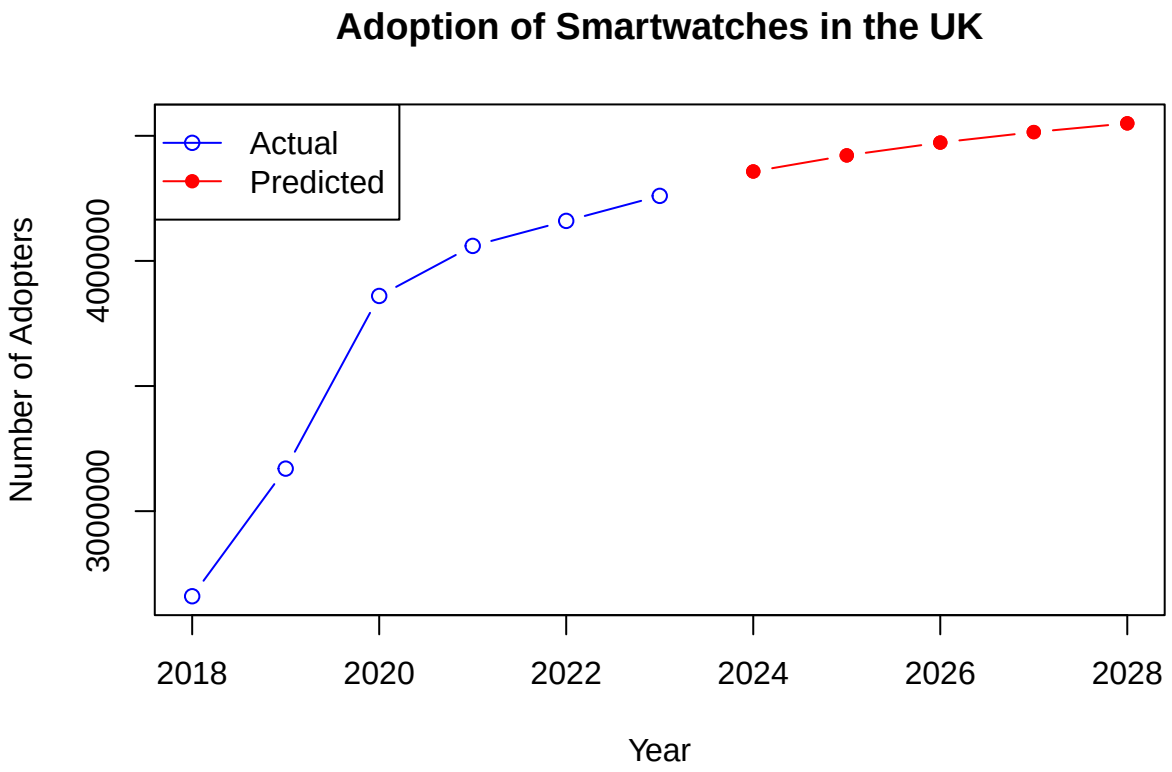
```r
# Load the necessary packages
library(minpack.lm)

# Smartwatch adoption data for the UK from 2018 to 2023
years <- c(2018:2023)
users <- c(2.66, 3.17, 3.86, 4.06, 4.16, 4.26)  # Users in millions

# Create a data frame from the yearly data
smartwatch_data <- data.frame(Year = years, Adopters = users * 1e6)  # Convert users from millions to a

# Define the Bass diffusion model function
bass_model <- function(t, m, p, q) {
  # The function models the adoption of a new product based on time t and parameters m, p, and q
  m * (1 - exp(-(p + q) * t)) / (1 + q / p * exp(-(p + q) * t))
}

# Initial guesses for the Bass model parameters
m_guess <- max(smartwatch_data$Adopters) * 1.2  # Market potential is 20% higher than the current max
p_guess <- 0.03  # Initial guess for the coefficient of innovation
q_guess <- 0.38  # Initial guess for the coefficient of imitation

# Fit the Bass model to the actual data using non-linear least squares
nls_fit <- nlsLM(Adopters ~ bass_model(Year - min(Year) + 1, m, p, q),
                 data = smartwatch_data,
                 start = list(m = m_guess, p = p_guess, q = q_guess),
                 control = nls.lm.control(maxiter = 500))

# Extract the estimated parameters from the fitted model
params <- coef(nls_fit)

# Predict future adoption using the estimated parameters for years 2024 to 2028
future_years <- seq(from = 2024, to = 2028)
predicted_adopters <- sapply(future_years, function(t) {
  bass_model(t - min(years) + 1, params["m"], params["p"], params["q"])
})

# Create a data frame for the predicted number of adopters
predictions <- data.frame(Year = future_years, Adopters = predicted_adopters)

# ... (the previous code remains the same)

# Plot the actual number of adopters
plot(smartwatch_data$Year, smartwatch_data$Adopters, type = "b", col = "blue",
     xlab = "Year", ylab = "Number of Adopters", main = "Adoption of Smartwatches in the UK",
     ylim = range(c(smartwatch_data$Adopters, predictions$Adopters)), xlim = c(2018, 2028))  # Set y-ax

# Add the predicted number of adopters to the plot with red color and circle type
lines(predictions$Year, predictions$Adopters, type = "b", col = "red", pch=16)
points(predictions$Year, predictions$Adopters, type = "p", col = "red", pch=16)

# Add a legend to the plot for clarity
legend("topleft", legend = c("Actual", "Predicted"), col = c("blue", "red"), lty = 1, pch = c(1, 16))
```




```
# Print the estimated parameters and the predicted values
print(params)
```

```
##             m             p             q
##  4.930472e+06  1.086209e+00 -1.086161e+00
```

```
print(predictions)
```

```
##   Year Adopters
## 1 2024  4357478
## 2 2025  4421723
## 3 2026  4473016
## 4 2027  4514915
## 5 2028  4549785
```