library(readxl)

library(glmnet)

data <- read\_xlsx("C:/Users/1/Desktop/ran\_date.xlsx", sheet = 1)

data\_N <- subset(data, nutrition\_fertilizer == "N")

data\_Y <- subset(data, nutrition\_fertilizer == "Y")

data <- na.omit(data)

# 标准化数据

scaled\_data <- scale(data[, c("BIO","SLA","plant\_height\_cm","leaf\_area\_cm","area\_top\_view\_cm","red\_top\_view")])

data <- as.data.frame(scaled\_data)

# 初始化存储精度的空数据框

results <- data.frame(n\_train = integer(),

lm\_accuracy = numeric(),

quad\_accuracy = numeric(),

LASSO\_accuracy = numeric(),

RIDGE\_accuracy = numeric())

for (k in c(30, 50, 100, 150, 200, 250)) { # 增加 k 的最小值

# 存储每种模型在该训练样本数量下的精度

lm\_accuracy <- numeric(100)

quad\_accuracy <- numeric(100)

LASSO\_accuracy <- numeric(100)

RIDGE\_accuracy <- numeric(100)

set.seed(2) # 放在外层循环中以保持一致性

for (n in 1:100) {

# 随机划分训练集和测试集

train\_index <- sample(seq\_len(nrow(data)), size = k)

train <- data[train\_index, ]

test <- data[-train\_index, ]

# 1. 线性回归

model\_lm <- lm(BIO ~ SLA+plant\_height\_cm+leaf\_area\_cm+area\_top\_view\_cm+red\_top\_view, data = train) # 将公式省略

test$predicted\_lm <- predict(model\_lm, newdata = test)

lm\_accuracy[n] <- (cor(test$BIO, test$predicted\_lm)^2)

# 2. 二次回归

model\_quad <- lm(BIO ~ poly(SLA, 2) +poly(plant\_height\_cm, 2)+poly(leaf\_area\_cm, 2)+poly(area\_top\_view\_cm,2)+poly(red\_top\_view,2), data = train) # 将公式省略

test$predicted\_quad <- predict(model\_quad, newdata = test)

quad\_accuracy[n] <- (cor(test$BIO, test$predicted\_quad)^2)

# 3. LASSO回归

model\_lasso <- cv.glmnet(x = as.matrix(train[, c("SLA","plant\_height\_cm","leaf\_area\_cm","area\_top\_view\_cm","red\_top\_view")]),

y = train$BIO,

alpha = 1, nfolds = 3) # 设置 nfolds=3

best\_lambda\_lasso <- model\_lasso$lambda.min

test$predicted\_lasso <- predict(model\_lasso, s = best\_lambda\_lasso,

newx = as.matrix(test[, c("SLA","plant\_height\_cm","leaf\_area\_cm","area\_top\_view\_cm","red\_top\_view")]))

LASSO\_accuracy[n] <- (cor(test$BIO, test$predicted\_lasso)^2)

# 4.RIDGE回归

model\_ridge <- cv.glmnet(x = as.matrix(train[, c("SLA","plant\_height\_cm","leaf\_area\_cm","area\_top\_view\_cm","red\_top\_view")]),

y = train$BIO,

alpha = 0, nfolds = 3) # 设置 nfolds=3

best\_lambda\_ridge <- model\_ridge$lambda.min

test$predicted\_ridge <- predict(model\_ridge, s = best\_lambda\_ridge,

newx = as.matrix(test[, c("SLA","plant\_height\_cm","leaf\_area\_cm","area\_top\_view\_cm","red\_top\_view")]))

RIDGE\_accuracy[n] <- (cor(test$BIO, test$predicted\_ridge)^2)

}

# 计算平均预测精度

results <- rbind(results, data.frame(

n\_train = k,

lm\_accuracy = mean(lm\_accuracy, na.rm = TRUE),

quad\_accuracy = mean(quad\_accuracy, na.rm = TRUE),

LASSO\_accuracy = mean(LASSO\_accuracy, na.rm = TRUE),

RIDGE\_accuracy = mean(RIDGE\_accuracy, na.rm = TRUE)

))

}

# 检查多重共线性

library(car)

vif(model\_lm)

vif(model\_quad)

# 查看结果

print(results)

warnings()

library(ggplot2)

ggplot(results, aes(x = n\_train)) +

geom\_line(aes(y = lm\_accuracy, color = "lm"), linewidth = 1) +

geom\_line(aes(y = quad\_accuracy, color = "quadratic"), linewidth = 1) +

geom\_line(aes(y = LASSO\_accuracy, color = "LASSO"), linewidth = 1) +

geom\_line(aes(y = RIDGE\_accuracy, color = "RIDGE"), linewidth = 1) +

geom\_point(aes(y = lm\_accuracy, color = "lm"), size = 2) +

geom\_point(aes(y = quad\_accuracy, color = "quadratic"), size = 2) +

geom\_point(aes(y = LASSO\_accuracy, color = "LASSO"), size = 2) +

geom\_point(aes(y = RIDGE\_accuracy, color = "RIDGE"), size = 2) +

labs(title = "Prediction accuracy (R²) vs. Number of training individuals",

x = "Number of training individuals",

y = "Prediction accuracy (R²)") +

scale\_color\_manual(values = c("lm" = "blue", "quadratic" = "yellow",

"LASSO" = "green", "RIDGE" = "orange")) +

theme\_minimal()