Average price Forecast in Brent, Croydon, Kensington&Chelsea by type

Yuling Xue

2024-06-13

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# Load necessary libraries
library(forecast)
```

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

```
library(tseries)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

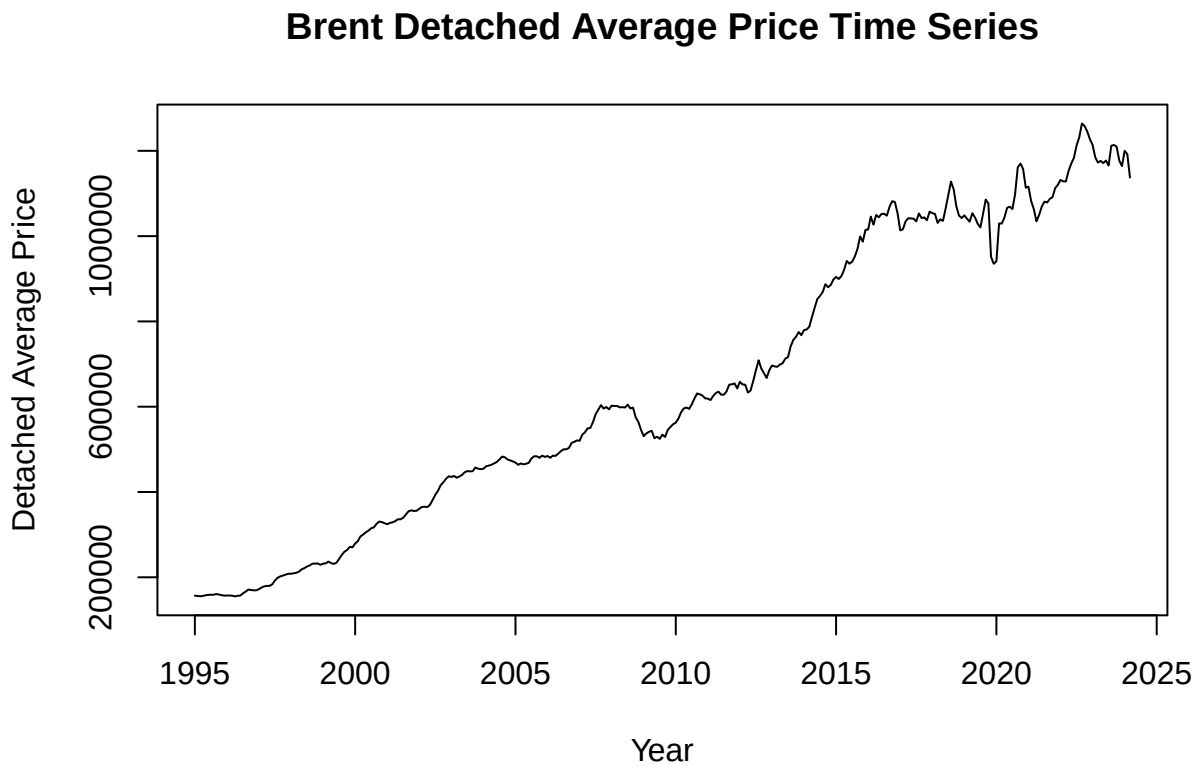
```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(Metrics)
```

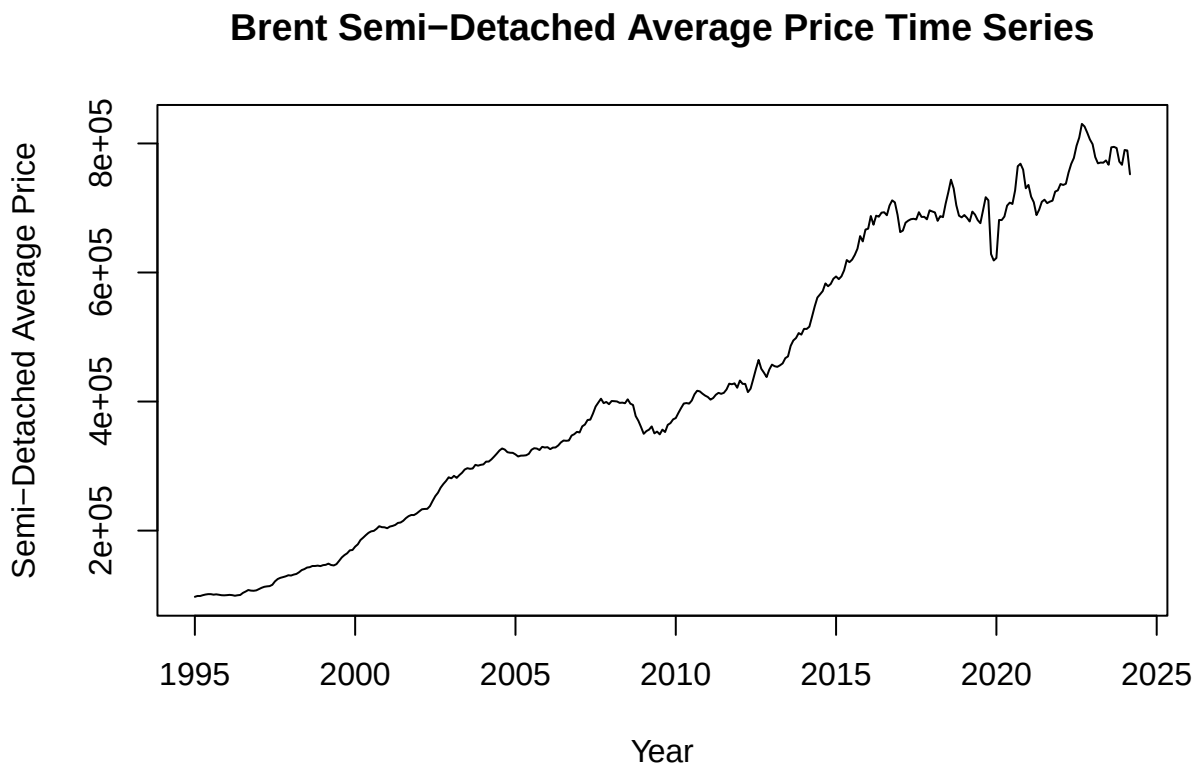
```
##
## Attaching package: 'Metrics'
```

```
## The following object is masked from 'package:forecast':
##
##     accuracy
```

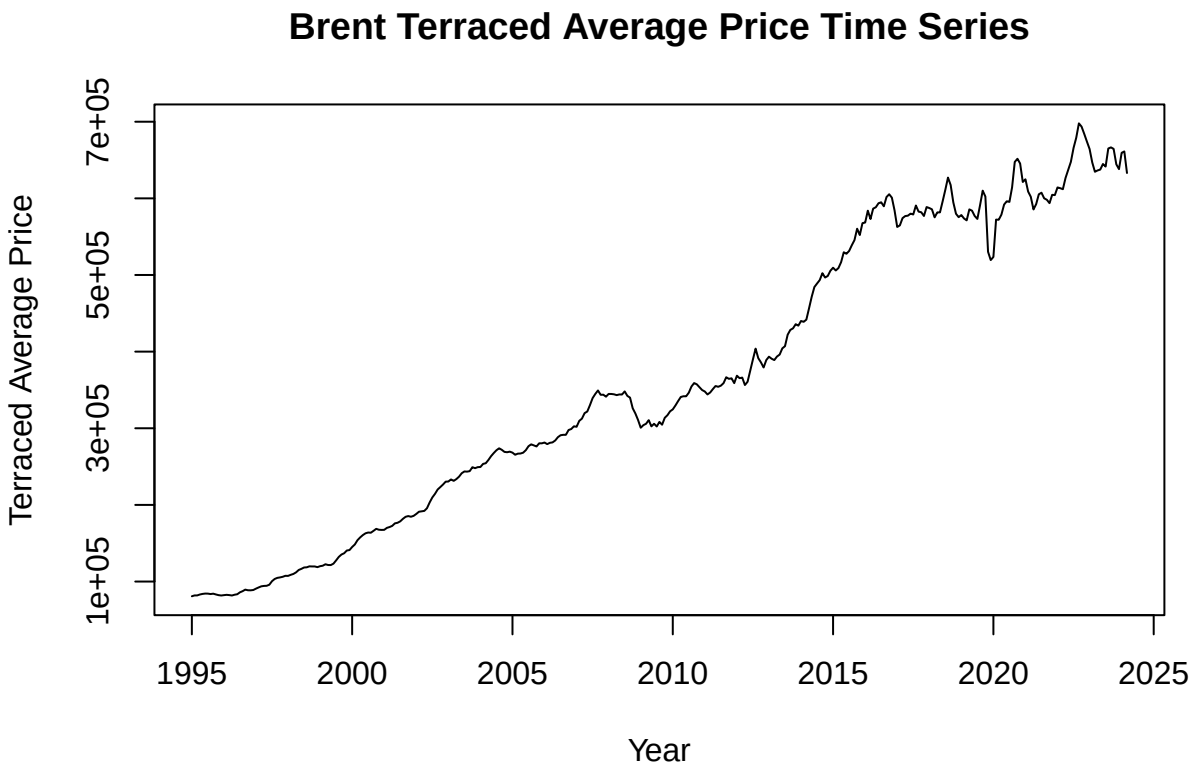
```r
### Brent Area
# Load the data of Brent Area
brent_data <- read.csv("~/Desktop/Updated_Brent_df.csv")

# Convert the Date column to Date type of Brent Area
brent_data$Date <- as.Date(brent_data$Date, format="%Y-%m-%d")

# Extract the Average Price for each property type and Date columns in Brent
brent_detached_price <- brent_data$Detached_Average_Price
brent_semi_detached_price <- brent_data$Semi_Detached_Average_Price
brent_terraced_price <- brent_data$Terraced_Average_Price
brent_flat_price <- brent_data$Flat_Average_Price
brent_dates <- brent_data$Date

# Create time series object for each property type of Brent Area
brent_detached_ts <- ts(brent_detached_price,
                        start = c(1995, 1), frequency = 12)
brent_semi_detached_ts <- ts(brent_semi_detached_price,
                             start = c(1995, 1), frequency = 12)
brent_terraced_ts <- ts(brent_terraced_price,
                        start = c(1995, 1), frequency = 12)
brent_flat_ts <- ts(brent_flat_price,
                    start = c(1995, 1), frequency = 12)

# Plot the time series of Brent Area
plot(brent_detached_ts, main = "Brent Detached Average Price Time Series",
     ylab = "Detached Average Price", xlab = "Year")
```



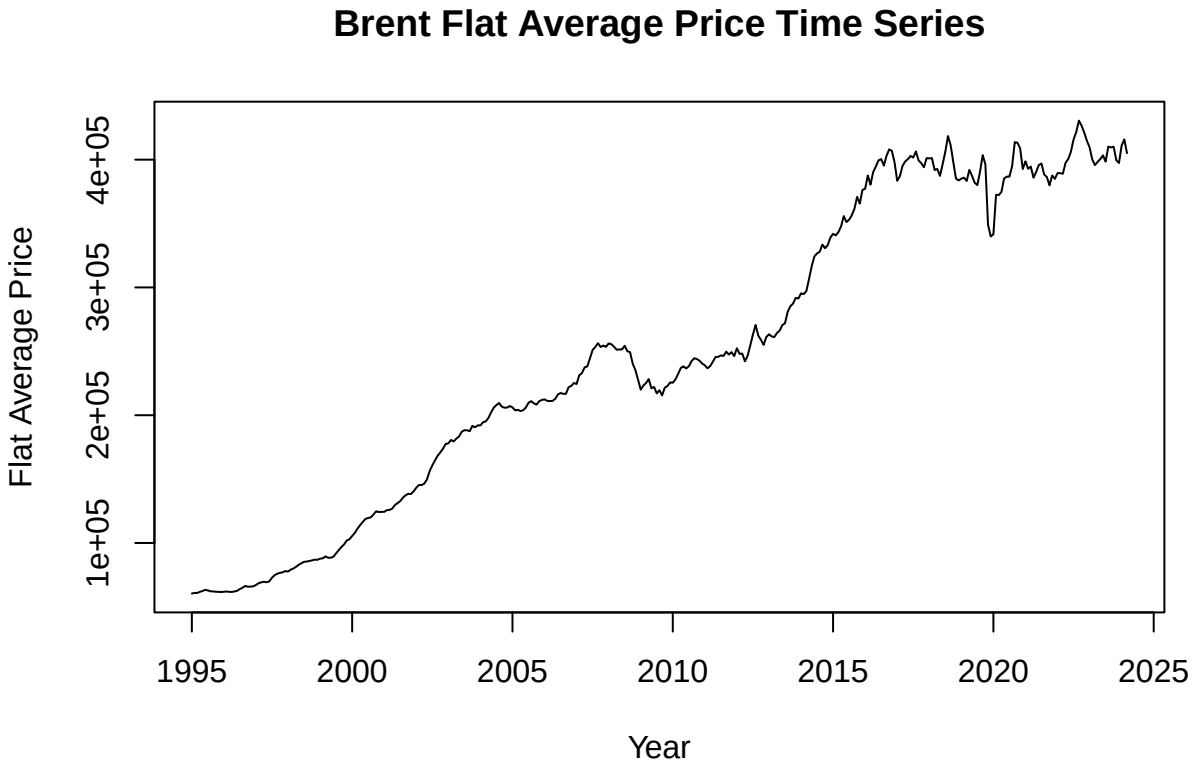
```
plot(brent_semi_detached_ts,
     main = "Brent Semi-Detached Average Price Time Series",
     ylab = "Semi-Detached Average Price", xlab = "Year")
```




```
plot(brent_terraced_ts, main = "Brent Terraced Average Price Time Series",
     ylab = "Terraced Average Price", xlab = "Year")
```




```
plot(brent_flat_ts, main = "Brent Flat Average Price Time Series",
     ylab = "Flat Average Price", xlab = "Year")
```



```r
# Calculate the p-value for the original series
#By using ADF test for each property type of Brent Area
adf_test_brent_d <- adf.test(brent_detached_ts)
adf_test_brent_sd <- adf.test(brent_semi_detached_ts)
adf_test_brent_t <- adf.test(brent_terraced_ts)
adf_test_brent_f <- adf.test(brent_flat_ts)
p_value_brent_d <- adf_test_brent_d$p.value
p_value_brent_sd <- adf_test_brent_sd$p.value
p_value_brent_t <- adf_test_brent_t$p.value
p_value_brent_f <- adf_test_brent_f$p.value
print(paste("Brent p-value:", p_value_brent_d, p_value_brent_sd,
            p_value_brent_t, p_value_brent_f))
```

```
## [1] "Brent p-value: 0.380002094355907 0.376728027975092 0.388089375457254 0.690335496911936"
```

```r
# Perform first-order differencing of Brent Area
brent_detached_ts_diff <- diff(brent_detached_ts)
brent_semi_detached_ts_diff <- diff(brent_semi_detached_ts)
brent_terraced_ts_diff <- diff(brent_terraced_ts)
brent_flat_ts_diff <- diff(brent_flat_ts)

# Perform ADF test on differenced series of Brent Area
adf_test_diff_brent_d <- adf.test(brent_detached_ts_diff)
```

```
## Warning in adf.test(brent_detached_ts_diff): p-value smaller than printed
## p-value
```

```r
adf_test_diff_brent_sd <- adf.test(brent_semi_detached_ts_diff)
```

```
## Warning in adf.test(brent_semi_detached_ts_diff): p-value smaller than printed
## p-value
```

```r
adf_test_diff_brent_t <- adf.test(brent_terraced_ts_diff)
```

```
## Warning in adf.test(brent_terraced_ts_diff): p-value smaller than printed
## p-value
```

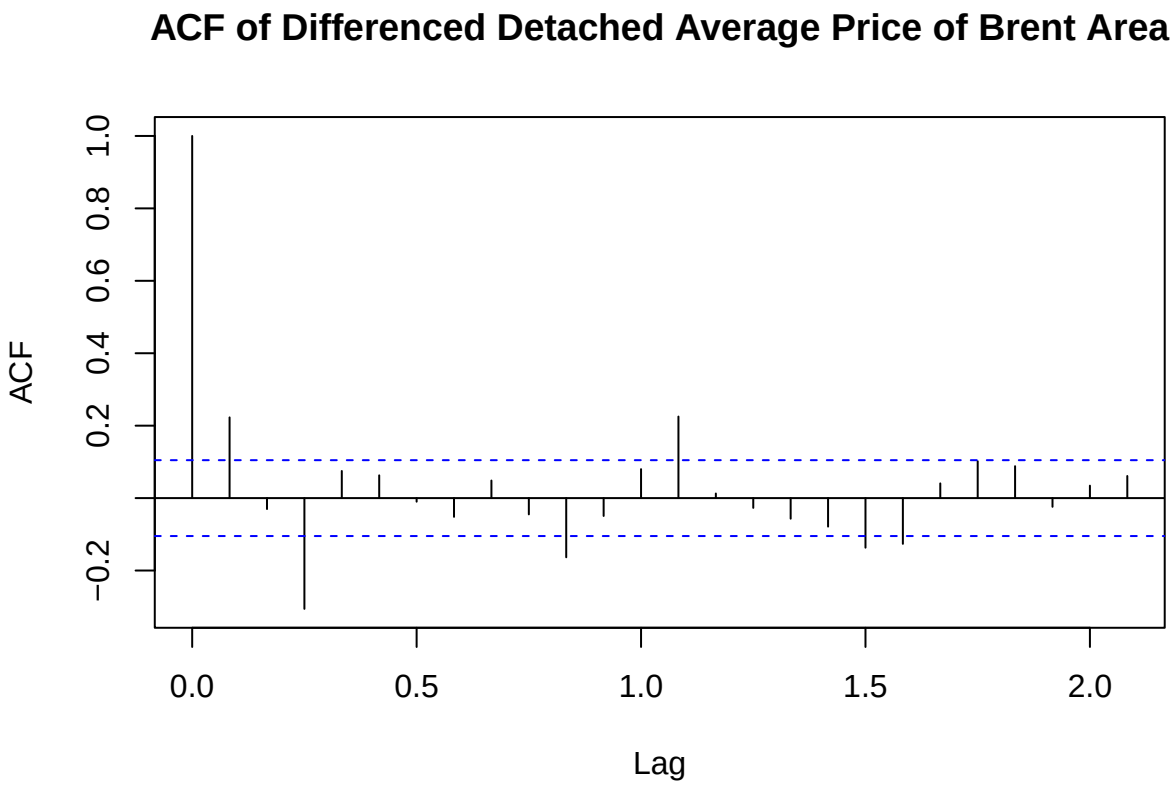
```r
adf_test_diff_brent_f <- adf.test(brent_flat_ts_diff)
```

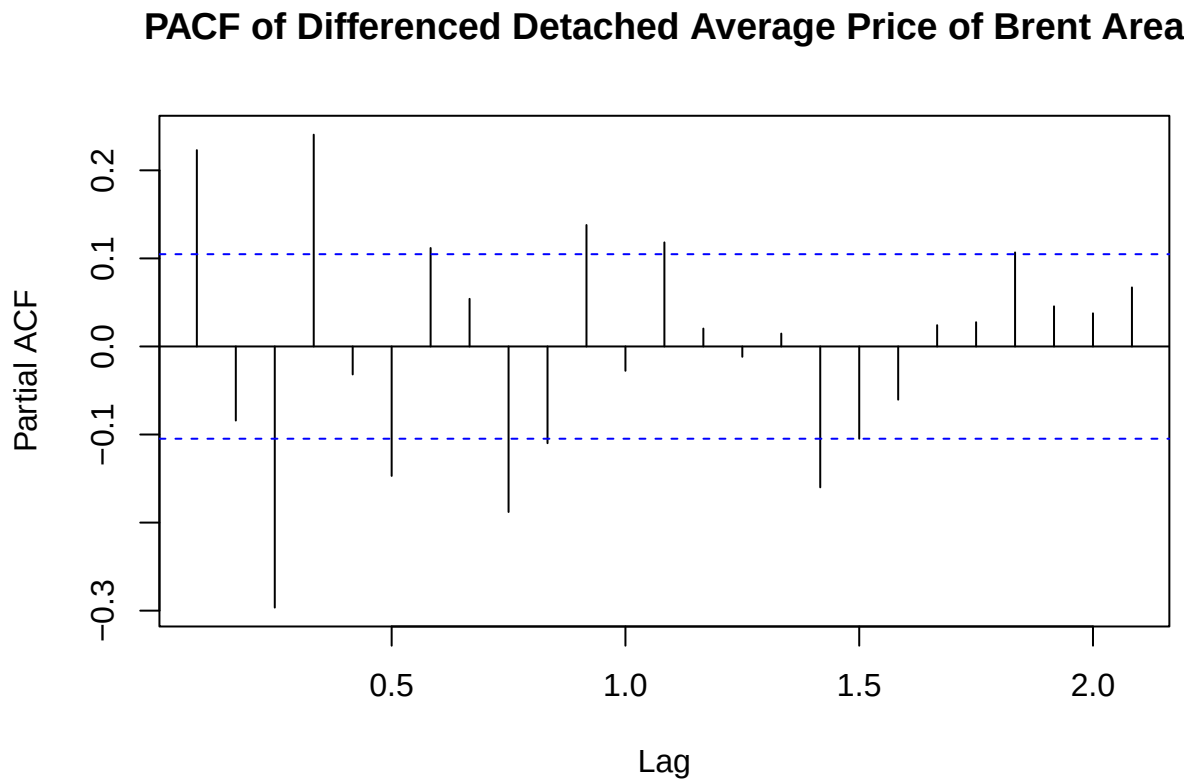
```
## Warning in adf.test(brent_flat_ts_diff): p-value smaller than printed p-value
```

```r
differenced_p_value_brent_d <- adf_test_diff_brent_d$p.value
differenced_p_value_brent_sd <- adf_test_diff_brent_sd$p.value
differenced_p_value_brent_t <- adf_test_diff_brent_t$p.value
differenced_p_value_brent_f <- adf_test_diff_brent_f$p.value
print(paste("Brent difference1_p-value:", differenced_p_value_brent_d,
            differenced_p_value_brent_sd, differenced_p_value_brent_t,
            differenced_p_value_brent_f))
```

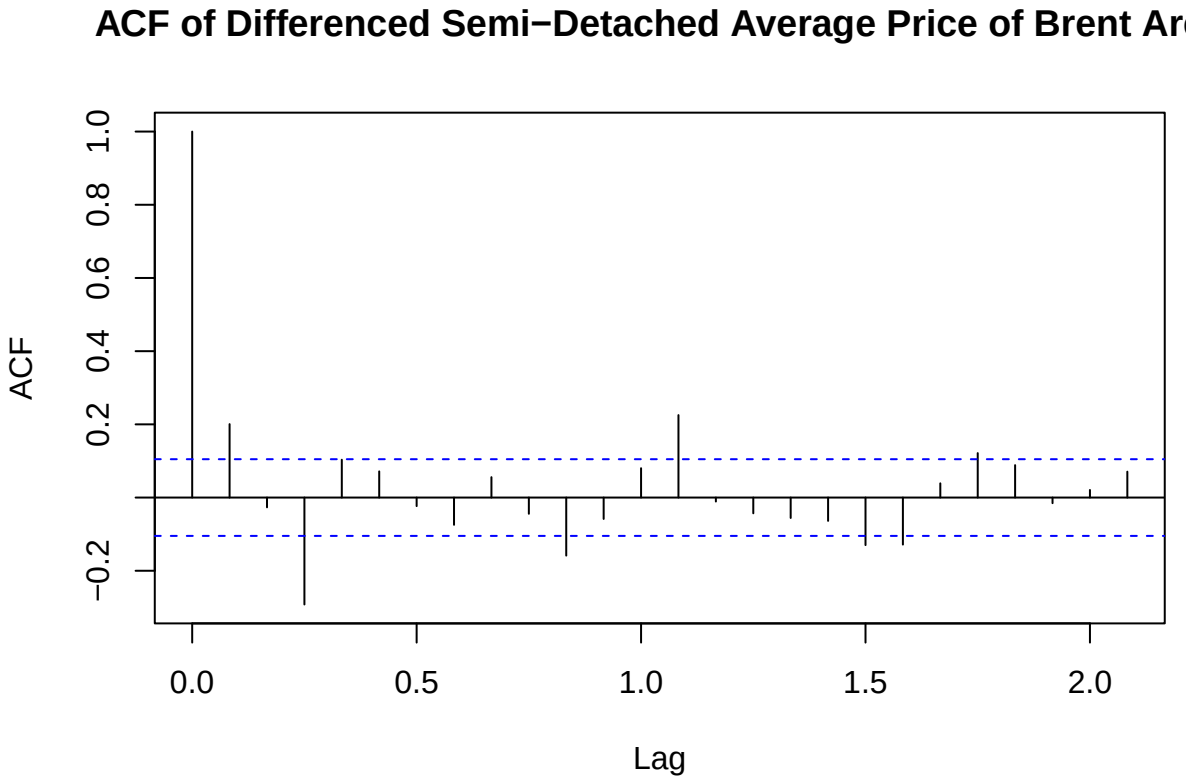
```
## [1] "Brent difference1_p-value: 0.01 0.01 0.01 0.01"
```

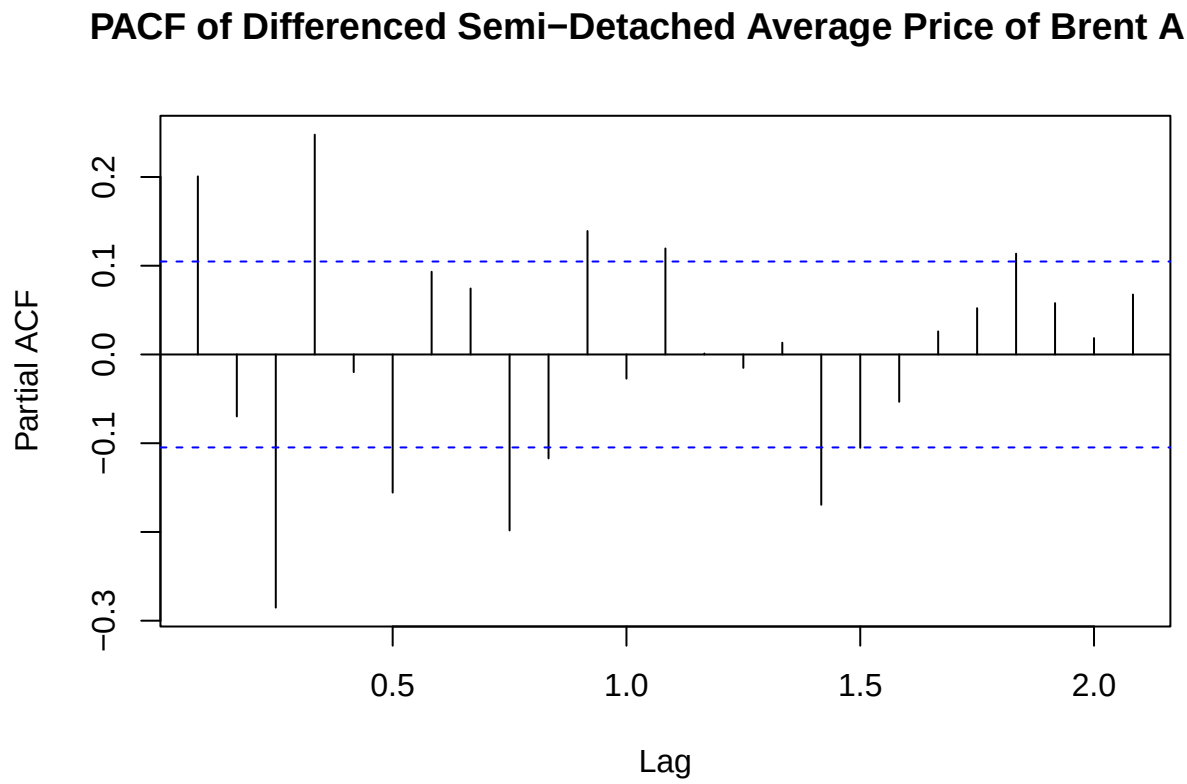
```r
# Define function to plot ACF and PACF of Brent Area
plot_acf_pacf <- function(ts_diff, title) {
  acf(ts_diff, main = paste("ACF of", title))
  pacf(ts_diff, main = paste("PACF of", title))
}

# Plot ACF and PACF for differenced series for each property type of Brent Area
plot_acf_pacf(brent_detached_ts_diff,
              "Differenced Detached Average Price of Brent Area")
```



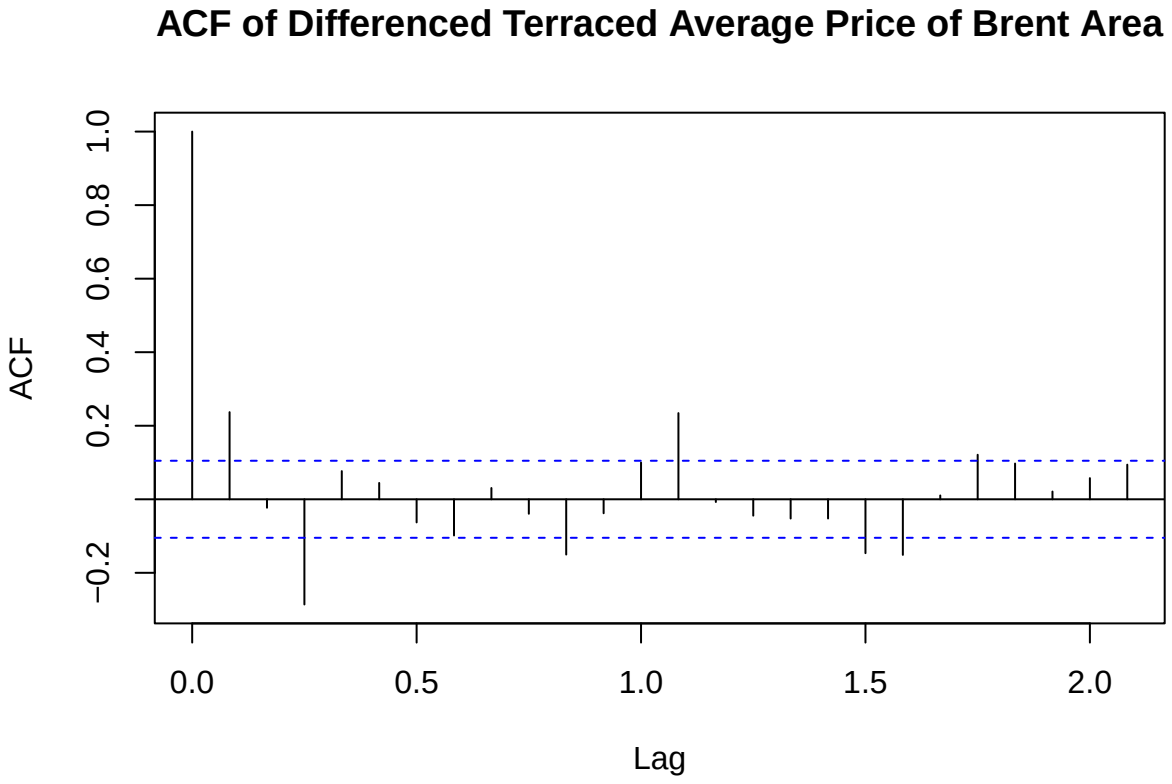


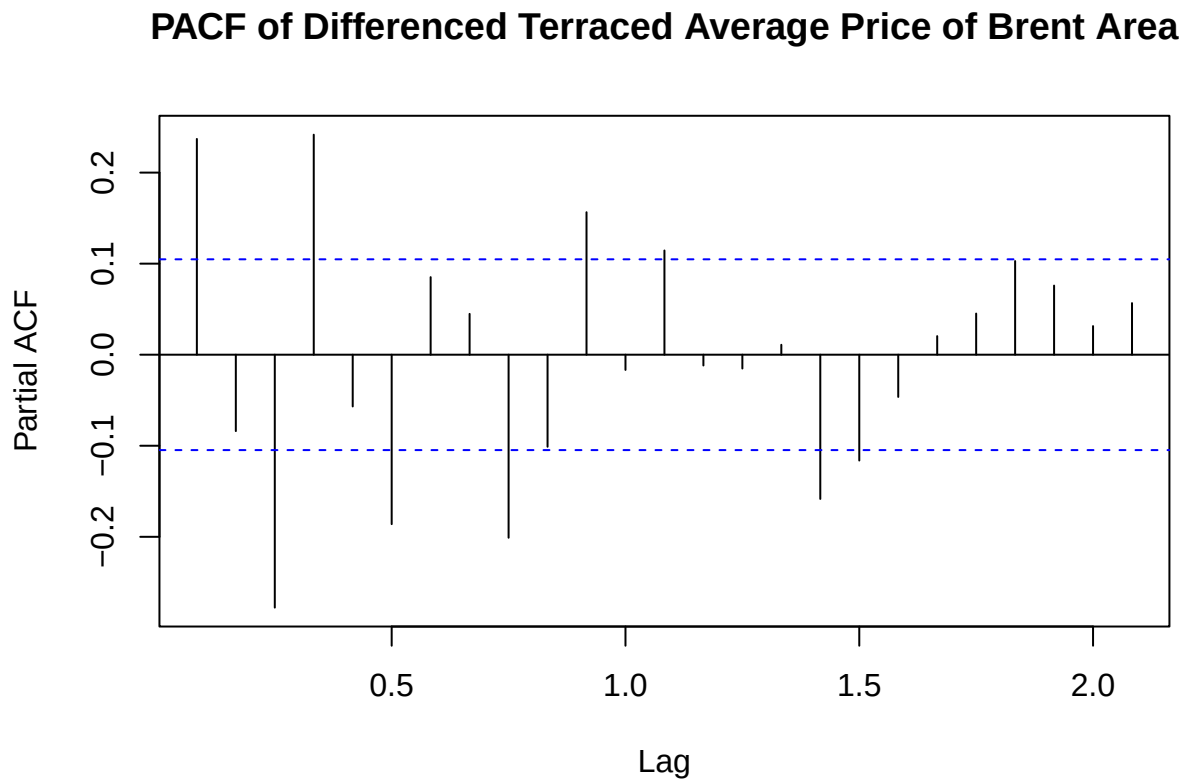

```
plot_acf_pacf(brent_semi_detached_ts_diff,
              "Differenced Semi-Detached Average Price of Brent Area")
```



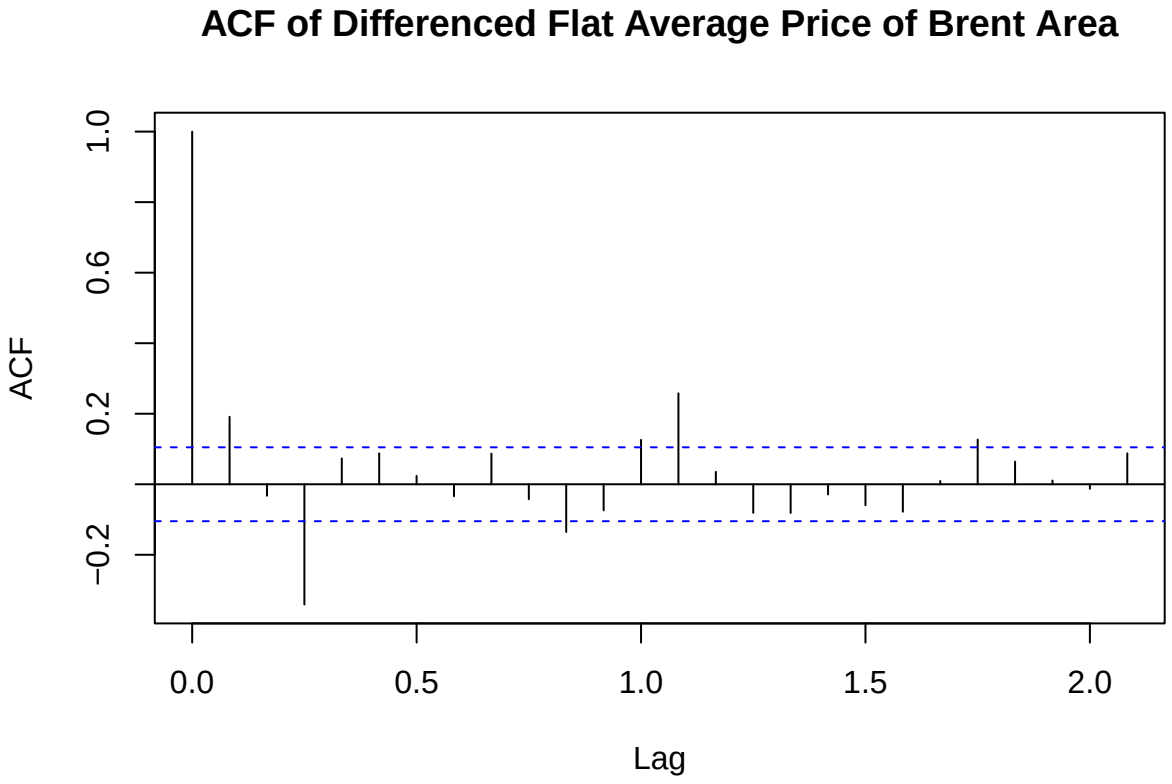


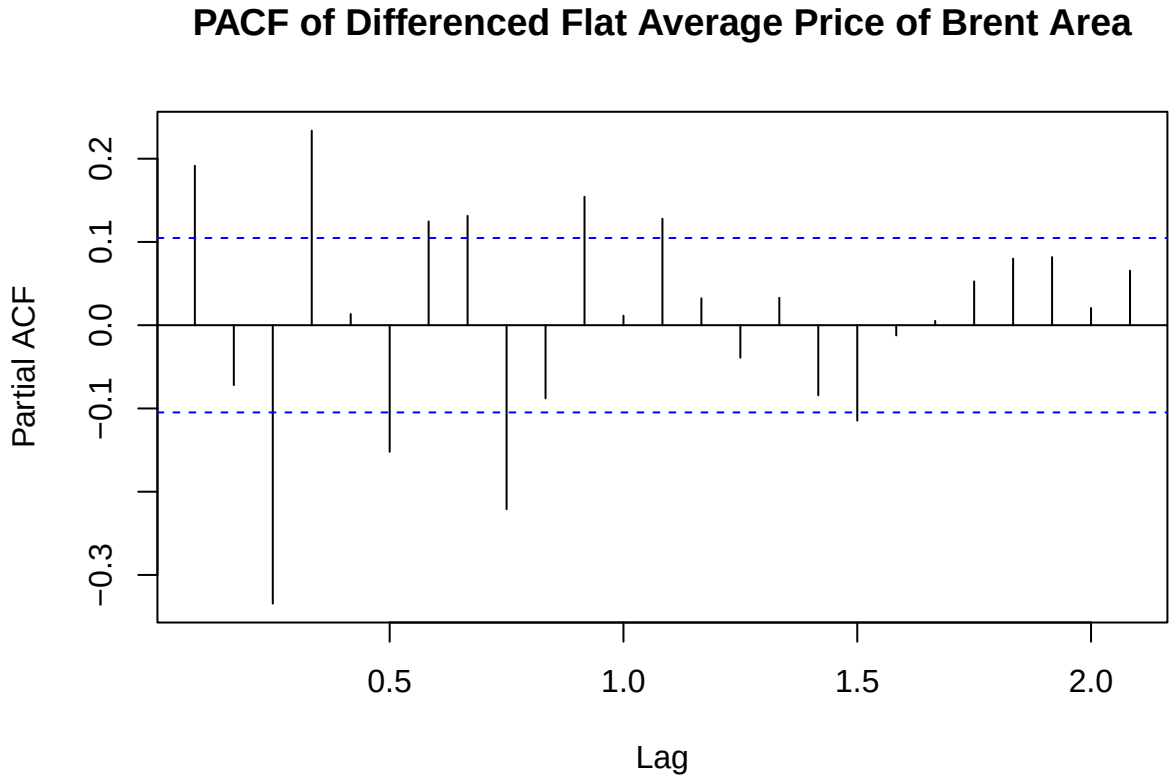

```
plot_acf_pacf(brent_terraced_ts_diff,
              "Differenced Terraced Average Price of Brent Area")
```






```
plot_acf_pacf(brent_flat_ts_diff,
              "Differenced Flat Average Price of Brent Area")
```






```
# Fit ARIMA models for Brent Area
fit_arima_brent_d <- auto.arima(brent_detached_ts_diff, stepwise = FALSE,
                          approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)              with zero mean     : 7740.719
##  ARIMA(0,0,0)              with non-zero mean : 7730.828
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 7738.43
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 7730.577
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 7739.675
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 7732.422
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 7737.926
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 7730.441
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 7738.689
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 7732.147
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)              with zero mean     : 7721.184
##  ARIMA(0,0,1)              with non-zero mean : 7714.912
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 7721.898
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 7716.362
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 7723.176
```

```
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 7718.134
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 7721.734
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 7716.317
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 7722.546
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 7717.92
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 7670.492
##  ARIMA(0,0,2)            with non-zero mean : 7671.266
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 7671.62
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 7672.161
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 7672.246
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 7672.638
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 7671.78
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 7672.376
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : 7673.484
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : 7673.991
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : 7672.856
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : 7673.24
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 7670.969
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 7671.166
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : 7671.181
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : 7671.422
##  ARIMA(0,0,3)            with zero mean     : 7601.334
##  ARIMA(0,0,3)            with non-zero mean : 7592.637
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 7602.418
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 7594.503
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : 7604.415
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 7596.371
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : 7602.435
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 7594.514
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : 7603.948
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : 7596.107
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : 7604.505
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 7596.451
##  ARIMA(0,0,4)            with zero mean     : 7602.754
##  ARIMA(0,0,4)            with non-zero mean : 7593.162
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 7604.167
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 7595.238
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : 7604.176
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 7595.238
##  ARIMA(0,0,5)            with zero mean     : 7597.691
##  ARIMA(0,0,5)            with non-zero mean : 7590.778
##  ARIMA(1,0,0)            with zero mean     : 7719.156
##  ARIMA(1,0,0)            with non-zero mean : 7714.272
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 7719.851
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 7715.56
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 7721.505
```

```
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 7717.492
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 7719.732
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 7715.522
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 7721.106
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 7717.356
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 7720.78
##  ARIMA(1,0,1)            with non-zero mean : 7715.648
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 7721.648
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 7717.121
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 7723.216
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 7718.998
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 7721.53
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 7717.084
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 7722.764
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 7718.833
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : 7633.682
##  ARIMA(1,0,2)            with non-zero mean : 7632.882
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : 7635.473
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 7634.845
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : 7635.445
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 7634.416
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : 7635.523
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 7634.868
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : 7635.726
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : 7635.169
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : 7635.517
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 7634.211
##  ARIMA(1,0,3)            with zero mean     : 7603.044
##  ARIMA(1,0,3)            with non-zero mean : 7593.761
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : 7604.319
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 7595.793
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : 7604.332
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 7595.796
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : 7593.232
##  ARIMA(2,0,0)            with zero mean     : 7719.828
##  ARIMA(2,0,0)            with non-zero mean : 7713.708
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 7721.005
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 7715.534
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 7722.22
```

```
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 7717.069
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 7720.893
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 7715.51
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 7721.58
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 7716.784
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 7718.276
##  ARIMA(2,0,1)            with non-zero mean : 7708.904
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 7719.598
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 7710.914
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 7720.071
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 7711.9
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 7719.471
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 7710.905
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 7719.008
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 7711.452
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 7693.597
##  ARIMA(3,0,0)            with non-zero mean : 7680.387
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 7693.13
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 7681.751
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 7693.405
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 7682.75
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 7692.664
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 7681.652
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 7691.516
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 7682.033
##  ARIMA(3,0,1)            with zero mean     : 7678.696
##  ARIMA(3,0,1)            with non-zero mean : 7667.206
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 7679.744
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 7669.184
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 7679.55
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 7669.17
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : Inf
##  ARIMA(4,0,0)            with zero mean     : 7661.856
##  ARIMA(4,0,0)            with non-zero mean : 7656.028
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 7663.443
```

```
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 7657.976
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 7663.374
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 7657.96
##  ARIMA(4,0,1)            with zero mean     : 7663.922
##  ARIMA(4,0,1)            with non-zero mean : 7658.039
##  ARIMA(5,0,0)            with zero mean     : 7663.918
##  ARIMA(5,0,0)            with non-zero mean : 7657.932
##
##
##
##  Best model: ARIMA(0,0,5)            with non-zero mean
```

```
fit_arima_brent_sd <- auto.arima(brent_semi_detached_ts_diff, stepwise = FALSE,
                          approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 7440.947
##  ARIMA(0,0,0)            with non-zero mean : 7430.444
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 7438.48
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 7430.136
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 7439.994
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 7432.119
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 7438.067
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 7430.062
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 7439.326
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 7432.013
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 7424.814
##  ARIMA(0,0,1)            with non-zero mean : 7417.924
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 7425.17
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 7419.165
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 7426.88
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 7421.178
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 7425.034
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 7419.141
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 7426.541
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 7421.123
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 7388.065
##  ARIMA(0,0,2)            with non-zero mean : 7388.812
```

```
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 7389.047
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 7389.536
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 7388.588
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 7388.872
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 7389.281
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 7389.844
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : 7390.756
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : 7391.187
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : 7388.876
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : 7389.167
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 7386.719
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 7386.768
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : 7386.483
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : 7386.611
##  ARIMA(0,0,3)            with zero mean     : 7312.148
##  ARIMA(0,0,3)            with non-zero mean : 7302.678
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 7313.098
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 7304.537
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : 7314.923
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 7306.17
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : 7313.149
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 7304.554
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : 7314.409
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : 7306.016
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : 7315.169
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 7306.251
##  ARIMA(0,0,4)            with zero mean     : 7314.037
##  ARIMA(0,0,4)            with non-zero mean : 7303.999
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 7315.152
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 7306.043
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : 7315.198
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 7306.047
##  ARIMA(0,0,5)            with zero mean     : 7306.096
##  ARIMA(0,0,5)            with non-zero mean : 7299.433
##  ARIMA(1,0,0)            with zero mean     : 7423.071
##  ARIMA(1,0,0)            with non-zero mean : 7417.448
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 7423.402
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 7418.516
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 7425.324
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 7420.571
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 7423.318
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 7418.507
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 7425.145
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 7420.555
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 7424.875
##  ARIMA(1,0,1)            with non-zero mean : 7419.04
```

```
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 7425.347
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 7420.279
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 7427.247
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 7422.336
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 7425.258
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 7420.265
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 7427.041
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 7422.307
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : 7346.976
##  ARIMA(1,0,2)            with non-zero mean : 7346.055
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : 7348.628
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 7347.931
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : 7347.468
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 7346.259
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : 7348.724
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 7347.981
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : 7348.165
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : 7347.536
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : 7347.543
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 7345.966
##  ARIMA(1,0,3)            with zero mean     : 7314.119
##  ARIMA(1,0,3)            with non-zero mean : 7304.314
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : 7315.16
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 7306.303
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : 7315.209
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 7306.31
##  ARIMA(1,0,4) with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : 7303.534
##  ARIMA(2,0,0)            with zero mean     : 7424.328
##  ARIMA(2,0,0)            with non-zero mean : 7417.64
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 7425.075
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 7419.267
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 7426.848
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 7421.223
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 7424.977
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 7419.246
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 7426.547
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 7421.134
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 7423.744
##  ARIMA(2,0,1)            with non-zero mean : 7414.163
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 7416.083
```

```
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 7426.068
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 7417.724
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 7416.069
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 7425.464
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 7417.511
##  ARIMA(2,0,2)            with zero mean     : 7313.483
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : 7308.38
##  ARIMA(2,0,3)            with non-zero mean : 7299.644
##  ARIMA(3,0,0)            with zero mean     : 7400.92
##  ARIMA(3,0,0)            with non-zero mean : 7387.187
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 7400.256
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 7388.524
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 7401.177
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 7389.951
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 7399.857
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 7388.445
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 7399.716
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 7389.445
##  ARIMA(3,0,1)            with zero mean     : 7386.318
##  ARIMA(3,0,1)            with non-zero mean : 7374.228
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 7387.257
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 7376.201
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 7387.083
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 7376.189
##  ARIMA(3,0,2)            with zero mean     : 7308.121
##  ARIMA(3,0,2)            with non-zero mean : 7300.901
##  ARIMA(4,0,0)            with zero mean     : 7368.243
##  ARIMA(4,0,0)            with non-zero mean : 7362.093
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 7369.794
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 7364.038
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 7369.736
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 7364.026
##  ARIMA(4,0,1)            with zero mean     : 7370.266
##  ARIMA(4,0,1)            with non-zero mean : 7364.16
##  ARIMA(5,0,0)            with zero mean     : 7370.215
##  ARIMA(5,0,0)            with non-zero mean : 7364.136
##
##
##
##  Best model: ARIMA(0,0,5)            with non-zero mean
```

```
fit_arima_brent_t <- auto.arima(brent_terraced_ts_diff, stepwise = FALSE,
                          approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 7339.597
##  ARIMA(0,0,0)            with non-zero mean : 7329.726
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 7336.031
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 7328.356
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 7336.746
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 7329.828
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 7335.179
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 7328.013
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 7334.969
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 7329.069
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 7317.811
##  ARIMA(0,0,1)            with non-zero mean : 7311.637
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 7317.925
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 7312.596
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 7319.14
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 7314.276
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 7317.67
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 7312.5
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 7318.23
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 7313.882
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 7270.304
##  ARIMA(0,0,2)            with non-zero mean : 7271.038
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 7271.71
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 7272.272
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 7271.945
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 7272.36
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 7271.829
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 7272.435
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : 7273.576
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : 7274.106
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : 7272.205
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : 7272.613
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 7270.552
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 7270.753
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : 7270.104
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : 7270.352
##  ARIMA(0,0,3)            with zero mean     : 7202.718
```

```
##  ARIMA(0,0,3)            with non-zero mean : 7194.391
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 7202.972
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 7195.719
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : 7205.008
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 7197.718
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : 7202.97
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 7195.736
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : 7204.974
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 7197.816
##  ARIMA(0,0,4)            with zero mean     : 7203.408
##  ARIMA(0,0,4)            with non-zero mean : 7193.665
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 7204.342
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 7195.624
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : 7204.335
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 7195.627
##  ARIMA(0,0,5)            with zero mean     : 7198.242
##  ARIMA(0,0,5)            with non-zero mean : 7191.564
##  ARIMA(1,0,0)            with zero mean     : 7315.546
##  ARIMA(1,0,0)            with non-zero mean : 7310.835
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 7315.585
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 7311.539
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 7317.249
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 7313.437
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 7315.404
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 7311.461
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 7316.688
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 7313.188
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 7317.098
##  ARIMA(1,0,1)            with non-zero mean : 7312.126
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 7317.381
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 7313.106
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 7318.968
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 7314.949
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 7317.202
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 7313.03
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 7318.345
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 7314.662
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : 7235.202
```

```
##  ARIMA(1,0,2)            with non-zero mean : 7234.454
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : 7236.507
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 7236.036
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : 7236.452
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 7235.634
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : 7236.649
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 7236.137
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : 7236.395
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : 7235.928
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : 7236.887
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 7235.827
##  ARIMA(1,0,3)            with zero mean     : 7204.034
##  ARIMA(1,0,3)            with non-zero mean : 7194.734
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : 7204.676
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 7196.524
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : 7204.672
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 7196.531
##  ARIMA(1,0,4)            with zero mean     : 7202.762
##  ARIMA(1,0,4)            with non-zero mean : 7193.37
##  ARIMA(2,0,0)            with zero mean     : 7316.089
##  ARIMA(2,0,0)            with non-zero mean : 7310.178
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 7316.787
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 7311.679
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 7318.08
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 7313.208
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 7316.614
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 7311.619
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 7317.247
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 7312.771
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 7314.776
##  ARIMA(2,0,1)            with non-zero mean : 7305.621
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 7315.676
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 7307.438
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 7316.37
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 7308.574
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 7315.485
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 7307.405
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 7315.083
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 7307.98
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
```

```
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 7293.59
##  ARIMA(3,0,0)            with non-zero mean : 7281.574
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 7292.46
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 7282.357
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 7292.781
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 7283.143
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 7291.863
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 7282.155
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 7290.24
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 7281.87
##  ARIMA(3,0,1)            with zero mean     : 7279.005
##  ARIMA(3,0,1)            with non-zero mean : 7268.407
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 7279.462
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 7270.045
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 7279.15
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 7269.973
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : Inf
##  ARIMA(4,0,0)            with zero mean     : 7262.866
##  ARIMA(4,0,0)            with non-zero mean : 7257.453
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 7263.856
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 7258.978
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 7263.676
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 7258.895
##  ARIMA(4,0,1)            with zero mean     : 7264.858
##  ARIMA(4,0,1)            with non-zero mean : 7259.236
##  ARIMA(5,0,0)            with zero mean     : 7264.752
##  ARIMA(5,0,0)            with non-zero mean : 7258.706
##
##
##
##  Best model: ARIMA(0,0,5)            with non-zero mean
```

```
fit_arima_brent_f <- auto.arima(brent_flat_ts_diff, stepwise = FALSE,
                          approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 7023.711
##  ARIMA(0,0,0)            with non-zero mean : 7014.322
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 7017.073
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 7010.275
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 7019.099
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 7012.215
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 7017.102
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 7010.601
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : 7019.095
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : 7012.22
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : 7014.272
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 7019.129
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 7012.332
```

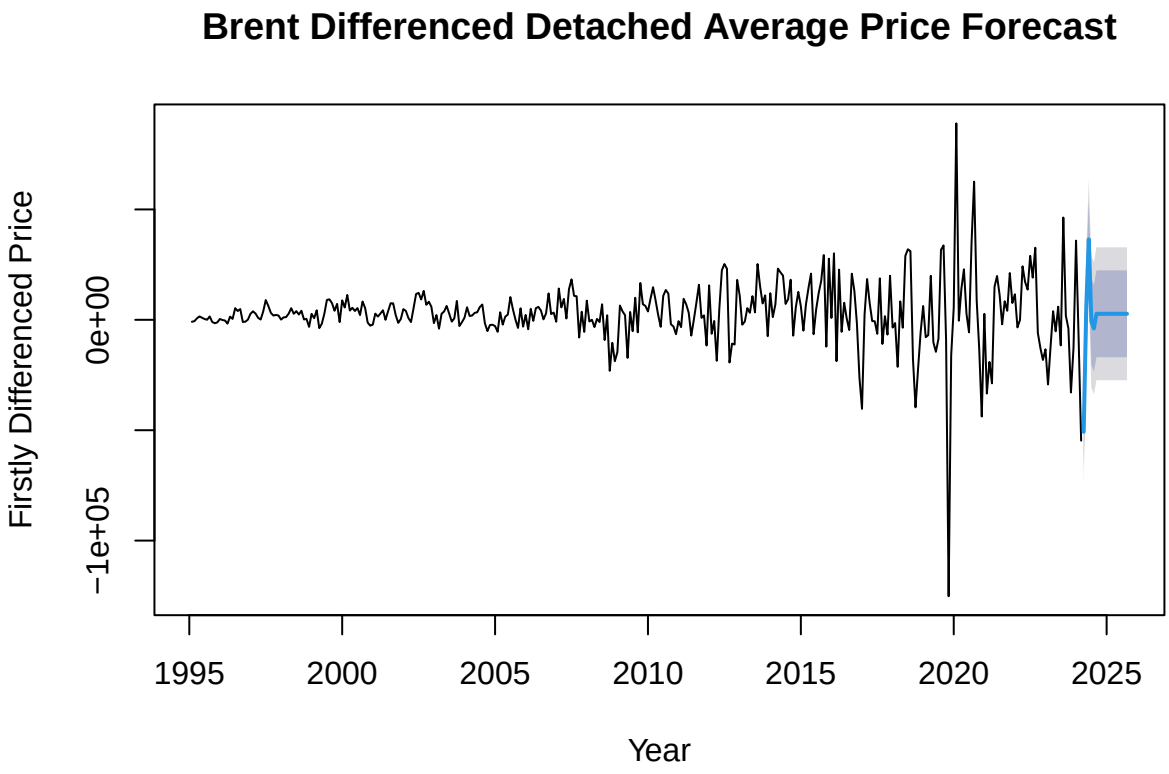
```
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : 7021.102
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : 7014.257
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 7009.818
##  ARIMA(0,0,1)            with non-zero mean : 7003.495
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 7006.653
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 7001.748
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 7008.656
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 7003.518
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 7006.839
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 7002.038
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : 7008.658
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : 7003.575
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : 7010.728
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : 7005.587
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 7008.765
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 7003.569
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : 7010.708
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : 7005.522
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 6969.779
##  ARIMA(0,0,2)            with non-zero mean : 6970.577
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 6971.529
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 6972.207
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 6967.025
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 6967.348
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 6971.623
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 6972.347
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : 6971.756
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : 6973.45
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : 6967.102
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : 6967.486
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 6965.209
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 6965.317
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : 6964
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : 6964.254
##  ARIMA(0,0,3)            with zero mean     : 6885.843
##  ARIMA(0,0,3)            with non-zero mean : 6876.254
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 6884.865
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 6876.756
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : 6885.754
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 6877.135
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : 6885.223
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 6876.988
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : 6885.226
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : 6877.125
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : 6886.631
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 6877.652
##  ARIMA(0,0,4)            with zero mean     : 6887.483
##  ARIMA(0,0,4)            with non-zero mean : 6877.364
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 6886.836
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 6878.311
```

```
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : 6887.172
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 6878.493
##  ARIMA(0,0,5)            with zero mean     : Inf
##  ARIMA(0,0,5)            with non-zero mean : Inf
##  ARIMA(1,0,0)            with zero mean     : 7008.478
##  ARIMA(1,0,0)            with non-zero mean : 7003.143
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 7005.415
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 7001.247
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 7007.32
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 7002.89
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 7005.708
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 7001.602
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : 7007.321
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : 7002.99
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : 7009.37
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : 7004.958
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 7007.459
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 7002.929
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : 7009.354
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : 7004.857
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 7010.283
##  ARIMA(1,0,1)            with non-zero mean : 7004.782
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 7007.376
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 7003.099
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 7009.31
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 7004.787
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 7007.645
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 7003.423
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : 7009.308
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : 7004.876
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : 7011.373
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : 7006.871
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 7009.442
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 7004.825
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : 7011.359
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : 7006.777
##  ARIMA(1,0,2)            with zero mean     : 6924.909
##  ARIMA(1,0,2)            with non-zero mean : 6924.042
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : 6925.96
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 6925.431
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : 6922.51
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 6921.343
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : 6926.258
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 6925.645
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : 6924.071
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : 6923.566
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : 6922.78
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 6921.227
##  ARIMA(1,0,3)            with zero mean     : 6887.676
##  ARIMA(1,0,3)            with non-zero mean : 6877.761
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : 6886.885
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 6878.544
```

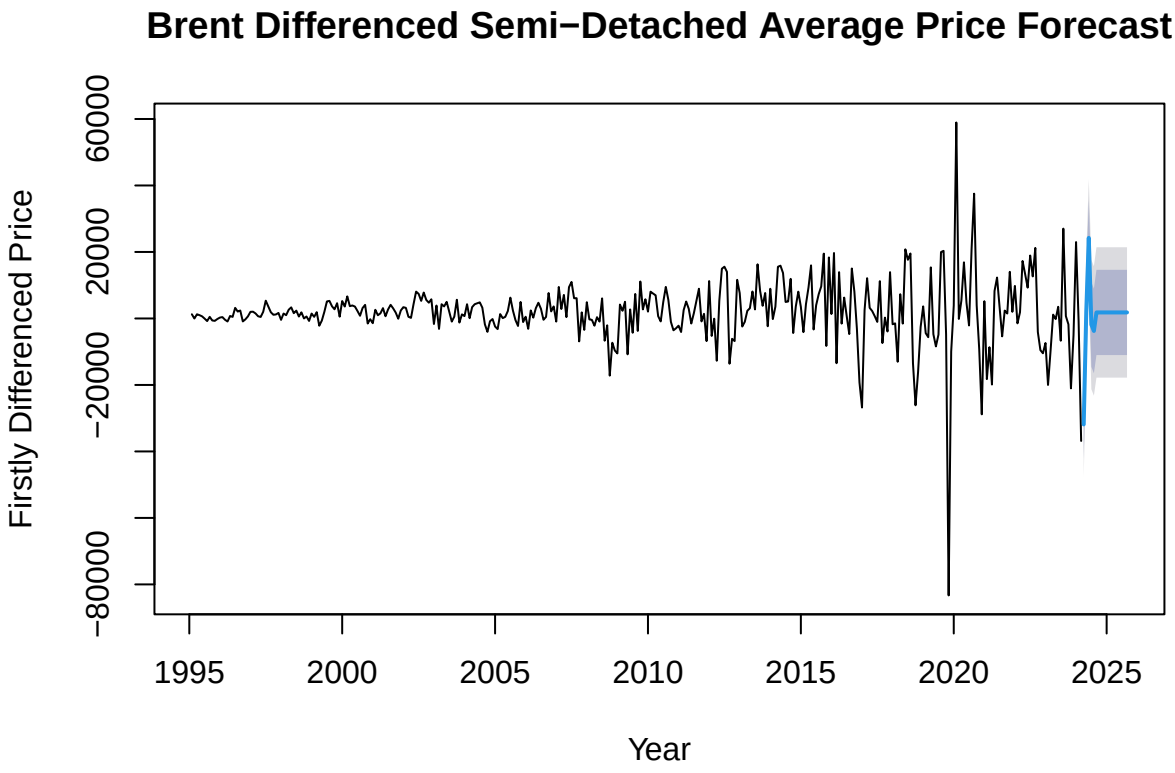
```
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : 6887.232
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 6878.747
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : Inf
##  ARIMA(2,0,0)            with zero mean     : 7009.608
##  ARIMA(2,0,0)            with non-zero mean : 7003.28
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 7007.09
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 7002.252
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 7009.074
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 7004.075
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 7007.281
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 7002.466
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : 7009.067
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : 7004.125
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : 7011.138
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : 7006.16
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 7009.185
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 7004.116
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : 7011.132
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : 7006.102
##  ARIMA(2,0,1)            with zero mean     : 7007.458
##  ARIMA(2,0,1)            with non-zero mean : 6997.616
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 6997.411
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 6999.79
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 6999.447
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 7005.341
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 6997.48
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : 6999.449
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 7000.03
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 6999.477
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 6976.311
##  ARIMA(3,0,0)            with non-zero mean : 6962.072
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 6972.497
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 6961.212
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 6974.495
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 6963.29
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 6972.276
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 6961.227
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 6974.131
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 6963.31
##  ARIMA(3,0,1)            with zero mean     : 6963.738
##  ARIMA(3,0,1)            with non-zero mean : 6951.468
```

```
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 6962.242
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 6951.97
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 6962.125
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 6951.983
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : Inf
##  ARIMA(4,0,0)            with zero mean     : 6949.401
##  ARIMA(4,0,0)            with non-zero mean : 6942.086
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 6949.254
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 6942.856
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 6949.249
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 6942.884
##  ARIMA(4,0,1)            with zero mean     : 6950.992
##  ARIMA(4,0,1)            with non-zero mean : 6944.082
##  ARIMA(5,0,0)            with zero mean     : 6950.542
##  ARIMA(5,0,0)            with non-zero mean : 6943.96
##
##
##
##  Best model: ARIMA(0,0,3)            with non-zero mean
```

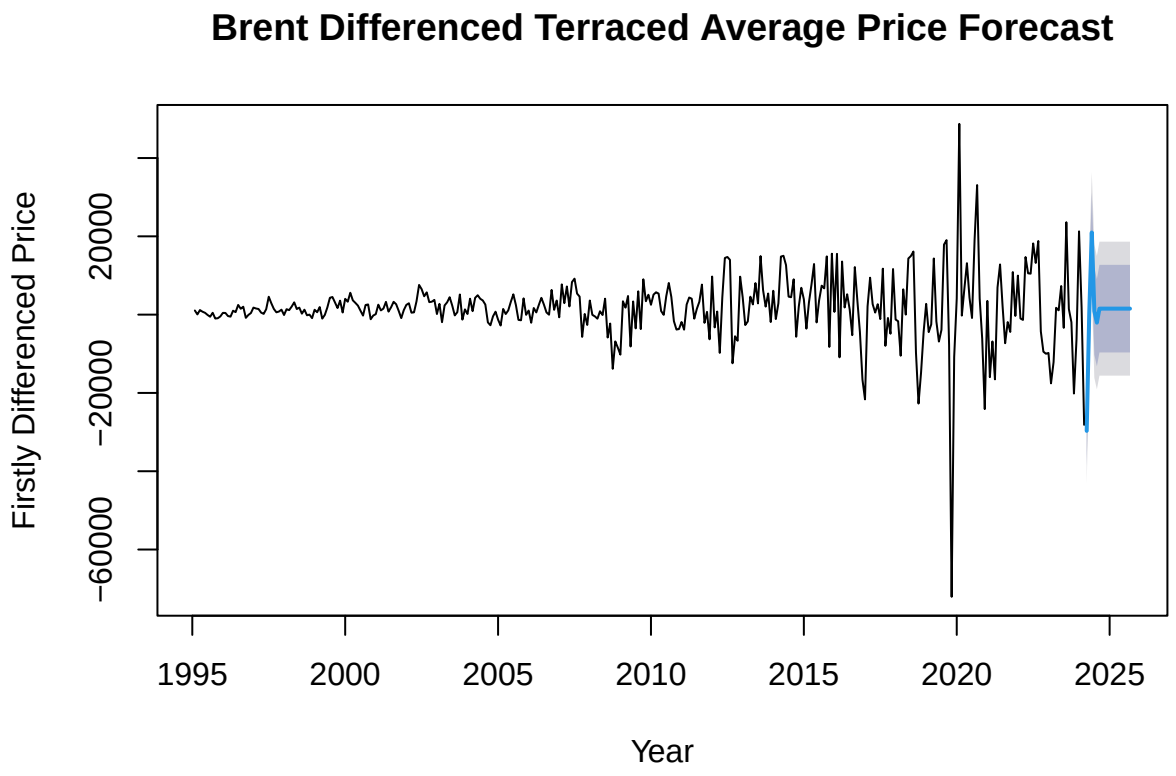
```r
# Forecast using the ARIMA model for each property type of Brent Area
forecasted_values_arima_brent_d <- forecast(fit_arima_brent_d, h = 18)
forecasted_values_arima_brent_sd <- forecast(fit_arima_brent_sd, h = 18)
forecasted_values_arima_brent_t <- forecast(fit_arima_brent_t, h = 18)
forecasted_values_arima_brent_f <- forecast(fit_arima_brent_f, h = 18)

# Plot the differenced forecast value for each property type of Brent Area
plot(forecasted_values_arima_brent_d,
     main = "Brent Differenced Detached Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```



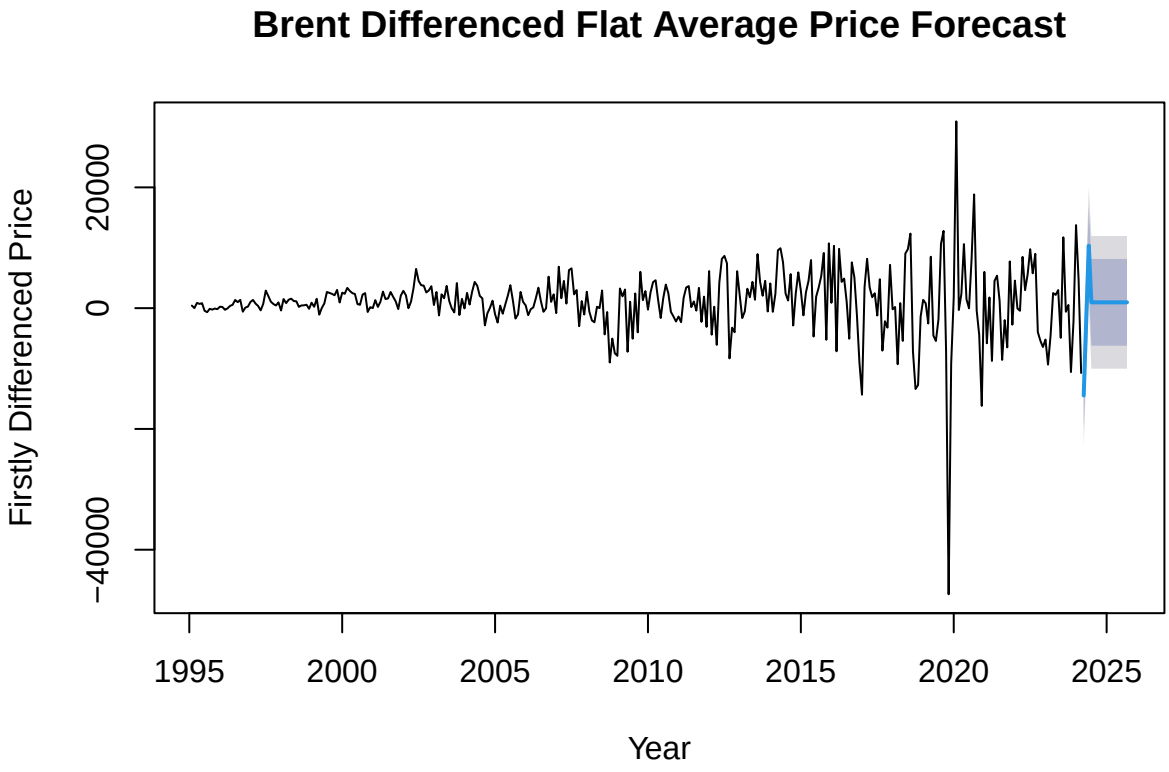

```
plot(forecasted_values_arima_brent_sd,
     main = "Brent Differenced Semi-Detached Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```



```
plot(forecasted_values_arima_brent_t,
     main = "Brent Differenced Terraced Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```




```
plot(forecasted_values_arima_brent_f,
     main = "Brent Differenced Flat Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```




```
# Summary of the model for four different type of properties of Brent Area
# For Detached of Brent Area
print(forecasted_values_arima_brent_d)
```

```
##          Point Forecast     Lo 80     Hi 80      Lo 95     Hi 95
## Apr 2024    -50716.7586 -66295.99 -35137.53 -74543.130 -26890.39
## May 2024      5506.6442 -11490.49  22503.78 -20488.226  31501.51
## Jun 2024     36470.6679  18877.45  54063.88   9564.167  63377.17
## Jul 2024      -731.6382 -20309.81  18846.53 -30673.867  29210.59
## Aug 2024     -3887.2727 -23467.26  15692.71 -33832.279  26057.73
## Sep 2024      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Oct 2024      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Nov 2024      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Dec 2024      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Jan 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Feb 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Mar 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Apr 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## May 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Jun 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Jul 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Aug 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
## Sep 2025      2723.8064 -16953.08  22400.70 -27369.401  32817.01
```

```
summary(fit_arima_brent_d)
```

```
## Series: brent_detached_ts_diff
## ARIMA(0,0,5) with non-zero mean
```

```
##
## Coefficients:
##          ma1     ma2      ma3      ma4     ma5       mean
##       0.4363  0.2915  -0.5514  -0.0171  0.1252  2723.8064
## s.e.  0.0545  0.0605   0.0585   0.0736  0.0581   828.4347
##
## sigma^2 = 147779433:  log likelihood = -3788.23
## AIC=7590.45   AICc=7590.78   BIC=7617.46
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.5480889 12051.81 7330.117 590.1545 821.9871 0.5884996
##                      ACF1
## Training set 0.004829623
```

```
# For Semi-Detached of Brent Area
print(forecasted_values_arima_brent_sd)
```

```
##          Point Forecast      Lo 80      Hi 80      Lo 95     Hi 95
## Apr 2024     -31875.313 -42152.062 -21598.565 -47592.244 -16158.38
## May 2024       2051.992  -9049.153  13153.138 -14925.745  19029.73
## Jun 2024      24238.808  12780.314  35697.301   6714.555  41763.06
## Jul 2024      -1825.539 -14546.212  10895.134 -21280.130  17629.05
## Aug 2024      -3767.677 -16490.335   8954.982 -23225.304  15689.95
## Sep 2024       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Oct 2024       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Nov 2024       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Dec 2024       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Jan 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Feb 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Mar 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Apr 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## May 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Jun 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Jul 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Aug 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
## Sep 2025       1812.175 -11007.526  14631.876 -17793.866  21418.22
```

```
summary(fit_arima_brent_sd)
```

```
## Series: brent_semi_detached_ts_diff
## ARIMA(0,0,5) with non-zero mean
##
## Coefficients:
##          ma1     ma2      ma3     ma4     ma5       mean
##       0.4085  0.2763  -0.5376  0.0219  0.1532  1812.1749
## s.e.  0.0540  0.0582   0.0593  0.0714  0.0581   562.1047
##
## sigma^2 = 64303470:  log likelihood = -3642.55
## AIC=7299.11   AICc=7299.43   BIC=7326.11
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
```

```
## Training set 0.8649025 7949.914 4764.676 5.444808 330.4916 0.5851739
##                     ACF1
## Training set 0.003683309
```

```
# For Terraced of Brent Area
print(forecasted_values_arima_brent_t)
```

```
##          Point Forecast      Lo 80      Hi 80      Lo 95     Hi 95
## Apr 2024    -29689.7385 -38496.904 -20882.573 -43159.136 -16220.34
## May 2024       424.8655  -9239.336  10089.067 -14355.256  15204.99
## Jun 2024     20991.5830  10970.099  31013.067   5665.046  36318.12
## Jul 2024       869.3539 -10261.280  11999.987 -16153.481  17892.19
## Aug 2024     -2047.4578 -13184.732   9089.817 -19080.450  14985.53
## Sep 2024      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Oct 2024      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Nov 2024      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Dec 2024      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Jan 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Feb 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Mar 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Apr 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## May 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Jun 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Jul 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Aug 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
## Sep 2025      1529.9638  -9658.311  12718.238 -15581.025  18640.95
```

```
summary(fit_arima_brent_t)
```

```
## Series: brent_terraced_ts_diff
## ARIMA(0,0,5) with non-zero mean
##
## Coefficients:
##          ma1     ma2      ma3      ma4     ma5       mean
##       0.4518  0.3011  -0.5500  -0.0437  0.1212  1529.9638
## s.e.  0.0540  0.0598   0.0634   0.0743  0.0585   466.8625
##
## sigma^2 = 47227370:  log likelihood = -3588.62
## AIC=7191.24   AICc=7191.56   BIC=7218.24
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.3876984 6813.058 4100.292 -324.8744 581.8374 0.5953967
##                     ACF1
## Training set 0.001820432
```

```
# For Flat of Brent Area
print(forecasted_values_arima_brent_f)
```

```
##          Point Forecast      Lo 80     Hi 80       Lo 95     Hi 95
## Apr 2024    -14457.2294 -20088.274 -8826.185 -23069.1684 -5845.290
## May 2024     -1861.1296  -7836.861  4114.602 -11000.2229  7277.964
```

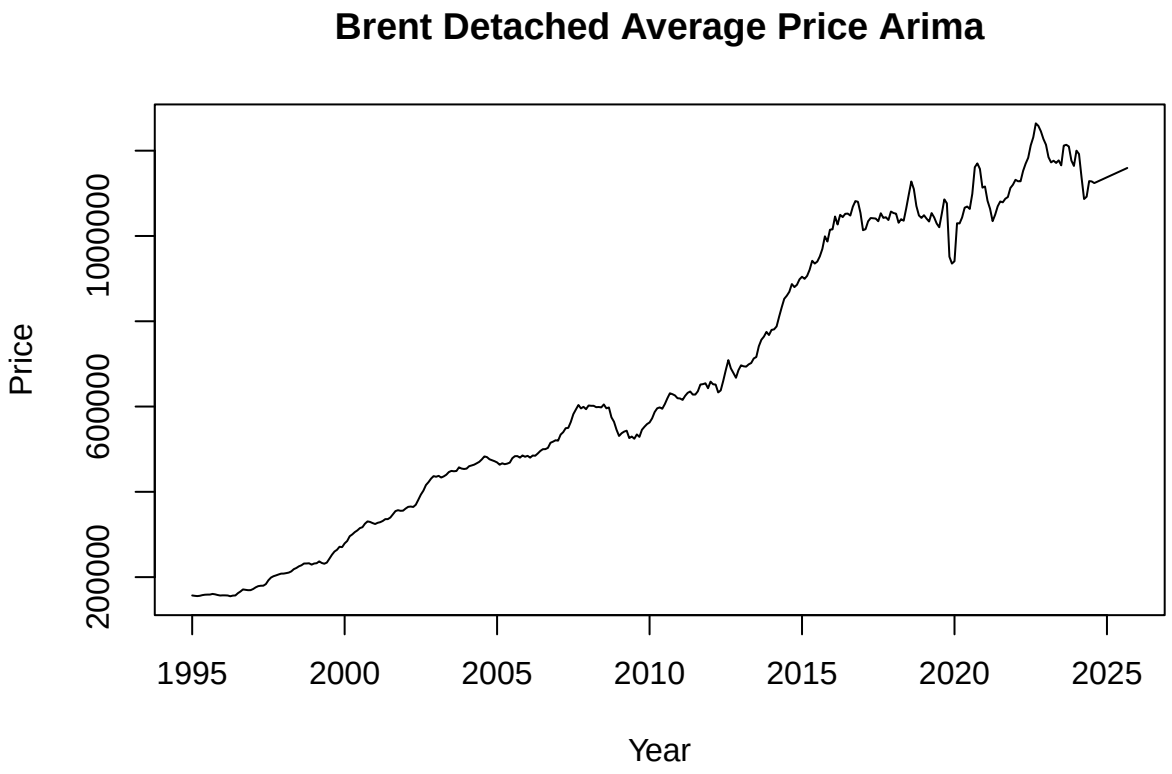
```
## Jun 2024     10336.5996   4019.521 16653.679    675.4609 19997.738
## Jul 2024       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Aug 2024       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Sep 2024       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Oct 2024       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Nov 2024       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Dec 2024       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Jan 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Feb 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Mar 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Apr 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## May 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Jun 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Jul 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Aug 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
## Sep 2025       960.5162  -6219.355  8140.388 -10020.1495 11941.182
```

```
summary(fit_arima_brent_f)
```

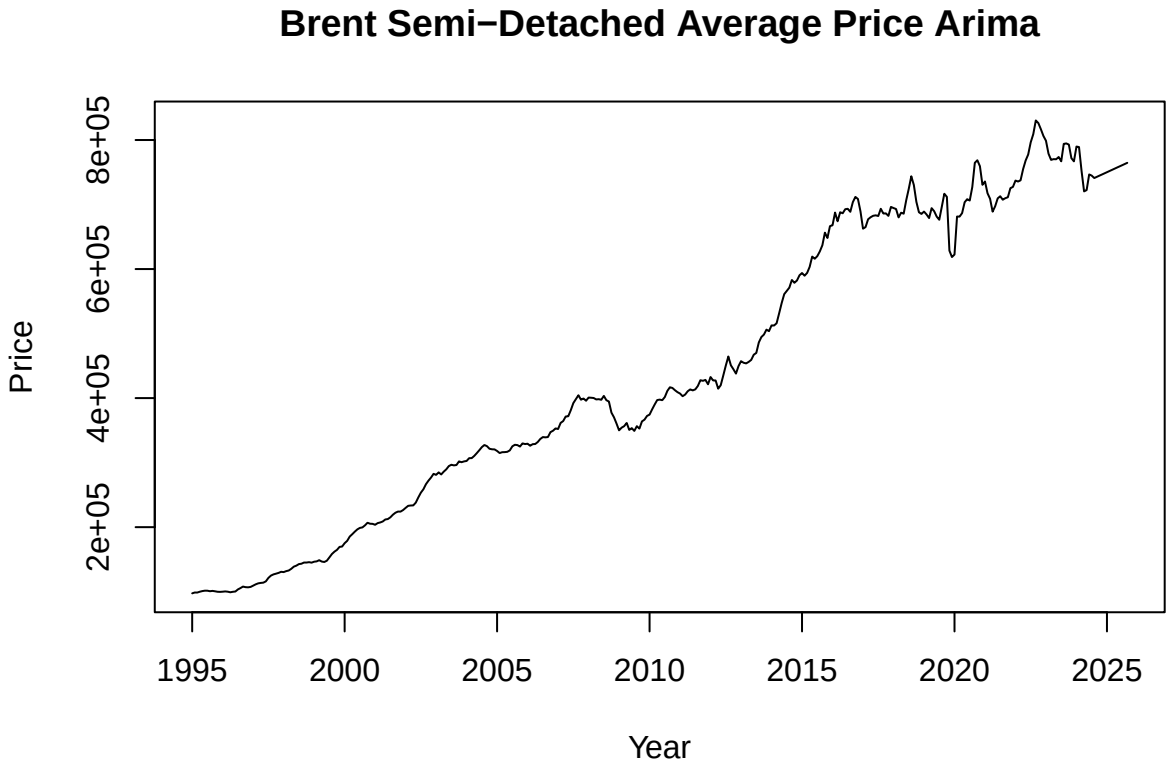
```
## Series: brent_flat_ts_diff
## ARIMA(0,0,3) with non-zero mean
##
## Coefficients:
##          ma1     ma2      ma3      mean
##       0.3552  0.3638  -0.6060  960.5162
## s.e.  0.0426  0.0453   0.0412  260.4608
##
## sigma^2 = 19306531:  log likelihood = -3433.04
## AIC=6876.08   AICc=6876.25   BIC=6895.37
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.8731979 4368.739 2704.925 953.4919 1223.537 0.5900422
##                     ACF1
## Training set 0.02600061
```

```
# Calculate the forecasted actual prices
# By adding the last observed price and the forecasted differences of Brent Area
# For detached_ts of Brent Area
last_value_brent_d <- as.numeric(tail(brent_detached_ts, n = 1))
forecasted_values_brent_d <- c(last_value_brent_d,
                               forecasted_values_arima_brent_d$mean)
cumulative_forecasted_values_brent_d <- cumsum(forecasted_values_brent_d)
forecasted_values_brent_d_ts <- ts(cumulative_forecasted_values_brent_d[-1],
                                   start = c(2024, 2), frequency = 12)

# For semi_detached_ts of Brent Area
last_value_brent_sd <- as.numeric(tail(brent_semi_detached_ts, n = 1))
forecasted_values_brent_sd <- c(last_value_brent_sd,
                                forecasted_values_arima_brent_sd$mean)
cumulative_forecasted_values_brent_sd <- cumsum(forecasted_values_brent_sd)
forecasted_values_brent_sd_ts <- ts(cumulative_forecasted_values_brent_sd[-1],
                                   start = c(2024, 2), frequency = 12)
```

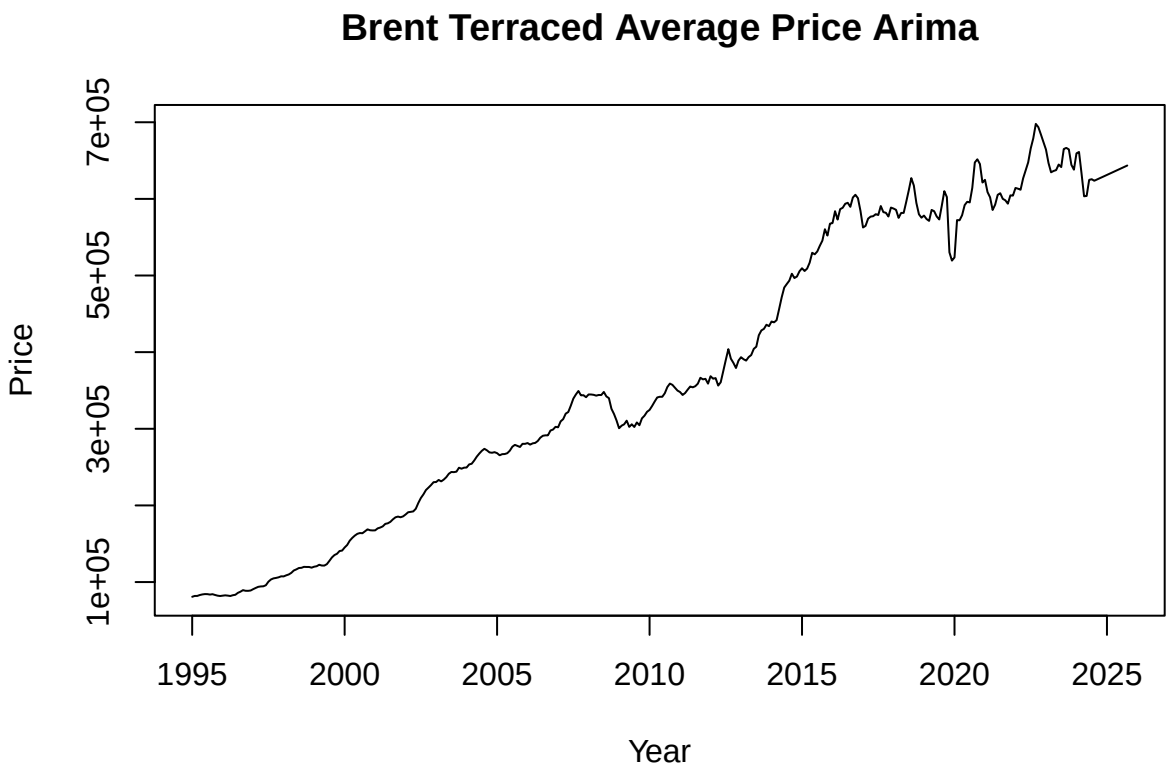
```r
# For terraced_ts of Brent Area
last_value_brent_t <- as.numeric(tail(brent_terraced_ts, n = 1))
forecasted_values_brent_t <- c(last_value_brent_t,
                               forecasted_values_arima_brent_t$mean)
cumulative_forecasted_values_brent_t <- cumsum(forecasted_values_brent_t)
forecasted_values_brent_t_ts <- ts(cumulative_forecasted_values_brent_t[-1],
                             start = c(2024, 2), frequency = 12)

# For flat_ts of Brent Area
last_value_brent_f <- as.numeric(tail(brent_flat_ts, n = 1))
forecasted_values_brent_f <- c(last_value_brent_f,
                               forecasted_values_arima_brent_f$mean)
cumulative_forecasted_values_brent_f <- cumsum(forecasted_values_brent_f)
forecasted_values_brent_f_ts <- ts(cumulative_forecasted_values_brent_f[-1],
                                   start = c(2024, 2), frequency = 12)

# Combine the original and forecasted time series of Brent Area
combined_brent_detached_ts_Arima <- ts(c(as.numeric(brent_detached_ts),
as.numeric(forecasted_values_brent_d_ts)), start = c(1995, 1), frequency = 12)
combined_brent_semi_detached_ts_Arima <-
  ts(c(as.numeric(brent_semi_detached_ts),
as.numeric(forecasted_values_brent_sd_ts)), start = c(1995, 1), frequency = 12)
combined_brent_terraced_ts_Arima <- ts(c(as.numeric(brent_terraced_ts),
as.numeric(forecasted_values_brent_t_ts)), start = c(1995, 1), frequency = 12)
combined_brent_flat_ts_Arima <- ts(c(as.numeric(brent_flat_ts),
as.numeric(forecasted_values_brent_f_ts)), start = c(1995, 1), frequency = 12)

# Plot the combined time series of Brent Area
plot(combined_brent_detached_ts_Arima,
     main = "Brent Detached Average Price Arima",
     ylab = "Price", xlab = "Year")
```



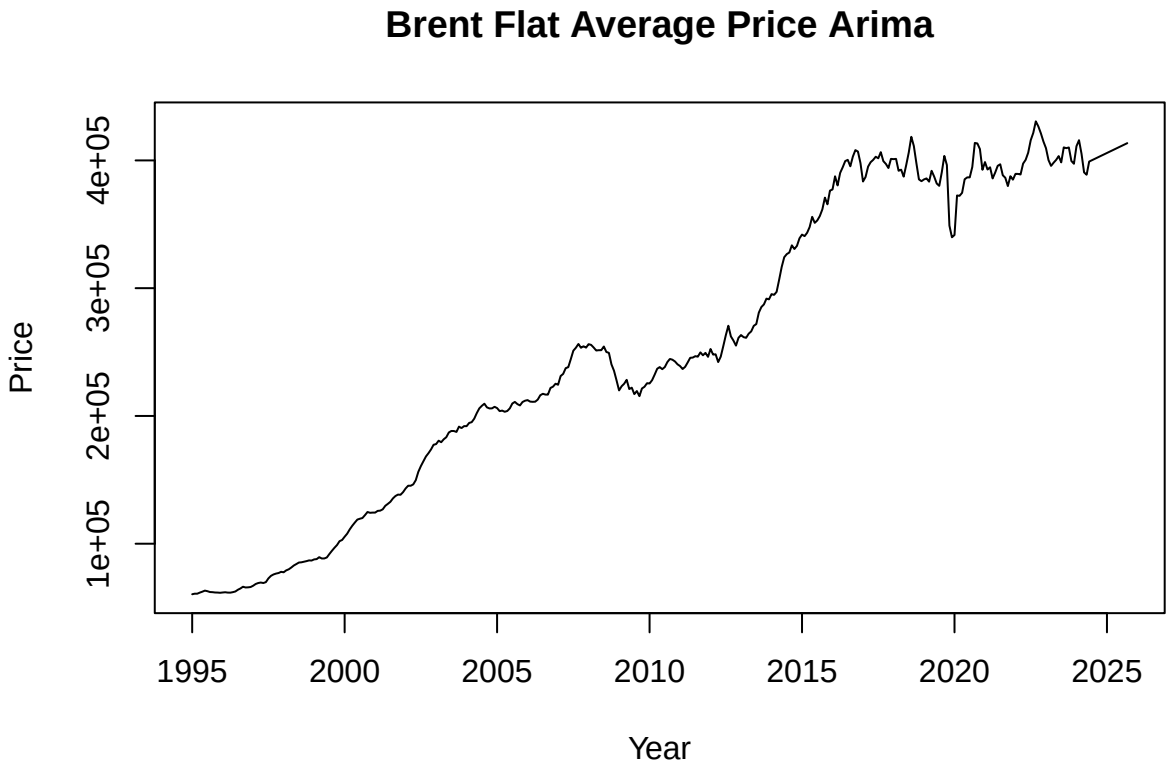

```
plot(combined_brent_semi_detached_ts_Arima,
     main = "Brent Semi-Detached Average Price Arima",
     ylab = "Price", xlab = "Year")
```



```
plot(combined_brent_terraced_ts_Arima,
     main = "Brent Terraced Average Price Arima",
     ylab = "Price", xlab = "Year")
```

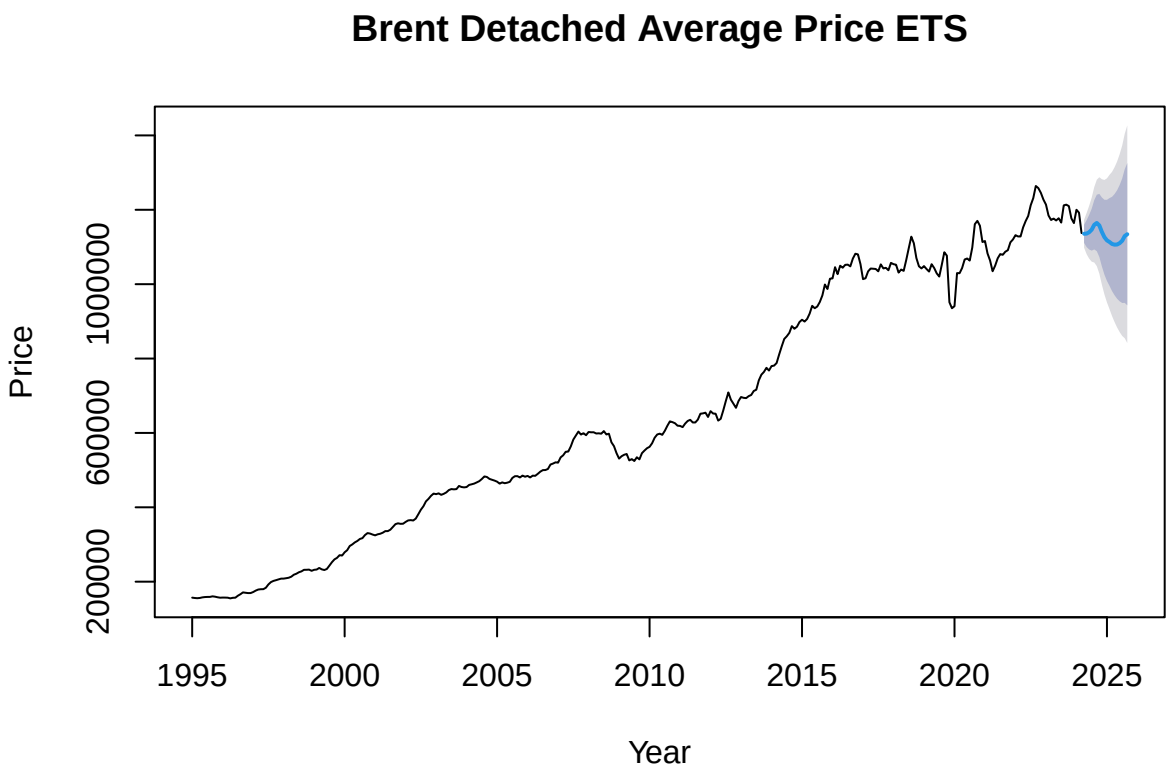



```
plot(combined_brent_flat_ts_Arima,
     main = "Brent Flat Average Price Arima",
     ylab = "Price", xlab = "Year")
```

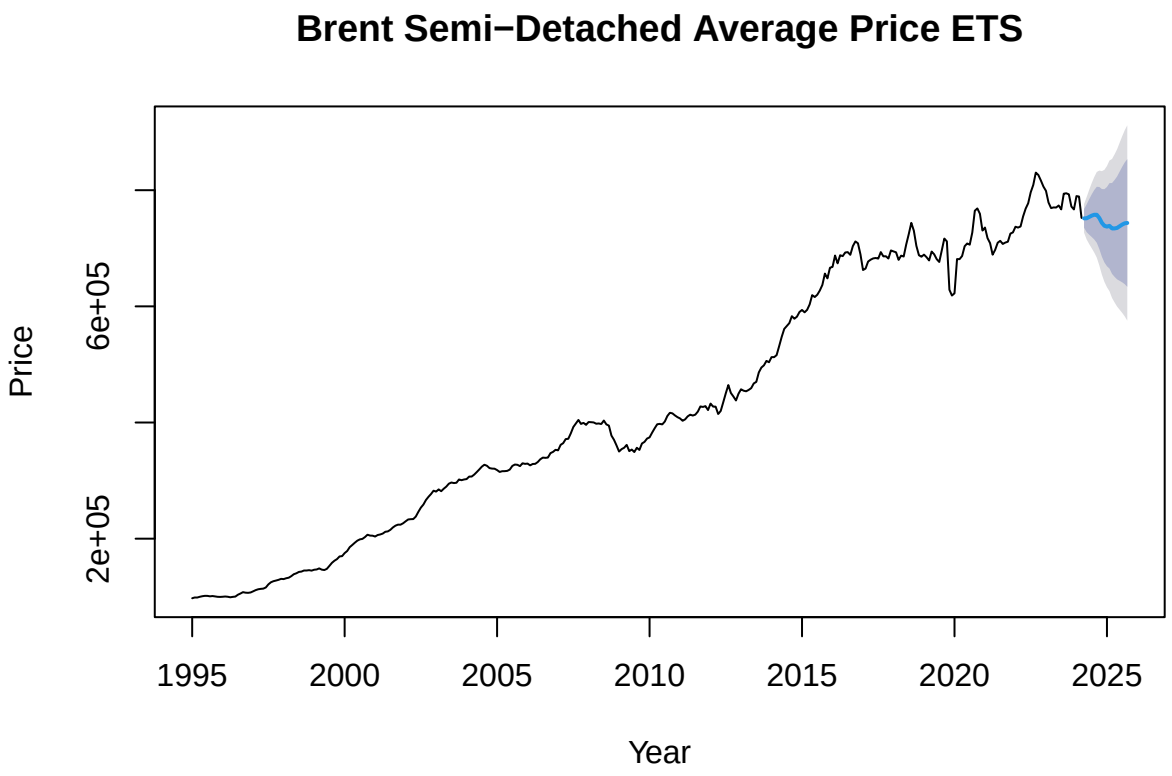



```r
# ETS model for Brent Area
fit_ets_brent_d <- ets(brent_detached_ts)
fit_ets_brent_sd <- ets(brent_semi_detached_ts)
fit_ets_brent_t <- ets(brent_terraced_ts)
fit_ets_brent_f <- ets(brent_flat_ts)

# Forecast using the ETS model for each property type of Brent Area
forecasted_values_ets_brent_d <- forecast(fit_ets_brent_d, h = 18)
forecasted_values_ets_brent_sd <- forecast(fit_ets_brent_sd, h = 18)
forecasted_values_ets_brent_t <- forecast(fit_ets_brent_t, h = 18)
forecasted_values_ets_brent_f <- forecast(fit_ets_brent_f, h = 18)

# Combine the historical and forecasted values
# For each property type by ETS of Brent Area
combined_brent_detached_ts_ets <- ts(c(brent_detached_price,
                                       forecasted_values_ets_brent_d$mean),
                                     start = c(1995, 1), frequency = 12)
combined_brent_semi_detached_ts_ets <- ts(c(brent_semi_detached_price,
                                            forecasted_values_ets_brent_sd$mean),
                                            start = c(1995, 1), frequency = 12)
combined_brent_terraced_ts_ets <- ts(c(brent_terraced_price,
                                       forecasted_values_ets_brent_t$mean),
                                     start = c(1995, 1), frequency = 12)
combined_brent_flat_ts_ets <- ts(c(brent_flat_price,
                                   forecasted_values_ets_brent_f$mean),
                                 start = c(1995, 1), frequency = 12)

# Plot the ETS forecast value for each property type of Brent Area
plot(forecasted_values_ets_brent_d,
```

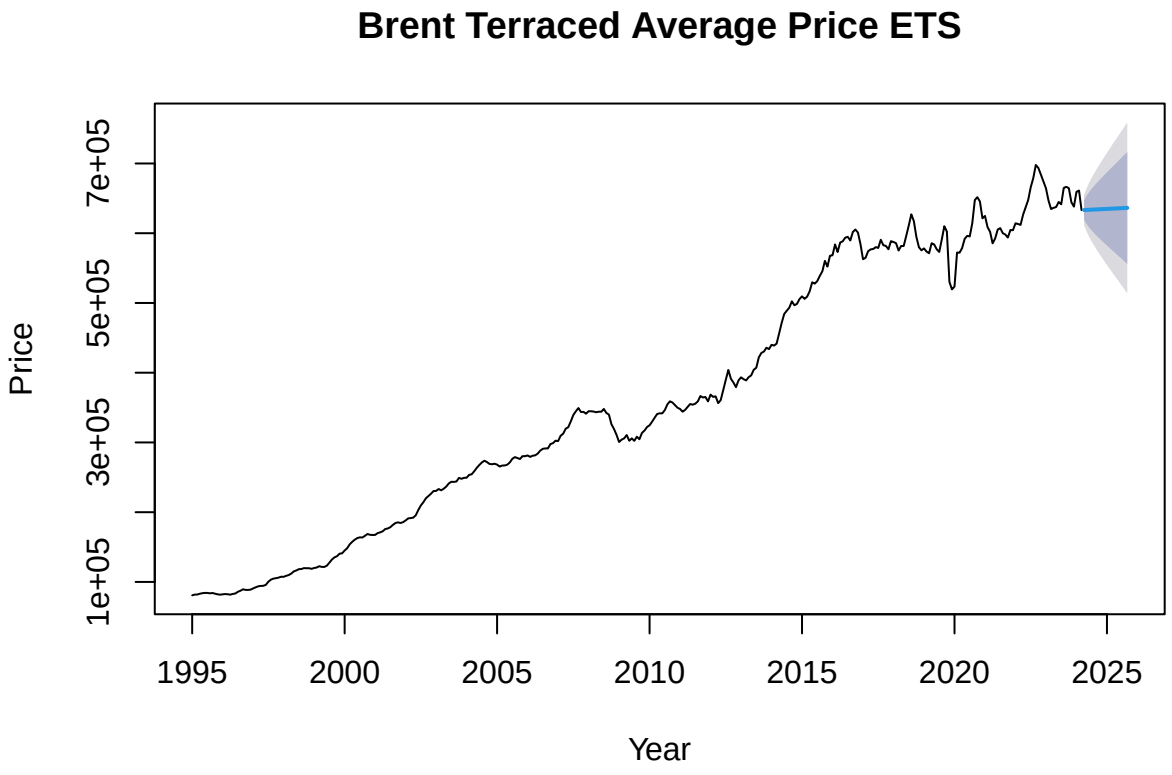
```
     main = "Brent Detached Average Price ETS",
     ylab = "Price", xlab = "Year")
```



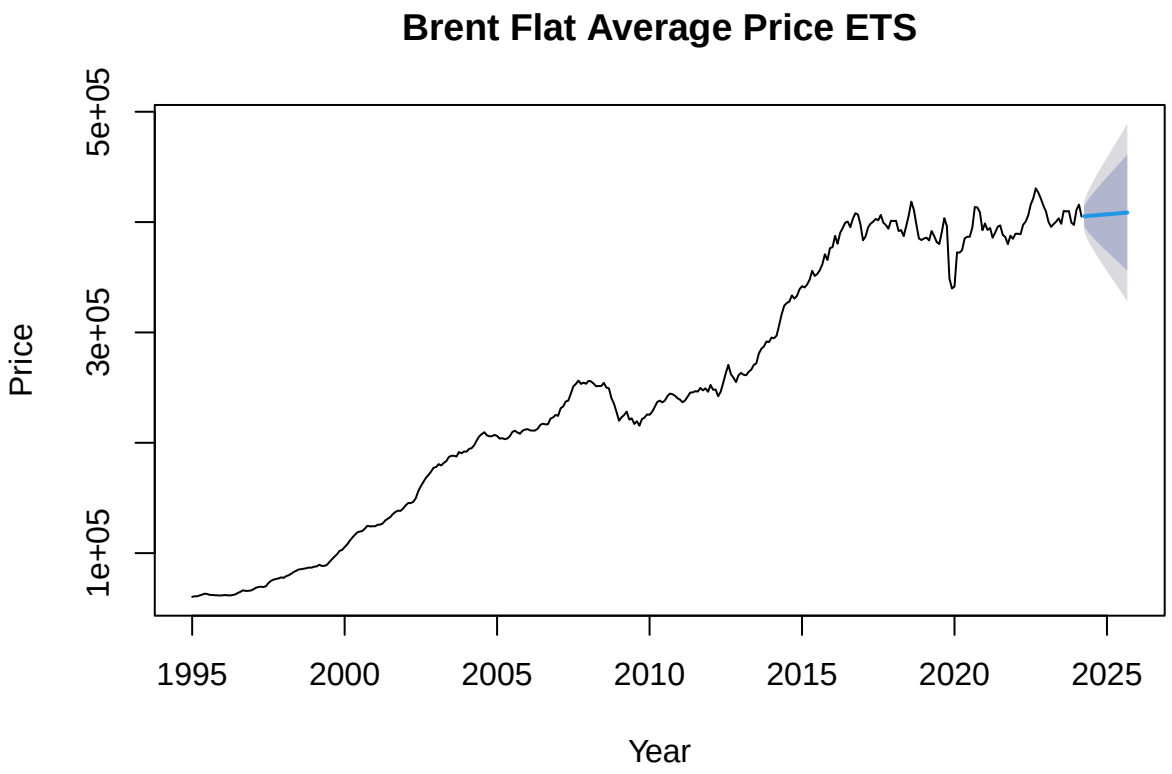

```
plot(forecasted_values_ets_brent_sd,
     main = "Brent Semi-Detached Average Price ETS",
     ylab = "Price", xlab = "Year")
```




```
plot(forecasted_values_ets_brent_t,
     main = "Brent Terraced Average Price ETS",
     ylab = "Price", xlab = "Year")
```



```
plot(forecasted_values_ets_brent_f,
     main = "Brent Flat Average Price ETS",
     ylab = "Price", xlab = "Year")
```




```
# Summary of the ETS model for four different type of properties of Brent Area
# For Detached of Brent Area
print(forecasted_values_ets_brent_d)
```

```
##          Point Forecast     Lo 80   Hi 80     Lo 95   Hi 95
## Apr 2024        1135562 1111079.9 1160045 1098119.6 1173005
## May 2024        1136016 1099802.3 1172229 1080632.0 1191400
## Jun 2024        1139763 1093294.9 1186231 1068696.3 1210829
## Jul 2024        1146579 1090281.8 1202875 1060480.1 1232677
## Aug 2024        1159692 1093374.9 1226009 1058268.8 1261115
## Sep 2024        1164638 1088727.1 1240549 1048542.3 1280734
## Oct 2024        1157962 1073233.0 1242691 1028380.2 1287544
## Nov 2024        1140536 1047911.8 1233161  998879.4 1282193
## Dec 2024        1126299 1025680.0 1226918  972415.6 1280182
## Jan 2025        1117764 1008706.0 1226822  950974.0 1284554
## Feb 2025        1113485  995539.4 1231431  933102.7 1293867
## Mar 2025        1108110  981322.3 1234897  914205.0 1302014
## Apr 2025        1106263  970129.6 1242397  898064.8 1314462
## May 2025        1106642  960734.7 1252549  883496.1 1329788
## Jun 2025        1110229  953916.0 1266541  871169.1 1349288
## Jul 2025        1116803  949395.0 1284212  860774.3 1372833
## Aug 2025        1129511  949727.2 1309295  854555.5 1404466
## Sep 2025        1134262  943020.3 1325505  841782.8 1426742
```

```
summary(fit_ets_brent_d)
```

```
## ETS(M,A,M)
##
## Call:
##  ets(y = brent_detached_ts)
##
##   Smoothing parameters:
##     alpha = 0.9998
##     beta  = 0.0869
##     gamma = 2e-04
##
##   Initial states:
##     l = 154738.8759
##     b = 1661.6836
##     s = 0.9958 1.0062 1.0194 1.023 1.0165 1.0028
##            0.9947 0.9893 0.9868 0.9862 0.9888 0.9904
##
##   sigma:  0.0168
##
##      AIC     AICc      BIC
## 8489.442 8491.280 8555.075
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -134.1591 14198.26 8450.961 0.00582066 1.178831 0.1730639
##                   ACF1
## Training set 0.1281662
```

```
# For Semi-Detached of Brent Area
print(forecasted_values_ets_brent_sd)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       751531.4 735402.7 767660.1 726864.6 776198.1
## May 2024       751644.7 727832.6 775456.7 715227.3 788062.0
## Jun 2024       753930.0 723470.3 784389.8 707345.9 800514.2
## Jul 2024       756138.6 719492.1 792785.1 700092.6 812184.6
## Aug 2024       757706.5 715144.8 800268.2 692614.0 822799.0
## Sep 2024       757640.5 709408.9 805872.1 683876.6 831404.3
## Oct 2024       752112.9 698709.1 805516.8 670438.8 833787.1
## Nov 2024       743787.2 685586.7 801987.7 654777.2 832797.2
## Dec 2024       738656.1 675562.4 801749.9 642162.6 835149.7
## Jan 2025       737296.4 669078.3 805514.5 632965.8 841627.0
## Feb 2025       738605.6 665054.1 812157.2 626118.2 851093.0
## Mar 2025       734117.4 655865.4 812369.5 614441.3 853793.6
## Apr 2025       734202.5 650823.7 817581.4 606685.6 861719.5
## May 2025       735044.2 646476.5 823612.0 599591.5 870496.9
## Jun 2025       737983.0 643976.9 831989.1 594213.1 881753.0
## Jul 2025       740822.6 641380.6 840264.6 588739.1 892906.1
## Aug 2025       743010.4 638215.6 847805.2 582740.5 903280.3
## Sep 2025       743570.9 633664.8 853477.1 575484.0 911657.9
```

```r
summary(fit_ets_brent_sd)
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = brent_semi_detached_ts)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.0832
##     gamma = 1e-04
##     phi   = 0.9555
##
##   Initial states:
##     l = 98065.8348
##     b = 1228.8995
##     s = 0.993 0.9981 1.0073 1.0128 1.0108 1.0065
##            1.0013 0.996 0.9934 0.9918 0.9963 0.9929
##
##   sigma:  0.0167
##
##      AIC     AICc      BIC
## 8187.164 8189.225 8256.659
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 611.9561 9342.572 5483.786 0.1953599 1.162433 0.1694913 0.1162961
```

```r
# For Terraced of Brent Area
print(forecasted_values_ets_brent_t)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       633269.0 619125.0 647413.1 611637.5 654900.5
## May 2024       633448.1 613066.4 653829.8 602277.0 664619.2
## Jun 2024       633627.2 608197.2 659057.2 594735.3 672519.0
## Jul 2024       633806.3 603898.9 663713.6 588067.0 679545.6
## Aug 2024       633985.3 599937.2 668033.4 581913.3 686057.4
## Sep 2024       634164.4 596194.2 672134.6 576094.0 692234.9
## Oct 2024       634343.5 592601.2 676085.8 570504.2 698182.8
## Nov 2024       634522.6 589114.5 679930.7 565076.9 703968.2
## Dec 2024       634701.7 585704.3 683699.0 559766.7 709636.6
## Jan 2025       634880.7 582349.4 687412.1 554541.0 715220.5
## Feb 2025       635059.8 579034.2 691085.5 549376.0 720743.7
## Mar 2025       635238.9 575746.7 694731.2 544253.4 726224.5
## Apr 2025       635418.0 572477.7 698358.3 539159.1 731676.9
## May 2025       635597.1 569219.9 701974.2 534082.0 737112.1
## Jun 2025       635776.1 565967.6 705584.7 529013.1 742539.2
## Jul 2025       635955.2 562715.8 709194.7 523945.2 747965.3
## Aug 2025       636134.3 559460.7 712807.9 518872.1 753396.5
## Sep 2025       636313.4 556199.0 716427.8 513789.0 758837.8
```

```
summary(fit_ets_brent_t)
```

```
## ETS(M,A,N)
##
## Call:
##  ets(y = brent_terraced_ts)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.0372
##
##   Initial states:
##     l = 81421.3009
##     b = 303.6323
##
##   sigma:  0.0174
##
##      AIC     AICc      BIC
## 8083.843 8084.017 8103.147
##
## Training set error measures:
##                      ME    RMSE      MAE        MPE     MAPE     MASE      ACF1
## Training set -9.535812 8572.18 5025.153 0.06702895 1.244866 0.182206 0.2276409
```

```
# For Flat of Brent Area
print(forecasted_values_ets_brent_f)
```

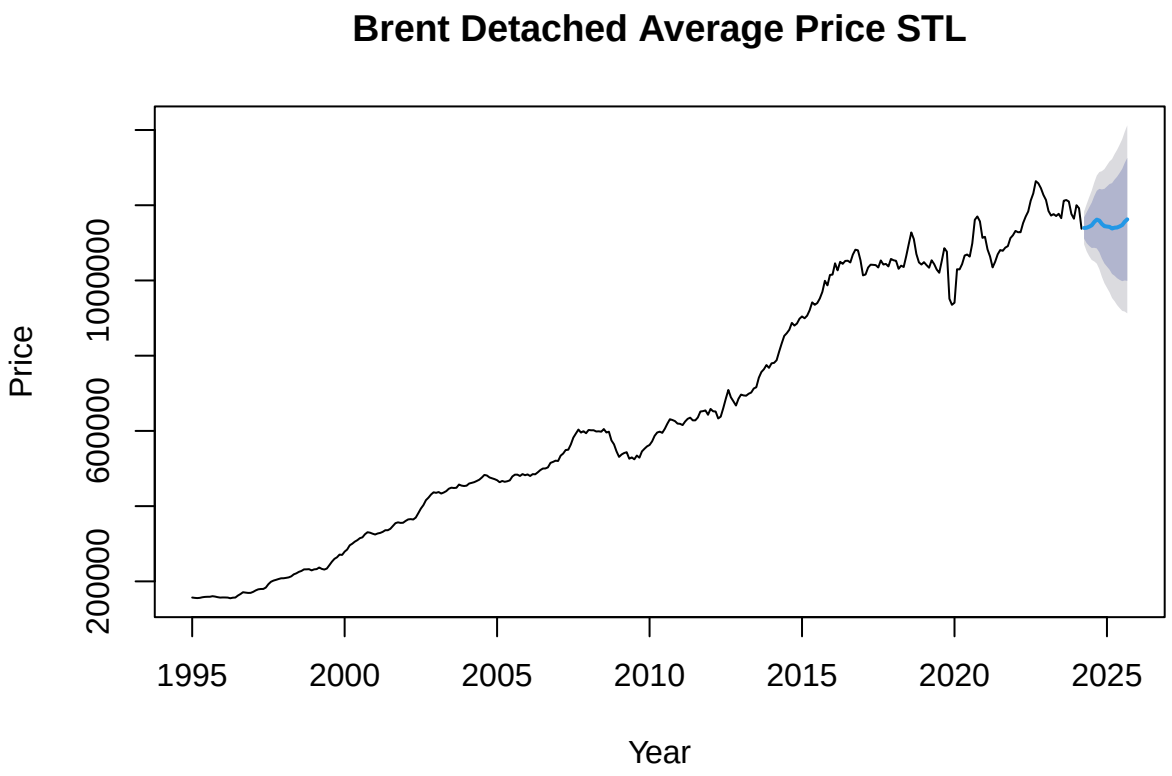
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       405281.5 396542.8 414020.2 391916.8 418646.2
## May 2024       405476.9 392827.9 418125.9 386131.9 424821.8
## Jun 2024       405672.2 389821.6 421522.9 381430.7 429913.7
## Jul 2024       405867.6 387147.5 424587.7 377237.7 434497.5
## Aug 2024       406062.9 384663.8 427462.1 373335.7 438790.2
## Sep 2024       406258.3 382299.4 430217.3 369616.2 442900.4
## Oct 2024       406453.7 380013.2 432894.1 366016.5 446890.9
## Nov 2024       406649.0 377779.3 435518.8 362496.5 450801.5
## Dec 2024       406844.4 375580.0 438108.8 359029.6 454659.2
## Jan 2025       407039.7 373402.8 440676.7 355596.5 458483.0
## Feb 2025       407235.1 371238.7 443231.5 352183.4 462286.8
## Mar 2025       407430.5 369080.9 445780.1 348779.8 466081.1
## Apr 2025       407625.8 366923.9 448327.7 345377.6 469874.0
## May 2025       407821.2 364763.7 450878.7 341970.5 473671.9
## Jun 2025       408016.5 362597.0 453436.1 338553.4 477479.7
## Jul 2025       408211.9 360421.2 456002.6 335122.3 481301.5
## Aug 2025       408407.3 358234.0 458580.5 331673.9 485140.6
## Sep 2025       408602.6 356033.8 461171.4 328205.6 488999.7
```

```
summary(fit_ets_brent_f)
```

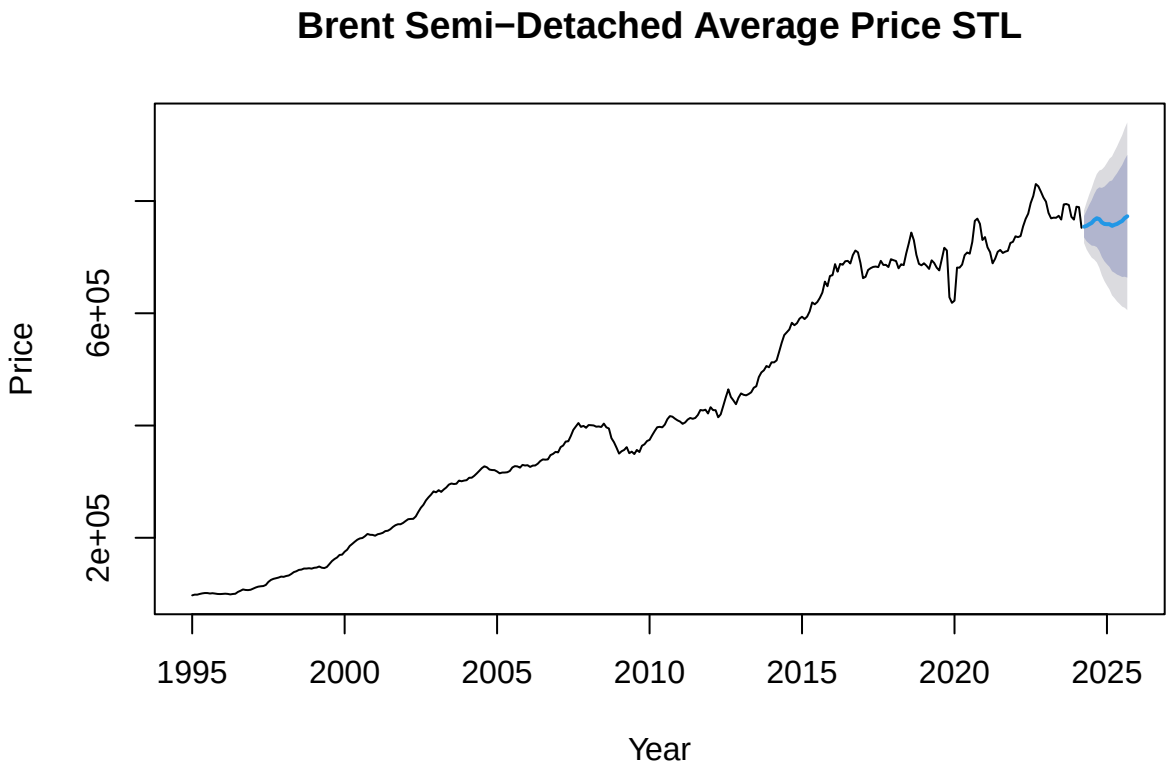
```
## ETS(M,A,N)
##
## Call:
```

```
##  ets(y = brent_flat_ts)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 0.046
##
##   Initial states:
##     l = 60823.0935
##     b = 188.9293
##
##   sigma:  0.0168
##
##      AIC     AICc      BIC
## 7811.656 7811.830 7830.960
##
## Training set error measures:
##                      ME     RMSE     MAE        MPE     MAPE      MASE      ACF1
## Training set 0.3984275 5460.828 3295.79 0.06046445 1.204346 0.1841529 0.1729913
```

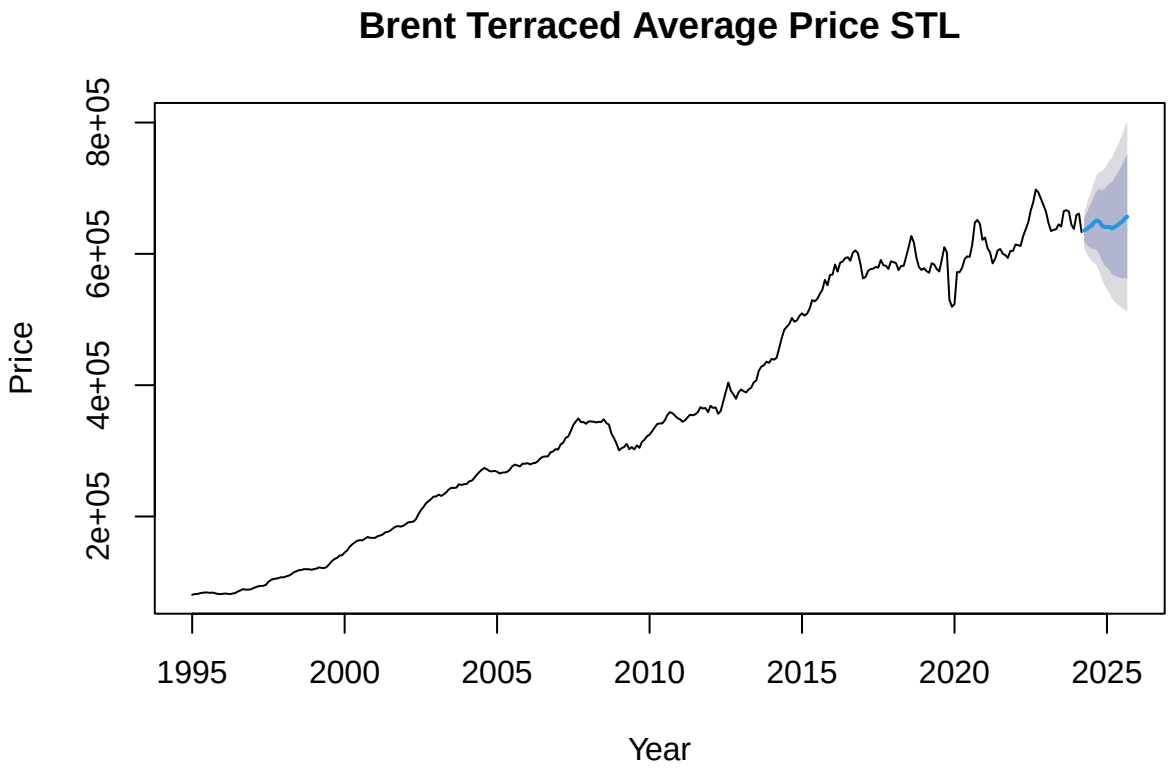
```
# STL model for Brent Area
stl_brent_d <- stl(brent_detached_ts, s.window = "periodic")
stl_brent_sd <- stl(brent_semi_detached_ts, s.window = "periodic")
stl_brent_t <- stl(brent_terraced_ts, s.window = "periodic")
stl_brent_f <- stl(brent_flat_ts, s.window = "periodic")

# Forecast using the STL model
# For each property type of Brent Area
forecasted_values_stl_brent_d <- forecast(stl_brent_d, method='ets',h = 18)
forecasted_values_stl_brent_sd <- forecast(stl_brent_sd, method='ets', h = 18)
forecasted_values_stl_brent_t <- forecast(stl_brent_t, method='ets', h = 18)
forecasted_values_stl_brent_f <- forecast(stl_brent_f, method='ets', h = 18)

# Combine the historical and forecasted values
# For each property type by STL of Brent Area
combined_brent_detached_ts_stl <- ts(c(brent_detached_price,
                                       forecasted_values_stl_brent_d$mean),
                                     start = c(1995, 1), frequency = 12)
combined_brent_semi_detached_ts_stl <- ts(c(brent_semi_detached_price,
                                          forecasted_values_stl_brent_sd$mean),
                                          start = c(1995, 1), frequency = 12)
combined_brent_terraced_ts_stl <- ts(c(brent_terraced_price,
                                       forecasted_values_stl_brent_t$mean),
                                     start = c(1995, 1), frequency = 12)
combined_brent_flat_ts_stl <- ts(c(brent_flat_price,
                                   forecasted_values_stl_brent_f$mean),
                                 start = c(1995, 1), frequency = 12)

# Plot the STL forecast value for each property type of Brent Area
plot(forecasted_values_stl_brent_d,
     main = "Brent Detached Average Price STL",
     ylab = "Price", xlab = "Year")
```



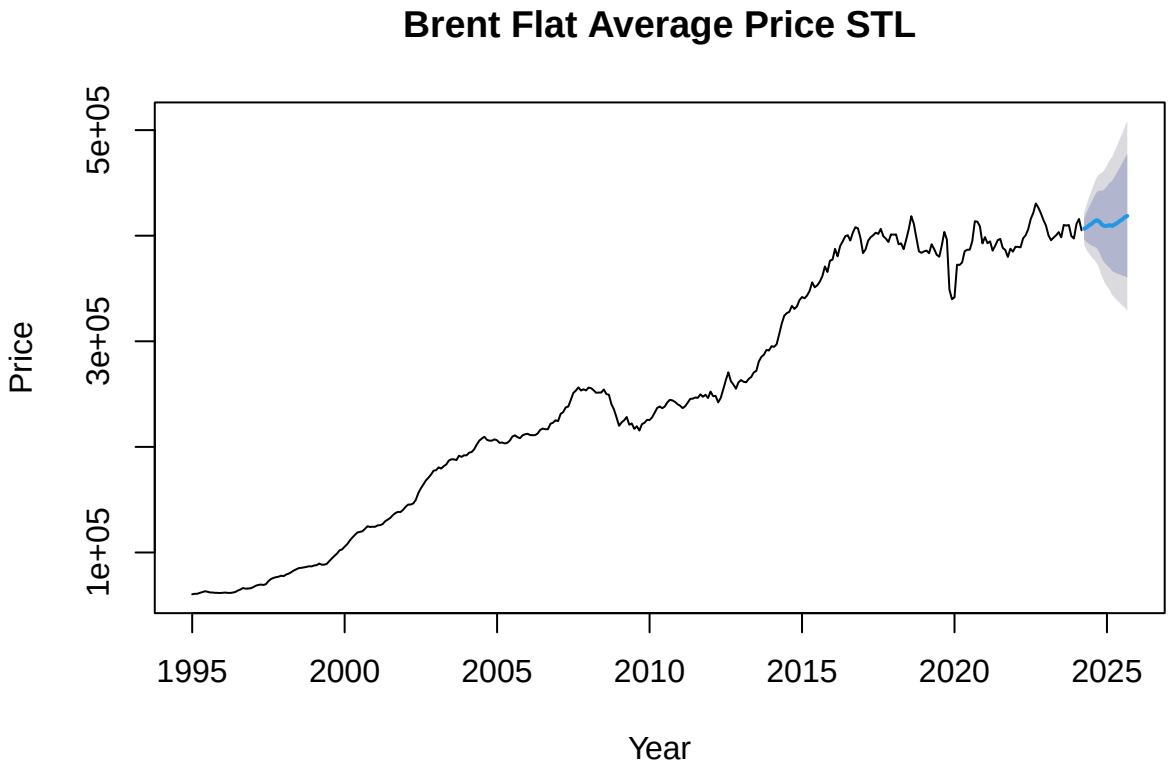

```
plot(forecasted_values_stl_brent_sd,
     main = "Brent Semi-Detached Average Price STL",
     ylab = "Price", xlab = "Year")
```



```
plot(forecasted_values_stl_brent_t,
     main = "Brent Terraced Average Price STL",
     ylab = "Price", xlab = "Year")
```




```
plot(forecasted_values_stl_brent_f,
     main = "Brent Flat Average Price STL",
     ylab = "Price", xlab = "Year")
```




```
# Summary of the STL model for four different type of properties of Brent Area
# For Detached of Brent Area
print(forecasted_values_stl_brent_d)
```

```
##          Point Forecast     Lo 80   Hi 80     Lo 95   Hi 95
## Apr 2024        1139640 1111273.4 1168006 1096257.1 1183023
## May 2024        1140200 1099273.6 1181127 1077608.2 1202793
## Jun 2024        1143017 1091891.1 1194142 1064826.9 1221206
## Jul 2024        1146637 1086439.9 1206834 1054573.4 1238701
## Aug 2024        1155583 1086973.3 1224193 1050653.5 1260513
## Sep 2024        1161761 1085162.7 1238359 1044614.2 1278907
## Oct 2024        1159243 1074945.3 1243541 1030320.6 1288166
## Nov 2024        1150246 1058448.9 1242043 1009854.5 1290638
## Dec 2024        1144399 1045244.6 1243552  992755.7 1296041
## Jan 2025        1143391 1036980.1 1249801  980649.7 1306132
## Feb 2025        1142562 1028963.7 1256160  968828.6 1316295
## Mar 2025        1138112 1017372.9 1258852  953457.3 1322767
## Apr 2025        1140357 1012503.8 1268210  944822.5 1335891
## May 2025        1140917 1005964.1 1275871  934524.1 1347311
## Jun 2025        1143734 1001682.0 1285785  926484.4 1360983
## Jul 2025        1147354  998197.0 1296511  919238.1 1375470
## Aug 2025        1156300 1000022.7 1312578  917294.4 1395306
## Sep 2025        1162478  999058.6 1325897  912549.7 1412406
```

```
summary(stl_brent_d)
```

```
##  Call:
##  stl(x = brent_detached_ts, s.window = "periodic")
```

```
##
##  Time.series components:
##     seasonal             trend              remainder
##  Min.   :-9282.223   Min.   : 157438.8   Min.   :-82598.44
##  1st Qu.:-5684.823   1st Qu.: 379753.8   1st Qu.: -6202.32
##  Median :-3840.205   Median : 588836.9   Median :   640.91
##  Mean   :  -51.110   Mean   : 651388.8   Mean   :     4.60
##  3rd Qu.: 3090.502   3rd Qu.:1032757.8   3rd Qu.:  5164.47
##  Max.   :14724.706   Max.   :1214038.2   Max.   : 55965.26
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        8775       653004     11367        662802
##    %   1.3          98.5       1.7         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Semi-Detached of Brent Area
print(forecasted_values_stl_brent_sd)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       754264.0 735160.5 773367.4 725047.7 783480.2
## May 2024       755745.5 728197.4 783293.7 713614.2 797876.8
## Jun 2024       758558.2 724162.3 792954.1 705954.2 811162.2
## Jul 2024       761019.7 720540.1 801499.4 699111.4 822928.1
## Aug 2024       766217.2 720102.0 812332.4 695690.1 836744.4
## Sep 2024       769336.0 717874.9 820797.0 690633.0 848038.9
## Oct 2024       767703.2 711093.7 824312.7 681126.4 854279.9
## Nov 2024       761822.6 700203.4 823441.9 667584.2 856061.1
## Dec 2024       759126.2 692595.9 825656.4 657377.0 860875.4
## Jan 2025       758711.7 687340.7 830082.6 649559.2 867864.1
## Feb 2025       758817.5 682655.2 834979.9 642337.3 875297.8
## Mar 2025       755718.5 674798.2 836638.8 631961.5 879475.5
## Apr 2025       757756.0 672098.7 843413.2 626754.5 888757.4
## May 2025       759237.5 668854.8 849620.3 621009.0 897466.1
## Jun 2025       762050.2 666945.3 857155.0 616599.9 907500.4
## Jul 2025       764511.7 664682.1 864341.4 611835.4 917188.1
## Aug 2025       769709.2 665146.6 874271.8 609794.5 929623.9
## Sep 2025       772828.0 663520.0 882135.9 605655.9 940000.0
```

```
summary(stl_brent_sd)
```

```
##  Call:
##  stl(x = brent_semi_detached_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend            remainder
```

```
##  Min.   :-6468.674   Min.   :100083.5   Min.   :-54077.07
##  1st Qu.:-3305.157   1st Qu.:243184.8   1st Qu.: -4417.18
##  Median :-2188.029   Median :393113.7   Median :   380.16
##  Mean   :  -35.444   Mean   :428151.6   Mean   :    -5.44
##  3rd Qu.: 1160.558   3rd Qu.:681466.3   3rd Qu.:  3511.04
##  Max.   : 8894.769   Max.   :797384.2   Max.   : 34888.13
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        4466       438281      7928        444330
##    %   1.0         98.6       1.8         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Terraced of Brent Area
print(forecasted_values_stl_brent_t)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       635552.3 618709.6 652394.9 609793.6 661310.9
## May 2024       637529.1 613276.7 661781.4 600438.2 674619.9
## Jun 2024       640815.4 610577.7 671053.1 594570.8 687060.0
## Jul 2024       643365.2 607828.6 678901.9 589016.6 697713.9
## Aug 2024       648250.1 607820.6 688679.5 586418.6 710081.6
## Sep 2024       650835.2 605778.1 695892.2 581926.4 719744.0
## Oct 2024       648842.5 599340.9 698344.2 573136.4 724548.7
## Nov 2024       643459.5 589643.9 697275.1 561155.6 725763.4
## Dec 2024       641005.2 582970.4 699040.0 552248.6 729761.7
## Jan 2025       640920.1 578735.4 703104.7 545816.8 736023.3
## Feb 2025       641131.9 574847.8 707416.0 539759.2 742504.7
## Mar 2025       638772.7 568425.4 709120.1 531185.8 746359.7
## Apr 2025       641235.3 566849.6 715620.9 527472.2 754998.3
## May 2025       643212.1 564804.2 721619.9 523297.6 763126.5
## Jun 2025       646498.4 564077.4 728919.4 520446.3 772550.4
## Jul 2025       649048.2 562617.2 735479.2 516863.4 781233.0
## Aug 2025       653933.1 563490.5 744375.6 515613.1 792253.0
## Sep 2025       656518.2 562058.5 750977.9 512054.5 800981.9
```

```
summary(stl_brent_t)
```

```
##  Call:
##  stl(x = brent_terraced_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend            remainder
##  Min.   :-6371.886   Min.   : 82716.5   Min.   :-47342.61
##  1st Qu.:-3539.148   1st Qu.:200131.7   1st Qu.: -4131.86
##  Median :-2718.710   Median :338972.8   Median :   540.76
```

```
##  Mean   :  -37.574   Mean   :362453.8   Mean   :   -18.86
##  3rd Qu.: 2009.260   3rd Qu.:577033.5   3rd Qu.:  3305.24
##  Max.   : 8532.049   Max.   :665299.7   Max.   : 31277.07
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        5548       376902      7437        379553
##    %   1.5         99.3       2.0         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Flat of Brent Area
print(forecasted_values_stl_brent_f)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       406546.1 396859.3 416232.9 391731.4 421360.7
## May 2024       407864.0 393837.8 421890.2 386412.8 429315.2
## Jun 2024       409856.5 392273.9 427439.0 382966.3 436746.6
## Jul 2024       411257.9 390485.5 432030.3 379489.3 443026.5
## Aug 2024       413458.9 389706.0 437211.7 377132.0 449785.8
## Sep 2024       414667.5 388064.9 441270.1 373982.3 455352.7
## Oct 2024       413539.9 384173.0 442906.9 368627.1 458452.8
## Nov 2024       410444.7 378370.0 442519.4 361390.7 459498.7
## Dec 2024       409212.3 374467.0 443957.7 356073.9 462350.8
## Jan 2025       409247.5 371854.8 446640.1 352060.3 466434.6
## Feb 2025       410118.1 370091.5 450144.8 348902.6 471333.6
## Mar 2025       409152.1 366497.2 451807.1 343917.0 474387.3
## Apr 2025       410612.2 365328.8 455895.6 341357.2 479867.2
## May 2025       411930.1 364013.6 459846.7 338648.1 485212.2
## Jun 2025       413922.6 363364.5 464480.6 336600.7 491244.5
## Jul 2025       415324.0 362113.2 468534.8 333945.1 496702.9
## Aug 2025       417525.0 361647.8 473402.2 332068.2 502981.8
## Sep 2025       418733.6 360174.5 477292.8 329175.1 508292.2
```

```
summary(stl_brent_f)
```

```
##  Call:
##  stl(x = brent_flat_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend            remainder
##  Min.   :-3158.617   Min.   : 61969.2   Min.   :-31347.912
##  1st Qu.:-2081.918   1st Qu.:152513.2   1st Qu.: -2441.169
##  Median :-1058.328   Median :241268.1   Median :    69.704
##  Mean   :  -21.077   Mean   :248156.1   Mean   :   -20.251
##  3rd Qu.: 1657.895   3rd Qu.:381088.6   3rd Qu.:  2132.059
##  Max.   : 4389.807   Max.   :412594.5   Max.   : 22837.044
```

```
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        3740       228575      4573        234549
##    %   1.6         97.5       1.9         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# Split the data into training and test sets of Brent Area
train_end <- c(2023, 6)
test_start <- c(2023, 6)

#Detached property of Brent Area
# By ARIMA model for Detached of Brent Area
brent_detached_train_arima <- window(brent_detached_ts_diff, end = train_end)

# Fit specified ARIMA models to the training data for Detached of Brent Area
fit_arima_brent_d_train <- Arima(brent_detached_train_arima,
                                 order = c(0, 0, 5))
forecasted_values_arima_brent_d_train <- forecast(fit_arima_brent_d_train,
                                                  h = 9)

# Add the forecasted differenced values to the last observed value of Brent Area
brent_detached_new_ts <- ts(brent_detached_price,
                            start = c(1995, 1), end = c(2023, 6), frequency = 12)
last_value_brent_detached <- as.numeric(tail(brent_detached_new_ts, n = 1))
forecasted_values_brent_detached_combined <- c(last_value_brent_detached,
                                   forecasted_values_arima_brent_d_train$mean)
cumulative_forecasted_values_brent_detached <-
  cumsum(forecasted_values_brent_detached_combined)
forecasted_values_arima_brent_d_test <-
  ts(cumulative_forecasted_values_brent_detached,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Detached by ARIMA of Brent Area
mse_brent_detached_arima <- mean((window(brent_detached_ts,
                                         start=test_start) -
                                   forecasted_values_arima_brent_d_test)^2)
mae_brent_detached_arima <- mean(abs(window(brent_detached_ts,
                                            start=test_start) -
                                       forecasted_values_arima_brent_d_test))

# By ETS model for Detached of Brent Area
brent_detached_train_ets <- window(brent_detached_ts, end = train_end)
brent_detached_test_ets <- window(brent_detached_ts, start = test_start)

# Fit ETS models to the training data for Detached of Brent Area
fit_ets_brent_d_train <- ets(brent_detached_train_ets)
```

```r
# Forecast the test period for Detached of Brent Area
forecasted_values_ets_brent_d_test <- forecast(fit_ets_brent_d_train, h = 9)

# Calculate MSE and MAE by ETS for Detached of Brent Area
mse_brent_detached_ets <- mean((brent_detached_test_ets -
                                  forecasted_values_ets_brent_d_test$mean)^2)
mae_brent_detached_ets <- mean(abs(brent_detached_test_ets -
                                     forecasted_values_ets_brent_d_test$mean))

# By STL model for Detached of Brent Area
brent_detached_train_stl <- window(brent_detached_ts, end = train_end)
brent_detached_test_stl <- window(brent_detached_ts, start = test_start)
fit_stl_brent_d_train <- stl(brent_detached_train_stl, s.window = "periodic")
forecasted_values_stl_brent_d_test <- forecast(fit_stl_brent_d_train,
                                               method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Detached of Brent Area
mse_brent_detached_stl <- mean((brent_detached_test_stl -
                                  forecasted_values_stl_brent_d_test$mean)^2)
mae_brent_detached_stl <- mean(abs(brent_detached_test_stl -
                                     forecasted_values_stl_brent_d_test$mean))

# Print MSE and MAE of Brent Area
print(paste("Brent Detached MSE for Arima:", mse_brent_detached_arima))
```

```
## [1] "Brent Detached MSE for Arima: 719073122.859176"
```

```r
print(paste("Brent Detached MAE for Arima:", mae_brent_detached_arima))
```

```
## [1] "Brent Detached MAE for Arima: 19391.7938886515"
```

```r
print(paste("Brent Detached MSE for ETS:", mse_brent_detached_ets))
```

```
## [1] "Brent Detached MSE for ETS: 765145364.383674"
```

```r
print(paste("Brent Detached MAE for ETS:", mae_brent_detached_ets))
```

```
## [1] "Brent Detached MAE for ETS: 22613.7702735772"
```

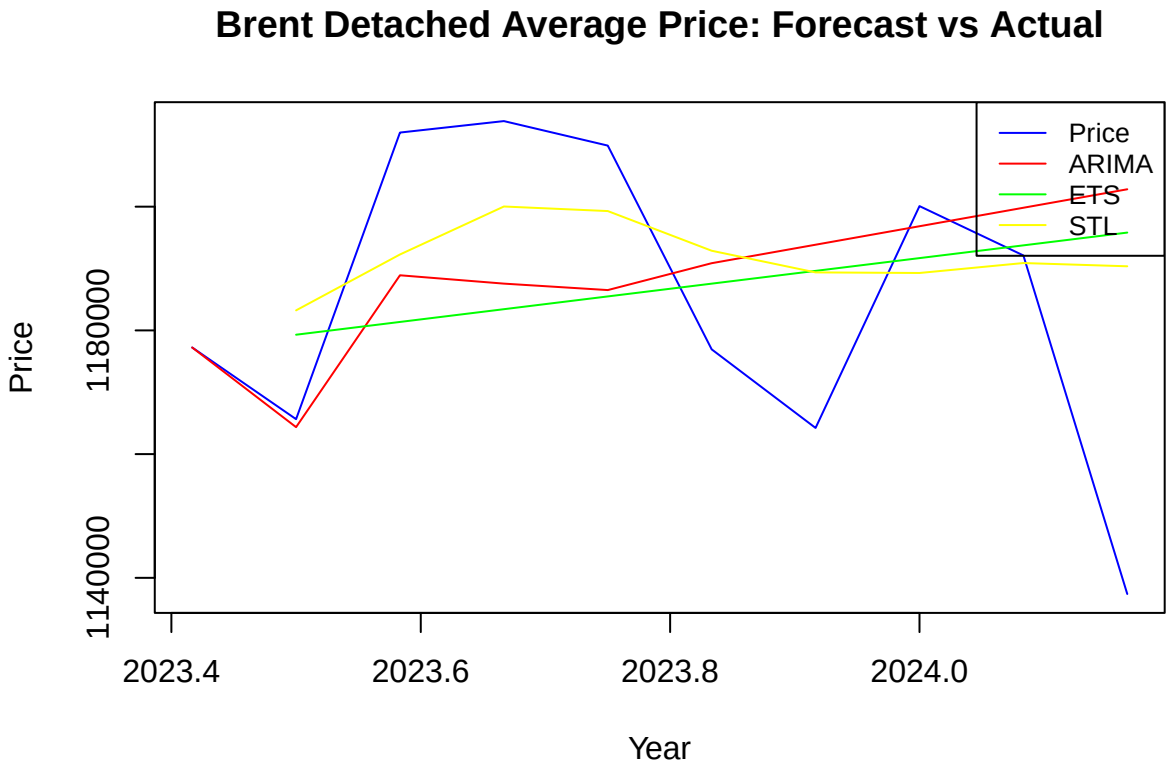
```r
print(paste("Brent Detached MSE for STL:", mse_brent_detached_stl))
```

```
## [1] "Brent Detached MSE for STL: 534549456.298435"
```

```r
print(paste("Brent Detached MAE for STL:", mae_brent_detached_stl))
```

```
## [1] "Brent Detached MAE for STL: 18642.00743622"
```

```
# Plot the combined time series with forecast for Detached of Brent Area
plot(window(brent_detached_ts, start = train_end), type = "l", col = "blue",
     main = "Brent Detached Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year",
     ylim = range(c(window(brent_detached_ts, start = train_end),
                    forecasted_values_arima_brent_d_test,
                    forecasted_values_ets_brent_d_test$mean,
                    forecasted_values_stl_brent_d_test$mean)))
lines(forecasted_values_arima_brent_d_test, col = "red")
lines(forecasted_values_ets_brent_d_test$mean, col = "green")
lines(forecasted_values_stl_brent_d_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```
# Semi-Detached property of Brent Area
# By ARIMA model for Semi-Detached of Brent Area
brent_semi_detached_train_arima <- window(brent_semi_detached_ts_diff,
                                          end = train_end)

# Fit best-fit ARIMA models to the training data for Semi-Detached of Brent Area
fit_arima_brent_sd_train <- Arima(brent_semi_detached_train_arima,
                                  order = c(0, 0, 5))
forecasted_values_arima_brent_sd_train <- forecast(fit_arima_brent_sd_train,
                                                   h = 9)

# Add the forecasted differenced values to the last observed value of Brent Area
brent_semi_detached_new_ts <- ts(brent_semi_detached_price, start = c(1995, 1),
                                 end = c(2023, 6),  frequency = 12)
```

```
last_value_brent_semi_detached <- as.numeric(tail(brent_semi_detached_new_ts,
                                                  n = 1))
forecasted_values_brent_semi_detached_combined <-
  c(last_value_brent_semi_detached,
    forecasted_values_arima_brent_sd_train$mean)
cumulative_forecasted_values_brent_semi_detached <-
  cumsum(forecasted_values_brent_semi_detached_combined)
forecasted_values_arima_brent_sd_test <-
  ts(cumulative_forecasted_values_brent_semi_detached,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Semi-Detached by ARIMA of Brent Area
mse_brent_semi_detached_arima <- mean((window(brent_semi_detached_ts,
                                              start=test_start) -
                                       forecasted_values_arima_brent_sd_test)^2)
mae_brent_semi_detached_arima <- mean(abs(window(brent_semi_detached_ts,
                                                 start=test_start) -
                                        forecasted_values_arima_brent_sd_test))

# By ETS model for Semi-Detached of Brent Area
brent_semi_detached_train_ets <- window(brent_semi_detached_ts, end = train_end)
brent_semi_detached_test_ets <- window(brent_semi_detached_ts,
                                       start = test_start)

# Fit ETS models to the training data for Semi-Detached of Brent Area
fit_ets_brent_sd_train <- ets(brent_semi_detached_train_ets)

# Forecast the test period for Semi-Detached of Brent Area
forecasted_values_ets_brent_sd_test <- forecast(fit_ets_brent_sd_train, h = 9)

# Calculate MSE and MAE by ETS for Semi-Detached of Brent Area
mse_brent_semi_detached_ets <- mean((brent_semi_detached_test_ets -
                                     forecasted_values_ets_brent_sd_test$mean)^2)
mae_brent_semi_detached_ets <- mean(abs(brent_semi_detached_test_ets -
                                       forecasted_values_ets_brent_sd_test$mean))

# By STL model for Semi-Detached of Brent Area
brent_semi_detached_train_stl <- window(brent_semi_detached_ts, end = train_end)
brent_semi_detached_test_stl <- window(brent_semi_detached_ts,
                                       start = test_start)
fit_stl_brent_sd_train <- stl(brent_semi_detached_train_stl,
                              s.window = "periodic")
forecasted_values_stl_brent_sd_test <- forecast(fit_stl_brent_sd_train,
                                                method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Semi-Detached of Brent Area
mse_brent_semi_detached_stl <- mean((brent_semi_detached_test_stl -
                                     forecasted_values_stl_brent_sd_test$mean)^2)
mae_brent_semi_detached_stl <- mean(abs(brent_semi_detached_test_stl -
                                       forecasted_values_stl_brent_sd_test$mean))

# Print MSE and MAE for Semi-Detached of Brent Area
print(paste("Brent Semi-Detached MSE for ARIMA:",
```

```
            mse_brent_semi_detached_arima))
```

```
## [1] "Brent Semi-Detached MSE for ARIMA: 240503401.526663"
```

```
print(paste("Brent Semi-Detached MAE for ARIMA:",
            mae_brent_semi_detached_arima))
```

```
## [1] "Brent Semi-Detached MAE for ARIMA: 11908.0549804835"
```

```
print(paste("Brent Semi-Detached MSE for ETS:",
            mse_brent_semi_detached_ets))
```

```
## [1] "Brent Semi-Detached MSE for ETS: 258701315.053434"
```

```
print(paste("Brent Semi-Detached MAE for ETS:",
            mae_brent_semi_detached_ets))
```

```
## [1] "Brent Semi-Detached MAE for ETS: 13885.9328324341"
```

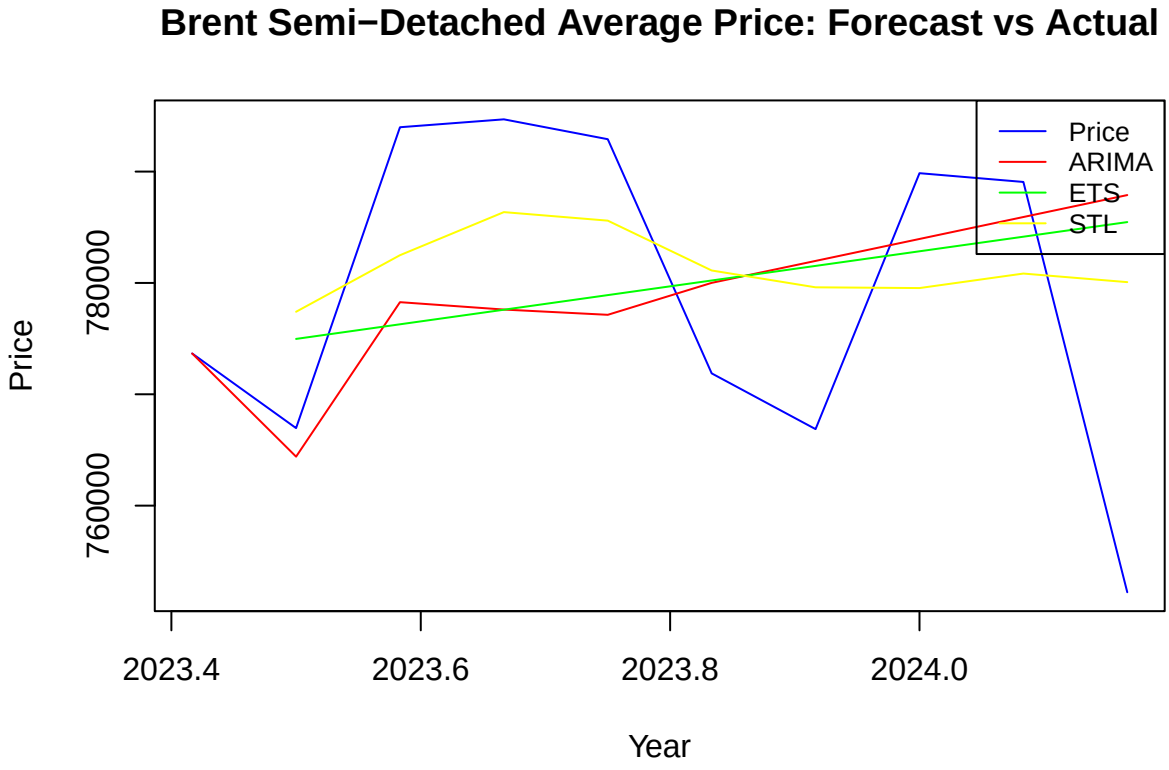
```
print(paste("Brent Semi-Detached MSE for STL:",
            mse_brent_semi_detached_stl))
```

```
## [1] "Brent Semi-Detached MSE for STL: 173467546.618505"
```

```
print(paste("Brent Semi-Detached MAE for STL:",
            mae_brent_semi_detached_stl))
```

```
## [1] "Brent Semi-Detached MAE for STL: 11771.8204146328"
```

```
# Plot the combined time series with forecast for Semi-Detached of Brent Area
plot(window(brent_semi_detached_ts, start = train_end), type = "l",
     col = "blue",
     main = "Brent Semi-Detached Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year",
     ylim = range(c(window(brent_semi_detached_ts, start = train_end),
                    forecasted_values_arima_brent_sd_test,
                    forecasted_values_ets_brent_sd_test$mean,
                    forecasted_values_stl_brent_sd_test$mean)))
lines(forecasted_values_arima_brent_sd_test, col = "red")
lines(forecasted_values_ets_brent_sd_test$mean, col = "green")
lines(forecasted_values_stl_brent_sd_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```
# Terraced property of Brent Area
# By ARIMA model for Terraced of Brent Area
brent_terraced_train_arima <- window(brent_terraced_ts_diff, end = train_end)

# Fit specified ARIMA models to the training data for Terraced of Brent Area
fit_arima_brent_t_train <- Arima(brent_terraced_train_arima,
                                 order = c(2, 0, 5))
forecasted_values_arima_brent_t_train <- forecast(fit_arima_brent_t_train,
                                                  h = 9)

# Add the forecasted differenced values of Terraced in Brent Area
brent_terraced_new_ts <- ts(brent_terraced_price, start = c(1995, 1),
                            end = c(2023, 6), frequency = 12)
last_value_brent_terraced <- as.numeric(tail(brent_terraced_new_ts, n = 1))
forecasted_values_brent_terraced_combined <-
  c(last_value_brent_terraced,
    forecasted_values_arima_brent_t_train$mean)
cumulative_forecasted_values_brent_terraced <-
  cumsum(forecasted_values_brent_terraced_combined)
forecasted_values_arima_brent_t_test <-
  ts(cumulative_forecasted_values_brent_terraced,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Terraced by ARIMA of Brent Area
mse_brent_terraced_arima <- mean((window(brent_terraced_ts, start=test_start) -
                                    forecasted_values_arima_brent_t_test)^2)
mae_brent_terraced_arima <- mean(abs(window(brent_terraced_ts,
                                            start=test_start) -
                                       forecasted_values_arima_brent_t_test))
```

```r
# By ETS model for Terraced of Brent Area
brent_terraced_train_ets <- window(brent_terraced_ts, end = train_end)
brent_terraced_test_ets <- window(brent_terraced_ts, start = test_start)

# Fit ETS models to the training data for Terraced of Brent Area
fit_ets_brent_t_train <- ets(brent_terraced_train_ets)

# Forecast the test period for Terraced of Brent Area
forecasted_values_ets_brent_t_test <- forecast(fit_ets_brent_t_train, h = 9)

# Calculate MSE and MAE by ETS for Terraced of Brent Area
mse_brent_terraced_ets <- mean((brent_terraced_test_ets -
                                  forecasted_values_ets_brent_t_test$mean)^2)
mae_brent_terraced_ets <- mean(abs(brent_terraced_test_ets -
                                     forecasted_values_ets_brent_t_test$mean))

# By STL model for Terraced of Brent Area
brent_terraced_train_stl <- window(brent_terraced_ts, end = train_end)
brent_terraced_test_stl <- window(brent_terraced_ts, start = test_start)
fit_stl_brent_t_train <- stl(brent_terraced_train_stl, s.window = "periodic")
forecasted_values_stl_brent_t_test <- forecast(fit_stl_brent_t_train,
                                               method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Terraced of Brent Area
mse_brent_terraced_stl <- mean((brent_terraced_test_stl -
                                  forecasted_values_stl_brent_t_test$mean)^2)
mae_brent_terraced_stl <- mean(abs(brent_terraced_test_stl -
                                     forecasted_values_stl_brent_t_test$mean))

# Print MSE and MAE of Terraced in Brent Area
print(paste("Brent Terraced MSE for ARIMA:", mse_brent_terraced_arima))
```

```
## [1] "Brent Terraced MSE for ARIMA: 200465734.613222"
```

```r
print(paste("Brent Terraced MAE for ARIMA:", mae_brent_terraced_arima))
```

```
## [1] "Brent Terraced MAE for ARIMA: 10693.0071390128"
```

```r
print(paste("Brent Terraced MSE for ETS:", mse_brent_terraced_ets))
```

```
## [1] "Brent Terraced MSE for ETS: 179735961.58828"
```

```r
print(paste("Brent Terraced MAE for ETS:", mae_brent_terraced_ets))
```

```
## [1] "Brent Terraced MAE for ETS: 10257.3682407471"
```

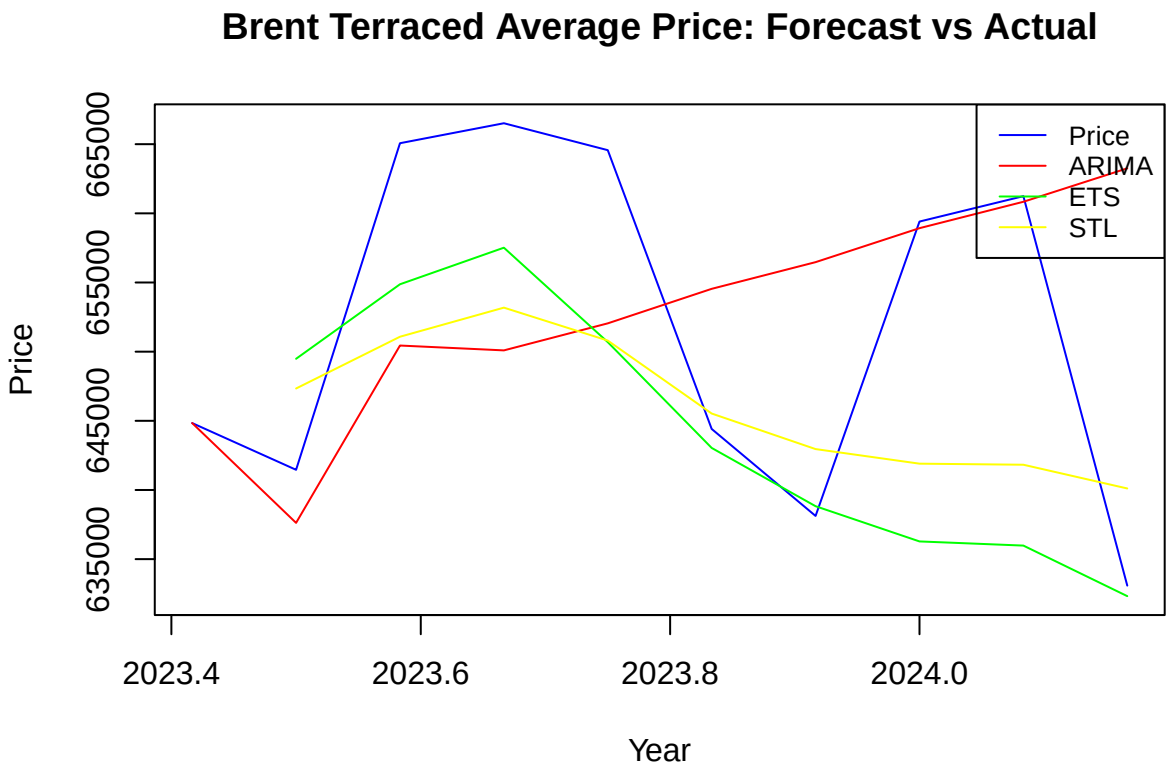
```r
print(paste("Brent Terraced MSE for STL:", mse_brent_terraced_stl))
```

```
## [1] "Brent Terraced MSE for STL: 150517030.927186"
```

```r
print(paste("Brent Terraced MAE for STL:", mae_brent_terraced_stl))
```

```
## [1] "Brent Terraced MAE for STL: 10761.2839519828"
```

```r
# Plot the combined time series with forecast for Terraced of Brent Area
plot(window(brent_terraced_ts, start = train_end), type = "l", col = "blue",
     main = "Brent Terraced Average Price: Forecast vs Actual", ylab = "Price",
     xlab = "Year", ylim = range(c(window(brent_terraced_ts, start = train_end),
                                   forecasted_values_arima_brent_t_test,
                                   forecasted_values_ets_brent_t_test$mean,
                                   forecasted_values_stl_brent_t_test$mean)))
lines(forecasted_values_arima_brent_t_test, col = "red")
lines(forecasted_values_ets_brent_t_test$mean, col = "green")
lines(forecasted_values_stl_brent_t_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```r
# Flat property of Brent Area
# By ARIMA model for Flat of Brent Area
brent_flat_train_arima <- window(brent_flat_ts_diff, end = train_end)

# Fit specified ARIMA models to the training data for Flat of Brent Area
fit_arima_brent_f_train <- Arima(brent_flat_train_arima, order = c(0, 0, 3))
forecasted_values_arima_brent_f_train <- forecast(fit_arima_brent_f_train,
                                                  h = 9)

# Add the forecasted differenced values to the last observed value of Brent Area
```

```r
brent_flat_new_ts <- ts(brent_flat_price, start = c(1995, 1),
                        end = c(2023, 6), frequency = 12)
last_value_brent_flat <- as.numeric(tail(brent_flat_new_ts, n = 1))
forecasted_values_brent_flat_combined <-
  c(last_value_brent_flat, forecasted_values_arima_brent_f_train$mean)
cumulative_forecasted_values_brent_flat <-
  cumsum(forecasted_values_brent_flat_combined)
forecasted_values_arima_brent_f_test <-
  ts(cumulative_forecasted_values_brent_flat, start = test_start,
     frequency = 12)

# Calculate MSE and MAE for Flat by ARIMA of Brent Area
mse_brent_flat_arima <- mean((window(brent_flat_ts, start=test_start) -
                                forecasted_values_arima_brent_f_test)^2)
mae_brent_flat_arima <- mean(abs(window(brent_flat_ts, start=test_start) -
                                   forecasted_values_arima_brent_f_test))

# By ETS model for Flat of Brent Area
brent_flat_train_ets <- window(brent_flat_ts, end = train_end)
brent_flat_test_ets <- window(brent_flat_ts, start = test_start)

# Fit ETS models to the training data for Flat of Brent Area
fit_ets_brent_f_train <- ets(brent_flat_train_ets)

# Forecast the test period for Flat of Brent Area
forecasted_values_ets_brent_f_test <- forecast(fit_ets_brent_f_train, h = 9)

# Calculate MSE and MAE by ETS for Flat of Brent Area
mse_brent_flat_ets <- mean((brent_flat_test_ets -
                            forecasted_values_ets_brent_f_test$mean)^2)
mae_brent_flat_ets <- mean(abs(brent_flat_test_ets -
                               forecasted_values_ets_brent_f_test$mean))

# By STL model for Flat of Brent Area
brent_flat_train_stl <- window(brent_flat_ts, end = train_end)
brent_flat_test_stl <- window(brent_flat_ts, start = test_start)
fit_stl_brent_f_train <- stl(brent_flat_train_stl, s.window = "periodic")
forecasted_values_stl_brent_f_test <- forecast(fit_stl_brent_f_train,
                                               method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Flat of Brent Area
mse_brent_flat_stl <- mean((brent_flat_test_stl -
                            forecasted_values_stl_brent_f_test$mean)^2)
mae_brent_flat_stl <- mean(abs(brent_flat_test_stl -
                               forecasted_values_stl_brent_f_test$mean))

# Print MSE and MAE of Flat in Brent Area
print(paste("Brent Flat MSE for ARIMA:", mse_brent_flat_arima))
```

```
## [1] "Brent Flat MSE for ARIMA: 30884602.9555907"
```

```r
print(paste("Brent Flat MAE for ARIMA:", mae_brent_flat_arima))
```

```
## [1] "Brent Flat MAE for ARIMA: 4673.23986867098"
```

```r
print(paste("Brent Flat MSE for ETS:", mse_brent_flat_ets))
```

```
## [1] "Brent Flat MSE for ETS: 39890845.0825861"
```

```r
print(paste("Brent Flat MAE for ETS:", mae_brent_flat_ets))
```

```
## [1] "Brent Flat MAE for ETS: 5810.26480880922"
```

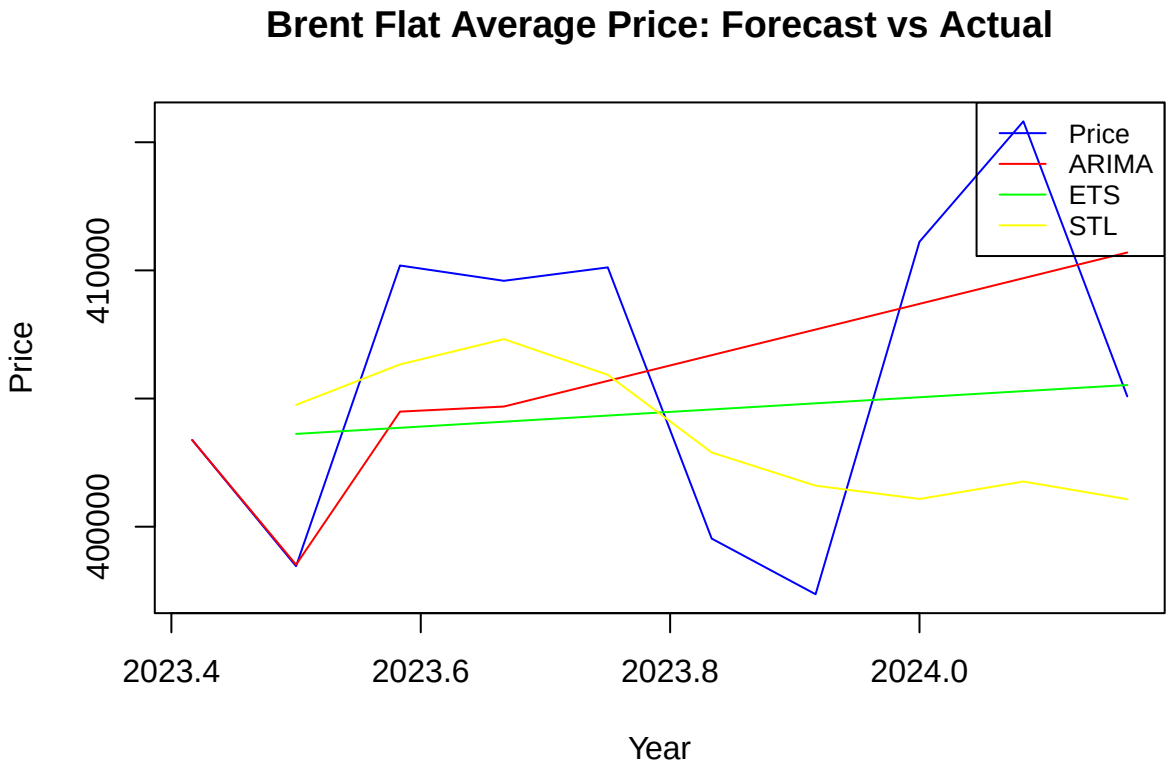
```r
print(paste("Brent Flat MSE for STL:", mse_brent_flat_stl))
```

```
## [1] "Brent Flat MSE for STL: 46753804.8973573"
```

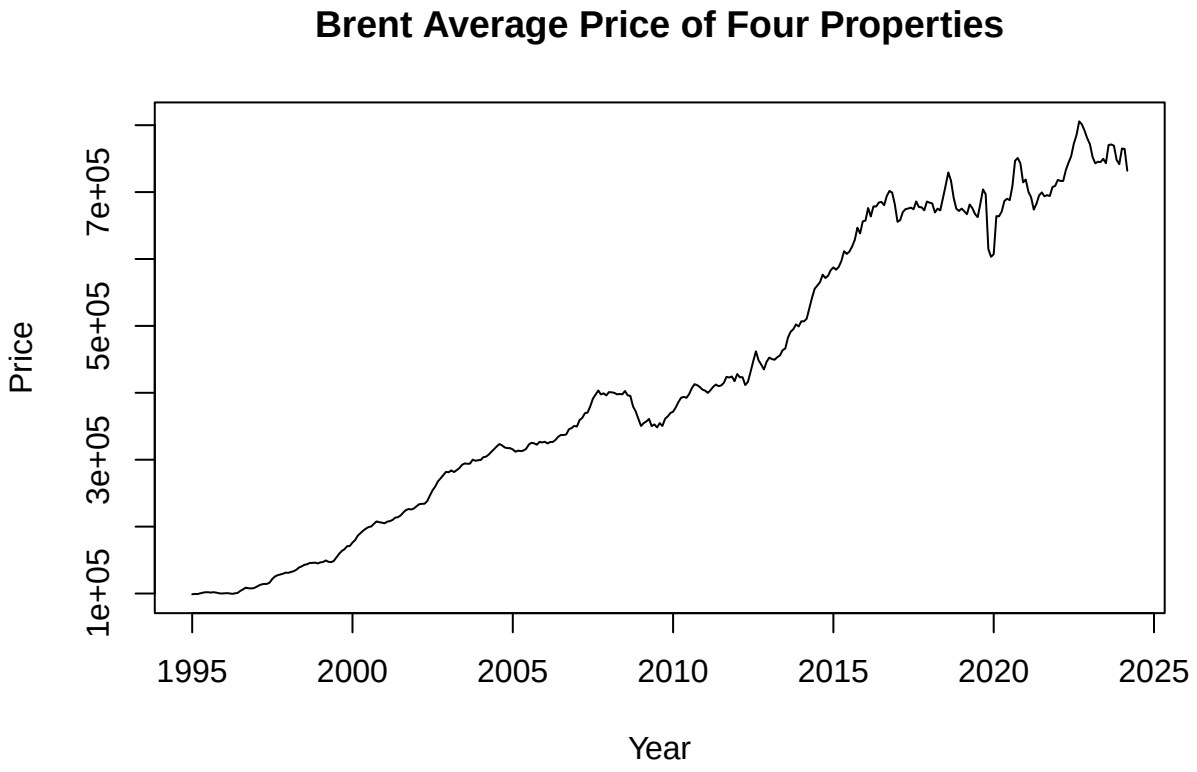
```r
print(paste("Brent Flat MAE for STL:", mae_brent_flat_stl))
```

```
## [1] "Brent Flat MAE for STL: 5812.86835251555"
```

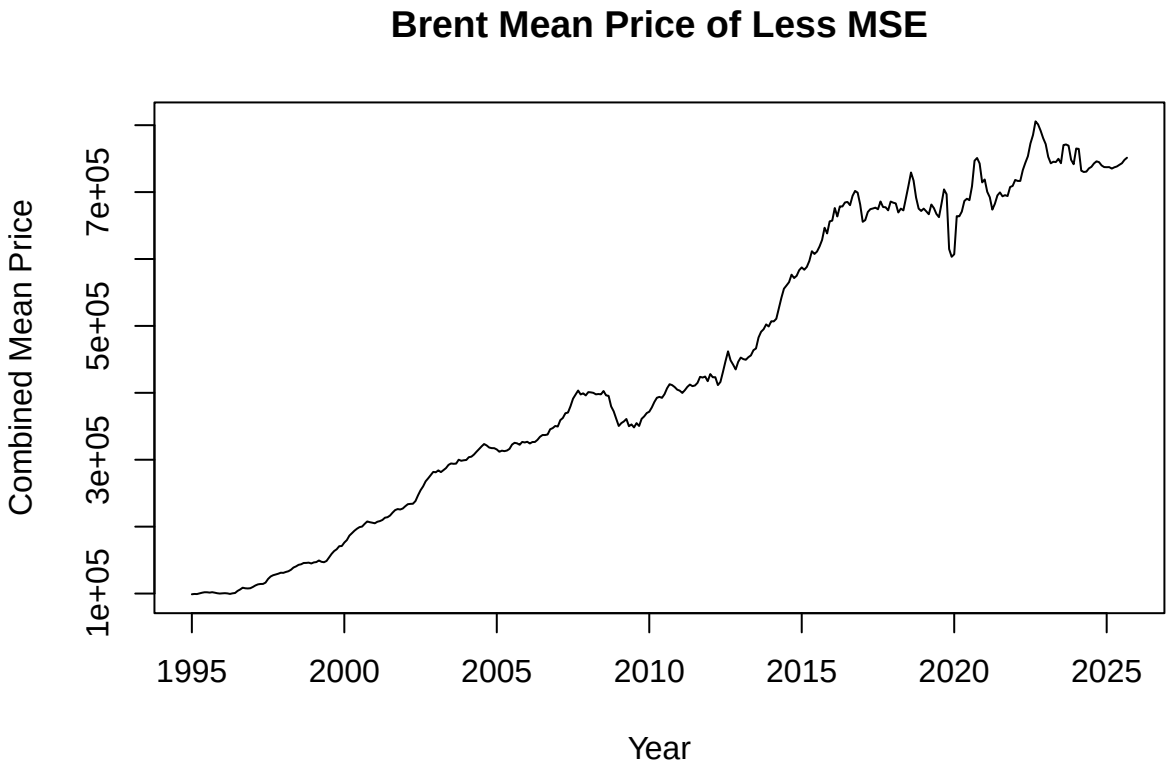
```r
# Plot the combined time series with forecast for Flat of Brent Area
plot(window(brent_flat_ts, start = train_end), type = "l", col = "blue",
     main = "Brent Flat Average Price: Forecast vs Actual", ylab = "Price",
     xlab = "Year", ylim = range(c(window(brent_flat_ts, start = train_end),
                                   forecasted_values_arima_brent_f_test,
                                   forecasted_values_ets_brent_f_test$mean,
                                   forecasted_values_stl_brent_f_test$mean)))
lines(forecasted_values_arima_brent_f_test, col = "red")
lines(forecasted_values_ets_brent_f_test$mean, col = "green")
lines(forecasted_values_stl_brent_f_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```



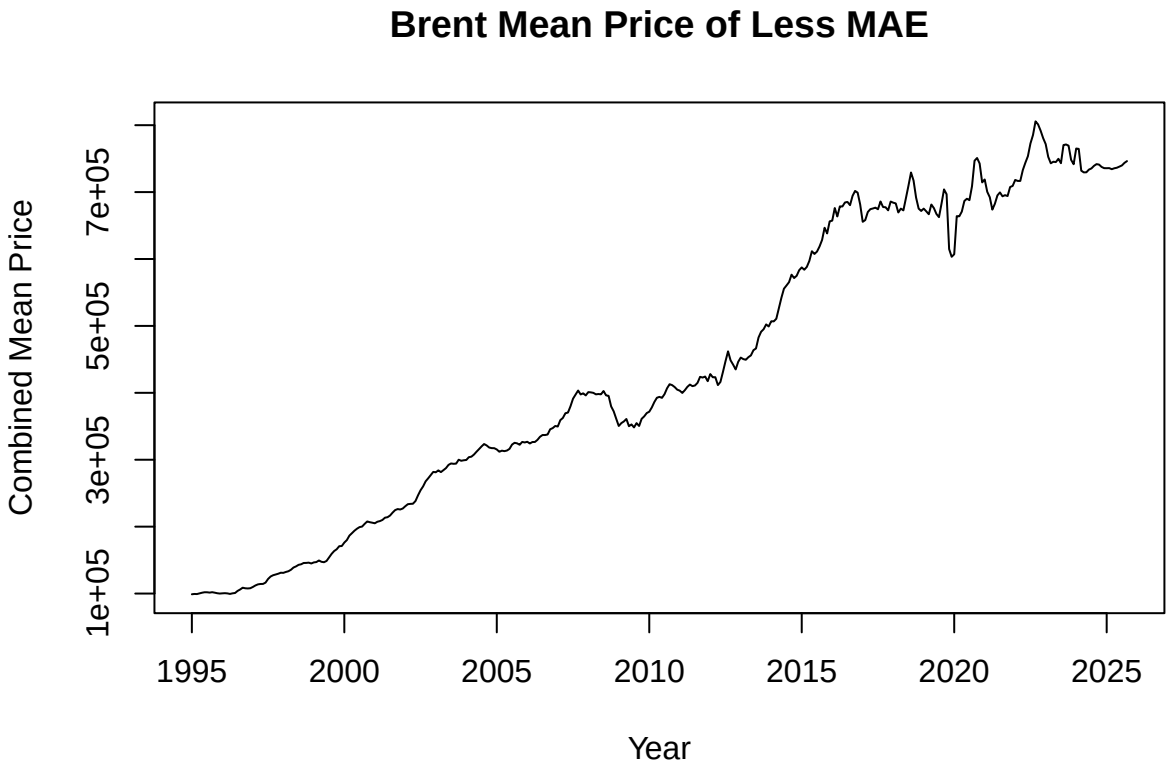

```
# Calculate the mean price of four property type of Brent Area
combined_mean_price_brent <- (brent_detached_price + brent_semi_detached_price
                              + brent_terraced_price + brent_flat_price)/4
combined_mean_price_brent_ts <- ts(combined_mean_price_brent,
                                   start = c(1995, 1), frequency = 12)
plot(combined_mean_price_brent_ts,
     main = "Brent Average Price of Four Properties",
     ylab = "Price", xlab = "Year")
```




```
# If we look for less MSE in Brent Area,
# STL model is the best for Detached
# STL model is the best for Semi-Detached
# STL model is the best for Terraced
# Arima model is the best for Flat

# Calculate the combined mean price of Brent Area with less MSE
combined_mean_price_brent_less_MSE <- (combined_brent_detached_ts_stl +
                                       combined_brent_semi_detached_ts_stl +
                                       combined_brent_terraced_ts_stl +
                                       combined_brent_flat_ts_Arima) / 4
combined_mean_price_brent_less_MSE_ts <- ts(combined_mean_price_brent_less_MSE,
                                            start = c(1995, 1), frequency = 12)
plot(combined_mean_price_brent_less_MSE_ts,
     main = "Brent Mean Price of Less MSE",
     ylab = "Combined Mean Price", xlab = "Year")
```




```r
# If we look for less MAE in Brent Area,
# STL model is the best for Detached
# STL model is the best for Semi-Detached
# ETS model is the best for Terraced
# Arima model is the best for Flat

# Calculate combined mean price for property types of Brent Area with less MAE
combined_mean_price_brent_less_MAE <- (combined_brent_detached_ts_stl +
                                       combined_brent_semi_detached_ts_stl +
                                       combined_brent_terraced_ts_ets +
                                       combined_brent_flat_ts_Arima) / 4
combined_mean_price_brent_less_MAE_ts <- ts(combined_mean_price_brent_less_MAE,
                                            start = c(1995, 1), frequency = 12)
plot(combined_mean_price_brent_less_MAE_ts,
     main = "Brent Mean Price of Less MAE",
     ylab = "Combined Mean Price", xlab = "Year")
```




```
# Compare two models for which one is better in Brent Area

combined_mean_price_brent_less_MSE_test <-
  (forecasted_values_stl_brent_d_test$mean +
     forecasted_values_stl_brent_sd_test$mean +
     forecasted_values_stl_brent_t_test$mean +
     forecasted_values_arima_brent_f_test) / 4
combined_mean_price_brent_less_MSE_test_ts <-
  ts(combined_mean_price_brent_less_MSE_test,
     start = test_start, frequency = 12)

combined_mean_price_brent_less_MAE_test <-
  (forecasted_values_stl_brent_d_test$mean +
     forecasted_values_stl_brent_sd_test$mean +
     forecasted_values_ets_brent_t_test$mean +
     forecasted_values_arima_brent_f_test) / 4
combined_mean_price_brent_less_MAE_test_ts <-
  ts(combined_mean_price_brent_less_MAE_test,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for combined mean prices in Brent Area
combined_mean_price_brent_test <- window(combined_mean_price_brent_ts,
                                         start = test_start)
mse_combined_brent_less_MSE <-
  mean((combined_mean_price_brent_test -
          combined_mean_price_brent_less_MSE_test_ts)^2)
mae_combined_brent_less_MSE <- mean(abs(combined_mean_price_brent_test -
                                    combined_mean_price_brent_less_MSE_test_ts))
```

```r
mse_combined_brent_less_MAE <- mean((combined_mean_price_brent_test -
                                     combined_mean_price_brent_less_MAE_test_ts)^2)
mae_combined_brent_less_MAE <- mean(abs(combined_mean_price_brent_test -
                                      combined_mean_price_brent_less_MAE_test_ts))

# Print MSE and MAE for combined mean prices in Brent Area
print(paste("Brent Combined Mean Price MSE for Less MSE Model:",
            mse_combined_brent_less_MSE))
```

```
## [1] "Brent Combined Mean Price MSE for Less MSE Model: 107820761.994977"
```

```r
print(paste("Brent Combined Mean Price MAE for Less MSE Model:",
            mae_combined_brent_less_MSE))
```

```
## [1] "Brent Combined Mean Price MAE for Less MSE Model: 9737.95323887907"
```

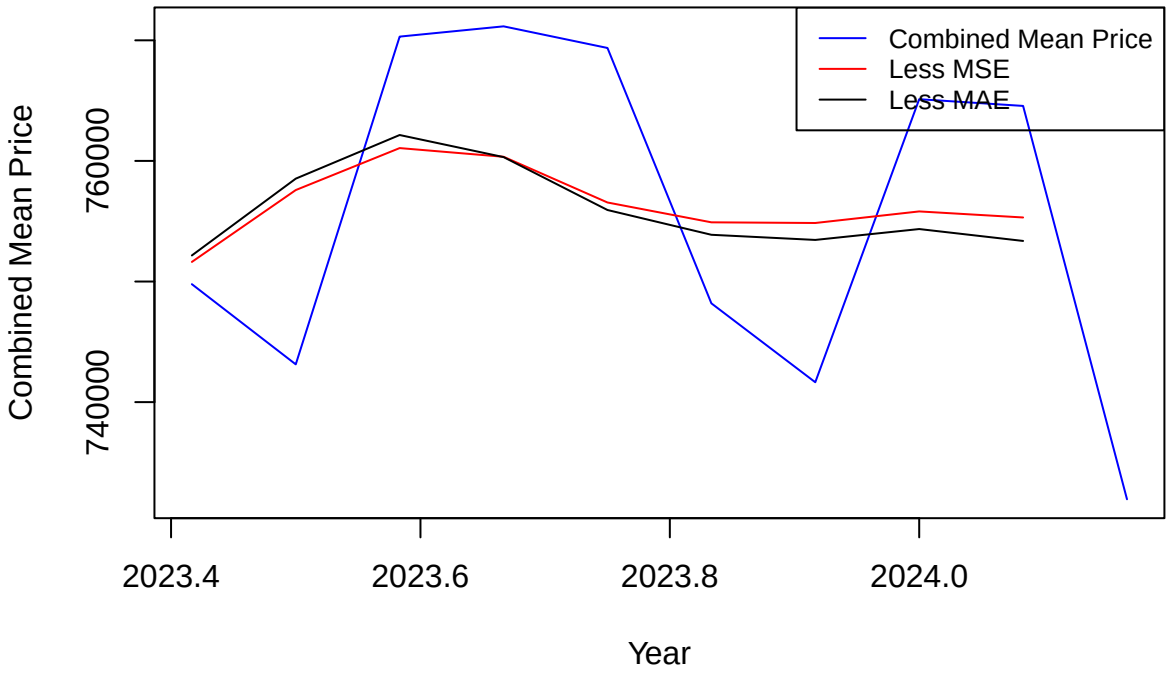
```r
print(paste("Brent Combined Mean Price MSE for Less MAE Model:",
            mse_combined_brent_less_MAE))
```

```
## [1] "Brent Combined Mean Price MSE for Less MAE Model: 113355278.670304"
```

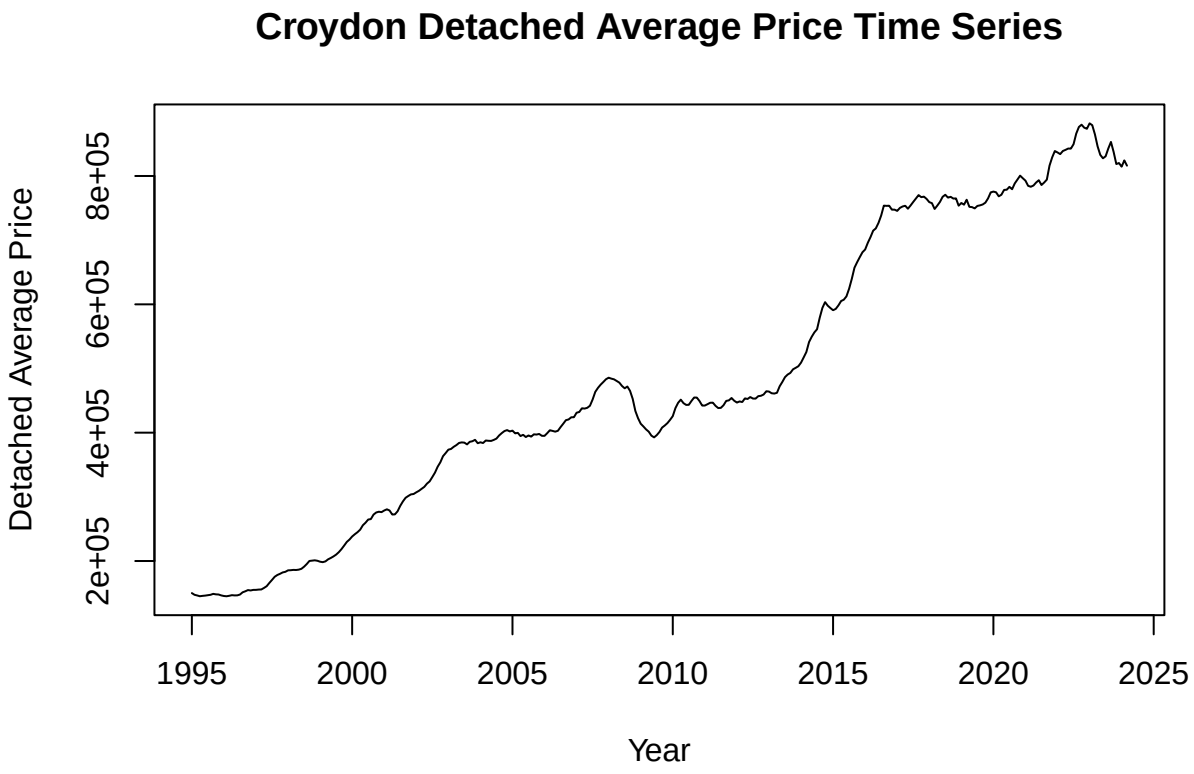
```r
print(paste("Brent Combined Mean Price MAE for Less MAE Model:",
            mae_combined_brent_less_MAE))
```

```
## [1] "Brent Combined Mean Price MAE for Less MAE Model: 9962.44868555034"
```

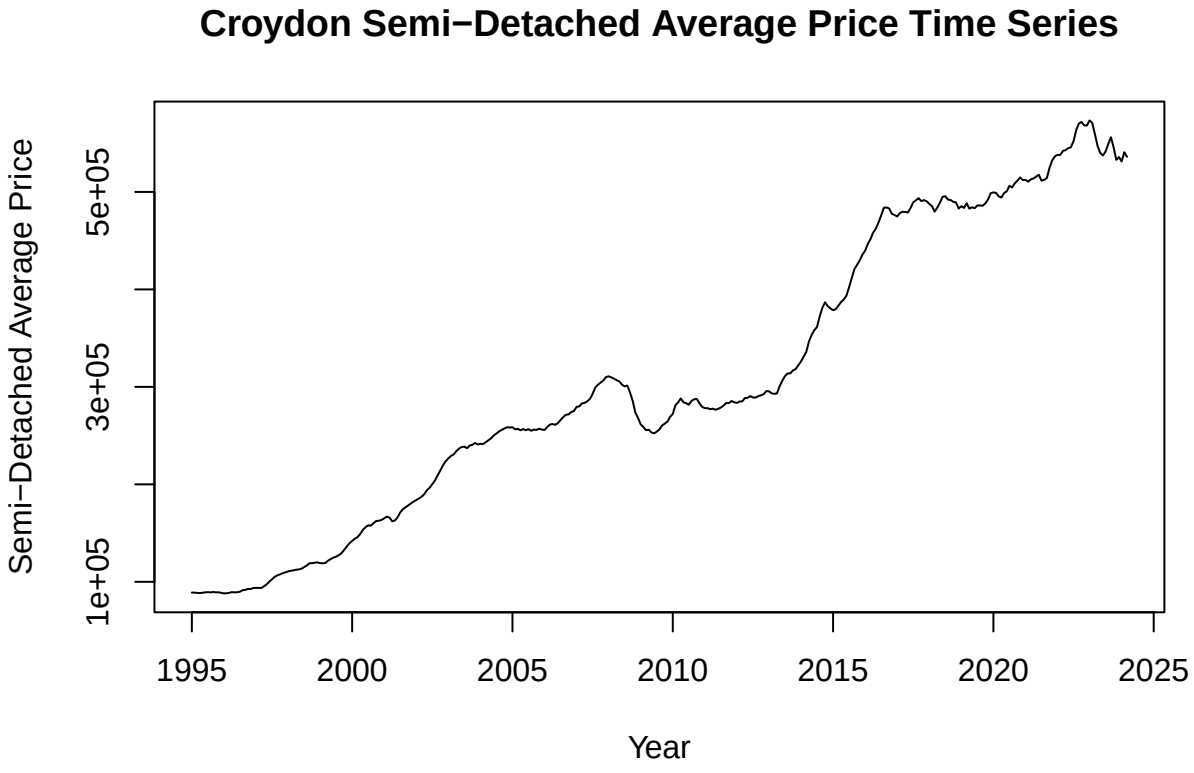
```r
# Plot all combined series together for Brent Area
plot(combined_mean_price_brent_test, type = "l", col = "blue",
     main = "Brent Combined Mean Prices Comparison",
     ylab = "Combined Mean Price", xlab = "Year",
     ylim = range(c(combined_mean_price_brent_test,
                    combined_mean_price_brent_less_MSE_test_ts,
                    combined_mean_price_brent_less_MAE_test_ts)))
lines(combined_mean_price_brent_less_MSE_test_ts, col = "red")
lines(combined_mean_price_brent_less_MAE_test_ts, col = "black")

legend("topright", legend = c("Combined Mean Price", "Less MSE", "Less MAE"),
       col = c("blue", "red", "black"), lty = 1, cex = 0.8)
```



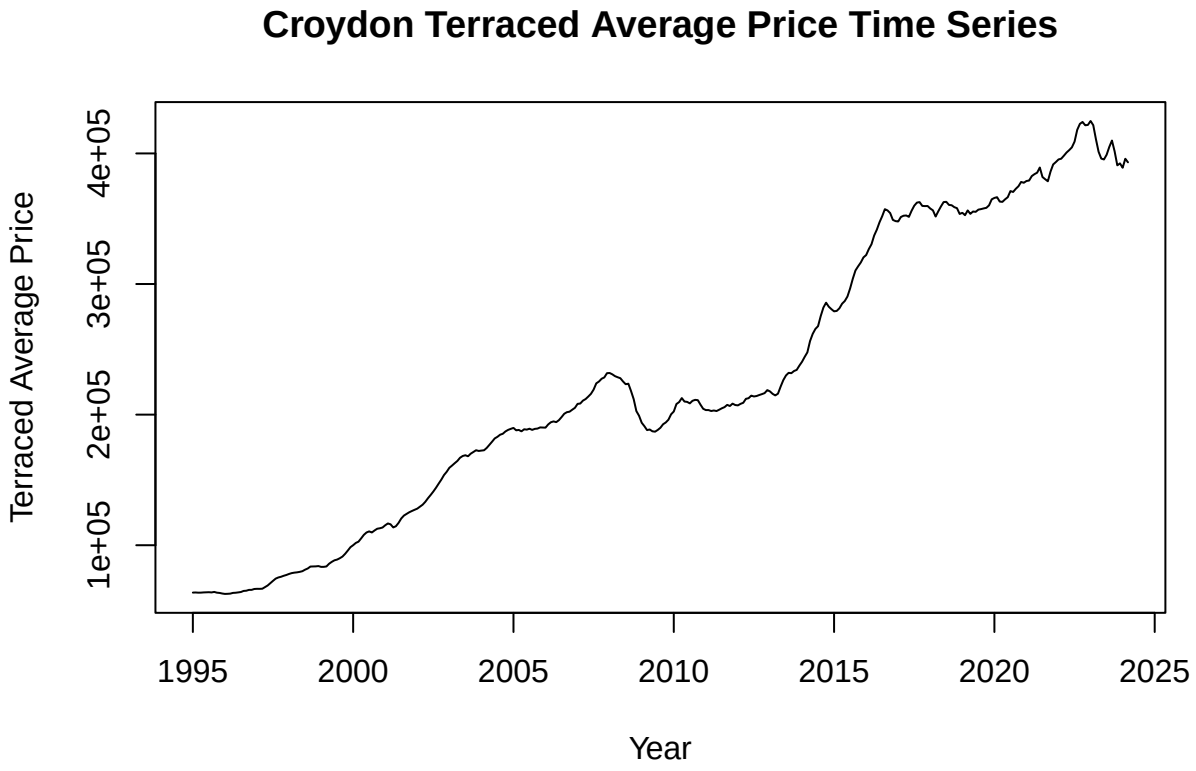

```
### Croydon Area
# Load the data of Croydon Area
croydon_data <- read.csv("~/Desktop/Updated_Croydon_df.csv")

# Convert the Date column to Date type of Croydon Area
croydon_data$Date <- as.Date(croydon_data$Date, format="%Y-%m-%d")

# Extract the Average Price for each property type and Date columns in Croydon
croydon_detached_price <- croydon_data$Detached_Average_Price
croydon_semi_detached_price <- croydon_data$Semi_Detached_Average_Price
croydon_terraced_price <- croydon_data$Terraced_Average_Price
croydon_flat_price <- croydon_data$Flat_Average_Price
croydon_dates <- croydon_data$Date

# Create time series object for each property type of Croydon Area
croydon_detached_ts <- ts(croydon_detached_price,
                          start = c(1995, 1), frequency = 12)
croydon_semi_detached_ts <- ts(croydon_semi_detached_price,
                               start = c(1995, 1), frequency = 12)
croydon_terraced_ts <- ts(croydon_terraced_price,
                          start = c(1995, 1), frequency = 12)
croydon_flat_ts <- ts(croydon_flat_price, start = c(1995, 1), frequency = 12)

# Plot the time series of Croydon Area
plot(croydon_detached_ts, main = "Croydon Detached Average Price Time Series",
     ylab = "Detached Average Price", xlab = "Year")
```



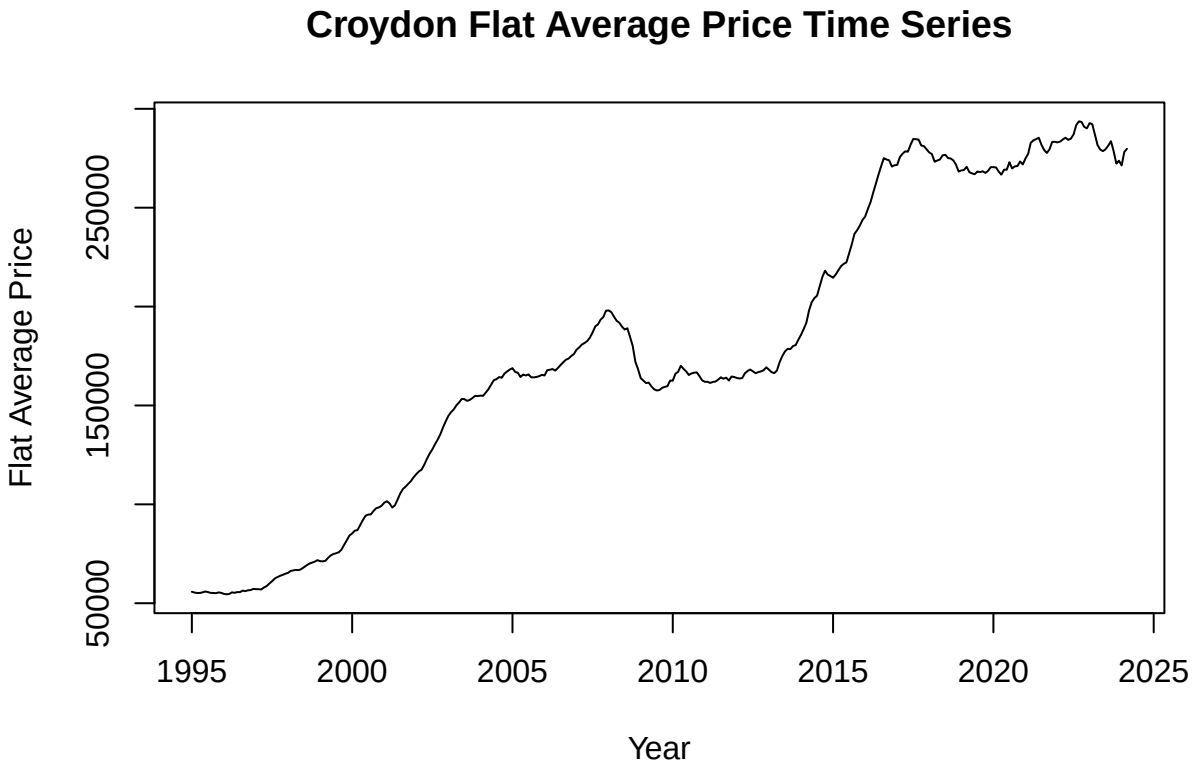

```
plot(croydon_semi_detached_ts,
     main = "Croydon Semi-Detached Average Price Time Series",
     ylab = "Semi-Detached Average Price", xlab = "Year")
```



```
plot(croydon_terraced_ts, main = "Croydon Terraced Average Price Time Series",
     ylab = "Terraced Average Price", xlab = "Year")
```




```
plot(croydon_flat_ts, main = "Croydon Flat Average Price Time Series",
     ylab = "Flat Average Price", xlab = "Year")
```




```
# Calculate the p-value for the original series
# By using ADF test for each property type of Croydon Area
adf_test_croydon_d <- adf.test(croydon_detached_ts)
adf_test_croydon_sd <- adf.test(croydon_semi_detached_ts)
adf_test_croydon_t <- adf.test(croydon_terraced_ts)
adf_test_croydon_f <- adf.test(croydon_flat_ts)
p_value_croydon_d <- adf_test_croydon_d$p.value
p_value_croydon_sd <- adf_test_croydon_sd$p.value
p_value_croydon_t <- adf_test_croydon_t$p.value
p_value_croydon_f <- adf_test_croydon_f$p.value
print(paste("Croydon p-value:", p_value_croydon_d, p_value_croydon_sd,
            p_value_croydon_t, p_value_croydon_f))
```

```
## [1] "Croydon p-value: 0.482159416344303 0.429833692588994 0.420094571650799 0.423259870045663"
```

```
# Perform first-order differencing of Croydon Area
croydon_detached_ts_diff <- diff(croydon_detached_ts)
croydon_semi_detached_ts_diff <- diff(croydon_semi_detached_ts)
croydon_terraced_ts_diff <- diff(croydon_terraced_ts)
croydon_flat_ts_diff <- diff(croydon_flat_ts)

# Perform ADF test on differenced series of Croydon Area
adf_test_diff_croydon_d <- adf.test(croydon_detached_ts_diff)
```

```
## Warning in adf.test(croydon_detached_ts_diff): p-value smaller than printed
## p-value
```

```r
adf_test_diff_croydon_sd <- adf.test(croydon_semi_detached_ts_diff)
```

```
## Warning in adf.test(croydon_semi_detached_ts_diff): p-value smaller than
## printed p-value
```

```r
adf_test_diff_croydon_t <- adf.test(croydon_terraced_ts_diff)
```

```
## Warning in adf.test(croydon_terraced_ts_diff): p-value smaller than printed
## p-value
```

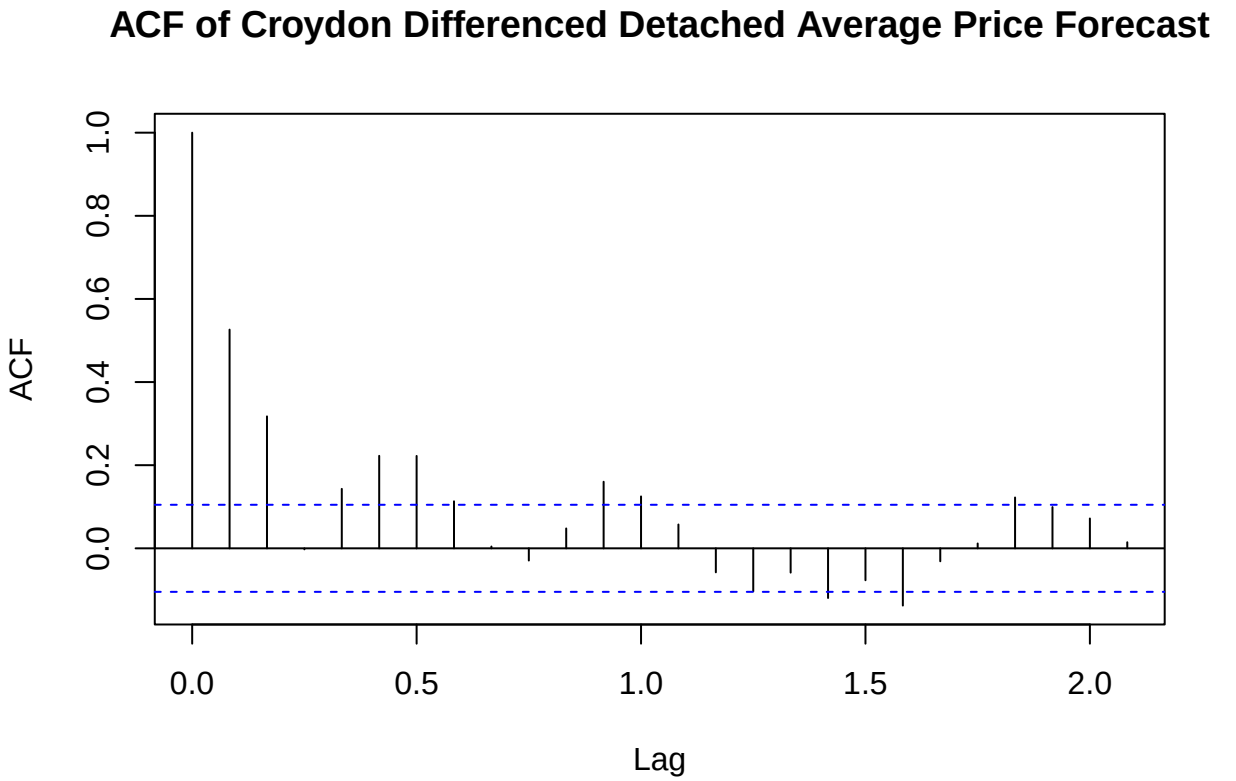
```r
adf_test_diff_croydon_f <- adf.test(croydon_flat_ts_diff)
```

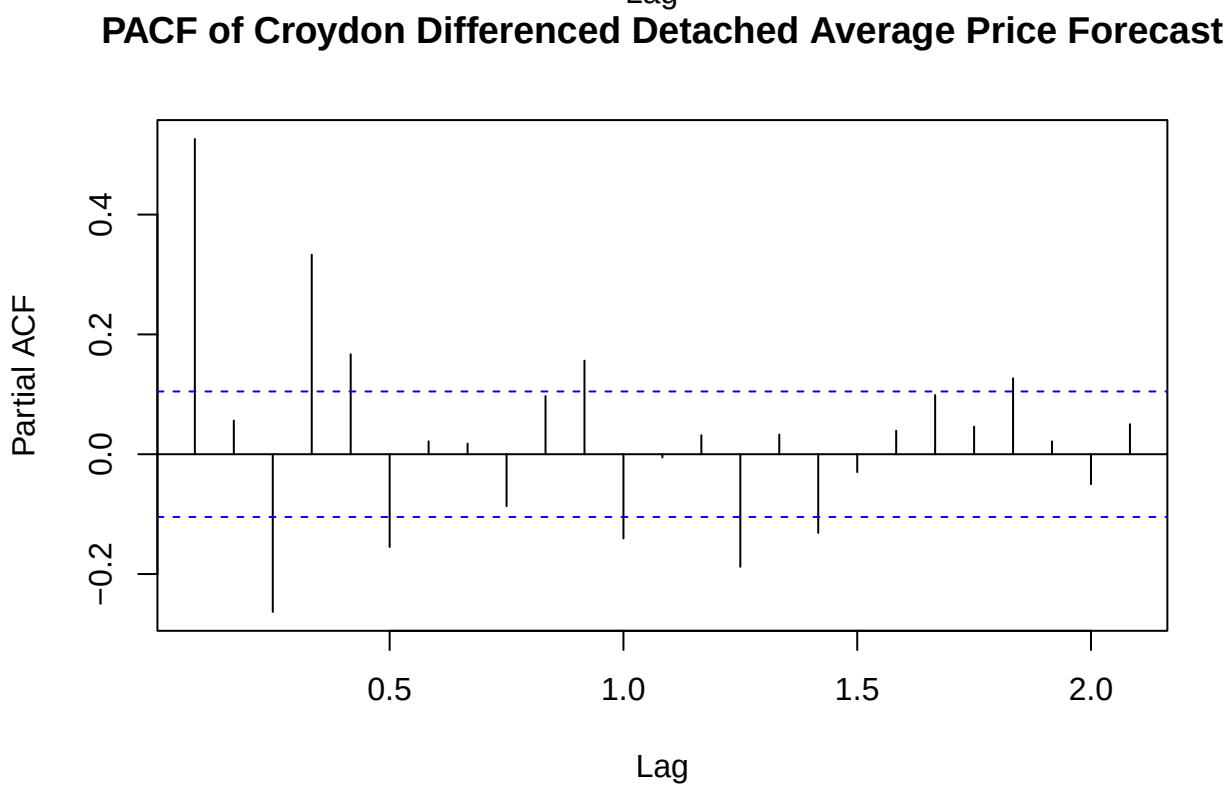
```
## Warning in adf.test(croydon_flat_ts_diff): p-value smaller than printed p-value
```

```r
differenced_p_value_croydon_d <- adf_test_diff_croydon_d$p.value
differenced_p_value_croydon_sd <- adf_test_diff_croydon_sd$p.value
differenced_p_value_croydon_t <- adf_test_diff_croydon_t$p.value
differenced_p_value_croydon_f <- adf_test_diff_croydon_f$p.value
print(paste("Croydon difference1_p-value:", differenced_p_value_croydon_d,
            differenced_p_value_croydon_sd, differenced_p_value_croydon_t,
            differenced_p_value_croydon_f))
```

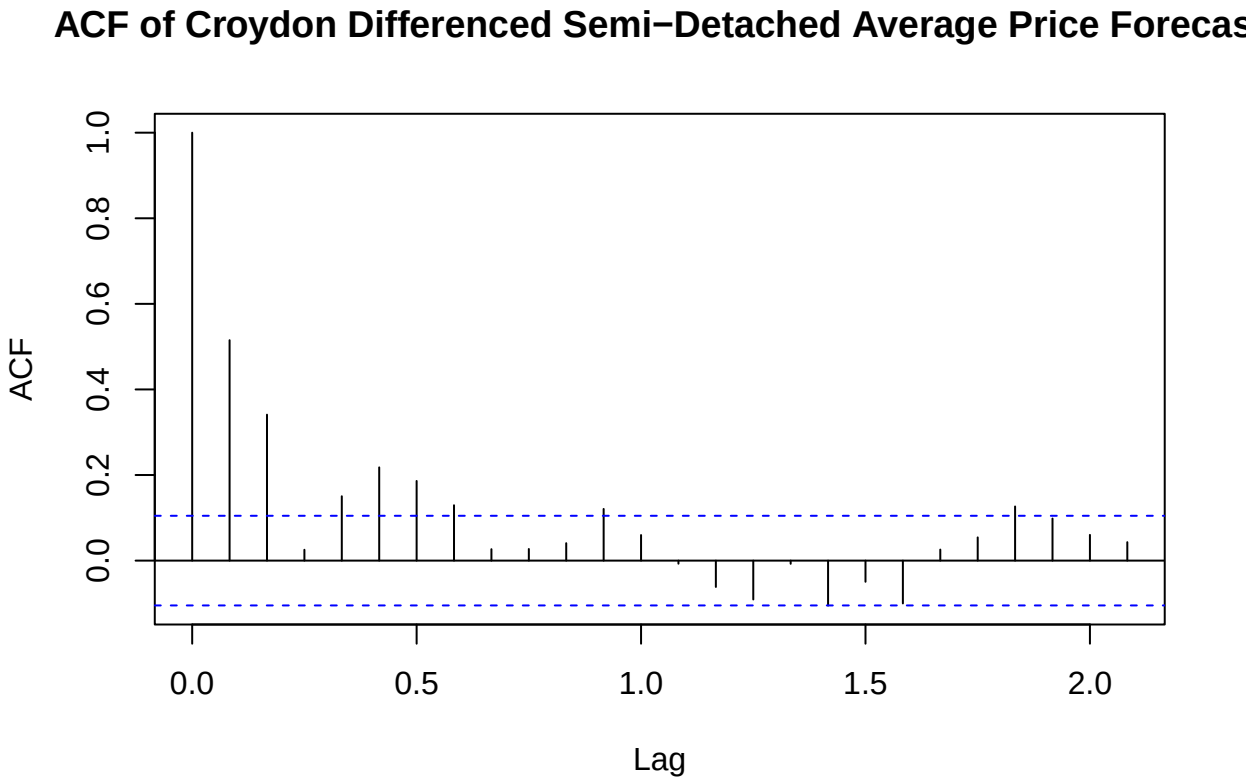
```
## [1] "Croydon difference1_p-value: 0.01 0.01 0.01 0.01"
```

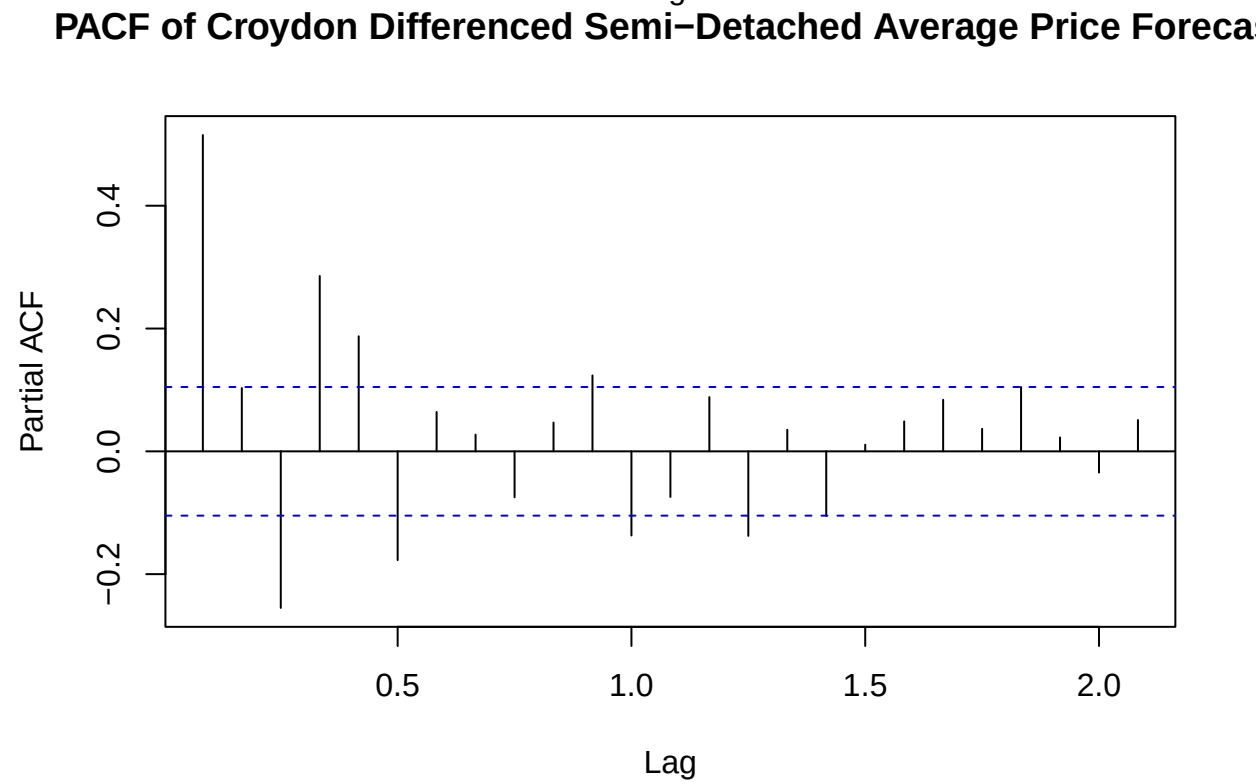
```r
# Define function to plot ACF and PACF of Croydon Area
plot_acf_pacf <- function(ts_diff, title) {
  acf(ts_diff, main = paste("ACF of", title))
  pacf(ts_diff, main = paste("PACF of", title))
}

# Plot ACF and PACF for differenced series for each property type in Croydon
plot_acf_pacf(croydon_detached_ts_diff,
              "Croydon Differenced Detached Average Price Forecast")
```



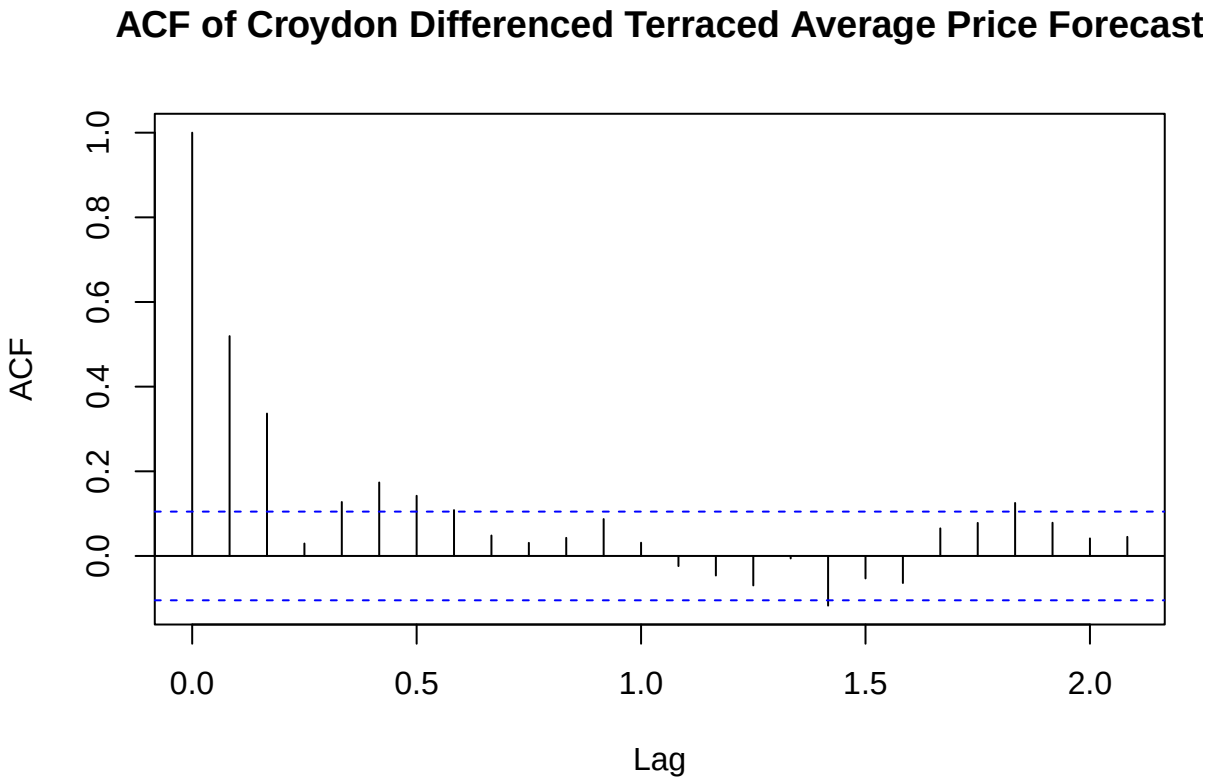


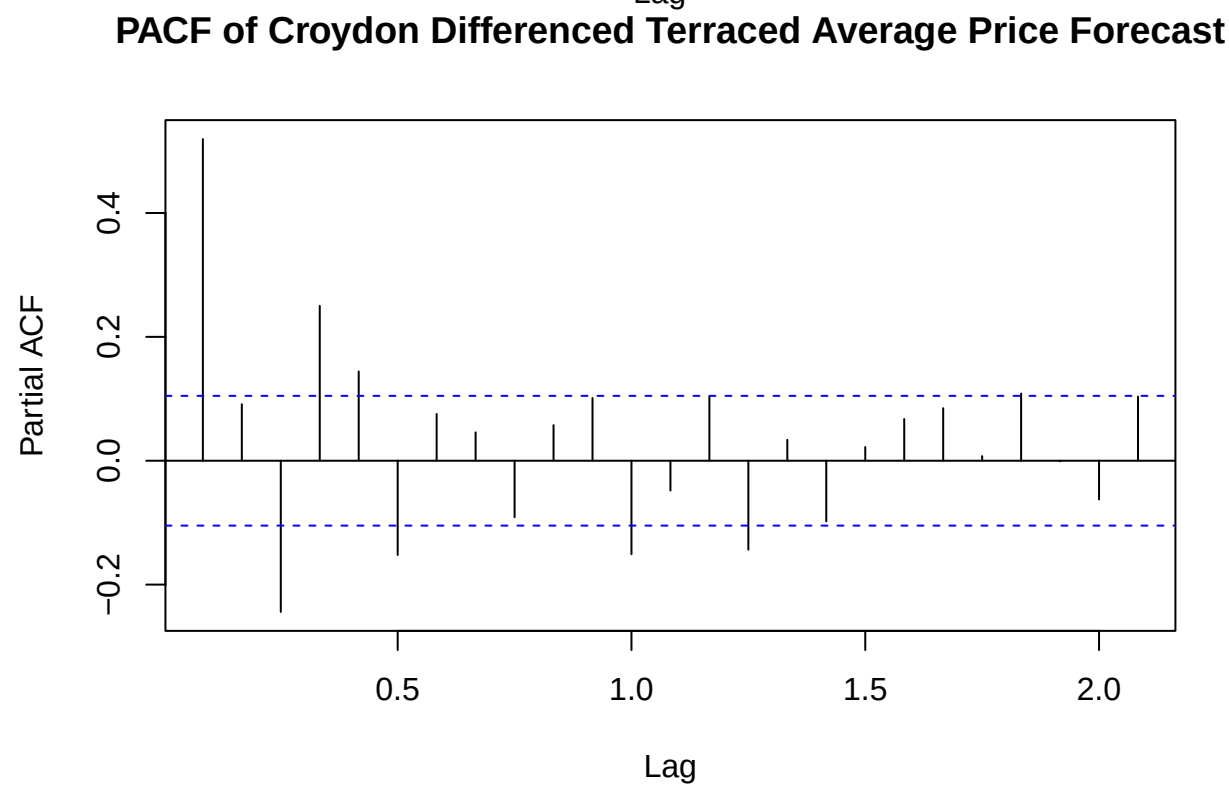

```
plot_acf_pacf(croydon_semi_detached_ts_diff,
              "Croydon Differenced Semi-Detached Average Price Forecast")
```



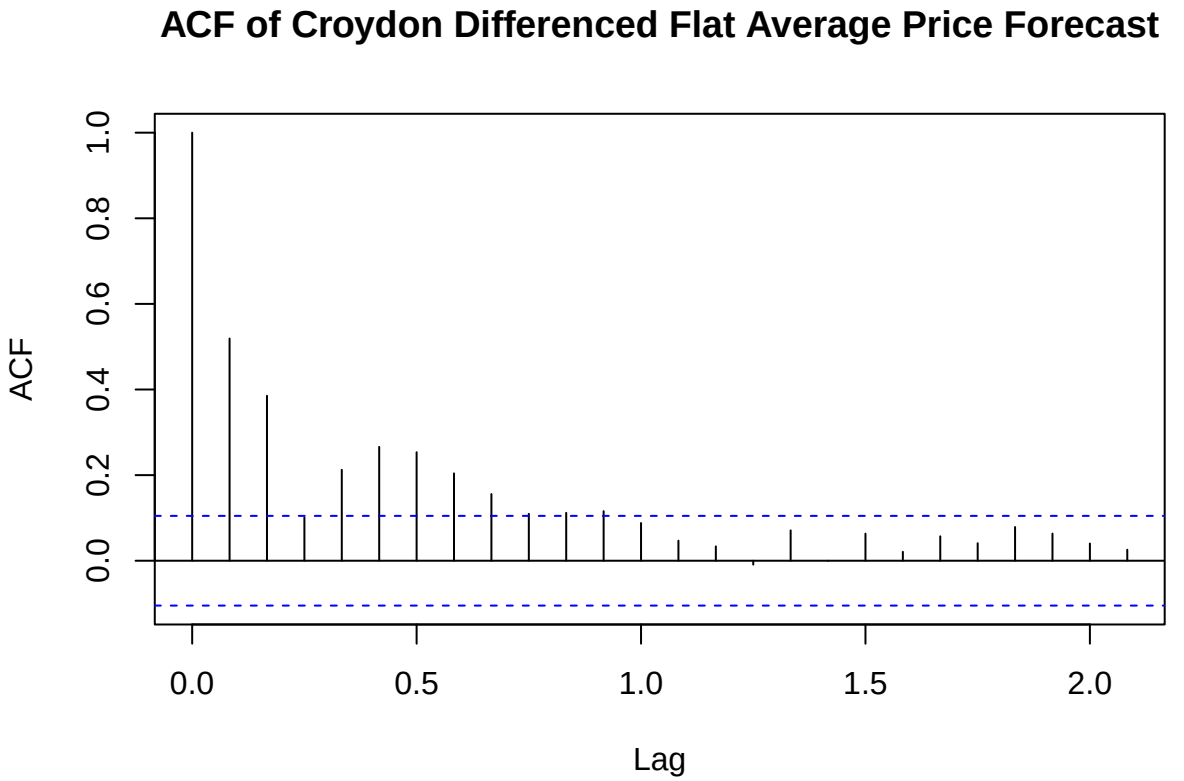


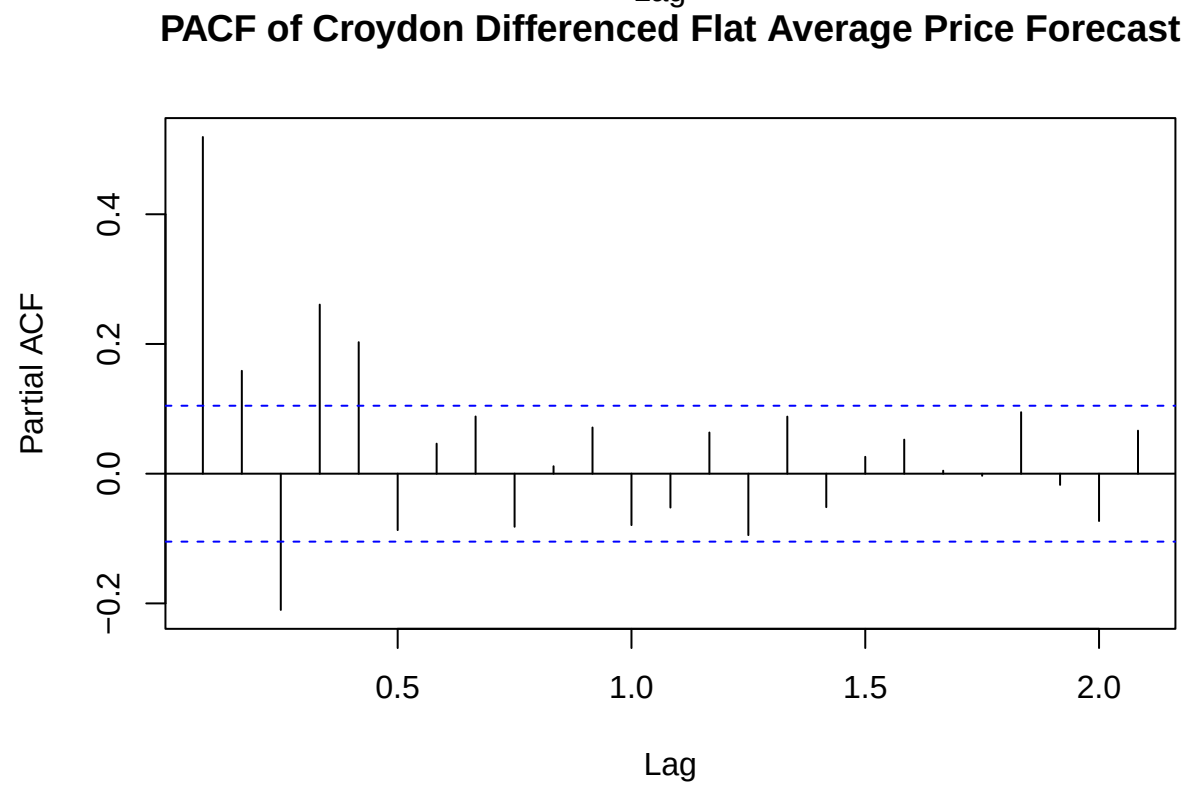

```
plot_acf_pacf(croydon_terraced_ts_diff,
              "Croydon Differenced Terraced Average Price Forecast")
```






```
plot_acf_pacf(croydon_flat_ts_diff,
              "Croydon Differenced Flat Average Price Forecast")
```






```
# Fit ARIMA models for Croydon Area
fit_arima_croydon_d <- auto.arima(croydon_detached_ts_diff, stepwise = FALSE,
                          approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 7061.172
##  ARIMA(0,0,0)            with non-zero mean : 7023.322
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 7047.076
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 7019.787
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 7039.692
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 7019.271
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 7041.604
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 7018.869
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : 7019.797
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : 7021.3
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 7036.096
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 7019.338
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 6975.332
##  ARIMA(0,0,1)            with non-zero mean : 6951.206
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 6970.373
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 6950.971
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 6964.57
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 6950.24
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 6968.086
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 6950.558
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : 6951.529
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : 6965.395
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : 6952.305
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 6963.085
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 6950.464
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 6826.691
##  ARIMA(0,0,2)            with non-zero mean : 6819.341
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 6828.159
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 6819.237
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 6827.977
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 6820.806
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 6828.052
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 6819.051
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 6827.8
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 6820.585
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)            with zero mean     : 6808.86
```

```
##  ARIMA(0,0,3)            with non-zero mean : 6793.625
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 6810.638
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 6795.63
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : 6808.371
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 6796.35
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : 6810.559
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 6795.62
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : 6807.755
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 6796.284
##  ARIMA(0,0,4)            with zero mean     : 6806.799
##  ARIMA(0,0,4)            with non-zero mean : 6793.595
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 6808.382
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 6795.675
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : 6808.263
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 6795.675
##  ARIMA(0,0,5)            with zero mean     : 6806.728
##  ARIMA(0,0,5)            with non-zero mean : 6794.595
##  ARIMA(1,0,0)            with zero mean     : 6919.41
##  ARIMA(1,0,0)            with non-zero mean : 6910.635
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 6918.242
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 6910.778
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 6917.508
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 6911.535
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 6917.648
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 6910.555
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : 6912.235
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : 6919.56
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 6917.556
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 6911.772
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : 6911.766
##  ARIMA(1,0,1)            with zero mean     : 6920.202
##  ARIMA(1,0,1)            with non-zero mean : 6912.265
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 6918.24
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 6911.973
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 6918.122
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 6912.926
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 6917.597
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 6911.715
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : 6913.484
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 6918.22
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 6913.159
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : 6806.642
```

```
##  ARIMA(1,0,2)            with non-zero mean : 6792.809
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : 6808.368
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 6794.829
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : 6806.73
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 6795.898
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : 6808.283
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 6794.823
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : 6806.238
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 6795.838
##  ARIMA(1,0,3)            with zero mean     : 6808.694
##  ARIMA(1,0,3)            with non-zero mean : 6794.273
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : 6810.41
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 6796.333
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : 6810.315
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 6796.33
##  ARIMA(1,0,4)            with zero mean     : 6793.361
##  ARIMA(1,0,4)            with non-zero mean : 6789.841
##  ARIMA(2,0,0)            with zero mean     : 6919.346
##  ARIMA(2,0,0)            with non-zero mean : 6911.873
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 6916.769
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 6911.131
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 6917.012
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 6912.255
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 6916.108
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 6910.852
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : 6917.401
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : 6912.698
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 6917.133
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 6912.48
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 6903.059
##  ARIMA(2,0,1)            with non-zero mean : 6896.354
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 6901.484
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 6896.139
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 6902.926
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 6898.077
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 6901.182
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 6896.065
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : 6902.987
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : 6898.135
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 6902.946
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 6898.126
##  ARIMA(2,0,2)            with zero mean     : 6808.688
##  ARIMA(2,0,2)            with non-zero mean : 6793.962
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : 6810.382
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : 6796.036
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : 6810.277
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : 6796.035
##  ARIMA(2,0,3)            with zero mean     : 6797.211
```

```
##  ARIMA(2,0,3)            with non-zero mean : 6795.03
##  ARIMA(3,0,0)            with zero mean     : 6902.816
##  ARIMA(3,0,0)            with non-zero mean : 6888.739
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 6900.451
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 6889.254
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 6901.068
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 6891.096
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 6899.908
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 6889.187
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : 6891.246
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 6901.12
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 6891.203
##  ARIMA(3,0,1)            with zero mean     : 6887.173
##  ARIMA(3,0,1)            with non-zero mean : 6874.016
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 6887.539
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 6875.908
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 6887.327
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 6875.9
##  ARIMA(3,0,2)            with zero mean     : 6805.979
##  ARIMA(3,0,2)            with non-zero mean : 6794.266
##  ARIMA(4,0,0)            with zero mean     : 6849.408
##  ARIMA(4,0,0)            with non-zero mean : 6844.313
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 6851.447
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 6846.18
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 6851.44
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 6846.148
##  ARIMA(4,0,1)            with zero mean     : 6844.674
##  ARIMA(4,0,1)            with non-zero mean : 6841.578
##  ARIMA(5,0,0)            with zero mean     : 6840.077
##  ARIMA(5,0,0)            with non-zero mean : 6837.409
##
##
##
##  Best model: ARIMA(1,0,4)            with non-zero mean
```

```
fit_arima_croydon_sd <- auto.arima(croydon_semi_detached_ts_diff,
                                   stepwise = FALSE,
                              approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 6740.57
##  ARIMA(0,0,0)            with non-zero mean : 6697.36
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 6732.55
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 6698.087
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 6723.276
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 6697.85
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 6728.743
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 6697.885
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : 6699.449
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : 6722.733
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 6720.349
```

```
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 6698.318
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 6658.426
##  ARIMA(0,0,1)            with non-zero mean : 6630.641
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 6655.841
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 6632.15
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 6651.326
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 6632.971
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 6654.466
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 6632.088
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : 6633.968
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : 6652.68
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 6650.492
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 6633.263
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : 6632.055
##  ARIMA(0,0,2)            with zero mean     : 6506.449
##  ARIMA(0,0,2)            with non-zero mean : 6497.795
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 6508.003
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 6497.547
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 6509.479
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 6499.44
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 6507.954
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 6497.603
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 6509.241
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 6499.673
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : 6498.216
##  ARIMA(0,0,3)            with zero mean     : 6487.525
##  ARIMA(0,0,3)            with non-zero mean : 6469.665
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 6489.423
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 6471.35
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : 6487.948
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 6473.05
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : 6489.384
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 6471.321
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 6472.9
##  ARIMA(0,0,4)            with zero mean     : 6482.361
##  ARIMA(0,0,4)            with non-zero mean : 6467.449
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 6484.177
```

```
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 6469.372
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : 6484.128
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 6469.362
##  ARIMA(0,0,5)            with zero mean     : 6482.757
##  ARIMA(0,0,5)            with non-zero mean : 6468.915
##  ARIMA(1,0,0)            with zero mean     : 6601.389
##  ARIMA(1,0,0)            with non-zero mean : 6590.641
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 6601.976
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 6592.286
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 6602.938
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 6594.24
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 6601.818
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 6592.275
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : 6603.627
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : 6594.331
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 6603.185
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 6594.308
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : 6602.14
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : 6592.524
##  ARIMA(1,0,1)            with zero mean     : 6599.955
##  ARIMA(1,0,1)            with non-zero mean : 6591.039
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 6600.223
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 6592.476
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 6601.67
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 6594.498
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 6600.093
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 6592.468
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : 6602.065
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : 6594.538
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 6601.902
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 6594.538
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : 6483.492
##  ARIMA(1,0,2)            with non-zero mean : 6466.676
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : 6485.315
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 6468.458
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 6470.258
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : 6485.262
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 6468.439
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 6470.138
##  ARIMA(1,0,3)            with zero mean     : 6485.395
##  ARIMA(1,0,3)            with non-zero mean : 6468.614
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : 6487.274
```

```
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 6470.432
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : 6487.23
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 6470.414
##  ARIMA(1,0,4)            with zero mean     : 6468.927
##  ARIMA(1,0,4)            with non-zero mean : 6464.268
##  ARIMA(2,0,0)            with zero mean     : 6597.791
##  ARIMA(2,0,0)            with non-zero mean : 6589.552
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 6597.759
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 6590.764
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 6599.377
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 6592.817
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 6597.641
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 6590.763
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : 6599.661
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : 6592.831
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 6599.589
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 6592.822
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : 6593.306
##  ARIMA(2,0,1)            with zero mean     : 6582.341
##  ARIMA(2,0,1)            with non-zero mean : 6574.379
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 6583.086
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 6576.031
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 6585.137
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 6577.881
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 6583.083
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 6576.056
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : 6585.151
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : 6578.088
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 6585.144
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 6577.78
##  ARIMA(2,0,2)            with zero mean     : 6485.265
##  ARIMA(2,0,2)            with non-zero mean : 6468.555
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : 6487.177
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : 6470.387
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : 6487.141
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : 6470.369
##  ARIMA(2,0,3)            with zero mean     : 6470.682
##  ARIMA(2,0,3)            with non-zero mean : 6468.065
##  ARIMA(3,0,0)            with zero mean     : 6581.938
##  ARIMA(3,0,0)            with non-zero mean : 6566.902
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 6581.538
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 6568.497
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 6582.897
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 6570.574
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 6581.361
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 6568.506
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : 6583.355
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : 6570.583
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 6583.208
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 6570.516
##  ARIMA(3,0,1)            with zero mean     : 6570.255
```

```
##  ARIMA(3,0,1)            with non-zero mean : 6556.121
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 6571.381
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 6558.198
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 6571.335
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 6558.198
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : 6468.973
##  ARIMA(4,0,0)            with zero mean     : 6538.531
##  ARIMA(4,0,0)            with non-zero mean : 6532.09
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 6540.423
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 6533.603
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 6540.413
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 6533.604
##  ARIMA(4,0,1)            with zero mean     : 6533.552
##  ARIMA(4,0,1)            with non-zero mean : 6529.431
##  ARIMA(5,0,0)            with zero mean     : 6528.134
##  ARIMA(5,0,0)            with non-zero mean : 6524.52
##
##
##
##  Best model: ARIMA(1,0,4)            with non-zero mean
```

```
fit_arima_croydon_t <- auto.arima(croydon_terraced_ts_diff, stepwise = FALSE,
                          approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 6553.457
##  ARIMA(0,0,0)            with non-zero mean : 6513.768
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 6548.55
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 6515.428
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 6542.477
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 6516.505
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 6546.288
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 6515.389
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : 6517.226
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 6539.858
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 6516.621
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 6470.465
##  ARIMA(0,0,1)            with non-zero mean : 6445.056
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 6469.091
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 6446.902
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 6467.16
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 6448.631
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 6468.299
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 6446.889
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : 6448.902
```

```
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 6466.381
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 6448.686
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : 6448.143
##  ARIMA(0,0,2)            with zero mean     : 6330.245
##  ARIMA(0,0,2)            with non-zero mean : 6322.811
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 6330.589
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 6320.614
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 6332.369
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 6322.494
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 6330.454
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 6320.758
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 6331.943
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 6322.823
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : 6323.812
##  ARIMA(0,0,3)            with zero mean     : 6311.059
##  ARIMA(0,0,3)            with non-zero mean : 6294.877
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 6313.019
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 6295.187
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : 6312.824
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 6297.176
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : 6312.999
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 6295.102
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : 6312.113
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 6296.908
##  ARIMA(0,0,4)            with zero mean     : 6308.09
##  ARIMA(0,0,4)            with non-zero mean : 6294.142
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 6310.059
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 6294.723
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : 6310.043
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 6294.671
##  ARIMA(0,0,5)            with zero mean     : 6305.536
##  ARIMA(0,0,5)            with non-zero mean : 6293.581
##  ARIMA(1,0,0)            with zero mean     : 6415.162
##  ARIMA(1,0,0)            with non-zero mean : 6405.334
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 6416.406
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 6407.286
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 6418.308
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 6409.258
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 6416.376
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 6407.29
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : 6418.392
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : 6409.347
```

```
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : 6410.844
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 6418.345
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 6409.225
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : 6410.733
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : 6417.252
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 6414.353
##  ARIMA(1,0,1)            with non-zero mean : 6406.036
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 6415.594
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 6407.957
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 6417.643
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 6409.847
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 6415.591
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 6407.964
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : 6417.65
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : 6410.009
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : 6411.397
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 6417.649
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 6409.791
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : 6419.697
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : 6411.255
##  ARIMA(1,0,2)            with zero mean     : 6308.909
##  ARIMA(1,0,2)            with non-zero mean : 6294.251
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : 6310.84
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 6294.417
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : 6310.911
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 6296.468
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : 6310.816
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 6294.352
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : 6310.261
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 6296.264
##  ARIMA(1,0,3)            with zero mean     : 6310.902
##  ARIMA(1,0,3)            with non-zero mean : 6295.389
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : 6312.869
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 6295.792
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : 6312.848
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 6295.718
##  ARIMA(1,0,4)            with zero mean     : 6296.382
##  ARIMA(1,0,4)            with non-zero mean : 6291.747
##  ARIMA(2,0,0)            with zero mean     : 6412.579
##  ARIMA(2,0,0)            with non-zero mean : 6404.864
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 6413.775
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 6406.746
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 6415.832
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 6408.532
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 6413.786
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 6406.758
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : 6415.833
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : 6408.787
```

```
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : 6410.037
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 6415.812
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 6408.454
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : 6417.82
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : 6409.865
##  ARIMA(2,0,1)            with zero mean     : 6399.116
##  ARIMA(2,0,1)            with non-zero mean : 6391.65
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 6400.946
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 6393.719
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 6402.721
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 6394.517
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 6400.962
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 6393.719
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : 6402.961
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : 6395.052
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 6402.642
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 6394.536
##  ARIMA(2,0,2)            with zero mean     : 6310.827
##  ARIMA(2,0,2)            with non-zero mean : 6294.732
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : 6312.818
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : 6295.338
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : 6312.802
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : 6295.267
##  ARIMA(2,0,3)            with zero mean     : 6299.346
##  ARIMA(2,0,3)            with non-zero mean : 6296.902
##  ARIMA(3,0,0)            with zero mean     : 6398.328
##  ARIMA(3,0,0)            with non-zero mean : 6384.605
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 6399.404
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 6386.652
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 6401.441
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 6388.32
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 6399.393
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 6386.654
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : 6401.467
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : 6388.182
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 6401.463
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 6388.285
##  ARIMA(3,0,1)            with zero mean     : 6389.03
##  ARIMA(3,0,1)            with non-zero mean : 6376.238
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 6390.902
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 6378.172
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 6390.904
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 6378.185
##  ARIMA(3,0,2)            with zero mean     : 6305.687
##  ARIMA(3,0,2)            with non-zero mean : 6293.547
##  ARIMA(4,0,0)            with zero mean     : 6366.194
##  ARIMA(4,0,0)            with non-zero mean : 6359.596
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 6367.975
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 6360.877
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 6367.976
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 6360.92
##  ARIMA(4,0,1)            with zero mean     : 6363.337
##  ARIMA(4,0,1)            with non-zero mean : 6358.766
```

```
##  ARIMA(5,0,0)            with zero mean     : 6360.186
##  ARIMA(5,0,0)            with non-zero mean : 6355.967
##
##
##
##  Best model: ARIMA(1,0,4)            with non-zero mean
```

```
fit_arima_croydon_f <- auto.arima(croydon_flat_ts_diff, stepwise = FALSE,
                          approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 6334.932
##  ARIMA(0,0,0)            with non-zero mean : 6301.256
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 6326.621
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 6300.491
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 6322.804
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 6301.576
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 6323.934
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 6300.236
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : 6321.22
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : 6302.077
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : 6323.116
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : 6303.361
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 6321.307
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 6301.833
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : 6303.476
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : 6305.24
##  ARIMA(0,0,1)            with zero mean     : 6259.07
##  ARIMA(0,0,1)            with non-zero mean : 6237.068
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 6255.305
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 6237.383
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 6253.927
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 6238.884
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 6254.157
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 6237.265
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : 6254.067
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : 6239.247
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 6253.699
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 6239.102
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 6134.803
##  ARIMA(0,0,2)            with non-zero mean : 6126.06
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 6132.504
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 6126.988
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 6133.142
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 6129.045
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 6132.28
```

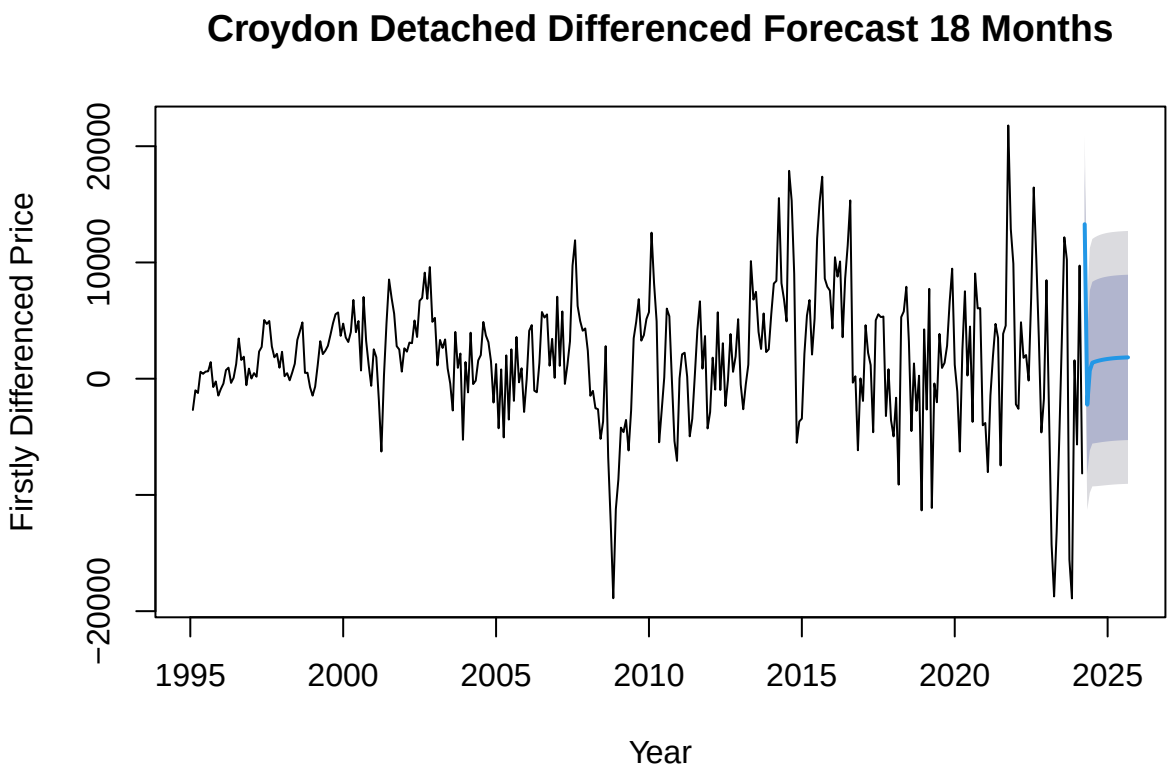
```
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 6126.95
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 6132.174
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 6128.905
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : 6128.84
##  ARIMA(0,0,3)            with zero mean     : Inf
##  ARIMA(0,0,3)            with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)            with zero mean     : Inf
##  ARIMA(0,0,4)            with non-zero mean : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,5)            with zero mean     : Inf
##  ARIMA(0,0,5)            with non-zero mean : Inf
##  ARIMA(1,0,0)            with zero mean     : 6202.159
##  ARIMA(1,0,0)            with non-zero mean : 6193.512
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 6202.827
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 6195.022
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 6204.155
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 6196.997
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 6202.706
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 6195.009
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : 6204.582
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : 6197.066
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 6204.32
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 6197.052
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 6195.664
##  ARIMA(1,0,1)            with non-zero mean : 6189.854
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 6196.859
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 6191.499
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 6198.671
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 6193.554
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 6196.818
```

```
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 6191.496
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : 6198.842
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : 6193.568
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 6198.763
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 6193.566
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : 6195.05
##  ARIMA(1,0,2)            with zero mean     : Inf
##  ARIMA(1,0,2)            with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)            with zero mean     : Inf
##  ARIMA(1,0,3)            with non-zero mean : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : Inf
##  ARIMA(2,0,0)            with zero mean     : 6191.577
##  ARIMA(2,0,0)            with non-zero mean : 6186.005
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 6192.822
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 6187.698
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 6194.69
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 6189.766
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 6192.789
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 6187.699
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : 6194.825
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : 6189.769
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 6194.77
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 6189.763
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : 6191.172
##  ARIMA(2,0,1)            with zero mean     : 6176.323
##  ARIMA(2,0,1)            with non-zero mean : 6170.366
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 6178.273
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 6172.436
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 6180.27
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 6174.488
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 6178.27
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 6172.436
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : 6180.34
```

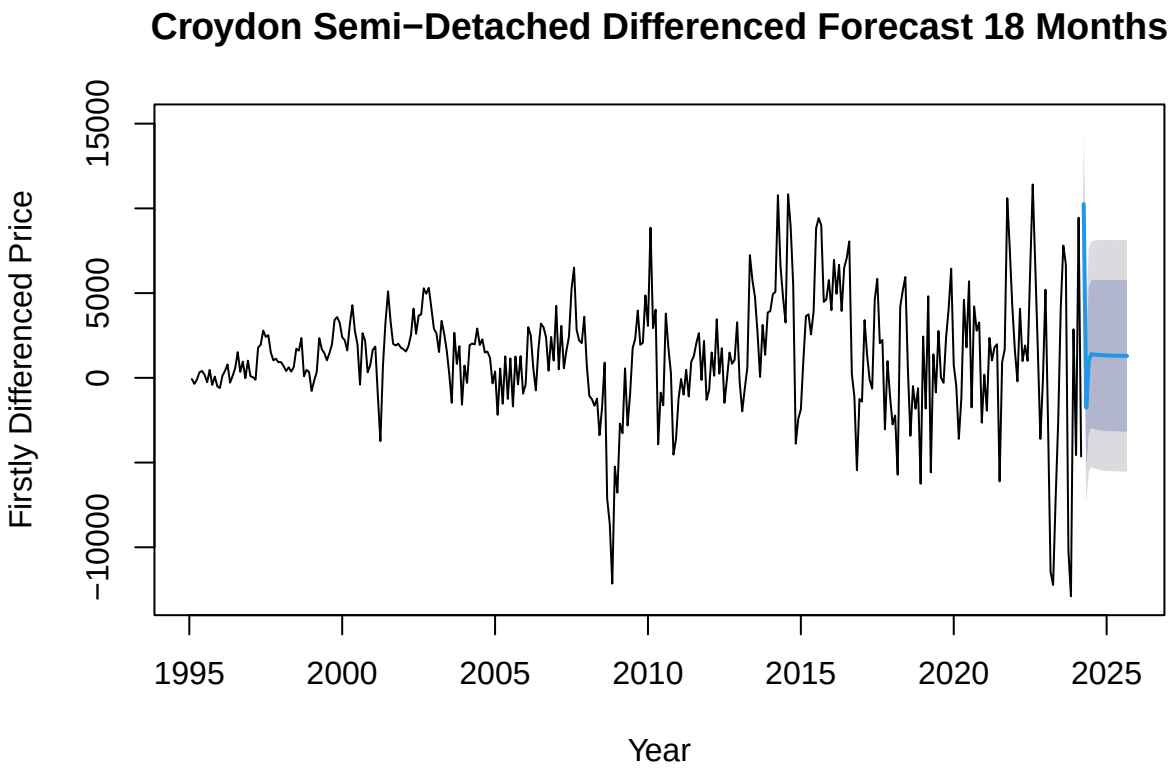
```
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 6180.285
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 6174.488
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 6179.337
##  ARIMA(3,0,0)            with non-zero mean : 6169.772
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 6180.585
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 6171.646
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 6182.107
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 6173.698
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 6180.523
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 6171.643
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : 6182.492
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : 6173.725
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 6182.244
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 6173.721
##  ARIMA(3,0,1)            with zero mean     : 6171.222
##  ARIMA(3,0,1)            with non-zero mean : 6162.23
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 6173.145
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 6164.308
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 6173.135
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 6164.308
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : Inf
##  ARIMA(4,0,0)            with zero mean     : 6147.349
##  ARIMA(4,0,0)            with non-zero mean : 6142.91
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 6148.669
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 6143.936
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 6148.619
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 6143.907
##  ARIMA(4,0,1)            with zero mean     : 6139.878
##  ARIMA(4,0,1)            with non-zero mean : 6137.741
##  ARIMA(5,0,0)            with zero mean     : 6135.574
##  ARIMA(5,0,0)            with non-zero mean : 6133.221
##
##
##
##  Best model: ARIMA(0,0,2)            with non-zero mean
```

```
# Forecast using the ARIMA model for each property type of Croydon Area
forecasted_values_arima_croydon_d <- forecast(fit_arima_croydon_d, h = 18)
forecasted_values_arima_croydon_sd <- forecast(fit_arima_croydon_sd, h = 18)
forecasted_values_arima_croydon_t <- forecast(fit_arima_croydon_t, h = 18)
forecasted_values_arima_croydon_f <- forecast(fit_arima_croydon_f, h = 18)

# Plot the differenced forecast value for each property type of Croydon Area
plot(forecasted_values_arima_croydon_d,
```

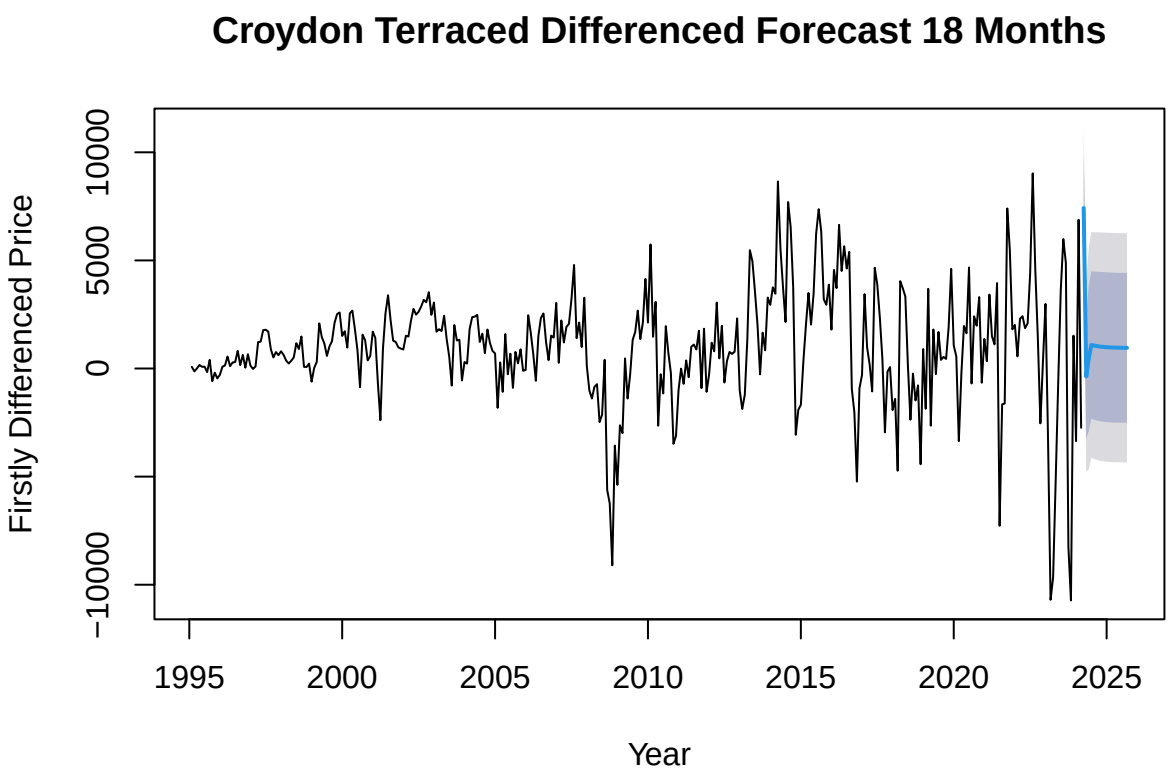
```
     main = "Croydon Detached Differenced Forecast 18 Months",
     ylab = "Firstly Differenced Price", xlab = "Year")
```



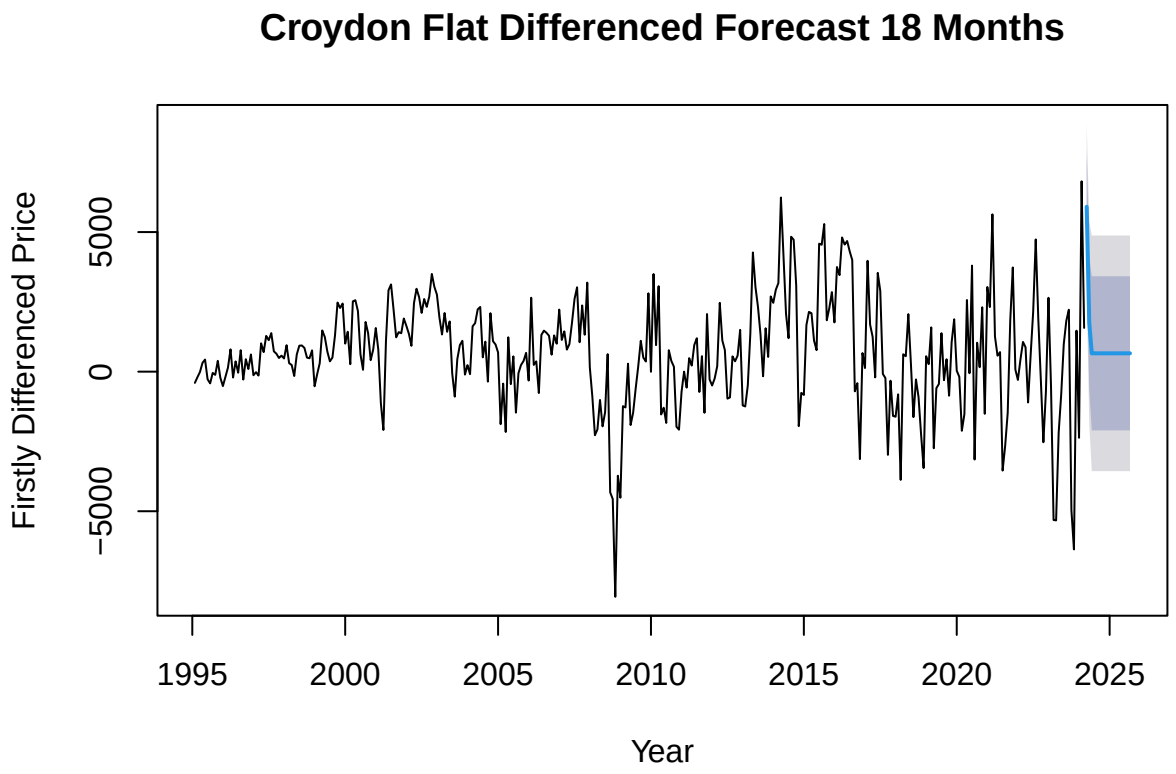

```
plot(forecasted_values_arima_croydon_sd,
     main = "Croydon Semi-Detached Differenced Forecast 18 Months",
     ylab = "Firstly Differenced Price", xlab = "Year")
```




```
plot(forecasted_values_arima_croydon_t,
     main = "Croydon Terraced Differenced Forecast 18 Months",
     ylab = "Firstly Differenced Price", xlab = "Year")
```



```
plot(forecasted_values_arima_croydon_f,
     main = "Croydon Flat Differenced Forecast 18 Months",
     ylab = "Firstly Differenced Price", xlab = "Year")
```




```
# Summary of the model for four different type of properties of Croydon Area
# For Detached of Croydon Area
print(forecasted_values_arima_croydon_d)
```

```
##          Point Forecast     Lo 80     Hi 80      Lo 95     Hi 95
## Apr 2024     13288.6646  8329.413 18247.916   5704.144 20873.185
## May 2024     -2234.5522 -8149.924  3680.820 -11281.333  6812.228
## Jun 2024       699.9471 -6180.803  7580.697  -9823.252 11223.146
## Jul 2024      1375.8584 -5582.263  8333.980  -9265.671 12017.388
## Aug 2024      1455.2830 -5548.294  8458.860  -9255.764 12166.330
## Sep 2024      1522.1370 -5513.468  8557.742  -9237.892 12282.166
## Oct 2024      1578.4102 -5479.799  8636.619  -9216.189 12373.009
## Nov 2024      1625.7770 -5448.403  8699.957  -9193.248 12444.802
## Dec 2024      1665.6471 -5419.827  8751.122  -9170.651 12501.945
## Jan 2025      1699.2071 -5394.259  8792.673  -9149.312 12547.727
## Feb 2025      1727.4555 -5371.666  8826.577  -9129.715 12584.626
## Mar 2025      1751.2331 -5351.894  8854.360  -9112.062 12614.528
## Apr 2025      1771.2474 -5334.716  8877.210  -9096.385 12638.880
## May 2025      1788.0940 -5319.878  8896.066  -9082.611 12658.799
## Jun 2025      1802.2744 -5307.120  8911.669  -9070.606 12675.155
## Jul 2025      1814.2104 -5296.192  8924.613  -9060.212 12688.633
## Aug 2025      1824.2574 -5286.859  8935.374  -9051.257 12699.772
## Sep 2025      1832.7142 -5278.908  8944.337  -9043.574 12709.002
```

```
summary(fit_arima_croydon_d)
```

```
## Series: croydon_detached_ts_diff
## ARIMA(1,0,4) with non-zero mean
##
## Coefficients:
##          ar1      ma1     ma2      ma3     ma4       mean
##       0.8417  -0.1915  0.1614  -0.8052  0.3363  1877.6905
## s.e.  0.0901   0.0978  0.0729   0.0726  0.0552   643.8804
##
## sigma^2 = 14974698:  log likelihood = -3387.76
## AIC=6789.51   AICc=6789.84   BIC=6816.52
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set 13.87798 3836.403 2725.986 23.72384 183.1522 0.5349603 0.00967617
```

```
# For Semi-Detached of Croydon Area
print(forecasted_values_arima_croydon_sd)
```

```
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Apr 2024      10255.874  7142.554 13369.194  5494.462 15017.287
## May 2024      -1761.047 -5451.477  1929.382 -7405.072  3882.977
## Jun 2024       1027.592 -3281.759  5336.943 -5562.992  7618.176
## Jul 2024       1394.208 -2961.506  5749.922 -5267.282  8055.697
## Aug 2024       1375.226 -3016.628  5767.080 -5341.535  8091.986
## Sep 2024       1359.684 -3056.233  5775.600 -5393.878  8113.245
## Oct 2024       1346.958 -3085.017  5778.933 -5431.163  8125.079
## Nov 2024       1336.538 -3106.170  5779.247 -5457.998  8131.074
## Dec 2024       1328.007 -3121.883  5777.896 -5477.512  8133.526
## Jan 2025       1321.022 -3133.676  5775.719 -5491.850  8133.893
## Feb 2025       1315.302 -3142.615  5773.220 -5502.494  8133.099
## Mar 2025       1310.619 -3149.456  5770.694 -5510.477  8131.715
## Apr 2025       1306.785 -3154.736  5768.306 -5516.523  8130.092
## May 2025       1303.645 -3158.845  5766.135 -5521.144  8128.435
## Jun 2025       1301.075 -3162.065  5764.214 -5524.708  8126.857
## Jul 2025       1298.970 -3164.605  5762.545 -5527.479  8125.419
## Aug 2025       1297.247 -3166.620  5761.113 -5529.649  8124.142
## Sep 2025       1295.836 -3168.227  5759.898 -5531.359  8123.030
```

```
summary(fit_arima_croydon_sd)
```

```
## Series: croydon_semi_detached_ts_diff
## ARIMA(1,0,4) with non-zero mean
##
## Coefficients:
##          ar1      ma1     ma2      ma3     ma4      mean
##       0.8188  -0.1823  0.1936  -0.7888  0.3473  1289.460
## s.e.  0.0968   0.1037  0.0764   0.0768  0.0547   401.359
##
## sigma^2 = 5901577:  log likelihood = -3224.97
## AIC=6463.94   AICc=6464.27   BIC=6490.95
```

```
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE     MASE        ACF1
## Training set 2.870847 2408.403 1654.117 -9.76907 175.3152 0.505261 0.002796679
```

```
# For Terraced of Croydon Area
print(forecasted_values_arima_croydon_t)
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95     Hi 95
## Apr 2024      7424.5797  4990.187 9858.973  3701.497 11147.663
## May 2024      -371.2671 -3254.111 2511.577 -4780.197  4037.662
## Jun 2024       480.4136 -2886.655 3847.483 -4669.074  5629.901
## Jul 2024      1089.2171 -2315.922 4494.357 -4118.494  6296.929
## Aug 2024      1060.9544 -2366.674 4488.582 -4181.150  6303.059
## Sep 2024      1038.4018 -2403.469 4480.273 -4225.485  6302.289
## Oct 2024      1020.4057 -2430.504 4471.315 -4257.305  6298.116
## Nov 2024      1006.0455 -2450.607 4462.698 -4280.448  6292.539
## Dec 2024       994.5866 -2465.717 4454.890 -4297.491  6286.665
## Jan 2025       985.4429 -2477.184 4448.070 -4310.188  6281.074
## Feb 2025       978.1465 -2485.959 4442.252 -4319.746  6276.039
## Mar 2025       972.3243 -2492.722 4437.371 -4327.007  6271.656
## Apr 2025       967.6783 -2497.967 4433.324 -4332.569  6267.926
## May 2025       963.9711 -2502.056 4429.998 -4336.860  6264.802
## Jun 2025       961.0128 -2505.257 4427.283 -4340.190  6262.215
## Jul 2025       958.6522 -2507.772 4425.077 -4342.787  6260.091
## Aug 2025       956.7686 -2509.755 4423.292 -4344.821  6258.358
## Sep 2025       955.2655 -2511.320 4421.851 -4346.420  6256.951
```

```
summary(fit_arima_croydon_t)
```

```
## Series: croydon_terraced_ts_diff
## ARIMA(1,0,4) with non-zero mean
##
## Coefficients:
##          ar1      ma1     ma2      ma3     ma4      mean
##       0.7980  -0.1636  0.2084  -0.7788  0.3275  949.3289
## s.e.  0.1234   0.1277  0.0943   0.0963  0.0557  293.6534
##
## sigma^2 = 3608337:  log likelihood = -3138.71
## AIC=6291.42   AICc=6291.75   BIC=6318.42
##
## Training set error measures:
##                     ME    RMSE      MAE      MPE     MAPE      MASE        ACF1
## Training set 3.069001 1883.21 1247.162 52.26089 169.4062 0.5013394 0.005014159
```

```
# For Flat of Croydon Area
print(forecasted_values_arima_croydon_f)
```

```
##          Point Forecast      Lo 80    Hi 80     Lo 95    Hi 95
## Apr 2024      5906.6194  3965.5691 7847.670  2938.039 8875.200
## May 2024      1748.8814  -628.0219 4125.785 -1886.279 5384.041
## Jun 2024       656.0727 -2103.7639 3415.909 -3564.733 4876.878
```

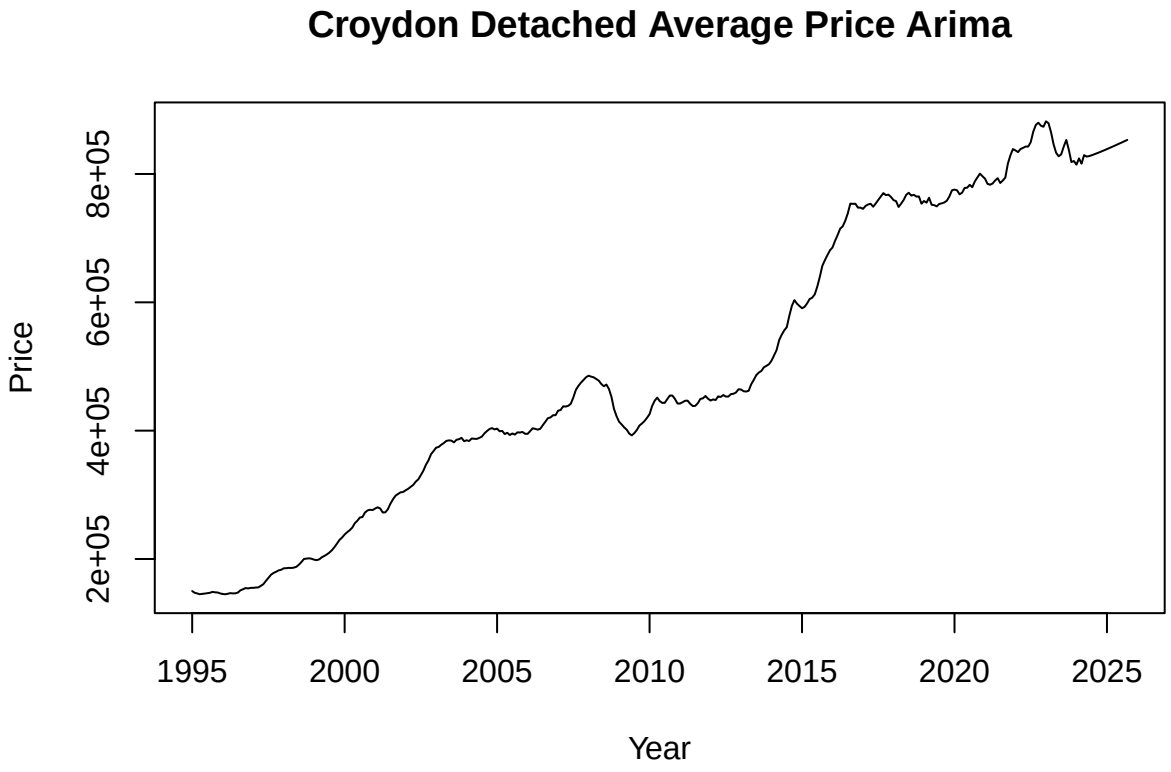
```
## Jul 2024       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Aug 2024       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Sep 2024       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Oct 2024       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Nov 2024       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Dec 2024       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Jan 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Feb 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Mar 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Apr 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## May 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Jun 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Jul 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Aug 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
## Sep 2025       656.0727 -2103.7639 3415.909 -3564.733 4876.878
```

```
summary(fit_arima_croydon_f)
```

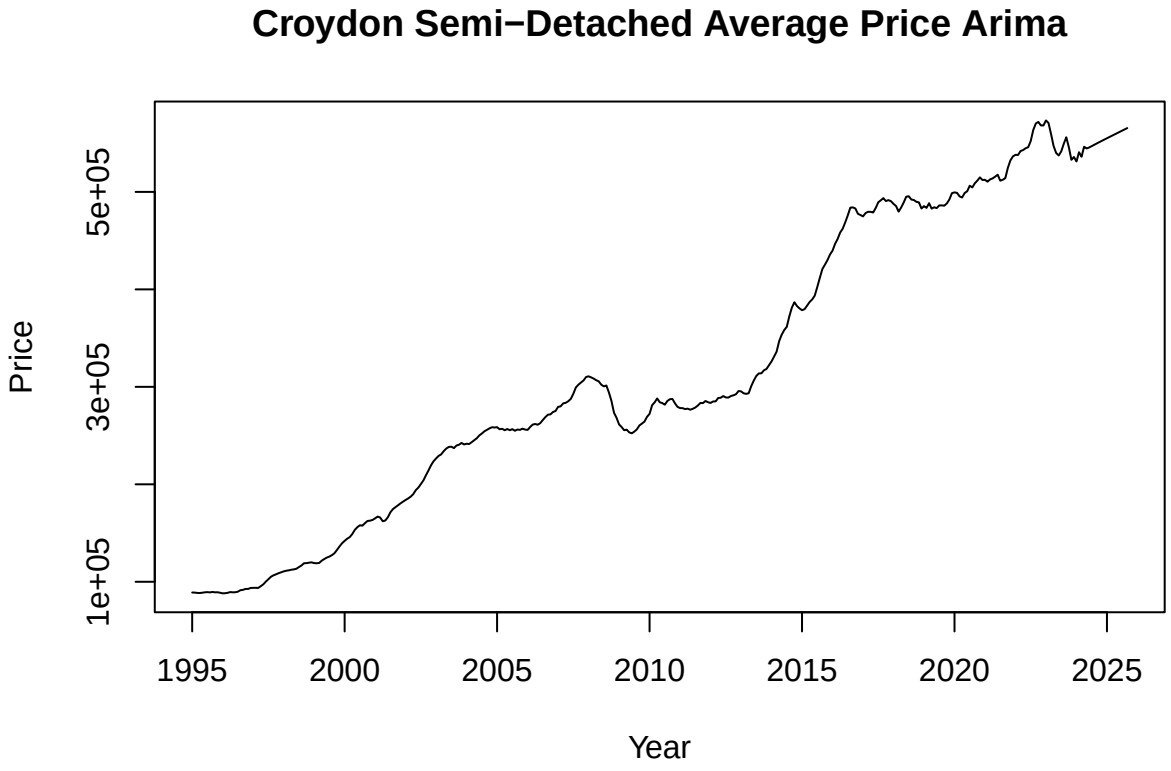
```
## Series: croydon_flat_ts_diff
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1     ma2      mean
##       0.7068  0.7226  656.0727
## s.e.  0.0712  0.0669  195.3703
##
## sigma^2 = 2294042:  log likelihood = -3058.97
## AIC=6125.94   AICc=6126.06   BIC=6141.38
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE    MAPE      MASE       ACF1
## Training set 1.178869 1508.104 1043.734 1122.885 1236.85 0.5475467 -0.1318704
```

```
# Calculate the forecasted actual prices
#By adding the last observed price and the forecasted differences in Croydon
# For detached_ts of Croydon Area
last_value_croydon_d <- as.numeric(tail(croydon_detached_ts, n = 1))
forecasted_values_croydon_d <- c(last_value_croydon_d,
                                 forecasted_values_arima_croydon_d$mean)
cumulative_forecasted_values_croydon_d <- cumsum(forecasted_values_croydon_d)
forecasted_values_croydon_d_ts <- ts(cumulative_forecasted_values_croydon_d[-1],
                             start = c(2024, 2), frequency = 12)

# For semi_detached_ts of Croydon Area
last_value_croydon_sd <- as.numeric(tail(croydon_semi_detached_ts, n = 1))
forecasted_values_croydon_sd <- c(last_value_croydon_sd,
                                  forecasted_values_arima_croydon_sd$mean)
cumulative_forecasted_values_croydon_sd <- cumsum(forecasted_values_croydon_sd)
forecasted_values_croydon_sd_ts <-
  ts(cumulative_forecasted_values_croydon_sd[-1],
     start = c(2024, 2), frequency = 12)

# For terraced_ts of Croydon Area
```

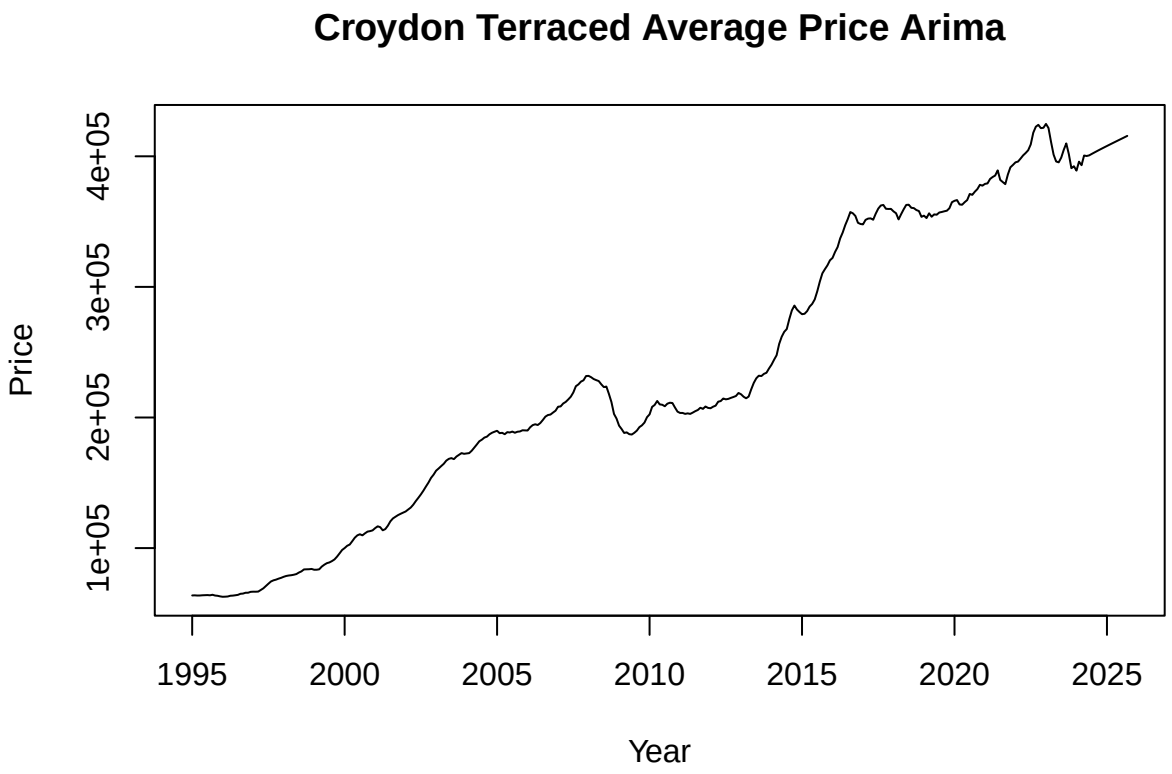
```
last_value_croydon_t <- as.numeric(tail(croydon_terraced_ts, n = 1))
forecasted_values_croydon_t <- c(last_value_croydon_t,
                                 forecasted_values_arima_croydon_t$mean)
cumulative_forecasted_values_croydon_t <- cumsum(forecasted_values_croydon_t)
forecasted_values_croydon_t_ts <- ts(cumulative_forecasted_values_croydon_t[-1],
                             start = c(2024, 2), frequency = 12)

# For flat_ts of Croydon Area
last_value_croydon_f <- as.numeric(tail(croydon_flat_ts, n = 1))
forecasted_values_croydon_f <- c(last_value_croydon_f,
                                 forecasted_values_arima_croydon_f$mean)
cumulative_forecasted_values_croydon_f <- cumsum(forecasted_values_croydon_f)
forecasted_values_croydon_f_ts <- ts(cumulative_forecasted_values_croydon_f[-1],
                                   start = c(2024, 2), frequency = 12)

# Combine the original and forecasted time series of Croydon Area
combined_croydon_detached_ts_Arima <-
  ts(c(as.numeric(croydon_detached_ts),
       as.numeric(forecasted_values_croydon_d_ts)),
     start = c(1995, 1), frequency = 12)
combined_croydon_semi_detached_ts_Arima <-
  ts(c(as.numeric(croydon_semi_detached_ts),
       as.numeric(forecasted_values_croydon_sd_ts)),
     start = c(1995, 1), frequency = 12)
combined_croydon_terraced_ts_Arima <-
  ts(c(as.numeric(croydon_terraced_ts),
       as.numeric(forecasted_values_croydon_t_ts)),
     start = c(1995, 1), frequency = 12)
combined_croydon_flat_ts_Arima <-
  ts(c(as.numeric(croydon_flat_ts),
       as.numeric(forecasted_values_croydon_f_ts)),
     start = c(1995, 1), frequency = 12)

# Plot the combined time series of Croydon Area
plot(combined_croydon_detached_ts_Arima,
     main = "Croydon Detached Average Price Arima",
     ylab = "Price", xlab = "Year")
```



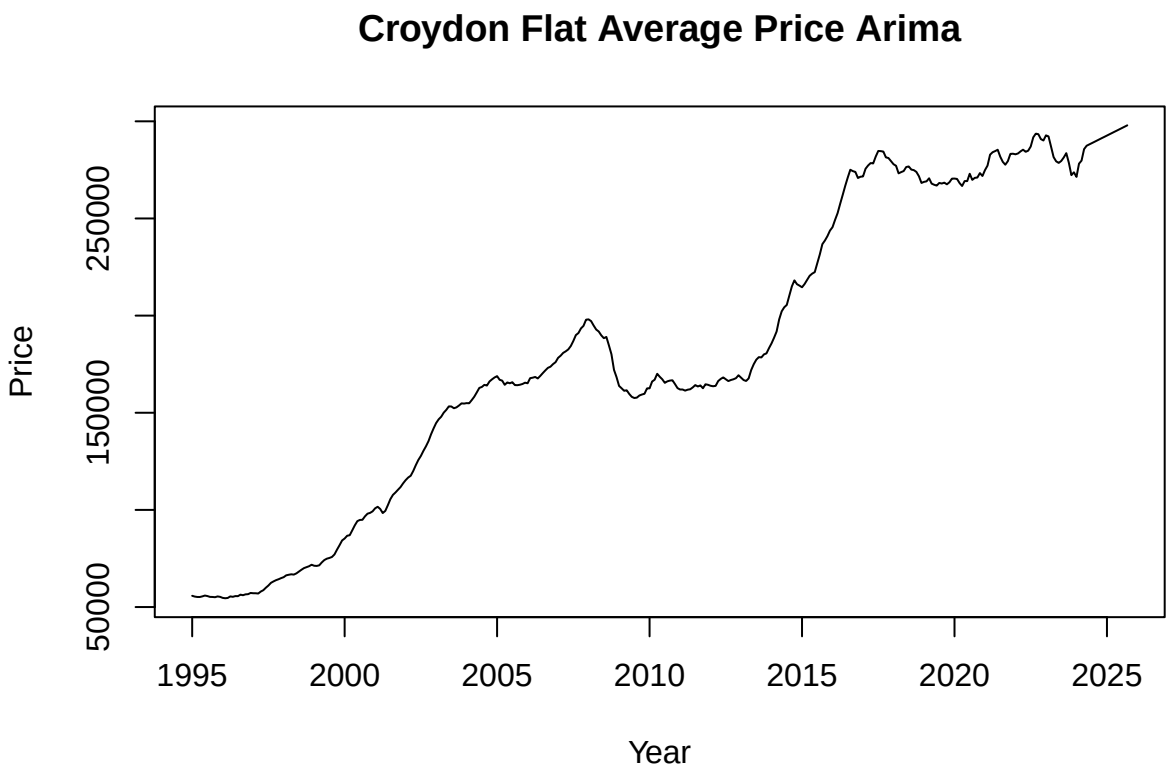

```
plot(combined_croydon_semi_detached_ts_Arima,
     main = "Croydon Semi-Detached Average Price Arima",
     ylab = "Price", xlab = "Year")
```



```
plot(combined_croydon_terraced_ts_Arima,
     main = "Croydon Terraced Average Price Arima",
     ylab = "Price", xlab = "Year")
```

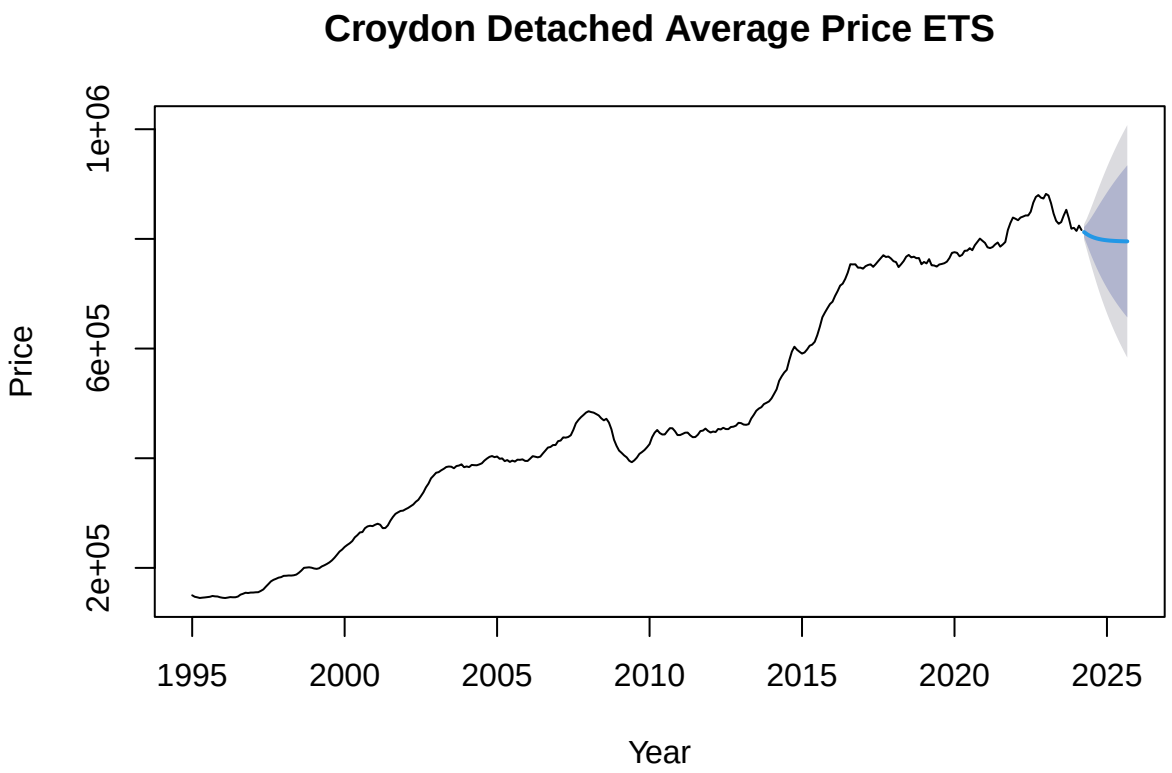



```
plot(combined_croydon_flat_ts_Arima,
     main = "Croydon Flat Average Price Arima",
     ylab = "Price", xlab = "Year")
```

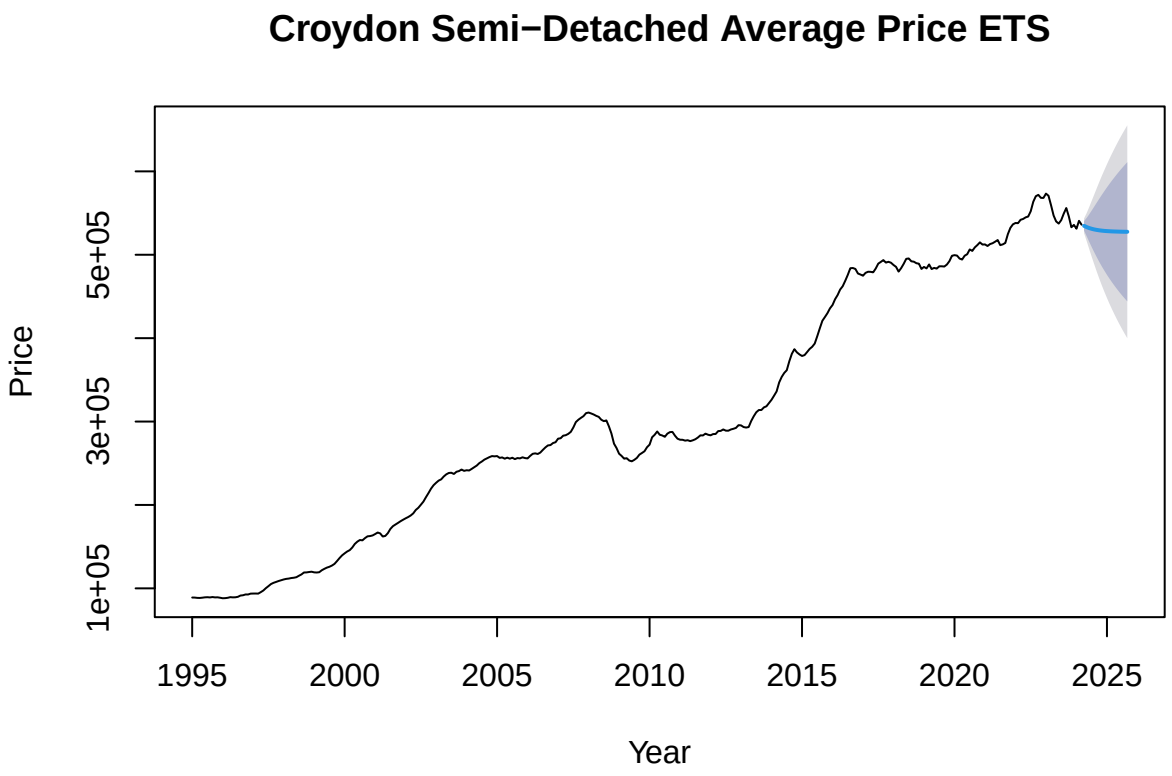



```r
# ETS model for Croydon Area
fit_ets_croydon_d <- ets(croydon_detached_ts)
fit_ets_croydon_sd <- ets(croydon_semi_detached_ts)
fit_ets_croydon_t <- ets(croydon_terraced_ts)
fit_ets_croydon_f <- ets(croydon_flat_ts)

# Forecast using the ETS model for each property type of Croydon Area
forecasted_values_ets_croydon_d <- forecast(fit_ets_croydon_d, h = 18)
forecasted_values_ets_croydon_sd <- forecast(fit_ets_croydon_sd, h = 18)
forecasted_values_ets_croydon_t <- forecast(fit_ets_croydon_t, h = 18)
forecasted_values_ets_croydon_f <- forecast(fit_ets_croydon_f, h = 18)

# Combine the historical and forecasted values by ETS of Croydon Area
combined_croydon_detached_ts_ets <-
  ts(c(croydon_detached_price, forecasted_values_ets_croydon_d$mean),
     start = c(1995, 1), frequency = 12)
combined_croydon_semi_detached_ts_ets <-
  ts(c(croydon_semi_detached_price, forecasted_values_ets_croydon_sd$mean),
     start = c(1995, 1), frequency = 12)
combined_croydon_terraced_ts_ets <-
  ts(c(croydon_terraced_price, forecasted_values_ets_croydon_t$mean),
     start = c(1995, 1), frequency = 12)
combined_croydon_flat_ts_ets <-
  ts(c(croydon_flat_price, forecasted_values_ets_croydon_f$mean),
     start = c(1995, 1), frequency = 12)

# Plot the ETS forecast value for each property type of Croydon Area
plot(forecasted_values_ets_croydon_d,
     main = "Croydon Detached Average Price ETS",
```

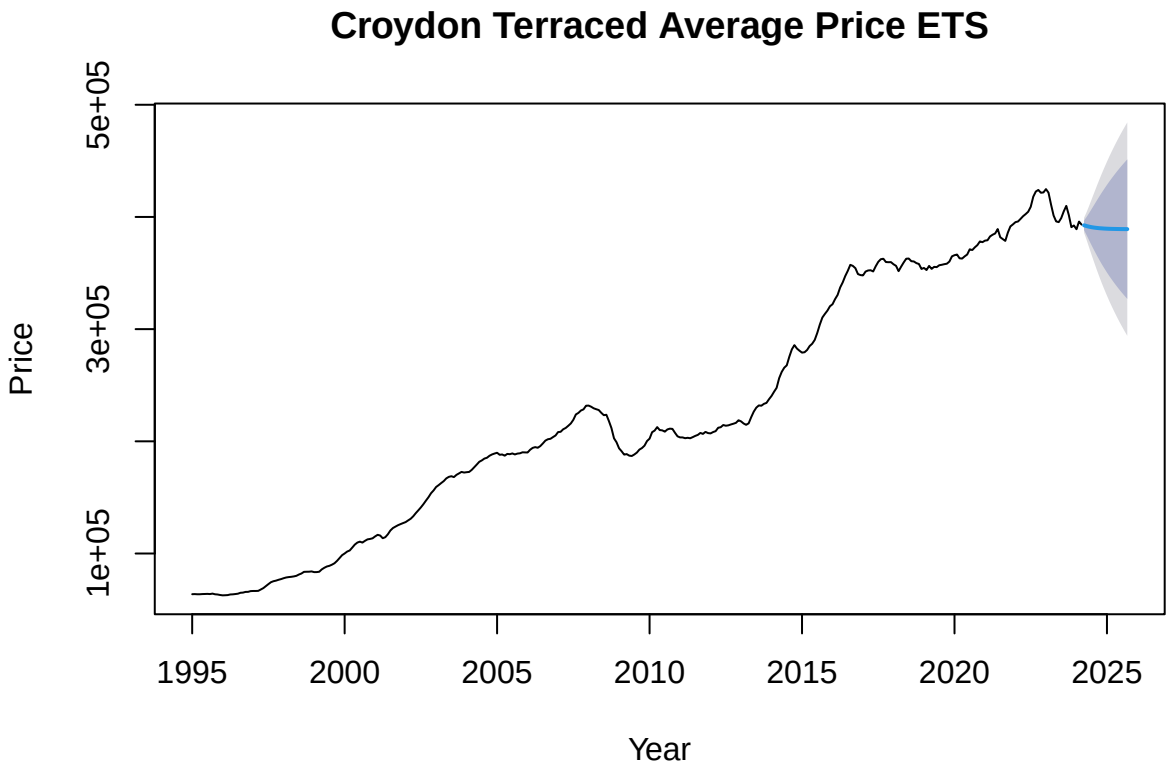
```
     ylab = "Price", xlab = "Year")
```



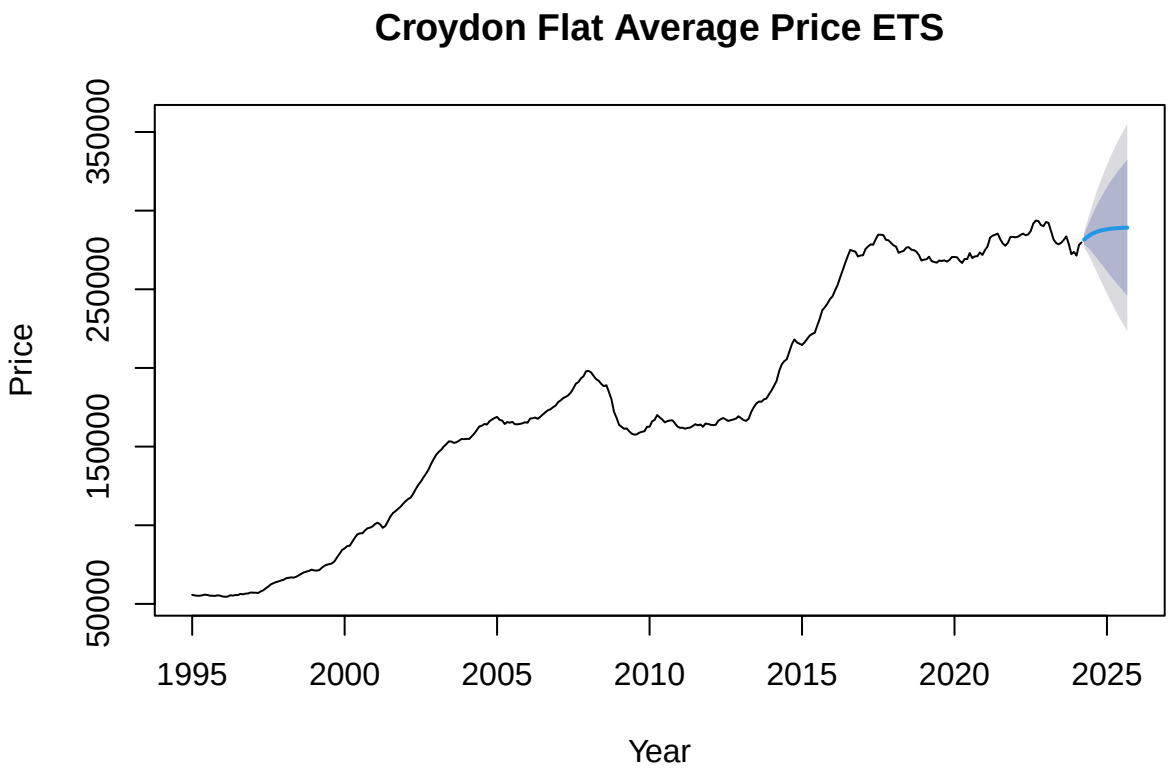

```
plot(forecasted_values_ets_croydon_sd,
     main = "Croydon Semi-Detached Average Price ETS",
     ylab = "Price", xlab = "Year")
```




```
plot(forecasted_values_ets_croydon_t,
     main = "Croydon Terraced Average Price ETS",
     ylab = "Price", xlab = "Year")
```



```
plot(forecasted_values_ets_croydon_f,
     main = "Croydon Flat Average Price ETS",
     ylab = "Price", xlab = "Year")
```




```
# Summary of the ETS model for four different type of properties of Croydon Area
# For Detached of Croydon Area
print(forecasted_values_ets_croydon_d)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## Apr 2024       812439.0 803109.9 821768.1 798171.4  826706.6
## May 2024       808969.2 791170.8 826767.5 781749.0  836189.4
## Jun 2024       806193.3 779295.3 833091.4 765056.3  847330.3
## Jul 2024       803972.6 767819.3 840126.0 748680.9  859264.4
## Aug 2024       802196.1 756873.0 847519.1 732880.5  871511.6
## Sep 2024       800774.8 746498.5 855051.1 717766.4  883783.3
## Oct 2024       799637.8 736693.9 862581.8 703373.3  895902.3
## Nov 2024       798728.2 727435.0 870021.4 689694.7  907761.7
## Dec 2024       798000.5 718687.2 877313.9 676701.3  919299.8
## Jan 2025       797418.4 710411.2 884425.7 664352.3  930484.5
## Feb 2025       796952.7 702567.0 891338.4 652602.2  941303.2
## Mar 2025       796580.1 695116.1 898044.1 641404.4  951755.9
## Apr 2025       796282.1 688022.4 904541.7 630713.3  961850.8
## May 2025       796043.6 681252.6 910834.6 620485.9  971601.3
## Jun 2025       795852.9 674776.3 916929.4 610682.2  981023.5
## Jul 2025       795700.2 668566.3 922834.2 601265.7  990134.8
## Aug 2025       795578.2 662598.2 928558.1 592202.9  998953.4
## Sep 2025       795480.5 656850.1 934110.8 583463.7 1007497.3
```

```
summary(fit_ets_croydon_d)
```

```
## ETS(M,Ad,N)
##
## Call:
##  ets(y = croydon_detached_ts)
##
##   Smoothing parameters:
##     alpha = 0.951
##     beta  = 0.8454
##     phi   = 0.8
##
##   Initial states:
##     l = 147048.4016
##     b = 19.5331
##
##   sigma:  0.009
##
##      AIC     AICc      BIC
## 7855.393 7855.637 7878.558
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE      MAPE      MASE
## Training set 426.0959 4785.155 3411.969 0.1175854 0.7001338 0.1042044
##                     ACF1
## Training set -0.04304432
```

```
# For Semi-Detached of Croydon Area
print(forecasted_values_ets_croydon_sd)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       534670.5 528912.2 540428.9 525863.9 543477.2
## May 2024       533212.2 522330.1 544094.3 516569.4 549855.0
## Jun 2024       532045.5 515693.4 548397.7 507037.1 557054.0
## Jul 2024       531112.2 509206.7 553017.8 497610.6 564613.9
## Aug 2024       530365.6 502960.9 557770.2 488453.8 572277.3
## Sep 2024       529768.2 496994.4 562542.0 479645.0 579891.5
## Oct 2024       529290.4 491317.5 567263.2 471215.9 587364.9
## Nov 2024       528908.1 485925.5 571890.7 463171.9 594644.3
## Dec 2024       528602.2 480805.5 576398.9 455503.5 601701.0
## Jan 2025       528357.6 475940.8 580774.3 448193.1 608522.1
## Feb 2025       528161.8 471312.8 585010.8 441218.8 615104.9
## Mar 2025       528005.3 466902.9 589107.6 434557.3 621453.3
## Apr 2025       527880.0 462692.9 593067.1 428184.9 627575.0
## May 2025       527779.8 458665.8 596893.7 422079.1 633480.4
## Jun 2025       527699.6 454805.8 600593.4 416218.2 639181.0
## Jul 2025       527635.5 451098.5 604172.4 410582.3 644688.6
## Aug 2025       527584.1 447530.7 607637.6 405153.0 650015.3
## Sep 2025       527543.1 444090.6 610995.6 399913.6 655172.6
```

```
summary(fit_ets_croydon_sd)
```

```
## ETS(M,Ad,N)
##
## Call:
##  ets(y = croydon_semi_detached_ts)
##
##   Smoothing parameters:
##     alpha = 0.957
##     beta  = 0.8103
##     phi   = 0.8
##
##   Initial states:
##     l = 88616.6438
##     b = 72.6588
##
##   sigma:  0.0084
##
##      AIC     AICc      BIC
## 7485.147 7485.391 7508.312
##
## Training set error measures:
##                    ME     RMSE     MAE       MPE      MAPE       MASE
## Training set 297.5689 3009.719 2045.24 0.1273933 0.6398718 0.09533882
##                     ACF1
## Training set -0.06498525
```

```
# For Terraced of Croydon Area
print(forecasted_values_ets_croydon_t)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       392566.1 388254.4 396877.8 385971.9 399160.3
## May 2024       391863.7 383737.4 399990.0 379435.6 404291.8
## Jun 2024       391301.7 379107.2 403496.2 372651.9 409951.5
## Jul 2024       390852.2 374527.9 407176.4 365886.4 415817.9
## Aug 2024       390492.5 370078.2 410906.8 359271.6 421713.4
## Sep 2024       390204.8 365796.0 414613.6 352874.7 427534.9
## Oct 2024       389974.6 361696.6 418252.7 346727.1 433222.2
## Nov 2024       389790.5 357782.8 421798.1 340839.0 438741.9
## Dec 2024       389643.2 354050.4 425235.9 335208.7 444077.6
## Jan 2025       389525.3 350490.8 428559.8 329827.2 449223.4
## Feb 2025       389431.0 347093.8 431768.3 324681.8 454180.3
## Mar 2025       389355.6 343848.0 434863.2 319757.7 458953.5
## Apr 2025       389295.3 340742.3 437848.3 315039.9 463550.7
## May 2025       389247.0 337765.6 440728.4 310513.0 467981.0
## Jun 2025       389208.4 334907.7 443509.0 306162.7 472254.0
## Jul 2025       389177.5 332159.0 446195.9 301975.3 476379.7
## Aug 2025       389152.8 329510.6 448794.9 297937.9 480367.6
## Sep 2025       389133.0 326954.4 451311.6 294039.0 484226.9
```

```
summary(fit_ets_croydon_t)
```

```
## ETS(M,Ad,N)
##
## Call:
```

```
##  ets(y = croydon_terraced_ts)
##
##   Smoothing parameters:
##     alpha = 0.9576
##     beta  = 0.8012
##     phi   = 0.8
##
##   Initial states:
##     l = 63709.1785
##     b = 25.2679
##
##   sigma:  0.0086
##
##      AIC     AICc      BIC
## 7275.403 7275.647 7298.568
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE      MAPE       MASE
## Training set 222.6095 2310.065 1527.998 0.1305258 0.6464408 0.09402495
##                     ACF1
## Training set -0.05248312
```

```
# For Flat of Croydon Area
print(forecasted_values_ets_croydon_f)
```

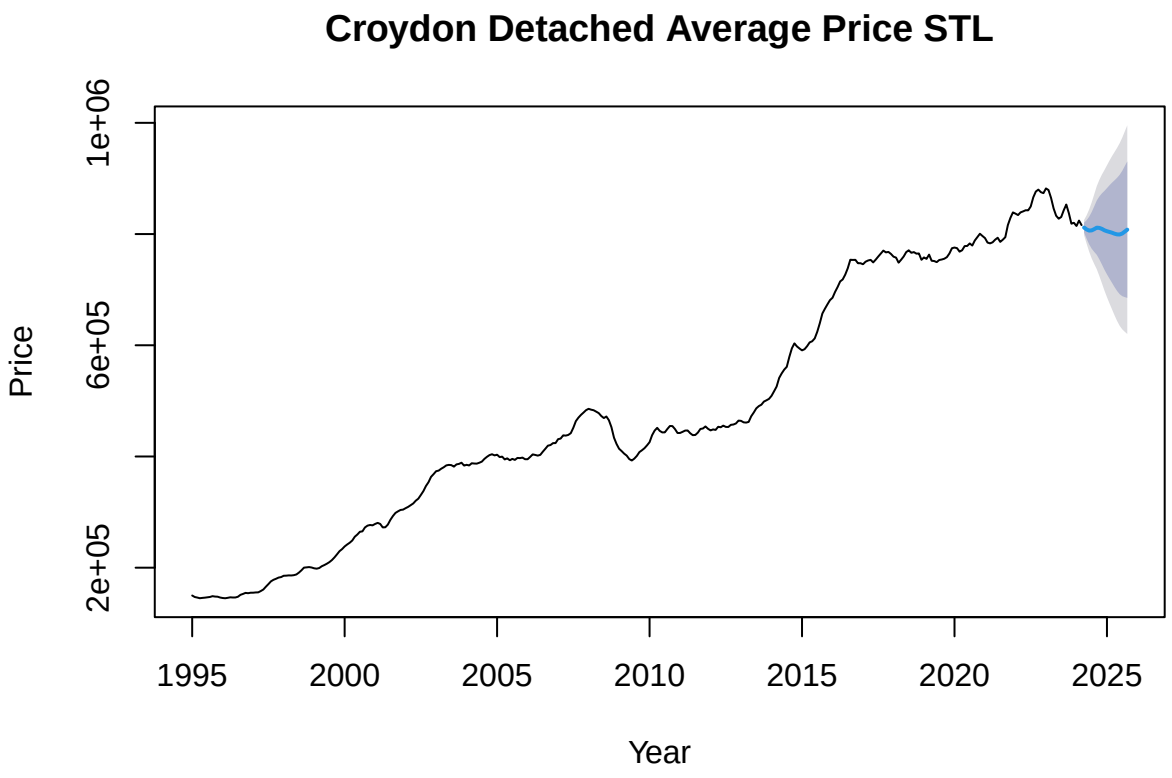
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       281533.4 278361.6 284705.1 276682.6 286384.2
## May 2024       282971.0 277171.9 288770.0 274102.0 291839.9
## Jun 2024       284147.1 275629.1 292665.0 271120.0 297174.1
## Jul 2024       285109.2 273840.1 296378.3 267874.7 302343.8
## Aug 2024       285896.4 271891.1 299901.6 264477.2 307315.6
## Sep 2024       286540.4 269844.5 303236.2 261006.2 312074.5
## Oct 2024       287067.2 267745.1 306389.2 257516.7 316617.7
## Nov 2024       287498.2 265625.1 309371.3 254046.2 320950.2
## Dec 2024       287850.8 263507.3 312194.4 250620.6 325081.1
## Jan 2025       288139.3 261407.7 314870.9 247256.8 329021.8
## Feb 2025       288375.3 259337.5 317413.1 243965.9 332784.7
## Mar 2025       288568.4 257304.4 319832.4 240754.2 336382.6
## Apr 2025       288726.3 255313.0 322139.7 237625.1 339827.6
## May 2025       288855.6 253366.4 324344.7 234579.6 343131.6
## Jun 2025       288961.3 251465.9 326456.6 231617.1 346305.4
## Jul 2025       289047.8 249612.1 328483.4 228736.1 349359.4
## Aug 2025       289118.5 247804.5 330432.6 225934.1 352302.9
## Sep 2025       289176.4 246042.2 332310.6 223208.3 355144.5
```

```
summary(fit_ets_croydon_f)
```

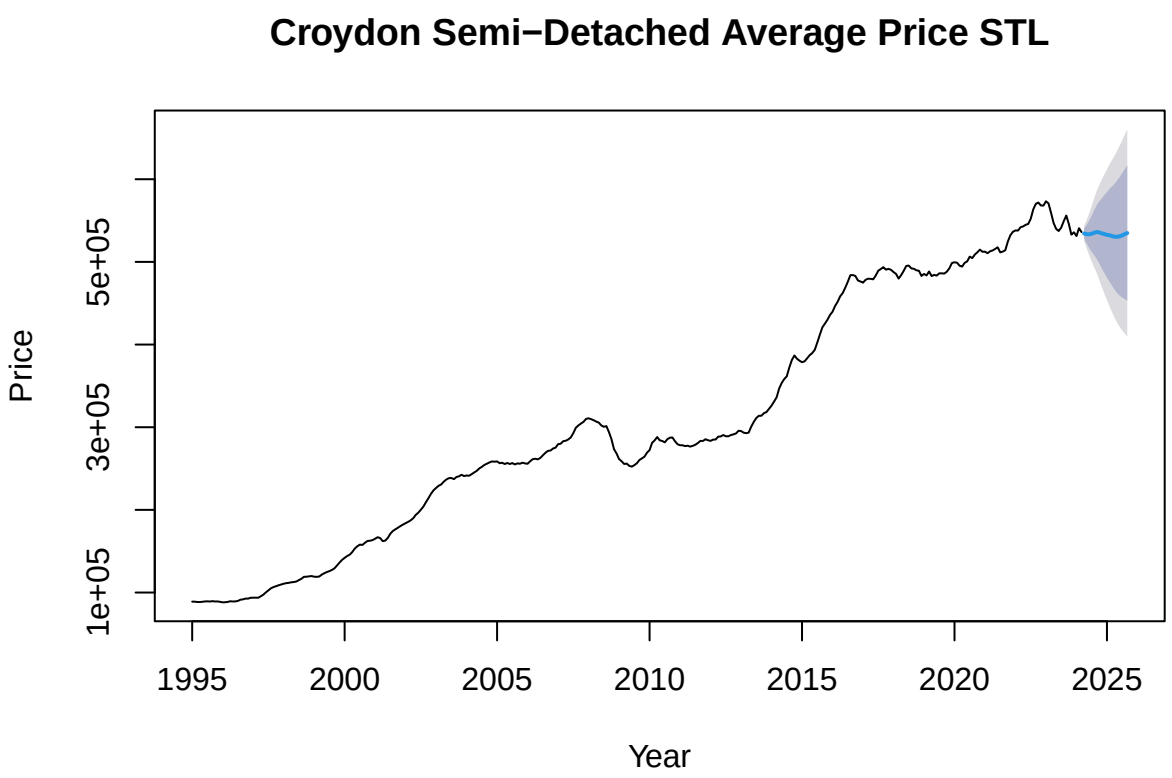
```
## ETS(M,Ad,N)
##
## Call:
##  ets(y = croydon_flat_ts)
##
##   Smoothing parameters:
```

```
##     alpha = 0.9999
##     beta  = 0.6446
##     phi   = 0.8181
##
##   Initial states:
##     l = 55603.6151
##     b = -35.4384
##
##   sigma:  0.0088
##
##      AIC     AICc      BIC
## 7145.956 7146.200 7169.121
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE      MAPE       MASE
## Training set 170.9752 1678.753 1205.753 0.1275841 0.6781077 0.09851299
##                      ACF1
## Training set -0.05581029
```

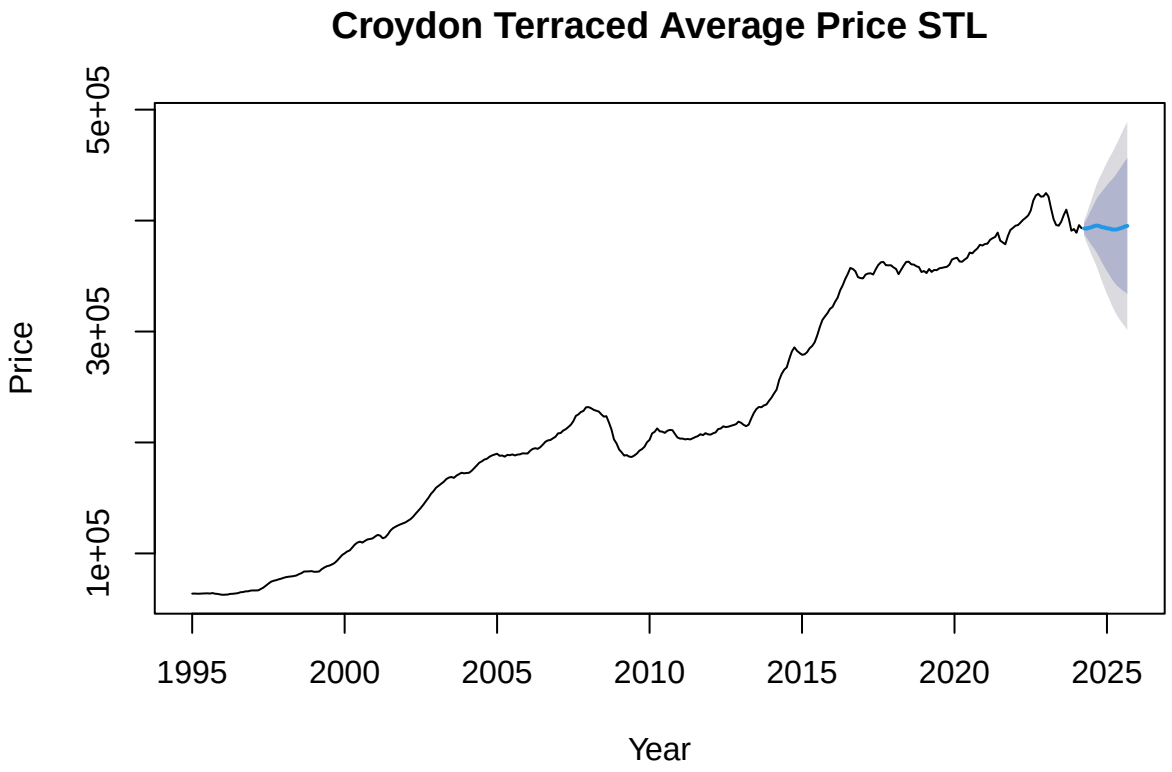
```r
# STL model for Croydon Area
stl_croydon_d <- stl(croydon_detached_ts, s.window = "periodic")
stl_croydon_sd <- stl(croydon_semi_detached_ts, s.window = "periodic")
stl_croydon_t <- stl(croydon_terraced_ts, s.window = "periodic")
stl_croydon_f <- stl(croydon_flat_ts, s.window = "periodic")

# Forecast using the STL model for each property type of Croydon Area
forecasted_values_stl_croydon_d <- forecast(stl_croydon_d, method='ets',h = 18)
forecasted_values_stl_croydon_sd <- forecast(stl_croydon_sd, method='ets',
                                             h = 18)
forecasted_values_stl_croydon_t <- forecast(stl_croydon_t, method='ets', h = 18)
forecasted_values_stl_croydon_f <- forecast(stl_croydon_f, method='ets', h = 18)

# Combine the historical and forecasted values by STL of Croydon Area
combined_croydon_detached_ts_stl <-
  ts(c(croydon_detached_price, forecasted_values_stl_croydon_d$mean),
     start = c(1995, 1), frequency = 12)
combined_croydon_semi_detached_ts_stl <-
  ts(c(croydon_semi_detached_price, forecasted_values_stl_croydon_sd$mean),
     start = c(1995, 1), frequency = 12)
combined_croydon_terraced_ts_stl <-
  ts(c(croydon_terraced_price, forecasted_values_stl_croydon_t$mean),
     start = c(1995, 1), frequency = 12)
combined_croydon_flat_ts_stl <-
  ts(c(croydon_flat_price, forecasted_values_stl_croydon_f$mean),
     start = c(1995, 1), frequency = 12)

# Plot the STL forecast value for each property type of Croydon Area
plot(forecasted_values_stl_croydon_d,
     main = "Croydon Detached Average Price STL",
     ylab = "Price", xlab = "Year")
```



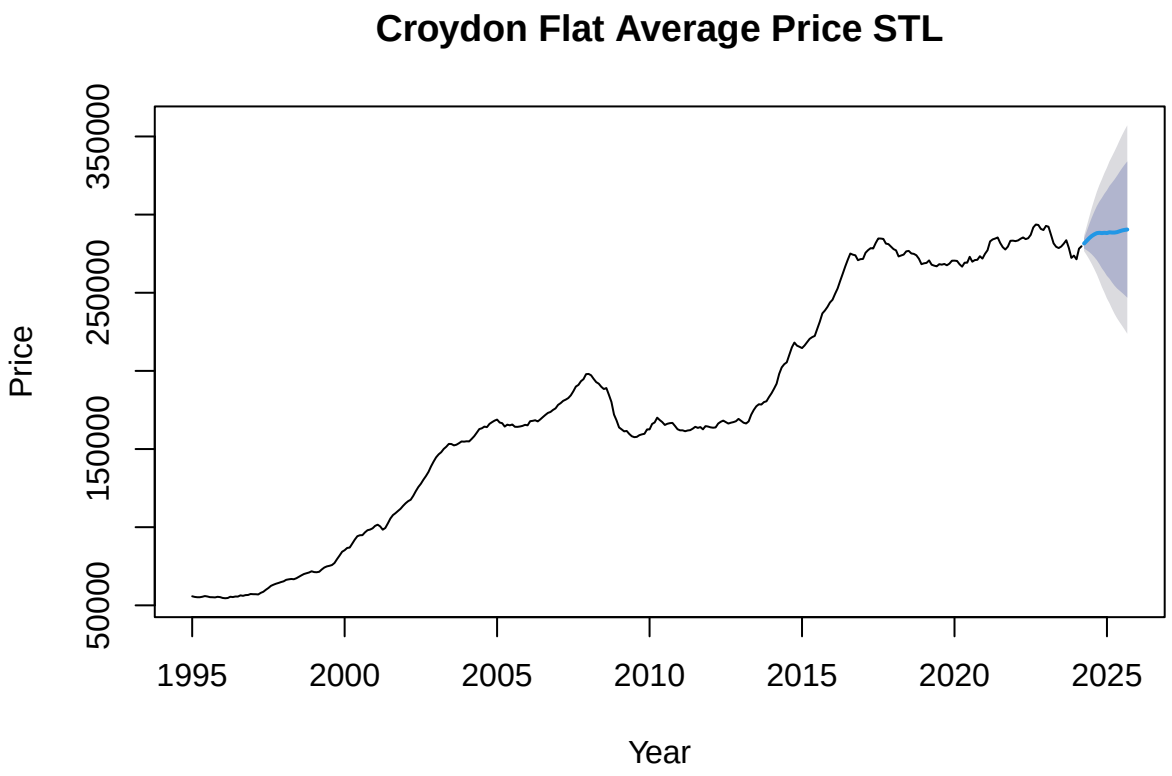

```
plot(forecasted_values_stl_croydon_sd,
     main = "Croydon Semi-Detached Average Price STL",
     ylab = "Price", xlab = "Year")
```



```
plot(forecasted_values_stl_croydon_t,
     main = "Croydon Terraced Average Price STL",
     ylab = "Price", xlab = "Year")
```




```
plot(forecasted_values_stl_croydon_f,
     main = "Croydon Flat Average Price STL",
     ylab = "Price", xlab = "Year")
```




```
# Summary of the STL model for four different type of properties of Croydon Area
# For Detached of Croydon Area
print(forecasted_values_stl_croydon_d)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       811664.6 802770.8 820558.5 798062.6 825266.6
## May 2024       808231.1 791523.1 824939.0 782678.5 833783.7
## Jun 2024       806267.2 781440.1 831094.4 768297.3 844237.1
## Jul 2024       806591.0 773597.6 839584.4 756131.9 857050.0
## Aug 2024       808862.9 767819.1 849906.6 746091.9 871633.9
## Sep 2024       811505.7 762621.2 860390.3 736743.2 886268.2
## Oct 2024       811032.6 754566.9 867498.4 724675.8 897389.5
## Nov 2024       809321.5 745557.4 873085.7 711802.7 906840.4
## Dec 2024       806756.4 735982.8 877529.9 698517.6 914995.2
## Jan 2025       805077.4 727579.3 882575.5 686554.3 923600.5
## Feb 2025       803759.2 719811.2 887707.1 675371.9 932146.5
## Mar 2025       802374.3 712237.4 892511.1 664521.8 940226.7
## Apr 2025       800628.4 704548.2 896708.6 653686.4 947570.4
## May 2025       799402.1 697608.1 901196.1 643721.6 955082.5
## Jun 2025       799204.0 691910.0 906498.1 635112.0 963296.1
## Jul 2025       800940.4 688344.8 913536.0 628740.3 973140.5
## Aug 2025       804342.4 686629.2 922055.6 624315.6 984369.2
## Sep 2025       807889.4 685229.0 930549.7 620296.6 995482.2
```

```
summary(stl_croydon_d)
```

```
##  Call:
##  stl(x = croydon_detached_ts, s.window = "periodic")
```

```
##
##  Time.series components:
##     seasonal             trend            remainder
##  Min.   :-4339.047   Min.   :146355.0   Min.   :-18365.644
##  1st Qu.:-2498.445   1st Qu.:322528.5   1st Qu.: -3021.939
##  Median : -535.162   Median :445723.6   Median :  -111.747
##  Mean   :   -4.945   Mean   :484139.7   Mean   :    50.986
##  3rd Qu.: 1746.294   3rd Qu.:747370.2   3rd Qu.:  2789.967
##  Max.   : 4904.186   Max.   :865562.7   Max.   : 18852.060
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        4245       424842      5812        430740
##    %   1.0         98.6       1.3         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Semi-Detached of Croydon Area
print(forecasted_values_stl_croydon_sd)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       534408.8 528481.1 540336.4 525343.2 543474.3
## May 2024       533533.7 522386.6 544680.7 516485.8 550581.6
## Jun 2024       533431.1 516870.4 549991.9 508103.7 558758.6
## Jul 2024       534085.4 512080.2 556090.7 500431.3 567739.6
## Aug 2024       535260.8 507886.4 562635.2 493395.3 577126.3
## Sep 2024       536168.8 503562.6 568775.1 486301.8 586035.8
## Oct 2024       535663.7 497996.2 573331.3 478056.2 593271.3
## Nov 2024       534466.9 491924.2 577009.5 469403.5 599530.2
## Dec 2024       533699.4 486472.3 580926.5 461471.7 605927.0
## Jan 2025       532685.0 480961.7 584408.3 453581.0 611789.0
## Feb 2025       532258.2 476220.4 588296.0 446555.8 617960.6
## Mar 2025       531169.8 470990.5 591349.2 439133.5 623206.2
## Apr 2025       530483.8 466325.8 594641.8 432362.6 628605.0
## May 2025       530393.0 462408.8 598377.2 426420.1 634365.9
## Jun 2025       530918.1 459249.8 602586.3 421310.9 640525.2
## Jul 2025       532074.6 456854.3 607294.8 417035.1 647114.1
## Aug 2025       533651.7 455002.0 612301.4 413367.4 653936.1
## Sep 2025       534881.3 452915.8 616846.8 409525.9 660236.7
```

```
summary(stl_croydon_sd)
```

```
##  Call:
##  stl(x = croydon_semi_detached_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend            remainder
```

```
##  Min.   :-2105.2105   Min.   : 88791.6   Min.   :-11333.493
##  1st Qu.:-1458.9340   1st Qu.:194142.4   1st Qu.: -1687.898
##  Median : -340.0459   Median :281840.0   Median :  -290.801
##  Mean   :   -5.8914   Mean   :308186.6   Mean   :    23.528
##  3rd Qu.: 1266.8504   3rd Qu.:478199.1   3rd Qu.:  1725.166
##  Max.   : 2520.2144   Max.   :561990.8   Max.   : 12831.278
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        2726       284057      3413        286640
##    %   1.0          99.1       1.2         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Terraced of Croydon Area
print(forecasted_values_stl_croydon_t)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       392892.9 388404.7 397381.1 386028.8 399757.1
## May 2024       392867.4 384485.0 401249.7 380047.6 405687.1
## Jun 2024       393539.3 381117.5 405961.1 374541.8 412536.9
## Jul 2024       394063.6 377579.1 410548.1 368852.7 419274.5
## Aug 2024       395083.4 374591.8 415575.0 363744.1 426422.6
## Sep 2024       395529.1 371131.8 419926.5 358216.6 432841.6
## Oct 2024       395081.3 366904.5 423258.1 351988.6 438174.0
## Nov 2024       394101.4 362283.0 425919.7 345439.4 442763.4
## Dec 2024       393740.3 358421.6 429058.9 339725.1 447755.4
## Jan 2025       393065.8 354386.6 431745.1 333911.0 452220.7
## Feb 2025       392838.1 350933.1 434743.1 328750.0 456926.2
## Mar 2025       391972.2 346970.0 436974.5 323147.2 460797.2
## Apr 2025       391913.1 343934.7 439891.5 318536.5 465289.8
## May 2025       392083.5 341242.3 442924.7 314328.6 469838.4
## Jun 2025       392912.2 339314.1 446510.4 310940.9 474883.6
## Jul 2025       393562.0 337305.2 449818.8 307524.6 479599.3
## Aug 2025       394682.1 335858.0 453506.2 304718.4 484645.8
## Sep 2025       395208.1 333901.4 456514.7 301447.6 488968.5
```

```
summary(stl_croydon_t)
```

```
##  Call:
##  stl(x = croydon_terraced_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal              trend            remainder
##  Min.   :-1482.1624   Min.   : 63378.2   Min.   :-10380.729
##  1st Qu.: -947.5821   1st Qu.:136112.1   1st Qu.: -1344.659
##  Median : -398.3617   Median :207566.1   Median :  -178.065
```

```
##  Mean   :   -6.9441   Mean   :225687.2   Mean   :    18.718
##  3rd Qu.:  557.7394   3rd Qu.:351423.7   3rd Qu.:  1282.306
##  Max.   : 1861.3801   Max.   :415834.3   Max.   : 10921.691
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        1505       215312      2627        216942
##    %   0.7         99.2       1.2         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Flat of Croydon Area
print(forecasted_values_stl_croydon_f)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Apr 2024       281530.3 278329.3 284731.3 276634.8 286425.9
## May 2024       283206.6 277257.5 289155.6 274108.2 292304.9
## Jun 2024       284892.5 276100.5 293684.4 271446.3 298338.6
## Jul 2024       286293.5 274640.4 297946.7 268471.6 304115.5
## Aug 2024       287353.5 272873.7 301833.3 265208.5 309498.5
## Sep 2024       288147.8 270907.3 305388.3 261780.7 314514.9
## Oct 2024       288349.4 268431.9 308267.0 257888.2 318810.7
## Nov 2024       288168.4 265666.2 310670.5 253754.3 322582.5
## Dec 2024       288361.4 263370.1 313352.7 250140.5 326582.3
## Jan 2025       288199.4 260813.9 315585.0 246316.9 330082.0
## Feb 2025       288673.2 258985.9 318360.6 243270.3 334076.2
## Mar 2025       288511.6 256610.8 320412.5 239723.5 337299.8
## Apr 2025       288518.7 254488.1 322549.3 236473.4 340563.9
## May 2025       288797.3 252715.8 324878.7 233615.4 343979.1
## Jun 2025       289365.0 251306.5 327423.5 231159.6 347570.5
## Jul 2025       289871.6 249905.0 329838.2 228747.9 350995.2
## Aug 2025       290216.0 248405.5 332026.4 226272.4 354159.5
## Sep 2025       290437.8 246843.3 334032.2 223765.8 357109.7
```

```
summary(stl_croydon_f)
```

```
##  Call:
##  stl(x = croydon_flat_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend             remainder
##  Min.   :-684.7788   Min.   : 55155.46   Min.   :-5271.367
##  1st Qu.:-512.5068   1st Qu.:122564.62   1st Qu.:-1066.964
##  Median : -98.7477   Median :166927.70   Median : -174.178
##  Mean   :  -3.7481   Mean   :177907.42   Mean   :    9.838
##  3rd Qu.: 416.4787   3rd Qu.:268511.29   3rd Qu.:  903.590
##  Max.   : 887.5969   Max.   :289338.74   Max.   : 6521.877
```

```
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##         929       145947      1971        146441
##    %   0.6         99.7       1.3         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```r
# Split the data into training and test sets of Croydon Area
train_end <- c(2023, 6)
test_start <- c(2023, 6)

#Detached property of Croydon Area
# By ARIMA model for Detached of Croydon Area
croydon_detached_train_arima <- window(croydon_detached_ts_diff,
                                       end = train_end)

# Fit specified ARIMA models to the training data for Detached of Croydon Area
fit_arima_croydon_d_train <- Arima(croydon_detached_train_arima,
                                   order = c(1, 0, 4))
forecasted_values_arima_croydon_d_train <- forecast(fit_arima_croydon_d_train,
                                                    h = 9)

# Add the forecasted differenced values to the last observed value in Croydon
croydon_detached_new_ts <- ts(croydon_detached_price, start = c(1995, 1),
                              end = c(2023, 6), frequency = 12)
last_value_croydon_detached <- as.numeric(tail(croydon_detached_new_ts, n = 1))
forecasted_values_croydon_detached_combined <- c(last_value_croydon_detached,
                                  forecasted_values_arima_croydon_d_train$mean)
cumulative_forecasted_values_croydon_detached <-
  cumsum(forecasted_values_croydon_detached_combined)
forecasted_values_arima_croydon_d_test <-
  ts(cumulative_forecasted_values_croydon_detached,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Detached by ARIMA of Croydon Area
mse_croydon_detached_arima <- mean((window(croydon_detached_ts,
                                           start=test_start) -
                                    forecasted_values_arima_croydon_d_test)^2)
mae_croydon_detached_arima <- mean(abs(window(croydon_detached_ts,
                                           start=test_start) -
                                     forecasted_values_arima_croydon_d_test))

# By ETS model for Detached of Croydon Area
croydon_detached_train_ets <- window(croydon_detached_ts, end = train_end)
croydon_detached_test_ets <- window(croydon_detached_ts, start = test_start)

# Fit ETS models to the training data for Detached of Croydon Area
```

```r
fit_ets_croydon_d_train <- ets(croydon_detached_train_ets)

# Forecast the test period for Detached of Croydon Area
forecasted_values_ets_croydon_d_test <- forecast(fit_ets_croydon_d_train, h = 9)

# Calculate MSE and MAE by ETS for Detached of Croydon Area
mse_croydon_detached_ets <- mean((croydon_detached_test_ets -
                                  forecasted_values_ets_croydon_d_test$mean)^2)
mae_croydon_detached_ets <- mean(abs(croydon_detached_test_ets -
                                    forecasted_values_ets_croydon_d_test$mean))

# By STL model for Detached of Croydon Area
croydon_detached_train_stl <- window(croydon_detached_ts, end = train_end)
croydon_detached_test_stl <- window(croydon_detached_ts, start = test_start)
fit_stl_croydon_d_train <- stl(croydon_detached_train_stl,
                               s.window = "periodic")
forecasted_values_stl_croydon_d_test <- forecast(fit_stl_croydon_d_train,
                                                 method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Detached of Croydon Area
mse_croydon_detached_stl <- mean((croydon_detached_test_stl -
                                  forecasted_values_stl_croydon_d_test$mean)^2)
mae_croydon_detached_stl <- mean(abs(croydon_detached_test_stl -
                                    forecasted_values_stl_croydon_d_test$mean))

# Print MSE and MAE of Croydon Area
print(paste("Croydon Detached MSE for Arima:", mse_croydon_detached_arima))
```

```
## [1] "Croydon Detached MSE for Arima: 330525392.463312"
```

```r
print(paste("Croydon Detached MAE for Arima:", mae_croydon_detached_arima))
```

```
## [1] "Croydon Detached MAE for Arima: 14054.1412276337"
```

```r
print(paste("Croydon Detached MSE for ETS:", mse_croydon_detached_ets))
```

```
## [1] "Croydon Detached MSE for ETS: 348654263.983745"
```

```r
print(paste("Croydon Detached MAE for ETS:", mae_croydon_detached_ets))
```

```
## [1] "Croydon Detached MAE for ETS: 15724.9505179042"
```

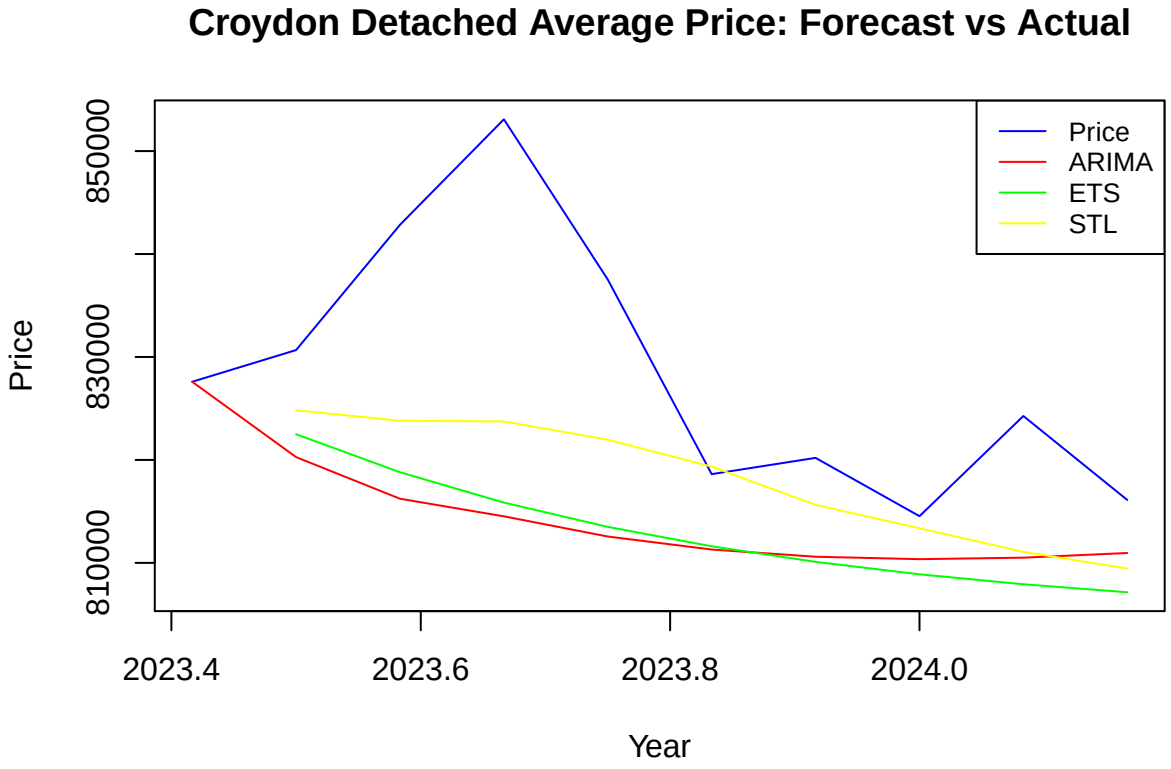
```r
print(paste("Croydon Detached MSE for STL:", mse_croydon_detached_stl))
```

```
## [1] "Croydon Detached MSE for STL: 193448958.22914"
```

```r
print(paste("Croydon Detached MAE for STL:", mae_croydon_detached_stl))
```

```
## [1] "Croydon Detached MAE for STL: 10678.8530999883"
```

```r
# Plot the combined time series with forecast for Detached of Croydon Area
plot(window(croydon_detached_ts, start = train_end), type = "l", col = "blue",
     main = "Croydon Detached Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year", ylim = range(c(window(croydon_detached_ts,
                                                          start = train_end),
                                       forecasted_values_arima_croydon_d_test,
                                       forecasted_values_ets_croydon_d_test$mean,
                                     forecasted_values_stl_croydon_d_test$mean)))
lines(forecasted_values_arima_croydon_d_test, col = "red")
lines(forecasted_values_ets_croydon_d_test$mean, col = "green")
lines(forecasted_values_stl_croydon_d_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```r
# Semi-Detached property of Croydon Area
# By ARIMA model for Semi-Detached of Croydon Area
croydon_semi_detached_train_arima <- window(croydon_semi_detached_ts_diff,
                                            end = train_end)

# Fit specified ARIMA models to the training data for Semi-Detached in Croydon
fit_arima_croydon_sd_train <- Arima(croydon_semi_detached_train_arima,
                                    order = c(1, 0, 4))
forecasted_values_arima_croydon_sd_train <- forecast(fit_arima_croydon_sd_train,
                                                     h = 9)

# Add the forecasted differenced values to the last observed value in Croydon
croydon_semi_detached_new_ts <- ts(croydon_semi_detached_price,
                                   start = c(1995, 1), end = c(2023, 6),
```

```
                                    frequency = 12)
last_value_croydon_semi_detached <-
  as.numeric(tail(croydon_semi_detached_new_ts, n = 1))
forecasted_values_croydon_semi_detached_combined <-
  c(last_value_croydon_semi_detached,
    forecasted_values_arima_croydon_sd_train$mean)
cumulative_forecasted_values_croydon_semi_detached <-
  cumsum(forecasted_values_croydon_semi_detached_combined)
forecasted_values_arima_croydon_sd_test <-
  ts(cumulative_forecasted_values_croydon_semi_detached,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Semi-Detached by ARIMA of Croydon Area
mse_croydon_semi_detached_arima <- mean((window(croydon_semi_detached_ts,
                                                start=test_start) -
                                     forecasted_values_arima_croydon_sd_test)^2)
mae_croydon_semi_detached_arima <- mean(abs(window(croydon_semi_detached_ts,
                                                   start=test_start) -
                                      forecasted_values_arima_croydon_sd_test))

# By ETS model for Semi-Detached of Croydon Area
croydon_semi_detached_train_ets <- window(croydon_semi_detached_ts,
                                          end = train_end)
croydon_semi_detached_test_ets <- window(croydon_semi_detached_ts,
                                         start = test_start)

# Fit ETS models to the training data for Semi-Detached of Croydon Area
fit_ets_croydon_sd_train <- ets(croydon_semi_detached_train_ets)

# Forecast the test period for Semi-Detached of Croydon Area
forecasted_values_ets_croydon_sd_test <- forecast(fit_ets_croydon_sd_train,
                                                  h = 9)

# Calculate MSE and MAE by ETS for Semi-Detached of Croydon Area
mse_croydon_semi_detached_ets <- mean((croydon_semi_detached_test_ets -
                                  forecasted_values_ets_croydon_sd_test$mean)^2)
mae_croydon_semi_detached_ets <- mean(abs(croydon_semi_detached_test_ets -
                                    forecasted_values_ets_croydon_sd_test$mean))

# By STL model for Semi-Detached of Croydon Area
croydon_semi_detached_train_stl <- window(croydon_semi_detached_ts,
                                          end = train_end)
croydon_semi_detached_test_stl <- window(croydon_semi_detached_ts,
                                         start = test_start)
fit_stl_croydon_sd_train <- stl(croydon_semi_detached_train_stl,
                                s.window = "periodic")
forecasted_values_stl_croydon_sd_test <- forecast(fit_stl_croydon_sd_train,
                                                  method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Semi-Detached of Croydon Area
mse_croydon_semi_detached_stl <- mean((croydon_semi_detached_test_stl -
                                  forecasted_values_stl_croydon_sd_test$mean)^2)
mae_croydon_semi_detached_stl <- mean(abs(croydon_semi_detached_test_stl -
```

```
                                    forecasted_values_stl_croydon_sd_test$mean))

# Print MSE and MAE for Semi-Detached of Croydon Area
print(paste("Croydon Semi-Detached MSE for ARIMA:",
            mse_croydon_semi_detached_arima))
```

```
## [1] "Croydon Semi-Detached MSE for ARIMA: 232690045.837641"
```

```
print(paste("Croydon Semi-Detached MAE for ARIMA:",
            mae_croydon_semi_detached_arima))
```

```
## [1] "Croydon Semi-Detached MAE for ARIMA: 13116.3959861087"
```

```
print(paste("Croydon Semi-Detached MSE for ETS:",
            mse_croydon_semi_detached_ets))
```

```
## [1] "Croydon Semi-Detached MSE for ETS: 161323955.940805"
```

```
print(paste("Croydon Semi-Detached MAE for ETS:",
            mae_croydon_semi_detached_ets))
```

```
## [1] "Croydon Semi-Detached MAE for ETS: 10852.5445499123"
```

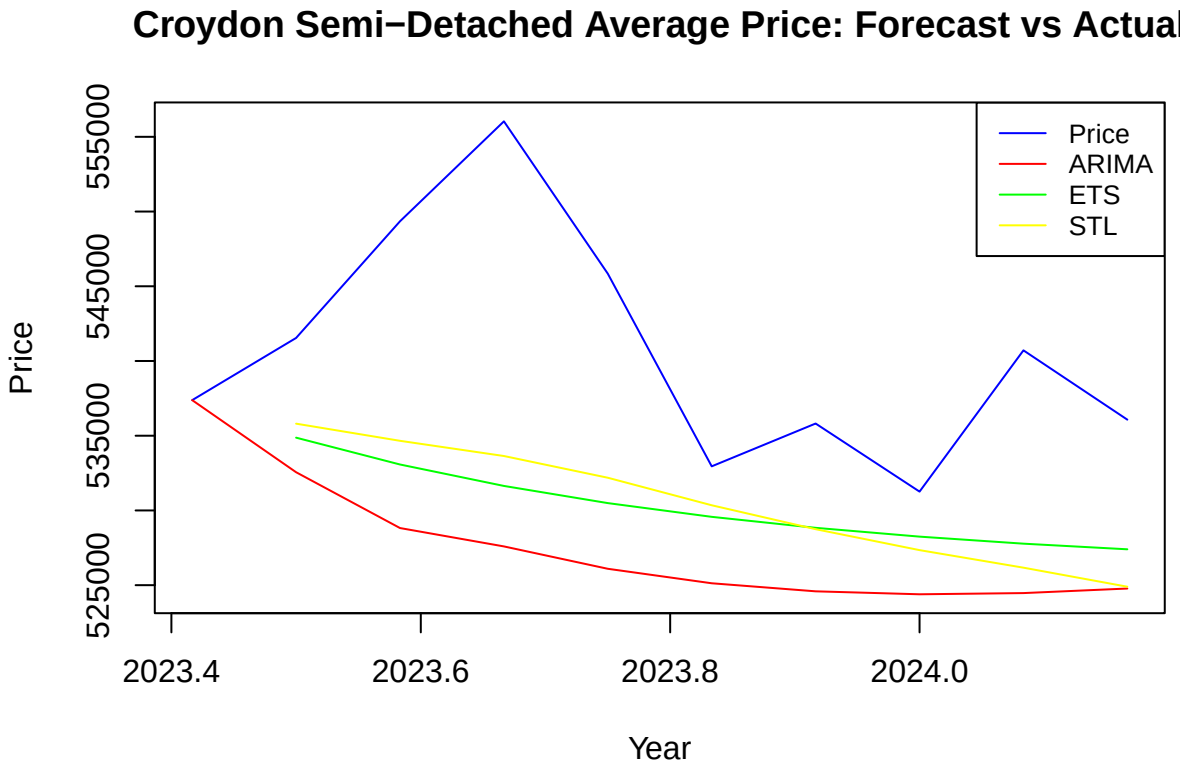
```
print(paste("Croydon Semi-Detached MSE for STL:",
            mse_croydon_semi_detached_stl))
```

```
## [1] "Croydon Semi-Detached MSE for STL: 149458286.2626"
```

```
print(paste("Croydon Semi-Detached MAE for STL:",
            mae_croydon_semi_detached_stl))
```

```
## [1] "Croydon Semi-Detached MAE for STL: 10642.5917819419"
```

```
# Plot the combined time series with forecast for Semi-Detached of Croydon
plot(window(croydon_semi_detached_ts, start = train_end), type = "l",
     col = "blue",
     main = "Croydon Semi-Detached Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year",
     ylim = range(c(window(croydon_semi_detached_ts, start = train_end),
                    forecasted_values_arima_croydon_sd_test,
                    forecasted_values_ets_croydon_sd_test$mean,
                    forecasted_values_stl_croydon_sd_test$mean)))
lines(forecasted_values_arima_croydon_sd_test, col = "red")
lines(forecasted_values_ets_croydon_sd_test$mean, col = "green")
lines(forecasted_values_stl_croydon_sd_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```
# Terraced property of Croydon Area
# By ARIMA model for Terraced of Croydon Area
croydon_terraced_train_arima <- window(croydon_terraced_ts_diff,
                                       end = train_end)

# Fit specified ARIMA models to the training data for Terraced of Croydon Area
fit_arima_croydon_t_train <- Arima(croydon_terraced_train_arima,
                                   order = c(1, 0, 4))
forecasted_values_arima_croydon_t_train <- forecast(fit_arima_croydon_t_train,
                                                    h = 9)

# Add forecasted differenced values of Terraced in Croydon Area
croydon_terraced_new_ts <- ts(croydon_terraced_price, start = c(1995, 1),
                              end = c(2023, 6), frequency = 12)
last_value_croydon_terraced <- as.numeric(tail(croydon_terraced_new_ts, n = 1))
forecasted_values_croydon_terraced_combined <- c(last_value_croydon_terraced,
                                  forecasted_values_arima_croydon_t_train$mean)
cumulative_forecasted_values_croydon_terraced <-
  cumsum(forecasted_values_croydon_terraced_combined)
forecasted_values_arima_croydon_t_test <-
  ts(cumulative_forecasted_values_croydon_terraced,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Terraced by ARIMA of Croydon Area
mse_croydon_terraced_arima <- mean((window(croydon_terraced_ts,
                                           start=test_start) -
                                    forecasted_values_arima_croydon_t_test)^2)
mae_croydon_terraced_arima <- mean(abs(window(croydon_terraced_ts,
                                              start=test_start) -
```

```
                                    forecasted_values_arima_croydon_t_test))

# By ETS model for Terraced of Croydon Area
croydon_terraced_train_ets <- window(croydon_terraced_ts, end = train_end)
croydon_terraced_test_ets <- window(croydon_terraced_ts, start = test_start)

# Fit ETS models to the training data for Terraced of Croydon Area
fit_ets_croydon_t_train <- ets(croydon_terraced_train_ets)

# Forecast the test period for Terraced of Croydon Area
forecasted_values_ets_croydon_t_test <- forecast(fit_ets_croydon_t_train, h = 9)

# Calculate MSE and MAE by ETS for Terraced of Croydon Area
mse_croydon_terraced_ets <- mean((croydon_terraced_test_ets -
                                  forecasted_values_ets_croydon_t_test$mean)^2)
mae_croydon_terraced_ets <- mean(abs(croydon_terraced_test_ets -
                                    forecasted_values_ets_croydon_t_test$mean))

# By STL model for Terraced of Croydon Area
croydon_terraced_train_stl <- window(croydon_terraced_ts, end = train_end)
croydon_terraced_test_stl <- window(croydon_terraced_ts, start = test_start)
fit_stl_croydon_t_train <- stl(croydon_terraced_train_stl,
                               s.window = "periodic")
forecasted_values_stl_croydon_t_test <- forecast(fit_stl_croydon_t_train,
                                                 method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Terraced of Croydon Area
mse_croydon_terraced_stl <- mean((croydon_terraced_test_stl -
                                  forecasted_values_stl_croydon_t_test$mean)^2)
mae_croydon_terraced_stl <- mean(abs(croydon_terraced_test_stl -
                                    forecasted_values_stl_croydon_t_test$mean))

# Print MSE and MAE of Terraced in Croydon Area
print(paste("Croydon Terraced MSE for ARIMA:", mse_croydon_terraced_arima))
```

```
## [1] "Croydon Terraced MSE for ARIMA: 111250169.644619"
```

```
print(paste("Croydon Terraced MAE for ARIMA:", mae_croydon_terraced_arima))
```

```
## [1] "Croydon Terraced MAE for ARIMA: 8382.52954868874"
```

```
print(paste("Croydon Terraced MSE for ETS:", mse_croydon_terraced_ets))
```

```
## [1] "Croydon Terraced MSE for ETS: 60651533.3796279"
```

```
print(paste("Croydon Terraced MAE for ETS:", mae_croydon_terraced_ets))
```

```
## [1] "Croydon Terraced MAE for ETS: 5826.38323999192"
```

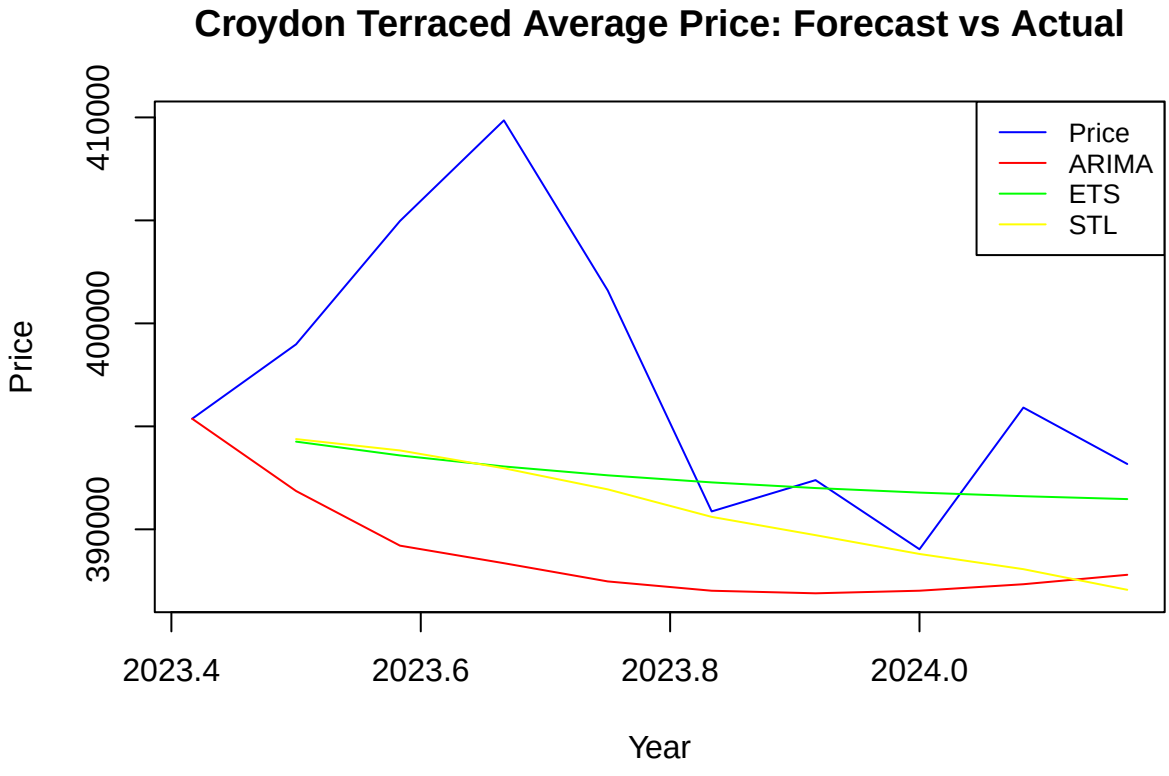
```
print(paste("Croydon Terraced MSE for STL:", mse_croydon_terraced_stl))
```

```
## [1] "Croydon Terraced MSE for STL: 69992263.5603383"
```

```
print(paste("Croydon Terraced MAE for STL:", mae_croydon_terraced_stl))
```

```
## [1] "Croydon Terraced MAE for STL: 6601.16091356683"
```

```
# Plot the combined time series with forecast for Terraced of Croydon Area
plot(window(croydon_terraced_ts, start = train_end), type = "l", col = "blue",
     main = "Croydon Terraced Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year", ylim = range(c(window(croydon_terraced_ts,
                                                          start = train_end),
                                       forecasted_values_arima_croydon_t_test,
                                     forecasted_values_ets_croydon_t_test$mean,
                                   forecasted_values_stl_croydon_t_test$mean)))
lines(forecasted_values_arima_croydon_t_test, col = "red")
lines(forecasted_values_ets_croydon_t_test$mean, col = "green")
lines(forecasted_values_stl_croydon_t_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```
# Flat property of Croydon Area
# By ARIMA model for Flat of Croydon Area
croydon_flat_train_arima <- window(croydon_flat_ts_diff, end = train_end)
```

```r
# Fit specified ARIMA models to the training data for Flat of Croydon Area
fit_arima_croydon_f_train <- Arima(croydon_flat_train_arima, order = c(0, 0, 2))
forecasted_values_arima_croydon_f_train <- forecast(fit_arima_croydon_f_train,
                                                    h = 9)

# Add the forecasted differenced values to the last observed value in Croydon
croydon_flat_new_ts <- ts(croydon_flat_price, start = c(1995, 1),
                          end = c(2023, 6), frequency = 12)
last_value_croydon_flat <- as.numeric(tail(croydon_flat_new_ts, n = 1))
forecasted_values_croydon_flat_combined <-
  c(last_value_croydon_flat, forecasted_values_arima_croydon_f_train$mean)
cumulative_forecasted_values_croydon_flat <-
  cumsum(forecasted_values_croydon_flat_combined)
forecasted_values_arima_croydon_f_test <-
  ts(cumulative_forecasted_values_croydon_flat,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Flat by ARIMA of Croydon Area
mse_croydon_flat_arima <- mean((window(croydon_flat_ts, start=test_start) -
                                forecasted_values_arima_croydon_f_test)^2)
mae_croydon_flat_arima <- mean(abs(window(croydon_flat_ts, start=test_start) -
                                   forecasted_values_arima_croydon_f_test))
# By ETS model for Flat of Croydon Area
croydon_flat_train_ets <- window(croydon_flat_ts, end = train_end)
croydon_flat_test_ets <- window(croydon_flat_ts, start = test_start)

# Fit ETS models to the training data for Flat of Croydon Area
fit_ets_croydon_f_train <- ets(croydon_flat_train_ets)

# Forecast the test period for Flat of Croydon Area
forecasted_values_ets_croydon_f_test <- forecast(fit_ets_croydon_f_train, h = 9)

# Calculate MSE and MAE by ETS for Flat of Croydon Area
mse_croydon_flat_ets <- mean((croydon_flat_test_ets -
                                forecasted_values_ets_croydon_f_test$mean)^2)
mae_croydon_flat_ets <- mean(abs(croydon_flat_test_ets -
                                   forecasted_values_ets_croydon_f_test$mean))

# By STL model for Flat of Croydon Area
croydon_flat_train_stl <- window(croydon_flat_ts, end = train_end)
croydon_flat_test_stl <- window(croydon_flat_ts, start = test_start)
fit_stl_croydon_f_train <- stl(croydon_flat_train_stl, s.window = "periodic")
forecasted_values_stl_croydon_f_test <- forecast(fit_stl_croydon_f_train,
                                                 method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Flat of Croydon Area
mse_croydon_flat_stl <- mean((croydon_flat_test_stl -
                                forecasted_values_stl_croydon_f_test$mean)^2)
mae_croydon_flat_stl <- mean(abs(croydon_flat_test_stl -
                                   forecasted_values_stl_croydon_f_test$mean))

# Print MSE and MAE of Flat in Croydon Area
print(paste("Croydon Flat MSE for ARIMA:", mse_croydon_flat_arima))
```

```
## [1] "Croydon Flat MSE for ARIMA: 20113708.1656028"
```

```
print(paste("Croydon Flat MAE for ARIMA:", mae_croydon_flat_arima))
```

```
## [1] "Croydon Flat MAE for ARIMA: 3554.18632357111"
```

```
print(paste("Croydon Flat MSE for ETS:", mse_croydon_flat_ets))
```

```
## [1] "Croydon Flat MSE for ETS: 18985222.1987678"
```

```
print(paste("Croydon Flat MAE for ETS:", mae_croydon_flat_ets))
```

```
## [1] "Croydon Flat MAE for ETS: 3822.54299288067"
```

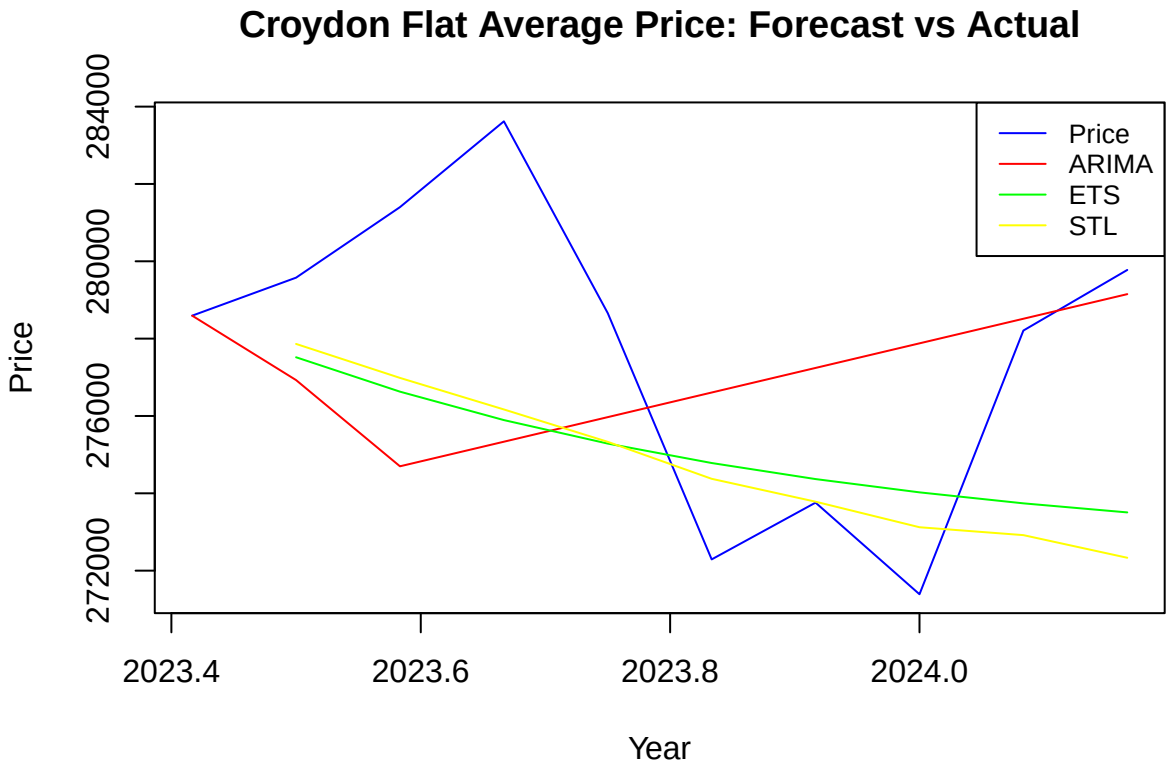
```
print(paste("Croydon Flat MSE for STL:", mse_croydon_flat_stl))
```

```
## [1] "Croydon Flat MSE for STL: 19972274.9261258"
```

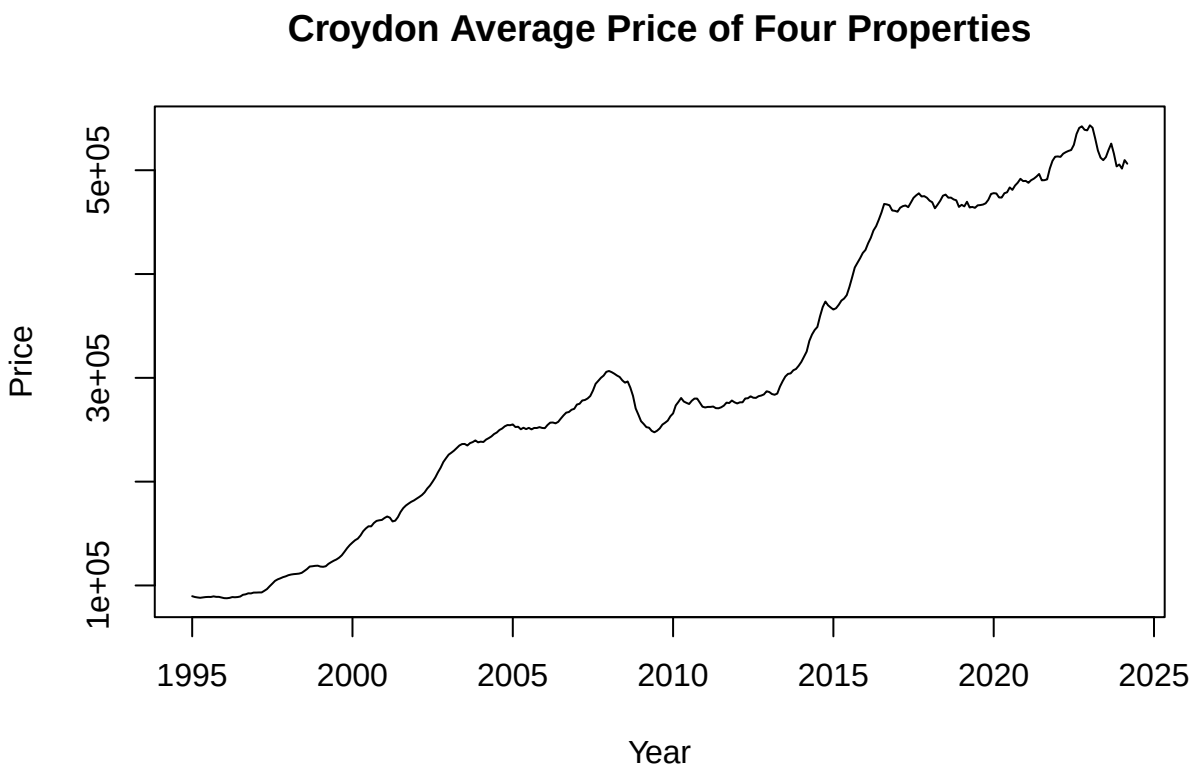
```
print(paste("Croydon Flat MAE for STL:", mae_croydon_flat_stl))
```

```
## [1] "Croydon Flat MAE for STL: 3720.21139188537"
```

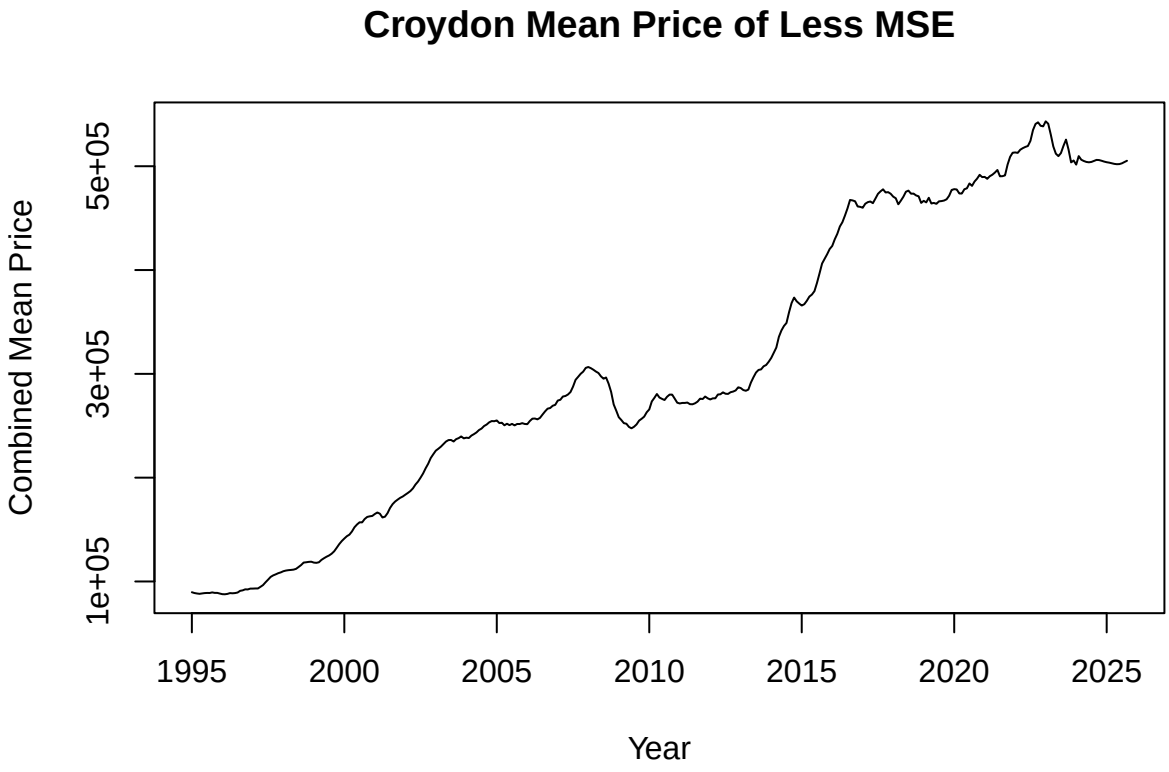
```
# Plot the combined time series with forecast for Flat of Croydon Area
plot(window(croydon_flat_ts, start = train_end), type = "l", col = "blue",
     main = "Croydon Flat Average Price: Forecast vs Actual", ylab = "Price",
     xlab = "Year", ylim = range(c(window(croydon_flat_ts, start = train_end),
                                   forecasted_values_arima_croydon_f_test,
                                   forecasted_values_ets_croydon_f_test$mean,
                                   forecasted_values_stl_croydon_f_test$mean)))
lines(forecasted_values_arima_croydon_f_test, col = "red")
lines(forecasted_values_ets_croydon_f_test$mean, col = "green")
lines(forecasted_values_stl_croydon_f_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```



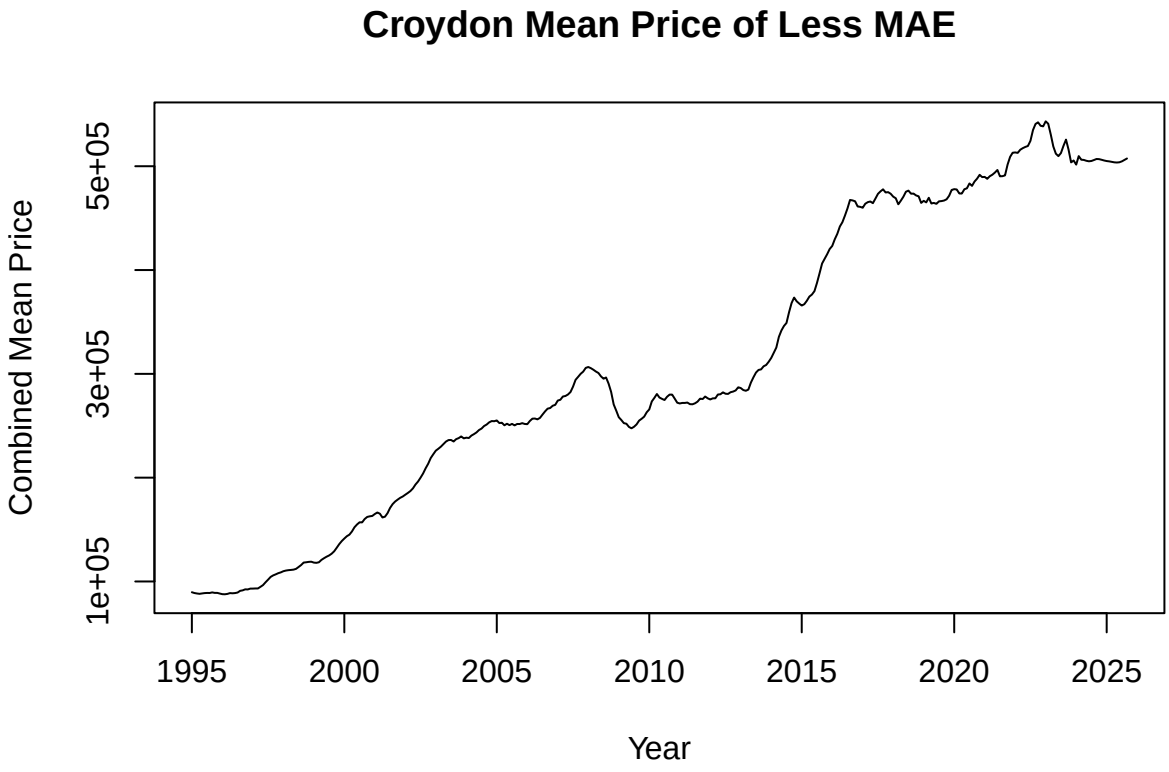

```
# Calculate the mean price of four property types of Croydon Area
combined_mean_price_croydon <- (croydon_detached_price +
                                  croydon_semi_detached_price +
                                  croydon_terraced_price +
                                  croydon_flat_price) / 4
combined_mean_price_croydon_ts <- ts(combined_mean_price_croydon,
                                     start = c(1995, 1), frequency = 12)
plot(combined_mean_price_croydon_ts,
     main = "Croydon Average Price of Four Properties",
     ylab = "Price", xlab = "Year")
```




```r
# If we look for less MSE in Croydon Area,
# STL model is the best for Detached
# STL model is the best for Semi-Detached
# ETS model is the best for Terraced
# ETS model is the best for Flat

# Calculate the combined mean price for property types in Croydon with less MSE
combined_mean_price_croydon_less_MSE <- (combined_croydon_detached_ts_stl +
                                           combined_croydon_semi_detached_ts_stl
                                         + combined_croydon_terraced_ts_ets +
                                           combined_croydon_flat_ts_ets) / 4
combined_mean_price_croydon_less_MSE_ts <-
  ts(combined_mean_price_croydon_less_MSE, start = c(1995, 1), frequency = 12)
plot(combined_mean_price_croydon_less_MSE_ts,
     main = "Croydon Mean Price of Less MSE",
     ylab = "Combined Mean Price", xlab = "Year")
```




```
# If we look for less MAE in Croydon Area,
# STL model is the best for Detached
# STL model is the best for Semi-Detached
# ETS model is the best for Terraced
# ARIMA model is the best for Flat

# Calculate the combined mean price for property types in Croydon with less MAE
combined_mean_price_croydon_less_MAE <- (combined_croydon_detached_ts_stl +
                                           combined_croydon_semi_detached_ts_stl
                                         + combined_croydon_terraced_ts_ets +
                                           combined_croydon_flat_ts_Arima) / 4
combined_mean_price_croydon_less_MAE_ts <-
  ts(combined_mean_price_croydon_less_MAE, start = c(1995, 1), frequency = 12)
plot(combined_mean_price_croydon_less_MAE_ts,
     main = "Croydon Mean Price of Less MAE",
     ylab = "Combined Mean Price", xlab = "Year")
```




```
# Compare two models for which one is better in Croydon Area
combined_mean_price_croydon_less_MSE_test <-
  (forecasted_values_stl_croydon_d_test$mean +
     forecasted_values_stl_croydon_sd_test$mean +
     forecasted_values_ets_croydon_t_test$mean +
     forecasted_values_ets_croydon_f_test$mean) / 4
combined_mean_price_croydon_less_MSE_test_ts <-
  ts(combined_mean_price_croydon_less_MSE_test,
     start = test_start, frequency = 12)

combined_mean_price_croydon_less_MAE_test <-
  (forecasted_values_stl_croydon_d_test$mean +
     forecasted_values_stl_croydon_sd_test$mean +
     forecasted_values_ets_croydon_t_test$mean +
     forecasted_values_arima_croydon_f_test) / 4
combined_mean_price_croydon_less_MAE_test_ts <-
  ts(combined_mean_price_croydon_less_MAE_test,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for combined mean prices in Croydon Area
combined_mean_price_croydon_test <- window(combined_mean_price_croydon_ts,
                                           start = test_start)
mse_combined_croydon_less_MSE <- mean((combined_mean_price_croydon_test -
                                 combined_mean_price_croydon_less_MSE_test_ts)^2)
mae_combined_croydon_less_MSE <- mean(abs(combined_mean_price_croydon_test -
                                   combined_mean_price_croydon_less_MSE_test_ts))

mse_combined_croydon_less_MAE <- mean((combined_mean_price_croydon_test -
                                 combined_mean_price_croydon_less_MAE_test_ts)^2)
```

```r
mae_combined_croydon_less_MAE <- mean(abs(combined_mean_price_croydon_test -
                                  combined_mean_price_croydon_less_MAE_test_ts))

# Print MSE and MAE for combined mean prices in Croydon Area
print(paste("Croydon Combined Mean Price MSE for Less MSE Model:",
            mse_combined_croydon_less_MSE))
```

```
## [1] "Croydon Combined Mean Price MSE for Less MSE Model: 95807599.7330046"
```

```r
print(paste("Croydon Combined Mean Price MAE for Less MSE Model:",
            mae_combined_croydon_less_MSE))
```

```
## [1] "Croydon Combined Mean Price MAE for Less MSE Model: 7535.4884935506"
```

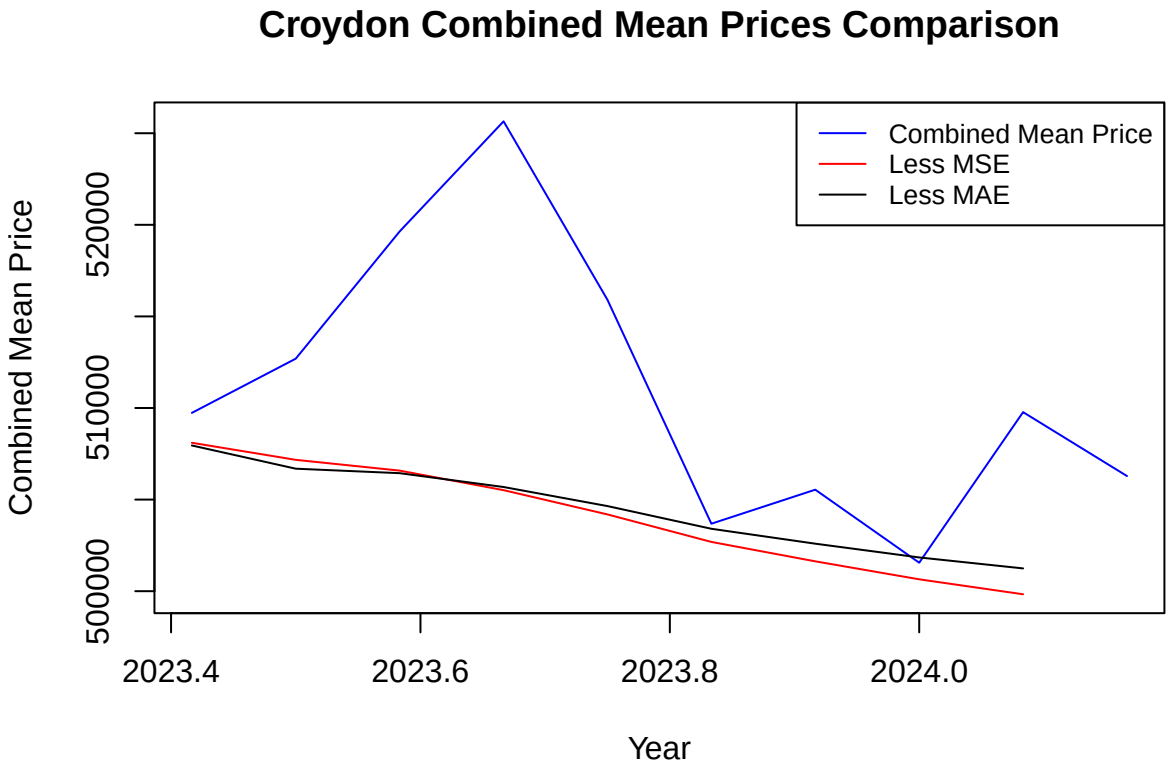
```r
print(paste("Croydon Combined Mean Price MSE for Less MAE Model:",
            mse_combined_croydon_less_MAE))
```

```
## [1] "Croydon Combined Mean Price MSE for Less MAE Model: 91148744.724732"
```

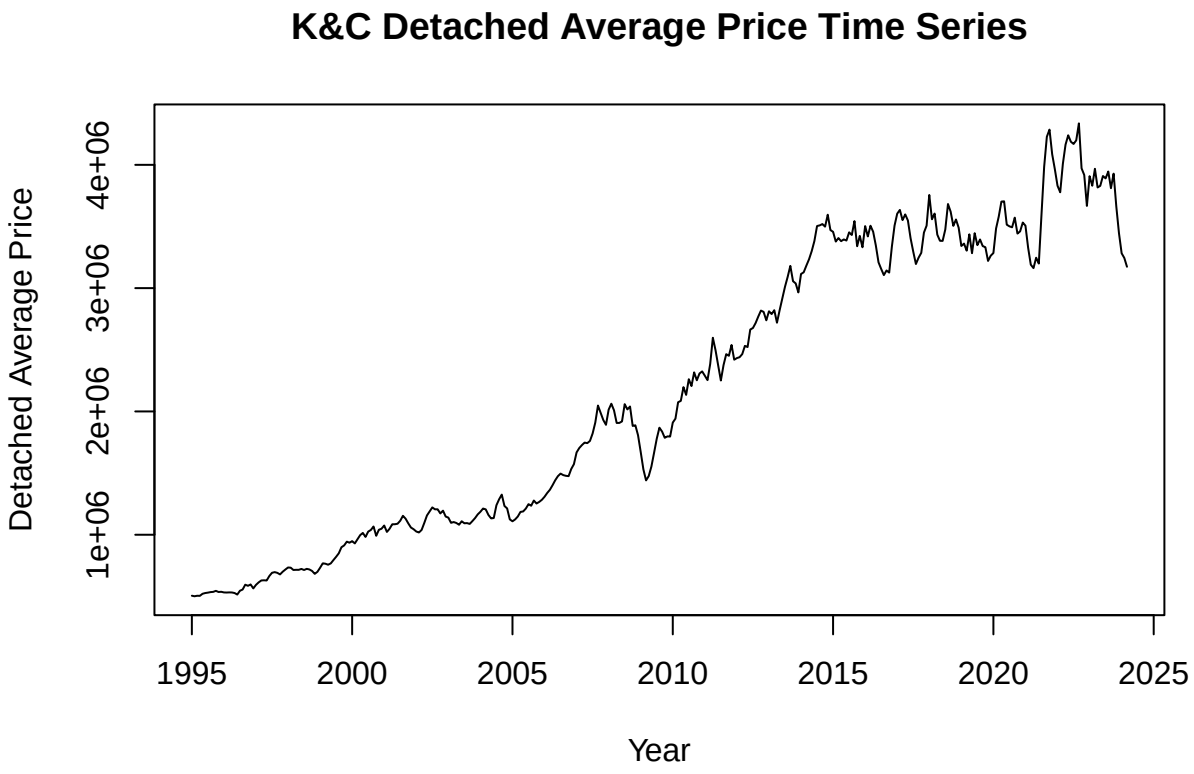
```r
print(paste("Croydon Combined Mean Price MAE for Less MAE Model:",
            mae_combined_croydon_less_MAE))
```

```
## [1] "Croydon Combined Mean Price MAE for Less MAE Model: 7138.67941386594"
```

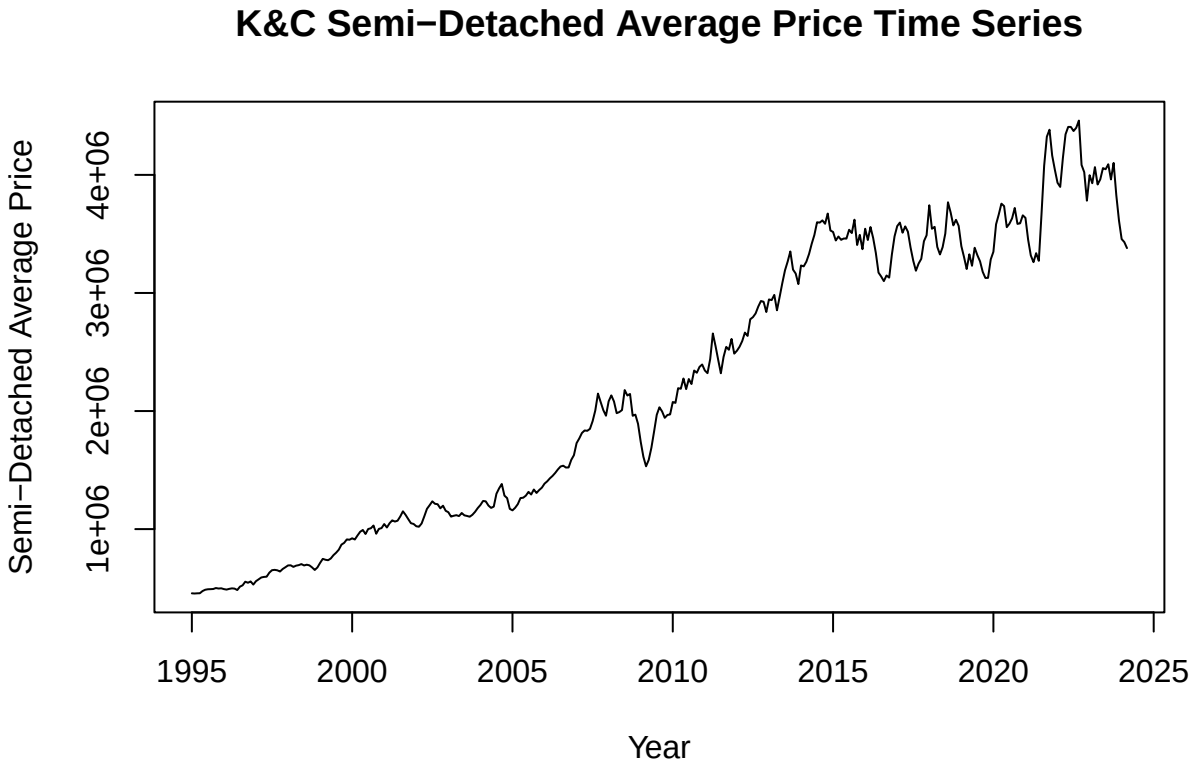
```r
# Plot all combined series together for Croydon Area
plot(combined_mean_price_croydon_test, type = "l", col = "blue",
     main = "Croydon Combined Mean Prices Comparison",
     ylab = "Combined Mean Price", xlab = "Year",
     ylim = range(c(combined_mean_price_croydon_test,
                    combined_mean_price_croydon_less_MSE_test_ts,
                    combined_mean_price_croydon_less_MAE_test_ts)))
lines(combined_mean_price_croydon_less_MSE_test_ts, col = "red")
lines(combined_mean_price_croydon_less_MAE_test_ts, col = "black")

legend("topright", legend = c("Combined Mean Price", "Less MSE", "Less MAE"),
       col = c("blue", "red", "black"), lty = 1, cex = 0.8)
```



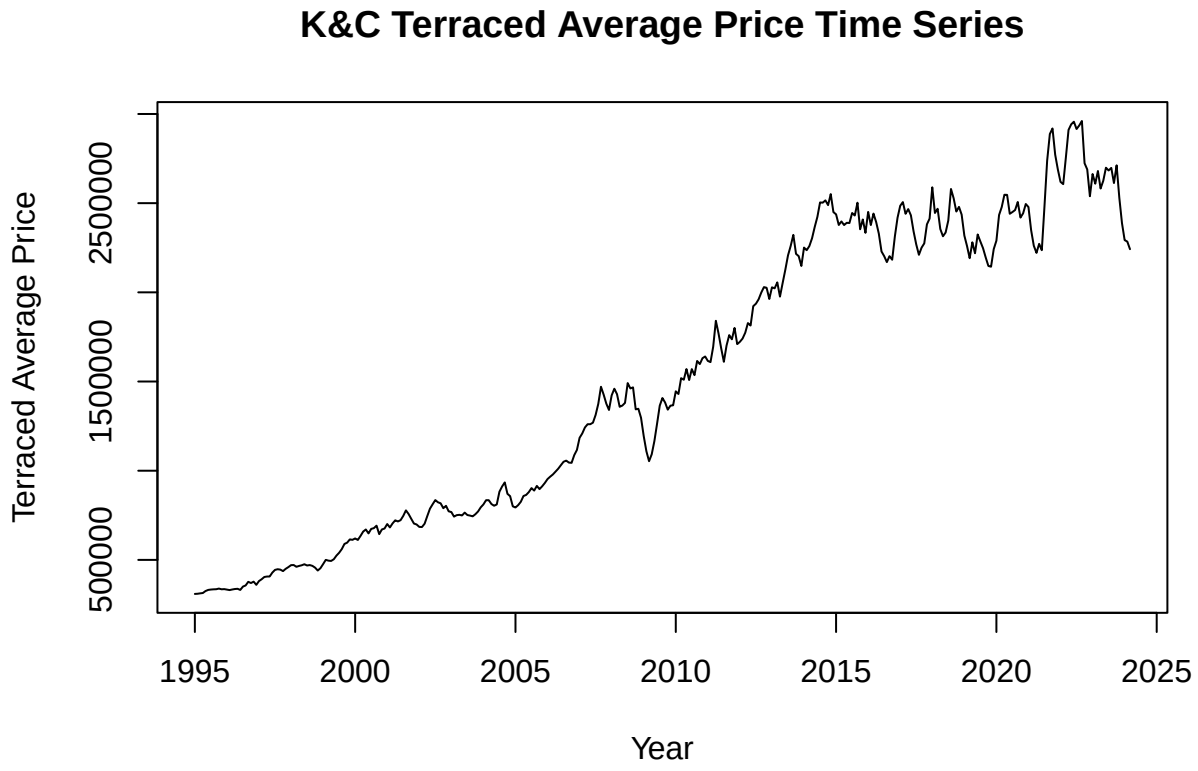

```
### Kensington and Chelsea Area
# Load the data of Kensington and Chelsea Area
kc_data <- read.csv("~/Desktop/Updated_KensingtonandChelsea_df.csv")

# Convert the Date column to Date type of Kensington and Chelsea Area
kc_data$Date <- as.Date(kc_data$Date, format="%Y-%m-%d")

# Extract the Average Price for each property type and Date columns
# In Kensington and Chelsea Area
kc_detached_price <- kc_data$Detached_Average_Price
kc_semi_detached_price <- kc_data$Semi_Detached_Average_Price
kc_terraced_price <- kc_data$Terraced_Average_Price
kc_flat_price <- kc_data$Flat_Average_Price
kc_dates <- kc_data$Date

# Create time series object for each property type in Kensington and Chelsea
kc_detached_ts <- ts(kc_detached_price, start = c(1995, 1), frequency = 12)
kc_semi_detached_ts <- ts(kc_semi_detached_price, start = c(1995, 1),
                          frequency = 12)
kc_terraced_ts <- ts(kc_terraced_price, start = c(1995, 1), frequency = 12)
kc_flat_ts <- ts(kc_flat_price, start = c(1995, 1), frequency = 12)

# Plot the time series of Kensington and Chelsea Area
plot(kc_detached_ts, main = "K&C Detached Average Price Time Series",
     ylab = "Detached Average Price", xlab = "Year")
```



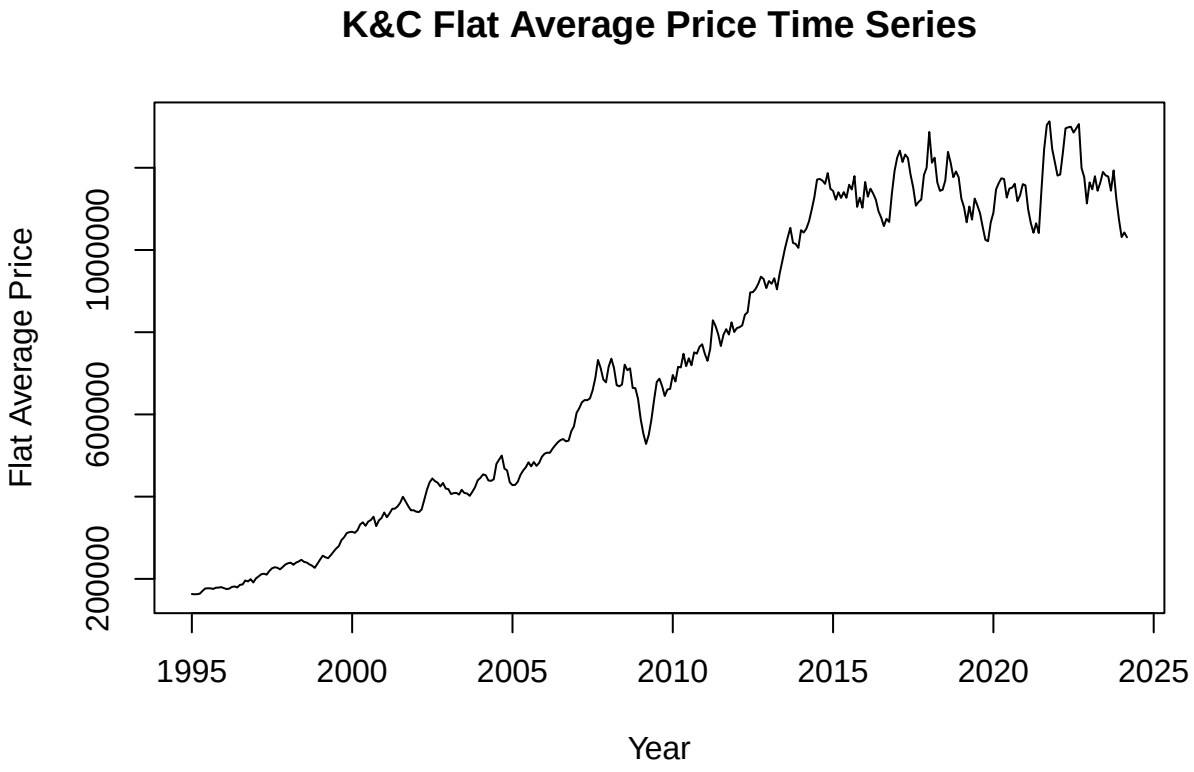

```
plot(kc_semi_detached_ts, main = "K&C Semi-Detached Average Price Time Series",
     ylab = "Semi-Detached Average Price", xlab = "Year")
```



```
plot(kc_terraced_ts, main = "K&C Terraced Average Price Time Series",
     ylab = "Terraced Average Price", xlab = "Year")
```




```
plot(kc_flat_ts, main = "K&C Flat Average Price Time Series",
     ylab = "Flat Average Price", xlab = "Year")
```




```
# Calculate the p-value for the original series
# By using ADF test for each property type of Kensington and Chelsea Area
adf_test_kc_d <- adf.test(kc_detached_ts)
adf_test_kc_sd <- adf.test(kc_semi_detached_ts)
adf_test_kc_t <- adf.test(kc_terraced_ts)
adf_test_kc_f <- adf.test(kc_flat_ts)
p_value_kc_d <- adf_test_kc_d$p.value
p_value_kc_sd <- adf_test_kc_sd$p.value
p_value_kc_t <- adf_test_kc_t$p.value
p_value_kc_f <- adf_test_kc_f$p.value
print(paste("Kensington and Chelsea p-value:", p_value_kc_d, p_value_kc_sd,
            p_value_kc_t, p_value_kc_f))
```

```
## [1] "Kensington and Chelsea p-value: 0.789333134209407 0.659399148825699 0.877662677907351 0.9783525
```

```
# Perform first-order differencing of Kensington and Chelsea Area
kc_detached_ts_diff <- diff(kc_detached_ts)
kc_semi_detached_ts_diff <- diff(kc_semi_detached_ts)
kc_terraced_ts_diff <- diff(kc_terraced_ts)
kc_flat_ts_diff <- diff(kc_flat_ts)

# Perform ADF test on differenced series of Kensington and Chelsea Area
adf_test_diff_kc_d <- adf.test(kc_detached_ts_diff)
```

```
## Warning in adf.test(kc_detached_ts_diff): p-value smaller than printed p-value
```

```
adf_test_diff_kc_sd <- adf.test(kc_semi_detached_ts_diff)
```

```
## Warning in adf.test(kc_semi_detached_ts_diff): p-value smaller than printed
## p-value
```

```
adf_test_diff_kc_t <- adf.test(kc_terraced_ts_diff)
```

```
## Warning in adf.test(kc_terraced_ts_diff): p-value smaller than printed p-value
```

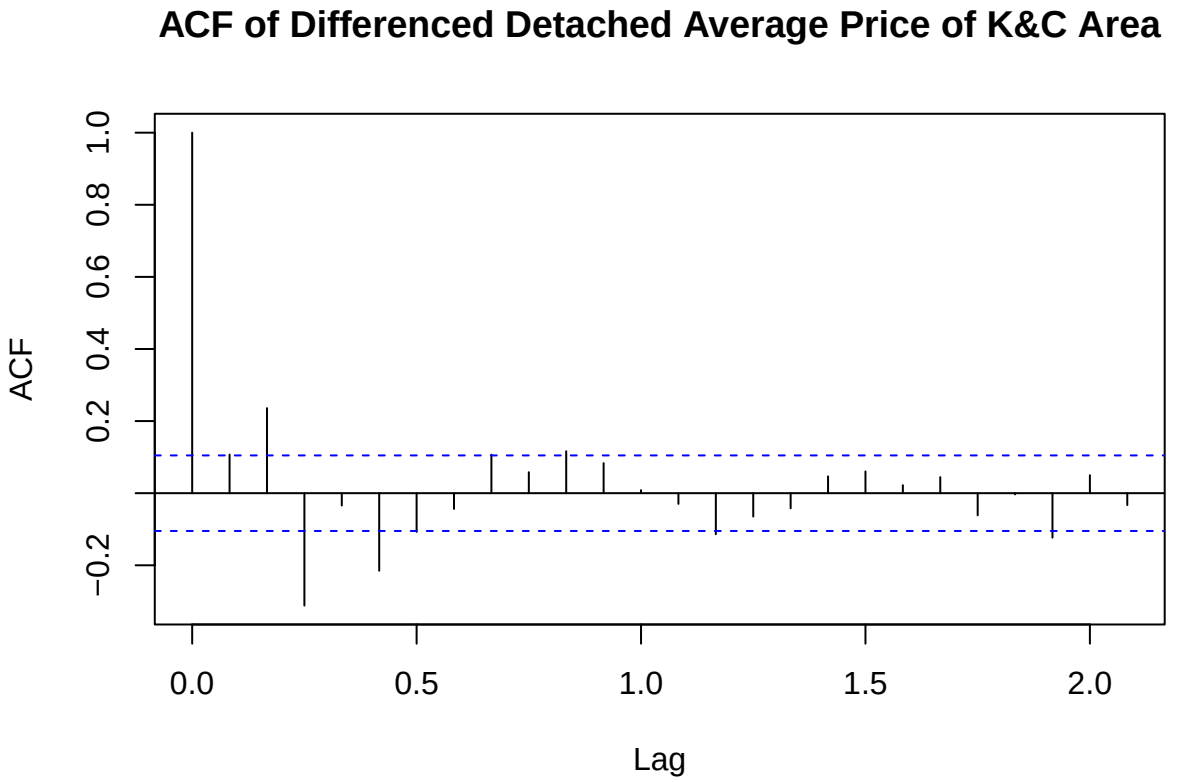
```
adf_test_diff_kc_f <- adf.test(kc_flat_ts_diff)
```

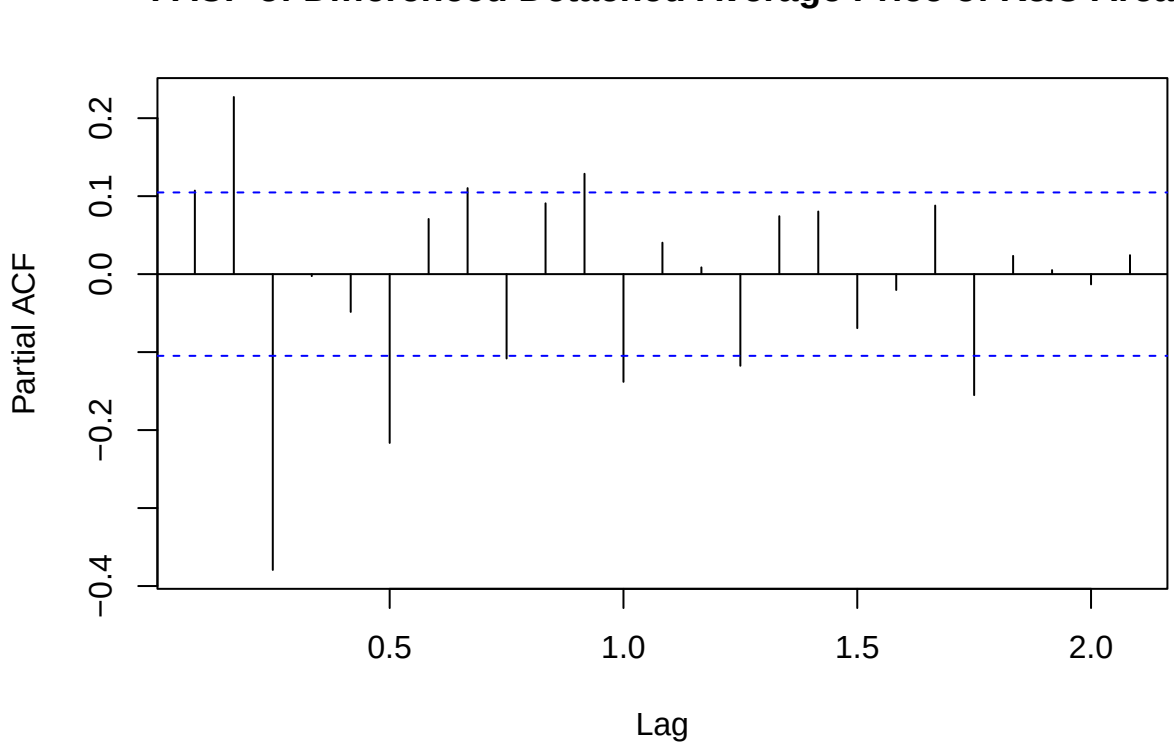
```
## Warning in adf.test(kc_flat_ts_diff): p-value smaller than printed p-value
```

```
differenced_p_value_kc_d <- adf_test_diff_kc_d$p.value
differenced_p_value_kc_sd <- adf_test_diff_kc_sd$p.value
differenced_p_value_kc_t <- adf_test_diff_kc_t$p.value
differenced_p_value_kc_f <- adf_test_diff_kc_f$p.value
print(paste("Kensington and Chelsea difference1_p-value:",
            differenced_p_value_kc_d,
            differenced_p_value_kc_sd,
            differenced_p_value_kc_t,
            differenced_p_value_kc_f))
```

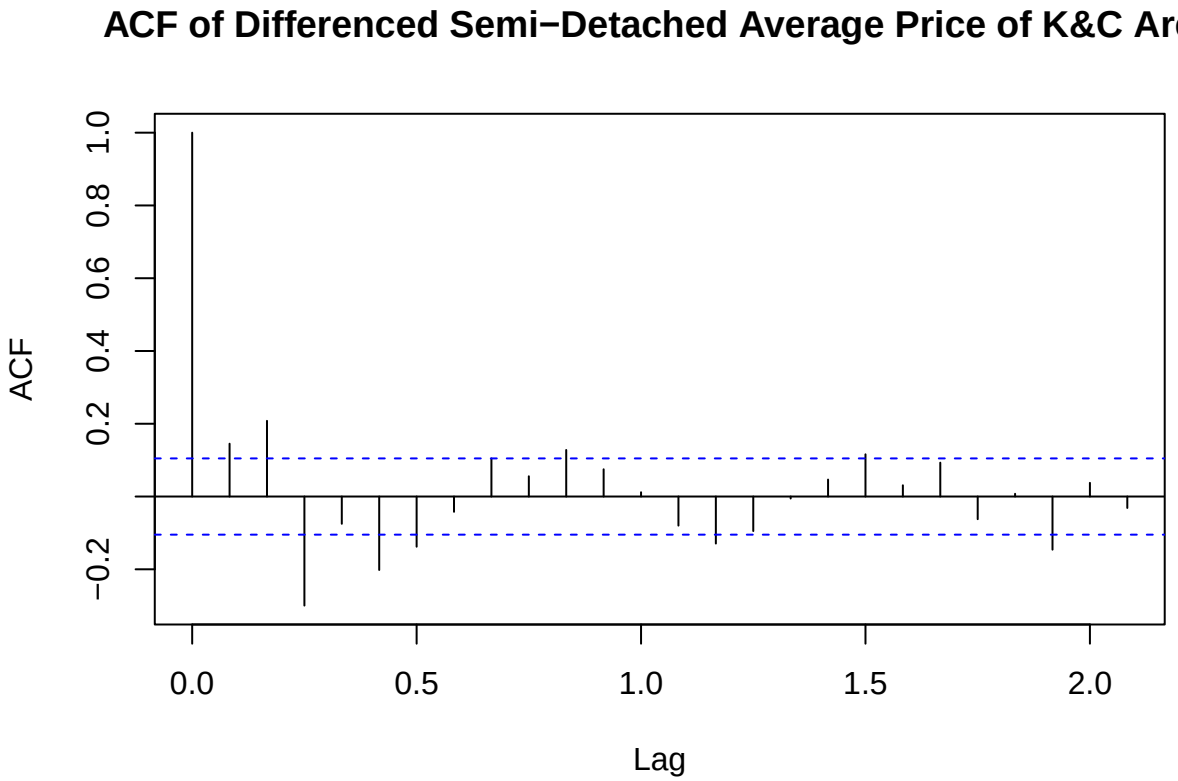
```
## [1] "Kensington and Chelsea difference1_p-value: 0.01 0.01 0.01 0.01"
```

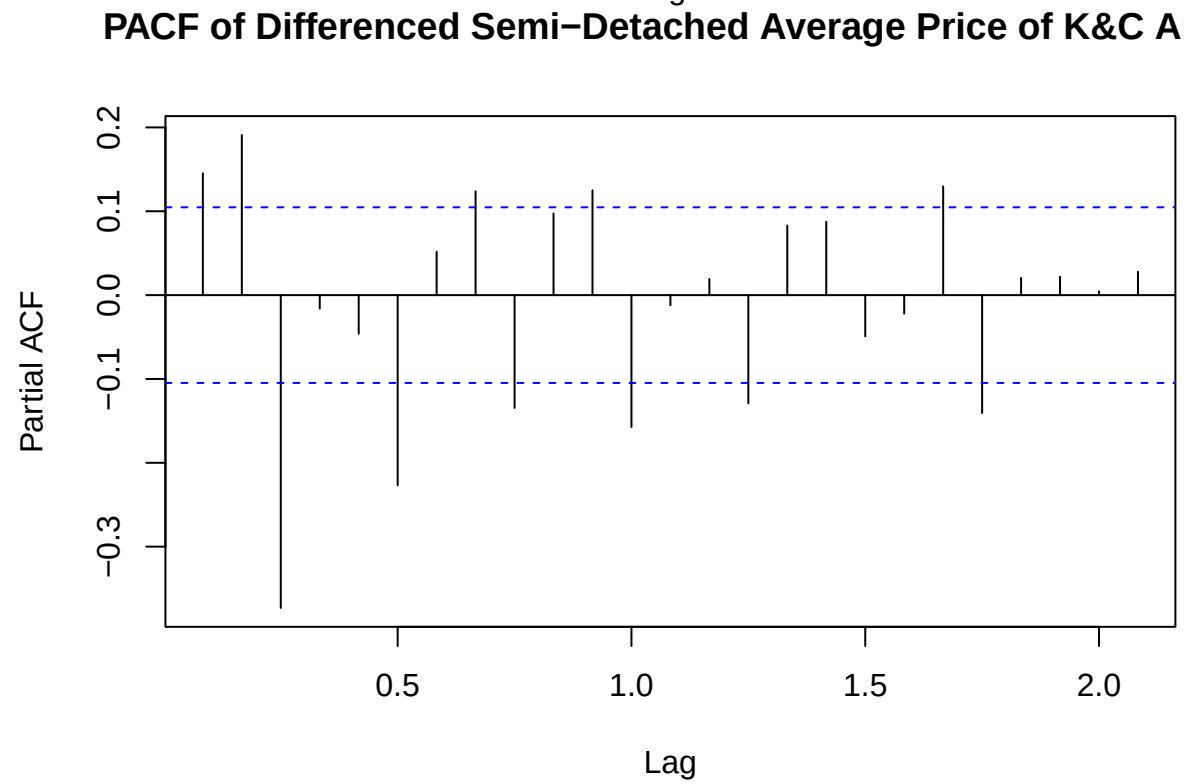
```
# Define function to plot ACF and PACF of Kensington and Chelsea Area
plot_acf_pacf <- function(ts_diff, title) {
  acf(ts_diff, main = paste("ACF of", title))
  pacf(ts_diff, main = paste("PACF of", title))
}

# Plot ACF and PACF for differenced series for each property type
# In Kensington and Chelsea
plot_acf_pacf(kc_detached_ts_diff,
              "Differenced Detached Average Price of K&C Area")
```



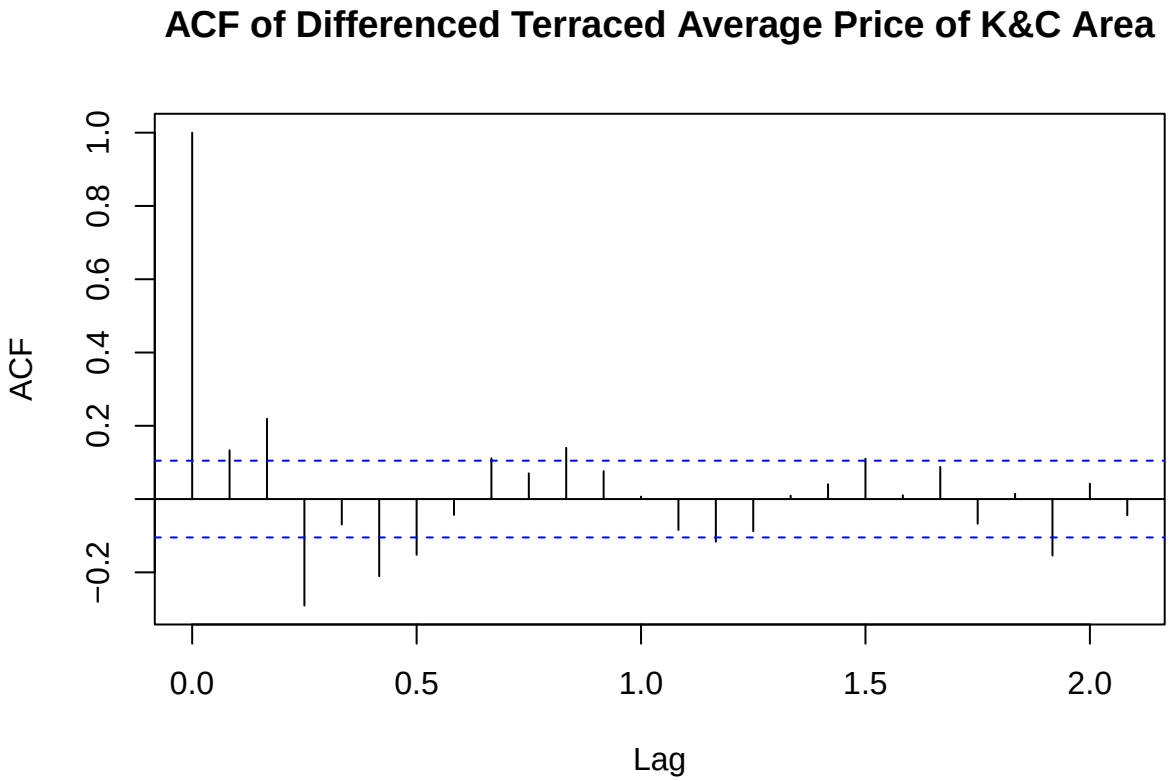


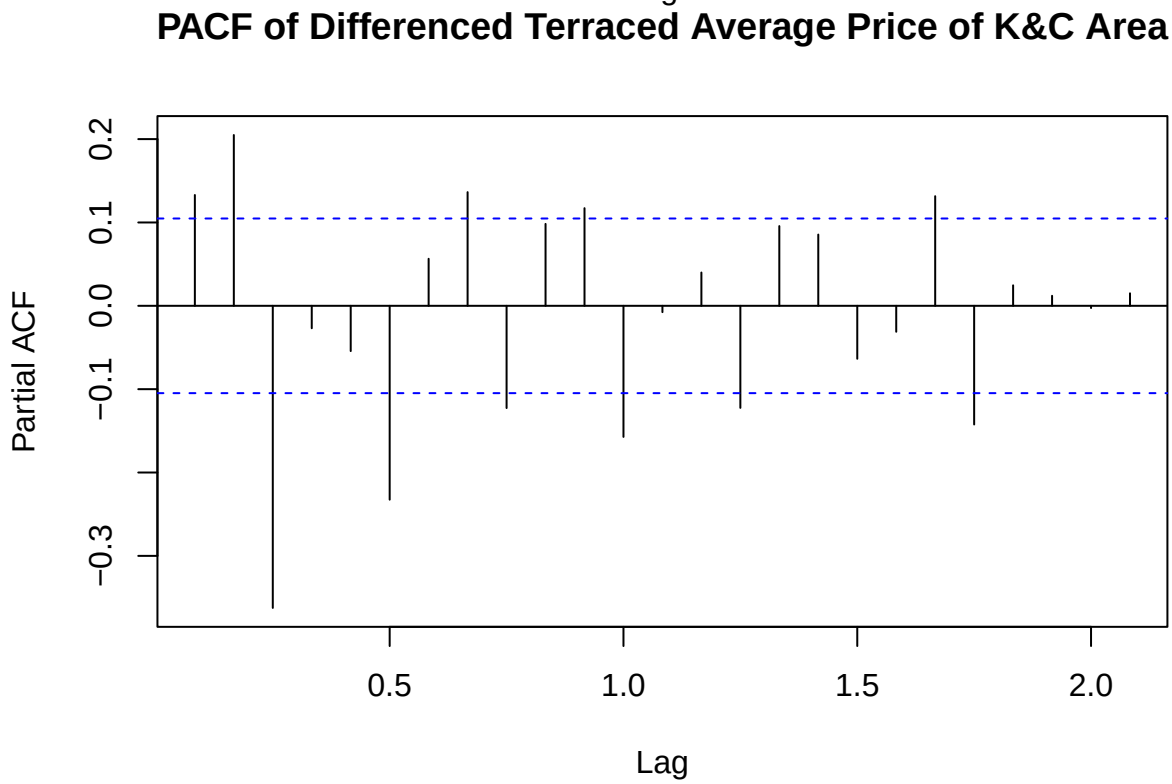

```
plot_acf_pacf(kc_semi_detached_ts_diff,
              "Differenced Semi-Detached Average Price of K&C Area")
```



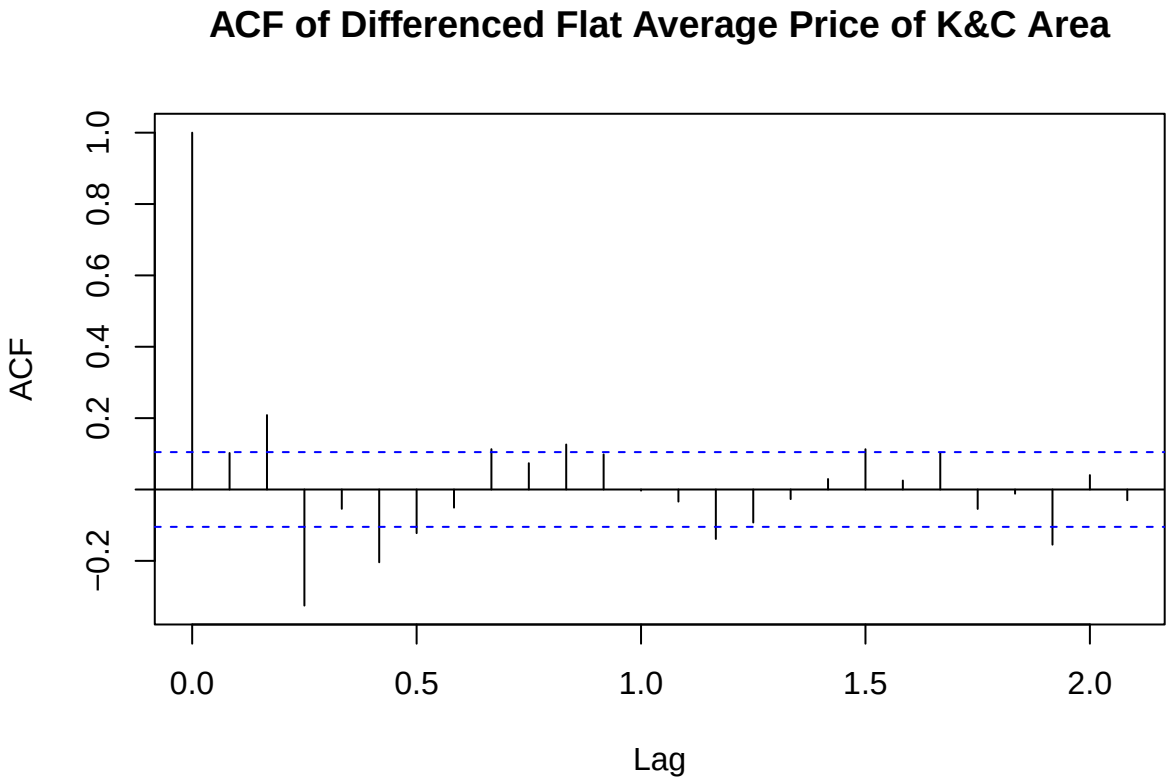


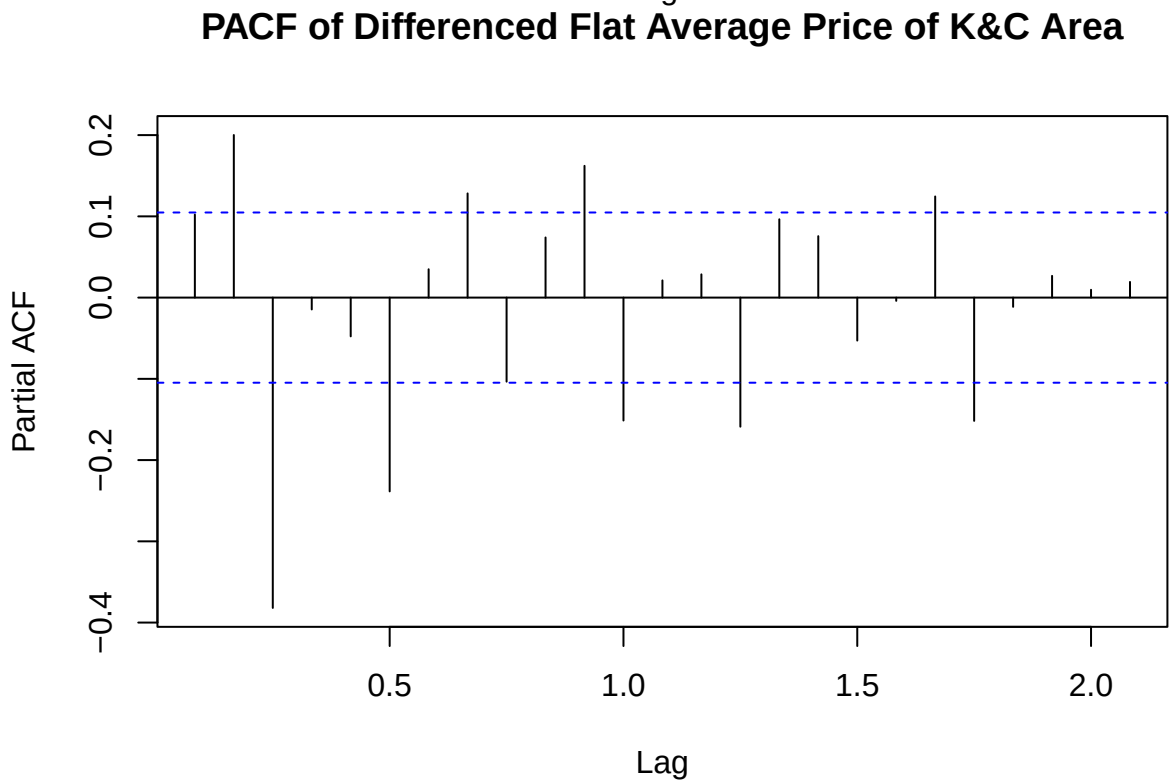

```
plot_acf_pacf(kc_terraced_ts_diff,
              "Differenced Terraced Average Price of K&C Area")
```






```
plot_acf_pacf(kc_flat_ts_diff, "Differenced Flat Average Price of K&C Area")
```






```
# Fit ARIMA models for Kensington and Chelsea Area
fit_arima_kc_d <- auto.arima(kc_detached_ts_diff, stepwise = FALSE,
                             approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 8962.552
##  ARIMA(0,0,0)            with non-zero mean : 8961.922
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 8964.464
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 8963.932
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 8965.157
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 8964.982
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 8964.45
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 8963.93
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 8965.069
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 8964.962
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 8961.567
##  ARIMA(0,0,1)            with non-zero mean : 8961.284
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 8963.541
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 8963.322
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 8963.62
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 8963.812
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 8963.532
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 8963.321
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 8963.515
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 8963.778
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 8922.995
##  ARIMA(0,0,2)            with non-zero mean : 8923.994
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 8925.04
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 8926.044
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 8925.062
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 8926.272
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 8925.04
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 8926.043
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : 8927.736
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 8924.829
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 8926.074
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)            with zero mean     : Inf
```

```
##  ARIMA(0,0,3)            with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)            with zero mean     : Inf
##  ARIMA(0,0,4)            with non-zero mean : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,5)            with zero mean     : Inf
##  ARIMA(0,0,5)            with non-zero mean : Inf
##  ARIMA(1,0,0)            with zero mean     : 8959.992
##  ARIMA(1,0,0)            with non-zero mean : 8959.919
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 8961.988
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 8961.961
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 8961.709
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 8962.117
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 8961.982
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 8961.961
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 8961.584
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 8962.062
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 8959.255
##  ARIMA(1,0,1)            with non-zero mean : 8959.491
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 8961.274
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 8961.546
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 8960.95
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 8961.607
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 8961.269
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 8961.546
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 8960.818
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 8961.537
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : Inf
```

```
##  ARIMA(1,0,2)            with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)            with zero mean     : Inf
##  ARIMA(1,0,3)            with non-zero mean : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : Inf
##  ARIMA(2,0,0)            with zero mean     : 8942.689
##  ARIMA(2,0,0)            with non-zero mean : 8943.454
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 8944.695
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 8945.498
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 8945.393
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 8946.407
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 8944.69
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 8945.497
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 8945.293
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 8946.341
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 8918.016
##  ARIMA(2,0,1)            with non-zero mean : 8918.691
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 8920.068
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 8920.732
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 8921.253
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 8922.103
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 8920.068
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 8920.729
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : 8922.666
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 8921.176
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 8922.058
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
```

```
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 8892.333
##  ARIMA(3,0,0)            with non-zero mean : 8891.283
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 8894.381
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 8893.344
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 8895.571
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 8894.906
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 8894.38
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 8893.344
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 8895.426
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 8894.838
##  ARIMA(3,0,1)            with zero mean     : 8894.386
##  ARIMA(3,0,1)            with non-zero mean : 8893.329
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 8896.447
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 8895.406
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 8896.446
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 8895.406
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : 8892.602
##  ARIMA(4,0,0)            with zero mean     : 8894.387
##  ARIMA(4,0,0)            with non-zero mean : 8893.337
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 8896.448
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 8895.413
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 8896.447
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 8895.413
##  ARIMA(4,0,1)            with zero mean     : 8896.45
##  ARIMA(4,0,1)            with non-zero mean : Inf
##  ARIMA(5,0,0)            with zero mean     : 8895.803
##  ARIMA(5,0,0)            with non-zero mean : 8894.334
##
##
##
##  Best model: ARIMA(3,0,0)            with non-zero mean
```

```
fit_arima_kc_sd <- auto.arima(kc_semi_detached_ts_diff, stepwise = FALSE,
                              approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 8984.021
##  ARIMA(0,0,0)            with non-zero mean : 8983.045
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 8985.876
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 8985.032
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 8987.051
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 8986.513
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 8985.86
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 8985.029
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 8986.996
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 8986.515
```

```
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 8980.463
##  ARIMA(0,0,1)            with non-zero mean : 8979.99
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 8982.343
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 8981.981
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 8982.658
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 8982.7
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 8982.322
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 8981.975
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 8982.597
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 8982.706
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 8942.882
##  ARIMA(0,0,2)            with non-zero mean : 8943.802
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 8944.867
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 8945.826
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 8945.328
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 8946.473
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 8944.859
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 8945.821
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 8945.189
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 8946.368
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)            with zero mean     : Inf
##  ARIMA(0,0,3)            with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)            with zero mean     : Inf
##  ARIMA(0,0,4)            with non-zero mean : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : Inf
```

```
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,5)            with zero mean     : Inf
##  ARIMA(0,0,5)            with non-zero mean : Inf
##  ARIMA(1,0,0)            with zero mean     : 8977.801
##  ARIMA(1,0,0)            with non-zero mean : 8977.63
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 8979.688
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 8979.616
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 8979.551
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 8979.907
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 8979.666
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 8979.608
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 8979.476
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 8979.9
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 8977.402
##  ARIMA(1,0,1)            with non-zero mean : 8977.506
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 8979.307
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 8979.5
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 8979.282
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 8979.843
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 8979.285
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 8979.491
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 8979.207
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 8979.826
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : Inf
##  ARIMA(1,0,2)            with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)            with zero mean     : Inf
##  ARIMA(1,0,3)            with non-zero mean : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : Inf
```

```
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : Inf
##  ARIMA(2,0,0)            with zero mean     : 8966.14
##  ARIMA(2,0,0)            with non-zero mean : 8966.72
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 8968.055
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 8968.707
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 8969.114
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 8969.96
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 8968.042
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 8968.7
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 8969.056
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 8969.933
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 8944.788
##  ARIMA(2,0,1)            with non-zero mean : 8945.308
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 8946.843
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 8947.376
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 8948.486
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 8949.16
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 8946.843
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 8947.376
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 8948.465
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 8949.157
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 8917.821
##  ARIMA(3,0,0)            with non-zero mean : 8916.455
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 8919.816
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 8918.525
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 8921.288
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 8920.308
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 8919.811
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 8918.525
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 8921.203
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 8920.283
##  ARIMA(3,0,1)            with zero mean     : 8919.843
##  ARIMA(3,0,1)            with non-zero mean : 8918.29
```

```
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 8921.83
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 8920.359
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 8921.823
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 8920.358
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : 8916.858
##  ARIMA(4,0,0)            with zero mean     : 8919.853
##  ARIMA(4,0,0)            with non-zero mean : 8918.382
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 8921.846
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 8920.459
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 8921.839
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 8920.458
##  ARIMA(4,0,1)            with zero mean     : Inf
##  ARIMA(4,0,1)            with non-zero mean : Inf
##  ARIMA(5,0,0)            with zero mean     : 8921.37
##  ARIMA(5,0,0)            with non-zero mean : 8919.473
##
##
##
##  Best model: ARIMA(3,0,0)            with non-zero mean
```

```
fit_arima_kc_t <- auto.arima(kc_terraced_ts_diff, stepwise = FALSE,
                         approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 8694.927
##  ARIMA(0,0,0)            with non-zero mean : 8693.957
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 8696.851
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 8695.977
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 8697.872
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 8697.318
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 8696.84
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 8695.975
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 8697.803
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 8697.324
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 8692.283
##  ARIMA(0,0,1)            with non-zero mean : 8691.78
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 8694.222
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 8693.805
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 8694.321
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 8694.323
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 8694.209
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 8693.802
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
```

```
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 8694.219
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 8694.316
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 8655.682
##  ARIMA(0,0,2)            with non-zero mean : 8656.548
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 8657.714
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 8658.604
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 8657.84
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 8658.942
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 8657.712
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 8658.603
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 8657.647
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 8658.796
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)            with zero mean     : Inf
##  ARIMA(0,0,3)            with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)            with zero mean     : Inf
##  ARIMA(0,0,4)            with non-zero mean : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,5)            with zero mean     : Inf
##  ARIMA(0,0,5)            with non-zero mean : Inf
##  ARIMA(1,0,0)            with zero mean     : 8689.976
##  ARIMA(1,0,0)            with non-zero mean : 8689.755
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 8691.923
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 8691.778
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 8691.523
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 8691.828
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 8691.909
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 8691.774
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
```

```
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 8691.395
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 8691.797
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 8689.323
##  ARIMA(1,0,1)            with non-zero mean : 8689.407
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 8691.298
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 8691.444
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 8691.028
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 8691.554
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 8691.287
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 8691.441
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 8690.909
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 8691.516
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : Inf
##  ARIMA(1,0,2)            with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)            with zero mean     : Inf
##  ARIMA(1,0,3)            with non-zero mean : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : Inf
##  ARIMA(2,0,0)            with zero mean     : 8676.246
##  ARIMA(2,0,0)            with non-zero mean : 8676.836
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 8678.268
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 8678.89
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 8679.37
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 8680.179
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 8678.265
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 8678.89
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
```

```
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 8679.316
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 8680.16
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 8655.741
##  ARIMA(2,0,1)            with non-zero mean : 8656.252
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 8657.78
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 8658.263
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 8659.398
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 8660.025
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 8657.779
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 8658.259
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : 8660.312
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 8659.398
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 8660.044
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 8630.979
##  ARIMA(3,0,0)            with non-zero mean : 8629.743
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 8633.02
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 8631.808
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 8634.268
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 8633.4
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 8633.019
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 8631.807
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 8634.144
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 8633.359
##  ARIMA(3,0,1)            with zero mean     : 8632.835
##  ARIMA(3,0,1)            with non-zero mean : 8631.069
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 8634.854
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 8633.136
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 8634.848
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 8633.136
##  ARIMA(3,0,2)            with zero mean     : 8632.699
##  ARIMA(3,0,2)            with non-zero mean : 8629.314
##  ARIMA(4,0,0)            with zero mean     : 8632.909
##  ARIMA(4,0,0)            with non-zero mean : 8631.492
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 8634.945
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 8633.575
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 8634.942
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 8633.575
##  ARIMA(4,0,1)            with zero mean     : Inf
##  ARIMA(4,0,1)            with non-zero mean : 8629.721
##  ARIMA(5,0,0)            with zero mean     : 8634.198
```

```
##  ARIMA(5,0,0)            with non-zero mean : 8632.287
##
##
##
##  Best model: ARIMA(3,0,2)            with non-zero mean
```

```
fit_arima_kc_f <- auto.arima(kc_flat_ts_diff, stepwise = FALSE,
                         approximation = FALSE, trace = TRUE)
```

```
##
##  ARIMA(0,0,0)            with zero mean     : 8119.486
##  ARIMA(0,0,0)            with non-zero mean : 8118.385
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : 8121.491
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 8120.416
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : 8122.475
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 8121.749
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : 8121.489
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 8120.416
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : 8122.518
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 8121.827
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)            with zero mean     : 8118.6
##  ARIMA(0,0,1)            with non-zero mean : 8117.9
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : 8120.634
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 8119.928
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : 8120.916
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 8120.651
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : 8120.634
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 8119.926
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : 8121.916
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : 8121.031
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 8120.8
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)            with zero mean     : 8084.096
##  ARIMA(0,0,2)            with non-zero mean : 8084.911
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 8086.057
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 8086.843
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : 8086.034
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 8087.058
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : 8086.043
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 8086.824
```

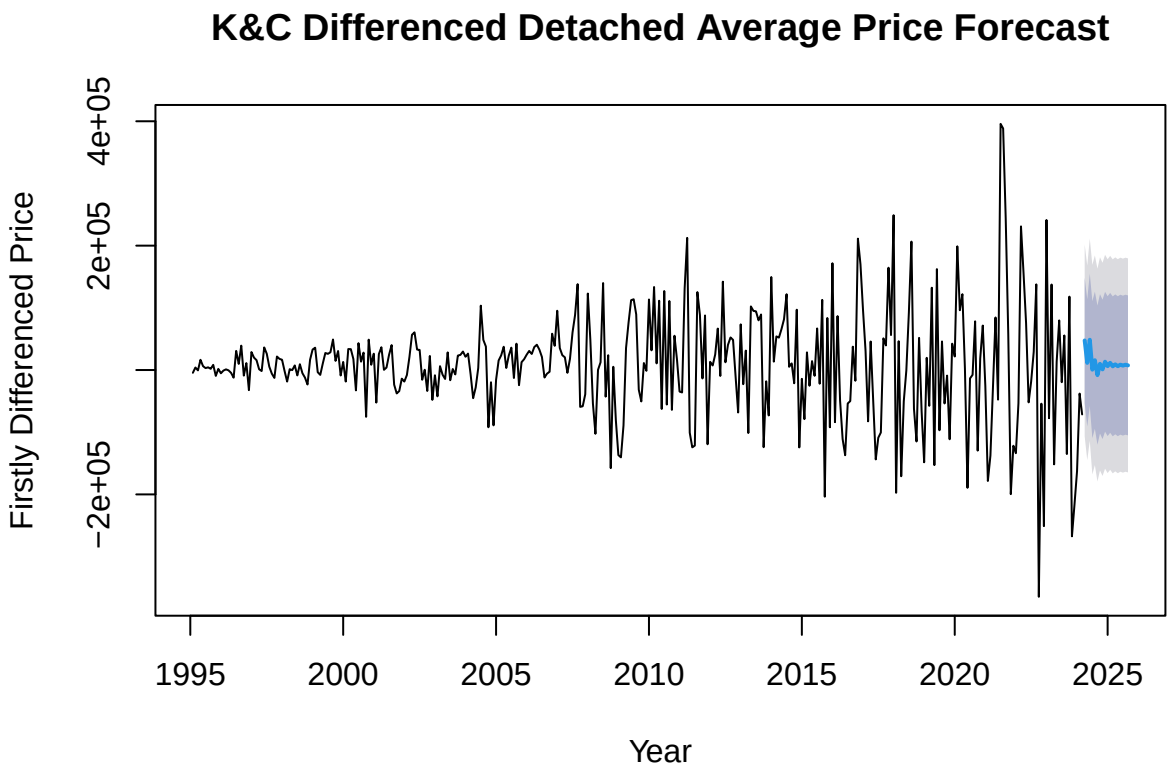
```
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : 8087.724
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : 8088.523
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : 8085.997
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 8087.046
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)            with zero mean     : Inf
##  ARIMA(0,0,3)            with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)            with zero mean     : Inf
##  ARIMA(0,0,4)            with non-zero mean : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,5)            with zero mean     : Inf
##  ARIMA(0,0,5)            with non-zero mean : Inf
##  ARIMA(1,0,0)            with zero mean     : 8117.231
##  ARIMA(1,0,0)            with non-zero mean : 8116.751
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 8119.265
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 8118.767
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 8119.181
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 8119.162
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 8119.265
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 8118.764
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : 8120.719
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 8119.327
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 8119.339
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : 8117.208
##  ARIMA(1,0,1)            with non-zero mean : 8117.013
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : 8119.252
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 8119.038
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : 8119.211
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 8119.42
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : 8119.251
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 8119.034
```

```
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : 8120.988
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 8119.349
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 8119.588
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : Inf
##  ARIMA(1,0,2)            with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)            with zero mean     : Inf
##  ARIMA(1,0,3)            with non-zero mean : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : Inf
##  ARIMA(2,0,0)            with zero mean     : 8104.179
##  ARIMA(2,0,0)            with non-zero mean : 8104.561
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 8106.226
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 8106.609
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 8107.405
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 8107.993
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 8106.226
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 8106.608
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : 8108.643
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 8107.436
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 8108.037
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : 8081.201
##  ARIMA(2,0,1)            with non-zero mean : 8081.516
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : 8083.074
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 8083.296
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : 8084.773
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 8085.142
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 8083.062
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 8083.281
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : 8085.068
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : 8085.321
```

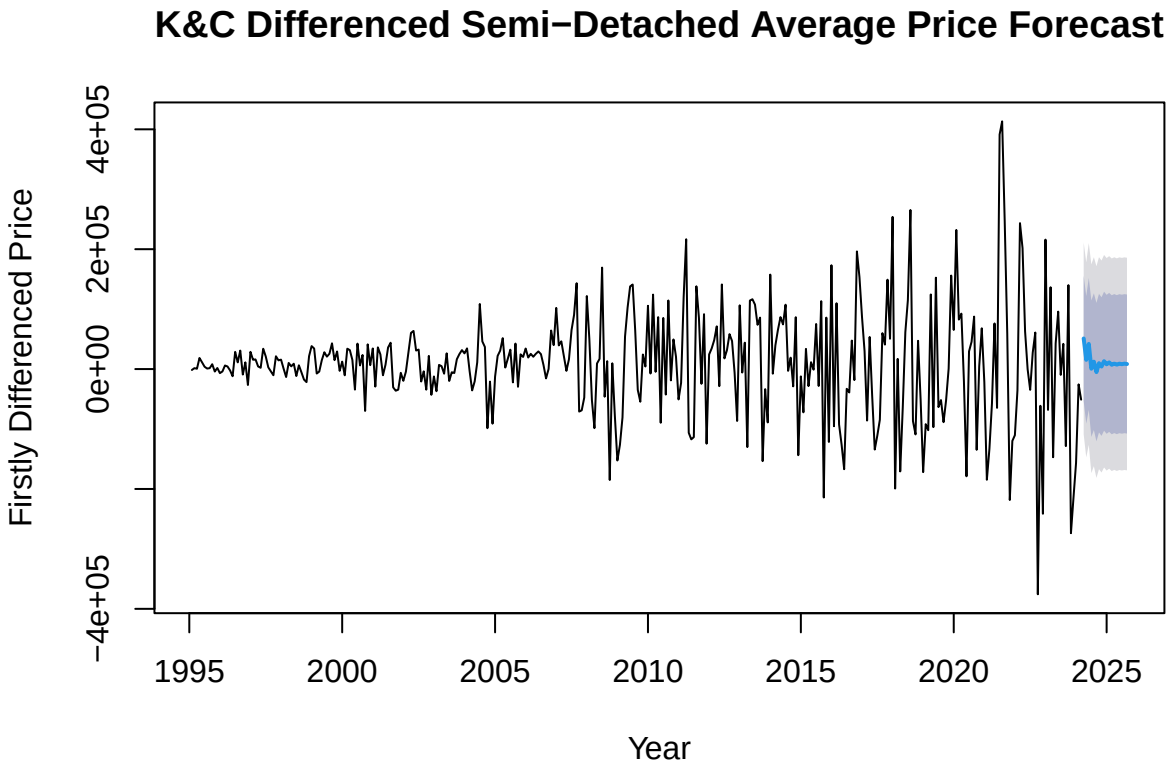
```
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 8084.812
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 8085.18
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3)            with zero mean     : Inf
##  ARIMA(2,0,3)            with non-zero mean : Inf
##  ARIMA(3,0,0)            with zero mean     : 8053.053
##  ARIMA(3,0,0)            with non-zero mean : 8051.258
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 8055.042
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 8053.077
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 8056.343
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 8054.766
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 8055.035
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 8053.06
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : 8056.953
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : 8055.028
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 8056.313
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 8054.779
##  ARIMA(3,0,1)            with zero mean     : 8055.098
##  ARIMA(3,0,1)            with non-zero mean : 8053.155
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : 8057.109
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 8055.059
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 8057.103
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 8055.045
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : 8051.717
##  ARIMA(4,0,0)            with zero mean     : 8055.101
##  ARIMA(4,0,0)            with non-zero mean : 8053.218
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : 8057.11
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 8055.094
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 8057.103
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 8055.08
##  ARIMA(4,0,1)            with zero mean     : Inf
##  ARIMA(4,0,1)            with non-zero mean : Inf
##  ARIMA(5,0,0)            with zero mean     : 8056.652
##  ARIMA(5,0,0)            with non-zero mean : 8054.335
##
##
##
##  Best model: ARIMA(3,0,0)            with non-zero mean
```

```
# Forecast using the ARIMA model for each property type
# In Kensington and Chelsea Area
forecasted_values_arima_kc_d <- forecast(fit_arima_kc_d, h = 18)
forecasted_values_arima_kc_sd <- forecast(fit_arima_kc_sd, h = 18)
forecasted_values_arima_kc_t <- forecast(fit_arima_kc_t, h = 18)
forecasted_values_arima_kc_f <- forecast(fit_arima_kc_f, h = 18)

# Plot the differenced forecast value for each property type
# In Kensington and Chelsea
```

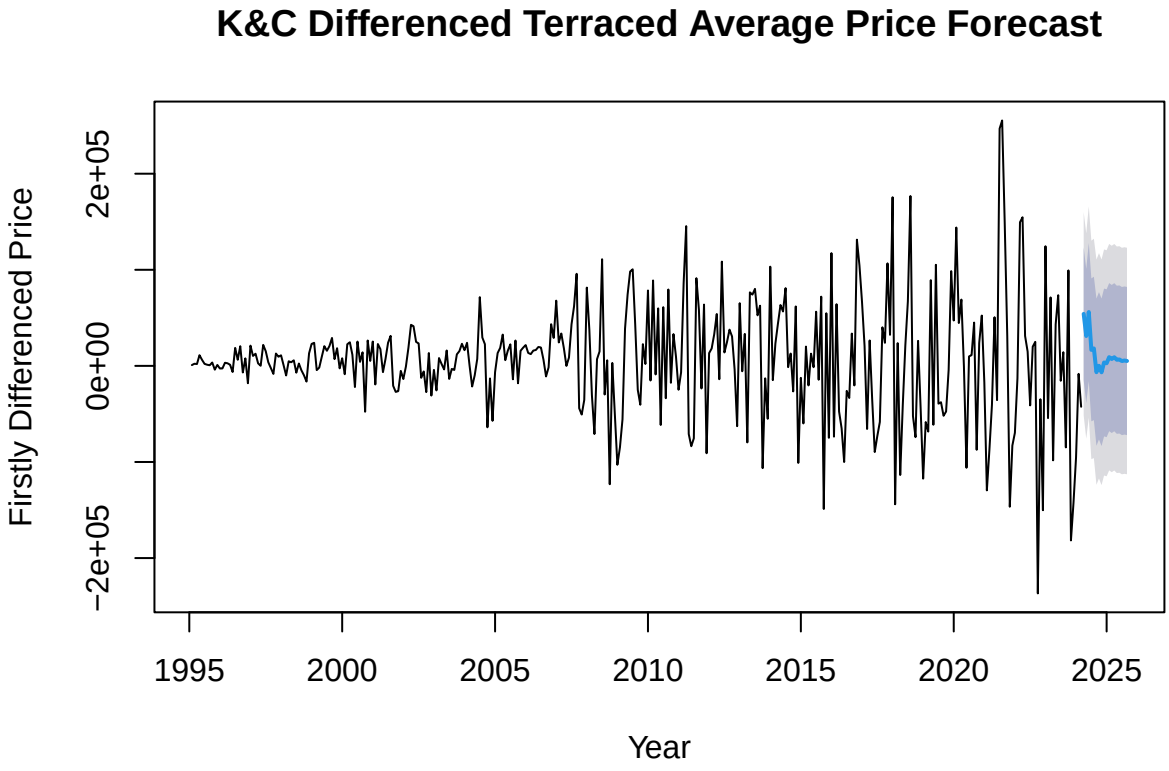
```
plot(forecasted_values_arima_kc_d,
     main = "K&C Differenced Detached Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```



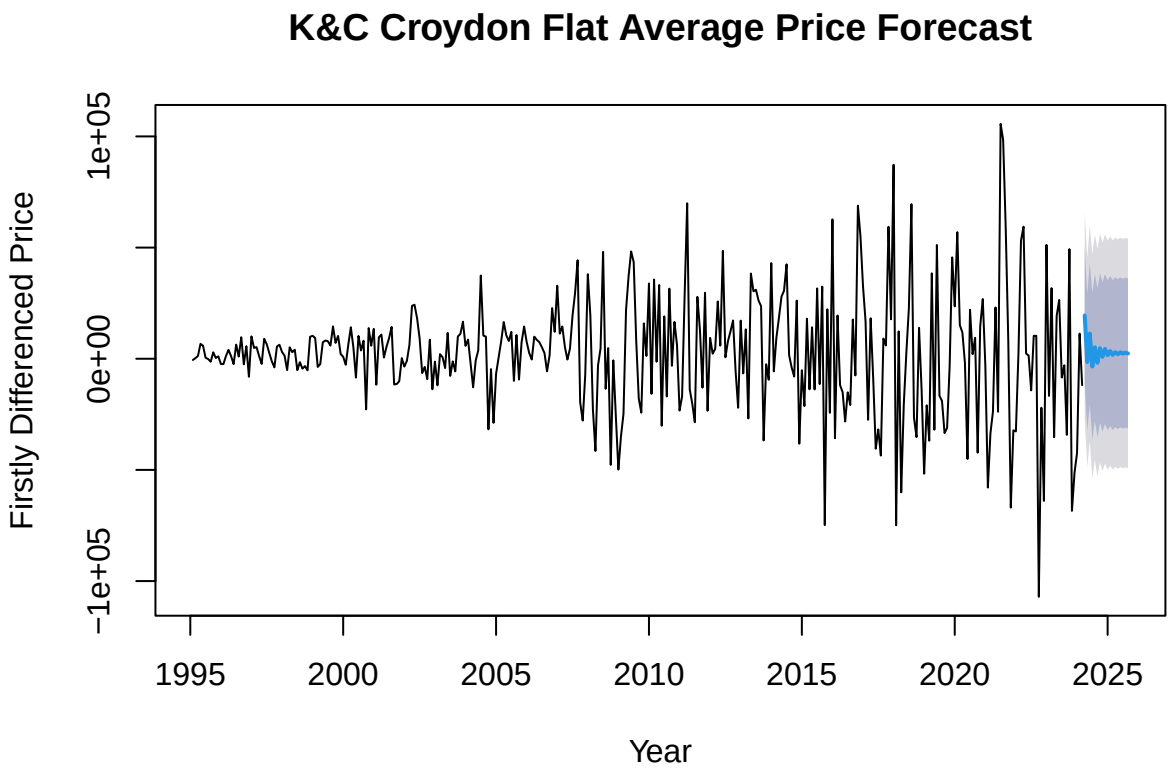

```
plot(forecasted_values_arima_kc_sd,
     main = "K&C Differenced Semi-Detached Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```




```
plot(forecasted_values_arima_kc_t,
     main = "K&C Differenced Terraced Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```



```
plot(forecasted_values_arima_kc_f,
     main = "K&C Croydon Flat Average Price Forecast",
     ylab = "Firstly Differenced Price", xlab = "Year")
```




```
# Summary of the model for four different type of properties
# In Kensington and Chelsea
# For Detached of Kensington and Chelsea Area
print(forecasted_values_arima_kc_d)
```

```
##          Point Forecast      Lo 80    Hi 80     Lo 95    Hi 95
## Apr 2024     47499.7497  -53345.94 148345.4 -106730.4 201729.9
## May 2024     11482.8000  -90773.59 113739.2 -144904.9 167870.5
## Jun 2024     48691.5231  -57530.56 154913.6 -113761.1 211144.2
## Jul 2024       593.1166 -109537.04 110723.3 -167836.4 169022.7
## Aug 2024     15742.8616  -94456.80 125942.5 -152793.0 184278.7
## Sep 2024     -8187.4323 -120019.04 103644.2 -179219.1 162844.3
## Oct 2024      9933.6240 -102106.32 121973.6 -161416.7 181283.9
## Nov 2024      1079.6126 -110983.14 113142.4 -170305.6 172464.8
## Dec 2024     13327.0818  -99064.87 125719.0 -158561.6 185215.7
## Jan 2025      6233.9650 -106171.28 118639.2 -165675.0 178143.0
## Feb 2025     11548.9503 -100890.57 123988.5 -160412.5 183510.4
## Mar 2025      5972.6968 -106513.46 118458.9 -166060.0 178005.4
## Apr 2025      9093.0255 -103396.16 121582.2 -162944.3 181130.4
## May 2025      6167.8330 -106334.92 118670.6 -165890.3 178225.9
## Jun 2025      8593.8331 -103914.77 121102.4 -163473.2 180660.9
## Jul 2025      7065.7808 -105444.26 119575.8 -165003.5 179135.0
## Aug 2025      8542.2639 -103971.08 121055.6 -163532.0 180616.6
## Sep 2025      7477.2099 -105036.98 119991.4 -164598.4 179552.8
```

```
summary(fit_arima_kc_d)
```

```
## Series: kc_detached_ts_diff
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2      ar3      mean
##       0.1678  0.2570  -0.3793  7844.422
## s.e.  0.0493  0.0482   0.0495  4387.469
##
## sigma^2 = 6.192e+09:  log likelihood = -4440.55
## AIC=8891.11   AICc=8891.28   BIC=8910.4
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -20.21112 78239.36 52211.78 102.1582 273.3586 0.6253291
##                      ACF1
## Training set -0.001850832
```

```
# For Semi-Detached of Kensington and Chelsea Area
print(forecasted_values_arima_kc_sd)
```

```
##          Point Forecast      Lo 80    Hi 80     Lo 95    Hi 95
## Apr 2024     51062.8851  -53481.01 155606.8 -108823.2 210949.0
## May 2024     15350.4654  -91019.83 121720.8 -147328.9 178029.8
## Jun 2024     41979.2391  -68037.14 151995.6 -126276.3 210234.8
## Jul 2024       584.2136 -113219.96 114388.4 -173464.2 174632.7
## Aug 2024     12341.5417 -101632.81 126315.9 -161967.2 186650.3
## Sep 2024     -5040.6344 -120496.48 110415.2 -181615.1 171533.8
## Oct 2024      9871.7062 -105736.81 125480.2 -166936.3 186679.7
## Nov 2024      4230.9202 -111380.86 119842.7 -172582.0 181043.9
## Dec 2024     13132.6068 -102762.31 129027.5 -164113.4 190378.6
## Jan 2025      7928.0806 -107970.30 123826.5 -169323.2 185179.4
## Feb 2025     11131.6776 -104784.75 127048.1 -166147.2 188410.6
## Mar 2025      7199.8734 -108751.42 123151.2 -170132.3 184532.1
## Apr 2025      9149.6705 -106801.96 125101.3 -168183.0 186482.4
## May 2025      7403.6895 -108556.08 123363.5 -169941.5 184748.9
## Jun 2025      8996.7315 -106966.28 124959.7 -168353.4 186346.8
## Jul 2025      8161.4006 -107801.90 124124.7 -169189.2 185512.0
## Aug 2025      9027.3040 -106937.91 124992.5 -168326.2 186380.8
## Sep 2025      8401.0256 -107564.48 124366.5 -168952.9 185755.0
```

```
summary(fit_arima_kc_sd)
```

```
## Series: kc_semi_detached_ts_diff
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2      ar3      mean
##       0.1877  0.2334  -0.3728  8569.609
## s.e.  0.0494  0.0488   0.0496  4562.365
##
```

```
## sigma^2 = 6.655e+09:  log likelihood = -4453.14
## AIC=8916.28   AICc=8916.46   BIC=8935.57
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -22.77273 81108.55 53360.84 63.16122 146.3736 0.6265387
##                      ACF1
## Training set -0.006765575
```

```
# For Terraced of Kensington and Chelsea Area
print(forecasted_values_arima_kc_t)
```

```
##          Point Forecast     Lo 80     Hi 80      Lo 95    Hi 95
## Apr 2024    54101.53089 -15048.90 123251.96  -51654.92 159858.0
## May 2024    31063.92607 -38765.72 100893.57  -75731.30 137859.2
## Jun 2024    56071.81721 -15779.23 127922.86  -53814.87 165958.5
## Jul 2024    16641.89395 -57682.77  90966.56  -97027.87 130311.7
## Aug 2024    18159.10921 -56455.85  92774.07  -95954.62 132272.8
## Sep 2024    -6567.95853 -83170.01  70034.10 -123720.69 110584.8
## Oct 2024       87.44254 -76538.38  76713.26 -117101.64 117276.5
## Nov 2024    -6668.37617 -83417.95  70081.20 -124046.72 110710.0
## Dec 2024     3536.71145 -73369.69  80443.12 -114081.48 121154.9
## Jan 2025     2737.32661 -74182.56  79657.21 -114901.49 120376.1
## Feb 2025     9077.92631 -67965.08  86120.94 -108749.19 126905.0
## Mar 2025     7179.34990 -69864.43  84223.13 -110648.94 125007.6
## Apr 2025     9004.54846 -68047.45  86056.55 -108836.31 126845.4
## May 2025     6358.93581 -70703.51  83421.38 -111497.90 124215.8
## Jun 2025     6668.91909 -70394.14  83731.98 -111188.85 124526.7
## Jul 2025     5039.13844 -72031.71  82109.99 -112830.55 122908.8
## Aug 2025     5575.37902 -71495.48  82646.24 -112294.34 123445.1
## Sep 2025     5082.28607 -71989.13  82153.70 -112788.27 122952.8
```

```
summary(fit_arima_kc_t)
```

```
## Series: kc_terraced_ts_diff
## ARIMA(3,0,2) with non-zero mean
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2      mean
##       0.4497  0.3675  -0.4367  -0.3092  -0.1859  5868.322
## s.e.  0.1076  0.0910   0.0493   0.1174   0.0940  2343.330
##
## sigma^2 = 2.912e+09:  log likelihood = -4307.49
## AIC=8628.99   AICc=8629.31   BIC=8655.99
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -9.779701 53493.86 36086.47 77.78488 169.5204 0.6362536
##                      ACF1
## Training set 0.006576417
```

```r
# For Flat of Kensington and Chelsea Area
print(forecasted_values_arima_kc_f)
```

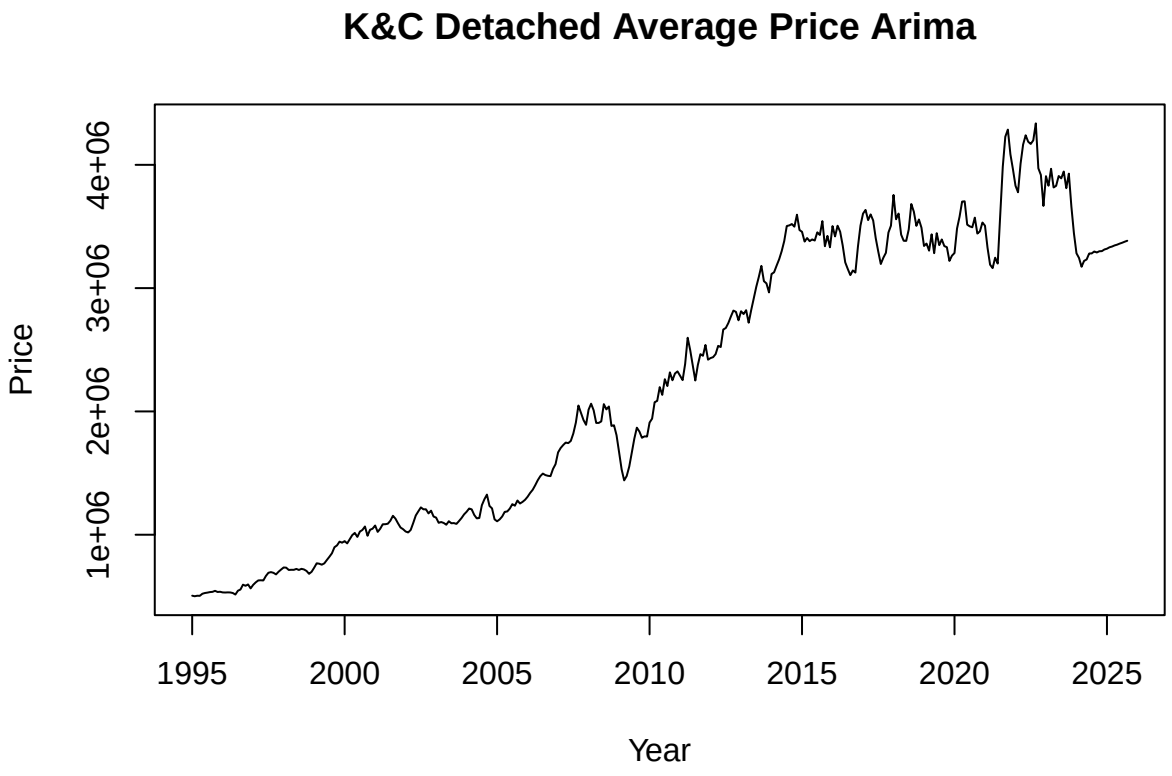
```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Apr 2024     19463.7545 -10909.92 49837.43 -26988.77 65916.28
## May 2024     -1499.4556 -32246.83 29247.92 -48523.51 45524.60
## Jun 2024     11320.7435 -20396.21 43037.70 -37186.16 59827.64
## Jul 2024     -3500.4757 -36549.81 29548.86 -54045.07 47044.12
## Aug 2024      5154.7605 -27928.09 38237.61 -45441.10 55750.62
## Sep 2024     -1818.5406 -35333.06 31695.98 -53074.57 49437.49
## Oct 2024      4757.7666 -28840.14 38355.67 -46625.79 56141.32
## Nov 2024       867.2335 -32732.00 34466.47 -50518.36 52252.83
## Dec 2024      4445.0164 -29250.97 38141.01 -47088.55 55978.59
## Jan 2025      1591.5518 -32109.31 35292.41 -49949.46 53132.57
## Feb 2025      3459.0924 -30247.38 37165.56 -48090.50 55008.69
## Mar 2025      1723.7192 -31997.94 35445.38 -49849.11 53296.55
## Apr 2025      2974.5818 -30747.68 36696.85 -48599.17 54548.34
## May 2025      2055.2620 -31669.84 35780.37 -49522.84 53633.36
## Jun 2025      2864.0408 -30863.04 36591.12 -48717.08 54445.16
## Jul 2025      2299.9682 -31427.33 36027.27 -49281.49 53881.42
## Aug 2025      2750.1196 -30977.98 36478.22 -48832.56 54332.80
## Sep 2025      2380.9727 -31347.39 36109.33 -49202.10 53964.05
```

```r
summary(fit_arima_kc_f)
```

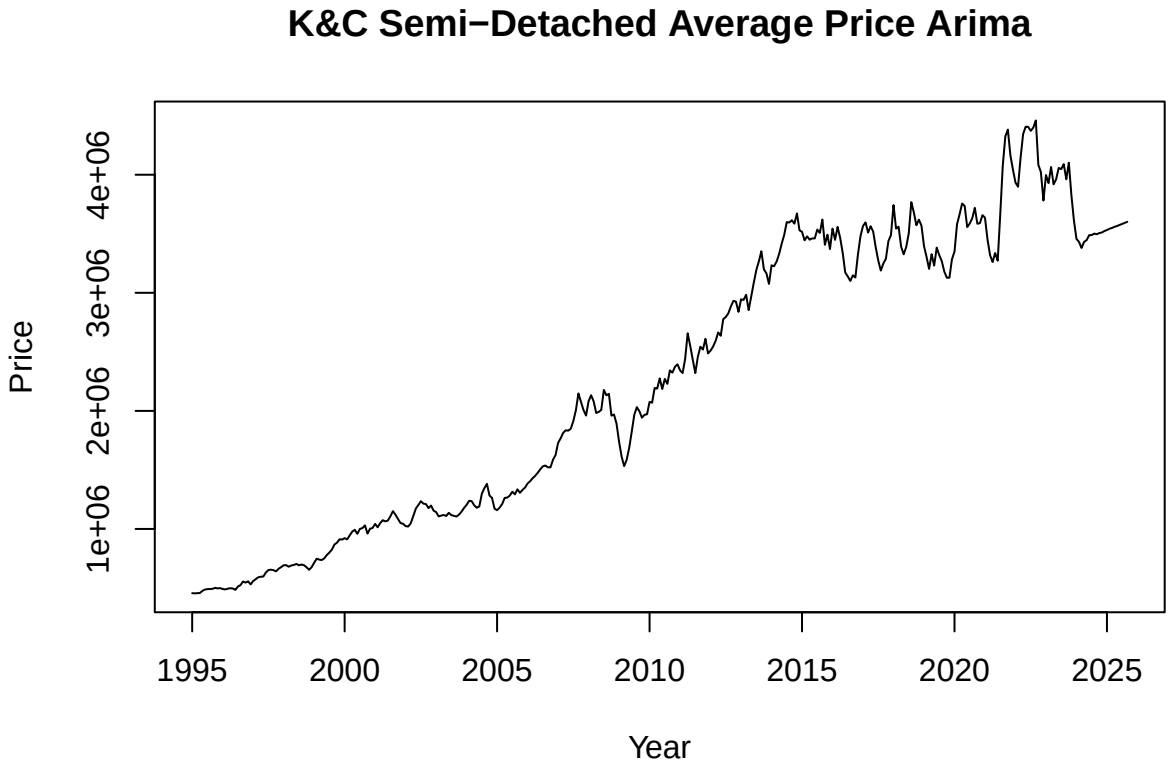
```
## Series: kc_flat_ts_diff
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2      ar3      mean
##       0.1573  0.2315  -0.3826  2527.359
## s.e.  0.0492  0.0484   0.0493  1269.449
##
## sigma^2 = 561723360:  log likelihood = -4020.54
## AIC=8051.08   AICc=8051.26   BIC=8070.37
##
## Training set error measures:
##                       ME     RMSE     MAE      MPE     MAPE      MASE
## Training set -2.447202 23564.88 15997.7 63.06536 134.4906 0.6157379
##                      ACF1
## Training set -0.006248376
```

```r
# Calculate the forecasted actual prices
# By adding the last observed price and the forecasted differences
# In Kensington and Chelsea
# For detached_ts of Kensington and Chelsea Area
last_value_kc_d <- as.numeric(tail(kc_detached_ts, n = 1))
forecasted_values_kc_d <- c(last_value_kc_d, forecasted_values_arima_kc_d$mean)
cumulative_forecasted_values_kc_d <- cumsum(forecasted_values_kc_d)
forecasted_values_kc_d_ts <- ts(cumulative_forecasted_values_kc_d[-1],
                                start = c(2024, 2), frequency = 12)
```

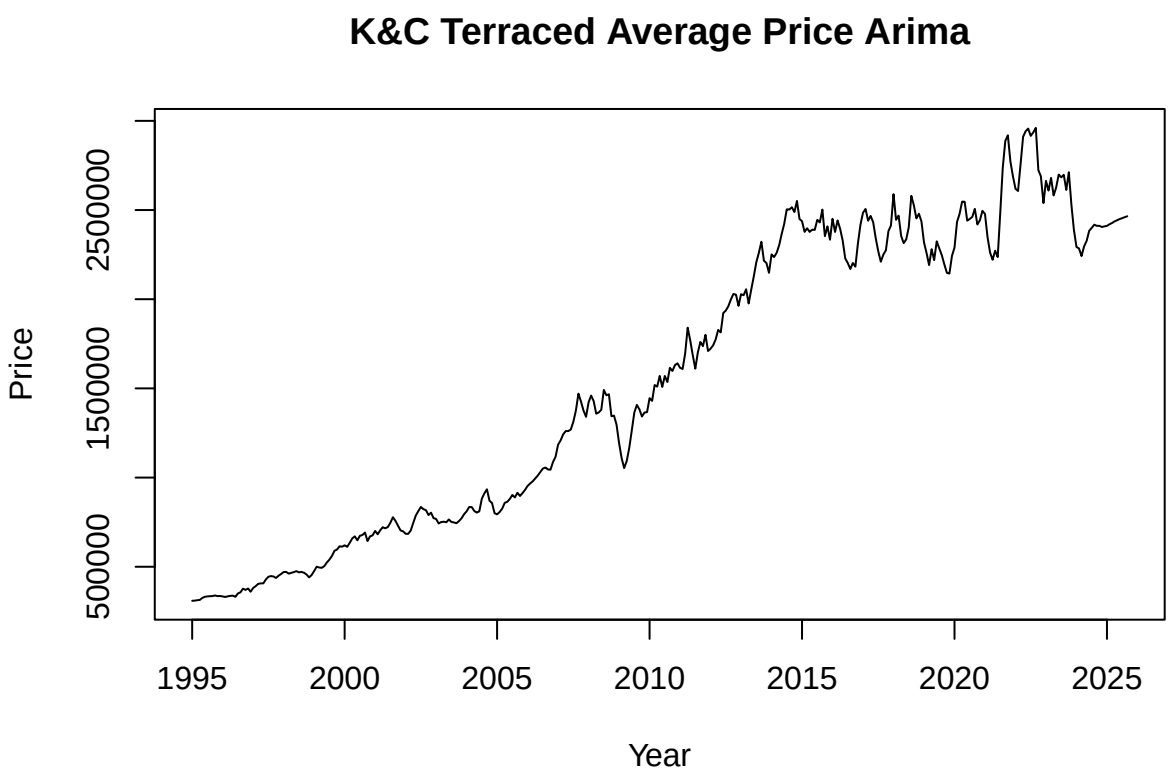
```r
# For semi_detached_ts of Kensington and Chelsea Area
last_value_kc_sd <- as.numeric(tail(kc_semi_detached_ts, n = 1))
forecasted_values_kc_sd <- c(last_value_kc_sd,
                             forecasted_values_arima_kc_sd$mean)
cumulative_forecasted_values_kc_sd <- cumsum(forecasted_values_kc_sd)
forecasted_values_kc_sd_ts <- ts(cumulative_forecasted_values_kc_sd[-1],
                              start = c(2024, 2), frequency = 12)

# For terraced_ts of Kensington and Chelsea Area
last_value_kc_t <- as.numeric(tail(kc_terraced_ts, n = 1))
forecasted_values_kc_t <- c(last_value_kc_t, forecasted_values_arima_kc_t$mean)
cumulative_forecasted_values_kc_t <- cumsum(forecasted_values_kc_t)
forecasted_values_kc_t_ts <- ts(cumulative_forecasted_values_kc_t[-1],
                             start = c(2024, 2), frequency = 12)

# For flat_ts of Kensington and Chelsea Area
last_value_kc_f <- as.numeric(tail(kc_flat_ts, n = 1))
forecasted_values_kc_f <- c(last_value_kc_f, forecasted_values_arima_kc_f$mean)
cumulative_forecasted_values_kc_f <- cumsum(forecasted_values_kc_f)
forecasted_values_kc_f_ts <- ts(cumulative_forecasted_values_kc_f[-1],
                                start = c(2024, 2), frequency = 12)

# Combine the original and forecasted time series of Kensington and Chelsea Area
combined_kc_detached_ts_Arima <- ts(c(as.numeric(kc_detached_ts),
                                      as.numeric(forecasted_values_kc_d_ts)),
                                    start = c(1995, 1), frequency = 12)
combined_kc_semi_detached_ts_Arima <- ts(c(as.numeric(kc_semi_detached_ts),
                                        as.numeric(forecasted_values_kc_sd_ts)),
                                        start = c(1995, 1), frequency = 12)
combined_kc_terraced_ts_Arima <- ts(c(as.numeric(kc_terraced_ts),
                                      as.numeric(forecasted_values_kc_t_ts)),
                                    start = c(1995, 1), frequency = 12)
combined_kc_flat_ts_Arima <- ts(c(as.numeric(kc_flat_ts),
                                  as.numeric(forecasted_values_kc_f_ts)),
                                start = c(1995, 1), frequency = 12)

# Plot the combined time series of Kensington and Chelsea Area
plot(combined_kc_detached_ts_Arima,
     main = "K&C Detached Average Price Arima",
     ylab = "Price", xlab = "Year")
```



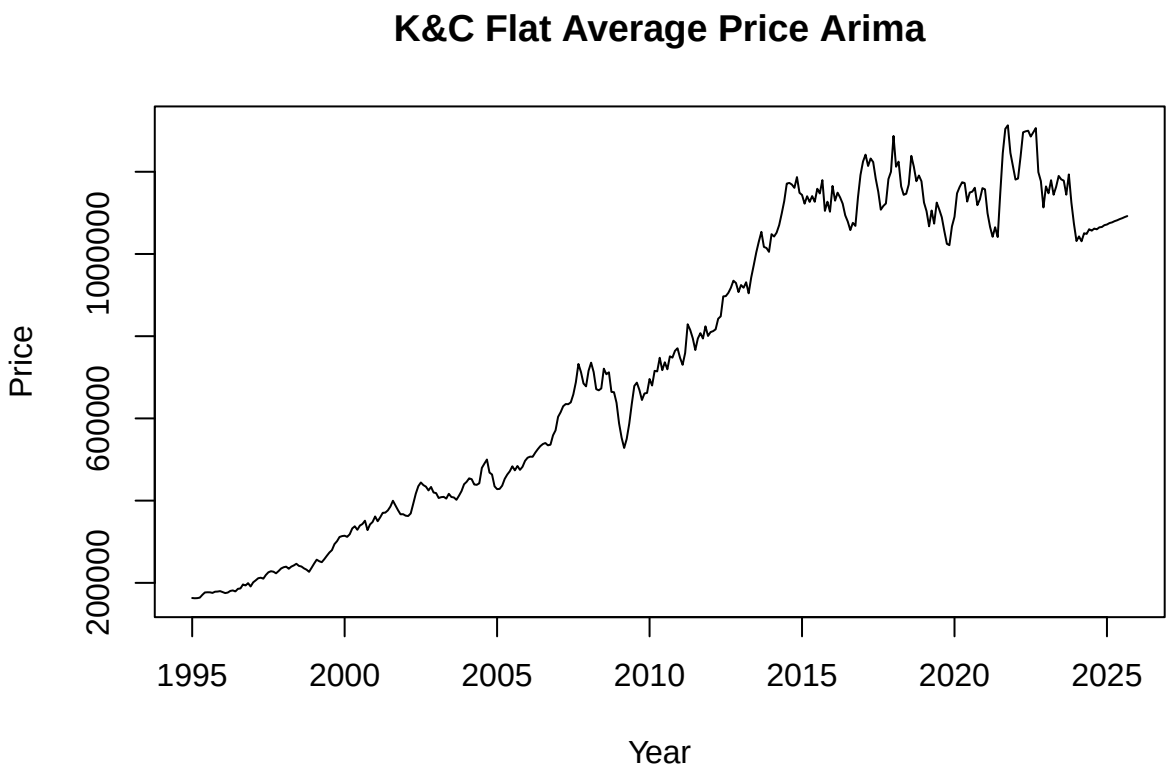

```
plot(combined_kc_semi_detached_ts_Arima,
     main = "K&C Semi-Detached Average Price Arima",
     ylab = "Price", xlab = "Year")
```



```
plot(combined_kc_terraced_ts_Arima,
     main = "K&C Terraced Average Price Arima",
     ylab = "Price", xlab = "Year")
```

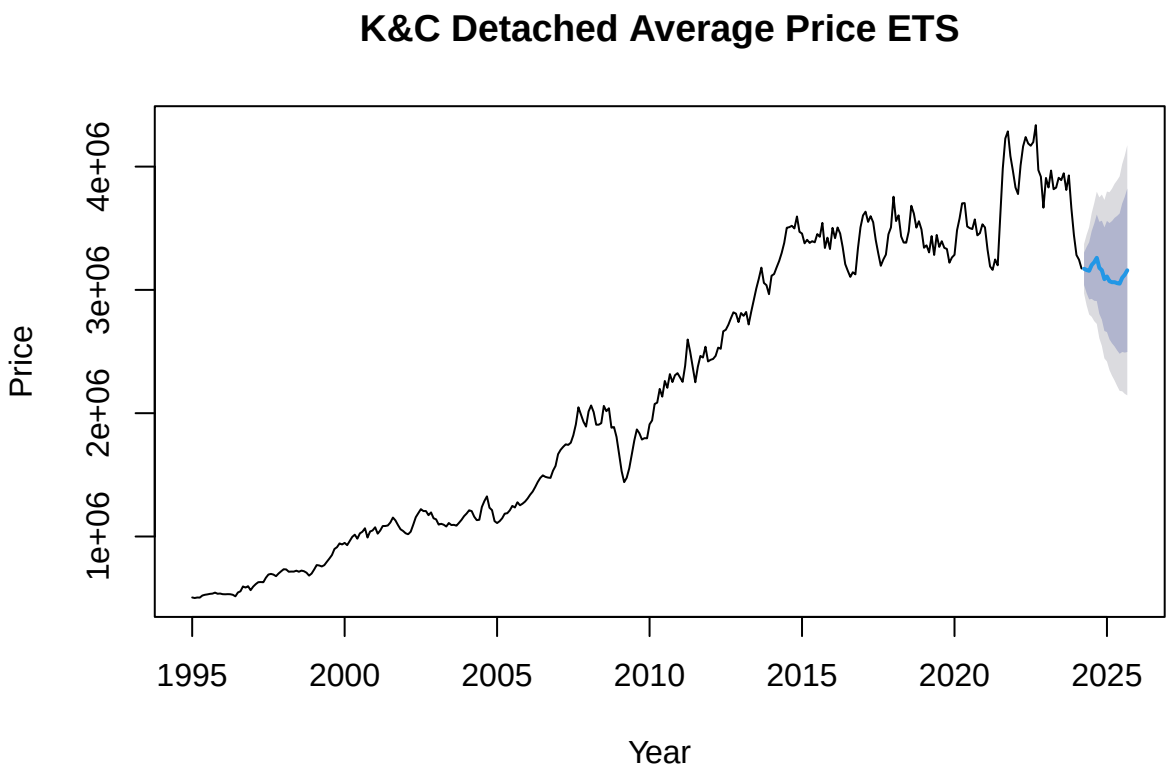



```
plot(combined_kc_flat_ts_Arima,
     main = "K&C Flat Average Price Arima",
     ylab = "Price", xlab = "Year")
```

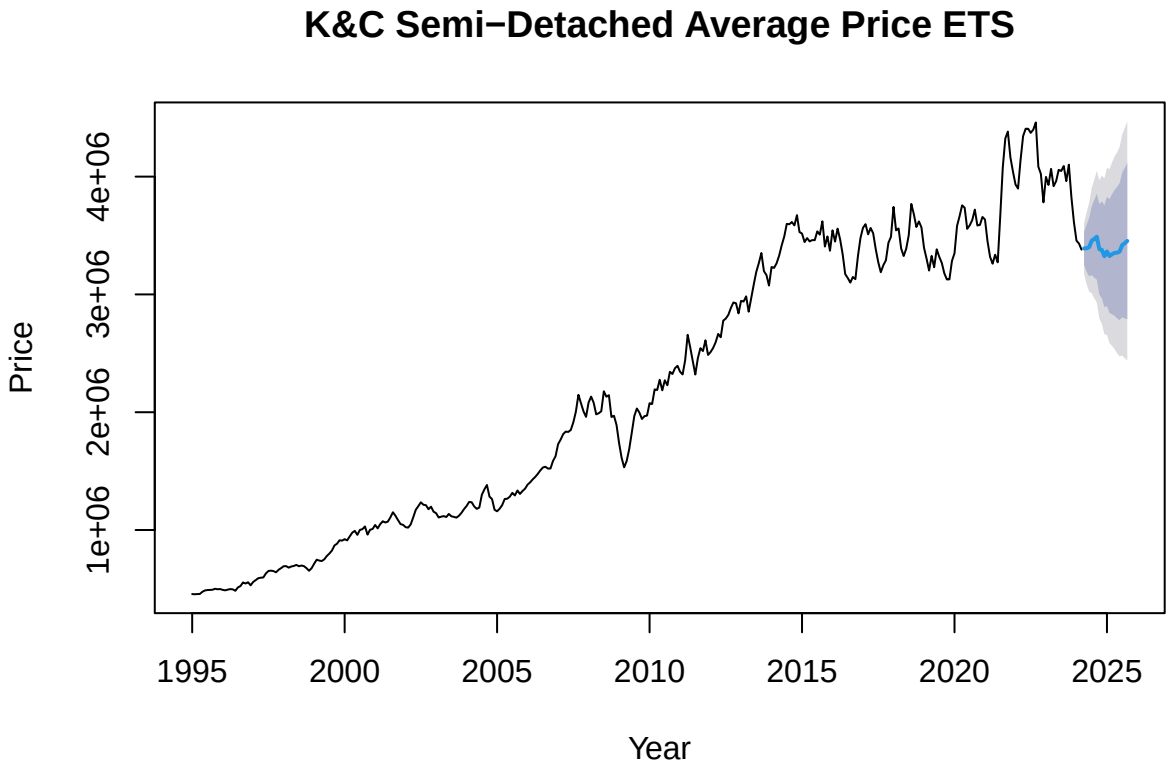



```
# ETS model for Kensington and Chelsea Area
fit_ets_kc_d <- ets(kc_detached_ts)
fit_ets_kc_sd <- ets(kc_semi_detached_ts)
fit_ets_kc_t <- ets(kc_terraced_ts)
fit_ets_kc_f <- ets(kc_flat_ts)

# Forecast using the ETS model for each property type in Kensington and Chelsea
forecasted_values_ets_kc_d <- forecast(fit_ets_kc_d, h = 18)
forecasted_values_ets_kc_sd <- forecast(fit_ets_kc_sd, h = 18)
forecasted_values_ets_kc_t <- forecast(fit_ets_kc_t, h = 18)
forecasted_values_ets_kc_f <- forecast(fit_ets_kc_f, h = 18)

# Combine the historical and forecasted values for each property type by ETS
# In Kensington and Chelsea
combined_kc_detached_ts_ets <- ts(c(kc_detached_price,
                                    forecasted_values_ets_kc_d$mean),
                                  start = c(1995, 1), frequency = 12)
combined_kc_semi_detached_ts_ets <- ts(c(kc_semi_detached_price,
                                         forecasted_values_ets_kc_sd$mean),
                                       start = c(1995, 1), frequency = 12)
combined_kc_terraced_ts_ets <- ts(c(kc_terraced_price,
                                    forecasted_values_ets_kc_t$mean),
                                  start = c(1995, 1), frequency = 12)
combined_kc_flat_ts_ets <- ts(c(kc_flat_price, forecasted_values_ets_kc_f$mean),
                              start = c(1995, 1), frequency = 12)

# Plot the ETS forecast value for each property type in Kensington and Chelsea
plot(forecasted_values_ets_kc_d,
     main = "K&C Detached Average Price ETS",
```

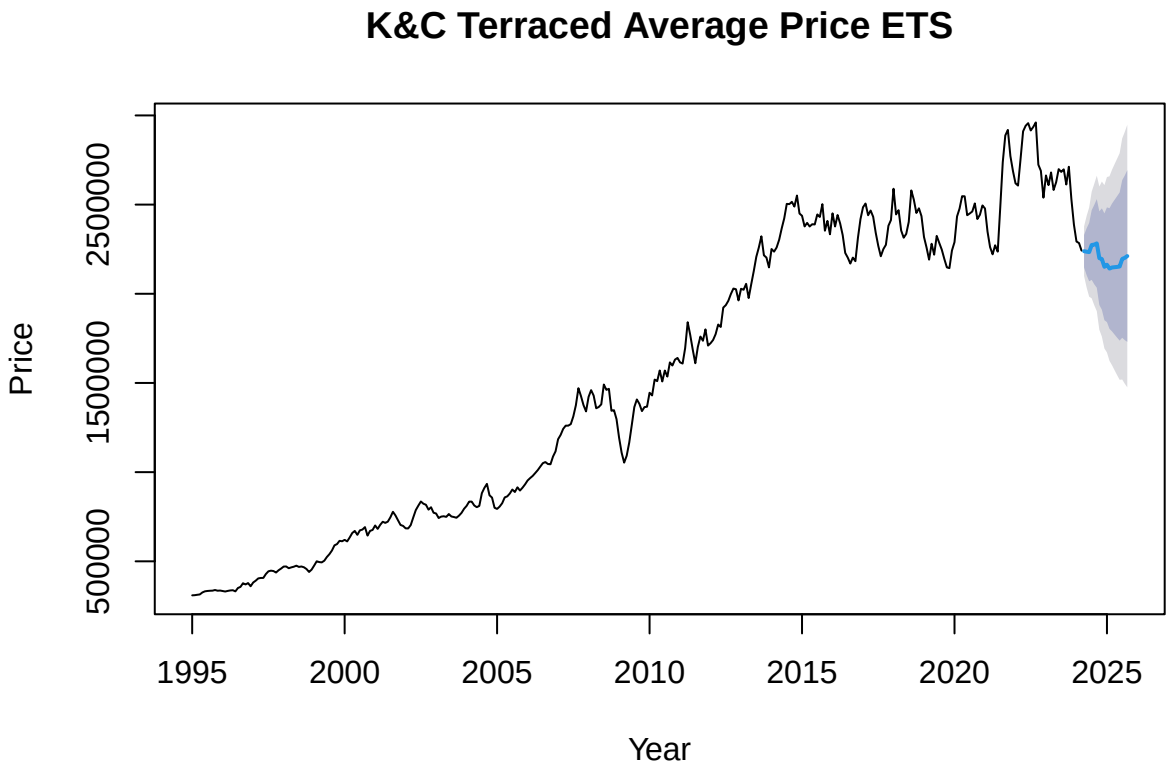
```
     ylab = "Price", xlab = "Year")
```



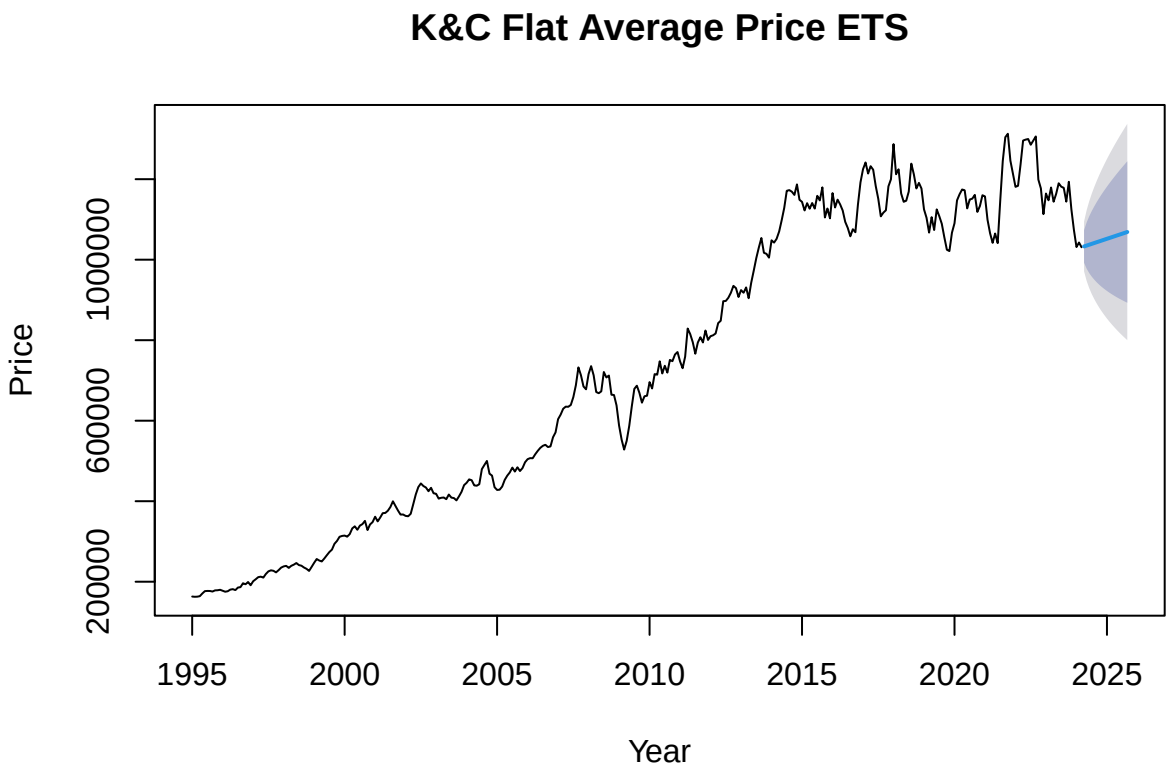

```
plot(forecasted_values_ets_kc_sd,
     main = "K&C Semi-Detached Average Price ETS",
     ylab = "Price", xlab = "Year")
```




```
plot(forecasted_values_ets_kc_t,
     main = "K&C Terraced Average Price ETS",
     ylab = "Price", xlab = "Year")
```



```
plot(forecasted_values_ets_kc_f,
     main = "K&C Flat Average Price ETS",
     ylab = "Price", xlab = "Year")
```




```
# Summary of the ETS model for four different type of properties
# In Kensington and Chelsea
# For Detached of Kensington and Chelsea Area
print(forecasted_values_ets_kc_d)
```

```
##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Apr 2024        3171263 3039947 3302579 2970433 3372093
## May 2024        3160932 2973562 3348303 2874374 3447491
## Jun 2024        3154898 2923094 3386702 2800385 3509411
## Jul 2024        3202143 2927258 3477027 2781743 3622542
## Aug 2024        3225000 2911888 3538111 2746137 3703863
## Sep 2024        3260057 2909393 3610721 2723762 3796351
## Oct 2024        3178978 2805526 3552430 2607832 3750124
## Nov 2024        3159441 2758329 3560553 2545994 3772889
## Dec 2024        3086839 2666729 3506948 2444336 3729341
## Jan 2025        3108606 2657978 3559234 2419430 3797782
## Feb 2025        3070424 2598816 3542032 2349162 3791686
## Mar 2025        3062417 2566183 3558651 2303493 3821341
## Apr 2025        3062622 2540998 3584247 2264866 3860379
## May 2025        3054738 2509603 3599872 2221026 3888449
## Jun 2025        3050960 2482058 3619862 2180900 3921020
## Jul 2025        3098699 2496414 3700985 2177583 4019816
## Aug 2025        3122850 2491504 3754195 2157290 4088409
## Sep 2025        3158816 2495833 3821798 2144871 4172760
```

```
summary(fit_ets_kc_d)
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = kc_detached_ts)
##
##   Smoothing parameters:
##     alpha = 0.9998
##     beta  = 0.0207
##     gamma = 1e-04
##     phi   = 0.9775
##
##   Initial states:
##     l = 502456.762
##     b = 8010.3765
##     s = 0.9863 1.0067 1.01 1.0327 1.0186 1.0083
##            0.9903 0.9891 0.9891 0.9865 0.9864 0.996
##
##   sigma:  0.0323
##
##      AIC     AICc      BIC
## 9777.663 9779.723 9847.157
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set 2847.821 82895.26 57372.68 0.1680941 2.512479 0.270424 0.1309965
```

```
# For Semi-Detached of Kensington and Chelsea Area
print(forecasted_values_ets_kc_sd)
```

```
##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Apr 2024        3390618 3250047 3531189 3175633 3605603
## May 2024        3391552 3191545 3591560 3085667 3697438
## Jun 2024        3400760 3153716 3647804 3022939 3778581
## Jul 2024        3456368 3164787 3747949 3010433 3902303
## Aug 2024        3470718 3141526 3799910 2967262 3974174
## Sep 2024        3489828 3125222 3854433 2932212 4047444
## Oct 2024        3383687 2999764 3767610 2796527 3970847
## Nov 2024        3378006 2966064 3789947 2747996 4008015
## Dec 2024        3323924 2891718 3756130 2662922 3984926
## Jan 2025        3364537 2900987 3828087 2655598 4073475
## Feb 2025        3325774 2842740 3808808 2587037 4064511
## Mar 2025        3341150 2831750 3850550 2562090 4120210
## Apr 2025        3352447 2817811 3887084 2534792 4170103
## May 2025        3354097 2796295 3911899 2501012 4207182
## Jun 2025        3363918 2782063 3945773 2474048 4253788
## Jul 2025        3419637 2805854 4033419 2480937 4358336
## Aug 2025        3434537 2796146 4072928 2458202 4410872
## Sep 2025        3454142 2790462 4117823 2439130 4469154
```

```
summary(fit_ets_kc_sd)
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = kc_semi_detached_ts)
##
##   Smoothing parameters:
##     alpha = 0.9998
##     beta  = 0.0102
##     gamma = 1e-04
##     phi   = 0.98
##
##   Initial states:
##     l = 454277.7251
##     b = 8465.0359
##     s = 0.982 0.9971 0.9978 1.0282 1.0215 1.0163
##            0.999 0.9952 0.9939 0.9897 0.9843 0.9949
##
##   sigma:  0.0324
##
##      AIC     AICc      BIC
## 9785.384 9787.444 9854.878
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 4556.844 85556.05 58638.91 0.2352319 2.499583 0.2643734 0.1811295
```

```
# For Terraced of Kensington and Chelsea Area
print(forecasted_values_ets_kc_t)
```

```
##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Apr 2024        2238761 2147664 2329858 2099440 2378081
## May 2024        2235765 2104617 2366912 2035192 2436338
## Jun 2024        2233755 2070306 2397205 1983781 2483730
## Jul 2024        2273485 2077998 2468972 1974513 2572456
## Aug 2024        2275923 2053439 2498408 1935663 2616184
## Sep 2024        2282338 2034002 2530675 1902541 2662136
## Oct 2024        2199818 1937317 2462320 1798357 2601280
## Nov 2024        2194696 1910626 2478766 1760249 2629143
## Dec 2024        2151158 1851709 2450607 1693191 2609125
## Jan 2025        2162821 1841233 2484410 1670994 2654649
## Feb 2025        2141433 1803233 2479633 1624201 2658665
## Mar 2025        2147446 1788910 2505982 1599113 2695779
## Apr 2025        2149019 1771228 2526810 1571237 2726801
## May 2025        2150563 1753875 2547251 1543882 2757244
## Jun 2025        2152846 1737439 2568253 1517536 2788156
## Jul 2025        2195231 1753315 2637146 1519380 2871082
## Aug 2025        2201496 1740246 2662746 1496075 2906917
## Sep 2025        2211440 1730246 2692635 1475517 2947363
```

```
summary(fit_ets_kc_t)
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = kc_terraced_ts)
##
##   Smoothing parameters:
##     alpha = 0.9998
##     beta  = 0.0334
##     gamma = 2e-04
##     phi   = 0.9465
##
##   Initial states:
##     l = 309676.1405
##     b = 5628.391
##     s = 0.9818 0.9986 0.9976 1.0315 1.0249 1.0199
##            0.998 0.9947 0.9916 0.9884 0.983 0.99
##
##   sigma:  0.0318
##
##      AIC     AICc      BIC
## 9503.477 9505.537 9572.971
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set 3016.14 56923.69 39055.75 0.2727458 2.441415 0.262582 0.1542435
```

```
# For Flat of Kensington and Chelsea Area
print(forecasted_values_ets_kc_f)
```

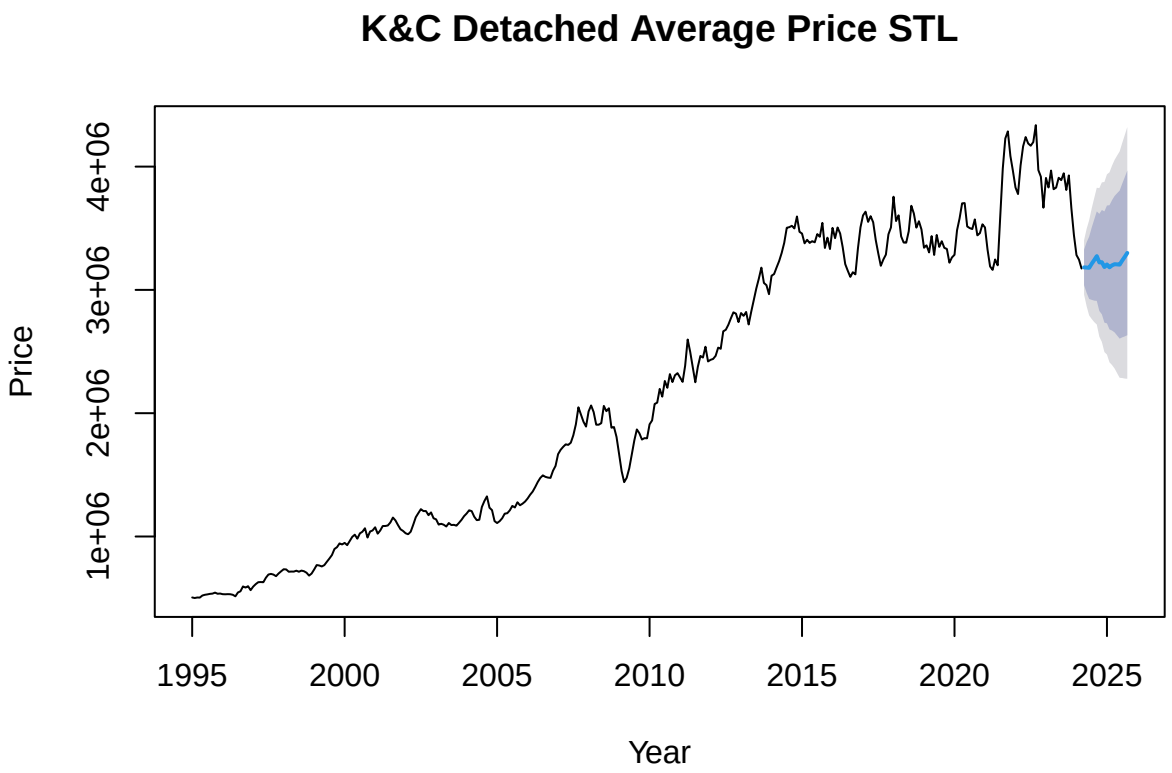
```
##          Point Forecast    Lo 80   Hi 80    Lo 95   Hi 95
## Apr 2024        1032996 992497.2 1073495 971058.3 1094934
## May 2024        1035100 977753.4 1092447 947395.8 1122804
## Jun 2024        1037204 966877.8 1107530 929649.3 1144759
## Jul 2024        1039308 957996.2 1120619 914952.5 1163663
## Aug 2024        1041412 950383.8 1132439 902196.6 1180627
## Sep 2024        1043515 943669.1 1143362 890813.7 1196217
## Oct 2024        1045619 937632.0 1153607 880467.0 1210772
## Nov 2024        1047723 932129.1 1163317 870937.3 1224509
## Dec 2024        1049827 927060.9 1172593 862072.5 1237582
## Jan 2025        1051931 922355.1 1181507 853761.9 1250100
## Feb 2025        1054035 917957.3 1190112 845922.3 1262147
## Mar 2025        1056139 913825.1 1198452 838489.0 1273788
## Apr 2025        1058242 909925.0 1206560 831410.5 1285074
## May 2025        1060346 906229.7 1214463 824645.3 1296047
## Jun 2025        1062450 902716.8 1222183 818159.2 1306741
## Jul 2025        1064554 899367.8 1229740 811923.5 1317184
## Aug 2025        1066658 896166.7 1237149 805914.2 1327401
## Sep 2025        1068762 893100.1 1244423 800110.5 1337413
```

```r
summary(fit_ets_kc_f)
```

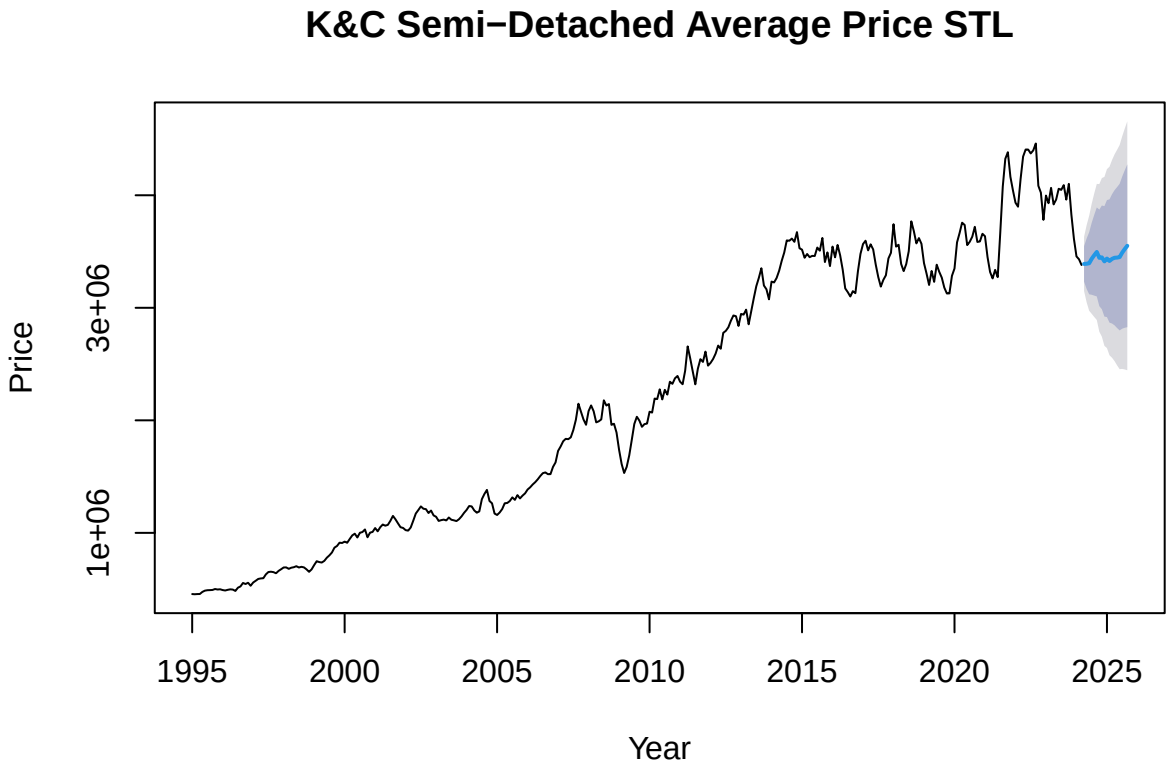
```
## ETS(M,A,N)
##
## Call:
##  ets(y = kc_flat_ts)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##
##   Initial states:
##     l = 160421.7673
##     b = 2087.8621
##
##   sigma:  0.0306
##
##      AIC     AICc      BIC
## 8969.709 8969.883 8989.013
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
## Training set 368.091 26153.64 18346.63 0.06002268 2.418989 0.2752429 0.1023466
```

```r
# STL model for Kensington and Chelsea Area
stl_kc_d <- stl(kc_detached_ts, s.window = "periodic")
stl_kc_sd <- stl(kc_semi_detached_ts, s.window = "periodic")
stl_kc_t <- stl(kc_terraced_ts, s.window = "periodic")
stl_kc_f <- stl(kc_flat_ts, s.window = "periodic")

# Forecast using the STL model for each property type in Kensington and Chelsea
forecasted_values_stl_kc_d <- forecast(stl_kc_d, method='ets',h = 18)
forecasted_values_stl_kc_sd <- forecast(stl_kc_sd, method='ets', h = 18)
forecasted_values_stl_kc_t <- forecast(stl_kc_t, method='ets', h = 18)
forecasted_values_stl_kc_f <- forecast(stl_kc_f, method='ets', h = 18)

# Combine the historical and forecasted values for each property type
# By STL of Kensington and Chelsea Area
combined_kc_detached_ts_stl <- ts(c(kc_detached_price,
                                    forecasted_values_stl_kc_d$mean),
                                  start = c(1995, 1), frequency = 12)
combined_kc_semi_detached_ts_stl <- ts(c(kc_semi_detached_price,
                                         forecasted_values_stl_kc_sd$mean),
                                       start = c(1995, 1), frequency = 12)
combined_kc_terraced_ts_stl <- ts(c(kc_terraced_price,
                                    forecasted_values_stl_kc_t$mean),
                                  start = c(1995, 1), frequency = 12)
combined_kc_flat_ts_stl <- ts(c(kc_flat_price, forecasted_values_stl_kc_f$mean),
                              start = c(1995, 1), frequency = 12)

# Plot the STL forecast value for each property type in Kensington and Chelsea
plot(forecasted_values_stl_kc_d,
     main = "K&C Detached Average Price STL",
```

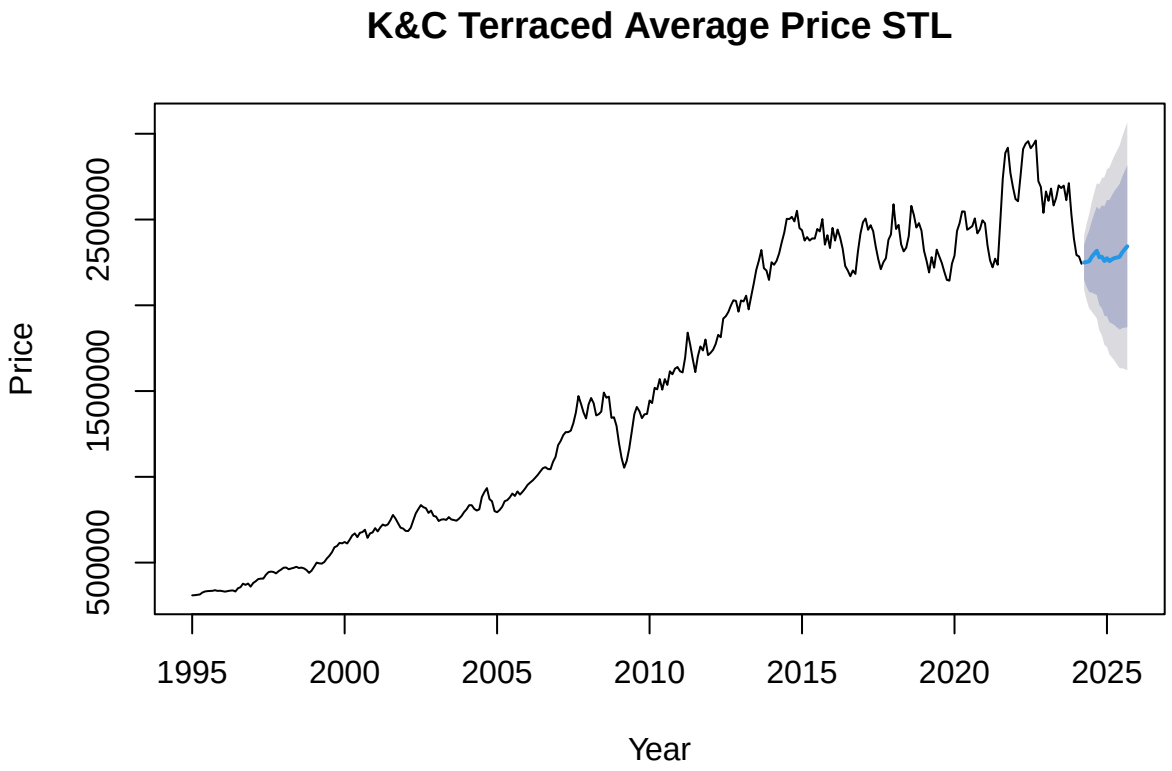
```
     ylab = "Price", xlab = "Year")
```



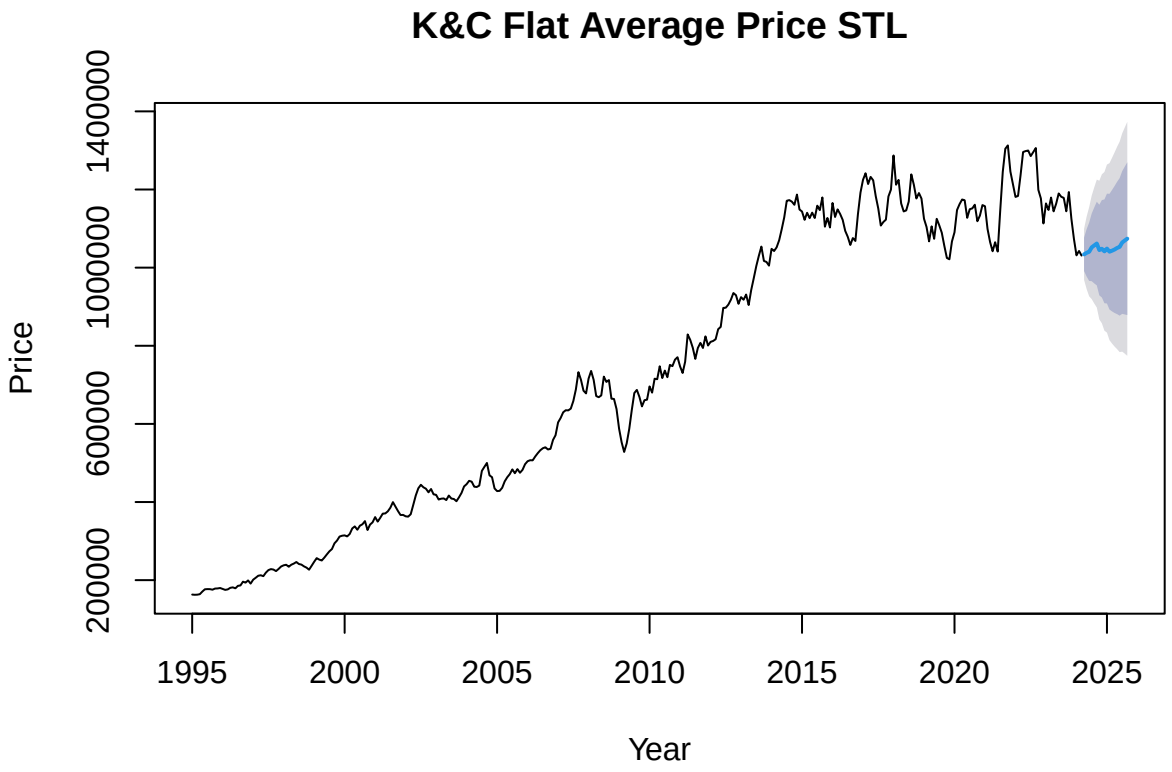

```
plot(forecasted_values_stl_kc_sd,
     main = "K&C Semi-Detached Average Price STL",
     ylab = "Price", xlab = "Year")
```




```
plot(forecasted_values_stl_kc_t,
     main = "K&C Terraced Average Price STL",
     ylab = "Price", xlab = "Year")
```



```
plot(forecasted_values_stl_kc_f,
     main = "K&C Flat Average Price STL",
     ylab = "Price", xlab = "Year")
```




```
# Summary of the STL model for four different type of properties
# In Kensington and Chelsea
# For Detached of Kensington and Chelsea Area
print(forecasted_values_stl_kc_d)
```

```
##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Apr 2024        3182557 3038263 3326851 2961878 3403236
## May 2024        3181108 2975992 3386224 2867410 3494806
## Jun 2024        3179092 2926575 3431609 2792900 3565283
## Jul 2024        3212258 2919167 3505349 2764014 3660502
## Aug 2024        3242200 2912821 3571578 2738459 3745940
## Sep 2024        3273789 2911114 3636464 2719125 3828452
## Oct 2024        3222875 2829128 3616623 2620691 3825060
## Nov 2024        3224703 2801612 3647794 2577641 3871766
## Dec 2024        3185489 2734438 3636540 2495667 3875311
## Jan 2025        3206357 2728484 3684231 2475513 3937201
## Feb 2025        3183030 2679283 3686776 2412616 3953444
## Mar 2025        3199100 2670285 3727915 2390348 4007852
## Apr 2025        3208390 2655199 3761582 2362357 4054424
## May 2025        3206941 2629971 3783911 2324542 4089340
## Jun 2025        3204925 2604700 3805149 2286960 4122889
## Jul 2025        3238091 2615072 3861111 2285266 4190917
## Aug 2025        3268033 2622626 3913439 2280969 4255097
## Sep 2025        3299622 2632190 3967054 2278874 4320370
```

```
summary(stl_kc_d)
```

```
##  Call:
##  stl(x = kc_detached_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend           remainder
##  Min.   :-34370.98   Min.   : 520691   Min.   :-382696.4
##  1st Qu.:-20770.07   1st Qu.:1108982   1st Qu.: -38333.2
##  Median :-13315.80   Median :1955043   Median :   -359.3
##  Mean   :  -181.52   Mean   :2165028   Mean   :   1180.7
##  3rd Qu.: 13760.79   3rd Qu.:3415029   3rd Qu.:  39923.2
##  Max.   : 67152.12   Max.   :4080701   Max.   : 417871.8
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        34531      2306047     78256       2260177
##    %   1.5        102.0       3.5         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Semi-Detached of Kensington and Chelsea Area
print(forecasted_values_stl_kc_sd)
```

```
##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Apr 2024        3389831 3232141 3547521 3148665 3630997
## May 2024        3391498 3167460 3615536 3048862 3734134
## Jun 2024        3395569 3119906 3671232 2973979 3817159
## Jul 2024        3435494 3115708 3755279 2946424 3924563
## Aug 2024        3467312 3108124 3826499 2917982 4016642
## Sep 2024        3496097 3100807 3891386 2891554 4100640
## Oct 2024        3443450 3014518 3872381 2787456 4099444
## Nov 2024        3448105 2987447 3908763 2743590 4152620
## Dec 2024        3412562 2921717 3903408 2661879 4163246
## Jan 2025        3437367 2917600 3957135 2642452 4232283
## Feb 2025        3415717 2868089 3963346 2578192 4253243
## Mar 2025        3434677 2860089 4009265 2555920 4313433
## Apr 2025        3444570 2843798 4045341 2525769 4363371
## May 2025        3446237 2819956 4072517 2488424 4404050
## Jun 2025        3450308 2799110 4101505 2454387 4446229
## Jul 2025        3490232 2814639 4165825 2457002 4523463
## Aug 2025        3522050 2822526 4221575 2452220 4591881
## Sep 2025        3550836 2827793 4273878 2445038 4656633
```

```
summary(stl_kc_sd)
```

```
##  Call:
##  stl(x = kc_semi_detached_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend           remainder
##  Min.   :-35449.80   Min.   : 472286   Min.   :-409376.9
##  1st Qu.:-21051.75   1st Qu.:1125069   1st Qu.: -42067.7
##  Median :-15720.21   Median :2045624   Median :   2944.8
##  Mean   :  -187.29   Mean   :2209426   Mean   :   1068.6
##  3rd Qu.: 10622.89   3rd Qu.:3418362   3rd Qu.:  36742.6
##  Max.   : 67737.72   Max.   :4229933   Max.   : 419285.0
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        31675      2293293     78810       2276849
##    %   1.4        100.7       3.5         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Terraced of Kensington and Chelsea Area
print(forecasted_values_stl_kc_t)
```

```
##          Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
## Apr 2024        2249534 2147151 2351916 2092953 2406115
## May 2024        2252383 2106870 2397896 2029840 2474926
## Jun 2024        2256683 2077574 2435791 1982760 2530605
## Jul 2024        2281759 2073909 2489610 1963879 2599640
## Aug 2024        2300522 2066978 2534066 1943348 2657696
## Sep 2024        2317567 2060459 2574676 1924354 2710780
## Oct 2024        2279812 2000724 2558900 1852984 2706640
## Nov 2024        2282973 1983137 2582809 1824414 2741533
## Dec 2024        2258253 1938658 2577849 1769474 2747033
## Jan 2025        2274422 1935878 2612967 1756664 2792181
## Feb 2025        2257549 1900735 2614363 1711850 2803249
## Mar 2025        2268075 1893567 2642582 1695315 2840834
## Apr 2025        2275648 1883942 2667354 1676585 2874711
## May 2025        2278497 1870021 2686973 1653787 2903207
## Jun 2025        2282797 1857927 2707667 1633015 2932580
## Jul 2025        2307874 1866940 2748807 1633524 2982223
## Aug 2025        2326636 1869933 2783340 1628168 3025105
## Sep 2025        2343682 1871468 2815895 1621494 3065870
```

```
summary(stl_kc_t)
```

```
##  Call:
##  stl(x = kc_terraced_ts, s.window = "periodic")
##
```

```
##  Time.series components:
##     seasonal             trend             remainder
##  Min.   :-25538.23   Min.   : 323090.4   Min.   :-254706.06
##  1st Qu.:-17188.70   1st Qu.: 757127.7   1st Qu.: -28412.40
##  Median : -8994.99   Median :1406792.9   Median :   1306.81
##  Mean   :  -140.22   Mean   :1509558.1   Mean   :    686.95
##  3rd Qu.:  6414.50   3rd Qu.:2365620.5   3rd Qu.:  26038.53
##  Max.   : 45361.01   Max.   :2823075.4   Max.   : 263768.26
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        23603      1608493     54451       1579266
##    %   1.5        101.9       3.4         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```
# For Flat of Kensington and Chelsea Area
print(forecasted_values_stl_kc_f)
```

```
##          Point Forecast    Lo 80   Hi 80    Lo 95   Hi 95
## Apr 2024        1033957 991263.6 1076651 968663.0 1099251
## May 2024        1037874 977221.9 1098526 945114.8 1130633
## Jun 2024        1040651 966028.7 1115272 926526.3 1154775
## Jul 2024        1051898 965339.0 1138456 919517.6 1184278
## Aug 2024        1056833 959617.4 1154049 908154.3 1205512
## Sep 2024        1061649 954670.9 1168628 898039.9 1225259
## Oct 2024        1044992 928918.9 1161066 867473.4 1222511
## Nov 2024        1048246 923597.0 1172894 857611.9 1238879
## Dec 2024        1041722 908916.1 1174528 838612.8 1244832
## Jan 2025        1048754 908134.7 1189374 833695.1 1263814
## Feb 2025        1040809 892663.6 1188954 814240.4 1267377
## Mar 2025        1043384 887959.2 1198810 805682.1 1281087
## Apr 2025        1046449 883955.0 1208943 797935.7 1294963
## May 2025        1050366 880986.2 1219745 791322.1 1309409
## Jun 2025        1053143 877038.9 1229246 783815.2 1322470
## Jul 2025        1064390 881704.2 1247075 784996.3 1343783
## Aug 2025        1069325 880184.6 1258466 780059.4 1358591
## Sep 2025        1074141 878658.2 1269624 775175.7 1373107
```

```
summary(stl_kc_f)
```

```
##  Call:
##  stl(x = kc_flat_ts, s.window = "periodic")
##
##  Time.series components:
##     seasonal             trend             remainder
##  Min.   :-9773.258   Min.   : 167197.8   Min.   :-109170.51
```

```
##  1st Qu.:-6777.819   1st Qu.: 402395.9   1st Qu.: -12929.44
##  Median :-1603.353   Median : 694884.8   Median :    720.86
##  Mean   :  -53.556   Mean   : 724963.3   Mean   :    248.35
##  3rd Qu.:  786.631   3rd Qu.:1123897.6   3rd Qu.:  12841.69
##  Max.   :16272.306   Max.   :1257556.6   Max.   : 110742.08
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##        7564       721502     25771        716898
##    %   1.1        100.6       3.6         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 3511 19 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 352 2 2
##  $ inner: int 2
##  $ outer: int 0
```

```r
# Split the data into training and test sets of Kensington and Chelsea Area
train_end <- c(2023, 6)
test_start <- c(2023, 6)

#Detached property of Kensington and Chelsea Area
# By ARIMA model for Detached of Kensington and Chelsea Area
kc_detached_train_arima <- window(kc_detached_ts_diff, end = train_end)

# Fit best-fit ARIMA models to the training data
# For Detached in Kensington and Chelsea
fit_arima_kc_d_train <- Arima(kc_detached_train_arima, order = c(3, 0, 0))
forecasted_values_arima_kc_d_train <- forecast(fit_arima_kc_d_train, h = 9)

# Add forecasted differenced values to the last observed value
# In Kensington and Chelsea
kc_detached_new_ts <- ts(kc_detached_price, start = c(1995, 1),
                         end = c(2023, 6), frequency = 12)
last_value_kc_detached <- as.numeric(tail(kc_detached_new_ts, n = 1))
forecasted_values_kc_detached_combined <- c(last_value_kc_detached,
                                            forecasted_values_arima_kc_d_train$mean)
cumulative_forecasted_values_kc_detached <-
  cumsum(forecasted_values_kc_detached_combined)
forecasted_values_arima_kc_d_test <-
  ts(cumulative_forecasted_values_kc_detached,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Detached by ARIMA of Kensington and Chelsea Area
mse_kc_detached_arima <- mean((window(kc_detached_ts, start=test_start) -
                                 forecasted_values_arima_kc_d_test)^2)
mae_kc_detached_arima <- mean(abs(window(kc_detached_ts, start=test_start) -
                                    forecasted_values_arima_kc_d_test))

# By ETS model for Detached of Kensington and Chelsea Area
kc_detached_train_ets <- window(kc_detached_ts, end = train_end)
kc_detached_test_ets <- window(kc_detached_ts, start = test_start)
```

```
# Fit ETS models to the training data for Detached in Kensington and Chelsea
fit_ets_kc_d_train <- ets(kc_detached_train_ets)

# Forecast the test period for Detached of Kensington and Chelsea Area
forecasted_values_ets_kc_d_test <- forecast(fit_ets_kc_d_train, h = 9)

# Calculate MSE and MAE by ETS for Detached of Kensington and Chelsea Area
mse_kc_detached_ets <- mean((kc_detached_test_ets -
                               forecasted_values_ets_kc_d_test$mean)^2)
mae_kc_detached_ets <- mean(abs(kc_detached_test_ets -
                                  forecasted_values_ets_kc_d_test$mean))

# By STL model for Detached of Kensington and Chelsea Area
kc_detached_train_stl <- window(kc_detached_ts, end = train_end)
kc_detached_test_stl <- window(kc_detached_ts, start = test_start)
fit_stl_kc_d_train <- stl(kc_detached_train_stl, s.window = "periodic")
forecasted_values_stl_kc_d_test <- forecast(fit_stl_kc_d_train,
                                            method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Detached of Kensington and Chelsea Area
mse_kc_detached_stl <- mean((kc_detached_test_stl -
                               forecasted_values_stl_kc_d_test$mean)^2)
mae_kc_detached_stl <- mean(abs(kc_detached_test_stl -
                                  forecasted_values_stl_kc_d_test$mean))

# Print MSE and MAE of Kensington and Chelsea Area
print(paste("Kensington and Chelsea Detached MSE for Arima:",
            mse_kc_detached_arima))
```

```
## [1] "Kensington and Chelsea Detached MSE for Arima: 256602175180.269"
```

```
print(paste("Kensington and Chelsea Detached MAE for Arima:",
            mae_kc_detached_arima))
```

```
## [1] "Kensington and Chelsea Detached MAE for Arima: 391420.528055581"
```

```
print(paste("Kensington and Chelsea Detached MSE for ETS:",
            mse_kc_detached_ets))
```

```
## [1] "Kensington and Chelsea Detached MSE for ETS: 240845944759.074"
```

```
print(paste("Kensington and Chelsea Detached MAE for ETS:",
            mae_kc_detached_ets))
```

```
## [1] "Kensington and Chelsea Detached MAE for ETS: 401449.559920602"
```

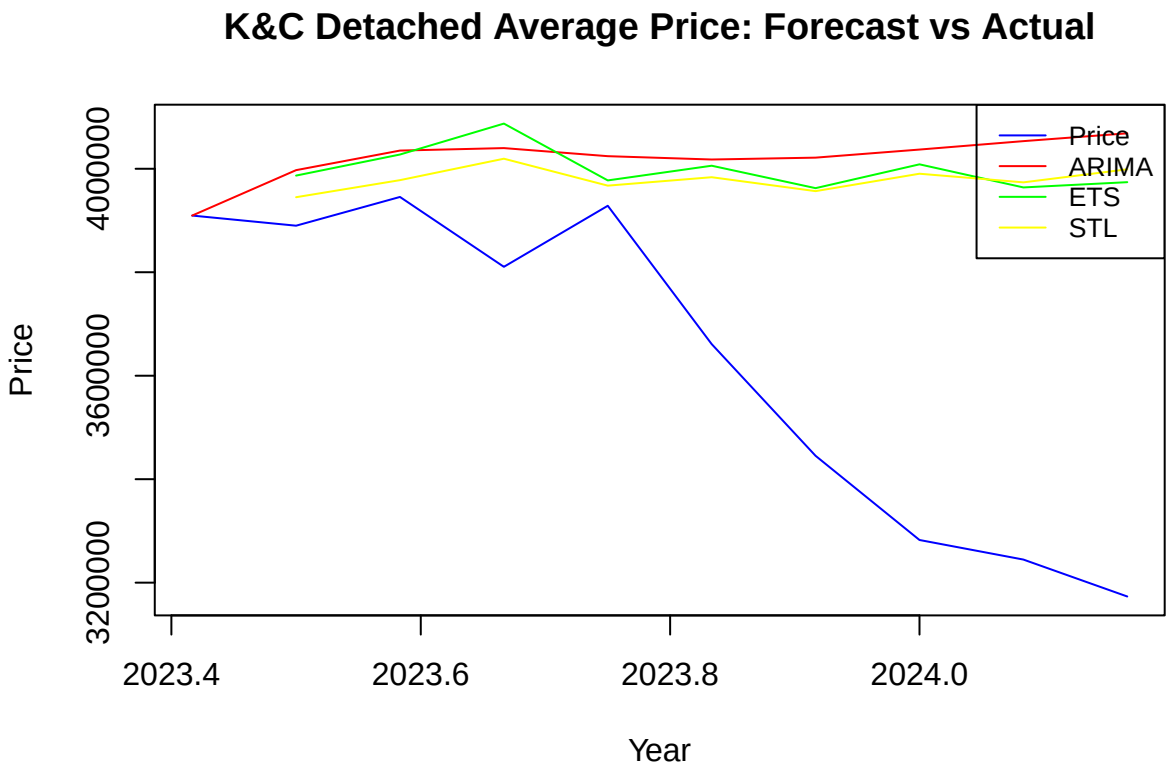
```
print(paste("Kensington and Chelsea Detached MSE for STL:",
            mse_kc_detached_stl))
```

```
## [1] "Kensington and Chelsea Detached MSE for STL: 236617504930.275"
```

```
print(paste("Kensington and Chelsea Detached MAE for STL:",
            mae_kc_detached_stl))
```

```
## [1] "Kensington and Chelsea Detached MAE for STL: 381362.8610051"
```

```
# Plot the combined time series with forecast for Detached
# In Kensington and Chelsea
plot(window(kc_detached_ts, start = train_end), type = "l", col = "blue",
     main = "K&C Detached Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year", ylim = range(c(window(kc_detached_ts,
                                                          start = train_end),
                                                   forecasted_values_arima_kc_d_test,
                                                 forecasted_values_ets_kc_d_test$mean,
                                               forecasted_values_stl_kc_d_test$mean)))
lines(forecasted_values_arima_kc_d_test, col = "red")
lines(forecasted_values_ets_kc_d_test$mean, col = "green")
lines(forecasted_values_stl_kc_d_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```
# Semi-Detached property of Kensington and Chelsea Area
# By ARIMA model for Semi-Detached of Kensington and Chelsea Area
kc_semi_detached_train_arima <-
  window(kc_semi_detached_ts_diff, end = train_end)

# Fit specified ARIMA models to the training data for Semi-Detached
# In Kensington and Chelsea
```

```
fit_arima_kc_sd_train <- Arima(kc_semi_detached_train_arima, order = c(3, 0, 0))
forecasted_values_arima_kc_sd_train <- forecast(fit_arima_kc_sd_train, h = 9)

# Add the forecasted differenced values to the last observed value
# In Kensington and Chelsea
kc_semi_detached_new_ts <- ts(kc_semi_detached_price, start = c(1995, 1),
                              end = c(2023, 6),  frequency = 12)
last_value_kc_semi_detached <- as.numeric(tail(kc_semi_detached_new_ts, n = 1))
forecasted_values_kc_semi_detached_combined <- c(last_value_kc_semi_detached,
                                    forecasted_values_arima_kc_sd_train$mean)
cumulative_forecasted_values_kc_semi_detached <-
  cumsum(forecasted_values_kc_semi_detached_combined)
forecasted_values_arima_kc_sd_test <-
  ts(cumulative_forecasted_values_kc_semi_detached,
     start = test_start, frequency = 12)

# Calculate MSE and MAE for Semi-Detached by ARIMA in Kensington and Chelsea
mse_kc_semi_detached_arima <- mean((window(kc_semi_detached_ts,
                                        start=test_start) -
                                    forecasted_values_arima_kc_sd_test)^2)
mae_kc_semi_detached_arima <- mean(abs(window(kc_semi_detached_ts,
                                           start=test_start) -
                                      forecasted_values_arima_kc_sd_test))

# By ETS model for Semi-Detached of Kensington and Chelsea Area
kc_semi_detached_train_ets <- window(kc_semi_detached_ts, end = train_end)
kc_semi_detached_test_ets <- window(kc_semi_detached_ts, start = test_start)

# Fit ETS models to the training data for Semi-Detached in Kensington and Chelsea
fit_ets_kc_sd_train <- ets(kc_semi_detached_train_ets)

# Forecast the test period for Semi-Detached of Kensington and Chelsea Area
forecasted_values_ets_kc_sd_test <- forecast(fit_ets_kc_sd_train, h = 9)

# Calculate MSE and MAE by ETS for Semi-Detached of Kensington and Chelsea Area
mse_kc_semi_detached_ets <- mean((kc_semi_detached_test_ets -
                                  forecasted_values_ets_kc_sd_test$mean)^2)
mae_kc_semi_detached_ets <- mean(abs(kc_semi_detached_test_ets -
                                     forecasted_values_ets_kc_sd_test$mean))

# By STL model for Semi-Detached of Kensington and Chelsea Area
kc_semi_detached_train_stl <- window(kc_semi_detached_ts, end = train_end)
kc_semi_detached_test_stl <- window(kc_semi_detached_ts, start = test_start)
fit_stl_kc_sd_train <- stl(kc_semi_detached_train_stl, s.window = "periodic")
forecasted_values_stl_kc_sd_test <- forecast(fit_stl_kc_sd_train,
                                             method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Semi-Detached of Kensington and Chelsea Area
mse_kc_semi_detached_stl <- mean((kc_semi_detached_test_stl -
                                  forecasted_values_stl_kc_sd_test$mean)^2)
mae_kc_semi_detached_stl <- mean(abs(kc_semi_detached_test_stl -
                                     forecasted_values_stl_kc_sd_test$mean))
```

```r
# Print MSE and MAE for Semi-Detached of Kensington and Chelsea Area
print(paste("Kensington and Chelsea Semi-Detached MSE for ARIMA:",
            mse_kc_semi_detached_arima))
```

```
## [1] "Kensington and Chelsea Semi-Detached MSE for ARIMA: 231299875931.127"
```

```r
print(paste("Kensington and Chelsea Semi-Detached MAE for ARIMA:",
            mae_kc_semi_detached_arima))
```

```
## [1] "Kensington and Chelsea Semi-Detached MAE for ARIMA: 371214.321803534"
```

```r
print(paste("Kensington and Chelsea Semi-Detached MSE for ETS:",
            mse_kc_semi_detached_ets))
```

```
## [1] "Kensington and Chelsea Semi-Detached MSE for ETS: 195888396227.247"
```

```r
print(paste("Kensington and Chelsea Semi-Detached MAE for ETS:",
            mae_kc_semi_detached_ets))
```

```
## [1] "Kensington and Chelsea Semi-Detached MAE for ETS: 359640.055757359"
```

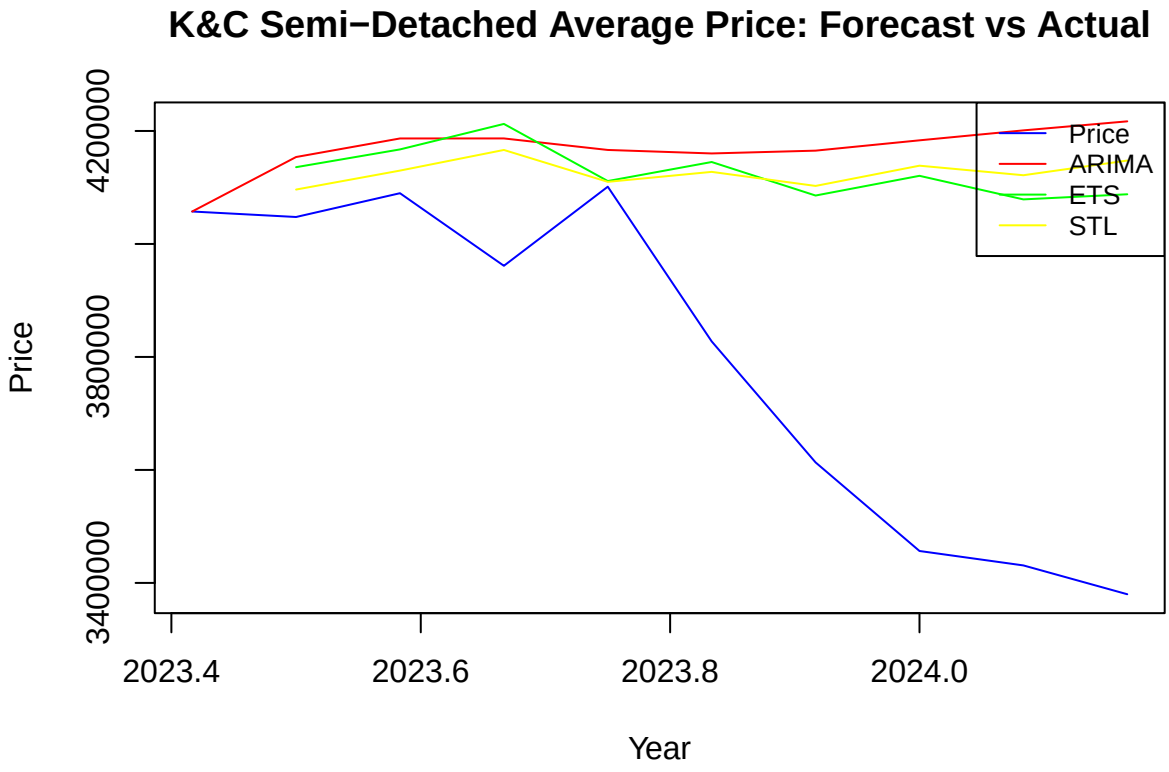
```r
print(paste("Kensington and Chelsea Semi-Detached MSE for STL:",
            mse_kc_semi_detached_stl))
```

```
## [1] "Kensington and Chelsea Semi-Detached MSE for STL: 211915795068.3"
```

```r
print(paste("Kensington and Chelsea Semi-Detached MAE for STL:",
            mae_kc_semi_detached_stl))
```

```
## [1] "Kensington and Chelsea Semi-Detached MAE for STL: 359126.391435254"
```

```r
# Plot the combined time series with forecast for Semi-Detached
# In Kensington and Chelsea
plot(window(kc_semi_detached_ts, start = train_end), type = "l", col = "blue",
     main = "K&C Semi-Detached Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year",
     ylim = range(c(window(kc_semi_detached_ts, start = train_end),
                    forecasted_values_arima_kc_sd_test,
                    forecasted_values_ets_kc_sd_test$mean,
                    forecasted_values_stl_kc_sd_test$mean)))
lines(forecasted_values_arima_kc_sd_test, col = "red")
lines(forecasted_values_ets_kc_sd_test$mean, col = "green")
lines(forecasted_values_stl_kc_sd_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```r
# Terraced property of Kensington and Chelsea Area
# By ARIMA model for Terraced of Kensington and Chelsea Area
kc_terraced_train_arima <- window(kc_terraced_ts_diff, end = train_end)

# Fit specified ARIMA models to the training data for Terraced
# In Kensington and Chelsea
fit_arima_kc_t_train <- Arima(kc_terraced_train_arima, order = c(3, 0, 2))
forecasted_values_arima_kc_t_train <- forecast(fit_arima_kc_t_train, h = 9)

# Add the forecasted differenced values to the last observed value of Terraced
# In Kensington and Chelsea
kc_terraced_new_ts <- ts(kc_terraced_price, start = c(1995, 1),
                         end = c(2023, 6), frequency = 12)
last_value_kc_terraced <- as.numeric(tail(kc_terraced_new_ts, n = 1))
forecasted_values_kc_terraced_combined <- c(last_value_kc_terraced,
                                            forecasted_values_arima_kc_t_train$mean)
cumulative_forecasted_values_kc_terraced <-
  cumsum(forecasted_values_kc_terraced_combined)
forecasted_values_arima_kc_t_test <- ts(cumulative_forecasted_values_kc_terraced
                                        , start = test_start, frequency = 12)

# Calculate MSE and MAE for Terraced by ARIMA of Kensington and Chelsea Area
mse_kc_terraced_arima <- mean((window(kc_terraced_ts, start=test_start) -
                                 forecasted_values_arima_kc_t_test)^2)
mae_kc_terraced_arima <- mean(abs(window(kc_terraced_ts, start=test_start) -
                                    forecasted_values_arima_kc_t_test))

# By ETS model for Terraced of Kensington and Chelsea Area
kc_terraced_train_ets <- window(kc_terraced_ts, end = train_end)
```

```
kc_terraced_test_ets <- window(kc_terraced_ts, start = test_start)

# Fit ETS models to the training data for Terraced in Kensington and Chelsea
fit_ets_kc_t_train <- ets(kc_terraced_train_ets)

# Forecast the test period for Terraced of Kensington and Chelsea Area
forecasted_values_ets_kc_t_test <- forecast(fit_ets_kc_t_train, h = 9)

# Calculate MSE and MAE by ETS for Terraced of Kensington and Chelsea Area
mse_kc_terraced_ets <- mean((kc_terraced_test_ets -
                               forecasted_values_ets_kc_t_test$mean)^2)
mae_kc_terraced_ets <- mean(abs(kc_terraced_test_ets -
                                  forecasted_values_ets_kc_t_test$mean))

# By STL model for Terraced of Kensington and Chelsea Area
kc_terraced_train_stl <- window(kc_terraced_ts, end = train_end)
kc_terraced_test_stl <- window(kc_terraced_ts, start = test_start)
fit_stl_kc_t_train <- stl(kc_terraced_train_stl, s.window = "periodic")
forecasted_values_stl_kc_t_test <- forecast(fit_stl_kc_t_train,
                                            method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Terraced of Kensington and Chelsea Area
mse_kc_terraced_stl <- mean((kc_terraced_test_stl -
                               forecasted_values_stl_kc_t_test$mean)^2)
mae_kc_terraced_stl <- mean(abs(kc_terraced_test_stl -
                                  forecasted_values_stl_kc_t_test$mean))

# Print MSE and MAE of Terraced in Kensington and Chelsea Area
print(paste("Kensington and Chelsea Terraced MSE for ARIMA:",
            mse_kc_terraced_arima))
```

```
## [1] "Kensington and Chelsea Terraced MSE for ARIMA: 100208878811.946"
```

```
print(paste("Kensington and Chelsea Terraced MAE for ARIMA:",
            mae_kc_terraced_arima))
```

```
## [1] "Kensington and Chelsea Terraced MAE for ARIMA: 253548.054587164"
```

```
print(paste("Kensington and Chelsea Terraced MSE for ETS:",
            mse_kc_terraced_ets))
```

```
## [1] "Kensington and Chelsea Terraced MSE for ETS: 81455270012.4159"
```

```
print(paste("Kensington and Chelsea Terraced MAE for ETS:",
            mae_kc_terraced_ets))
```

```
## [1] "Kensington and Chelsea Terraced MAE for ETS: 239365.286497315"
```

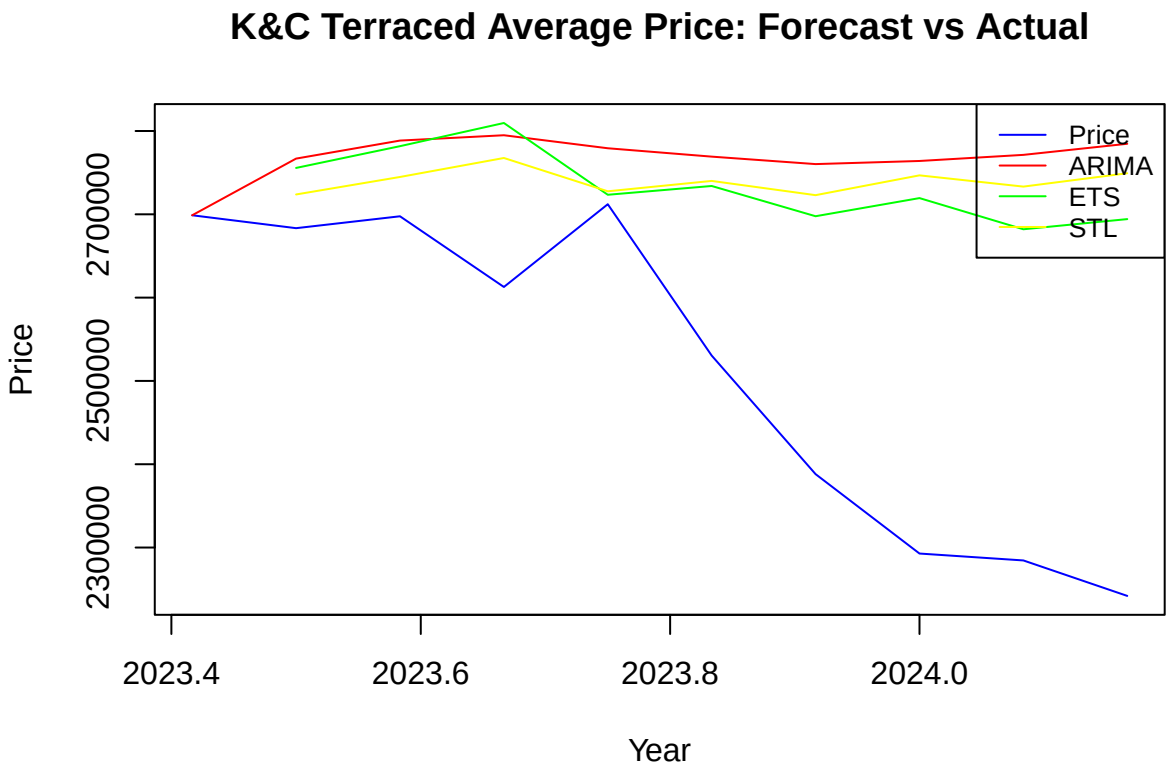
```
print(paste("Kensington and Chelsea Terraced MSE for STL:",
            mse_kc_terraced_stl))
```

```
## [1] "Kensington and Chelsea Terraced MSE for STL: 94334251220.8665"
```

```
print(paste("Kensington and Chelsea Terraced MAE for STL:",
            mae_kc_terraced_stl))
```

```
## [1] "Kensington and Chelsea Terraced MAE for STL: 245836.40431311"
```

```
# Plot the combined time series with forecast for Terraced
# In Kensington and Chelsea
plot(window(kc_terraced_ts, start = train_end), type = "l", col = "blue",
     main = "K&C Terraced Average Price: Forecast vs Actual",
     ylab = "Price", xlab = "Year", ylim = range(c(window(kc_terraced_ts,
                                                          start = train_end),
                                        forecasted_values_arima_kc_t_test,
                                        forecasted_values_ets_kc_t_test$mean,
                                        forecasted_values_stl_kc_t_test$mean)))
lines(forecasted_values_arima_kc_t_test, col = "red")
lines(forecasted_values_ets_kc_t_test$mean, col = "green")
lines(forecasted_values_stl_kc_t_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```




```
# Flat property of Kensington and Chelsea Area
# By ARIMA model for Flat of Kensington and Chelsea Area
kc_flat_train_arima <- window(kc_flat_ts_diff, end = train_end)

# Fit specified ARIMA models to the training data for Flat
```

```r
# In Kensington and Chelsea
fit_arima_kc_f_train <- Arima(kc_flat_train_arima, order = c(3, 0, 2))
forecasted_values_arima_kc_f_train <- forecast(fit_arima_kc_f_train, h = 9)

# Add the forecasted differenced values to the last observed value
# In Kensington and Chelsea
kc_flat_new_ts <- ts(kc_flat_price, start = c(1995, 1),
                     end = c(2023, 6), frequency = 12)
last_value_kc_flat <- as.numeric(tail(kc_flat_new_ts, n = 1))
forecasted_values_kc_flat_combined <-
  c(last_value_kc_flat, forecasted_values_arima_kc_f_train$mean)
cumulative_forecasted_values_kc_flat <-
  cumsum(forecasted_values_kc_flat_combined)
forecasted_values_arima_kc_f_test <- ts(cumulative_forecasted_values_kc_flat,
                                        start = test_start, frequency = 12)

# Calculate MSE and MAE for Flat by ARIMA of Kensington and Chelsea Area
mse_kc_flat_arima <- mean((window(kc_flat_ts, start=test_start) -
                             forecasted_values_arima_kc_f_test)^2)
mae_kc_flat_arima <- mean(abs(window(kc_flat_ts, start=test_start) -
                                forecasted_values_arima_kc_f_test))

# By ETS model for Flat of Kensington and Chelsea Area
kc_flat_train_ets <- window(kc_flat_ts, end = train_end)
kc_flat_test_ets <- window(kc_flat_ts, start = test_start)

# Fit ETS models to the training data for Flat of Kensington and Chelsea Area
fit_ets_kc_f_train <- ets(kc_flat_train_ets)

# Forecast the test period for Flat of Kensington and Chelsea Area
forecasted_values_ets_kc_f_test <- forecast(fit_ets_kc_f_train, h = 9)

# Calculate MSE and MAE by ETS for Flat of Kensington and Chelsea Area
mse_kc_flat_ets <- mean((kc_flat_test_ets -
                           forecasted_values_ets_kc_f_test$mean)^2)
mae_kc_flat_ets <- mean(abs(kc_flat_test_ets -
                              forecasted_values_ets_kc_f_test$mean))

# By STL model for Flat of Kensington and Chelsea Area
kc_flat_train_stl <- window(kc_flat_ts, end = train_end)
kc_flat_test_stl <- window(kc_flat_ts, start = test_start)
fit_stl_kc_f_train <- stl(kc_flat_train_stl, s.window = "periodic")
forecasted_values_stl_kc_f_test <- forecast(fit_stl_kc_f_train,
                                            method = 'ets', h = 9)

# Calculate MSE and MAE by STL for Flat of Kensington and Chelsea Area
mse_kc_flat_stl <- mean((kc_flat_test_stl -
                           forecasted_values_stl_kc_f_test$mean)^2)
mae_kc_flat_stl <- mean(abs(kc_flat_test_stl -
                              forecasted_values_stl_kc_f_test$mean))

# Print MSE and MAE of Flat in Kensington and Chelsea Area
print(paste("Kensington and Chelsea Flat MSE for ARIMA:", mse_kc_flat_arima))
```

```
## [1] "Kensington and Chelsea Flat MSE for ARIMA: 13701932617.2996"
```

```
print(paste("Kensington and Chelsea Flat MAE for ARIMA:", mae_kc_flat_arima))
```

```
## [1] "Kensington and Chelsea Flat MAE for ARIMA: 95129.9942786225"
```

```
print(paste("Kensington and Chelsea Flat MSE for ETS:", mse_kc_flat_ets))
```

```
## [1] "Kensington and Chelsea Flat MSE for ETS: 12957579481.7158"
```

```
print(paste("Kensington and Chelsea Flat MAE for ETS:", mae_kc_flat_ets))
```

```
## [1] "Kensington and Chelsea Flat MAE for ETS: 90353.5516287361"
```

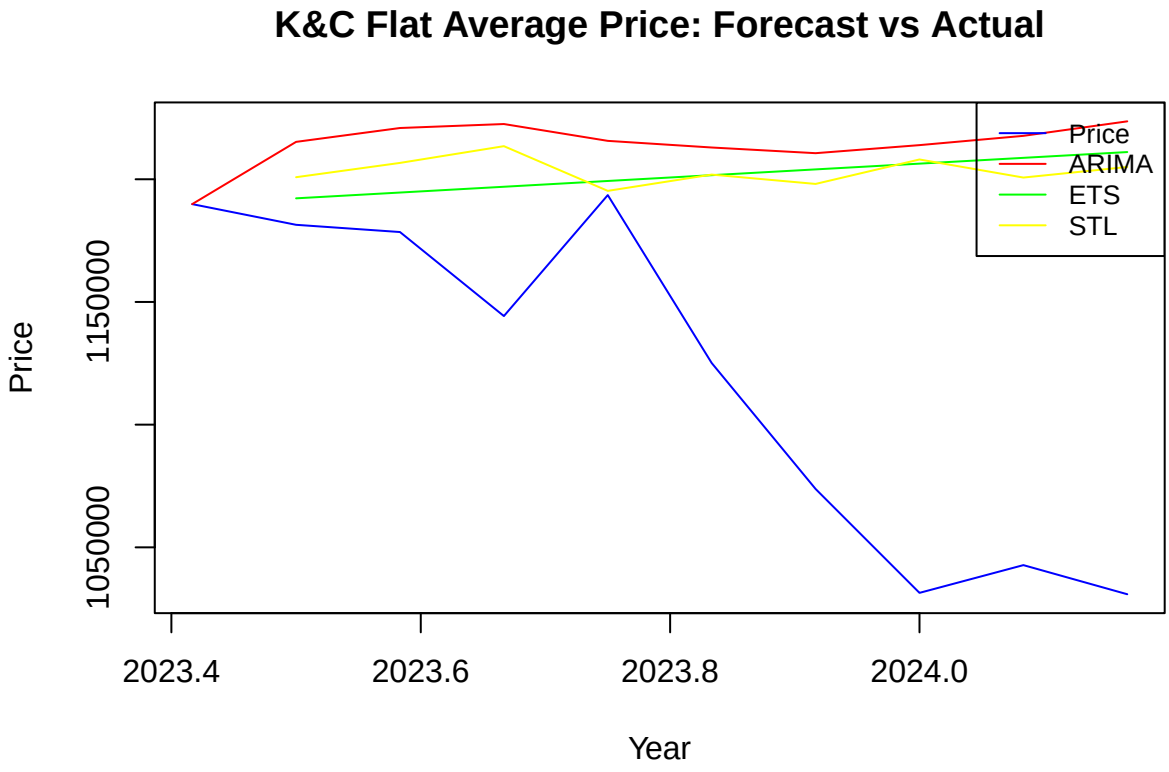
```
print(paste("Kensington and Chelsea Flat MSE for STL:", mse_kc_flat_stl))
```

```
## [1] "Kensington and Chelsea Flat MSE for STL: 12640829974.4374"
```

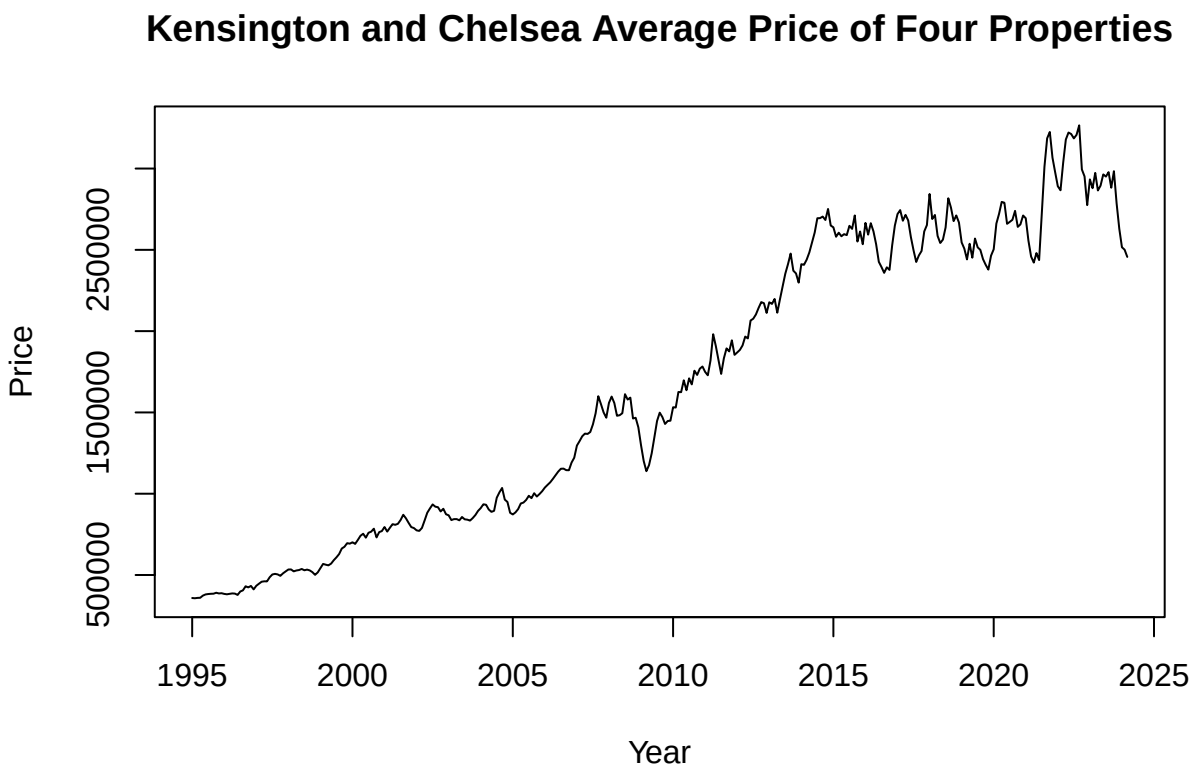
```
print(paste("Kensington and Chelsea Flat MAE for STL:", mae_kc_flat_stl))
```

```
## [1] "Kensington and Chelsea Flat MAE for STL: 92030.6816328988"
```

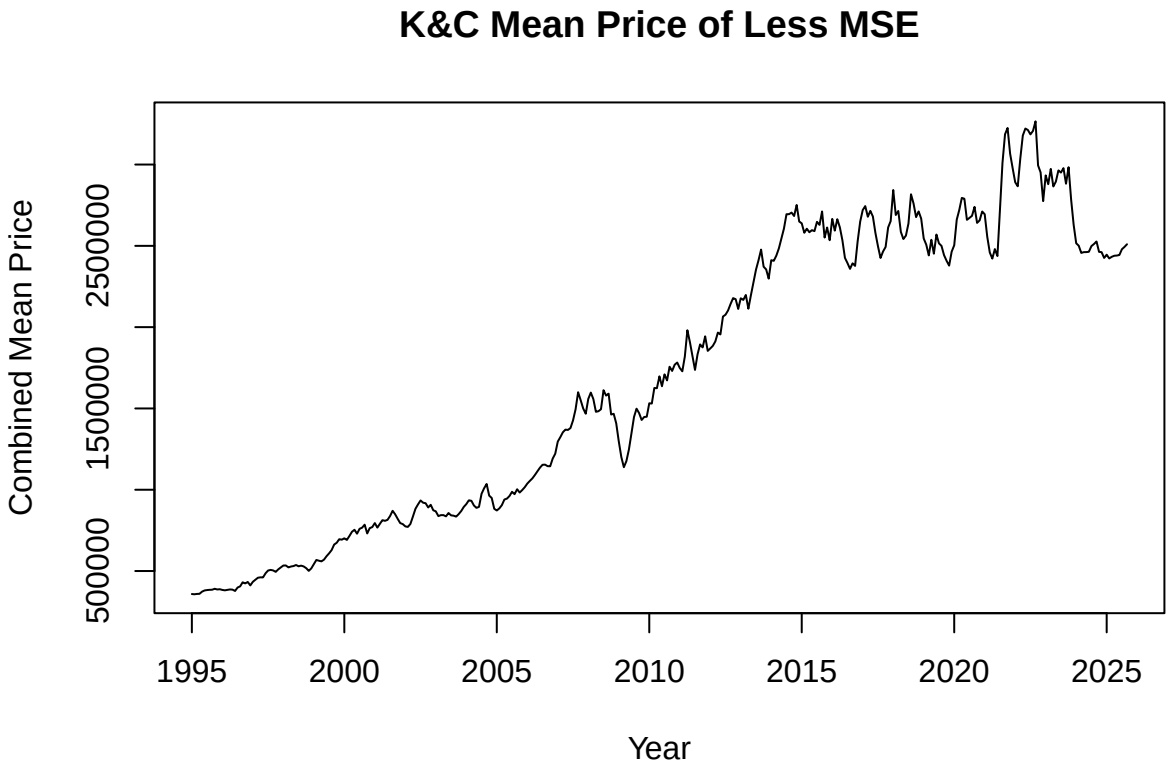
```
# Plot the combined time series with forecast for Flat
# In Kensington and Chelsea Area
plot(window(kc_flat_ts, start = train_end), type = "l", col = "blue",
     main = "K&C Flat Average Price: Forecast vs Actual", ylab = "Price",
     xlab = "Year", ylim = range(c(window(kc_flat_ts, start = train_end),
                                   forecasted_values_arima_kc_f_test,
                                   forecasted_values_ets_kc_f_test$mean,
                                   forecasted_values_stl_kc_f_test$mean)))
lines(forecasted_values_arima_kc_f_test, col = "red")
lines(forecasted_values_ets_kc_f_test$mean, col = "green")
lines(forecasted_values_stl_kc_f_test$mean, col = "yellow")

legend("topright", legend = c("Price", "ARIMA", "ETS", "STL"),
       col = c("blue", "red", "green", "yellow"), lty = 1, cex = 0.8)
```



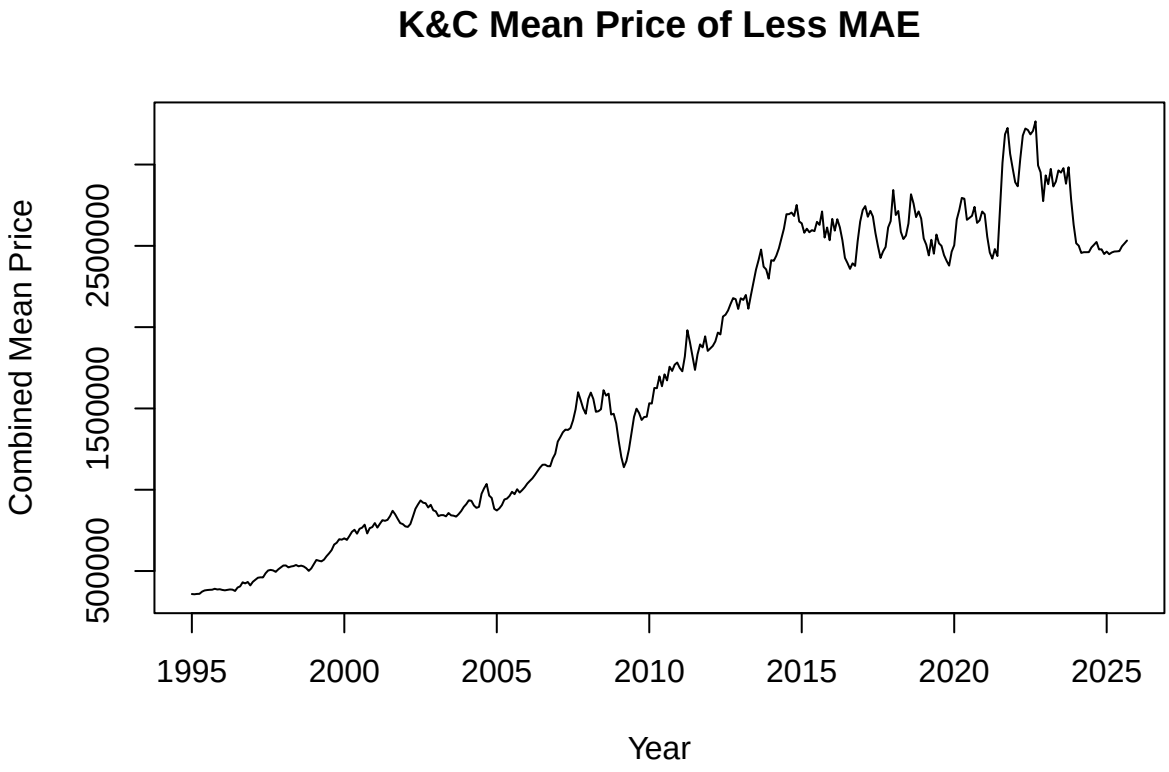

```
# Calculate the mean price of four property type of Kensington and Chelsea Area
combined_mean_price_kc <- (kc_detached_price + kc_semi_detached_price +
                             kc_terraced_price + kc_flat_price)/4
combined_mean_price_kc_ts <- ts(combined_mean_price_kc, start = c(1995, 1),
                                frequency = 12)
plot(combined_mean_price_kc_ts,
     main = "Kensington and Chelsea Average Price of Four Properties",
     ylab = "Price", xlab = "Year")
```




```
# If we look for less MSE in Kensington and Chelsea Area,
# STL model is the best for Detached
# ETS model is the best for Semi-Detached
# ETS model is the best for Terraced
# STL model is the best for Flat

# Calculate the combined mean price for all property types
# of Kensington and Chelsea Area with less MSE
combined_mean_price_kc_less_MSE <- (combined_kc_detached_ts_stl +
                                      combined_kc_semi_detached_ts_ets +
                                      combined_kc_terraced_ts_ets +
                                      combined_kc_flat_ts_stl) / 4
combined_mean_price_kc_less_MSE_ts <- ts(combined_mean_price_kc_less_MSE,
                                         start = c(1995, 1), frequency = 12)
plot(combined_mean_price_kc_less_MSE_ts,
     main = "K&C Mean Price of Less MSE",
     ylab = "Combined Mean Price", xlab = "Year")
```




```
# If we look for less MAE in Kensington and Chelsea Area,
# STL model is the best for Detached
# STL model is the best for Semi-Detached
# ETS model is the best for Terraced
# ETS model is the best for Flat

# Calculate the combined mean price for all property types
# of Kensington and Chelsea Area with less MAE
combined_mean_price_kc_less_MAE <- (combined_kc_detached_ts_stl +
                                    combined_kc_semi_detached_ts_stl +
                                    combined_kc_terraced_ts_ets +
                                    combined_kc_flat_ts_ets) / 4
combined_mean_price_kc_less_MAE_ts <- ts(combined_mean_price_kc_less_MAE,
                                         start = c(1995, 1), frequency = 12)
plot(combined_mean_price_kc_less_MAE_ts, main = "K&C Mean Price of Less MAE",
     ylab = "Combined Mean Price", xlab = "Year")
```




```r
# Compare two models for which one is better in Kensington and Chelsea Area

combined_mean_price_kc_less_MSE_test <- (forecasted_values_stl_kc_d_test$mean
                                         + forecasted_values_ets_kc_sd_test$mean
                                         + forecasted_values_ets_kc_t_test$mean
                                       + forecasted_values_stl_kc_f_test$mean) / 4
combined_mean_price_kc_less_MSE_test_ts <-
  ts(combined_mean_price_kc_less_MSE_test, start = test_start, frequency = 12)

combined_mean_price_kc_less_MAE_test <- (forecasted_values_stl_kc_d_test$mean +
                                           forecasted_values_stl_kc_sd_test$mean
                                         + forecasted_values_ets_kc_t_test$mean
                                       + forecasted_values_ets_kc_f_test$mean) / 4
combined_mean_price_kc_less_MAE_test_ts <-
  ts(combined_mean_price_kc_less_MAE_test, start = test_start, frequency = 12)

# Calculate MSE and MAE for combined mean prices in Kensington and Chelsea Area
combined_mean_price_kc_test <- window(combined_mean_price_kc_ts,
                                      start = test_start)
mse_combined_kc_less_MSE <- mean((combined_mean_price_kc_test -
                                    combined_mean_price_kc_less_MSE_test_ts)^2)
mae_combined_kc_less_MSE <- mean(abs(combined_mean_price_kc_test -
                                       combined_mean_price_kc_less_MSE_test_ts))

mse_combined_kc_less_MAE <- mean((combined_mean_price_kc_test -
                                    combined_mean_price_kc_less_MAE_test_ts)^2)
mae_combined_kc_less_MAE <- mean(abs(combined_mean_price_kc_test -
                                       combined_mean_price_kc_less_MAE_test_ts))
```

```r
# Print MSE and MAE for combined mean prices in Kensington and Chelsea Area
print(paste("Kensington and Chelsea Combined Mean Price MSE for Less MSE Model:"
            , mse_combined_kc_less_MSE))
```

```
## [1] "Kensington and Chelsea Combined Mean Price MSE for Less MSE Model: 75489443198.045"
```

```r
print(paste("Kensington and Chelsea Combined Mean Price MAE for Less MSE Model:"
            , mae_combined_kc_less_MSE))
```

```
## [1] "Kensington and Chelsea Combined Mean Price MAE for Less MSE Model: 211719.110112057"
```

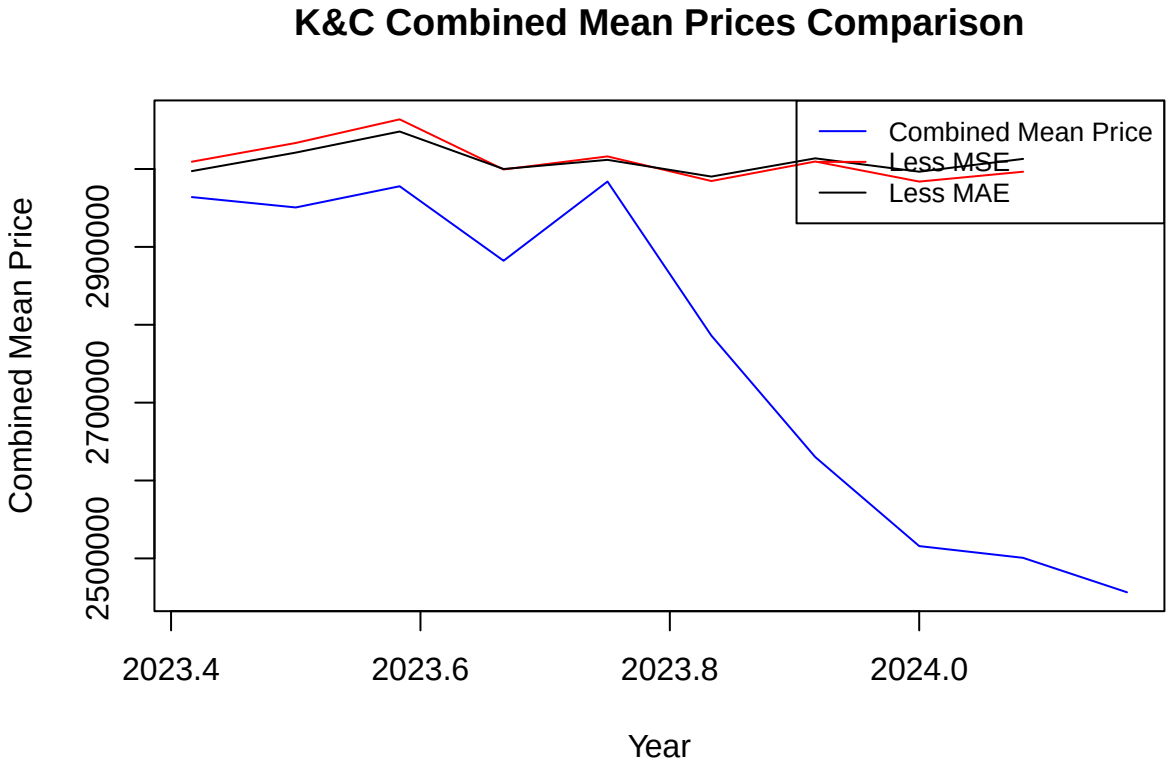
```r
print(paste("Kensington and Chelsea Combined Mean Price MSE for Less MAE Model:"
            , mse_combined_kc_less_MAE))
```

```
## [1] "Kensington and Chelsea Combined Mean Price MSE for Less MAE Model: 78671284799.6368"
```

```r
print(paste("Kensington and Chelsea Combined Mean Price MAE for Less MAE Model:"
            , mae_combined_kc_less_MAE))
```

```
## [1] "Kensington and Chelsea Combined Mean Price MAE for Less MAE Model: 211171.41153049"
```

```r
# Plot all combined series together for Kensington and Chelsea Area
plot(combined_mean_price_kc_test, type = "l", col = "blue",
     main = "K&C Combined Mean Prices Comparison",ylab = "Combined Mean Price",
     xlab = "Year", ylim = range(c(combined_mean_price_kc_test,
                                   combined_mean_price_kc_less_MSE_test_ts,
                                   combined_mean_price_kc_less_MAE_test_ts)))
lines(combined_mean_price_kc_less_MSE_test_ts, col = "red")
lines(combined_mean_price_kc_less_MAE_test_ts, col = "black")

legend("topright", legend = c("Combined Mean Price", "Less MSE", "Less MAE"),
       col = c("blue", "red", "black"), lty = 1, cex = 0.8)
```


K&C Combined Mean Prices Comparison
Combined Mean Price
Less MSE
Less MAE
Combined Mean Price
2500000
2700000
2900000
2023.4
2023.6
2023.8
2024.0
Year