Exercise 8

Advanced Methods for Regression and Classification

Rita Selimi

09/12/2024

Data Preparation

Load the `ozone` dataset and split it into training and test sets.

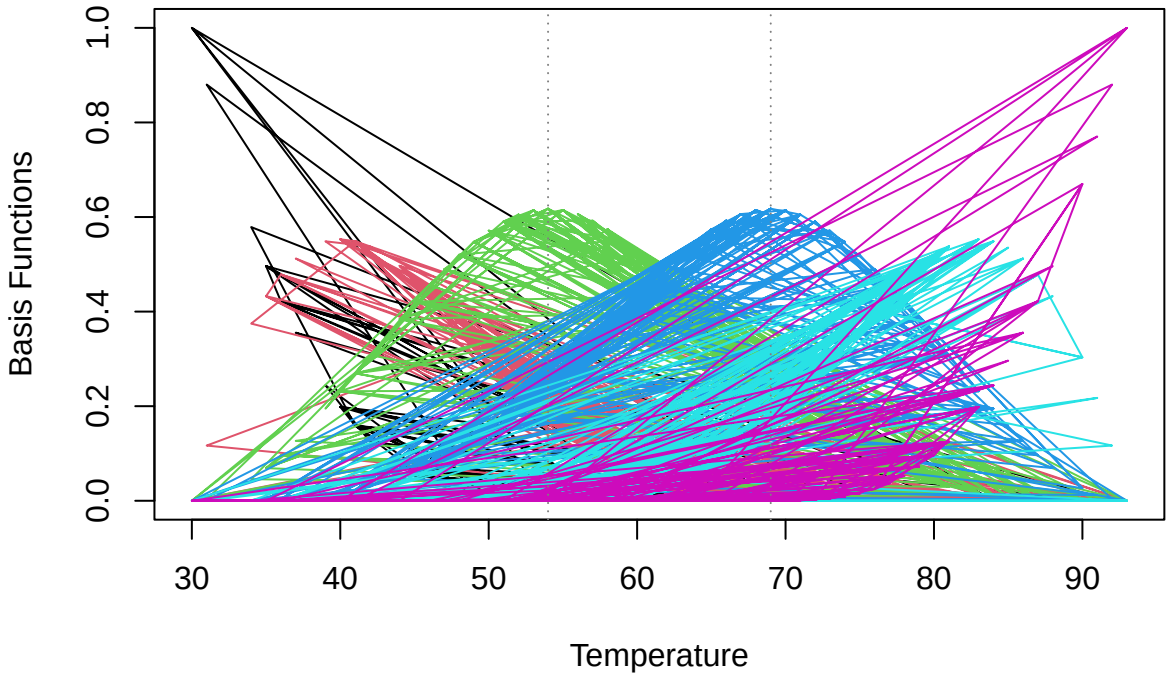
```
# Load data
data("ozone", package = "gclus")

# Random seed for reproducibility
set.seed(123)

# Split data into training (2/3) and test (1/3) sets
n <- nrow(ozone)
train_indices <- sample(1:n, size = round(2 * n/3))
train_data <- ozone[train_indices, ]
test_data <- ozone[-train_indices, ]
```

Task 1: Construct and Visualize Basis Functions

Use `lecturespl()` to construct basis functions and visualize them.

```
# Spline function
lecturespl <- function(x, knots, order = 4) {
    xquantiles <- quantile(x, probs = seq(0, 1, length.out = knots + 2)[-c(1, knots +
        2)])
    X <- bs(x, knots = xquantiles, degree = order - 1, intercept = TRUE)
    list(X = X, x = x, xquantiles = xquantiles)
}

# Construct basis functions for training data
knots <- 2
basis <- lecturespl(train_data$Temp, knots = knots, order = 4)

# Plot the basis functions
plotspl <- function(splobj, ...) {
    matplot(splobj$x, splobj$X, type = "l", lty = 1, xlab = "Temperature", ylab = "Basis Functions",
        ...)
    abline(v = splobj$xquantiles, lty = 3, col = gray(0.5))
}
```

```r
plotspl(basis)
```

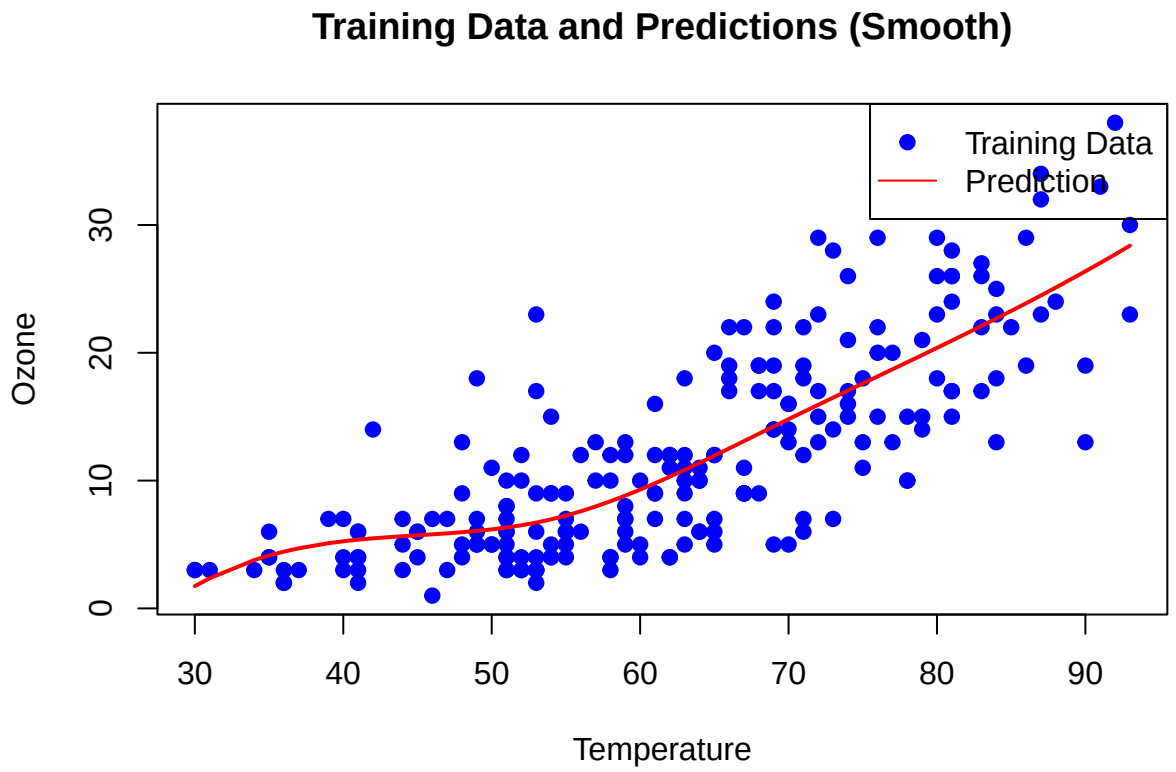


Task 2: Fit the Model and Visualize Training Predictions

Fit the model using the basis functions and visualize predictions on the training set.

```r
# Sort training data by temperature for smooth plotting
train_sorted <- train_data[order(train_data$Temp), ]

# Generate the sorted basis functions
train_basis_sorted <- lecturespl(train_sorted$Temp, knots = knots, order = 4)

# Create a data frame with response variable and basis predictors
train_model_data <- as.data.frame(train_basis_sorted$X)
train_model_data$Ozone <- train_sorted$Ozone

# Fit the model using the basis predictors and Ozone response
model <- lm(Ozone ~ ., data = train_model_data)

# Predict with the sorted basis functions
train_pred_sorted <- predict(model)

# Plot training data and sorted predictions
plot(train_sorted$Temp, train_sorted$Ozone, main = "Training Data and Predictions (Smooth)",
```

```
    xlab = "Temperature", ylab = "Ozone", pch = 19, col = "blue")
lines(train_sorted$Temp, train_pred_sorted, col = "red", lwd = 2)
legend("topright", legend = c("Training Data", "Prediction"), col = c("blue", "red"),
    pch = c(19, NA), lty = c(NA, 1))
```



The red prediction curve follows a smooth trend that aligns with the general pattern of the training data. It captures the relationship between the temperature and ozone concentration without being overly noisy or irregular.

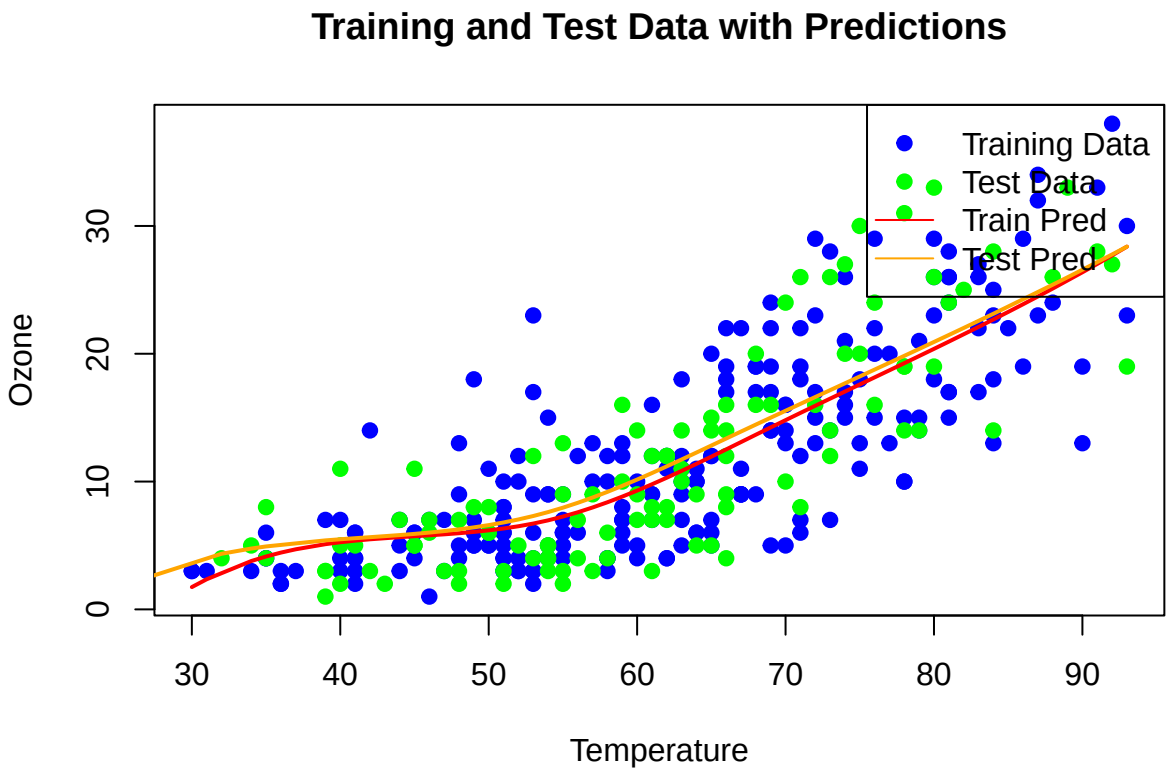
This smoothness indicates that the spline basis functions are working as intended, and the model is appropriately fitted to the sorted data.

Task 3: Predict and Visualize Test Data

Predict the test data and visualize alongside training data.

```
# Sort test data by temperature for smooth plotting
test_sorted <- test_data[order(test_data$Temp), ]

# Generate the sorted basis functions for test data
test_basis_sorted <- lecturespl(test_sorted$Temp, knots = knots, order = 4)

# Create a data frame with sorted test basis functions
test_model_data_sorted <- as.data.frame(test_basis_sorted$X)

# Predict ozone levels for the sorted test data
```

```r
test_pred_sorted <- predict(model, newdata = test_model_data_sorted)

# Plot training and test data with predictions
plot(train_sorted$Temp, train_sorted$Ozone, col = "blue", pch = 19, xlab = "Temperature",
    ylab = "Ozone", main = "Training and Test Data with Predictions")
points(test_sorted$Temp, test_sorted$Ozone, col = "green", pch = 19)
lines(train_sorted$Temp, train_pred_sorted, col = "red", lwd = 2)
lines(test_sorted$Temp, test_pred_sorted, col = "orange", lwd = 2)
legend("topright", legend = c("Training Data", "Test Data", "Train Pred", "Test Pred"),
    col = c("blue", "green", "red", "orange"), pch = c(19, 19, NA, NA), lty = c(NA,
        NA, 1, 1))
```

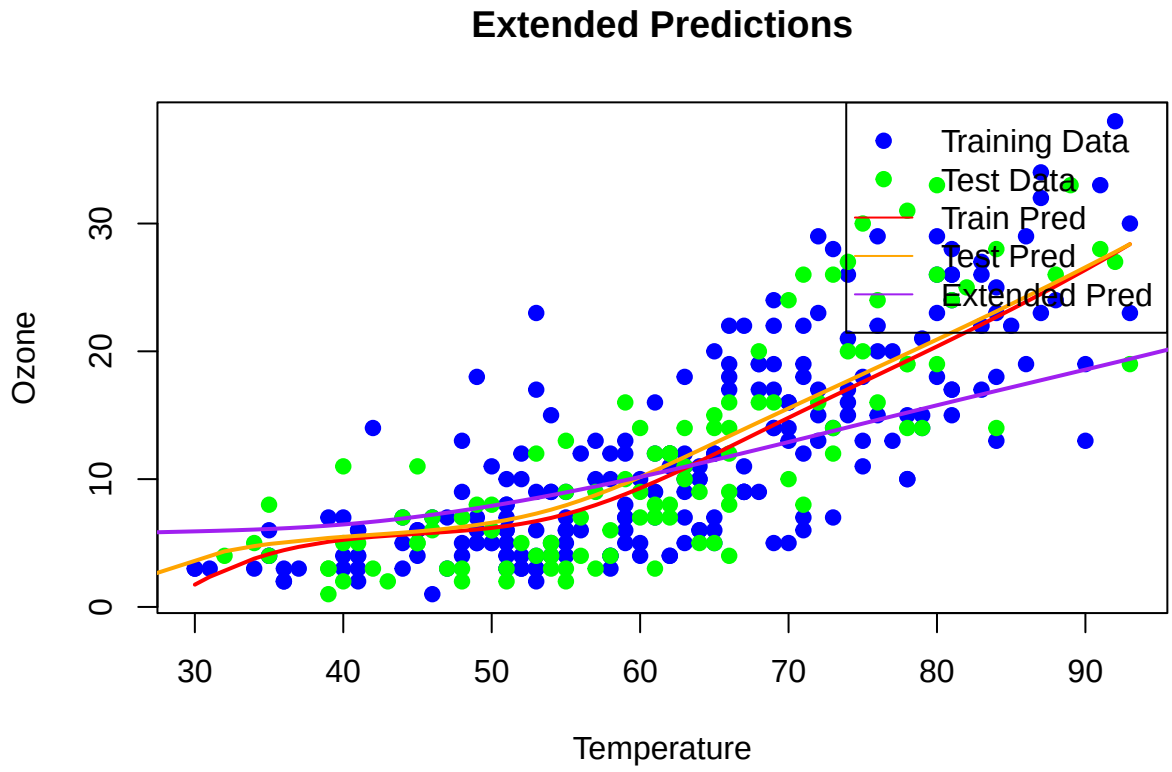


Task 4: Extend the Range and Predict

Extend the range of temperature data and predict.

```r
# Generate new temperature data
new_temps <- seq(0, 120, length.out = 100)

# Generate the basis functions for the extended range using the training knots
new_basis <- lecturespl(new_temps, knots = knots, order = 4)

# Create a data frame for the new basis functions
new_model_data <- as.data.frame(new_basis$X)
```

```r
# Predict ozone levels for the extended temperature range
new_preds <- predict(model, newdata = new_model_data)

# Plot the extended predictions along with training and test data
plot(train_sorted$Temp, train_sorted$Ozone, col = "blue", pch = 19, xlab = "Temperature",
    ylab = "Ozone", main = "Extended Predictions")
points(test_sorted$Temp, test_sorted$Ozone, col = "green", pch = 19)
lines(train_sorted$Temp, train_pred_sorted, col = "red", lwd = 2)
lines(test_sorted$Temp, test_pred_sorted, col = "orange", lwd = 2)
lines(new_temps, new_preds, col = "purple", lwd = 2)
legend("topright", legend = c("Training Data", "Test Data", "Train Pred", "Test Pred",
    "Extended Pred"), col = c("blue", "green", "red", "orange", "purple"), pch = c(19,
    19, NA, NA, NA), lty = c(NA, NA, 1, 1, 1))
```



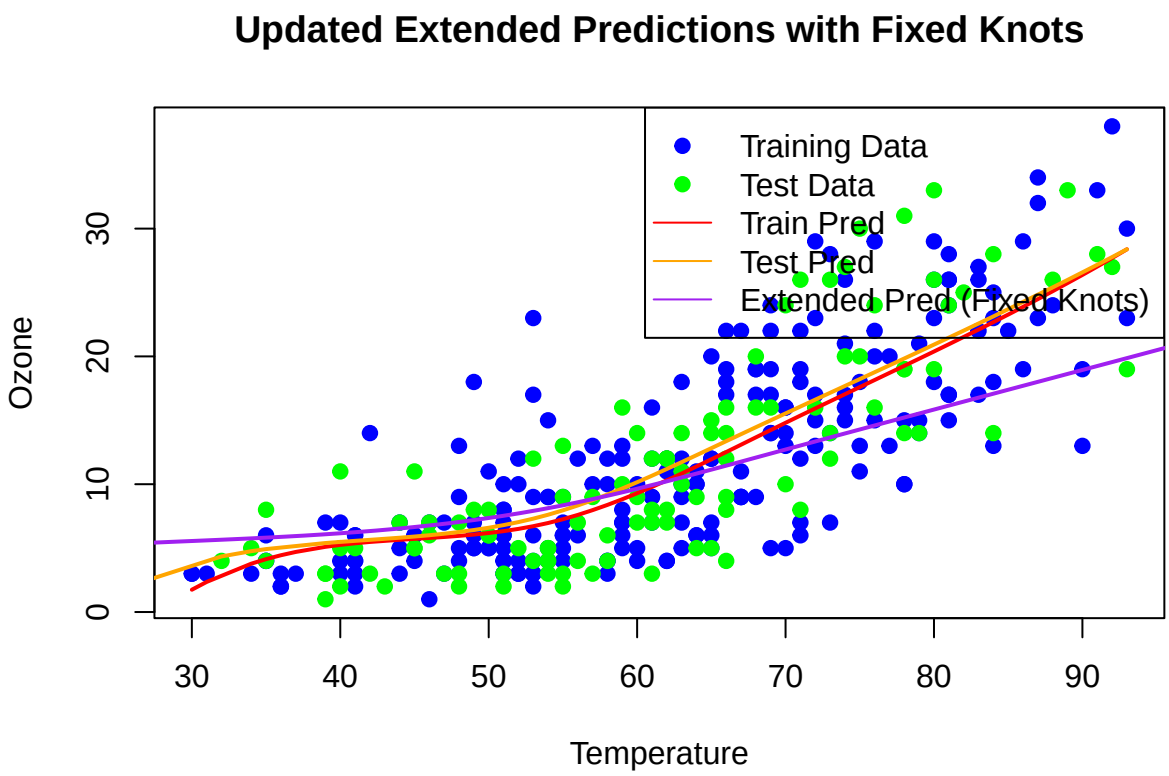
In this task, we extended the range of the explanatory variable (temperature) to include values from 0 to 120 using `seq(0, 120)`. The spline basis functions for this extended range were calculated using `lecturespl()`, which determines knots based on the quantiles of the input data. As a result, the knots for this task were placed at approximately 40 and 80.

The predictions for the extended range align well with the general trend observed in the training and test data. This suggests that the quantile-based knots worked well in this scenario, as the extended range still resembles the distribution of the training data. The resulting plot includes the extended predictions (purple line), which behave reasonably and show no significant inconsistencies.

Task 5: Modify `lecturespl` for Training Knots

Adjust `lecturespl` to use training knots for predictions.

```
# Modified lecturespl function to use precomputed training knots
modified_lecturespl <- function(x, train_knots, order = 4) {
    # Create basis functions using precomputed knots from the training data
    X <- bs(x, knots = train_knots, degree = order - 1, intercept = TRUE)
    list(X = X, x = x)
}

# Extract training knots from the training spline basis
train_knots <- attr(basis$X, "knots")  # Save the knots from the training data

# Generate new basis functions for the extended range using training knots
new_basis_fixed <- modified_lecturespl(new_temps, train_knots, order = 4)

# Create a data frame for the updated basis functions
new_model_data_fixed <- as.data.frame(new_basis_fixed$X)

# Predict ozone levels for the extended temperature range using fixed knots
new_preds_fixed <- predict(model, newdata = new_model_data_fixed)

# Visualize the updated results
plot(train_sorted$Temp, train_sorted$Ozone, col = "blue", pch = 19, xlab = "Temperature",
    ylab = "Ozone", main = "Updated Extended Predictions with Fixed Knots")
points(test_sorted$Temp, test_sorted$Ozone, col = "green", pch = 19)
lines(train_sorted$Temp, train_pred_sorted, col = "red", lwd = 2)
lines(test_sorted$Temp, test_pred_sorted, col = "orange", lwd = 2)
lines(new_temps, new_preds_fixed, col = "purple", lwd = 2)
legend("topright", legend = c("Training Data", "Test Data", "Train Pred", "Test Pred",
    "Extended Pred (Fixed Knots)"), col = c("blue", "green", "red", "orange", "purple"),
    pch = c(19, 19, NA, NA, NA), lty = c(NA, NA, 1, 1, 1))
```



To ensure robustness and consistency, this task reused the knots derived from the training data instead of recalculating them for the extended range. The knots from the training data, located at approximately 54 and 69, were applied to the extended range (0 to 120) using a modified `lecturespl` function. This approach avoids dependency on the distribution of new data and ensures consistent predictions across all scenarios.

The predictions for the extended range using fixed training knots align closely with those from Task 4. This is expected because the extended range (0 to 120) overlaps significantly with the training range (30 to 90), and splines inherently provide smooth extrapolation. However, using fixed knots eliminates potential variability introduced by recalculating knots, making it the more robust and methodologically sound approach.

```
# Knots from Task 4
knots_task4 <- attr(new_basis$X, "knots")

# Knots from Task 5
knots_task5 <- train_knots

# Compare the two
all.equal(knots_task4, knots_task5)
```

```
## [1] "Mean relative difference: 0.2083333"
```

```
print(knots_task4)
```

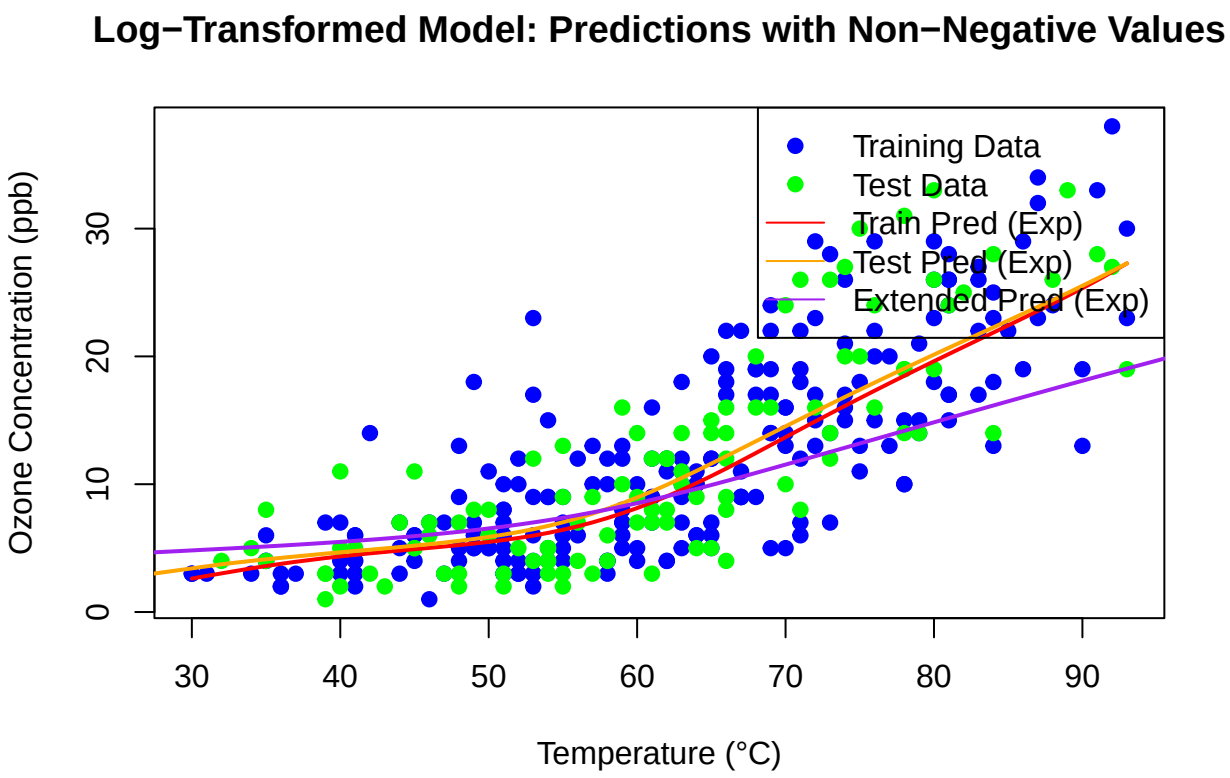
```
## 33.33333% 66.66667%
##        40        80
```

```r
print(knots_task5)
```

```
## 33.33333% 66.66667%
##        54        69
```

Task 5: Log-Transformed Response

```r
## Task 5: Log-Transformed Response

# Add log-transformed Ozone to the training data
train_model_data <- as.data.frame(train_basis_sorted$X)  # Recreate predictors for training
train_model_data$log_Ozone <- log(train_sorted$Ozone)  # Add log-transformed response

# Fit the model using the log-transformed response
log_model <- lm(log_Ozone ~ ., data = train_model_data)

# Predict for training data and back-transform to the original scale
log_train_pred <- predict(log_model, newdata = train_model_data)
train_pred_exp <- exp(log_train_pred)

# Predict for test data and back-transform to the original scale
test_model_data <- as.data.frame(test_basis_sorted$X)  # Recreate predictors for test
log_test_pred <- predict(log_model, newdata = test_model_data)
test_pred_exp <- exp(log_test_pred)

# Predict for the extended range and back-transform to the original scale
log_new_pred <- predict(log_model, newdata = new_model_data_fixed)
new_pred_exp <- exp(log_new_pred)

# Visualize the results
plot(train_sorted$Temp, train_sorted$Ozone, col = "blue", pch = 19, xlab = "Temperature (°C)",
    ylab = "Ozone Concentration (ppb)", main = "Log-Transformed Model: Predictions with Non-Negative Val
points(test_sorted$Temp, test_sorted$Ozone, col = "green", pch = 19)
lines(train_sorted$Temp, train_pred_exp, col = "red", lwd = 2)
lines(test_sorted$Temp, test_pred_exp, col = "orange", lwd = 2)
lines(new_temps, new_pred_exp, col = "purple", lwd = 2)
legend("topright", legend = c("Training Data", "Test Data", "Train Pred (Exp)", "Test Pred (Exp)",
    "Extended Pred (Exp)"), col = c("blue", "green", "red", "orange", "purple"),
    pch = c(19, 19, NA, NA, NA), lty = c(NA, NA, 1, 1, 1))
```


Log−Transformed Model: Predictions with Non−Negative Values
Training Data
Test Data
Train Pred (Exp)
Test Pred (Exp)
Extended Pred (Exp)
Ozone Concentration (ppb)
0
10
20
30
30
40
50
60
70
80
90
Temperature (°C)