```
# Generación de los datos
# set.seed(789)                    # Si queremos replicar resultados
X <- matrix(rnorm(2000), ncol=40)    # Matrix de variables "ruido"
buena <- rnorm(50)
buena2 <- rnorm(50)
#buena3 <- rnorm(50)
Y <- 1.2*buena+0.4*buena2+rnorm(50)  # Variable respuesta a partir de buena y buena2
prueba <- data.frame(X, buena, buena2, Y)   # Data frame con todas las variables

# Modelo lineal con selección stepwise
lm1 <- lm(Y ~., data=prueba)    # Ajuste del modelo
lm1.1 <- step(lm1)              # Algoritmo stepwise
# summary(lm1.1)
# plot(lm1.1)
summary(lm1.1)$coef[,4]         # Variables seleccionadas

# Screening de variables (Aplica un modelo lineal a cada variable predictora por separado)
p.values <- sapply(prueba[,-43], function(x) summary(lm(prueba$Y ~ x))$coef[2,4])
names(p.values) [p.values<0.05]   # P-valores sin ajustar
adj.pvalues <- p.adjust(p.values, method="fdr")   # Corrección comparaciones múltiples
names(adj.pvalues)[adj.pvalues<0.05]   # Variables seleccionadas

# Modelo de proyección PLS
library(mixOmics)
pls.fit <- pls(prueba[,-43], prueba$Y, ncomp=2, near.zero.var = TRUE)  # Ajuste con dos
componentes
pls.sel <- vip(pls.fit)    # Importancia de las variables
dotplot(pls.sel)           # Gráfica de selección de variables

# Modelo de regresión con penalización L1 (LASSO)
library(glmnet)
cv <- cv.glmnet(as.matrix(prueba[,-43]), prueba$Y, alpha=1, nfolds=10, family="gaussian",
penalty.factor = rep(1, 42))
plot(cv)                   # Gráfica de validación cruzada para selección de lambda
l <- cv$lambda.1se
lasso.fit <- glmnet(as.matrix(prueba[,-43]), prueba$Y)   # Ajuste del modelo
res <- predict(lasso.fit, s=l, type="coef")
res              # Variables seleccionadas

# Modelo de Random Forest
library(randomForest)
rf.fit <- randomForest(Y ~., data=prueba, ntree=500, mtry=14)   # Ajuste del modelo
plot(rf.fit)
varImpPlot(rf.fit)  # Gráfica de importancia de las variables

# Mejora de LASSO con Relaxed Lasso:
# library(relaxo)

# rlasso.fit <- cvrelaxo(as.matrix(prueba[,-43]), prueba$Y)
# rlasso.fit$beta
```