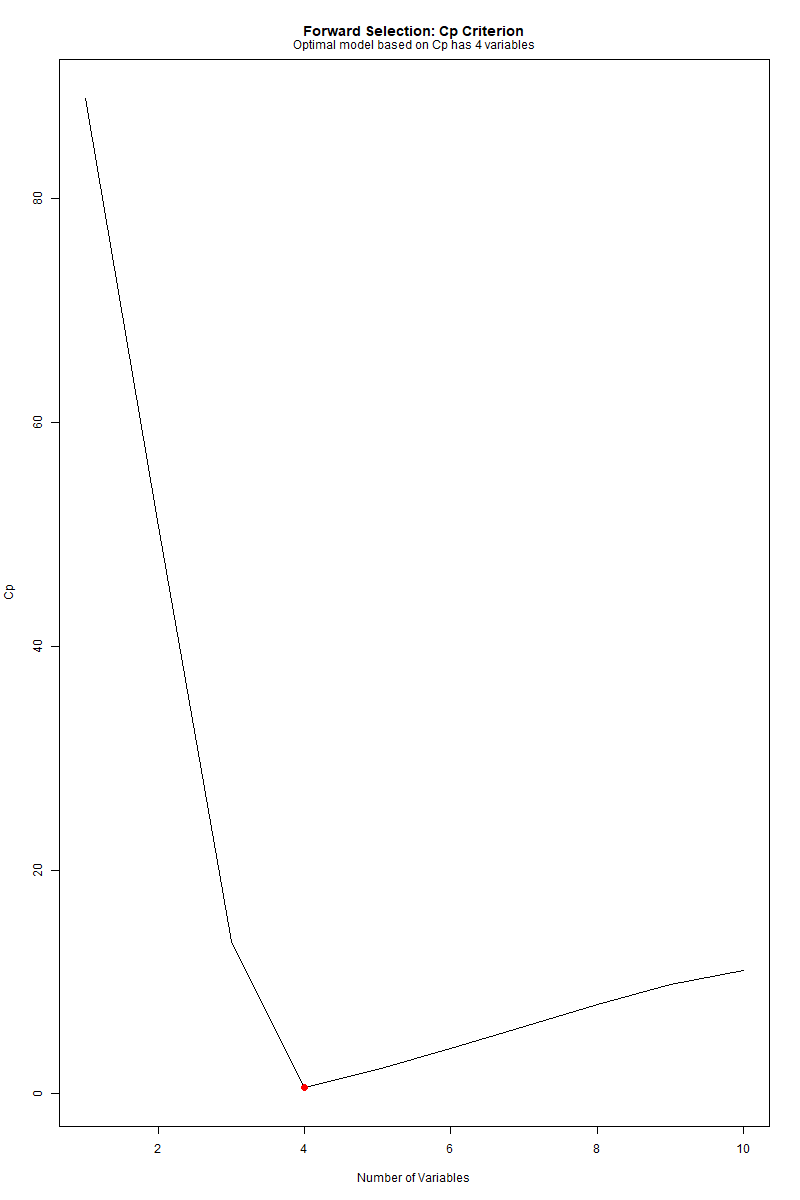
# 同样绘制Cp、BIC、调整R^2图并选择最优模型

# Cp for Forward

plot(summary\_regfit\_fwd$cp, type = "l", xlab = "Number of Variables", ylab = "Cp", main = "Forward Selection: Cp Criterion")

points(which.min(summary\_regfit\_fwd$cp), summary\_regfit\_fwd$cp[which.min(summary\_regfit\_fwd$cp)], col = "red", cex = 2, pch = 20)

mtext(paste("Optimal model based on Cp has", which.min(summary\_regfit\_fwd$cp), "variables"), side = 3, line = 0.5)

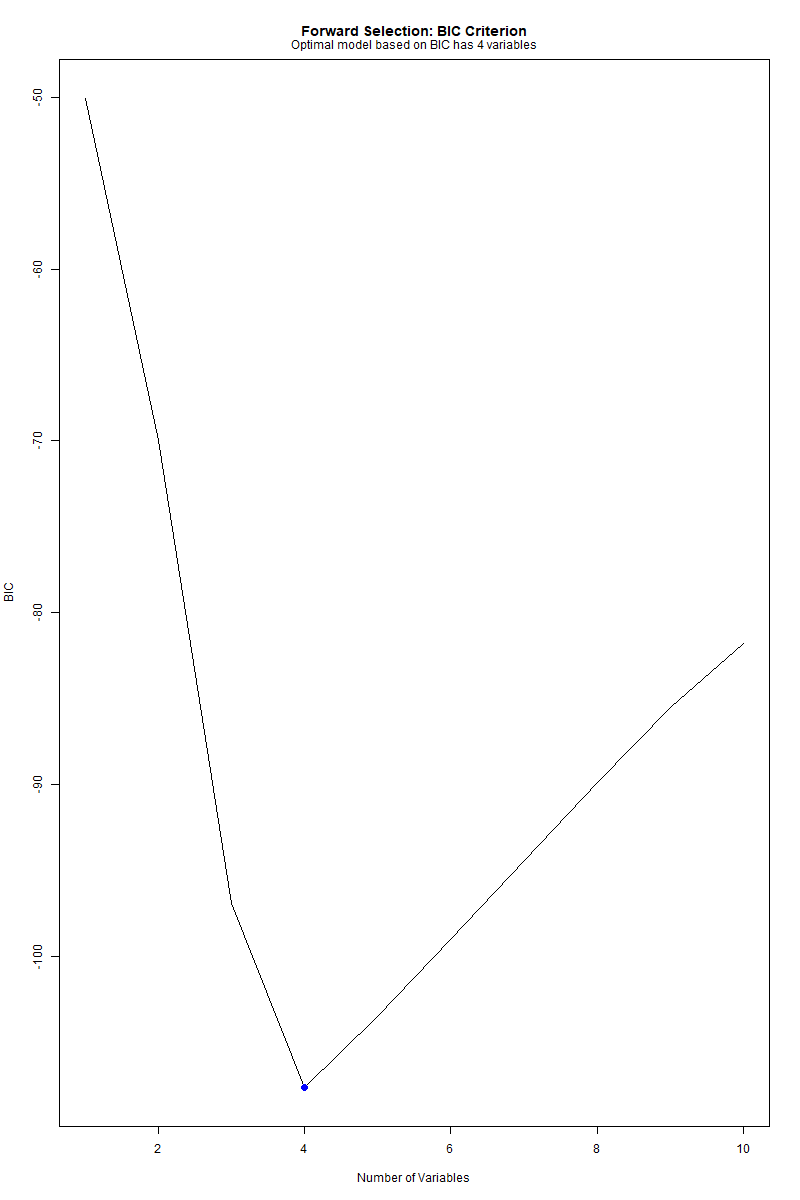


# BIC for Forward

plot(summary\_regfit\_fwd$bic, type = "l", xlab = "Number of Variables", ylab = "BIC", main = "Forward Selection: BIC Criterion")

points(which.min(summary\_regfit\_fwd$bic), summary\_regfit\_fwd$bic[which.min(summary\_regfit\_fwd$bic)], col = "blue", cex = 2, pch = 20)

mtext(paste("Optimal model based on BIC has", which.min(summary\_regfit\_fwd$bic), "variables"), side = 3, line = 0.5)

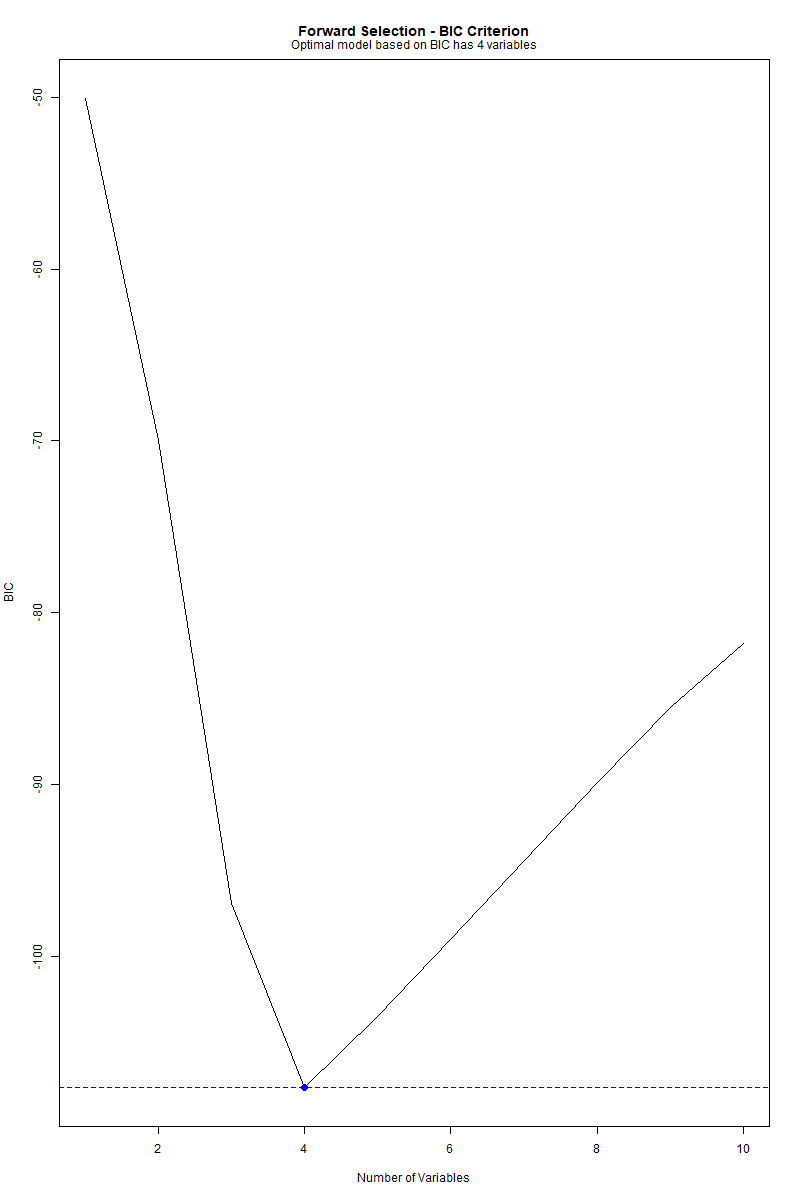


# Adjusted R^2 for Forward

plot(summary\_regfit\_fwd$adjr2, type = "l", xlab = "Number of Variables", ylab = "Adjusted R^2", main = "Forward Selection: Adjusted R^2 Criterion")

points(which.max(summary\_regfit\_fwd$adjr2), summary\_regfit\_fwd$adjr2[which.max(summary\_regfit\_fwd$adjr2)], col = "green", cex = 2, pch = 20)

mtext(paste("Optimal model based on Adjusted R^2 has", which.max(summary\_regfit\_fwd$adjr2), "variables"), side = 3, line = 0.5)



# 调整R^2

which.max(summary\_forward$adjr2)

plot(summary\_forward$adjr2, type = "l", xlab = "Number of Variables", ylab = "Adjusted R^2",

    main = "Forward Selection - Adjusted R^2 Criterion")

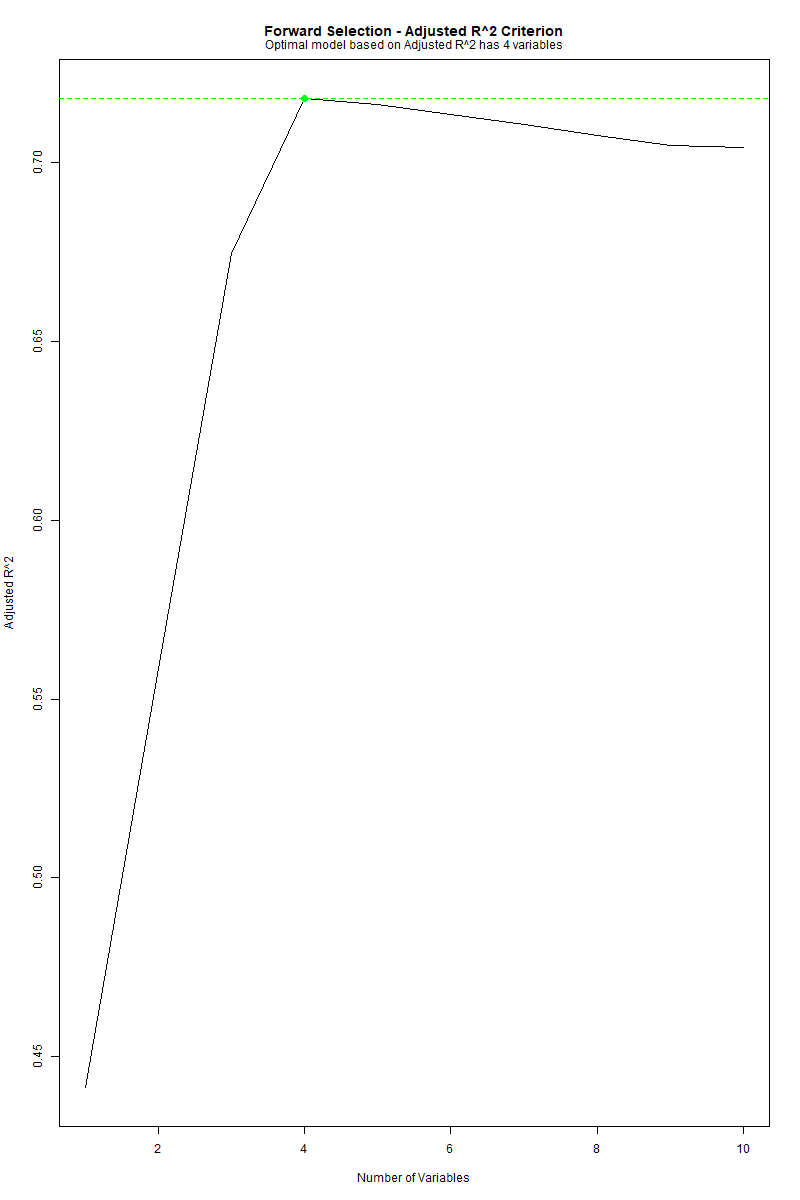
points(which.max(summary\_forward$adjr2), summary\_forward$adjr2[which.max(summary\_forward$adjr2)],

    col = "green", cex = 2, pch = 20)

abline(h = max(summary\_forward$adjr2), col = "green", lty = 2)

mtext(paste("Optimal model based on Adjusted R^2 has", which.max(summary\_forward$adjr2), "variables"),

    side = 3, line = 0.5)



# 输出最优模型系数（假设选择3个变量）

print(coef(regfit.forward, 3))



# 向后逐步选择 (Backward Selection)

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

summary\_backward <- summary(regfit.backward)

# 基于Cp、BIC和调整R^2选择最优模型

# Cp准则

which.min(summary\_backward$cp)

plot(summary\_backward$cp, type = "l", xlab = "Number of Variables", ylab = "Cp",

    main = "Backward Selection - Cp Criterion")

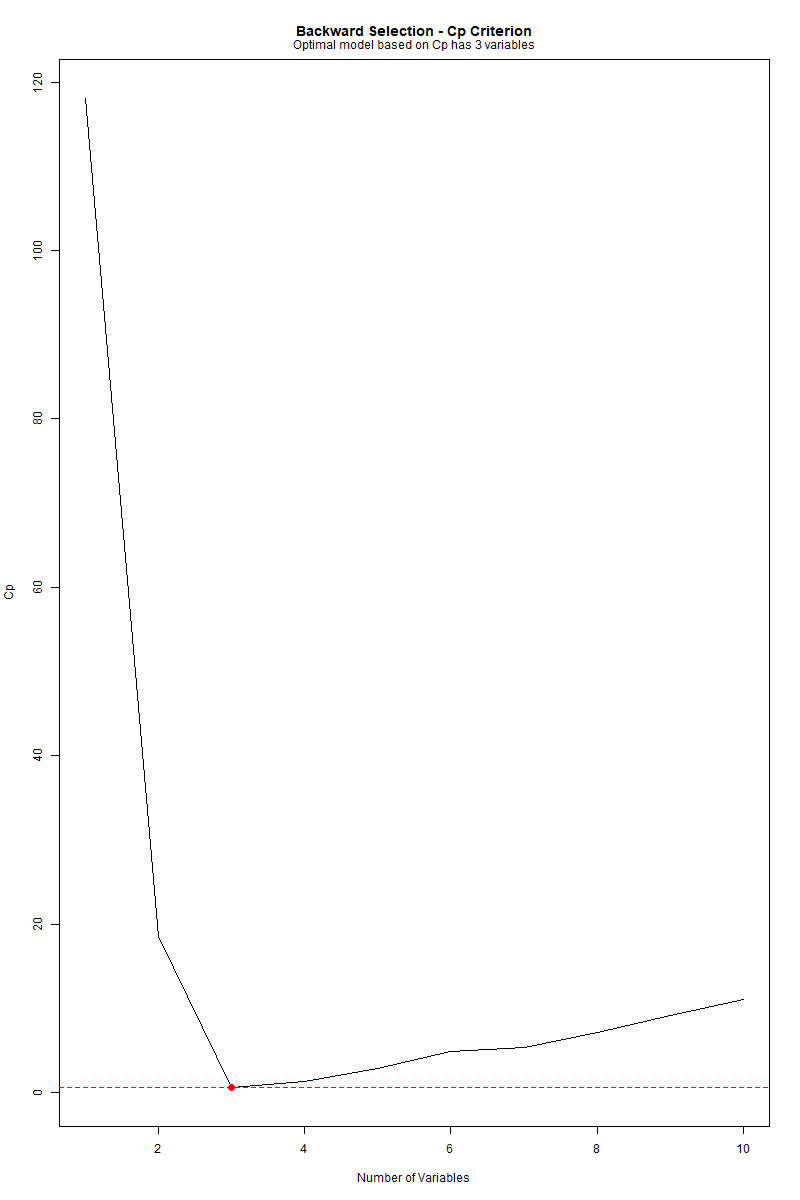
points(which.min(summary\_backward$cp), summary\_backward$cp[which.min(summary\_backward$cp)],

    col = "red", cex = 2, pch = 20)

abline(h = min(summary\_backward$cp), col = "red", lty = 2)

mtext(paste("Optimal model based on Cp has", which.min(summary\_backward$cp), "variables"),

    side = 3, line = 0.5)



# BIC准则

which.min(summary\_backward$bic)

plot(summary\_backward$bic, type = "l", xlab = "Number of Variables", ylab = "BIC",

    main = "Backward Selection - BIC Criterion")

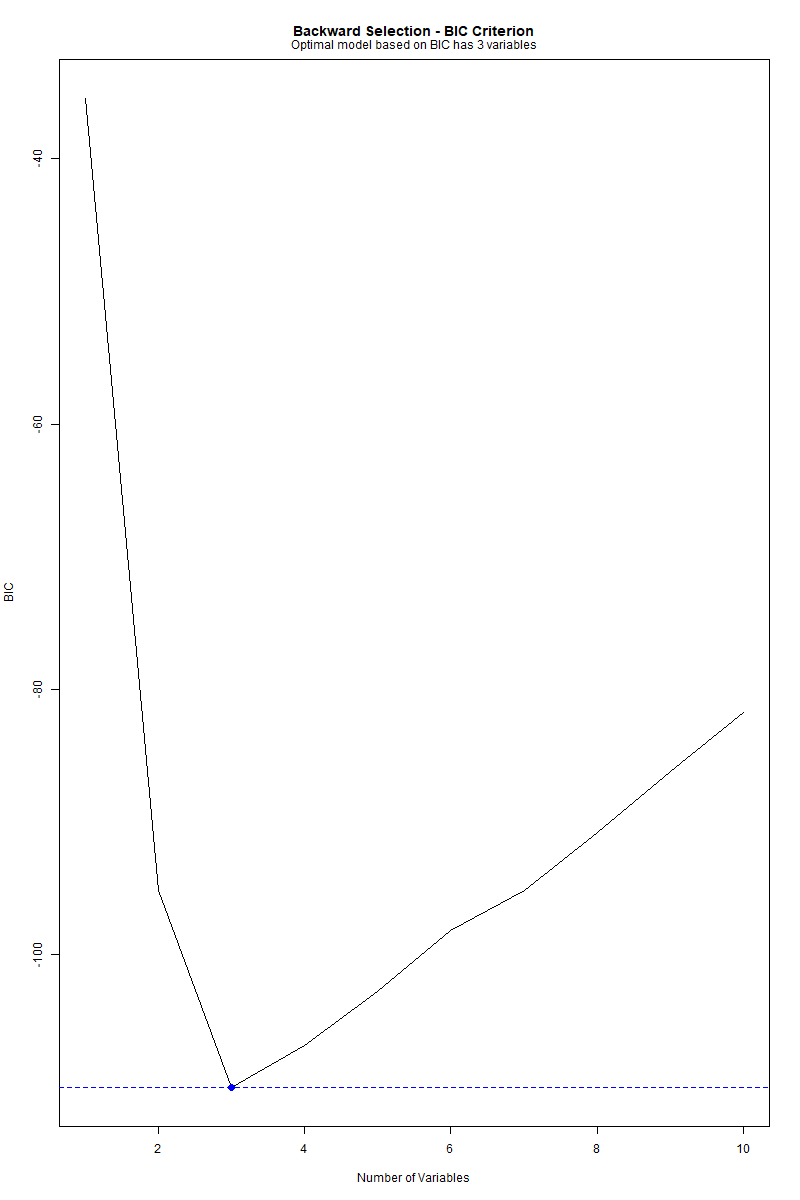
points(which.min(summary\_backward$bic), summary\_backward$bic[which.min(summary\_backward$bic)],

    col = "blue", cex = 2, pch = 20)

abline(h = min(summary\_backward$bic), col = "blue", lty = 2)

mtext(paste("Optimal model based on BIC has", which.min(summary\_backward$bic), "variables"),

    side = 3, line = 0.5)



# 调整R^2

which.max(summary\_backward$adjr2)

plot(summary\_backward$adjr2, type = "l", xlab = "Number of Variables", ylab = "Adjusted R^2",

    main = "Backward Selection - Adjusted R^2 Criterion")

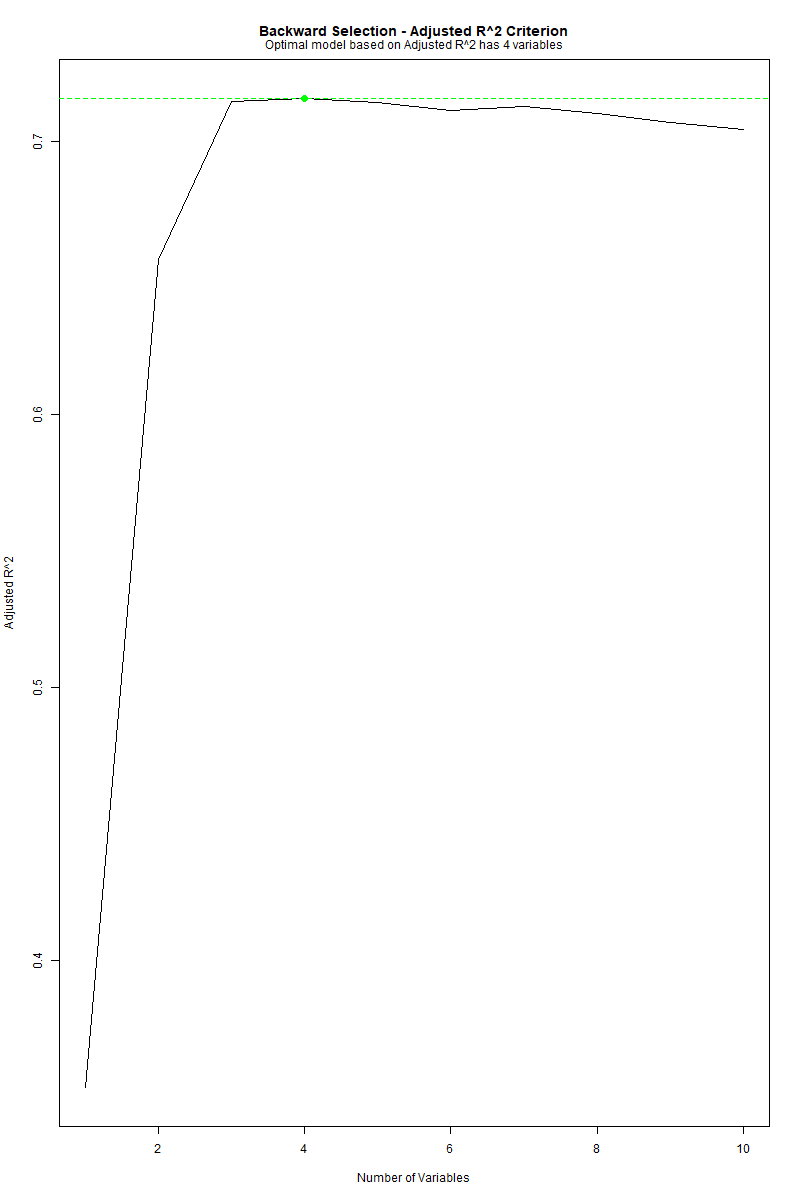
points(which.max(summary\_backward$adjr2), summary\_backward$adjr2[which.max(summary\_backward$adjr2)],

    col = "green", cex = 2, pch = 20)

abline(h = max(summary\_backward$adjr2), col = "green", lty = 2)

mtext(paste("Optimal model based on Adjusted R^2 has", which.max(summary\_backward$adjr2), "variables"),

    side = 3, line = 0.5)



# 输出最优模型系数（假设选择3个变量）

print(coef(regfit.backward, 3))



基本一致，只有系数有一定区别

# (e) lasso 拟合数据集

library(glmnet)

# 创建X的多项式项（包含X^1到X^10）

X\_poly <- model.matrix(Y ~ poly(X, 10, raw = TRUE))[,-1]  # 移除截距项

# 将X的多项式项转换为矩阵形式

X\_matrix <- as.matrix(X\_poly)

# 确保Y是向量

Y\_vector <- as.vector(Y)

# 执行交叉验证以选择最优的 lambda

cv\_lasso <- cv.glmnet(X\_matrix, Y\_vector, alpha = 1, nfolds = 10)

# 查看交叉验证结果

print(cv\_lasso)

# 绘制 CV 误差曲线

plot(cv\_lasso, main = "Cross-Validation Error vs Lambda for Lasso", xlab = "log(Lambda)", ylab = "CV Error")

abline(v = log(cv\_lasso$lambda.min), col = "red", lty = 2)

abline(v = log(cv\_lasso$lambda.1se), col = "blue", lty = 2)

mtext(paste("Optimal lambda (min):", round(cv\_lasso$lambda.min, 4)), side = 3, line = 0.5, col = "red")

mtext(paste("Lambda with 1SE:", round(cv\_lasso$lambda.1se, 4)), side = 3, line = 1.5, col = "blue")

# 使用最优 lambda 得到系数估计

lasso\_coef\_min <- coef(cv\_lasso, s = "lambda.min")

lasso\_coef\_1se <- coef(cv\_lasso, s = "lambda.1se")

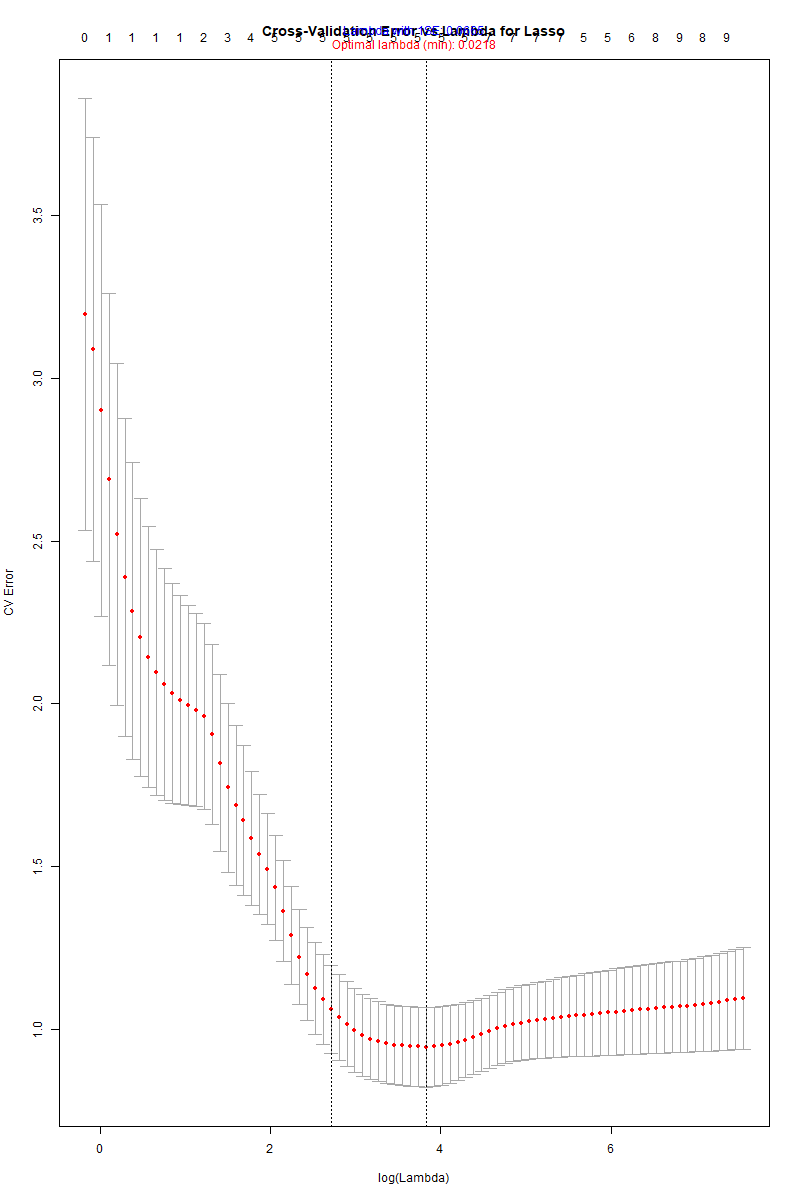
# 输出结果

print("Lasso Coefficients (lambda.min):")

print(lasso\_coef\_min)

print("\nLasso Coefficients (lambda.1se):")

print(lasso\_coef\_1se)



横轴：$\log(\lambda)$，表示正则化强度的对数值。

$\lambda$ 越小 → 正则化越弱 → 模型更复杂（可能过拟合）。

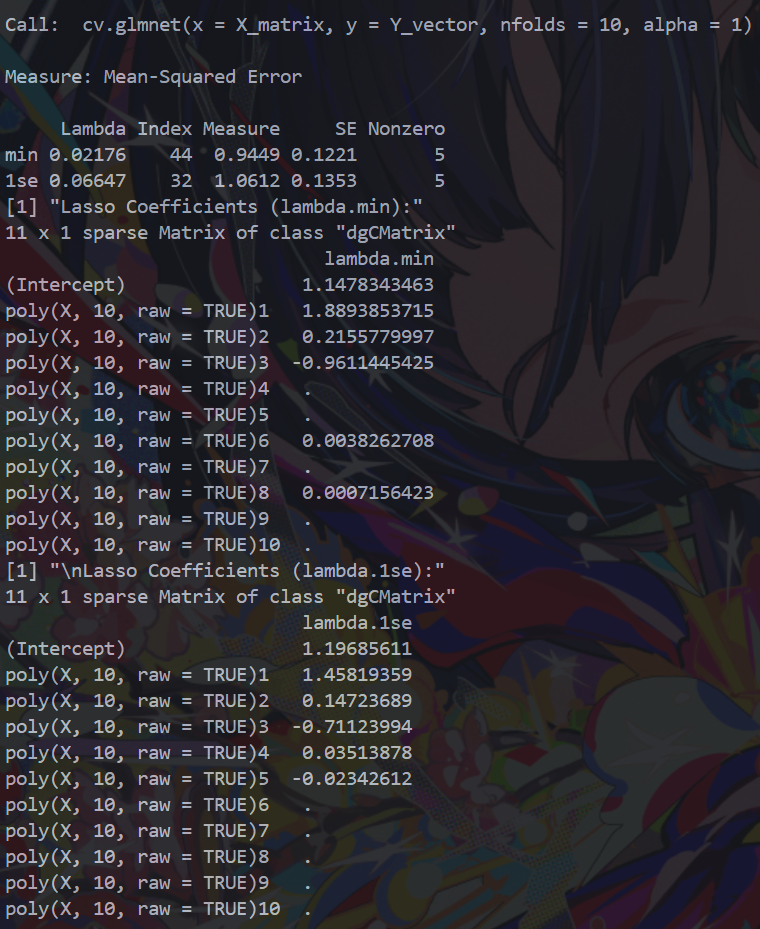
$\lambda$ 越大 → 正则化越强 → 模型更简单（可能欠拟合）。

纵轴：CV Error（交叉验证误差），即模型在验证集上的均方误差（MSE）。

误差越低 → 模型预测性能越好。

红色点：每个 $\lambda$ 对应的平均 CV 误差。

灰色竖线：表示每个 $\lambda$ 下 CV 误差的标准误（SE），反映估计的稳定性。



(1) lambda.min 对应的系数

仅前 3 项（X¹、X²、X³）和第 6、8 项有显著系数。

其他高阶项被压缩为零，说明它们对预测无贡献。

1. lambda.1se 对应的系数

更多高阶项被压缩为零，模型更加简洁。

主要保留了 X¹、X²、X³ 和 X⁴、X⁵ 的影响。

# (f) 依据新模型 Y = β0 + β1\*X^2 + ε 生成响应变量并进行模型选择

# 重新设置随机种子以确保结果可重现

set.seed(1)

n <- 100

X <- rnorm(n)

epsilon <- rnorm(n)

# 定义新的参数（仅含 X^2 项）

beta0 <- 1

beta1 <- 2

# 生成新的响应变量 Y

Y <- beta0 + beta1 \* X^2 + epsilon

# 创建X的多项式项（包含X^1到X^10）

X\_poly <- model.matrix(Y ~ poly(X, 10, raw = TRUE))[,-1]  # 移除截距项

colnames(X\_poly) <- paste0("X", 1:10)

# 将X的多项式项和Y组合成数据框

data\_df <- data.frame(Y, X\_poly)

# 使用最优子集选择法（regsubsets）

library(leaps)

regfit.full <- regsubsets(Y ~ ., data = data\_df, nvmax = 10)

# 查看选择结果的统计量

summary\_regfit <- summary(regfit.full)

# 根据Cp、BIC和调整R^2选择最优模型

# Cp准则

which.min(summary\_regfit$cp)

plot(summary\_regfit$cp, type = "l", xlab = "Number of Variables", ylab = "Cp",

     main = "Cp Criterion for Optimal Subset Selection")

points(which.min(summary\_regfit$cp), summary\_regfit$cp[which.min(summary\_regfit$cp)], col = "red", cex = 2, pch = 20)

abline(h = min(summary\_regfit$cp), col = "red", lty = 2)

mtext(paste("Optimal model based on Cp has", which.min(summary\_regfit$cp), "variables"), side = 3, line = 0.5)

# BIC准则

which.min(summary\_regfit$bic)

plot(summary\_regfit$bic, type = "l", xlab = "Number of Variables", ylab = "BIC",

     main = "BIC Criterion for Optimal Subset Selection")

points(which.min(summary\_regfit$bic), summary\_regfit$bic[which.min(summary\_regfit$bic)], col = "blue", cex = 2, pch = 20)

abline(h = min(summary\_regfit$bic), col = "blue", lty = 2)

mtext(paste("Optimal model based on BIC has", which.min(summary\_regfit$bic), "variables"), side = 3, line = 0.5)

# 调整R^2

which.max(summary\_regfit$adjr2)

plot(summary\_regfit$adjr2, type = "l", xlab = "Number of Variables", ylab = "Adjusted R^2",

     main = "Adjusted R^2 Criterion for Optimal Subset Selection")

points(which.max(summary\_regfit$adjr2), summary\_regfit$adjr2[which.max(summary\_regfit$adjr2)], col = "green", cex = 2, pch = 20)

abline(h = max(summary\_regfit$adjr2), col = "green", lty = 2)

mtext(paste("Optimal model based on Adjusted R^2 has", which.max(summary\_regfit$adjr2), "variables"), side = 3, line = 0.5)

# 输出最优模型系数（假设为2个变量：截距和X^2）

print("Optimal Subset Selection - Coefficients:")

print(coef(regfit.full, 2))

# 使用Lasso方法

library(glmnet)

X\_matrix <- as.matrix(X\_poly)

Y\_vector <- as.vector(Y)

# 执行交叉验证以选择最优的 lambda

cv\_lasso <- cv.glmnet(X\_matrix, Y\_vector, alpha = 1, nfolds = 10)

# 绘制CV误差曲线

plot(cv\_lasso, main = "Cross-Validation Error vs Lambda for Lasso", xlab = "log(Lambda)", ylab = "CV Error")

abline(v = log(cv\_lasso$lambda.min), col = "red", lty = 2)

abline(v = log(cv\_lasso$lambda.1se), col = "blue", lty = 2)

mtext(paste("Optimal lambda (min):", round(cv\_lasso$lambda.min, 4)), side = 3, line = 0.5, col = "red")

mtext(paste("Lambda with 1SE:", round(cv\_lasso$lambda.1se, 4)), side = 3, line = 1.5, col = "blue")

# 获取最优模型系数

lasso\_coef\_min <- coef(cv\_lasso, s = "lambda.min")

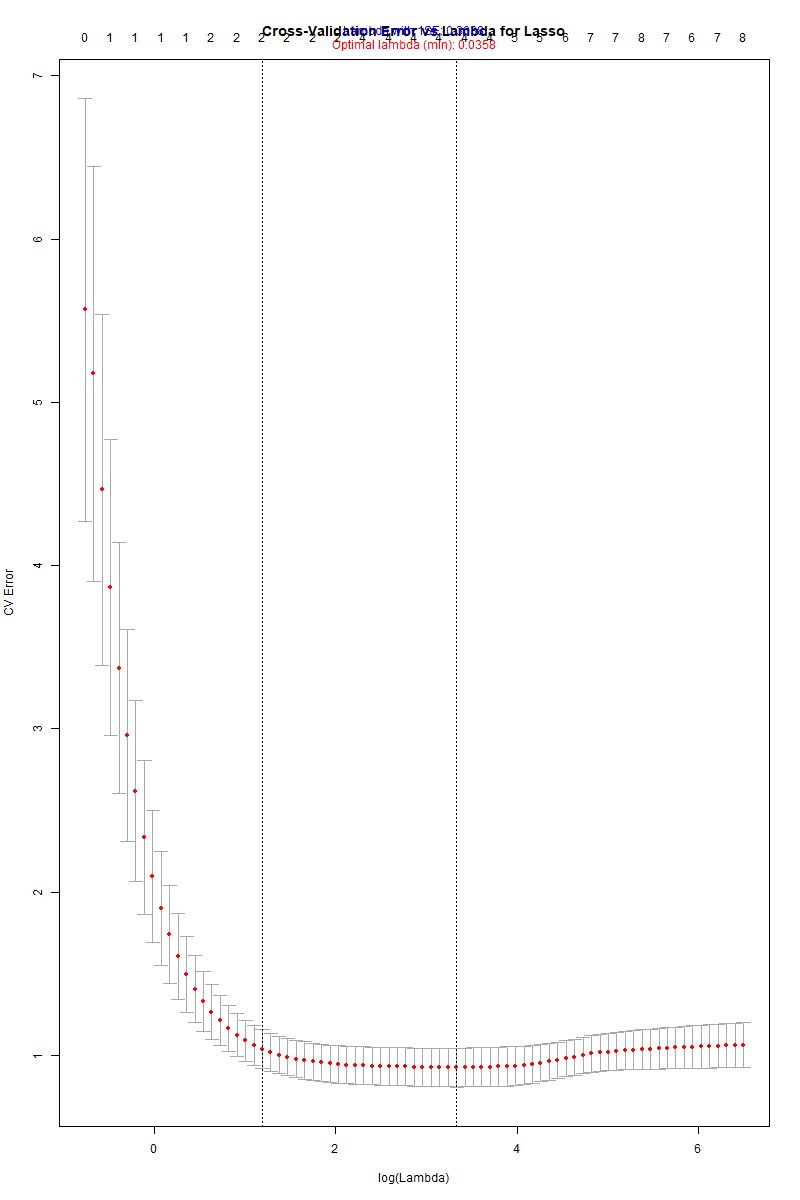
lasso\_coef\_1se <- coef(cv\_lasso, s = "lambda.1se")

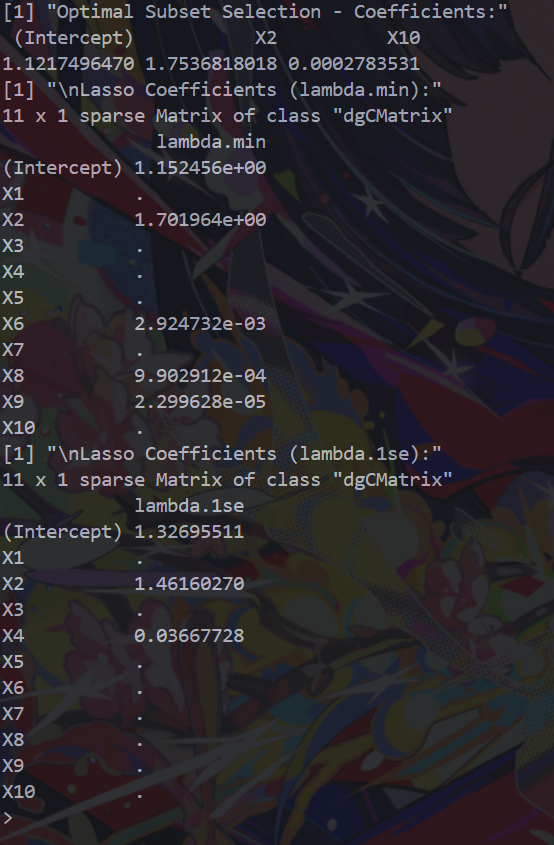
print("\nLasso Coefficients (lambda.min):")

print(lasso\_coef\_min)

print("\nLasso Coefficients (lambda.1se):")

print(lasso\_coef\_1se)





最优子集法

截距项 (Intercept)：约为 1.1217，接近真实值 1。

X2（即 $X^2$）：系数为 1.7537，接近真实值 2。

X10（即 $X^{10}$）：系数极小（0.000278），说明其影响可忽略。

结论：最优子集选择法成功识别出主要变量 $X^2$，但误将 $X^{10}$ 纳入模型，可能是由于随机噪声导致的偶然相关。

Min

X2（即 $X^2$）：系数为 1.701964，非常接近真实值 2。

其他高阶项（如 $X^6, X^8$）有微小非零系数，但数值极小，可视为噪声。

多数项被压缩为零，体现了 Lasso 的稀疏性优势。

1se

X2：系数为 1.4616，略低于真实值，但仍显著。

X4：出现一个小的非零系数（0.0367），可能由噪声引起。

其余项均为零，模型更加简洁。