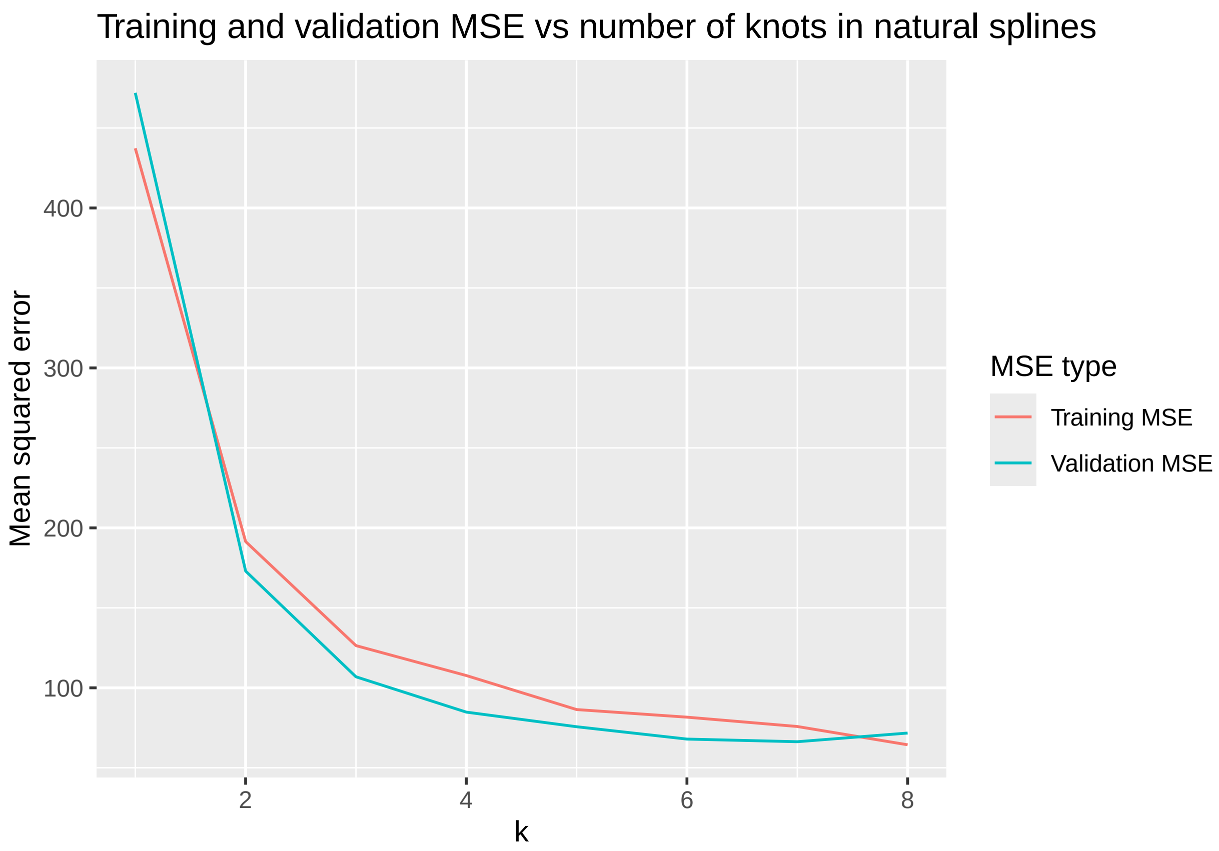
HW4

2024-11-17

Natural Spline Fitting

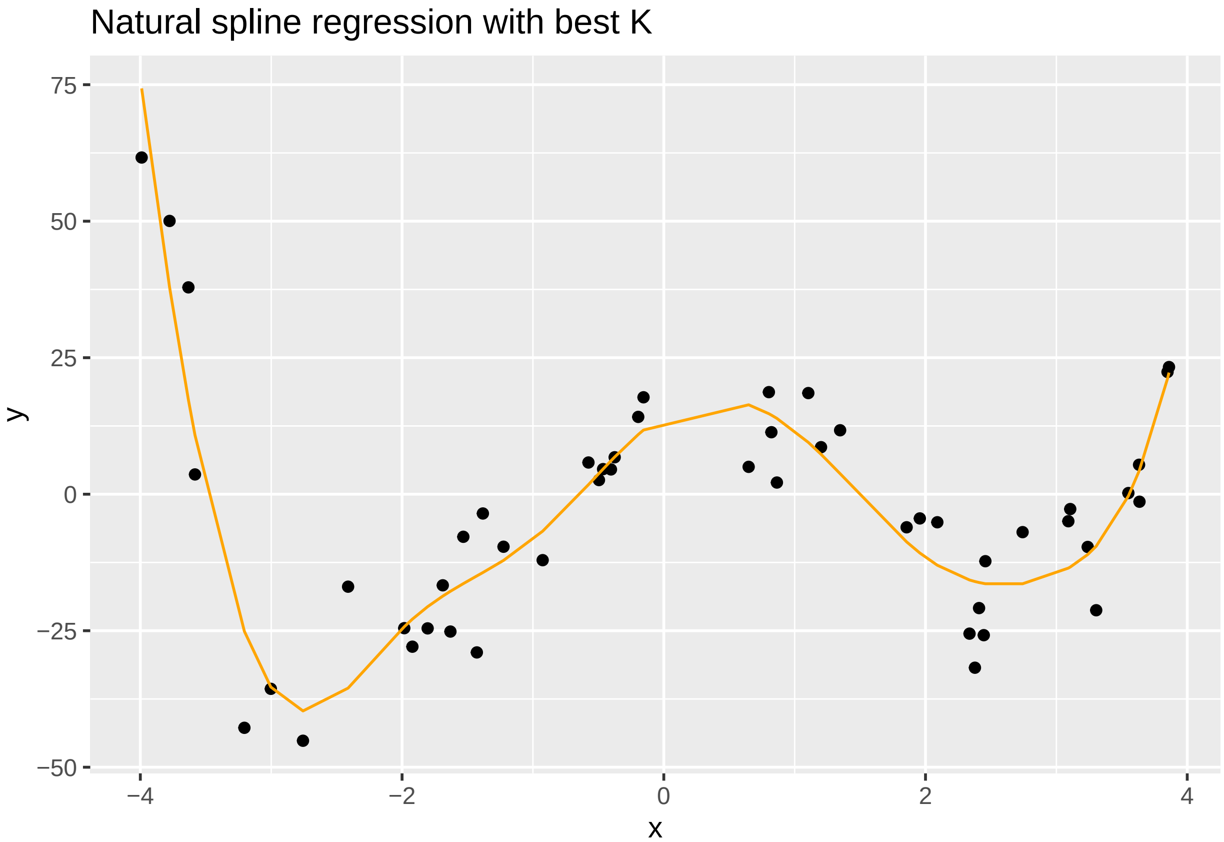
```r
natural_spline <- function(x, k) {
  knots <- seq(-4, 4, length.out = k + 2)[-c(1, k + 2)]

  basis <- sapply(knots, function(kn) pmax(x - kn, 0)^3)

  return(cbind(1, x, x^2, basis))
}

k_values <- 1:8
train_mse <- c()
valid_mse <- c()

for (k in k_values) {
  X_train_natural <- natural_spline(train_data$x, k)

  beta_natural <- solve(t(X_train_natural) %*% X_train_natural) %*% (t(X_train_natural) %*% train_data$y

  train_pred_natural <- X_train_natural %*% beta_natural
  train_mse <- c(train_mse, calculate_mse(train_data$y, train_pred_natural))

  X_valid_natural <- natural_spline(validation_data$x, k)

  valid_pred_natural <- X_valid_natural %*% beta_natural

  valid_mse <- c(valid_mse, calculate_mse(validation_data$y, valid_pred_natural))
}

mse_df <- data.frame(k = k_values, train_mse = train_mse, valid_mse = valid_mse)

ggplot(mse_df, aes(x=k)) +
  geom_line(aes(y=train_mse, color="Training MSE")) +
  geom_line(aes(y=valid_mse, color="Validation MSE")) +
  labs(y="Mean squared error", color="MSE type") +
  ggtitle("Training and validation MSE vs number of knots in natural splines")
```



```r
best_k <- k_values[which.min(valid_mse)]

cat("Best K:", best_k, "\n")
```

```
## Best K: 7
```

```r
final_natural_spline <- natural_spline(data$x, best_k)
final_natural_coeff <- solve(t(final_natural_spline) %*% final_natural_spline) %*% (t(final_natural_spli

data$y_pred_natural_final <- final_natural_spline %*% final_natural_coeff

ggplot(data, aes(x=x, y=y)) +
  geom_point() +
  geom_line(aes(y=y_pred_natural_final), color="orange") +
  ggtitle("Natural spline regression with best K")
```


Natural spline regression with best K
y
75
50
25
0
−25
−50
−4
−2
0
2
4
x