sdm2_teamproject2

2024-11-28

Team members:

- Tharun Teja Mogili (UB Person Number: 50559877)
- Prathibha Vuyyala (UB Person Number: 50559983)
- Manasa Lakshmi Gunampalli (UB Person Number: 50559593)
- Divyakanth Reddy Buchupalle (UB Person Number: 50559937)

1. You will be using a data set accessed via the link https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data. The data is contained in the oil.csv file.

```
library(ggplot2)
library(dplyr)
```

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

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

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

```
library(zoo)
```

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

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

```
library(forecast)
```

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

```
library(TTR)
```

```
oil_data <- read.csv("oil.csv")
```

Dataset Details:

The dataset contains historical oil price data for Ecuador's economy. This information is critical for understanding economic trends and forecasting retail sales.

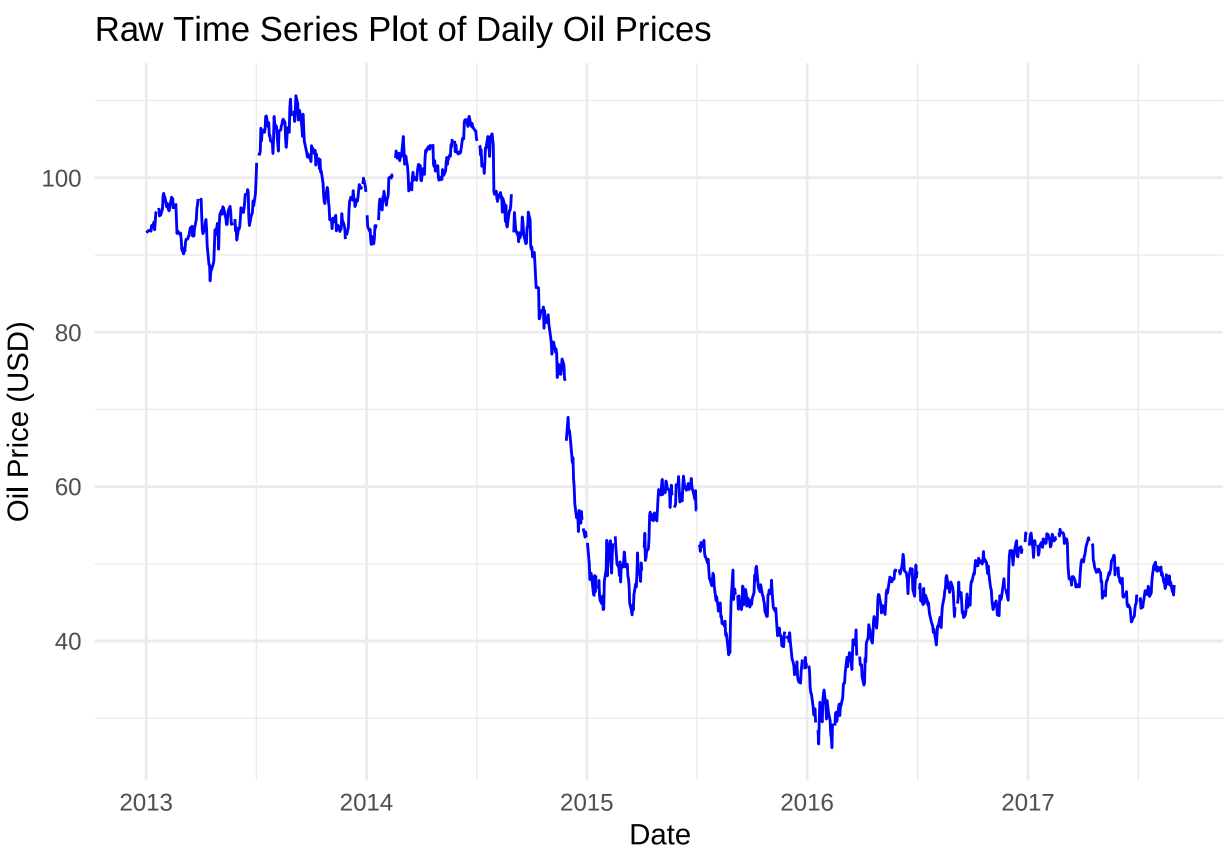
Columns:

- date: Date of the oil price (YYYY-MM-DD).
- dcoilwtico: Daily WTI crude oil price (in USD per barrel). Missing values occur for non-recording days (e.g., holidays).

2. Plot the time series as is.

```r
# Convert the date column to Date type (assuming the column is named 'date')
oil_data$date <- as.Date(oil_data$date)
ggplot(oil_data, aes(x = date, y = dcoilwtico)) +
  geom_line(color = "blue") +
  labs(title = "Raw Time Series Plot of Daily Oil Prices",
       x = "Date",
       y = "Oil Price (USD)") +
  theme_minimal()
```

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



This plot illustrates daily oil price fluctuations from 2013 to 2017, using WTI crude oil as the standard benchmark. A continuous line graph effectively highlights key trends over this timeframe.

Key Features:

- Time Frame: The data spans from early 2013 to mid-2017.
- Price Range: Oil prices ranged from over $100 per barrel in 2013 to a low of approximately $26 in early 2016.

Trends:

- Prices remained stable above $90 per barrel from 2013 to 2014.
- A sharp decline occurred between late 2014 and early 2016, hitting a low point in early 2016.
- A gradual recovery followed, with prices stabilizing near $50 by mid-2017.

Analysis:

The plot highlights significant ***volatility***, particularly during the 2014-2016 price drop. It also reveals occasional gaps in the data, representing ***missing values***. This visualization provides insights into historical oil price trends and their potential economic impacts.

This visualization captures the historical volatility of oil prices and serves as a valuable tool for examining economic impacts during this period.

3. Read the literature and find out how to fill the missing data. Impute the data using your preferred method.

Handling missing data is a critical step in ensuring the accuracy and reliability of time series analysis. Several imputation methods are available, each with its own strengths and limitations, depending on the nature of the dataset and the analysis objectives. Below is an overview of the methods considered for this dataset and the rationale for selecting the most suitable approach:

Linear Interpolation:

In the method, Missing values are estimated by drawing a straight line between the surrounding data points. This approach is ideal for time series data as it preserves trends and patterns, ensuring a smooth and continuous dataset.

Forward Fill (Last Observation Carried Forward - LOCF):

This method propagates the last observed value forward to fill missing data points.While it maintains continuity, it may introduce bias if the underlying trend changes significantly , this might have been useful when the first value in dataset was Null , since we dont have any Null values in the beginning of the dataset we are not using Forward Fill .

Backward Fill (Next Observation Carried Backward - NOCB):

In this Method, missing values are replaced by the next available data point.It is useful for datasets with gradual changes but can distort trends, particularly in highly volatile series , in our case of the dataset of Oil.csv values at the end of the dataset are not null , so there is no need for us to use the Backward Fill .

Recommended Method: Linear Interpolation

Here we have used ***Linear interpolation*** for imputing missing values in this dataset due to its ability to maintain the continuity and inherent trends of oil prices over time. By estimating missing values based on the surrounding data, this method minimizes the risk of introducing bias or distorting the overall trend.

This approach ensures that the imputed data aligns closely with the original series, facilitating accurate analysis and reliable forecasting. Its effectiveness in addressing missing data makes it particularly well-suited for time series datasets where gradual changes and patterns are critical to understanding and modeling the underlying behavior.

```
sum(is.na(oil_data$dcoilwtico))
```

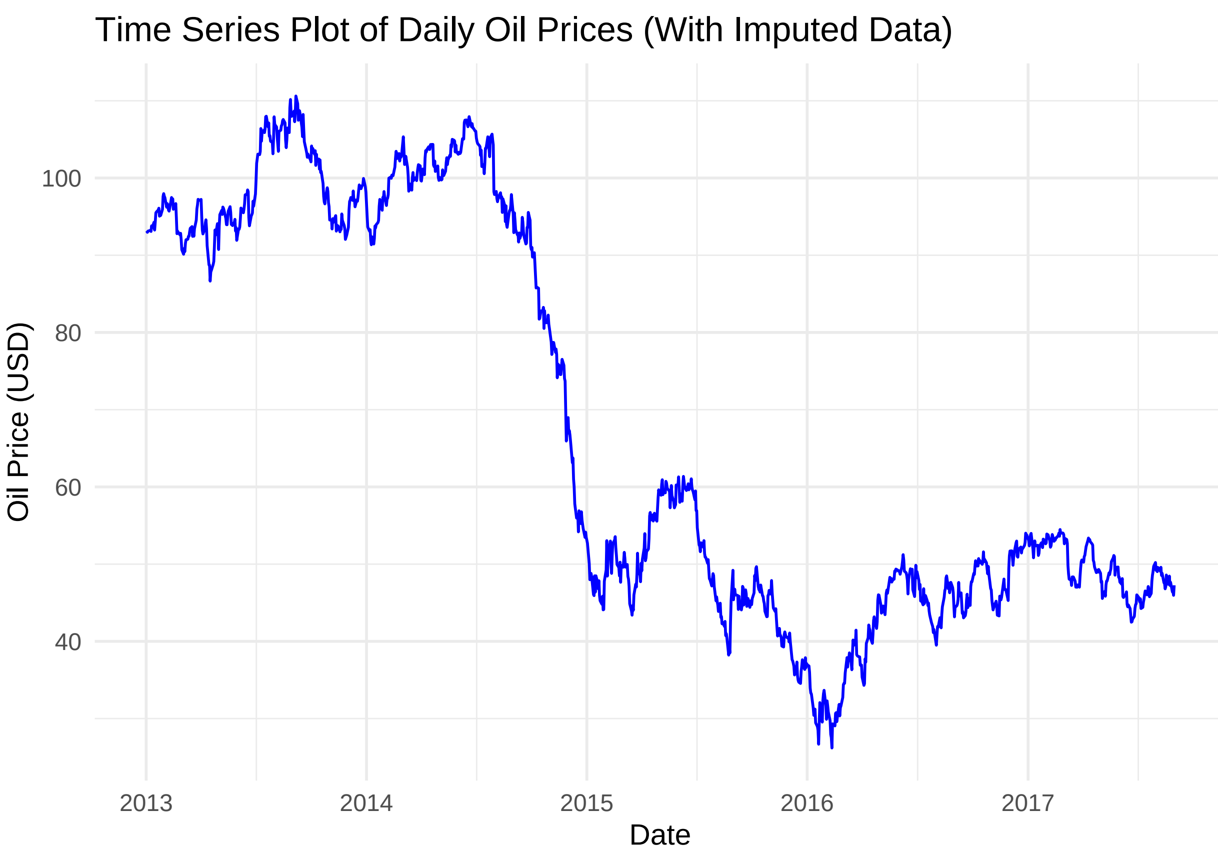
```
## [1] 43
```

```
oil_data <- oil_data %>%
  arrange(date) %>%
  mutate(dcoilwtico = na.approx(dcoilwtico, na.rm = FALSE))
```

4. Plot the time series with imputed data. Do you see a trend and/or seasonality in the data?

```
# Plot the time series with imputed data
ggplot(oil_data, aes(x = date, y = dcoilwtico)) +
  geom_line(color = "blue") +
  labs(title = "Time Series Plot of Daily Oil Prices (With Imputed Data)",
       x = "Date",
       y = "Oil Price (USD)") +
  theme_minimal()
```

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



Observations:

- ***Trend*** : The plot shows a clear downward trend from 2014 to early 2016, followed by a gradual recovery.
- ***Seasonality*** : There is no distinct seasonal pattern visible in the data.Oil prices are primarily driven by global economic factors, geopolitical events, and supply-demand dynamics, which may not follow regular seasonal cycles.
- Volatility: Significant price fluctuations are evident, especially during the sharp decline in 2014-2016.

Imputation Method:

- Linear Interpolation: This method was chosen to maintain continuity and accurately reflect trends without introducing bias. It effectively fills gaps by estimating values between known data points.This approach ensures a comprehensive analysis of historical oil price trends and supports accurate forecasting.

5.Learn about the ETS models and about Holt-Winters models (provide all relevant specifics with respect to theoretical aspects and running them). This will expand your toolkit of the candidate models.

ETS Models:

The ETS (Error-Trend-Seasonality) models are a comprehensive framework for time series forecasting.They are characterized by their decomposition into three components: error, trend, and seasonality.These models are particularly effective for handling non-stationary data with trends and seasonal patterns.

Components:

- Error (E): Represents random variations or noise in the data. It can be modeled as additive (A) or multiplicative (M).
- Trend (T): Captures the long-term progression in the data. Options include none (N), additive (A), additive damped (Ad), multiplicative (M), or multiplicative damped (Md).
- Seasonality (S): Accounts for repeating patterns over fixed periods and can be none (N), additive (A), or multiplicative (M).

Key Features:

- Automatic Selection: ETS models can automatically select the best combination of components based on the data characteristics.
- Flexibility: They are suitable for data exhibiting trends, seasonality, or both.
- State Space Formulation: Each model is represented in a state space form, allowing for efficient estimation and forecasting.

Model Notation:

An ETS model is denoted as ETS(E,T,S).

Some of the examples of ETS Models:

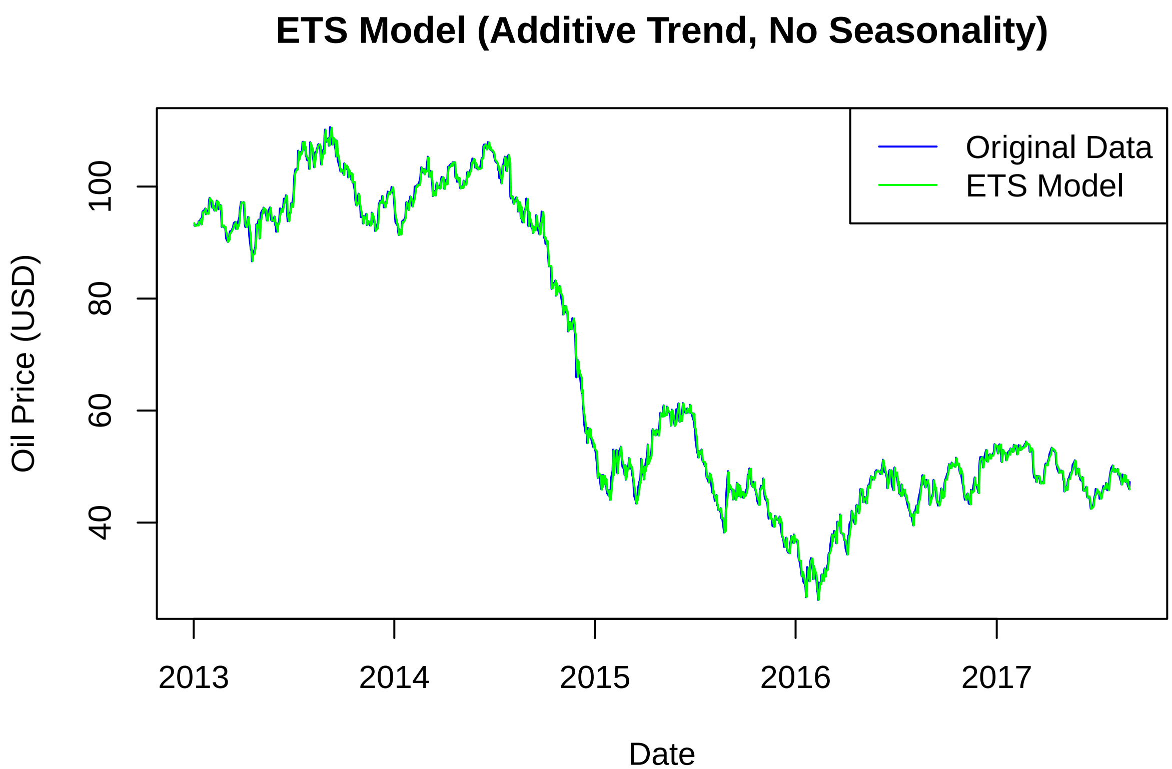
- ETS(A,N,N): A model with additive error, no trend, and no seasonality.
- ETS(A,A,N): Incorporates additive error and trend, but no seasonality.

- ETS(A,A,A): Includes additive error, trend, and seasonality, suitable for complex time series with varying levels and seasonal patterns15.
- ETS(N,N,N): This model assumes no error, no trend, and no seasonality. It represents a completely flat model without any systematic changes or noise.

Properties:

- Additive models assume constant variance, while multiplicative models can handle heteroscedasticity (variance changes with the level of the series).
- Damped trends are used to model slowing growth or decline over time.

```
oil_data$date <- as.Date(oil_data$date)
oil_data$dcoilwtico <- na.approx(oil_data$dcoilwtico, na.rm = FALSE)
oil_data_clean <- na.omit(oil_data)
ets_model <- ets(oil_data_clean$dcoilwtico, model = "AAN")
plot(oil_data_clean$date, oil_data_clean$dcoilwtico, type = "l", col = "blue", xlab = "Date",
     ylab = "Oil Price (USD)", main = "ETS Model (Additive Trend, No Seasonality)")
lines(oil_data_clean$date, fitted(ets_model), col = "green")
legend("topright", legend = c("Original Data", "ETS Model"), col = c("blue", "green"), lty = 1)
```



Holt-Winters Models:

The Holt-Winters method, also known as triple exponential smoothing, is a popular forecasting technique that extends exponential smoothing to account for both trend and seasonal variations in time series data.

Components:

- Level: The baseline value of the series at time t.
- Trend: Represents the slope or direction of the series.

- Seasonality : Captures cyclical behavior within the data.

Smoothing Parameters:

- Alpha: Controls the smoothing of the level component.
- Beta: Controls the smoothing of the trend component.
- Gamma: Controls the smoothing of the seasonal component.

Types of Holt-Winters Models

Additive Holt-Winters:

Suitable for data with constant seasonal variations.

```r
cat("Model Form: \n","yt = lt + bt + st + et\n\n")
```

```
## Model Form:
##  yt = lt + bt + st + et
```

The seasonal component is added to the level and trend components.

Multiplicative Holt-Winters:

Suitable for data where seasonal variations change proportionally with the level.

```r
cat("Model Form: \n","yt = (lt + bt) * st * et\n\n")
```

```
## Model Form:
##  yt = (lt + bt) * st * et
```

The seasonal component multiplies the level and trend components, making it ideal for series where seasonal effects increase with the level.

Key Features:

- Versatility: Effective for time series with both trend and seasonality.
- Manual Specification: Requires defining the length of the seasonal cycle (e.g., quarterly or monthly).
- Damping: Both additive and multiplicative methods can incorporate damping to model slowing trends over time.

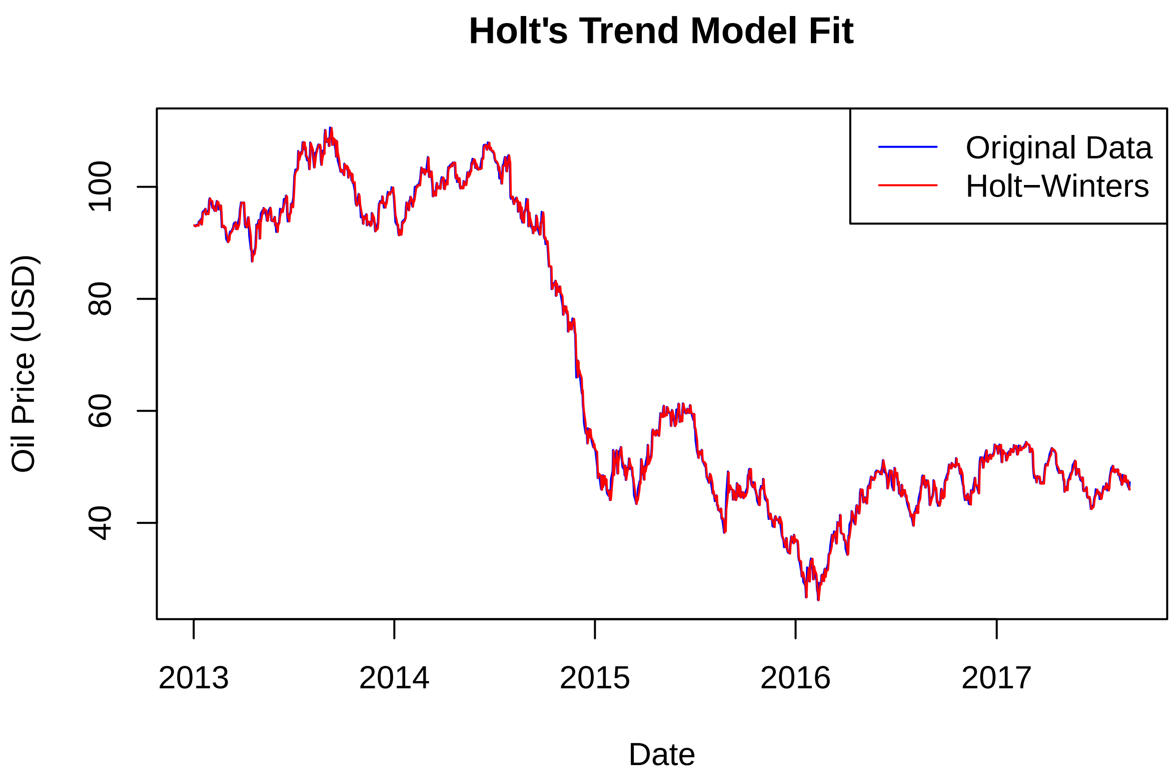
Properties:

- Additive Method: Preferred when seasonal variations are roughly constant throughout the series. It expresses seasonality in absolute terms, adjusting forecasts by adding or subtracting seasonal effects.
- Multiplicative Method: Preferred when seasonal variations change proportionally with the level of the series. It expresses seasonality as a percentage, adjusting forecasts by multiplying or dividing by seasonal factors.

The Holt-Winters models are powerful tools for generating accurate forecasts in time series data exhibiting trends and seasonality, making them widely used in various applications such as sales forecasting, inventory management, and economic forecasting.

```r
oil_data_clean <- na.omit(oil_data)

# Holt-Winters Model
holt_model <- holt(oil_data_clean$dcoilwtico, h = 30, damped = FALSE)
plot(oil_data_clean$date, oil_data_clean$dcoilwtico, type = "l", col = "blue", xlab = "Date", ylab = "O
lines(oil_data_clean$date, fitted(holt_model), col = "red")
legend("topright", legend = c("Original Data", "Holt-Winters"), col = c("blue", "red"), lty = 1)
```



6. Based on your answer to the question 4, suggest suitable model(s) for the data.

Based on our dataset of oil.csv these are the models that we found to be suitable :

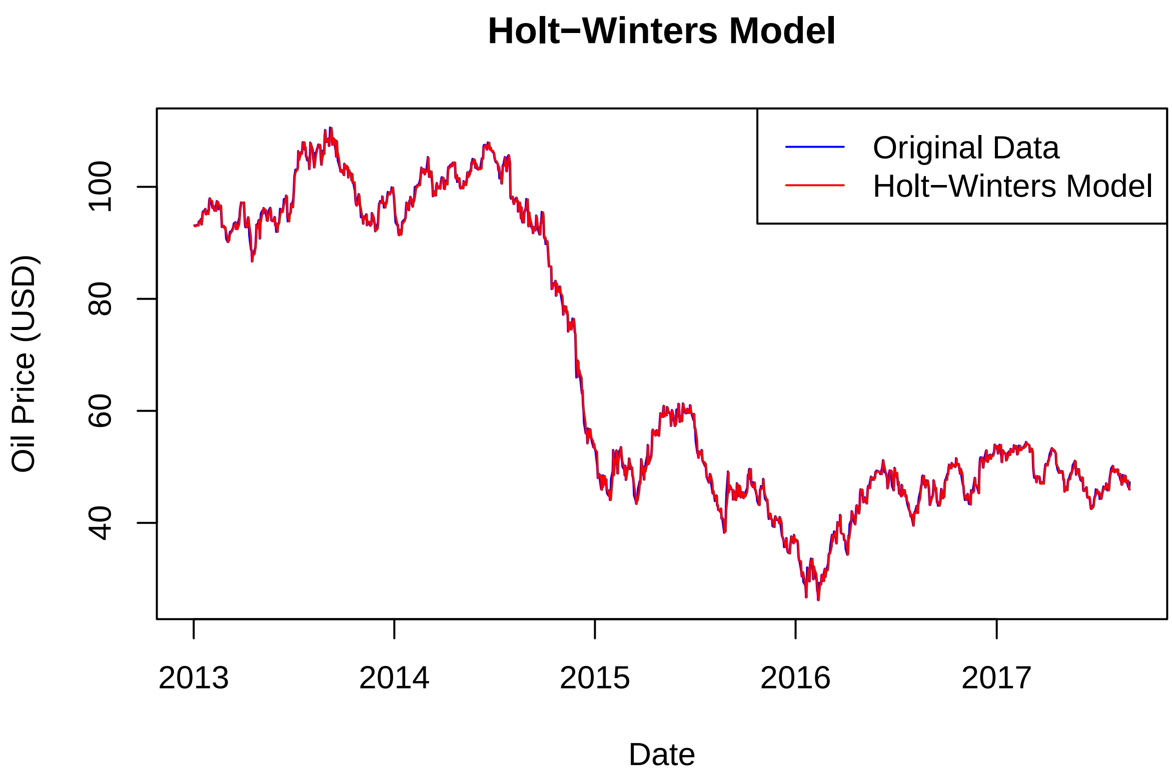
1. Holt-Winters Model
2. ETS Model (Additive Trend, No Seasonality)
3. ARIMA Model
4. Simple Moving Average (SMA)
5. Weighted Moving Average (WMA)

Holt-Winters Model:

- Holt's Linear Trend Model is designed to capture linear trends in time series data, making it suitable for datasets with a clear upward or downward trajectory.

```
oil_data_clean <- na.omit(oil_data)

# Holt-Winters Model
holt_model <- holt(oil_data_clean$dcoilwtico, h = 30, damped = FALSE)
plot(oil_data_clean$date, oil_data_clean$dcoilwtico, type = "l", col = "blue", xlab = "Date", ylab = "Oi
lines(oil_data_clean$date, fitted(holt_model), col = "red")
legend("topright", legend = c("Original Data", "Holt-Winters Model"), col = c("blue", "red"), lty = 1)
```

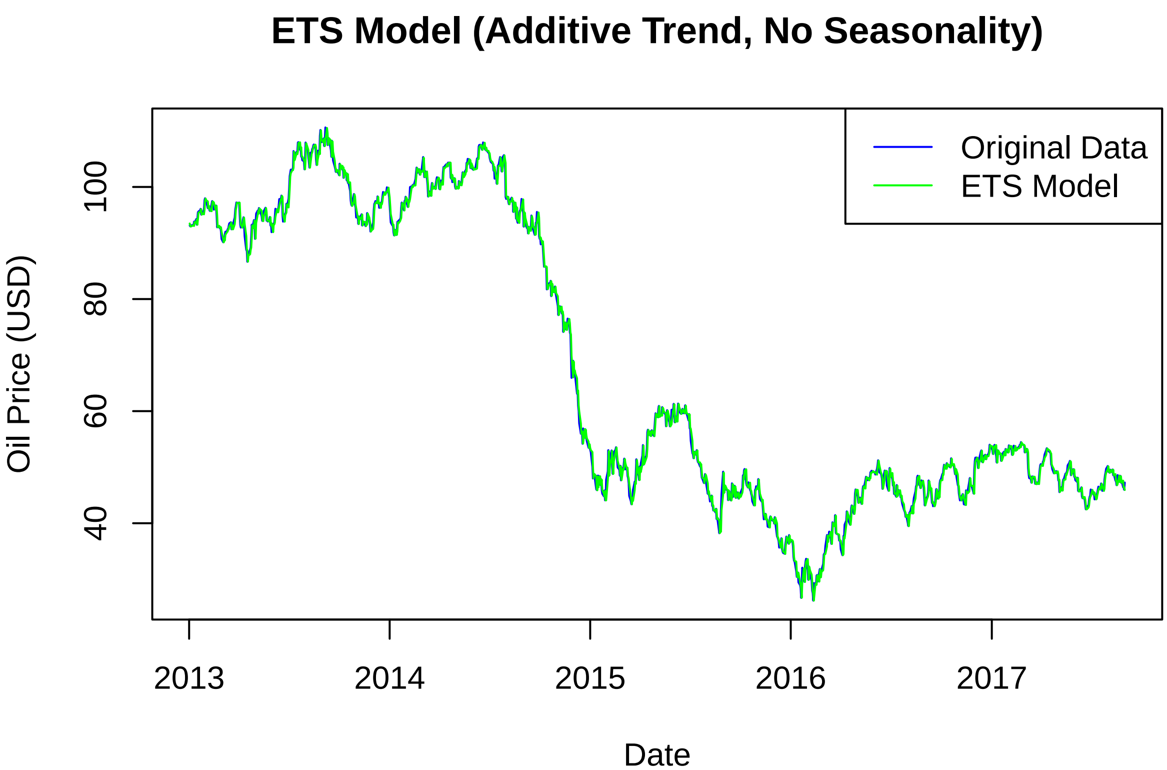


- Graph Insights: This model effectively follows the overall trend in the data but may not capture short-term fluctuations or volatility.
- Use Case: Best for datasets with a clear linear trend and minimal volatility.

ETS Model:

- The ETS model with an additive trend component and no seasonality is ideal for data exhibiting a clear trend without regular seasonal patterns.

```
oil_data$dcoilwtico <- na.approx(oil_data$dcoilwtico, na.rm = FALSE)
oil_data_clean <- na.omit(oil_data)
ets_model <- ets(oil_data_clean$dcoilwtico, model = "AAN")
plot(oil_data_clean$date, oil_data_clean$dcoilwtico, type = "l", col = "blue", xlab = "Date",
     ylab = "Oil Price (USD)", main = "ETS Model (Additive Trend, No Seasonality)")
lines(oil_data_clean$date, fitted(ets_model), col = "green")
legend("topright", legend = c("Original Data", "ETS Model"), col = c("blue", "green"), lty = 1)
```

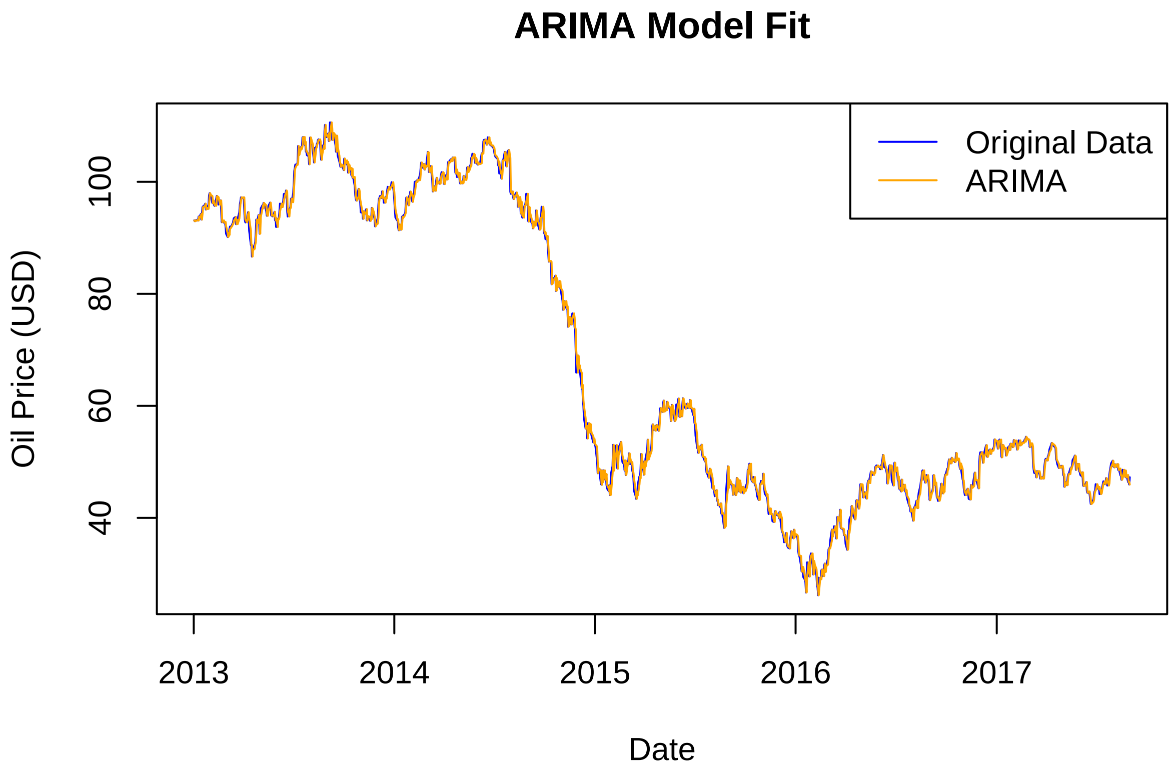


- Graph Insights: The ETS model captures the overall trend well, aligning closely with the original data. - Use Case: Ideal for datasets with irregular trends where automatic model selection is preferred.

ARIMA Model:

- ARIMA (AutoRegressive Integrated Moving Average) models are versatile and can handle various types of time series data, including those with trends.

```
oil_data$dcoilwtico <- na.approx(oil_data$dcoilwtico, na.rm = FALSE)
oil_data_clean <- na.omit(oil_data)
arima_model <- auto.arima(oil_data_clean$dcoilwtico)
plot(oil_data_clean$date, oil_data_clean$dcoilwtico, type = "l", col = "blue", xlab = "Date",
     ylab = "Oil Price (USD)", main = "ARIMA Model Fit")
lines(oil_data_clean$date, fitted(arima_model), col = "orange")
legend("topright", legend = c("Original Data", "ARIMA"), col = c("blue", "orange"), lty = 1)
```

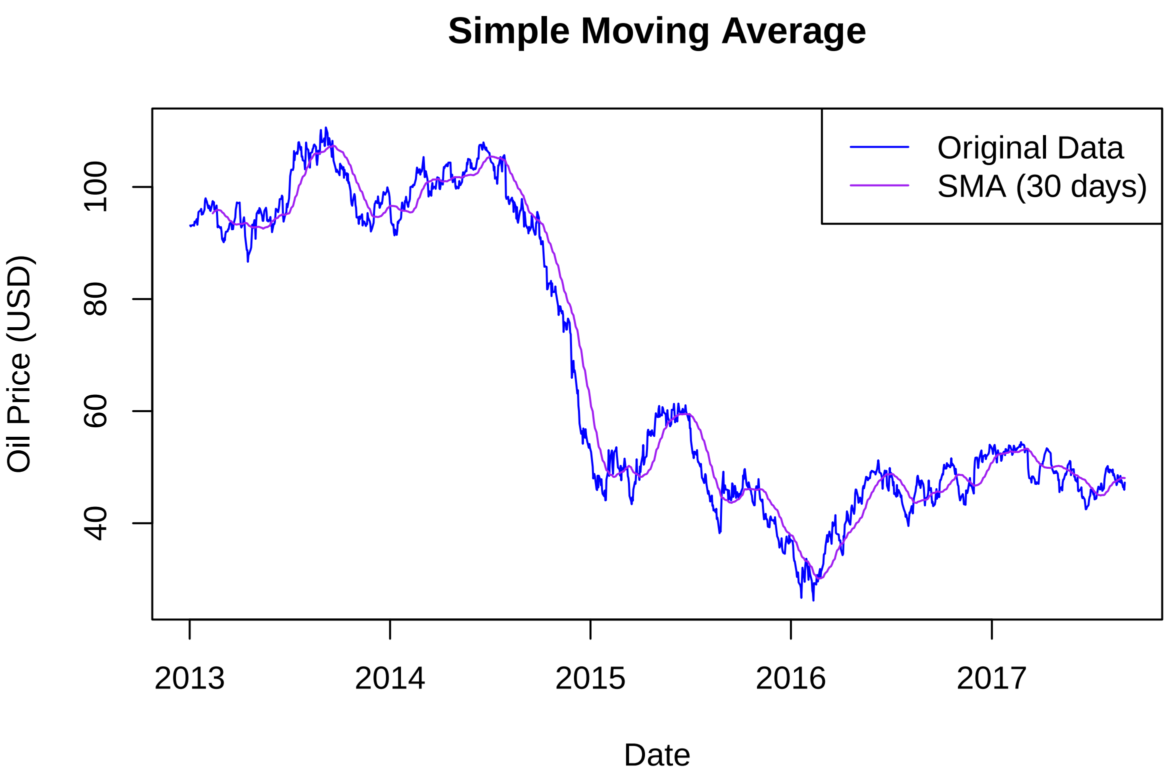


- Graph Insights: The ARIMA fit closely aligns with the original data, effectively modeling both long-term trends and short-term variations.
- Use Case: Suitable for datasets with significant trends and variability, offering robust forecasting capabilities.

Simple Moving Average (SMA)

- SMA smooths out short-term fluctuations by averaging data points over a specified period.

```
oil_data$dcoilwtico <- na.approx(oil_data$dcoilwtico, na.rm = FALSE)
sma <- SMA(oil_data$dcoilwtico, n = 30)
plot(oil_data$date, oil_data$dcoilwtico, type = "l", col = "blue", xlab = "Date",
     ylab = "Oil Price (USD)", main = "Simple Moving Average")
lines(oil_data$date, sma, col = "purple")
legend("topright", legend = c("Original Data", "SMA (30 days)"), col = c("blue", "purple"), lty = 1)
```

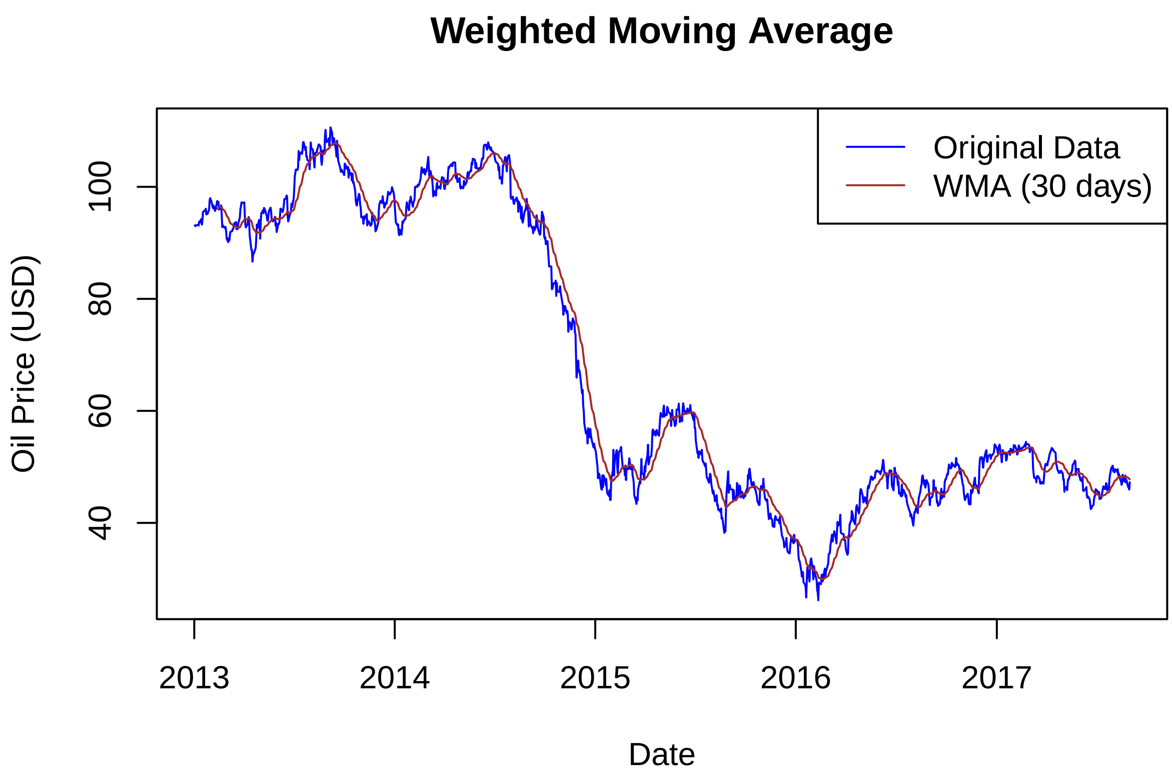


- Graph Insights: The SMA provides a smoothed trend but lags behind rapid changes in the data compared to WMA.
- Use Case: Best for identifying long-term trends without focusing on recent fluctuations.

Weighted Moving Average (WMA):

- WMA assigns different weights to past observations, giving more importance to recent data points.

```
wma <- WMA(oil_data$dcoilwtico, n = 30)
plot(oil_data$date, oil_data$dcoilwtico, type = "l", col = "blue", xlab = "Date", ylab = "Oil Price (US
lines(oil_data$date, wma, col = "brown")
legend("topright", legend = c("Original Data", "WMA (30 days)"), col = c("blue", "brown"), lty = 1)
```



- Graph Insights: The WMA closely follows the original data, smoothing short-term fluctuations while being more responsive than SMA.
- Use Case: Ideal for datasets where recent data points are more relevant, such as financial markets with rapid changes.

Best model is ARIMA Model - The ARIMA model provides the best balance between capturing long-term trends and short-term fluctuations. It is flexible and can be tailored to handle various time series characteristics without assuming seasonality, which aligns well with the observed data patterns.

7. Run the models and check their adequacy.

To check the adequacy of our models we use these tests that us help evaluate how well each model captures the underlying patterns in the data and whether there are any systematic patterns left in the residuals.

- Ljung-Box Test: Tests whether any group of autocorrelations of a time series is different from zero. A significant p-value suggests that there is autocorrelation in the residuals.
- Shapiro-Wilk Test: Tests the normality of the residual distribution. A significant p-value indicates that the residuals deviate from a normal distribution.

1. Holt's Model:

```r
# Calculate residuals
holt_residuals <- residuals(holt_model)

# Ljung-Box test
ljung_box_holt <- Box.test(holt_residuals, lag = 20, type = "Ljung-Box")
print("Ljung-Box Test for Holt's Linear Trend Model:")
```

```
## [1] "Ljung-Box Test for Holt's Linear Trend Model:"
```

```r
print(ljung_box_holt)
```

```
##
##  Box-Ljung test
##
## data:  holt_residuals
## X-squared = 11.398, df = 20, p-value = 0.9352
```

```r
# Shapiro-Wilk test
shapiro_holt <- shapiro.test(holt_residuals)
print("Shapiro-Wilk Test for Holt's Linear Trend Model:")
```

```
## [1] "Shapiro-Wilk Test for Holt's Linear Trend Model:"
```

```r
print(shapiro_holt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  holt_residuals
## W = 0.99094, p-value = 8.08e-07
```

- Suitability: This model is appropriate for data with a clear linear trend but no seasonality. It effectively captures the level and trend components using exponential smoothing.
- Tests: The Ljung-Box test indicates no significant autocorrelation in residuals (p-value = 0.9352), while the Shapiro-Wilk test suggests non-normality (p-value = 8.08e-07).

2. ETS Model (Additive Trend, No Seasonality):

```r
# Calculate residuals
ets_residuals <- residuals(ets_model)

# Ljung-Box test
ljung_box_ets <- Box.test(ets_residuals, lag = 20, type = "Ljung-Box")
print("Ljung-Box Test for ETS Model:")
```

```
## [1] "Ljung-Box Test for ETS Model:"
```

```r
print(ljung_box_ets)
```

```
##
##  Box-Ljung test
##
## data:  ets_residuals
## X-squared = 11.384, df = 20, p-value = 0.9356
```

```r
# Shapiro-Wilk test
shapiro_ets <- shapiro.test(ets_residuals)
print("Shapiro-Wilk Test for ETS Model:")
```

```
## [1] "Shapiro-Wilk Test for ETS Model:"
```

```r
print(shapiro_ets)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ets_residuals
```

```
## W = 0.99096, p-value = 8.269e-07
```

- Suitability: Ideal for data exhibiting a clear trend without regular seasonal patterns, this model captures the additive trend component effectively.
- Tests: The Ljung-Box test shows no significant autocorrelation (p-value = 0.9356), and the Shapiro-Wilk test indicates non-normality (p-value = 8.269e-07).

3. ARIMA Model:

```r
# Calculate residuals
arima_residuals <- residuals(arima_model)

# Ljung-Box test
ljung_box_arima <- Box.test(arima_residuals, lag = 20, type = "Ljung-Box")
print("Ljung-Box Test for ARIMA Model:")
```

```
## [1] "Ljung-Box Test for ARIMA Model:"
```

```r
print(ljung_box_arima)
```

```
##
##  Box-Ljung test
##
## data:  arima_residuals
## X-squared = 12.459, df = 20, p-value = 0.8994
```

```r
# Shapiro-Wilk test
shapiro_arima <- shapiro.test(arima_residuals)
print("Shapiro-Wilk Test for ARIMA Model:")
```

```
## [1] "Shapiro-Wilk Test for ARIMA Model:"
```

```r
print(shapiro_arima)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  arima_residuals
## W = 0.99078, p-value = 6.424e-07
```

- Suitability: ARIMA models are versatile and can handle various types of time series data, including those with trends. They capture both autoregressive and moving average components.
- Tests: The Ljung-Box test suggests no significant autocorrelation (p-value = 0.8994), but the Shapiro-Wilk test indicates non-normality (p-value = 6.424e-07).

4. Simple Moving Average (SMA):

```r
# Calculate residuals (ignoring initial NA values due to window size)
sma_residuals <- oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - sma[30:length(sma)]
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
# Ljung-Box test
ljung_box_sma <- Box.test(sma_residuals, lag = 20, type = "Ljung-Box")
print("Ljung-Box Test for SMA:")
```

```
## [1] "Ljung-Box Test for SMA:"
```

```r
print(ljung_box_sma)
```

```
##
##  Box-Ljung test
##
## data:  sma_residuals
## X-squared = 7538.9, df = 20, p-value < 2.2e-16
```

```r
# Shapiro-Wilk test
shapiro_sma <- shapiro.test(sma_residuals)
print("Shapiro-Wilk Test for SMA:")
```

```
## [1] "Shapiro-Wilk Test for SMA:"
```

```r
print(shapiro_sma)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sma_residuals
## W = 0.92626, p-value < 2.2e-16
```

- Suitability: SMA smooths out short-term fluctuations by averaging data points over a specified period. It provides a smoothed view of overall price movement but may not capture trends effectively.
- Tests: The Ljung-Box test reveals significant autocorrelation (p-value $<$ 2.2e-16), and the Shapiro-Wilk test indicates non-normality (p-value $<$ 2.2e-16).

5. Weighted Moving Average (WMA):

```r
# Calculate residuals (ignoring initial NA values due to window size)
wma_residuals <- oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - wma[30:length(wma)]
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
# Ljung-Box test
ljung_box_wma <- Box.test(wma_residuals, lag = 20, type = "Ljung-Box")
print("Ljung-Box Test for WMA:")
```

```
## [1] "Ljung-Box Test for WMA:"
```

```r
print(ljung_box_wma)
```

```
##
##  Box-Ljung test
##
## data:  wma_residuals
## X-squared = 4378.8, df = 20, p-value < 2.2e-16
```

```r
# Shapiro-Wilk test
shapiro_wma <- shapiro.test(wma_residuals)
print("Shapiro-Wilk Test for WMA:")
```

```
## [1] "Shapiro-Wilk Test for WMA:"
```

```r
print(shapiro_wma)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  wma_residuals
## W = 0.88255, p-value < 2.2e-16
```

- Suitability: WMA assigns different weights to past observations, emphasizing recent data points. It can highlight recent trends but may not fully capture long-term trends or seasonality.
- Tests: The Ljung-Box test shows significant autocorrelation (p-value < 2.2e-16), and the Shapiro-Wilk test indicates non-normality (p-value < 2.2e-16).

For this dataset, models that emphasize trend components, such as Holt's Linear Trend Model and the ETS model with an additive trend, are likely more effective due to the observed long-term progression in oil prices without strong seasonal patterns. ARIMA can also be considered for its flexibility in handling various data structures.

8. Compare the models' performance by the metrics that you think are relevant. Try to identify a model with a low RMSE.

To compare the performance of the models based on relevant metrics such as RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error), we can evaluate which model provides the most accurate predictions.

```r
# RMSE for Holt-Winters Model
holt_rmse <- sqrt(mean((oil_data_clean$dcoilwtico - fitted(holt_model))^2))
print(paste("Holt-Winters RMSE:", holt_rmse))
```

```
## [1] "Holt-Winters RMSE: 1.17607497276414"
```

```r
#MAE for Holt-Winters  Model
mae_holt <- mean(abs(oil_data_clean$dcoilwtico - fitted(holt_model)))
print(paste("Holt-Winters MAE:", mae_holt))
```

```
## [1] "Holt-Winters MAE: 0.895398177117463"
```

```r
# MAPE for Holt-Winters Model
mape_holt <- mean(abs((oil_data_clean$dcoilwtico - fitted(holt_model)) / oil_data_clean$dcoilwtico)) * 
print(paste("Holt-Winters MAPE:", mape_holt))
```

```
## [1] "Holt-Winters MAPE: 1.54701831101658"
```

Holt's Linear Trend Model provides a balanced approach to forecasting with a relatively low RMSE and MAE, indicating it captures both trend and level effectively in the data. The MAPE is also low, suggesting good percentage accuracy.

```r
# RMSE for ETS Model
ets_rmse <- sqrt(mean((oil_data_clean$dcoilwtico - fitted(ets_model))^2))
print(paste("ETS Model RMSE:", ets_rmse))
```

```
## [1] "ETS Model RMSE: 1.17611690218487"
```

```r
# MAE for ETS Model
mae_ets <- mean(abs(oil_data_clean$dcoilwtico - fitted(ets_model)))
print(paste("ETS Model MAE:", mae_ets))
```

```
## [1] "ETS Model MAE: 0.89567915821199"
```

```r
# MAPE for ETS Model
mape_ets <- mean(abs((oil_data_clean$dcoilwtico - fitted(ets_model)) / oil_data_clean$dcoilwtico)) * 10
print(paste("ETS Model MAPE:", mape_ets))
```

```
## [1] "ETS Model MAPE: 1.54731107510737"
```

The ETS Model performs similarly to Holt's Linear Trend Model with very comparable RMSE, MAE, and MAPE values. This indicates that it is equally effective in capturing the underlying patterns in the data, making it a reliable choice for forecasting.

```r
# RMSE for ARIMA Model
arima_rmse <- sqrt(mean((oil_data_clean$dcoilwtico - fitted(arima_model))^2))
print(paste("ARIMA Model RMSE:", arima_rmse))
```

```
## [1] "ARIMA Model RMSE: 1.17723846058683"
```

```r
# MAE for ARIMA Model
mae_arima <- mean(abs(oil_data_clean$dcoilwtico - fitted(arima_model)))
print(paste("ARIMA Model MAE:", mae_arima))
```

```
## [1] "ARIMA Model MAE: 0.895639391909147"
```

```r
# MAPE for ARIMA Model
mape_arima <- mean(abs((oil_data_clean$dcoilwtico - fitted(arima_model)) / oil_data_clean$dcoilwtico))
print(paste("ARIMA Model MAPE:", mape_arima))
```

```
## [1] "ARIMA Model MAPE: 1.54860449830115"
```

The ARIMA model shows slightly higher RMSE compared to Holt's and ETS models, suggesting it might not be as precise in this context. However, its MAE and MAPE are close to those of Holt's and ETS models, indicating its potential effectiveness in certain scenarios.

```r
# RMSE for SMA
sma_rmse <- sqrt(mean((oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - sma[30:length(
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
print(paste("SMA RMSE:", sma_rmse))
```

```
## [1] "SMA RMSE: 4.27029790873737"
```

```r
#  MAE for SMA
mae_sma <- mean(abs(oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - sma[30:length(sma
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
print(paste("SMA MAE:", mae_sma))
```

```
## [1] "SMA MAE: 3.19904190048634"
```

```r
# MAPE for SMA
mape_sma <- mean(abs((oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - sma[30:length(s
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length

## Warning in (oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
print(paste("SMA MAPE:", mape_sma))
```

```
## [1] "SMA MAPE: 5.52123696658253"
```

The SMA model has significantly higher RMSE, MAE, and MAPE values compared to other models, indicating less accuracy in forecasting due to its simplistic nature that does not account for trends or seasonality.

```r
# RMSE for WMA
wma_rmse <- sqrt(mean((oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - wma[30:length(w
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
print(paste("WMA RMSE:", wma_rmse))
```

```
## [1] "WMA RMSE: 3.44574040763888"
```

```r
# MAE for WMA
mae_wma <- mean(abs(oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - wma[30:length(wma
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
print(paste("WMA MAE:", mae_wma))
```

```
## [1] "WMA MAE: 2.53628446954612"
```

```r
# MAPE for WMA
mape_wma <- mean(abs((oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - wma[30:length(wm
```

```
## Warning in oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length

## Warning in (oil_data_clean$dcoilwtico[30:length(oil_data_clean$dcoilwtico)] - :
## longer object length is not a multiple of shorter object length
```

```r
print(paste("WMA MAPE:", mape_wma))
```

```
## [1] "WMA MAPE: 4.35380887388959"
```

The WMA model improves upon SMA by assigning different weights to past observations, resulting in lower RMSE, MAE, and MAPE than SMA but still higher than more complex models like Holt's or ETS.

Among the evaluated models, Holt's Trend Model and the ETS Model exhibit the lowest RMSE values, suggesting they are the most accurate for this dataset. Both models are suitable choices for forecasting tasks where capturing trends and levels accurately is crucial.