gdp-ntl-model

2024-07-02

```r
# Load necessary libraries
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

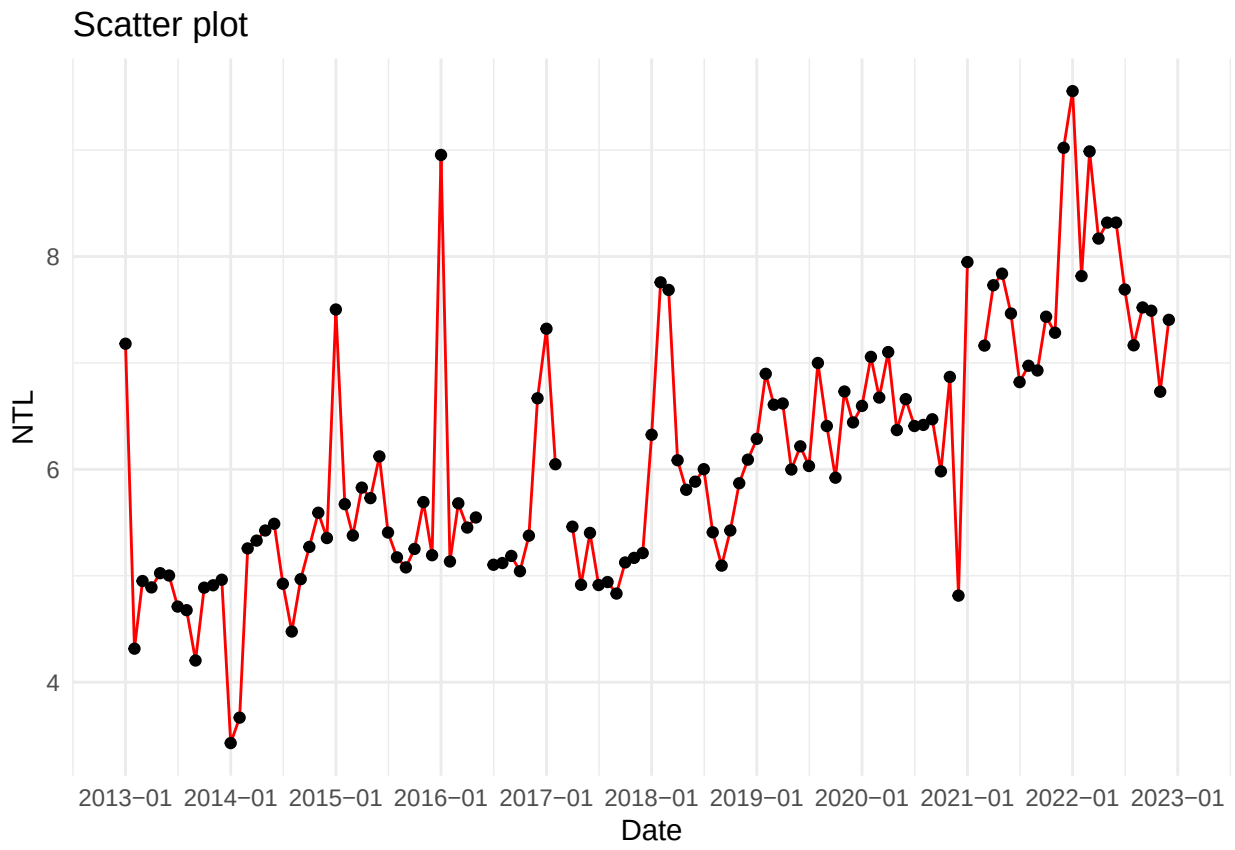
```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
# Load the data
data <- read.csv("gdp_ntl_merged.csv")

data$Date <- paste0(data$Date, "-01")
data$Date <- as.Date(data$Date)

data$log_gdp <- log(data$gdp)
data$log_NTL <- log(data$NTL)



ggplot(data, aes(x = Date)) +
  geom_line(aes(y = NTL), color = "red") +
  geom_point(aes(y = NTL)) +
  labs(x = "Date", y = "NTL", title = "Scatter plot") +
  scale_x_date(breaks = seq(as.Date("2013-01-01"), as.Date("2023-01-01"), by = "year"), date_labels = "
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

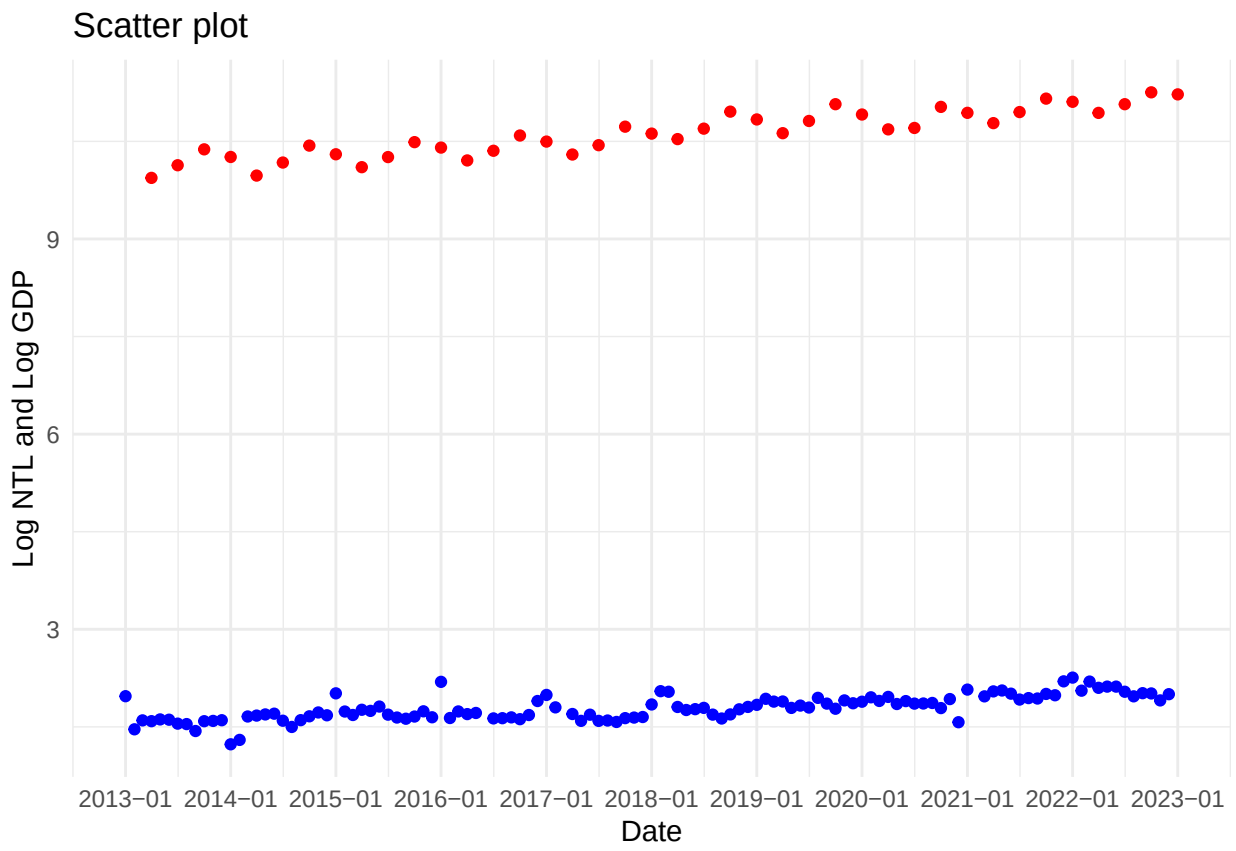
```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



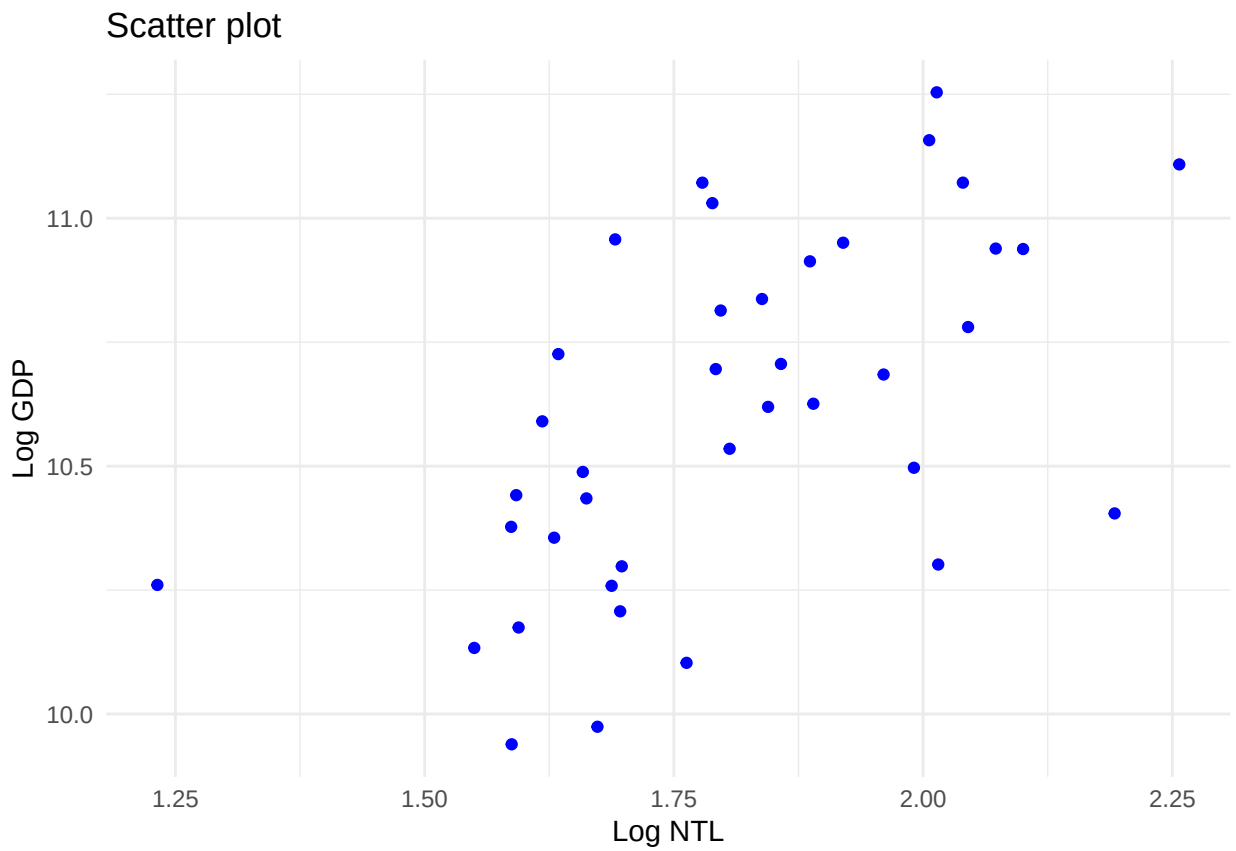

```
ggplot(data, aes(x=Date)) +
  geom_point(aes(y=log_NTL), color="blue") +
  geom_point(aes(y=log_gdp), color="red") +
  labs(x = "Date", y = "Log NTL and Log GDP", title = "Scatter plot") +
  scale_x_date(breaks = seq(as.Date("2013-01-01"), as.Date("2023-01-01"), by = "year"), date_labels = "
  theme_minimal()
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

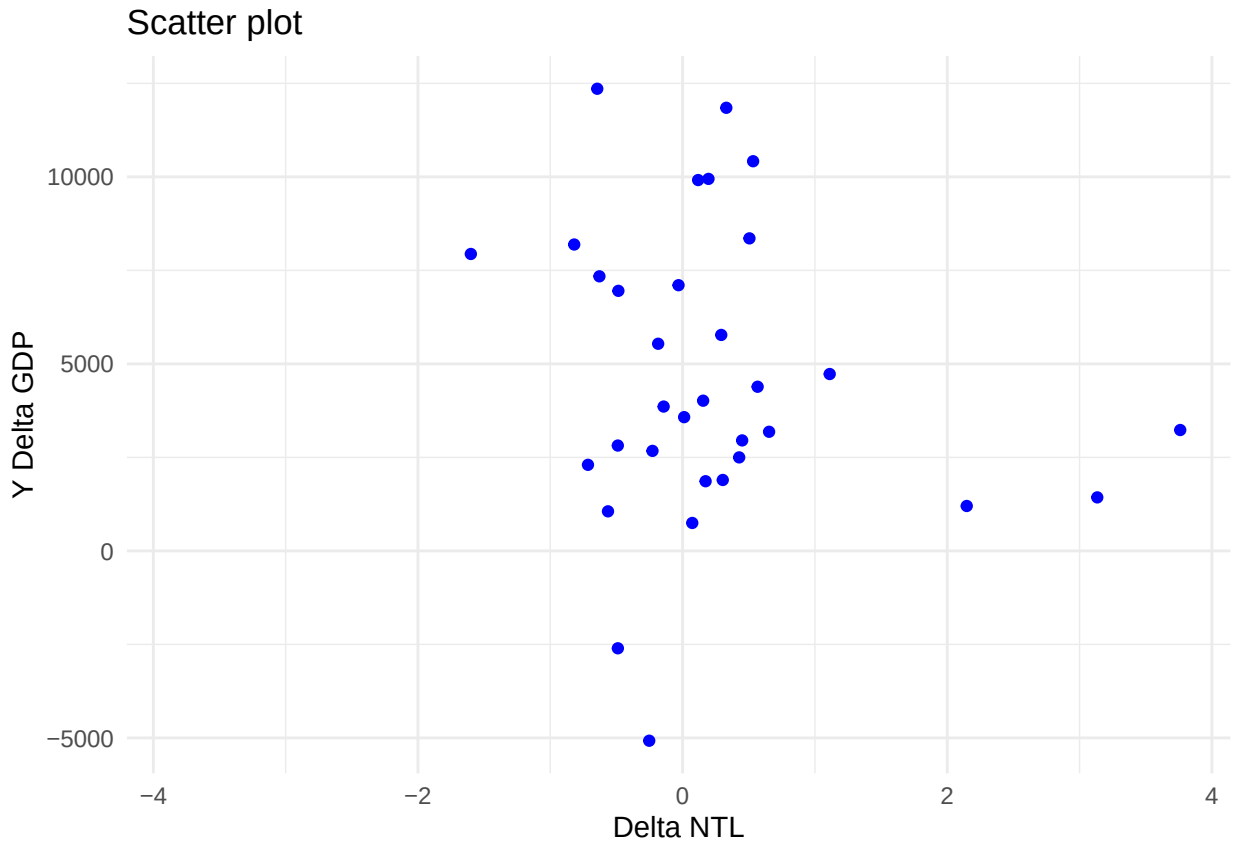
```
## Warning: Removed 81 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



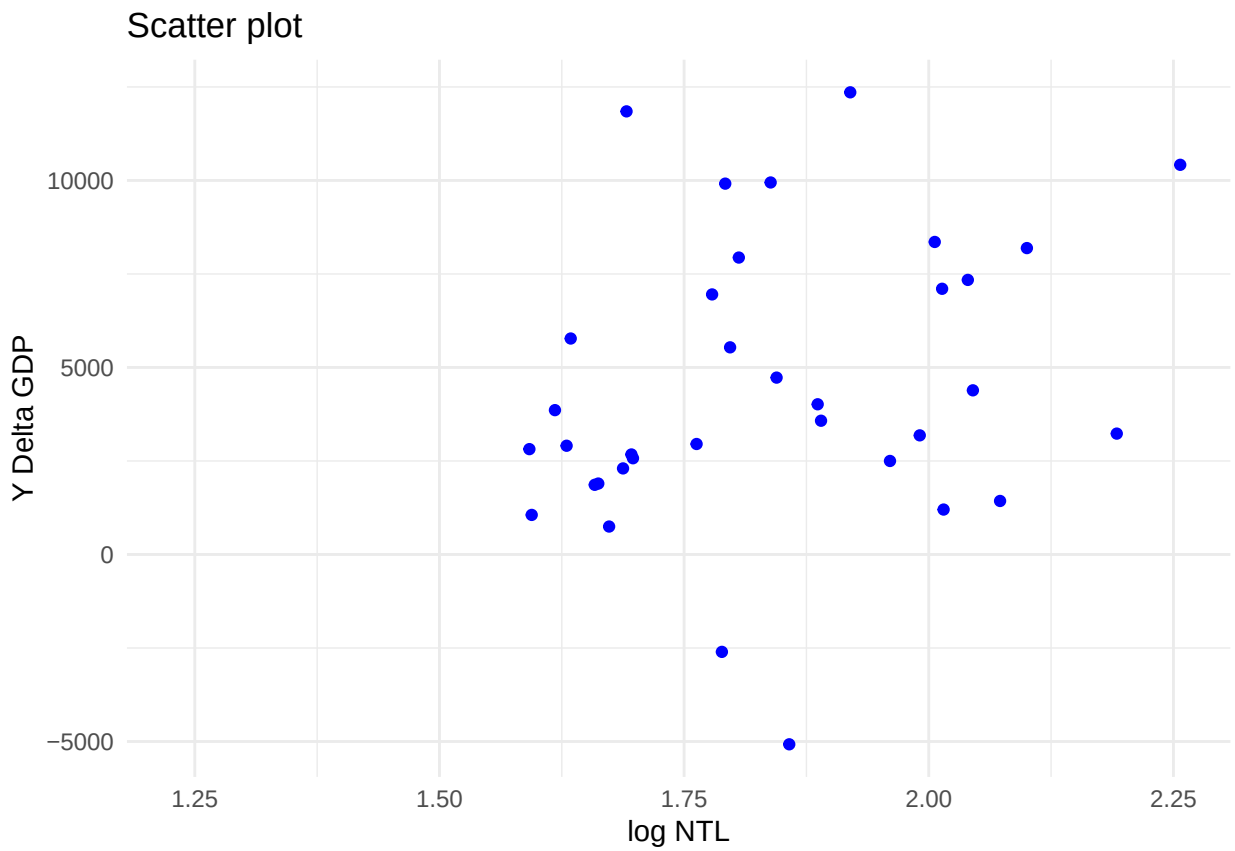

```r
data_clean <- data[complete.cases(data), ]



# Create the plot
ggplot() +
  geom_point(data = data_clean, aes(x = log_NTL, y = log_gdp), color = "blue", size = 1.5) +
  #geom_line(data = data_clean, aes(x = Date, y = predicted_values), color = "red", size = 1) +
  labs(x = "Log NTL", y = "Log GDP", title = "Scatter plot") +
  theme_minimal()
```




```
combined_data <- read.csv("combined_data.csv")


start_date <- as.Date("2013-04-01")
end_date <- as.Date("2023-01-01")

# Create a sequence of dates by quarter
date_sequence <- seq.Date(from = start_date, to = end_date, by = "quarter")


temp_data <- data.frame(Date = date_sequence, y_delta_gdp = combined_data$y_delta_gdp)


merged_df <- merge(data, temp_data, by = "Date", all.x = TRUE)

diff_ntl <- diff(merged_df$NTL)

merged_df$delta_NTL <- c(NA, diff_ntl)


# Create the plot
ggplot() +
  geom_point(data = merged_df, aes(x = delta_NTL, y = y_delta_gdp), color = "blue", size = 1.5) +
  #geom_line(data = data_clean, aes(x = Date, y = predicted_values), color = "red", size = 1) +
  labs(x = "Delta NTL", y = "Y Delta GDP", title = "Scatter plot") +
  theme_minimal()
```




```
# Create the plot
ggplot() +
  geom_point(data = merged_df, aes(x = log_NTL, y = y_delta_gdp), color = "blue", size = 1.5) +
  #geom_line(data = data_clean, aes(x = Date, y = predicted_values), color = "red", size = 1) +
  labs(x = "log NTL", y = "Y Delta GDP", title = "Scatter plot") +
  theme_minimal()
```

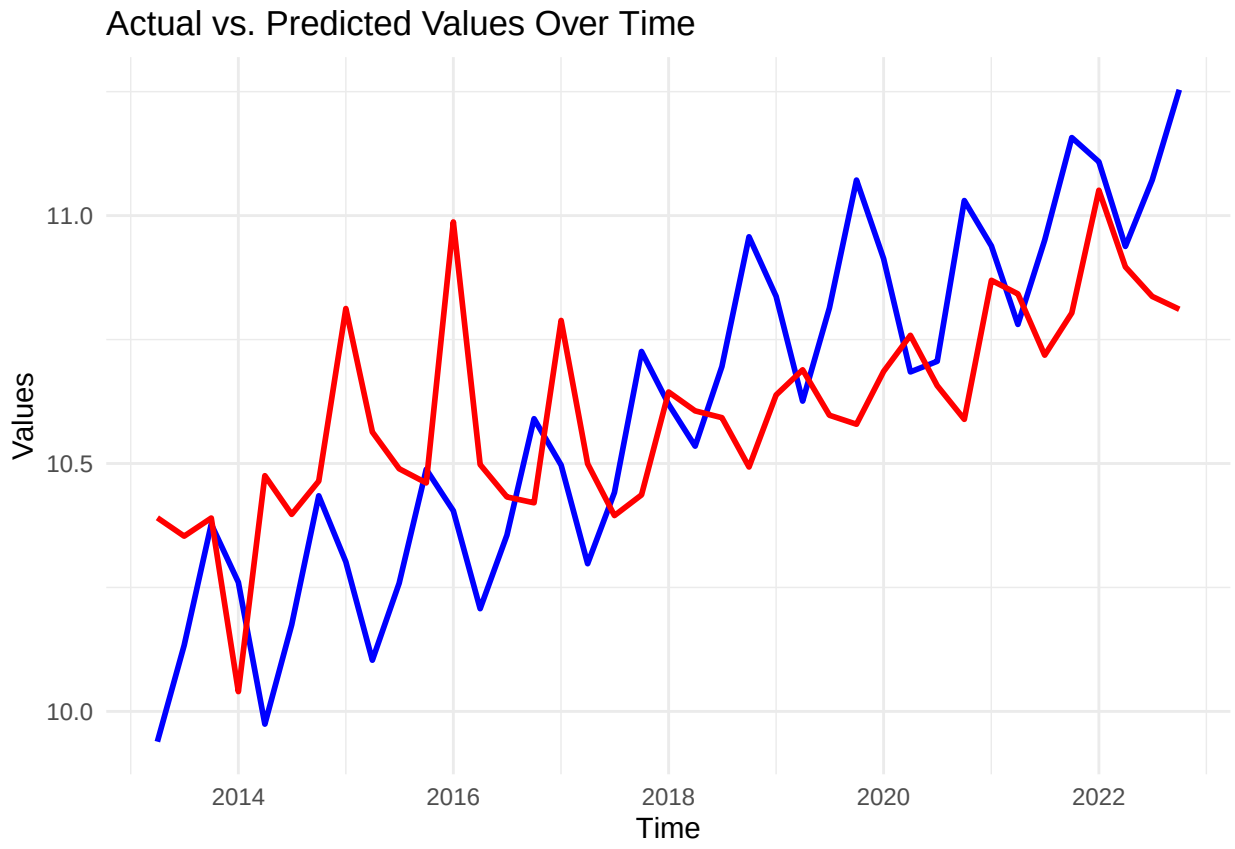



```r
# Perform linear regression
model <- lm(log_gdp ~ log_NTL, data = data_clean)

# Print the summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = log_gdp ~ log_NTL, data = data_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58254 -0.21084  0.02732  0.21848  0.49247
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.8247     0.4096  21.545  < 2e-16 ***
## log_NTL       0.9865     0.2253   4.378 9.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2877 on 37 degrees of freedom
## Multiple R-squared:  0.3413, Adjusted R-squared:  0.3235
## F-statistic: 19.17 on 1 and 37 DF,  p-value: 9.441e-05
```

```r
data_clean$predicted_values <- predict(model, newdata = data_clean)

# Create the plot
ggplot() +
  geom_line(data = data_clean, aes(x = Date, y = log_gdp), color = "blue", size = 1) +
  geom_line(data = data_clean, aes(x = Date, y = predicted_values), color = "red", size = 1) +
  labs(x = "Time", y = "Values", title = "Actual vs. Predicted Values Over Time") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```r
# Load the necessary library
library(midasr)
```

```
## Loading required package: sandwich
```

```
## Loading required package: optimx
```

```
## Loading required package: quantreg
```

```
## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# Generate synthetic quarterly data for the dependent variable
set.seed(123)
n_quarters <- 40
y <- na.omit(data$log_gdp)

# Generate synthetic monthly data for the independent variable
n_months <- n_quarters * 3
x <- data$log_NTL[-1]

# Define the date sequence
quarter_dates <- seq.Date(from = as.Date("2013-04-01"), by = "quarter", length.out = n_quarters)
month_dates <- seq.Date(from = as.Date("2013-04-01"), by = "month", length.out = n_months)

# Create data frames
quarter_data <- data.frame(Date = quarter_dates, y = na.omit(data$log_gdp))
month_data <- data.frame(Date = month_dates, x = data$log_NTL[-1])

start_values <- list(x = rep(0, 12))

# Define the MIDAS regression model
#midas_model <- midas_r(y ~ mls(x, 0:11, 3), data = list(y = y, x = x))

midas_model <- midas_r(y ~ mls(x, 0:11, 3), data = list(y = y, x = x), start = start_values)
```

```
## Warning in midas_r(y ~ mls(x, 0:11, 3), data = list(y = y, x = x), start =
## start_values): There are NAs in the middle of the time series
```

```
# Summarize the model
summary(midas_model)
```

```
##
## MIDAS regression model with "numeric" data:
## Start = 4, End = 39
##
##  Formula y ~ mls(x, 0:11, 3)
##
##  Parameters:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.48497    0.30875  24.243 1.46e-11 ***
```

```
## x1          -0.43737    0.07307  -5.986 6.35e-05 ***
## x2          -0.56286    0.26572  -2.118 0.055707 .
## x3           1.08583    0.16879   6.433 3.24e-05 ***
## x4          -0.59666    0.13202  -4.519 0.000702 ***
## x5           0.65301    0.26006   2.511 0.027356 *
## x6           0.34656    0.45567   0.761 0.461605
## x7          -0.18710    0.34902  -0.536 0.601703
## x8           0.67625    0.30483   2.218 0.046567 *
## x9           0.49987    0.54559   0.916 0.377607
## x10         -0.49438    0.36814  -1.343 0.204152
## x11          0.67655    0.28765   2.352 0.036579 *
## x12          0.12977    0.11634   1.115 0.286516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Residual standard error: 0.1513 on 12 degrees of freedom
```

```
# Summarize the model
summary(midas_model)
```

```
##
## MIDAS regression model with "numeric" data:
## Start = 4, End = 39
##
##  Formula y ~ mls(x, 0:11, 3)
##
##  Parameters:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.48497    0.30875  24.243 1.46e-11 ***
## x1          -0.43737    0.07307  -5.986 6.35e-05 ***
## x2          -0.56286    0.26572  -2.118 0.055707 .
## x3           1.08583    0.16879   6.433 3.24e-05 ***
## x4          -0.59666    0.13202  -4.519 0.000702 ***
## x5           0.65301    0.26006   2.511 0.027356 *
## x6           0.34656    0.45567   0.761 0.461605
## x7          -0.18710    0.34902  -0.536 0.601703
## x8           0.67625    0.30483   2.218 0.046567 *
## x9           0.49987    0.54559   0.916 0.377607
## x10         -0.49438    0.36814  -1.343 0.204152
## x11          0.67655    0.28765   2.352 0.036579 *
## x12          0.12977    0.11634   1.115 0.286516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Residual standard error: 0.1513 on 12 degrees of freedom
```

```
predicted_values <- predict(midas_model)
```

```
# Generate predicted values
predicted_values <- predict(midas_model)

# Calculate the number of initial NA values due to lag
n_lags <- length(predicted_values)
```

```r
y_trimmed <- y[(length(y) - n_lags + 1):length(y)]
dates_trimmed <- quarter_dates[(length(quarter_dates) - n_lags + 1):length(quarter_dates)]


results_df <- data.frame(
  Date = dates_trimmed,
  Actual = y_trimmed,
  Predicted = predicted_values
)

ggplot(results_df, aes(x = Date)) +
  geom_line(aes(y = Actual, color = "Actual")) +
  geom_line(aes(y = Predicted, color = "Predicted")) +
  labs(title = "Actual vs Predicted Values",
       x = "Date",
       y = "Value",
       color = "Legend") +
  theme_minimal()
```