Stats1\_Project

Sandeep Daddala

2024-11-13

getwd()

## [1] "/Users/sandeepdaddala/Desktop/Stats1/AdvStats1\_HW4"

library(ggplot2)  
library(corrplot)

## corrplot 0.94 loaded

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(readr)  
library(gridExtra)  
library(caret)

## Loading required package: lattice

library(car)

## Loading required package: carData

library(leaps)  
  
# Read the data and remove Cooling Load (Y2)  
building\_data <- read\_csv("/Users/sandeepdaddala/Desktop/Stats1/ENB2012\_data.csv")

## Rows: 768 Columns: 10

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (10): X1, X2, X3, X4, X5, X6, X7, X8, Y1, Y2  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

building\_data\_no\_y2 <- building\_data[, !names(building\_data) %in% c("Y2")]  
  
# Rename variables for clarity  
names(building\_data\_no\_y2) <- c("Relative\_Compactness", "Surface\_Area", "Wall\_Area", "Roof\_Area",  
 "Overall\_Height", "Orientation", "Glazing\_Area",   
 "Glazing\_Area\_Distribution", "Heating\_Load")  
  
# Summary of a linear model for checking multicollinearity  
summary(lm(Heating\_Load ~ ., data = building\_data\_no\_y2))

##   
## Call:  
## lm(formula = Heating\_Load ~ ., data = building\_data\_no\_y2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.8965 -1.3196 -0.0252 1.3532 7.7052   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 84.014521 19.033607 4.414 1.16e-05 \*\*\*  
## Relative\_Compactness -64.773991 10.289445 -6.295 5.19e-10 \*\*\*  
## Surface\_Area -0.087290 0.017075 -5.112 4.04e-07 \*\*\*  
## Wall\_Area 0.060813 0.006648 9.148 < 2e-16 \*\*\*  
## Roof\_Area NA NA NA NA   
## Overall\_Height 4.169939 0.337990 12.337 < 2e-16 \*\*\*  
## Orientation -0.023328 0.094705 -0.246 0.80550   
## Glazing\_Area 19.932680 0.813986 24.488 < 2e-16 \*\*\*  
## Glazing\_Area\_Distribution 0.203772 0.069918 2.914 0.00367 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.934 on 760 degrees of freedom  
## Multiple R-squared: 0.9162, Adjusted R-squared: 0.9154   
## F-statistic: 1187 on 7 and 760 DF, p-value: < 2.2e-16

# Remove Surface Area and Roof Area due to high correlation  
building\_data\_reduced <- building\_data\_no\_y2[, !names(building\_data\_no\_y2) %in% c("Surface\_Area", "Roof\_Area")]  
  
# Check for NULL values and duplicates  
sum(is.na(building\_data\_reduced))

## [1] 0

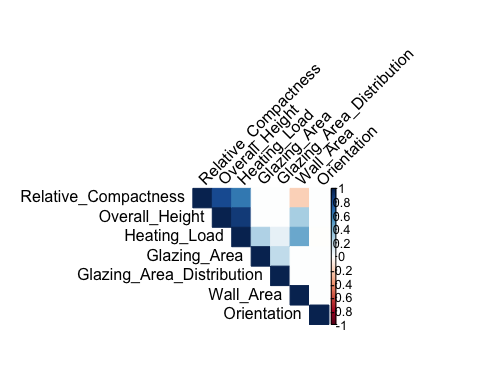
colSums(is.na(building\_data\_reduced))

## Relative\_Compactness Wall\_Area Overall\_Height   
## 0 0 0   
## Orientation Glazing\_Area Glazing\_Area\_Distribution   
## 0 0 0   
## Heating\_Load   
## 0

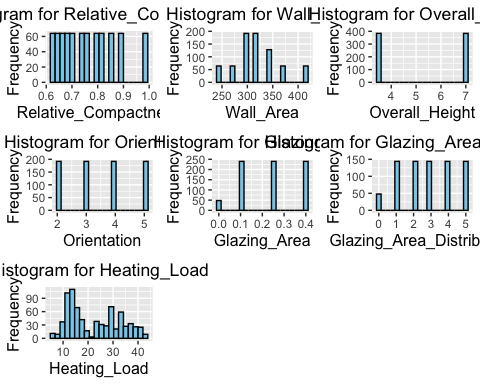
duplicate\_rows <- sum(duplicated(building\_data\_reduced))  
summary(building\_data\_reduced)

## Relative\_Compactness Wall\_Area Overall\_Height Orientation   
## Min. :0.6200 Min. :245.0 Min. :3.50 Min. :2.00   
## 1st Qu.:0.6825 1st Qu.:294.0 1st Qu.:3.50 1st Qu.:2.75   
## Median :0.7500 Median :318.5 Median :5.25 Median :3.50   
## Mean :0.7642 Mean :318.5 Mean :5.25 Mean :3.50   
## 3rd Qu.:0.8300 3rd Qu.:343.0 3rd Qu.:7.00 3rd Qu.:4.25   
## Max. :0.9800 Max. :416.5 Max. :7.00 Max. :5.00   
## Glazing\_Area Glazing\_Area\_Distribution Heating\_Load   
## Min. :0.0000 Min. :0.000 Min. : 6.01   
## 1st Qu.:0.1000 1st Qu.:1.750 1st Qu.:12.99   
## Median :0.2500 Median :3.000 Median :18.95   
## Mean :0.2344 Mean :2.812 Mean :22.31   
## 3rd Qu.:0.4000 3rd Qu.:4.000 3rd Qu.:31.67   
## Max. :0.4000 Max. :5.000 Max. :43.10

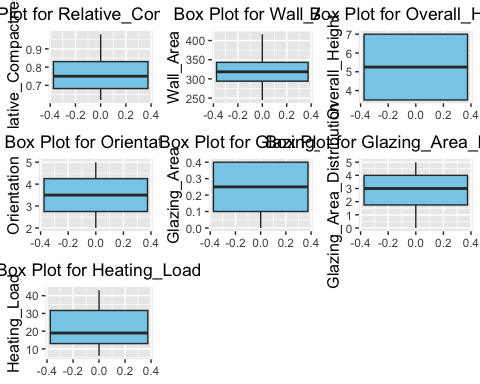
# 1. Correlation Heatmap  
cor\_matrix <- cor(building\_data\_reduced)  
corrplot(cor\_matrix, method = "color", type = "upper", order = "hclust",   
 tl.col = "black", tl.srt = 45)



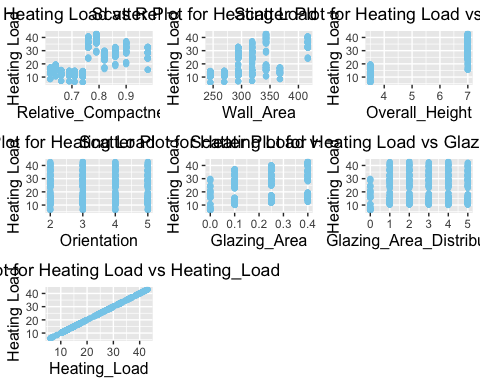
# 2. Histograms  
gg\_list\_hist <- lapply(names(building\_data\_reduced), function(var) {  
 ggplot(building\_data\_reduced, aes(x = .data[[var]])) +  
 geom\_histogram(fill = "skyblue", color = "black", bins = 20) +  
 labs(title = paste("Histogram for", var),  
 x = var, y = "Frequency") +  
 theme(plot.title = element\_text(hjust = 0.5),  
 axis.title.x = element\_text(size = 12),  
 axis.title.y = element\_text(size = 12))  
})  
grid.arrange(grobs = gg\_list\_hist, ncol = 3)



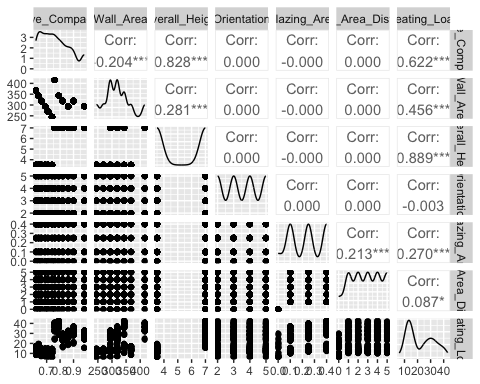
# Box Plot  
gg\_list\_box <- lapply(names(building\_data\_reduced), function(var) {  
 ggplot(building\_data\_reduced, aes(y = .data[[var]])) +  
 geom\_boxplot(fill = "skyblue") +  
 labs(title = paste("Box Plot for", var),  
 y = var) +  
 theme(plot.title = element\_text(hjust = 0.5),  
 axis.title.x = element\_text(size = 12),  
 axis.title.y = element\_text(size = 12))  
})  
grid.arrange(grobs = gg\_list\_box, ncol = 3)



# 4. Scatter plots for Heating Load (Y1)  
scatter\_y1 <- lapply(names(building\_data\_reduced)[1:7], function(var) {  
 ggplot(building\_data\_reduced, aes(x = .data[[var]], y = Heating\_Load)) +  
 geom\_point(color = "skyblue") +  
 labs(title = paste("Scatter Plot for Heating Load vs", var),  
 x = var, y = "Heating Load") +  
 theme(plot.title = element\_text(hjust = 0.5),  
 axis.title.x = element\_text(size = 12),  
 axis.title.y = element\_text(size = 12))  
})  
grid.arrange(grobs = scatter\_y1, ncol = 3)



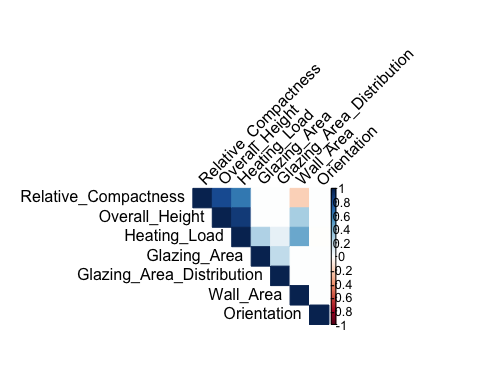
# 5. Pair plot  
ggpairs(building\_data\_reduced)



# Check the correlation matrix of the reduced dataset  
cor\_matrix\_reduced <- cor(building\_data\_reduced)  
print(cor\_matrix\_reduced)

## Relative\_Compactness Wall\_Area Overall\_Height  
## Relative\_Compactness 1.000000e+00 -2.037817e-01 8.277473e-01  
## Wall\_Area -2.037817e-01 1.000000e+00 2.809757e-01  
## Overall\_Height 8.277473e-01 2.809757e-01 1.000000e+00  
## Orientation 0.000000e+00 0.000000e+00 0.000000e+00  
## Glazing\_Area -4.558152e-17 -8.766699e-18 -1.861418e-18  
## Glazing\_Area\_Distribution 5.735014e-17 0.000000e+00 0.000000e+00  
## Heating\_Load 6.222722e-01 4.556712e-01 8.894307e-01  
## Orientation Glazing\_Area Glazing\_Area\_Distribution  
## Relative\_Compactness 0.000000000 -4.558152e-17 5.735014e-17  
## Wall\_Area 0.000000000 -8.766699e-18 0.000000e+00  
## Overall\_Height 0.000000000 -1.861418e-18 0.000000e+00  
## Orientation 1.000000000 0.000000e+00 0.000000e+00  
## Glazing\_Area 0.000000000 1.000000e+00 2.129642e-01  
## Glazing\_Area\_Distribution 0.000000000 2.129642e-01 1.000000e+00  
## Heating\_Load -0.002586534 2.698410e-01 8.736759e-02  
## Heating\_Load  
## Relative\_Compactness 0.622272179  
## Wall\_Area 0.455671157  
## Overall\_Height 0.889430674  
## Orientation -0.002586534  
## Glazing\_Area 0.269840996  
## Glazing\_Area\_Distribution 0.087367594  
## Heating\_Load 1.000000000

# Visualize the new correlation matrix  
corrplot(cor\_matrix\_reduced, method = "color", type = "upper", order = "hclust",   
 tl.col = "black", tl.srt = 45)



# Split the data into training and testing sets  
set.seed(123)   
trainIndex <- createDataPartition(building\_data\_reduced$Heating\_Load, p=0.8, list=FALSE)  
train\_data <- building\_data\_reduced[trainIndex, ]  
test\_data <- building\_data\_reduced[-trainIndex, ]

# Create a simple linear model  
lm\_model <- lm(Heating\_Load ~ ., data = train\_data)  
  
# Print the summary of the model  
summary(lm\_model)

##   
## Call:  
## lm(formula = Heating\_Load ~ ., data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.9124 -1.4316 0.1062 1.4812 7.5995   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -13.500296 3.014258 -4.479 8.96e-06 \*\*\*  
## Relative\_Compactness -13.327622 3.437977 -3.877 0.000117 \*\*\*  
## Wall\_Area 0.037481 0.004919 7.619 9.80e-14 \*\*\*  
## Overall\_Height 5.524573 0.211862 26.076 < 2e-16 \*\*\*  
## Orientation -0.057680 0.106822 -0.540 0.589419   
## Glazing\_Area 19.570301 0.930147 21.040 < 2e-16 \*\*\*  
## Glazing\_Area\_Distribution 0.237600 0.079583 2.986 0.002944 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.985 on 609 degrees of freedom  
## Multiple R-squared: 0.9128, Adjusted R-squared: 0.912   
## F-statistic: 1063 on 6 and 609 DF, p-value: < 2.2e-16

# Make predictions on the test data  
test\_predictions <- predict(lm\_model, newdata = test\_data)  
  
# Evaluate the model's performance on the test data  
rmse <- sqrt(mean((test\_data$Heating\_Load - test\_predictions)^2))  
rsq <- 1 - sum((test\_data$Heating\_Load - test\_predictions)^2) / sum((test\_data$Heating\_Load - mean(test\_data$Heating\_Load))^2)  
  
# Print the test set performance metrics  
cat("Test set RMSE:", rmse, "\n")

## Test set RMSE: 2.978433

cat("Test set R-squared:", rsq, "\n")

## Test set R-squared: 0.9148274

# Set up cross-validation  
train\_control <- trainControl(method="cv", number=10)   
  
# Train the model with cross-validation  
lm\_model\_cv <- train(Heating\_Load ~ ., data=train\_data, method="lm", trControl=train\_control)  
  
# Print the cross-validation results  
print(lm\_model\_cv)

## Linear Regression   
##   
## 616 samples  
## 6 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 553, 555, 555, 553, 555, 555, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 2.969655 0.913055 2.195585  
##   
## Tuning parameter 'intercept' was held constant at a value of TRUE

# Get the final model  
final\_model <- lm\_model\_cv$finalModel  
  
# Print the summary of the final model  
summary(final\_model)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.9124 -1.4316 0.1062 1.4812 7.5995   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -13.500296 3.014258 -4.479 8.96e-06 \*\*\*  
## Relative\_Compactness -13.327622 3.437977 -3.877 0.000117 \*\*\*  
## Wall\_Area 0.037481 0.004919 7.619 9.80e-14 \*\*\*  
## Overall\_Height 5.524573 0.211862 26.076 < 2e-16 \*\*\*  
## Orientation -0.057680 0.106822 -0.540 0.589419   
## Glazing\_Area 19.570301 0.930147 21.040 < 2e-16 \*\*\*  
## Glazing\_Area\_Distribution 0.237600 0.079583 2.986 0.002944 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.985 on 609 degrees of freedom  
## Multiple R-squared: 0.9128, Adjusted R-squared: 0.912   
## F-statistic: 1063 on 6 and 609 DF, p-value: < 2.2e-16

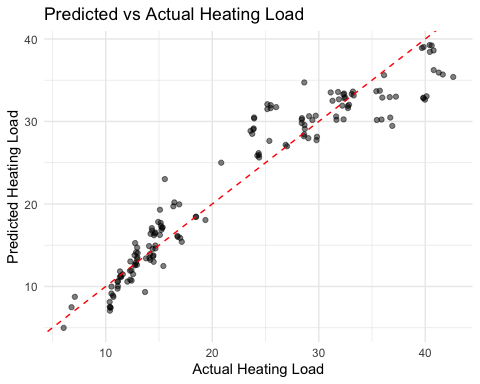
# Make predictions on the test data  
test\_predictions <- predict(lm\_model\_cv, newdata=test\_data)  
  
# Evaluate the model's performance on the test data  
rmse <- sqrt(mean((test\_data$Heating\_Load - test\_predictions)^2))  
rsq <- 1 - sum((test\_data$Heating\_Load - test\_predictions)^2) / sum((test\_data$Heating\_Load - mean(test\_data$Heating\_Load))^2)  
# Print the test set performance metrics  
cat("Test set RMSE:", rmse, "\n")

## Test set RMSE: 2.978433

cat("Test set R-squared:", rsq, "\n")

## Test set R-squared: 0.9148274

# Visualization 1: Predicted vs Actual Plot  
ggplot(data.frame(Predicted = test\_predictions, Actual = test\_data$Heating\_Load), aes(x = Actual, y = Predicted)) +  
 geom\_point(alpha = 0.5) +  
 geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  
 labs(title = "Predicted vs Actual Heating Load",  
 x = "Actual Heating Load",  
 y = "Predicted Heating Load") +  
 theme\_minimal()



# Set up Leave-One-Out Cross-Validation  
train\_control <- trainControl(method = "LOOCV")  
  
# Train the model with LOOCV  
lm\_model\_loocv <- train(Heating\_Load ~ ., data = train\_data, method = "lm", trControl = train\_control)  
  
# Print the LOOCV results  
print(lm\_model\_loocv)

## Linear Regression   
##   
## 616 samples  
## 6 predictor  
##   
## No pre-processing  
## Resampling: Leave-One-Out Cross-Validation   
## Summary of sample sizes: 615, 615, 615, 615, 615, 615, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 3.00365 0.910731 2.200825  
##   
## Tuning parameter 'intercept' was held constant at a value of TRUE

# Get the final model  
final\_model <- lm\_model\_loocv$finalModel  
  
# Print the summary of the final model  
summary(final\_model)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.9124 -1.4316 0.1062 1.4812 7.5995   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -13.500296 3.014258 -4.479 8.96e-06 \*\*\*  
## Relative\_Compactness -13.327622 3.437977 -3.877 0.000117 \*\*\*  
## Wall\_Area 0.037481 0.004919 7.619 9.80e-14 \*\*\*  
## Overall\_Height 5.524573 0.211862 26.076 < 2e-16 \*\*\*  
## Orientation -0.057680 0.106822 -0.540 0.589419   
## Glazing\_Area 19.570301 0.930147 21.040 < 2e-16 \*\*\*  
## Glazing\_Area\_Distribution 0.237600 0.079583 2.986 0.002944 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.985 on 609 degrees of freedom  
## Multiple R-squared: 0.9128, Adjusted R-squared: 0.912   
## F-statistic: 1063 on 6 and 609 DF, p-value: < 2.2e-16

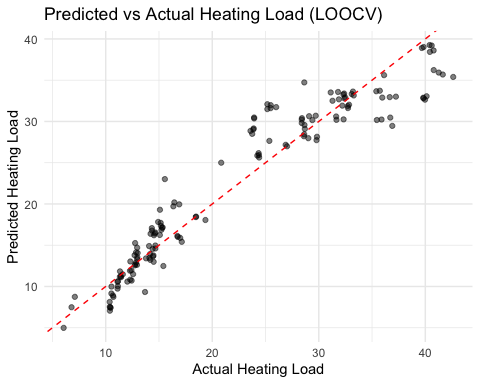
# Make predictions on the test data  
test\_predictions <- predict(lm\_model\_loocv, newdata = test\_data)  
  
# Evaluate the model's performance on the test data  
rmse <- sqrt(mean((test\_data$Heating\_Load - test\_predictions)^2))  
rsq <- 1 - sum((test\_data$Heating\_Load - test\_predictions)^2) / sum((test\_data$Heating\_Load - mean(test\_data$Heating\_Load))^2)  
  
# Print the test set performance metrics  
cat("LOOCV Test set RMSE:", rmse, "\n")

## LOOCV Test set RMSE: 2.978433

cat("LOOCV Test set R-squared:", rsq, "\n")

## LOOCV Test set R-squared: 0.9148274

# Visualization: Predicted vs Actual Plot  
ggplot(data.frame(Predicted = test\_predictions, Actual = test\_data$Heating\_Load), aes(x = Actual, y = Predicted)) +  
 geom\_point(alpha = 0.5) +  
 geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  
 labs(title = "Predicted vs Actual Heating Load (LOOCV)",  
 x = "Actual Heating Load",  
 y = "Predicted Heating Load") +  
 theme\_minimal()



# Backward selection  
  
regfit.bwd <- regsubsets(Heating\_Load ~ ., data=train\_data, nvmax=6, method="backward")  
regfit.bwd.summary <- summary(regfit.bwd)  
print(regfit.bwd.summary)

## Subset selection object  
## Call: regsubsets.formula(Heating\_Load ~ ., data = train\_data, nvmax = 6,   
## method = "backward")  
## 6 Variables (and intercept)  
## Forced in Forced out  
## Relative\_Compactness FALSE FALSE  
## Wall\_Area FALSE FALSE  
## Overall\_Height FALSE FALSE  
## Orientation FALSE FALSE  
## Glazing\_Area FALSE FALSE  
## Glazing\_Area\_Distribution FALSE FALSE  
## 1 subsets of each size up to 6  
## Selection Algorithm: backward  
## Relative\_Compactness Wall\_Area Overall\_Height Orientation Glazing\_Area  
## 1 ( 1 ) " " " " "\*" " " " "   
## 2 ( 1 ) " " " " "\*" " " "\*"   
## 3 ( 1 ) " " "\*" "\*" " " "\*"   
## 4 ( 1 ) "\*" "\*" "\*" " " "\*"   
## 5 ( 1 ) "\*" "\*" "\*" " " "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## Glazing\_Area\_Distribution  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) "\*"   
## 6 ( 1 ) "\*"

which.min(regfit.bwd.summary$bic)

## [1] 5

coef(regfit.bwd, which.min(regfit.bwd.summary$bic))

## (Intercept) Relative\_Compactness Wall\_Area   
## -13.68086004 -13.35680683 0.03743411   
## Overall\_Height Glazing\_Area Glazing\_Area\_Distribution   
## 5.52701504 19.57350701 0.23853042

cv.errors.bwd <- rep(NA,6)  
mae.bwd <- rep(NA,6)  
cv.r2.bwd <- rep(NA,6)  
test.mat=model.matrix(Heating\_Load ~ ., data=building\_data\_reduced[-trainIndex,])  
for (i in 1:6) {  
   
 coefi=coef(regfit.bwd,id=i)  
 pred=test.mat[,names(coefi)] %\*% coefi  
 cv.errors.bwd[i]=mean((building\_data\_reduced$Heating\_Load[-trainIndex]-pred)^2)  
 cv.r2.bwd[i] <- 1 - sum((pred-building\_data\_reduced$Heating\_Load[-trainIndex])^2) / sum((building\_data\_reduced$Heating\_Load[-trainIndex] - mean(building\_data\_reduced$Heating\_Load[-trainIndex]))^2)  
 mae.bwd[i] <- mean(abs(building\_data\_reduced$Heating\_Load[-trainIndex]-pred))  
}  
rmse\_bwd <- sqrt(cv.errors.bwd)  
# Print the best model and its performance metrics  
best\_model\_index <- which.min(regfit.bwd.summary$bic)  
best\_model\_coefs <- coef(regfit.bwd, best\_model\_index)  
  
cat("Best Model:\n")

## Best Model:

print(best\_model\_coefs)

## (Intercept) Relative\_Compactness Wall\_Area   
## -13.68086004 -13.35680683 0.03743411   
## Overall\_Height Glazing\_Area Glazing\_Area\_Distribution   
## 5.52701504 19.57350701 0.23853042

cat("\nPerformance Metrics for the Best Model:\n")

##   
## Performance Metrics for the Best Model:

cat("RMSE:", rmse\_bwd[best\_model\_index], "\n")

## RMSE: 2.974837

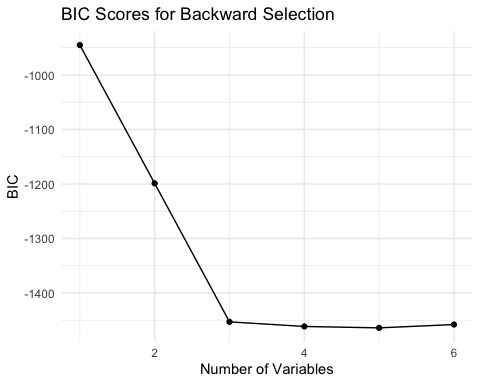
cat("MAE:", mae.bwd[best\_model\_index], "\n")

## MAE: 2.161401

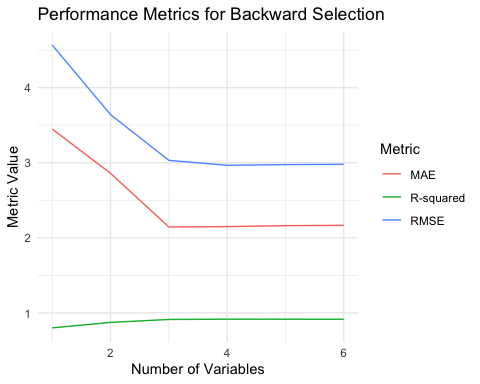
cat("R-squared:", cv.r2.bwd[best\_model\_index], "\n")

## R-squared: 0.9150329

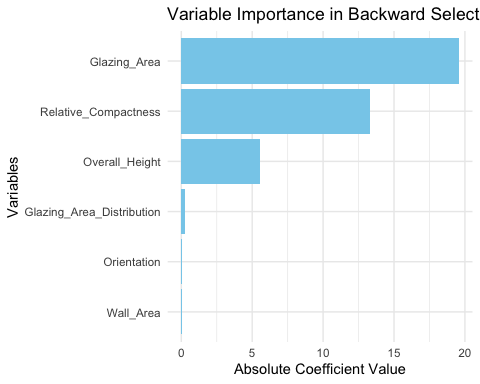
# Visualization: Plot of BIC scores  
ggplot(data.frame(vars = 1:6, bic = regfit.bwd.summary$bic), aes(x = vars, y = bic)) +  
 geom\_line() +  
 geom\_point() +  
 labs(title = "BIC Scores for Backward Selection",  
 x = "Number of Variables",  
 y = "BIC") +  
 theme\_minimal()



# Visualization: Plot of RMSE, MAE, and R-squared  
performance\_df <- data.frame(  
 vars = 1:6,  
 RMSE = rmse\_bwd,  
 MAE = mae.bwd,  
 R\_squared = cv.r2.bwd  
)  
  
ggplot(performance\_df, aes(x = vars)) +  
 geom\_line(aes(y = RMSE, color = "RMSE")) +  
 geom\_line(aes(y = MAE, color = "MAE")) +  
 geom\_line(aes(y = R\_squared, color = "R-squared")) +  
 labs(title = "Performance Metrics for Backward Selection",  
 x = "Number of Variables",  
 y = "Metric Value",  
 color = "Metric") +  
 theme\_minimal()



# Visualization: Variable Importance Plot  
var\_importance <- data.frame(  
 Variable = names(coef(regfit.bwd, 6))[-1],  
 Importance = abs(coef(regfit.bwd, 6))[-1]  
)  
var\_importance <- var\_importance[order(-var\_importance$Importance),]  
  
ggplot(var\_importance, aes(x = reorder(Variable, Importance), y = Importance)) +  
 geom\_bar(stat = "identity", fill = "skyblue") +  
 coord\_flip() +  
 labs(title = "Variable Importance in Backward Selection",  
 x = "Variables",  
 y = "Absolute Coefficient Value") +  
 theme\_minimal()



# Forward selection  
regfit.fwd <- regsubsets(Heating\_Load ~ ., data=train\_data, nvmax=6, method="forward")  
regfit.fwd.summary <- summary(regfit.fwd)  
print(regfit.fwd.summary)

## Subset selection object  
## Call: regsubsets.formula(Heating\_Load ~ ., data = train\_data, nvmax = 6,   
## method = "forward")  
## 6 Variables (and intercept)  
## Forced in Forced out  
## Relative\_Compactness FALSE FALSE  
## Wall\_Area FALSE FALSE  
## Overall\_Height FALSE FALSE  
## Orientation FALSE FALSE  
## Glazing\_Area FALSE FALSE  
## Glazing\_Area\_Distribution FALSE FALSE  
## 1 subsets of each size up to 6  
## Selection Algorithm: forward  
## Relative\_Compactness Wall\_Area Overall\_Height Orientation Glazing\_Area  
## 1 ( 1 ) " " " " "\*" " " " "   
## 2 ( 1 ) " " " " "\*" " " "\*"   
## 3 ( 1 ) " " "\*" "\*" " " "\*"   
## 4 ( 1 ) "\*" "\*" "\*" " " "\*"   
## 5 ( 1 ) "\*" "\*" "\*" " " "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## Glazing\_Area\_Distribution  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) "\*"   
## 6 ( 1 ) "\*"

which.min(regfit.fwd.summary$bic)

## [1] 5

coef(regfit.fwd, which.min(regfit.fwd.summary$bic))

## (Intercept) Relative\_Compactness Wall\_Area   
## -13.68086004 -13.35680683 0.03743411   
## Overall\_Height Glazing\_Area Glazing\_Area\_Distribution   
## 5.52701504 19.57350701 0.23853042

cv.errors.fwd <- rep(NA,6)  
mae.fwd <- rep(NA,6)  
cv.r2.fwd <- rep(NA,6)  
test.mat=model.matrix(Heating\_Load ~ ., data=building\_data\_reduced[-trainIndex,])  
for (i in 1:6) {  
   
 coefi=coef(regfit.fwd,id=i)  
 pred=test.mat[,names(coefi)] %\*% coefi  
 cv.errors.fwd[i]=mean((building\_data\_reduced$Heating\_Load[-trainIndex]-pred)^2)  
 cv.r2.fwd[i] <- 1 - sum((pred-building\_data\_reduced$Heating\_Load[-trainIndex])^2) / sum((building\_data\_reduced$Heating\_Load[-trainIndex] - mean(building\_data\_reduced$Heating\_Load[-trainIndex]))^2)  
 mae.fwd[i] <- mean(abs(building\_data\_reduced$Heating\_Load[-trainIndex]-pred))  
}  
rmse\_fwd <- sqrt(cv.errors.fwd)  
  
# Print the best model and its performance metrics  
best\_model\_index <- which.min(regfit.fwd.summary$bic)  
best\_model\_coefs <- coef(regfit.fwd, best\_model\_index)  
  
cat("Best Model:\n")

## Best Model:

print(best\_model\_coefs)

## (Intercept) Relative\_Compactness Wall\_Area   
## -13.68086004 -13.35680683 0.03743411   
## Overall\_Height Glazing\_Area Glazing\_Area\_Distribution   
## 5.52701504 19.57350701 0.23853042

cat("\nPerformance Metrics for the Best Model:\n")

##   
## Performance Metrics for the Best Model:

cat("RMSE:", rmse\_fwd[best\_model\_index], "\n")

## RMSE: 2.974837

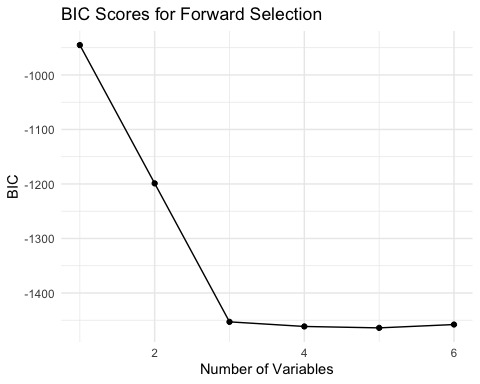
cat("MAE:", mae.fwd[best\_model\_index], "\n")

## MAE: 2.161401

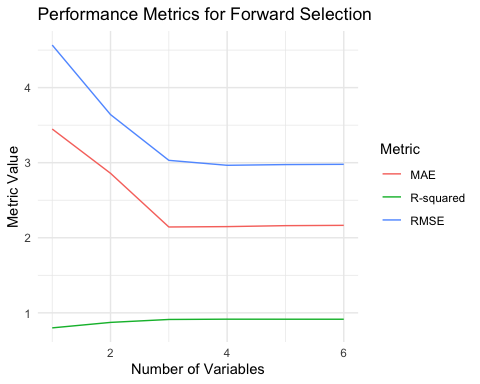
cat("R-squared:", cv.r2.fwd[best\_model\_index], "\n")

## R-squared: 0.9150329

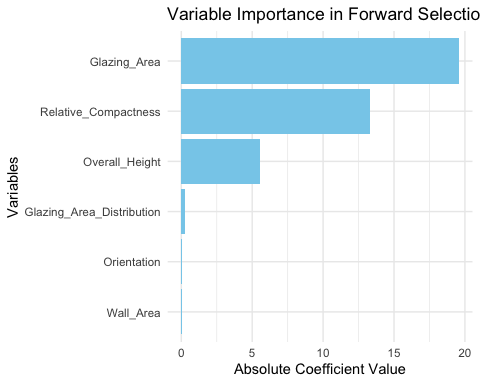
# Visualization: Plot of BIC scores  
ggplot(data.frame(vars = 1:6, bic = regfit.fwd.summary$bic), aes(x = vars, y = bic)) +  
 geom\_line() +  
 geom\_point() +  
 labs(title = "BIC Scores for Forward Selection",  
 x = "Number of Variables",  
 y = "BIC") +  
 theme\_minimal()



# Visualization: Plot of RMSE, MAE, and R-squared  
performance\_df <- data.frame(  
 vars = 1:6,  
 RMSE = rmse\_fwd,  
 MAE = mae.fwd,  
 R\_squared = cv.r2.fwd  
)  
  
ggplot(performance\_df, aes(x = vars)) +  
 geom\_line(aes(y = RMSE, color = "RMSE")) +  
 geom\_line(aes(y = MAE, color = "MAE")) +  
 geom\_line(aes(y = R\_squared, color = "R-squared")) +  
 labs(title = "Performance Metrics for Forward Selection",  
 x = "Number of Variables",  
 y = "Metric Value",  
 color = "Metric") +  
 theme\_minimal()



# Visualization: Variable Importance Plot  
var\_importance <- data.frame(  
 Variable = names(coef(regfit.fwd, 6))[-1],  
 Importance = abs(coef(regfit.fwd, 6))[-1]  
)  
var\_importance <- var\_importance[order(-var\_importance$Importance),]  
  
ggplot(var\_importance, aes(x = reorder(Variable, Importance), y = Importance)) +  
 geom\_bar(stat = "identity", fill = "skyblue") +  
 coord\_flip() +  
 labs(title = "Variable Importance in Forward Selection",  
 x = "Variables",  
 y = "Absolute Coefficient Value") +  
 theme\_minimal()



# Create interaction terms for all pairs of predictors  
interaction\_formula <- as.formula(paste("Heating\_Load ~ (", paste(names(building\_data\_reduced)[-which(names(building\_data\_reduced) == "Heating\_Load")], collapse=" + "), ")^2"))

# Backward selection with interactions  
regfit.bwd.int <- regsubsets(interaction\_formula, data=train\_data, nvmax=20, method="backward")  
regfit.bwd.int.summary <- summary(regfit.bwd.int)  
print(regfit.bwd.int.summary)

## Subset selection object  
## Call: regsubsets.formula(interaction\_formula, data = train\_data, nvmax = 20,   
## method = "backward")  
## 21 Variables (and intercept)  
## Forced in Forced out  
## Relative\_Compactness FALSE FALSE  
## Wall\_Area FALSE FALSE  
## Overall\_Height FALSE FALSE  
## Orientation FALSE FALSE  
## Glazing\_Area FALSE FALSE  
## Glazing\_Area\_Distribution FALSE FALSE  
## Relative\_Compactness:Wall\_Area FALSE FALSE  
## Relative\_Compactness:Overall\_Height FALSE FALSE  
## Relative\_Compactness:Orientation FALSE FALSE  
## Relative\_Compactness:Glazing\_Area FALSE FALSE  
## Relative\_Compactness:Glazing\_Area\_Distribution FALSE FALSE  
## Wall\_Area:Overall\_Height FALSE FALSE  
## Wall\_Area:Orientation FALSE FALSE  
## Wall\_Area:Glazing\_Area FALSE FALSE  
## Wall\_Area:Glazing\_Area\_Distribution FALSE FALSE  
## Overall\_Height:Orientation FALSE FALSE  
## Overall\_Height:Glazing\_Area FALSE FALSE  
## Overall\_Height:Glazing\_Area\_Distribution FALSE FALSE  
## Orientation:Glazing\_Area FALSE FALSE  
## Orientation:Glazing\_Area\_Distribution FALSE FALSE  
## Glazing\_Area:Glazing\_Area\_Distribution FALSE FALSE  
## 1 subsets of each size up to 20  
## Selection Algorithm: backward  
## Relative\_Compactness Wall\_Area Overall\_Height Orientation  
## 1 ( 1 ) " " " " "\*" " "   
## 2 ( 1 ) " " " " "\*" " "   
## 3 ( 1 ) " " " " "\*" " "   
## 4 ( 1 ) " " " " "\*" " "   
## 5 ( 1 ) " " " " "\*" " "   
## 6 ( 1 ) " " " " "\*" " "   
## 7 ( 1 ) "\*" " " "\*" " "   
## 8 ( 1 ) "\*" " " "\*" " "   
## 9 ( 1 ) "\*" " " "\*" " "   
## 10 ( 1 ) "\*" " " "\*" " "   
## 11 ( 1 ) "\*" " " "\*" " "   
## 12 ( 1 ) "\*" " " "\*" " "   
## 13 ( 1 ) "\*" " " "\*" " "   
## 14 ( 1 ) "\*" " " "\*" " "   
## 15 ( 1 ) "\*" " " "\*" " "   
## 16 ( 1 ) "\*" " " "\*" " "   
## 17 ( 1 ) "\*" " " "\*" "\*"   
## 18 ( 1 ) "\*" " " "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*" "\*"   
## 20 ( 1 ) "\*" "\*" "\*" "\*"   
## Glazing\_Area Glazing\_Area\_Distribution Relative\_Compactness:Wall\_Area  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " "\*"   
## 5 ( 1 ) " " " " "\*"   
## 6 ( 1 ) " " " " "\*"   
## 7 ( 1 ) " " " " "\*"   
## 8 ( 1 ) " " " " "\*"   
## 9 ( 1 ) "\*" " " "\*"   
## 10 ( 1 ) "\*" " " "\*"   
## 11 ( 1 ) "\*" "\*" "\*"   
## 12 ( 1 ) "\*" "\*" "\*"   
## 13 ( 1 ) "\*" "\*" "\*"   
## 14 ( 1 ) "\*" "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*"   
## 17 ( 1 ) "\*" "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*"   
## 20 ( 1 ) "\*" "\*" "\*"   
## Relative\_Compactness:Overall\_Height Relative\_Compactness:Orientation  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) "\*" " "   
## 4 ( 1 ) "\*" " "   
## 5 ( 1 ) "\*" " "   
## 6 ( 1 ) "\*" " "   
## 7 ( 1 ) "\*" " "   
## 8 ( 1 ) "\*" " "   
## 9 ( 1 ) "\*" " "   
## 10 ( 1 ) "\*" " "   
## 11 ( 1 ) "\*" " "   
## 12 ( 1 ) "\*" " "   
## 13 ( 1 ) "\*" " "   
## 14 ( 1 ) "\*" " "   
## 15 ( 1 ) "\*" "\*"   
## 16 ( 1 ) "\*" "\*"   
## 17 ( 1 ) "\*" "\*"   
## 18 ( 1 ) "\*" "\*"   
## 19 ( 1 ) "\*" "\*"   
## 20 ( 1 ) "\*" "\*"   
## Relative\_Compactness:Glazing\_Area  
## 1 ( 1 ) " "   
## 2 ( 1 ) "\*"   
## 3 ( 1 ) "\*"   
## 4 ( 1 ) "\*"   
## 5 ( 1 ) "\*"   
## 6 ( 1 ) "\*"   
## 7 ( 1 ) "\*"   
## 8 ( 1 ) "\*"   
## 9 ( 1 ) "\*"   
## 10 ( 1 ) "\*"   
## 11 ( 1 ) "\*"   
## 12 ( 1 ) "\*"   
## 13 ( 1 ) "\*"   
## 14 ( 1 ) "\*"   
## 15 ( 1 ) "\*"   
## 16 ( 1 ) "\*"   
## 17 ( 1 ) "\*"   
## 18 ( 1 ) "\*"   
## 19 ( 1 ) "\*"   
## 20 ( 1 ) "\*"   
## Relative\_Compactness:Glazing\_Area\_Distribution  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) " "   
## 6 ( 1 ) " "   
## 7 ( 1 ) " "   
## 8 ( 1 ) " "   
## 9 ( 1 ) " "   
## 10 ( 1 ) " "   
## 11 ( 1 ) " "   
## 12 ( 1 ) " "   
## 13 ( 1 ) " "   
## 14 ( 1 ) "\*"   
## 15 ( 1 ) "\*"   
## 16 ( 1 ) "\*"   
## 17 ( 1 ) "\*"   
## 18 ( 1 ) "\*"   
## 19 ( 1 ) "\*"   
## 20 ( 1 ) "\*"   
## Wall\_Area:Overall\_Height Wall\_Area:Orientation Wall\_Area:Glazing\_Area  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " " "   
## 5 ( 1 ) " " " " " "   
## 6 ( 1 ) " " " " " "   
## 7 ( 1 ) " " " " " "   
## 8 ( 1 ) "\*" " " " "   
## 9 ( 1 ) "\*" " " " "   
## 10 ( 1 ) "\*" " " "\*"   
## 11 ( 1 ) "\*" " " "\*"   
## 12 ( 1 ) "\*" " " "\*"   
## 13 ( 1 ) "\*" " " "\*"   
## 14 ( 1 ) "\*" " " "\*"   
## 15 ( 1 ) "\*" " " "\*"   
## 16 ( 1 ) "\*" " " "\*"   
## 17 ( 1 ) "\*" " " "\*"   
## 18 ( 1 ) "\*" " " "\*"   
## 19 ( 1 ) "\*" " " "\*"   
## 20 ( 1 ) "\*" " " "\*"   
## Wall\_Area:Glazing\_Area\_Distribution Overall\_Height:Orientation  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " " "   
## 4 ( 1 ) " " " "   
## 5 ( 1 ) " " " "   
## 6 ( 1 ) " " " "   
## 7 ( 1 ) " " " "   
## 8 ( 1 ) " " " "   
## 9 ( 1 ) " " " "   
## 10 ( 1 ) " " " "   
## 11 ( 1 ) " " " "   
## 12 ( 1 ) " " " "   
## 13 ( 1 ) "\*" " "   
## 14 ( 1 ) "\*" " "   
## 15 ( 1 ) "\*" " "   
## 16 ( 1 ) "\*" " "   
## 17 ( 1 ) "\*" " "   
## 18 ( 1 ) "\*" "\*"   
## 19 ( 1 ) "\*" "\*"   
## 20 ( 1 ) "\*" "\*"   
## Overall\_Height:Glazing\_Area Overall\_Height:Glazing\_Area\_Distribution  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " " "   
## 4 ( 1 ) " " " "   
## 5 ( 1 ) " " "\*"   
## 6 ( 1 ) " " "\*"   
## 7 ( 1 ) " " "\*"   
## 8 ( 1 ) " " "\*"   
## 9 ( 1 ) " " "\*"   
## 10 ( 1 ) " " "\*"   
## 11 ( 1 ) " " "\*"   
## 12 ( 1 ) "\*" "\*"   
## 13 ( 1 ) "\*" "\*"   
## 14 ( 1 ) "\*" "\*"   
## 15 ( 1 ) "\*" "\*"   
## 16 ( 1 ) "\*" "\*"   
## 17 ( 1 ) "\*" "\*"   
## 18 ( 1 ) "\*" "\*"   
## 19 ( 1 ) "\*" "\*"   
## 20 ( 1 ) "\*" "\*"   
## Orientation:Glazing\_Area Orientation:Glazing\_Area\_Distribution  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " " "   
## 4 ( 1 ) " " " "   
## 5 ( 1 ) " " " "   
## 6 ( 1 ) " " " "   
## 7 ( 1 ) " " " "   
## 8 ( 1 ) " " " "   
## 9 ( 1 ) " " " "   
## 10 ( 1 ) " " " "   
## 11 ( 1 ) " " " "   
## 12 ( 1 ) " " " "   
## 13 ( 1 ) " " " "   
## 14 ( 1 ) " " " "   
## 15 ( 1 ) " " " "   
## 16 ( 1 ) " " "\*"   
## 17 ( 1 ) " " "\*"   
## 18 ( 1 ) " " "\*"   
## 19 ( 1 ) " " "\*"   
## 20 ( 1 ) "\*" "\*"   
## Glazing\_Area:Glazing\_Area\_Distribution  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) " "   
## 6 ( 1 ) "\*"   
## 7 ( 1 ) "\*"   
## 8 ( 1 ) "\*"   
## 9 ( 1 ) "\*"   
## 10 ( 1 ) "\*"   
## 11 ( 1 ) "\*"   
## 12 ( 1 ) "\*"   
## 13 ( 1 ) "\*"   
## 14 ( 1 ) "\*"   
## 15 ( 1 ) "\*"   
## 16 ( 1 ) "\*"   
## 17 ( 1 ) "\*"   
## 18 ( 1 ) "\*"   
## 19 ( 1 ) "\*"   
## 20 ( 1 ) "\*"

# Forward selection with interactions  
regfit.fwd.int <- regsubsets(interaction\_formula, data=train\_data, nvmax=20, method="forward")  
regfit.fwd.int.summary <- summary(regfit.fwd.int)  
print(regfit.fwd.int.summary)

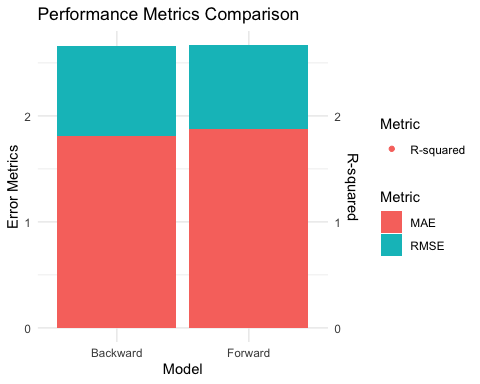
## Subset selection object  
## Call: regsubsets.formula(interaction\_formula, data = train\_data, nvmax = 20,   
## method = "forward")  
## 21 Variables (and intercept)  
## Forced in Forced out  
## Relative\_Compactness FALSE FALSE  
## Wall\_Area FALSE FALSE  
## Overall\_Height FALSE FALSE  
## Orientation FALSE FALSE  
## Glazing\_Area FALSE FALSE  
## Glazing\_Area\_Distribution FALSE FALSE  
## Relative\_Compactness:Wall\_Area FALSE FALSE  
## Relative\_Compactness:Overall\_Height FALSE FALSE  
## Relative\_Compactness:Orientation FALSE FALSE  
## Relative\_Compactness:Glazing\_Area FALSE FALSE  
## Relative\_Compactness:Glazing\_Area\_Distribution FALSE FALSE  
## Wall\_Area:Overall\_Height FALSE FALSE  
## Wall\_Area:Orientation FALSE FALSE  
## Wall\_Area:Glazing\_Area FALSE FALSE  
## Wall\_Area:Glazing\_Area\_Distribution FALSE FALSE  
## Overall\_Height:Orientation FALSE FALSE  
## Overall\_Height:Glazing\_Area FALSE FALSE  
## Overall\_Height:Glazing\_Area\_Distribution FALSE FALSE  
## Orientation:Glazing\_Area FALSE FALSE  
## Orientation:Glazing\_Area\_Distribution FALSE FALSE  
## Glazing\_Area:Glazing\_Area\_Distribution FALSE FALSE  
## 1 subsets of each size up to 20  
## Selection Algorithm: forward  
## Relative\_Compactness Wall\_Area Overall\_Height Orientation  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) " " " " " " " "   
## 3 ( 1 ) " " " " " " " "   
## 4 ( 1 ) " " " " " " " "   
## 5 ( 1 ) " " " " " " " "   
## 6 ( 1 ) " " " " " " " "   
## 7 ( 1 ) " " " " " " " "   
## 8 ( 1 ) " " " " " " " "   
## 9 ( 1 ) " " " " " " " "   
## 10 ( 1 ) " " " " "\*" " "   
## 11 ( 1 ) " " " " "\*" " "   
## 12 ( 1 ) "\*" " " "\*" " "   
## 13 ( 1 ) "\*" " " "\*" " "   
## 14 ( 1 ) "\*" " " "\*" " "   
## 15 ( 1 ) "\*" "\*" "\*" " "   
## 16 ( 1 ) "\*" "\*" "\*" " "   
## 17 ( 1 ) "\*" "\*" "\*" " "   
## 18 ( 1 ) "\*" "\*" "\*" " "   
## 19 ( 1 ) "\*" "\*" "\*" " "   
## 20 ( 1 ) "\*" "\*" "\*" "\*"   
## Glazing\_Area Glazing\_Area\_Distribution Relative\_Compactness:Wall\_Area  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " " "   
## 5 ( 1 ) "\*" " " " "   
## 6 ( 1 ) "\*" " " "\*"   
## 7 ( 1 ) "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*"   
## 11 ( 1 ) "\*" "\*" "\*"   
## 12 ( 1 ) "\*" "\*" "\*"   
## 13 ( 1 ) "\*" "\*" "\*"   
## 14 ( 1 ) "\*" "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*"   
## 17 ( 1 ) "\*" "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*"   
## 20 ( 1 ) "\*" "\*" "\*"   
## Relative\_Compactness:Overall\_Height Relative\_Compactness:Orientation  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " " "   
## 4 ( 1 ) " " " "   
## 5 ( 1 ) " " " "   
## 6 ( 1 ) " " " "   
## 7 ( 1 ) " " " "   
## 8 ( 1 ) " " " "   
## 9 ( 1 ) " " " "   
## 10 ( 1 ) " " " "   
## 11 ( 1 ) "\*" " "   
## 12 ( 1 ) "\*" " "   
## 13 ( 1 ) "\*" " "   
## 14 ( 1 ) "\*" " "   
## 15 ( 1 ) "\*" " "   
## 16 ( 1 ) "\*" "\*"   
## 17 ( 1 ) "\*" "\*"   
## 18 ( 1 ) "\*" "\*"   
## 19 ( 1 ) "\*" "\*"   
## 20 ( 1 ) "\*" "\*"   
## Relative\_Compactness:Glazing\_Area  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) " "   
## 6 ( 1 ) " "   
## 7 ( 1 ) "\*"   
## 8 ( 1 ) "\*"   
## 9 ( 1 ) "\*"   
## 10 ( 1 ) "\*"   
## 11 ( 1 ) "\*"   
## 12 ( 1 ) "\*"   
## 13 ( 1 ) "\*"   
## 14 ( 1 ) "\*"   
## 15 ( 1 ) "\*"   
## 16 ( 1 ) "\*"   
## 17 ( 1 ) "\*"   
## 18 ( 1 ) "\*"   
## 19 ( 1 ) "\*"   
## 20 ( 1 ) "\*"   
## Relative\_Compactness:Glazing\_Area\_Distribution  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) " "   
## 6 ( 1 ) " "   
## 7 ( 1 ) " "   
## 8 ( 1 ) " "   
## 9 ( 1 ) " "   
## 10 ( 1 ) " "   
## 11 ( 1 ) " "   
## 12 ( 1 ) " "   
## 13 ( 1 ) " "   
## 14 ( 1 ) "\*"   
## 15 ( 1 ) "\*"   
## 16 ( 1 ) "\*"   
## 17 ( 1 ) "\*"   
## 18 ( 1 ) "\*"   
## 19 ( 1 ) "\*"   
## 20 ( 1 ) "\*"   
## Wall\_Area:Overall\_Height Wall\_Area:Orientation Wall\_Area:Glazing\_Area  
## 1 ( 1 ) "\*" " " " "   
## 2 ( 1 ) "\*" " " " "   
## 3 ( 1 ) "\*" " " " "   
## 4 ( 1 ) "\*" " " " "   
## 5 ( 1 ) "\*" " " " "   
## 6 ( 1 ) "\*" " " " "   
## 7 ( 1 ) "\*" " " " "   
## 8 ( 1 ) "\*" " " " "   
## 9 ( 1 ) "\*" " " "\*"   
## 10 ( 1 ) "\*" " " "\*"   
## 11 ( 1 ) "\*" " " "\*"   
## 12 ( 1 ) "\*" " " "\*"   
## 13 ( 1 ) "\*" " " "\*"   
## 14 ( 1 ) "\*" " " "\*"   
## 15 ( 1 ) "\*" " " "\*"   
## 16 ( 1 ) "\*" " " "\*"   
## 17 ( 1 ) "\*" " " "\*"   
## 18 ( 1 ) "\*" "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*"   
## 20 ( 1 ) "\*" "\*" "\*"   
## Wall\_Area:Glazing\_Area\_Distribution Overall\_Height:Orientation  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " " "   
## 4 ( 1 ) " " " "   
## 5 ( 1 ) " " " "   
## 6 ( 1 ) " " " "   
## 7 ( 1 ) " " " "   
## 8 ( 1 ) " " " "   
## 9 ( 1 ) " " " "   
## 10 ( 1 ) " " " "   
## 11 ( 1 ) " " " "   
## 12 ( 1 ) " " " "   
## 13 ( 1 ) "\*" " "   
## 14 ( 1 ) "\*" " "   
## 15 ( 1 ) "\*" " "   
## 16 ( 1 ) "\*" " "   
## 17 ( 1 ) "\*" " "   
## 18 ( 1 ) "\*" " "   
## 19 ( 1 ) "\*" "\*"   
## 20 ( 1 ) "\*" "\*"   
## Overall\_Height:Glazing\_Area Overall\_Height:Glazing\_Area\_Distribution  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) "\*" " "   
## 3 ( 1 ) "\*" "\*"   
## 4 ( 1 ) "\*" "\*"   
## 5 ( 1 ) "\*" "\*"   
## 6 ( 1 ) "\*" "\*"   
## 7 ( 1 ) "\*" "\*"   
## 8 ( 1 ) "\*" "\*"   
## 9 ( 1 ) "\*" "\*"   
## 10 ( 1 ) "\*" "\*"   
## 11 ( 1 ) "\*" "\*"   
## 12 ( 1 ) "\*" "\*"   
## 13 ( 1 ) "\*" "\*"   
## 14 ( 1 ) "\*" "\*"   
## 15 ( 1 ) "\*" "\*"   
## 16 ( 1 ) "\*" "\*"   
## 17 ( 1 ) "\*" "\*"   
## 18 ( 1 ) "\*" "\*"   
## 19 ( 1 ) "\*" "\*"   
## 20 ( 1 ) "\*" "\*"   
## Orientation:Glazing\_Area Orientation:Glazing\_Area\_Distribution  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " " "   
## 4 ( 1 ) " " " "   
## 5 ( 1 ) " " " "   
## 6 ( 1 ) " " " "   
## 7 ( 1 ) " " " "   
## 8 ( 1 ) " " " "   
## 9 ( 1 ) " " " "   
## 10 ( 1 ) " " " "   
## 11 ( 1 ) " " " "   
## 12 ( 1 ) " " " "   
## 13 ( 1 ) " " " "   
## 14 ( 1 ) " " " "   
## 15 ( 1 ) " " " "   
## 16 ( 1 ) " " " "   
## 17 ( 1 ) " " "\*"   
## 18 ( 1 ) " " "\*"   
## 19 ( 1 ) " " "\*"   
## 20 ( 1 ) " " "\*"   
## Glazing\_Area:Glazing\_Area\_Distribution  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) "\*"   
## 5 ( 1 ) "\*"   
## 6 ( 1 ) "\*"   
## 7 ( 1 ) "\*"   
## 8 ( 1 ) "\*"   
## 9 ( 1 ) "\*"   
## 10 ( 1 ) "\*"   
## 11 ( 1 ) "\*"   
## 12 ( 1 ) "\*"   
## 13 ( 1 ) "\*"   
## 14 ( 1 ) "\*"   
## 15 ( 1 ) "\*"   
## 16 ( 1 ) "\*"   
## 17 ( 1 ) "\*"   
## 18 ( 1 ) "\*"   
## 19 ( 1 ) "\*"   
## 20 ( 1 ) "\*"

# Function to calculate performance metrics  
calculate\_metrics <- function(predictions, actual) {  
 mse <- mean((actual - predictions)^2)  
 rmse <- sqrt(mse)  
 mae <- mean(abs(actual - predictions))  
 r\_squared <- 1 - sum((actual - predictions)^2) / sum((actual - mean(actual))^2)  
   
 return(list(mse=mse, rmse=rmse, mae=mae, r\_squared=r\_squared))  
}

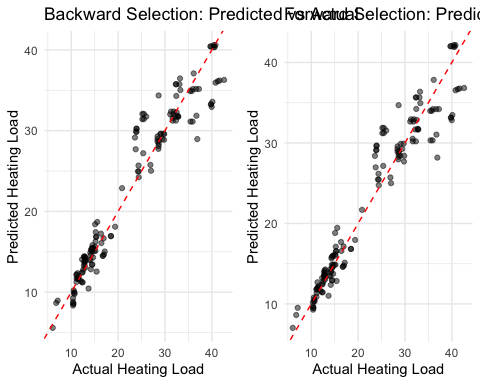
# Evaluate backward selection model with interactions  
best\_bwd\_model\_int <- lm(as.formula(paste("Heating\_Load ~", paste(names(coef(regfit.bwd.int, which.min(regfit.bwd.int.summary$bic)))[-1], collapse=" + "))), data=train\_data)   
  
bwd\_predictions\_int <- predict(best\_bwd\_model\_int,newdata=test\_data)   
  
bwd\_metrics\_int<-calculate\_metrics(bwd\_predictions\_int,test\_data$Heating\_Load)

# Evaluate forward selection model with interactions  
best\_fwd\_model\_int<-lm(as.formula(paste("Heating\_Load ~", paste(names(coef(regfit.fwd.int, which.min(regfit.fwd.int.summary$bic)))[-1], collapse=" + "))), data=train\_data)   
  
fwd\_predictions\_int<-predict(best\_fwd\_model\_int,newdata=test\_data)   
  
fwd\_metrics\_int<-calculate\_metrics(fwd\_predictions\_int,test\_data$Heating\_Load)

# Visualization : Performance Metrics Comparison  
metrics\_df <- data.frame(  
 Model = c("Backward", "Forward"),  
 RMSE = c(bwd\_metrics\_int$rmse, fwd\_metrics\_int$rmse),  
 MAE = c(bwd\_metrics\_int$mae, fwd\_metrics\_int$mae),  
 R\_squared = c(bwd\_metrics\_int$r\_squared, fwd\_metrics\_int$r\_squared)  
)  
  
ggplot(metrics\_df, aes(x = Model)) +  
 geom\_bar(aes(y = RMSE, fill = "RMSE"), stat = "identity", position = "dodge") +  
 geom\_bar(aes(y = MAE, fill = "MAE"), stat = "identity", position = "dodge") +  
 geom\_point(aes(y = R\_squared, color = "R-squared")) +  
 scale\_y\_continuous(sec.axis = sec\_axis(~., name = "R-squared")) +  
 labs(title = "Performance Metrics Comparison",  
 y = "Error Metrics",  
 fill = "Metric",  
 color = "Metric") +  
 theme\_minimal()



# Visualization: Predicted vs Actual Plot  
plot\_predictions <- function(predictions, actual, title) {  
 ggplot(data.frame(Predicted = predictions, Actual = actual), aes(x = Actual, y = Predicted)) +  
 geom\_point(alpha = 0.5) +  
 geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  
 labs(title = title,  
 x = "Actual Heating Load",  
 y = "Predicted Heating Load") +  
 theme\_minimal()  
}  
  
bwd\_plot <- plot\_predictions(bwd\_predictions\_int, test\_data$Heating\_Load, "Backward Selection: Predicted vs Actual")  
fwd\_plot <- plot\_predictions(fwd\_predictions\_int, test\_data$Heating\_Load, "Forward Selection: Predicted vs Actual")  
  
grid.arrange(bwd\_plot, fwd\_plot, ncol = 2)



library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8

# Prepare the data  
x <- as.matrix(train\_data[, !names(train\_data) %in% c("Heating\_Load")])  
y <- train\_data$Heating\_Load  
  
# Fit Lasso model  
lasso\_model <- cv.glmnet(x, y, alpha = 1)  
  
# Make predictions on test data  
x\_test <- as.matrix(test\_data[, !names(test\_data) %in% c("Heating\_Load")])  
lasso\_predictions <- predict(lasso\_model, newx = x\_test, s = "lambda.min")  
  
# Calculate performance metrics  
lasso\_rmse <- sqrt(mean((test\_data$Heating\_Load - lasso\_predictions)^2))  
lasso\_rsq <- 1 - sum((test\_data$Heating\_Load - lasso\_predictions)^2) / sum((test\_data$Heating\_Load - mean(test\_data$Heating\_Load))^2)  
  
cat("Lasso Regression Results:\n")

## Lasso Regression Results:

cat("RMSE:", lasso\_rmse, "\n")

## RMSE: 2.982335

cat("R-squared:", lasso\_rsq, "\n")

## R-squared: 0.9146041

# Fit Ridge model  
ridge\_model <- cv.glmnet(x, y, alpha = 0)  
  
# Make predictions on test data  
ridge\_predictions <- predict(ridge\_model, newx = x\_test, s = "lambda.min")  
  
# Calculate performance metrics  
ridge\_rmse <- sqrt(mean((test\_data$Heating\_Load - ridge\_predictions)^2))  
ridge\_rsq <- 1 - sum((test\_data$Heating\_Load - ridge\_predictions)^2) / sum((test\_data$Heating\_Load - mean(test\_data$Heating\_Load))^2)  
  
cat("Ridge Regression Results:\n")

## Ridge Regression Results:

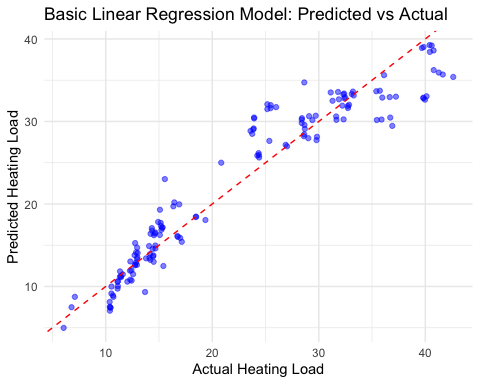
cat("RMSE:", ridge\_rmse, "\n")

## RMSE: 3.244605

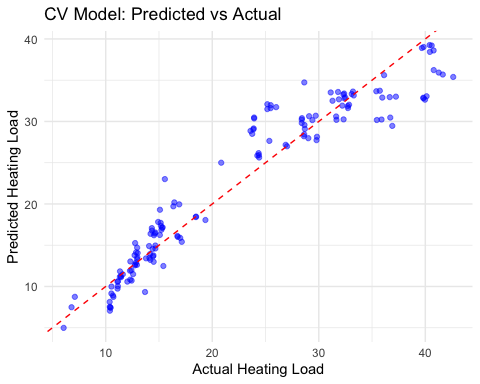
cat("R-squared:", ridge\_rsq, "\n")

## R-squared: 0.898924

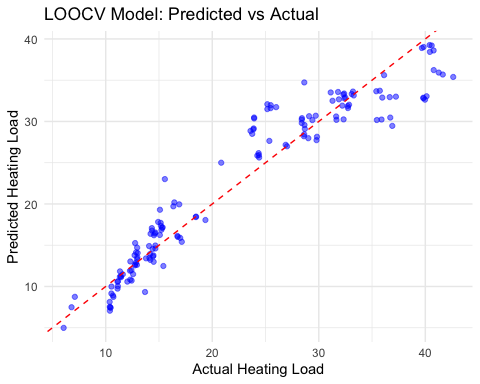
# Visualizations for all models  
  
# Function to create predicted vs actual plots  
plot\_predictions\_vs\_actual <- function(predictions, actual\_values, title) {  
 data <- data.frame(Predicted = as.vector(predictions), Actual = actual\_values)  
 ggplot(data, aes(x = Actual, y = Predicted)) +  
 geom\_point(color = 'blue', alpha = 0.5) +  
 geom\_abline(slope = 1, intercept = 0, color = 'red', linetype = 'dashed') +  
 labs(title = title, x = "Actual Heating Load", y = "Predicted Heating Load") +  
 theme\_minimal()  
}  
  
po\_lm <- plot\_predictions\_vs\_actual(predict(lm\_model, newdata = test\_data),   
 test\_data$Heating\_Load,   
 "Basic Linear Regression Model: Predicted vs Actual")  
  
p1\_cv <- plot\_predictions\_vs\_actual(predict(lm\_model\_cv, newdata = test\_data),   
 test\_data$Heating\_Load,   
 "CV Model: Predicted vs Actual")  
  
p2\_loocv <- plot\_predictions\_vs\_actual(predict(lm\_model\_loocv, newdata = test\_data),   
 test\_data$Heating\_Load,   
 "LOOCV Model: Predicted vs Actual")  
  
p3\_backward <- plot\_predictions\_vs\_actual(predict(best\_bwd\_model\_int, newdata = test\_data),   
 test\_data$Heating\_Load,   
 "Backward Selection: Predicted vs Actual")  
  
p4\_forward <- plot\_predictions\_vs\_actual(predict(best\_fwd\_model\_int, newdata = test\_data),   
 test\_data$Heating\_Load,   
 "Forward Selection: Predicted vs Actual")  
  
p5\_backward\_int <- plot\_predictions\_vs\_actual(bwd\_predictions\_int,   
 test\_data$Heating\_Load,   
 "Backward Selection with Interactions: Predicted vs Actual")  
  
p6\_forward\_int <- plot\_predictions\_vs\_actual(fwd\_predictions\_int,   
 test\_data$Heating\_Load,   
 "Forward Selection with Interactions: Predicted vs Actual")  
  
p7\_lasso <- plot\_predictions\_vs\_actual(lasso\_predictions,   
 test\_data$Heating\_Load,   
 "Lasso Regression: Predicted vs Actual")  
  
p8\_ridge <- plot\_predictions\_vs\_actual(ridge\_predictions,   
 test\_data$Heating\_Load,   
 "Ridge Regression: Predicted vs Actual")  
  
po\_lm



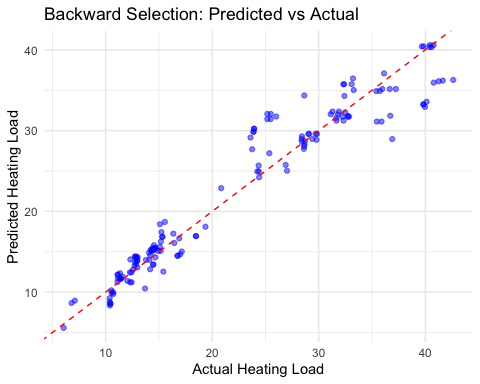
p1\_cv



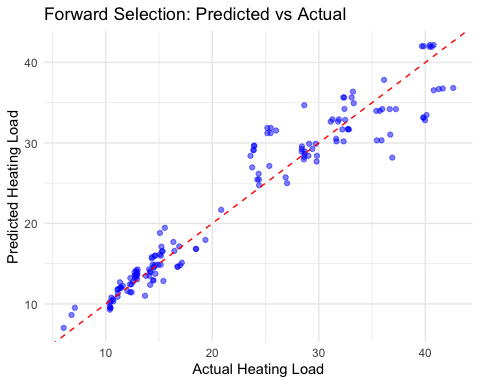
p2\_loocv



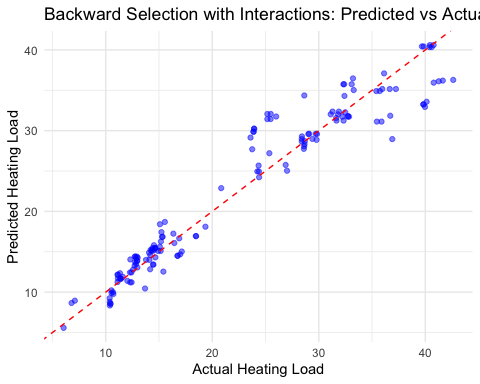
p3\_backward



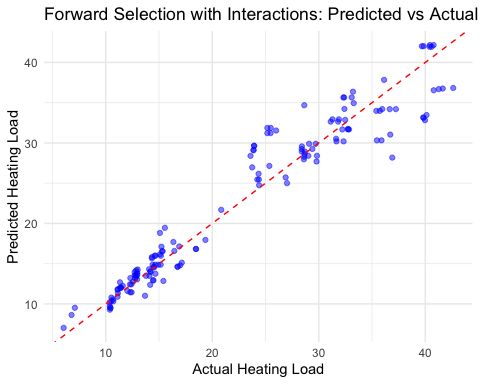
p4\_forward



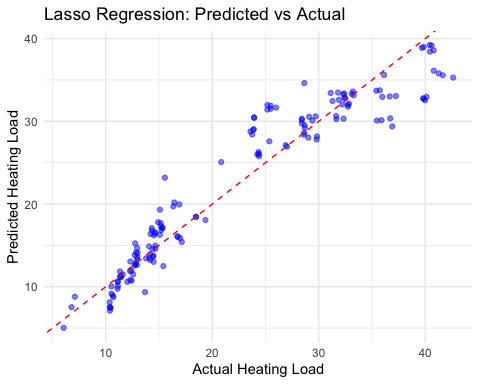
p5\_backward\_int



p6\_forward\_int



p7\_lasso



p8\_ridge

