```r
# Load necessary libraries
library(glmnet)
library(caret)
library(dplyr)

# Load the Hitters dataset (assuming you have it)
data("Hitters")
# Ensure there are no missing values
Hitters <- na.omit(Hitters)

# Select predictor variables and response variable
predictors <- c("CAtBat", "CHits", "CHmRun", "CRuns", "CRBI", "CWalks", "League",
"Division", "PutOuts", "Assists", "Errors", "NewLeague")
response <- "Salary"

# Encode categorical variables (League, Division, NewLeague)
Hitters$League <- as.numeric(as.factor(Hitters$League))
Hitters$Division <- as.numeric(as.factor(Hitters$Division))
Hitters$NewLeague <- as.numeric(as.factor(Hitters$NewLeague))

# Split the data into training and testing sets
set.seed(123) # for reproducibility
trainIndex <- createDataPartition(Hitters[, response], p = 0.8,
                                  list = FALSE,
                                  times = 1)
trainData <- Hitters[ trainIndex,]
testData  <- Hitters[-trainIndex,]

# Fit Ridge Regression model with lambda = 1
alpha <- 0 # Ridge regression
lambda <- 1
ridge_model <- glmnet(x = as.matrix(trainData[, predictors]),
                      y = trainData[, response],
                      alpha = alpha,
                      lambda = lambda)

# Predict values on the test set
ridge_predictions <- predict(ridge_model,
                             newx = as.matrix(testData[, predictors]))

# Plot the graph of predicted values
plot(testData[, response], ridge_predictions,
     xlab = "Actual Salary", ylab = "Predicted Salary",
     main = "Ridge Regression: Actual vs. Predicted Salary", col=c("red","blue"))

# Show the predicted values
head(data.frame(Actual = testData[, response], Predicted = ridge_predictions))

summary(ridge_model)
summary(ridge_predictions)

# Calculate the difference between actual and predicted values
difference <- testData[, response] - ridge_predictions

# Calculate the Mean Absolute Error (MAE)
mae <- mean(abs(difference))

# Calculate the Mean Squared Error (MSE)
mse <- mean(difference^2)

# Calculate the Root Mean Squared Error (RMSE)
rmse <- sqrt(mse)

# Print the accuracy metrics
```

```
cat("Mean Absolute Error (MAE):", mae, "\n")
cat("Mean Squared Error (MSE):", mse, "\n")
cat("Root Mean Squared Error (RMSE):", rmse, "\n")

# Calculate the Mean Absolute Percentage Error (MAPE)
mape <- mean(abs(testData[, response] - ridge_predictions) / testData[, response]) * 100

cat("Mean Absolute Percentage Error (MAPE):", mape, "%\n")
```

Output File

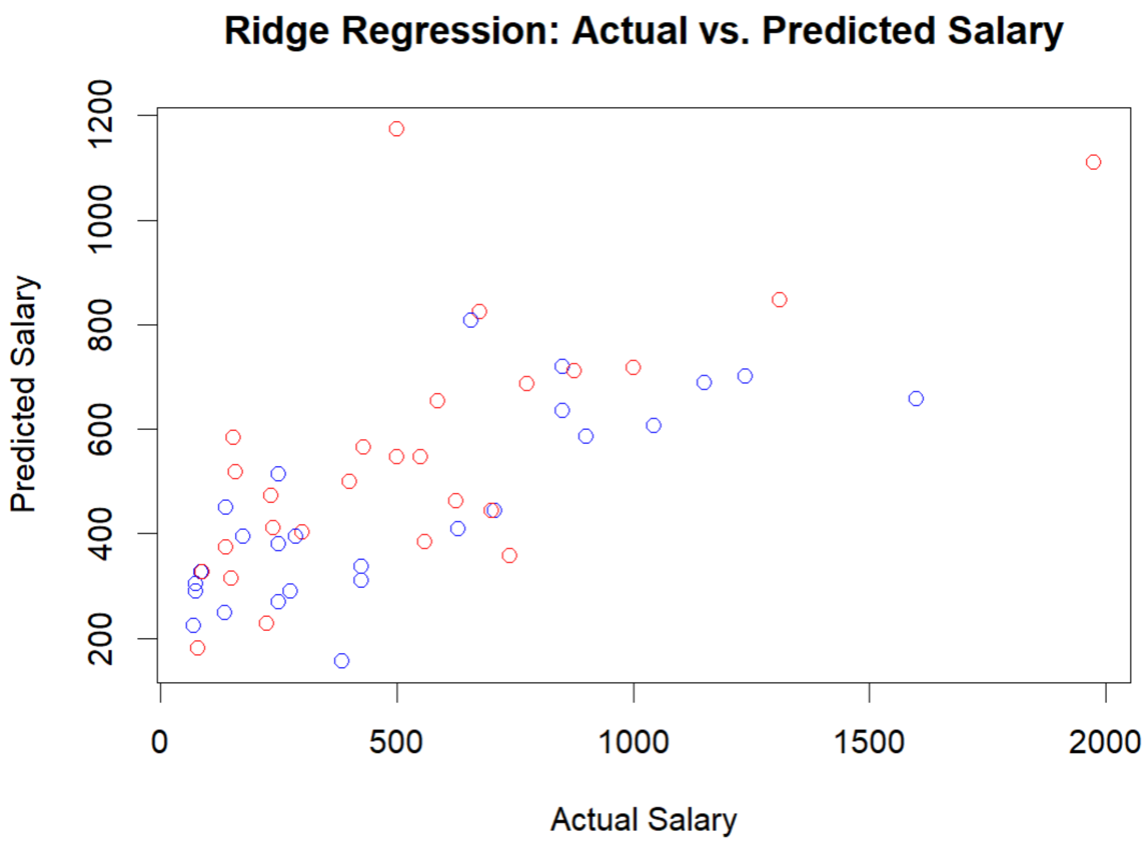
```
> # Load necessary libraries
> library(glmnet)
> library(caret)
> library(dplyr)
>
> # Load the Hitters dataset (assuming you have it)
> data("Hitters")
> # Ensure there are no missing values
> Hitters <- na.omit(Hitters)
>
> # Select predictor variables and response variable
> predictors <- c("CAtBat", "CHits", "CHmRun", "CRuns", "CRBI", "CWalks", "League", "
Division", "PutOuts", "Assists", "Errors", "NewLeague")
> response <- "Salary"
> # Encode categorical variables (League, Division, NewLeague)
> Hitters$League <- as.numeric(as.factor(Hitters$League))
> Hitters$Division <- as.numeric(as.factor(Hitters$Division))
> Hitters$NewLeague <- as.numeric(as.factor(Hitters$NewLeague))
>
> # Split the data into training and testing sets
> set.seed(123) # for reproducibility
> trainIndex <- createDataPartition(Hitters[, response], p = 0.8,
+                                   list = FALSE,
+                                   times = 1)
> trainData <- Hitters[ trainIndex,]
> testData  <- Hitters[-trainIndex,]
> # Fit Ridge Regression model with lambda = 1
> alpha <- 0 # Ridge regression
> lambda <- 1
> ridge_model <- glmnet(x = as.matrix(trainData[, predictors]),
+                       y = trainData[, response],
+                       alpha = alpha,
+                       lambda = lambda)
> # Predict values on the test set
> ridge_predictions <- predict(ridge_model,
+                              newx = as.matrix(testData[, predictors]))
>
> # Plot the graph of predicted values
> plot(testData[, response], ridge_predictions,
+      xlab = "Actual Salary", ylab = "Predicted Salary",
+      main = "Ridge Regression: Actual vs. Predicted Salary", col=c("red","blue"))
> # Show the predicted values
> head(data.frame(Actual = testData[, response], Predicted = ridge_predictions))
                 Actual        s0
-Andre Dawson   500.000 1173.9601
-Andres Thomas   75.000  304.7835
-Bill Almon     240.000  411.1497
-Bob Dernier    708.333  443.6684
-Bill Doran     625.000  461.8411
-Brian Downing  900.000  585.1582
>
> summary(ridge_model)
          Length Class     Mode
a0         1     -none-    numeric
beta      12     dgCMatrix S4
df         1     -none-    numeric
dim        2     -none-    numeric
lambda     1     -none-    numeric
dev.ratio  1     -none-    numeric
nulldev    1     -none-    numeric
npasses    1     -none-    numeric
jerr       1     -none-    numeric
offset     1     -none-    logical
call       5     -none-    call
nobs       1     -none-    numeric
> summary(ridge_predictions)
```

```
        s0
 Min.   : 155.8
 1st Qu.: 331.6
 Median : 449.3
 Mean   : 499.6
 3rd Qu.: 645.1
 Max.   :1174.0
>
> # Calculate the difference between actual and predicted values
> difference <- testData[, response] - ridge_predictions
>
> # Calculate the Mean Absolute Error (MAE)
> mae <- mean(abs(difference))
>
> # Calculate the Mean Squared Error (MSE)
> mse <- mean(difference^2)
> # Calculate the Root Mean Squared Error (RMSE)
> rmse <- sqrt(mse)
>
> # Print the accuracy metrics
> cat("Mean Absolute Error (MAE):", mae, "\n")
Mean Absolute Error (MAE): 238.6258
> cat("Mean Squared Error (MSE):", mse, "\n")
Mean Squared Error (MSE): 94171.06
> cat("Root Mean Squared Error (RMSE):", rmse, "\n")
Root Mean Squared Error (RMSE): 306.873
>
> # Calculate the Mean Absolute Percentage Error (MAPE)
> mape <- mean(abs(testData[, response] - ridge_predictions) / testData[, response])
* 100
>
> cat("Mean Absolute Percentage Error (MAPE):", mape, "%\n")
Mean Absolute Percentage Error (MAPE): 79.90111 %

>
```

Graph on next page…

Graph



Interpretation

A high error signifies that this ridge model is unsuitable for the Hitter dataset. It cannot capture the underlying pattern—insufficient data, multicollinearity, incorrect feature selection, etc. are some of the reasons!