Mini Project

Priyanka Bhosale

2024-03-15

Project-3: Using 2 datasets: Home Depot Ad Spending and Google Trends for the analysis.

```
# Load the necessary libraries
library(ggplot2)
library(forecast)
```

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

```
# Set a CRAN mirror manually
options(repos = structure(c(CRAN = "https://cloud.r-project.org/")))
install.packages('prophet')
```

```
##
## The downloaded binary packages are in
##  /var/folders/g_/zw8t9hgn0p197fxcrflghncc0000gn/T//RtmpMKVxb8/downloaded_packages
```

```
library(prophet)
```

```
## Loading required package: Rcpp

## Loading required package: rlang
```

```
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
```

```
# Read the Google Trends data
google_trends = read.csv("homedepot_googletrends.csv")

# Read the Home Depot ad spending data
homedepot_adspend = read.csv("homedepot.adspend.csv")
```

Original Datasets details:

There are 1821 rows and 5 columns in Google Trends CSV dataset. There are 756 rows and 13 columns in Home Depot Ad Spending dataset.

```
str(google_trends)
```

```
## 'data.frame':    1821 obs. of  5 variables:
##  $ X           : int  1820 1819 1818 1817 1816 1815 1814 1813 1812 1811 ...
##  $ date        : chr  "Oct 01 2018" "Sep 30 2018" "Sep 29 2018" "Sep 28 2018" ...
##  $ value       : num  15349 19085 19124 14549 14161 ...
##  $ period      : chr  "2018-10-01" "2018-09-30" "2018-09-29" "2018-09-28" ...
##  $ onediffvalue: num  NA 3736.4 39.1 -4575 -387.9 ...
```

```
str(homedepot_adspend)
```

```
## 'data.frame':    756 obs. of  13 variables:
##  $ TIME.PERIOD             : chr  "WEEK OF OCT 07, 2013 (B)" "WEEK OF OCT 07, 2013 (B)" "WEEK OF OC
##  $ PRODUCT                 : chr  "Home Depot Home Center & Kidde : Combo Vignette" "Home Depot Hom
##  $ TOTAL.DOLS..000.        : num  6.3 5128.7 2.6 6.1 4760.4 ...
##  $ NETWORK.TV.DOLS..000.   : num  0 1465 0 0 1753 ...
##  $ CABLE.TV.DOLS..000.     : num  0 1211.8 2.6 0 1105.8 ...
##  $ SYNDICATION.DOLS..000.  : num  0 171.9 0 0 91.1 ...
##  $ SPOT.TV.DOLS..000.      : num  6.3 348.3 0 6.1 116.9 ...
##  $ MAGAZINES.DOLS..000.    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NATL.NEWSP.DOLS..000.   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NEWSPAPER.DOLS..000.    : num  0 238 0 0 0 ...
##  $ NETWORK.RADIO.DOLS..000. : num  0 1635 0 0 1635 ...
##  $ NAT.SPOT.RADIO.DOLS..000.: num  0 58.3 0 0 58.3 0 0 58.3 0 0 ...
##  $ OUTDOOR.DOLS..000.      : num  0 0 0 0 0 0 0 0 0 0 ...
```

Pre-processing on Google Trends dataset:

There is one missing value, and after dropping this missing row, the dataset now has 1820 rows and 4 columns. The first column is renamed to “id”. Column “date” is a character datatype, so converted this into Date datatype. Columns “date” and “period” hold the same data, i.e., the date. So dropped “period” column to avoid having duplicate and irrelevant columns in this dataset. Converted columns “value” and “onediffvalue” to numeric datatypes.

Pre-processing on Home Depot Ad Spending dataset:

There is no missing value. The column names are renamed to make more sense and meaningful, for eg. DOLS renamed to Dollars, removed (000) from the column names. Column TIME_PERIOD is in a format that cannot be used directly for analysis. It is in the format “WEEK OF OCT 07, 2013 (B)”. So removed WEEK OF and the last (B) from such values and converted this character datatype to Date datatype with a consistent format.

```
sum(is.na(google_trends))
```

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

```
sum(is.na(homedepot_adspend))
```

```
## [1] 0
```

```
# Get location of null values
which(is.na(google_trends))
```

```
## [1] 7285
```

```
# Remove rows with missing values
google_trends = na.omit(google_trends)

# Check if the missing data is deleted. This should print 0 now
sum(is.na(google_trends))
```

```
## [1] 0
```

```
# Add column name to the first column of Google Trends dataset
colnames(google_trends)[1] = "id"

# Rename columns of Home Depot Ad Spending
names(homedepot_adspend) <- c("TIME_PERIOD", "PRODUCT", "TOTAL_DOLLARS", "NETWORK_TV_DOLLARS",
                              "CABLE_TV_DOLLARS", "SYNDICATION_DOLLARS", "SPOT_TV_DOLLARS", "MAGAZINES_DOLLARS",
                              "NATL_NEWSP_DOLLARS", "NEWSPAPER_DOLLARS", "NETWORK_RADIO_DOLLARS",
                              "NAT_SPOT_RADIO_DOLLARS", "OUTDOOR_DOLLARS")

# Check the new column names
colnames(homedepot_adspend)
```

```
##  [1] "TIME_PERIOD"            "PRODUCT"                "TOTAL_DOLLARS"
##  [4] "NETWORK_TV_DOLLARS"     "CABLE_TV_DOLLARS"       "SYNDICATION_DOLLARS"
##  [7] "SPOT_TV_DOLLARS"        "MAGAZINES_DOLLARS"      "NATL_NEWSP_DOLLARS"
## [10] "NEWSPAPER_DOLLARS"      "NETWORK_RADIO_DOLLARS"  "NAT_SPOT_RADIO_DOLLARS"
## [13] "OUTDOOR_DOLLARS"
```

```
# Convert date column to Date format
google_trends$date = as.Date(google_trends$date, format = "%b %d %Y")

# Remove label "WEEK OF" and the last "(B)" from the "TIME PERIOD" column of
# Home Depot Ad Spending dataset
homedepot_adspend$`TIME_PERIOD` = trimws(sub("WEEK OF ", "", homedepot_adspend$`TIME_PERIOD`))
homedepot_adspend$`TIME_PERIOD` = trimws(sub(" \\(B\\)$", "", homedepot_adspend$`TIME_PERIOD`))
homedepot_adspend$`TIME_PERIOD` = as.Date(homedepot_adspend$`TIME_PERIOD`, format = "%b %d, %Y")

# Since we converted date column to Date datatype, "period" column is a duplicate and
# irrelevant column. Drop "period".
drop = c('period')
google_trends = google_trends[,!(names(google_trends) %in% drop)]

# Convert the value and onediffvalue columns to numeric
google_trends$value <- as.numeric(google_trends$value)
google_trends$onediffvalue <- as.numeric(google_trends$onediffvalue)

str(google_trends)
```

```
## 'data.frame':    1820 obs. of  4 variables:
##  $ id          : int  1819 1818 1817 1816 1815 1814 1813 1812 1811 1810 ...
##  $ date        : Date, format: "2018-09-30" "2018-09-29" ...
##  $ value       : num  19085 19124 14549 14161 15195 ...
##  $ onediffvalue: num  3736.4 39.1 -4575 -387.9 1033.2 ...
```

```r
str(homedepot_adspend)
```

```
## 'data.frame':    756 obs. of  13 variables:
##  $ TIME_PERIOD          : Date, format: "2013-10-07" "2013-10-07" ...
##  $ PRODUCT              : chr  "Home Depot Home Center & Kidde : Combo Vignette" "Home Depot Home C
##  $ TOTAL_DOLLARS        : num  6.3 5128.7 2.6 6.1 4760.4 ...
##  $ NETWORK_TV_DOLLARS   : num  0 1465 0 0 1753 ...
##  $ CABLE_TV_DOLLARS     : num  0 1211.8 2.6 0 1105.8 ...
##  $ SYNDICATION_DOLLARS  : num  0 171.9 0 0 91.1 ...
##  $ SPOT_TV_DOLLARS      : num  6.3 348.3 0 6.1 116.9 ...
##  $ MAGAZINES_DOLLARS    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NATL_NEWSP_DOLLARS   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NEWSPAPER_DOLLARS    : num  0 238 0 0 0 ...
##  $ NETWORK_RADIO_DOLLARS : num  0 1635 0 0 1635 ...
##  $ NAT_SPOT_RADIO_DOLLARS: num  0 58.3 0 0 58.3 0 0 58.3 0 0 ...
##  $ OUTDOOR_DOLLARS      : num  0 0 0 0 0 0 0 0 0 0 ...
```

Summarize Google Trends Data:

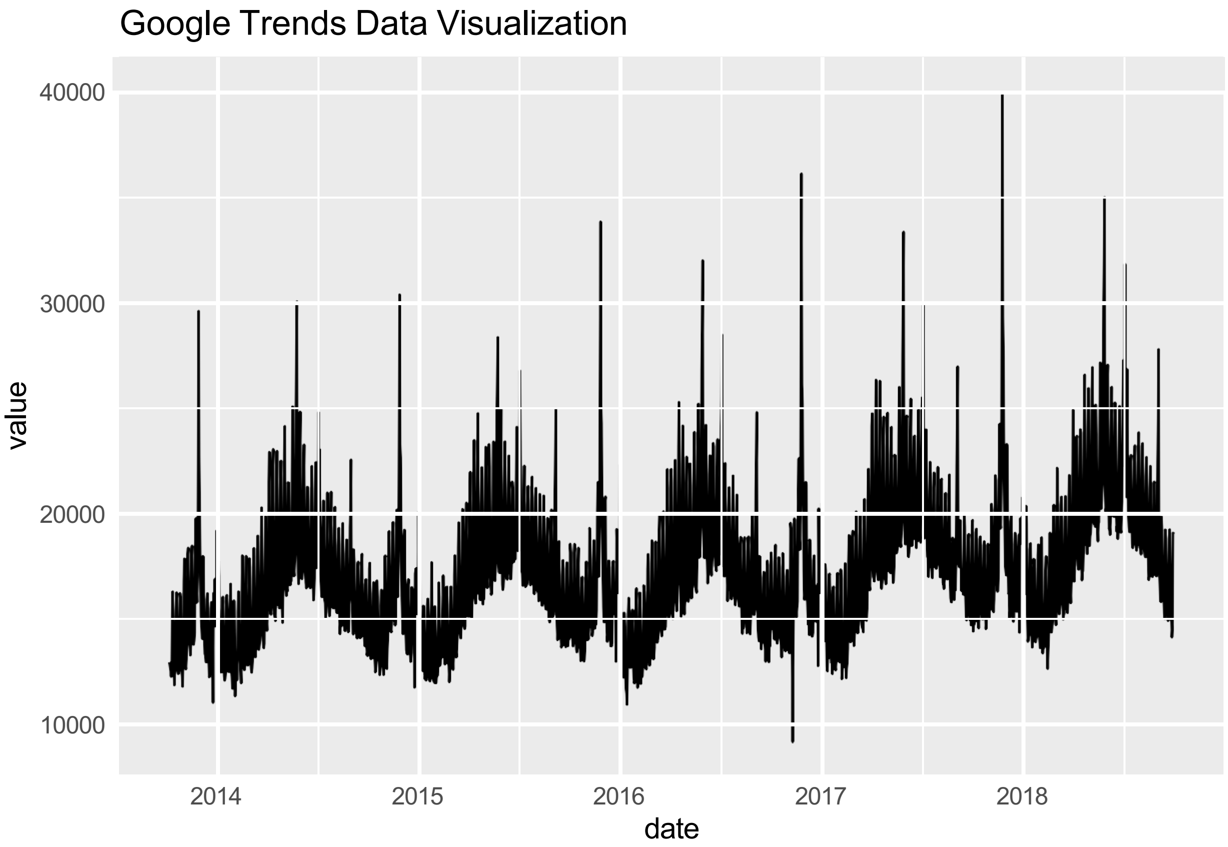
```r
summary(google_trends)
```

```
##        id              date                value         onediffvalue
##  Min.   :   0.0   Min.   :2013-10-07   Min.   : 9205   Min.   :-10129.249
##  1st Qu.: 454.8   1st Qu.:2015-01-04   1st Qu.:14686   1st Qu.:  -892.936
##  Median : 909.5   Median :2016-04-03   Median :16660   Median :    21.625
##  Mean   : 909.5   Mean   :2016-04-03   Mean   :17214   Mean   :    -1.292
##  3rd Qu.:1364.2   3rd Qu.:2017-07-02   3rd Qu.:18861   3rd Qu.:   853.101
##  Max.   :1819.0   Max.   :2018-09-30   Max.   :40141   Max.   : 13500.754
```

Visualize Google Trends Data:

This plot shows the Google Trends data, specifically the trend in the search volume for "Home Depot" over time. The x-axis represents the date, and the y-axis represents the value. The plot shows a clear seasonal pattern, with value peaking around the same time each year, likely due to seasonal factors affecting home improvement projects.

```r
# Visualize the Google Trends data -> Date vs Value
ggplot(data = google_trends, aes(x = date, y = value)) +
  geom_line() +
  labs(title = "Google Trends Data Visualization")
```



```
# Performing adf test for checking the stationarity
library(zoo)
```

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

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

```
library(tseries)

# Perform the ADF test
adf_result <- adf.test(google_trends$value)
```

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

```
# Print the ADF test result
print(adf_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  google_trends$value
## Dickey-Fuller = -5.3037, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

Even though p-value is less than

$$\alpha$$

(where

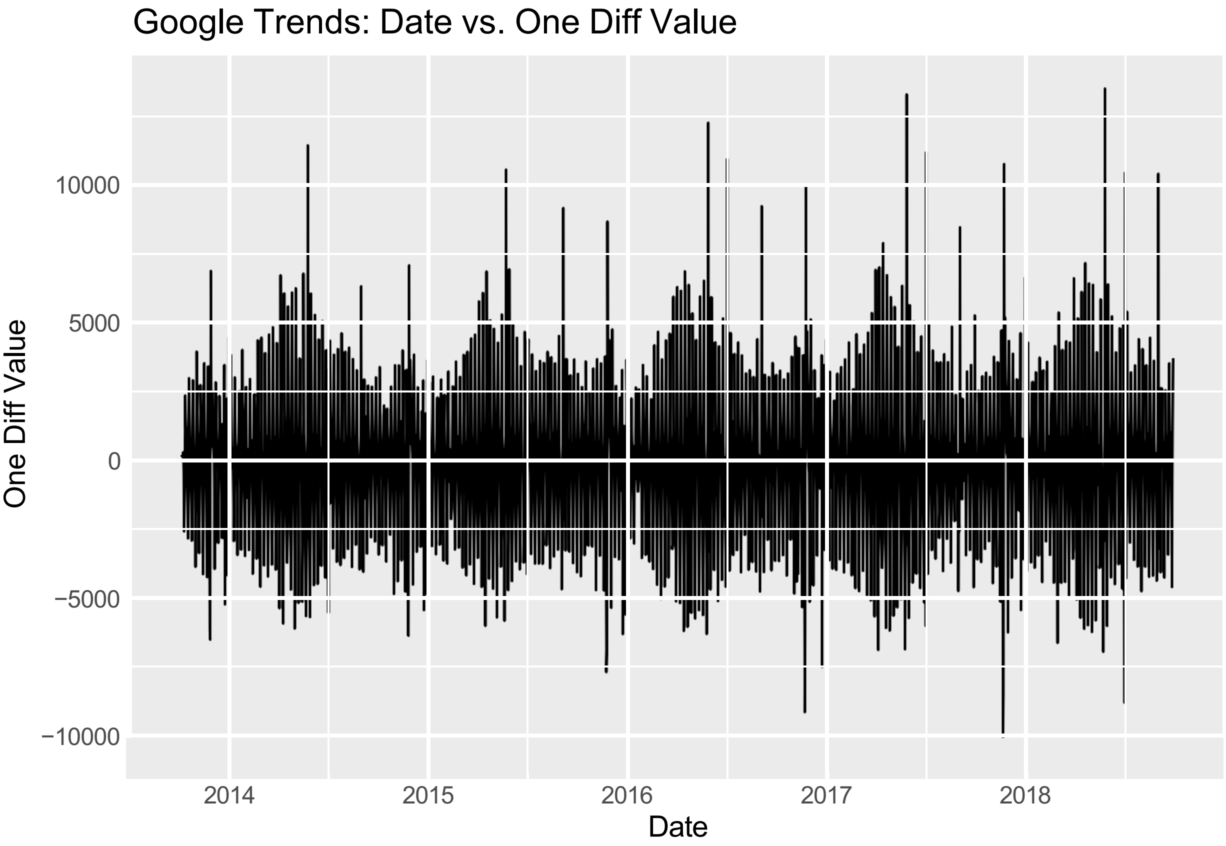
α

=0.05), we can see a significant variance in data. So to get rid off the variance we need to difference the data. There is a differenced data in google trends dataset and is stored in "onediffvalue" column. We now will perform plot the graph for onediffvalue.

Visualize Google Trends Data -> Date vs One Difference Value:

This plot shows the first-order differenced values of the Google Trends data, which helps to remove the trend and make the time series stationary. The x-axis represents the date, and the y-axis represents the differenced value. The plot shows a more stable pattern around a constant mean, indicating that the differencing has removed the trend and seasonality from the original data.

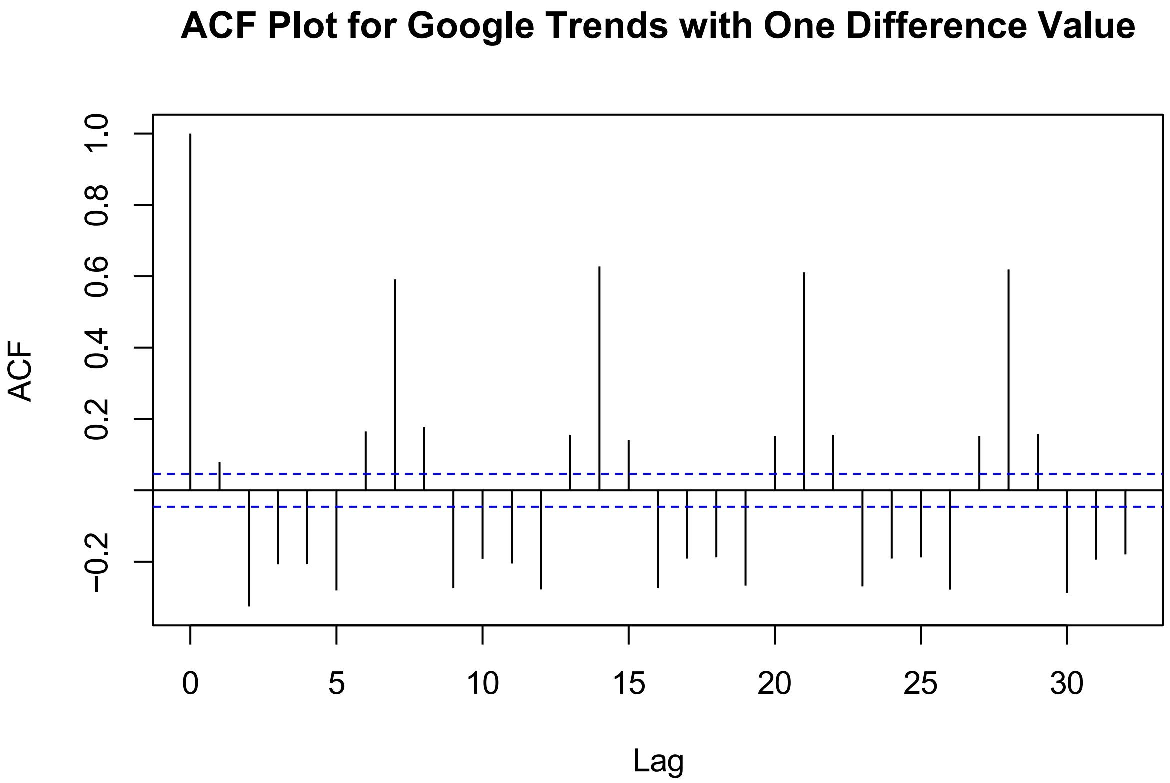
```r
# Line plot for x = date and y = onediffvalue
ggplot(data = google_trends, aes(x = date, y = onediffvalue)) +
  geom_line() +
  labs(title = "Google Trends: Date vs. One Diff Value", x = "Date", y = "One Diff Value")
```



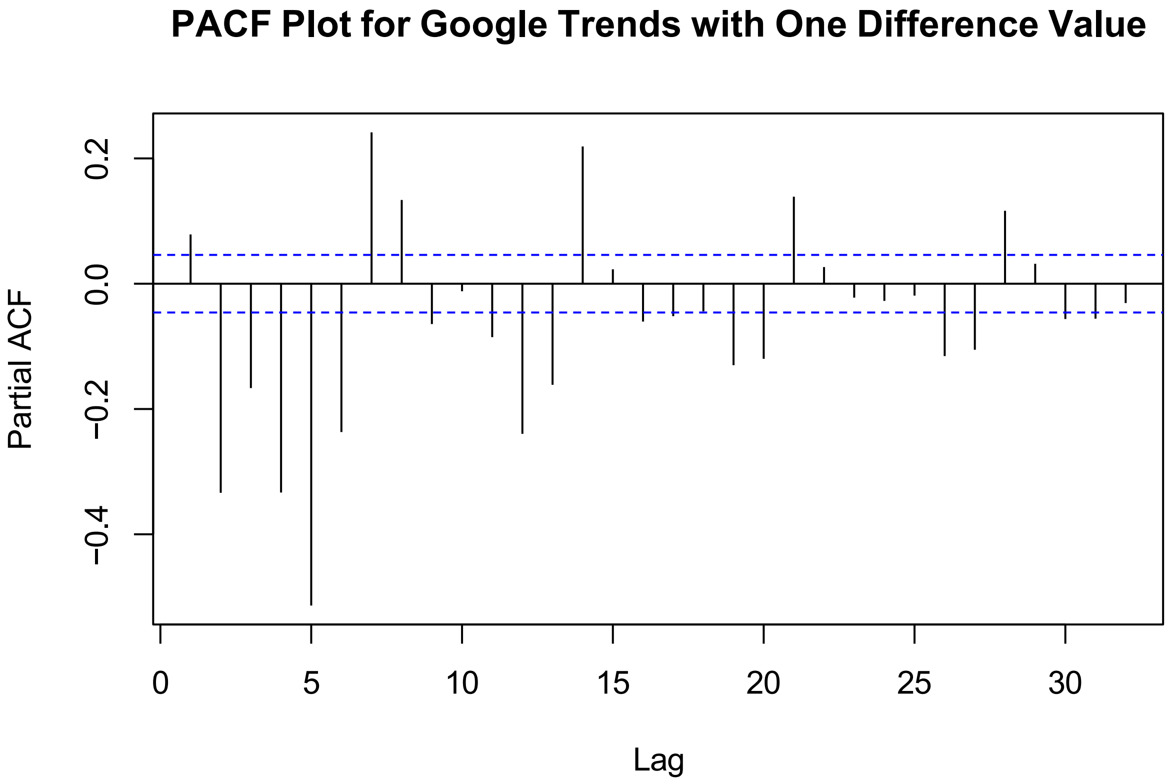
ACF and PACF Plots for Google Trends with One Difference Value:

These plots show the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) for the differenced Google Trends data. The ACF plot shows a significant spike at lag 1, suggesting the presence of a first-order moving average (MA) component in the time series. The PACF plot shows a decaying pattern, indicating the potential presence of an autoregressive (AR) component. These plots help identify the appropriate orders for the ARIMA/SARIMA models.

```
# ACF and PACF plots for Google Trends
acf(google_trends$onediffvalue, main = "ACF Plot for Google Trends with One Difference Value")
```



```
pacf(google_trends$onediffvalue, main = "PACF Plot for Google Trends with One Difference Value")
```



Summarize Home Depot Advertisement Spending Data:

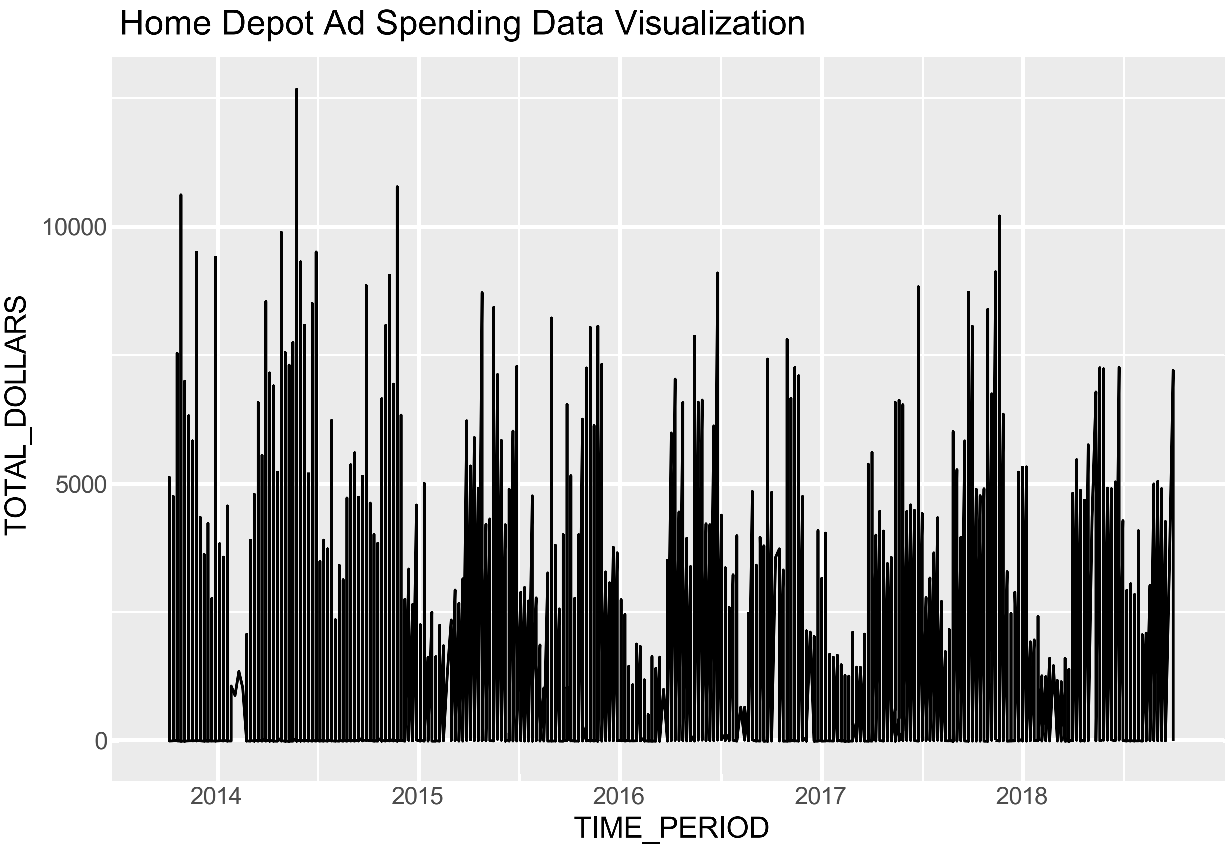
```
summary(homedepot_adspend)
```

```
##    TIME_PERIOD           PRODUCT          TOTAL_DOLLARS     NETWORK_TV_DOLLARS
##  Min.   :2013-10-07   Length:756         Min.   :    0.0   Min.   :   0  
##  1st Qu.:2014-10-20   Class :character   1st Qu.:    4.1   1st Qu.:   0  
##  Median :2015-12-28   Mode  :character   Median :   13.1   Median :   0  
##  Mean   :2016-02-06                      Mean   : 1583.4   Mean   : 398  
##  3rd Qu.:2017-05-02                      3rd Qu.: 2862.2   3rd Qu.:   2  
##  Max.   :2018-10-01                      Max.   :12679.8   Max.   :6558  
##  CABLE_TV_DOLLARS SYNDICATION_DOLLARS SPOT_TV_DOLLARS  MAGAZINES_DOLLARS
##  Min.   :   0.0   Min.   :  0.0       Min.   :  0.00   Min.   :   0.0   
##  1st Qu.:   0.0   1st Qu.:  0.0       1st Qu.:  0.00   1st Qu.:   0.0   
##  Median :  10.3   Median :  0.0       Median :  0.30   Median :   0.0   
##  Mean   : 552.9   Mean   : 12.2       Mean   : 27.76   Mean   : 125.9   
##  3rd Qu.:1058.5   3rd Qu.: 10.5       3rd Qu.: 23.07   3rd Qu.:   0.0   
##  Max.   :4208.1   Max.   :418.3       Max.   :849.70   Max.   :4440.9   
##  NATL_NEWSP_DOLLARS NEWSPAPER_DOLLARS NETWORK_RADIO_DOLLARS
##  Min.   : 0.0000    Min.   :  0.00    Min.   :   0.0       
##  1st Qu.: 0.0000    1st Qu.:  0.00    1st Qu.:   0.0       
##  Median : 0.0000    Median :  0.00    Median :   0.0       
##  Mean   : 0.3985    Mean   : 40.45    Mean   : 372.1       
##  3rd Qu.: 0.0000    3rd Qu.:  2.55    3rd Qu.: 784.2       
##  Max.   :61.3000    Max.   :958.40    Max.   :2485.6       
##  NAT_SPOT_RADIO_DOLLARS OUTDOOR_DOLLARS  
##  Min.   :  0.00         Min.   :  0.000  
##  1st Qu.:  0.00         1st Qu.:  0.000  
##  Median :  0.00         Median :  0.000  
##  Mean   : 51.57         Mean   :  2.043  
##  3rd Qu.: 18.80         3rd Qu.:  0.000  
##  Max.   :848.40         Max.   :226.300  
```

Visualize Home Depot Advertisement Spending Data -> Time Period vs Total Dollars:

This plot shows the Home Depot advertisement spending over time, represented by the Total Dollars spent on advertising. The x-axis represents the time period, and the y-axis represents the total advertising dollars spent. The plot reveals a cyclical pattern, with peaks and valleys in advertising spending, possibly related to seasonal demand or marketing campaigns.

```
# Visualize the Home Depot Ad Spending data
ggplot(data = homedepot_adspend, aes(x = TIME_PERIOD, y = TOTAL_DOLLARS)) +
  geom_line() +
  labs(title = "Home Depot Ad Spending Data Visualization")
```



ADF test for Home Depot Ad Spending TOTAL_DOLLARS:

The plot and the ADF test signifies stationarity in the data. We go ahead and take log and differencing to remove/minimize stationarity.

```
# Performing adf test for checking the stationarity

adf_result <- adf.test(homedepot_adspend$TOTAL_DOLLARS)
```

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

```
# Print the ADF test result
print(adf_result)
```

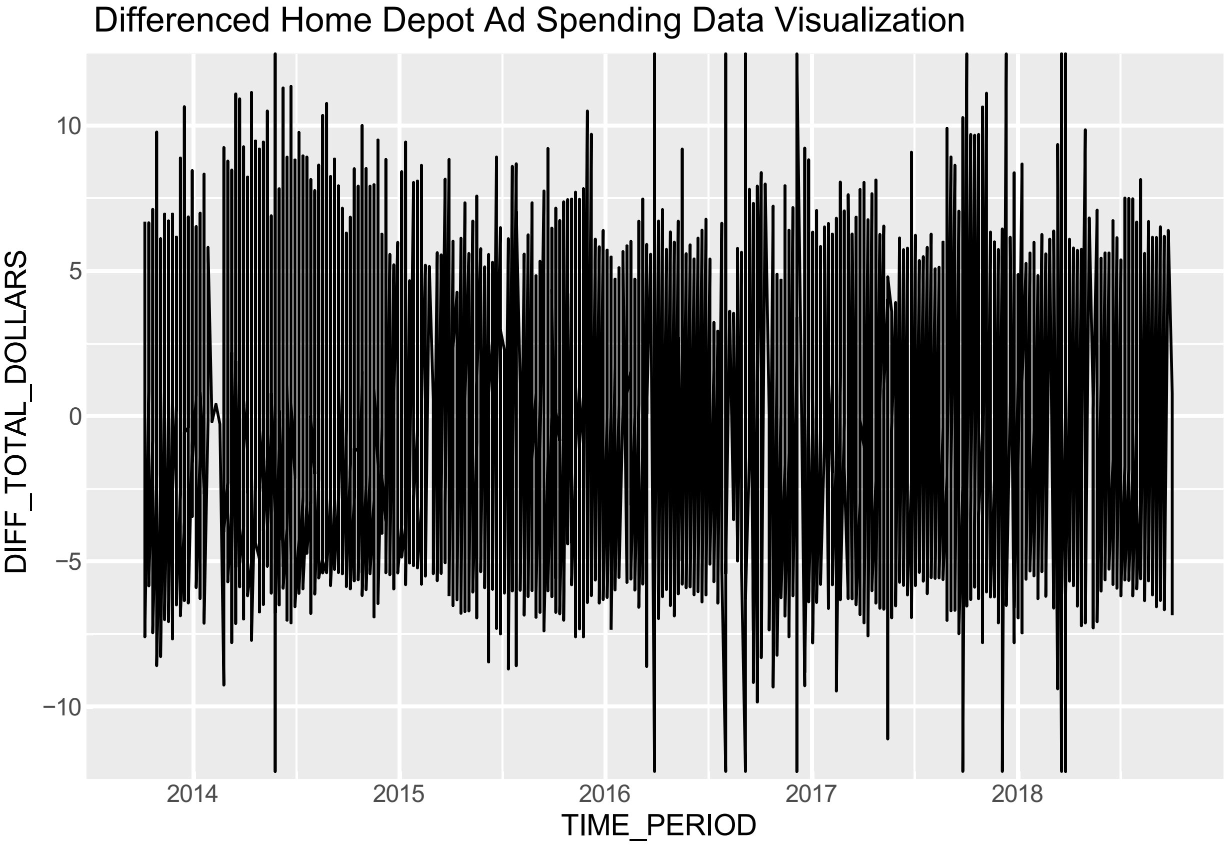
```
##
##  Augmented Dickey-Fuller Test
##
## data:  homedepot_adspend$TOTAL_DOLLARS
## Dickey-Fuller = -5.8114, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

First-order differencing for TOTAL_DOLLARS:

This plot shows the first-order differenced values of the Home Depot advertisement spending, which helps to remove the trend and make the time series stationary. The x-axis represents the time period, and the y-axis represents the differenced total advertising dollars. The plot shows a more stable pattern around a constant mean, indicating that the differencing has removed the trend and seasonality from the original data.

```
homedepot_adspend$DIFF_TOTAL_DOLLARS = c(NA, diff(log(homedepot_adspend$TOTAL_DOLLARS)))

# Visualize the Home Depot Ad Spending data
ggplot(data = homedepot_adspend, aes(x = TIME_PERIOD, y = DIFF_TOTAL_DOLLARS)) +
  geom_line() +
  labs(title = "Differenced Home Depot Ad Spending Data Visualization")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



ACF and PACF with DIFF_TOTAL_DOLLARS:

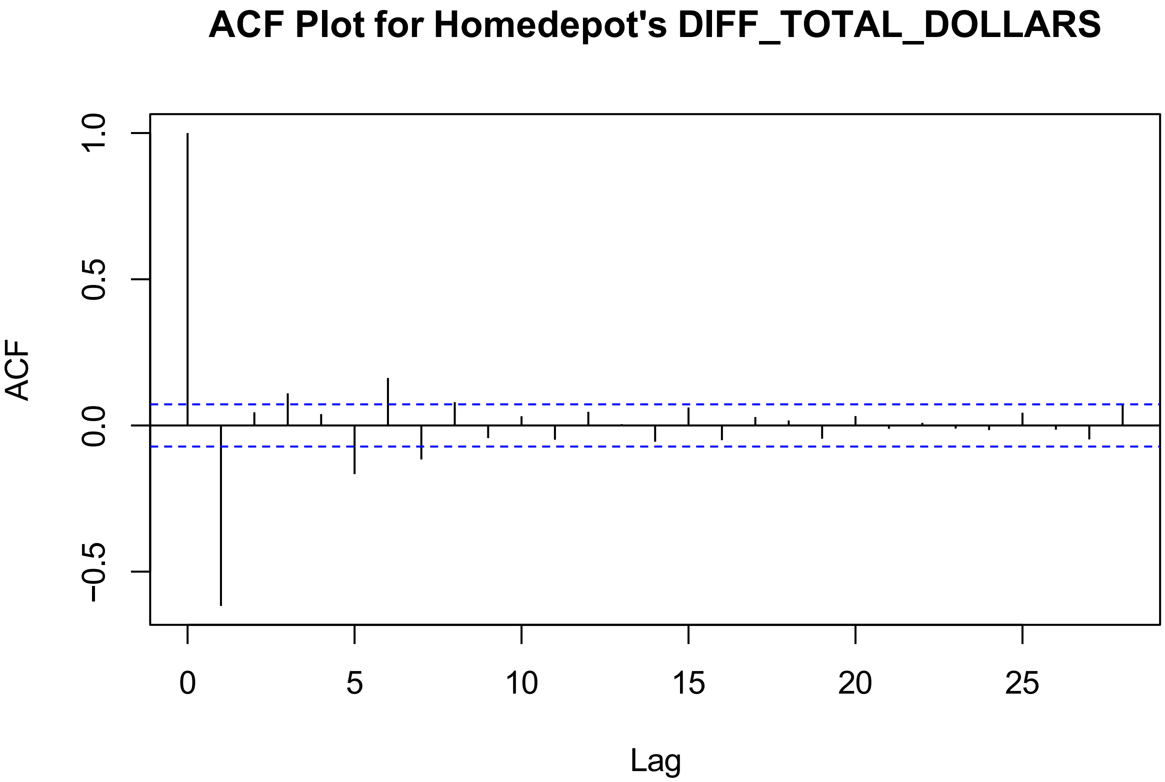
These plots show the ACF and PACF for the differenced Home Depot advertisement spending data. The ACF plot shows a significant spike at lag 1, suggesting the presence of a first-order moving average (MA) component in the time series. The PACF plot shows a decaying pattern, indicating the potential presence of an autoregressive (AR) component. These plots help identify the appropriate orders for the ARIMA/SARIMA models for the advertisement spending data.

```
sum(is.na(homedepot_adspend))
```

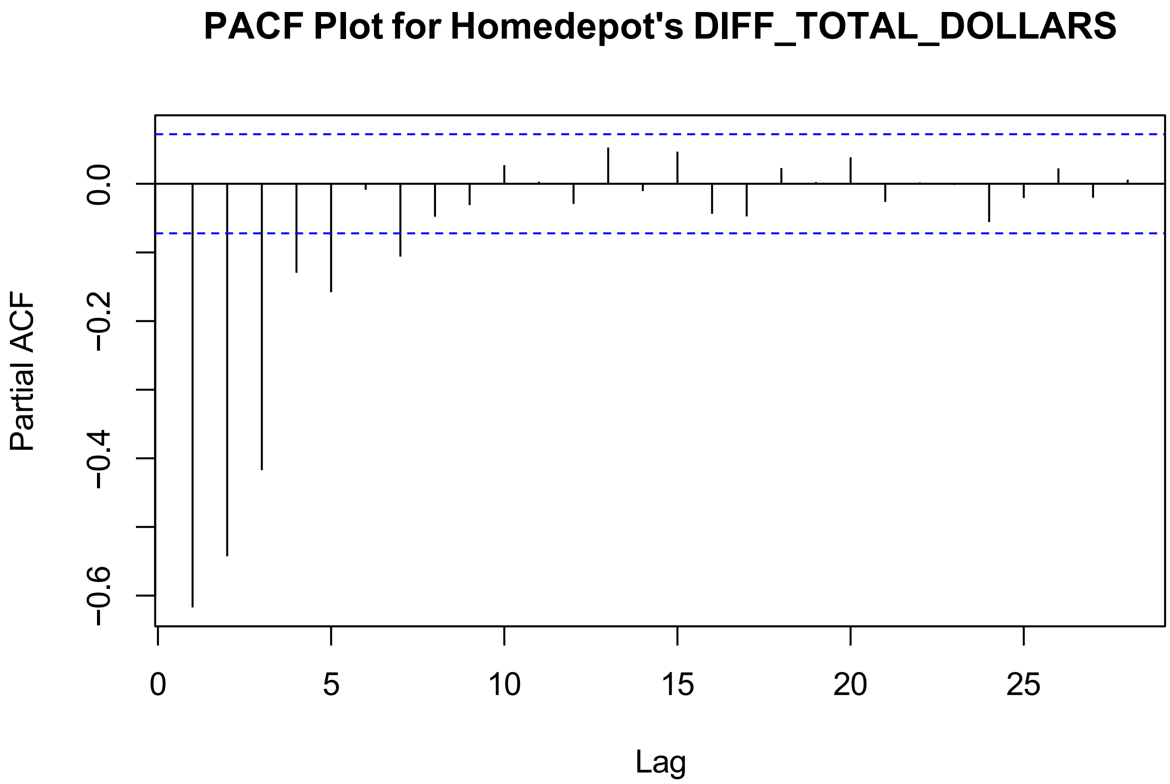
```
## [1] 1
```

```
homedepot_adspend = na.omit(homedepot_adspend)

# Filter rows with infinite values in DIFF_TOTAL_DOLLARS
inf_rows <- which(!is.finite(homedepot_adspend$DIFF_TOTAL_DOLLARS))
```

```
# Remove rows with infinite values
homedepot_adspend <- homedepot_adspend[-inf_rows, ]

# ACF and PACF plots for Google Trends
acf(homedepot_adspend$DIFF_TOTAL_DOLLARS, main = "ACF Plot for Homedepot's DIFF_TOTAL_DOLLARS")
```



```
pacf(homedepot_adspend$DIFF_TOTAL_DOLLARS, main = "PACF Plot for Homedepot's DIFF_TOTAL_DOLLARS")
```


PACF Plot for Homedepot's DIFF_TOTAL_DOLLARS

Relationship between Google Trends and advertisement spending:

Merge the 2 datasets into one, on columns “date” of Google Trends and “Time_Period” of Home Depot Ads Spending. Removed missing values.

Google Trends’s “date” column has daily data and Home Depot Ads Spending’s “TIME_PERIOD”. So when these 2 datasets are merged on date and TIME_PERIOD columns, we get the merged dataset with weekly data with 735 rows. Head() of this merged data is also printed for reference that shows date as weekly.

```
# Merge the datasets based on the date column
merged_data <- merge(google_trends, homedepot_adspend, by.x = "date", by.y = "TIME_PERIOD", all = TRUE)
sum(is.na(merged_data))
```

```
## [1] 20286
```

```
# Remove rows with missing values
merged_data = na.omit(merged_data)

print(nrow(merged_data))
```

```
## [1] 735
```

```
head(merged_data)
```

```
##            date id    value onediffvalue
## 1  2013-10-07  0 12997.16     222.3070
## 2  2013-10-07  0 12997.16     222.3070
## 9  2013-10-14  7 13957.23    1377.4804
## 10 2013-10-14  7 13957.23    1377.4804
## 11 2013-10-14  7 13957.23    1377.4804
## 18 2013-10-21 14 13282.89     733.4774
##                                            PRODUCT TOTAL_DOLLARS
## 1  Home Depot Home Center : Home Center/Hardware Store        5128.7
## 2                 Home Depot Home Center : Vignette           2.6
## 9                 Home Depot Home Center : Vignette          13.9
## 10 Home Depot Home Center : Home Center/Hardware Store        4760.4
## 11       Home Depot Home Center & Kidde : Combo Vignette        6.1
## 18                Home Depot Home Center : Vignette           4.4
##    NETWORK_TV_DOLLARS CABLE_TV_DOLLARS SYNDICATION_DOLLARS SPOT_TV_DOLLARS
## 1              1465.2           1211.8               171.9           348.3
## 2                 0.0              2.6                 0.0             0.0
## 9                 0.0             13.9                 0.0             0.0
## 10             1753.1           1105.8                91.1           116.9
## 11                0.0              0.0                 0.0             6.1
## 18                0.0              3.9                 0.0             0.5
##    MAGAZINES_DOLLARS NATL_NEWSP_DOLLARS NEWSPAPER_DOLLARS NETWORK_RADIO_DOLLARS
## 1                  0                  0             238.2                1635.2
## 2                  0                  0               0.0                   0.0
## 9                  0                  0               0.0                   0.0
## 10                 0                  0               0.0                1635.2
## 11                 0                  0               0.0                   0.0
## 18                 0                  0               0.0                   0.0
##    NAT_SPOT_RADIO_DOLLARS OUTDOOR_DOLLARS DIFF_TOTAL_DOLLARS
## 1                    58.3               0          6.7020579
## 2                     0.0               0         -7.5870960
## 9                     0.0               0         -5.8361981
## 10                   58.3               0          6.6597982
## 11                    0.0               0          0.8527773
## 18                    0.0               0         -7.4470756
```

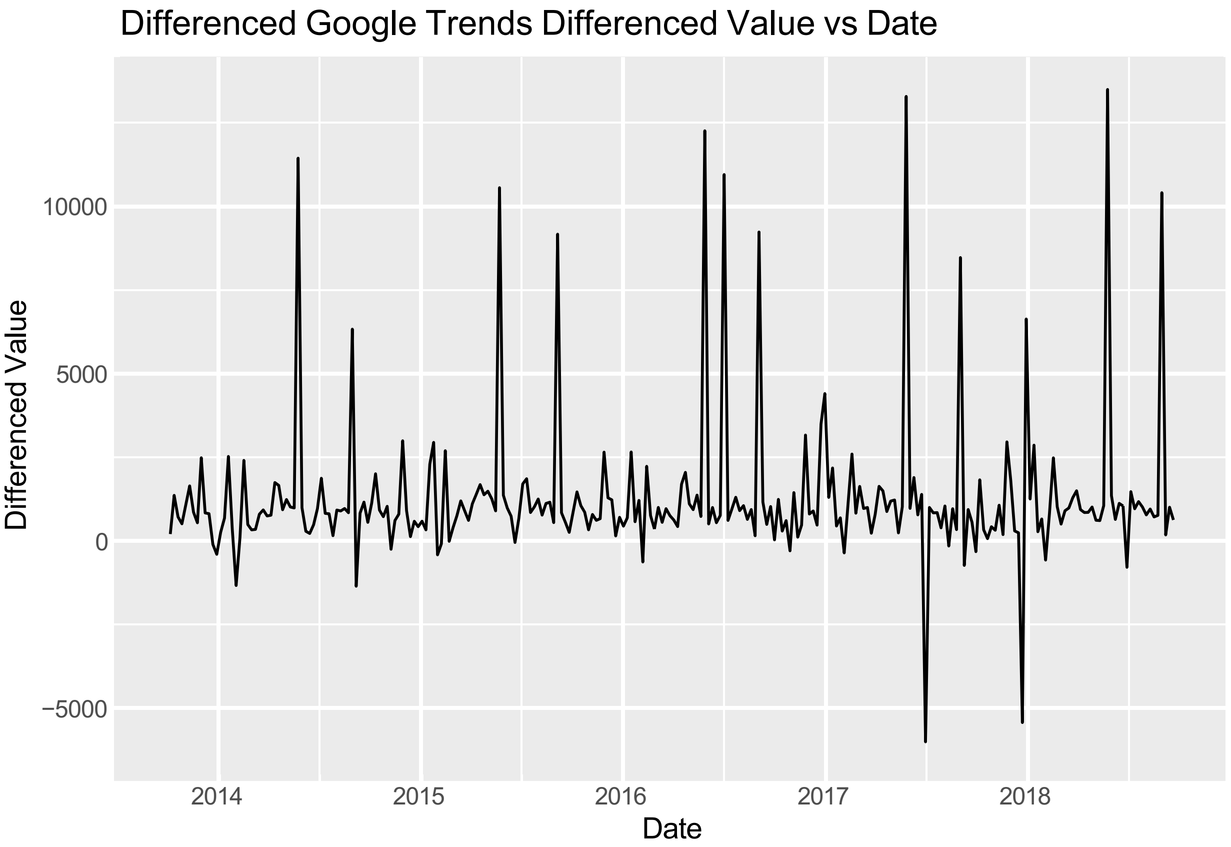
Visualize Merged Data:

1. Plot-1 (Differenced Google Trends Differenced Value vs. Date): This plot shows the differenced Google Trends data over time. The x-axis represents the date, and the y-axis represents the differenced value. This plot is similar to the "Google Trends: Date vs. One Diff Value" plot but is presented after merging the Google Trends and Home Depot advertisement spending datasets.

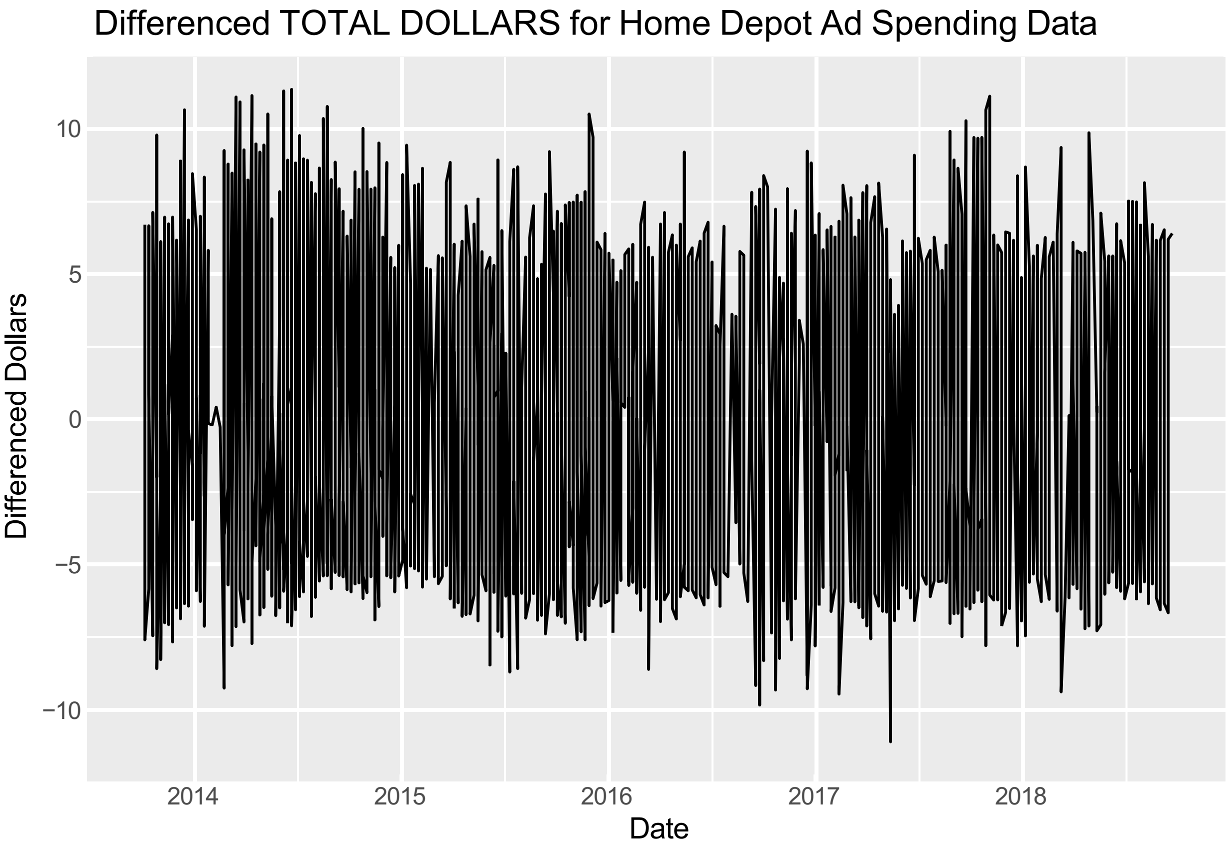
2. Plot-2 (Differenced TOTAL DOLLARS for Home Depot Ad Spending Data): This plot shows the differenced Home Depot advertisement spending over time. The x-axis represents the date, and the y-axis represents the differenced total advertising dollars. This plot is similar to the "Differenced Home Depot Ad Spending Data Visualization" but is presented after merging the Google Trends and Home Depot advertisement spending datasets.

3. Plot-3 (Google Trends Differenced Value vs. Total Advertisement Spending Differenced Total Dollars): This plot shows the relationship between the differenced Google Trends data (y-axis) and the differenced Home Depot advertisement spending (x-axis). It helps visualize the potential correlation or relationship between the two variables, which is useful for building regression models like SARIMAX.

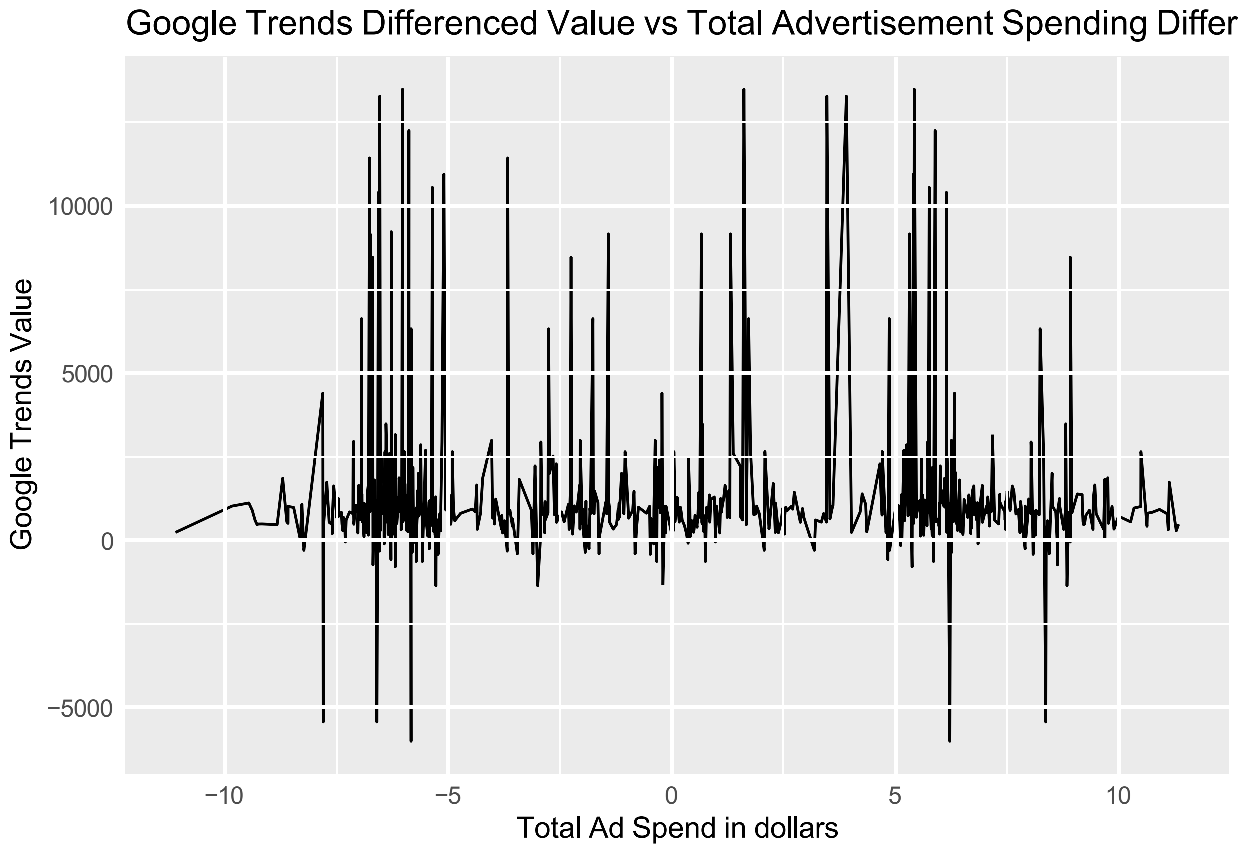
```r
# Plot the differenced Google Trends data
ggplot(merged_data, aes(x = date, y = onediffvalue)) +
  geom_line() +
  labs(title = "Differenced Google Trends Differenced Value vs Date", x = "Date",
                    y = "Differenced Value")
```



```
# Plotting the differenced Home Depot Ad Spending data
ggplot(merged_data, aes(x = date, y = DIFF_TOTAL_DOLLARS)) +
  geom_line() +
  labs(title = "Differenced TOTAL DOLLARS for Home Depot Ad Spending Data", x = "Date",
                                y = "Differenced Dollars")
```

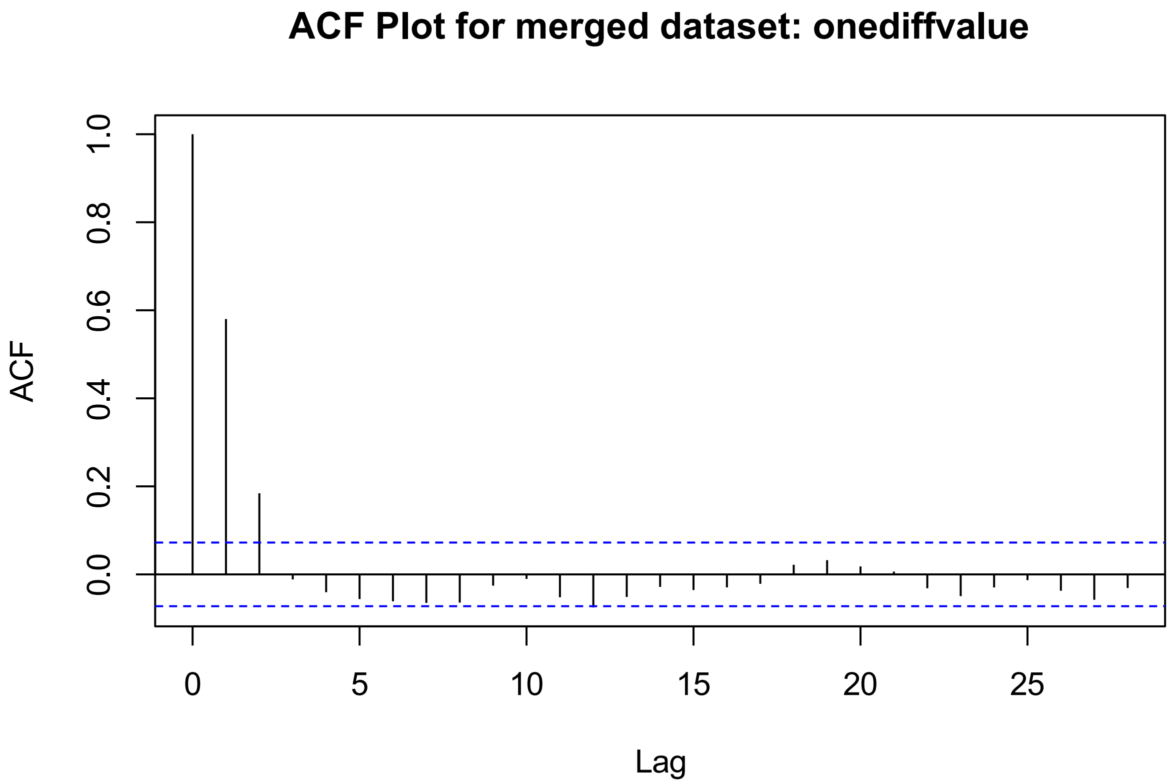


```
# Plot Google Trends data over time
ggplot(data = merged_data, aes(x = DIFF_TOTAL_DOLLARS, y = onediffvalue)) +
  geom_line() +
  labs(title = "Google Trends Differenced Value vs Total Advertisement Spending Differenced Total Dolla
                      x = "Total Ad Spend in dollars", y = "Google Trends Value")
```

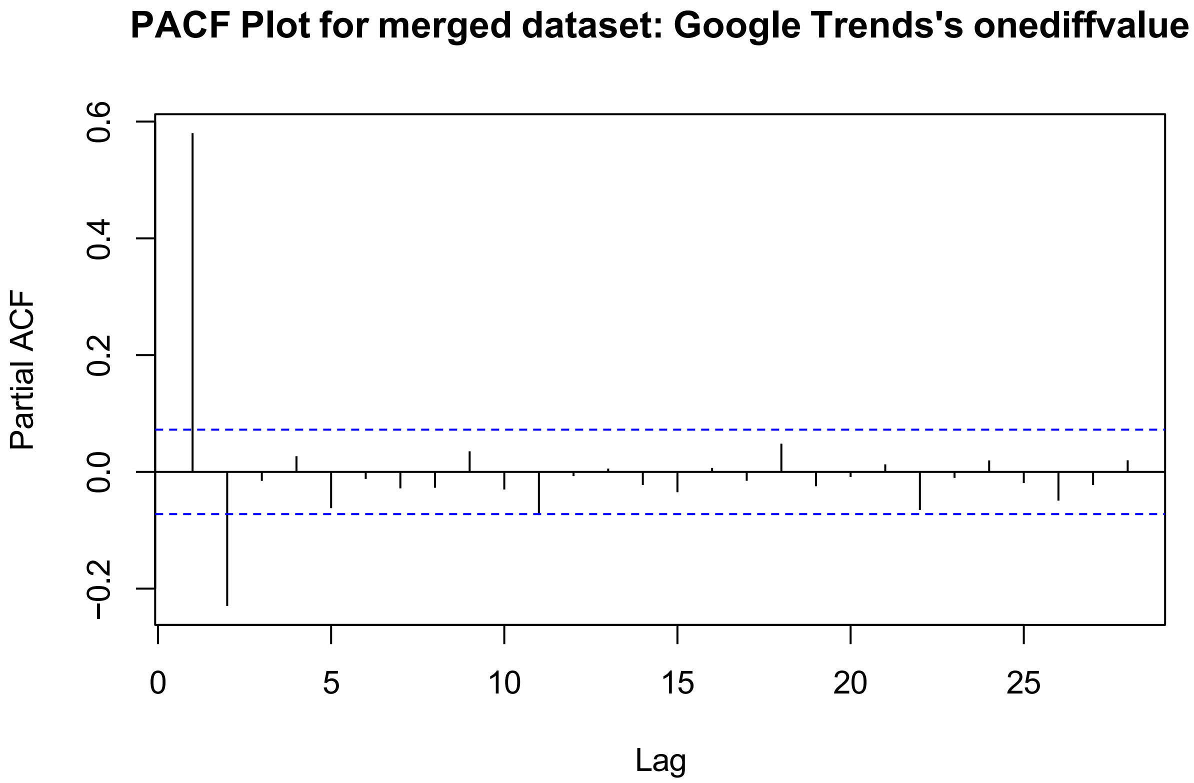


ACF and PACF plots for the merged dataset - Differenced values of Google Trends (onediffvalue) and Home Depot Ad Spending (DIFF_TOTAL_DOLLARS)

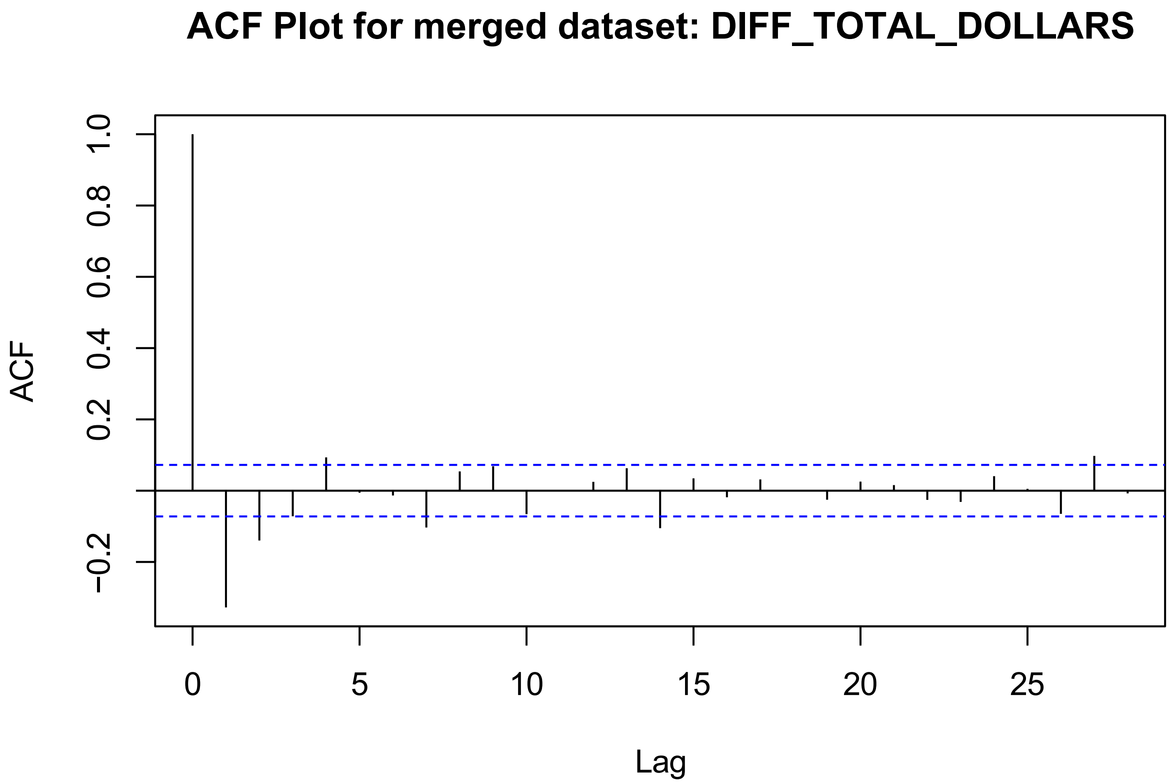
```
acf(merged_data$onediffvalue, main = "ACF Plot for merged dataset: onediffvalue")
```



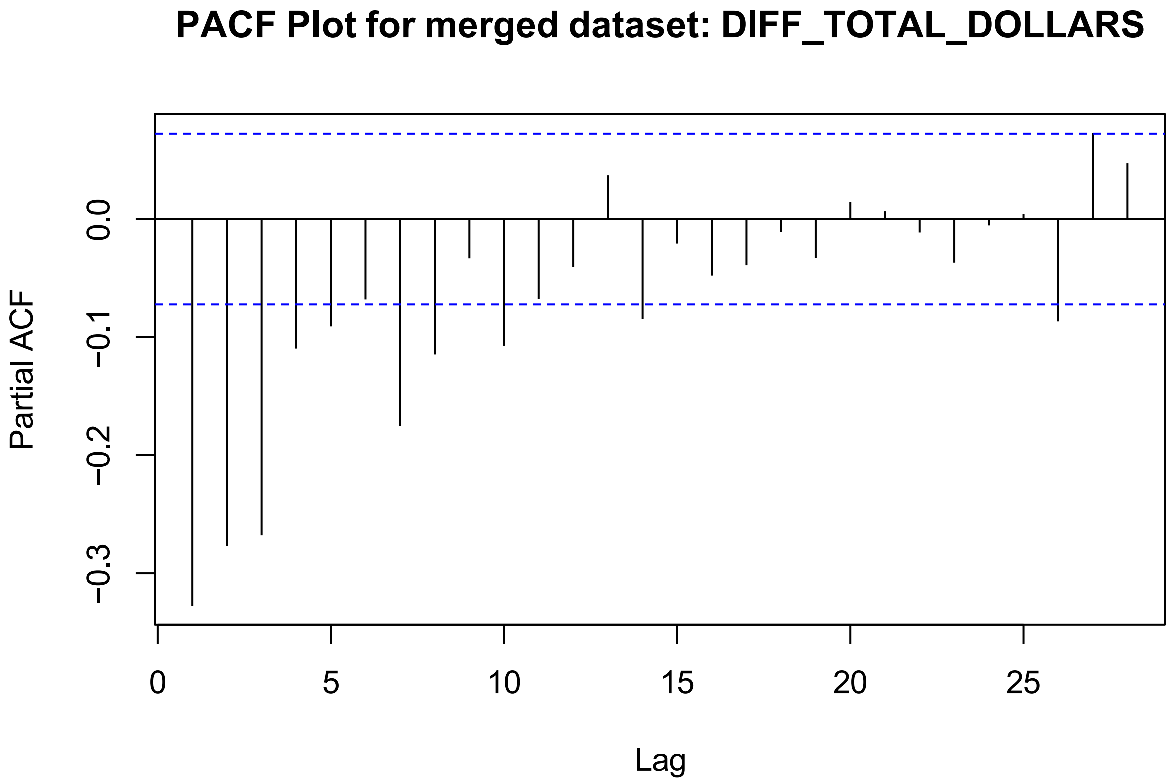
```
pacf(merged_data$onediffvalue, main = "PACF Plot for merged dataset: Google Trends's onediffvalue")
```



```
acf(merged_data$DIFF_TOTAL_DOLLARS, main = "ACF Plot for merged dataset: DIFF_TOTAL_DOLLARS")
```



```
pacf(merged_data$DIFF_TOTAL_DOLLARS, main = "PACF Plot for merged dataset: DIFF_TOTAL_DOLLARS")
```



Split the data into Training and Test (80% and 20%)

```
# Split the data into training and testing sets
train_size <- floor(0.8 * nrow(merged_data))
train_data <- merged_data[1:train_size, ]
test_data <- merged_data[(train_size + 1):nrow(merged_data), ]
```

Train models on train dataset:

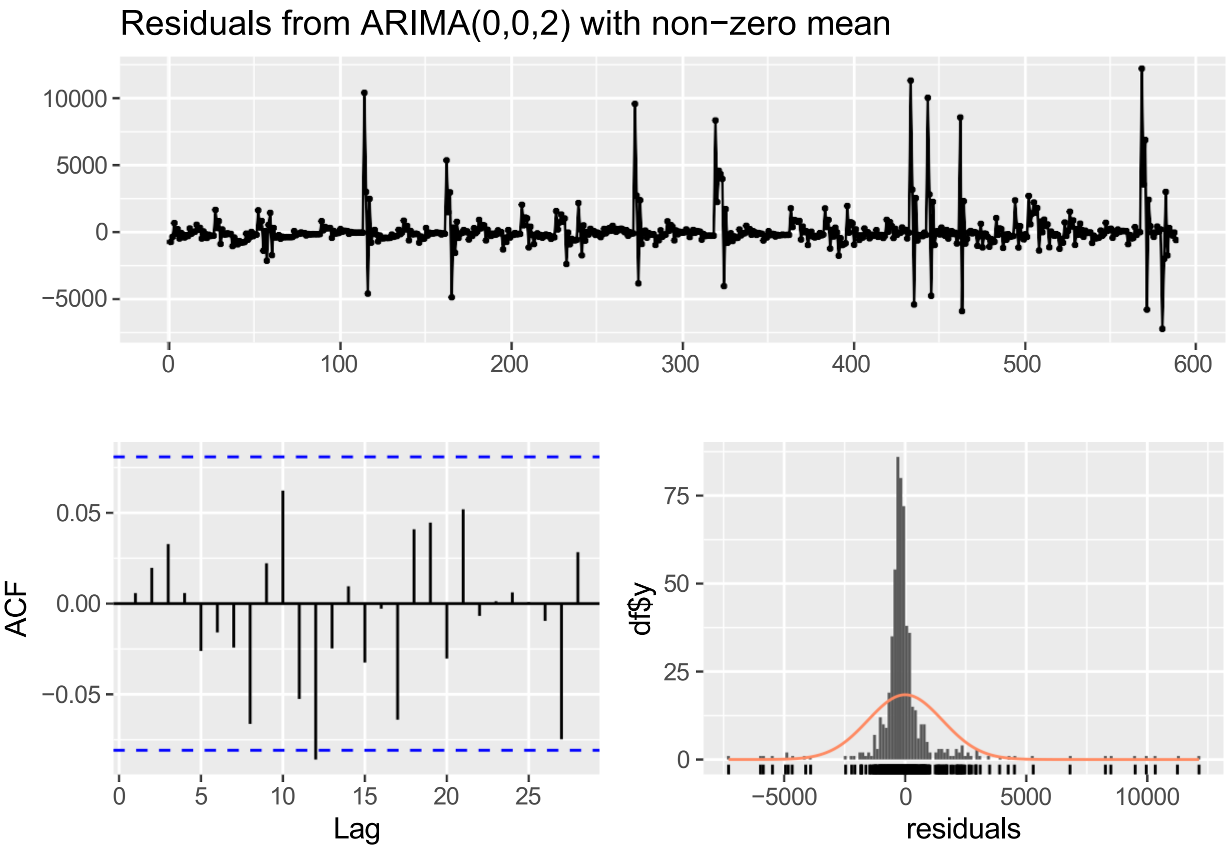
We chose to fit 2 models:

1. SARIMA without additional variables: We use auto.arima() to train the model. It correctly identifies the model to be a (0,0,2) model that matches well with the ACF and PACF plots mentioned above for the merged dataset’s onediffvalue.
2. SARIMAX with additional variables: We are using DIFF_TOTAL_DOLLARS of the merged dataset to be our external regressor and onediffvalue to be out target variable. The auto.arima() chooses the model to be a (0,0,2) model.

```
# Train SARIMA model without additional variables
sarima_model_wo_additional <- auto.arima(train_data$onediffvalue)
summary(sarima_model_wo_additional)
```

```
## Series: train_data$onediffvalue
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1     ma2       mean
##       0.7082  0.2296  1229.8776
## s.e.  0.0400  0.0392   122.7897
##
## sigma^2 = 2377761:  log likelihood = -5149.5
## AIC=10307   AICc=10307.07   BIC=10324.5
##
## Training set error measures:
##                     ME    RMSE      MAE       MPE     MAPE     MASE        ACF1
## Training set 0.3020978 1538.06 702.5965 -82.11875 176.9225 1.347009 0.005785303
```

```
checkresiduals(sarima_model_wo_additional)
```

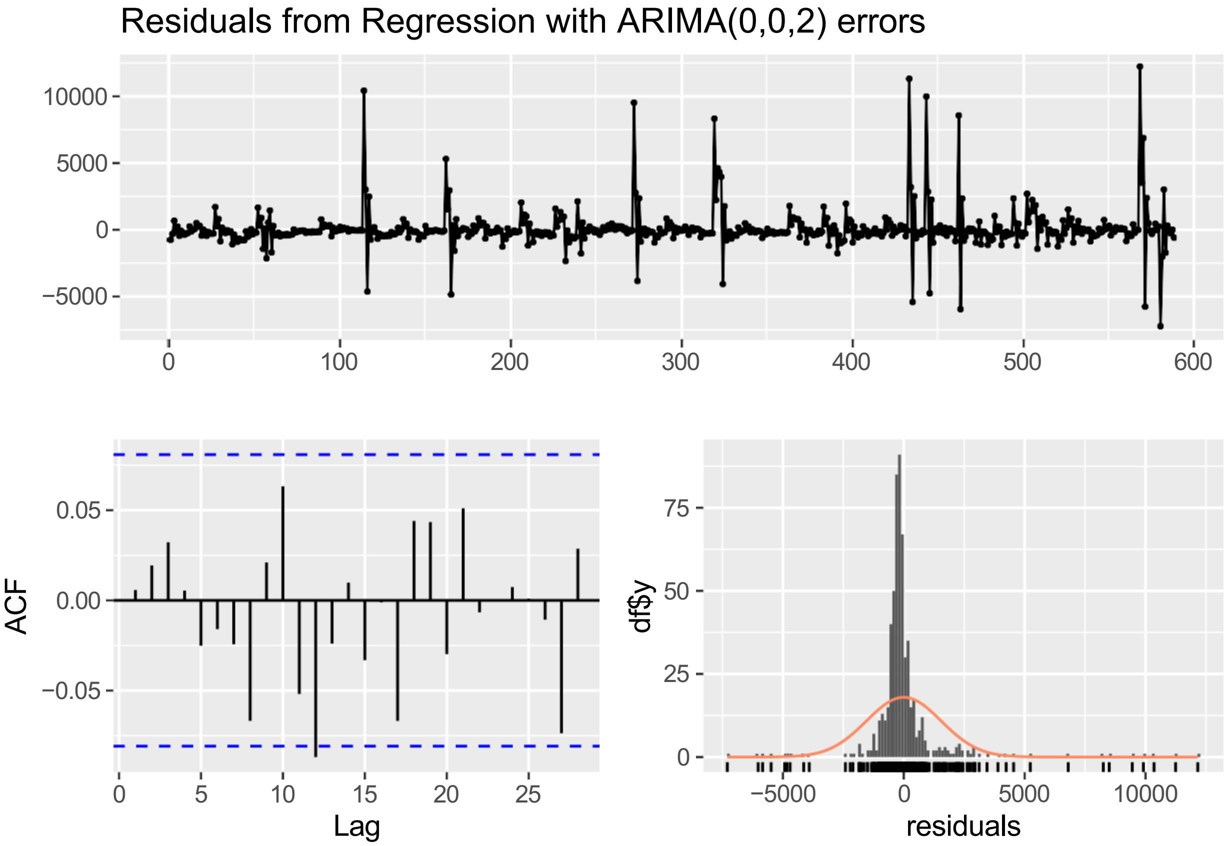


```
##
##   Ljung-Box test
##
## data:   Residuals from ARIMA(0,0,2) with non-zero mean
## Q* = 7.0604, df = 8, p-value = 0.5301
##
## Model df: 2.    Total lags used: 10
```

```
# Train SARIMAX model with additional variables
sarimax_model_additional <- auto.arima(train_data$onediffvalue,
                                       xreg = train_data$DIFF_TOTAL_DOLLARS, seasonal = TRUE)
summary(sarimax_model_additional)
```

```
## Series: train_data$onediffvalue
## Regression with ARIMA(0,0,2) errors
##
## Coefficients:
##          ma1     ma2  intercept    xreg
##       0.7091  0.2307  1229.7295  3.7555
## s.e.  0.0400  0.0393   122.8847  7.5876
##
## sigma^2 = 2380836:  log likelihood = -5149.38
## AIC=10308.75   AICc=10308.86   BIC=10330.64
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 0.4637945 1537.738 703.3406 -83.77338 178.2107 1.348436
##                     ACF1
## Training set 0.005749775
```

```
checkresiduals(sarimax_model_additional)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,0,2) errors
## Q* = 7.0919, df = 8, p-value = 0.5267
##
## Model df: 2.   Total lags used: 10
```

Evaluate models on test dataset

```
# Evaluate SARIMA model without additional variables
sarima_model_wo_additional_pred <- forecast(sarima_model_wo_additional, h = nrow(test_data))

# Evaluate SARIMAX model with additional variables
sarimax_model_additional_pred <- forecast(sarimax_model_additional, h = nrow(test_data),
                                          xreg = test_data$DIFF_TOTAL_DOLLARS)
```

MSPE for all models:

The MSPE values are calculated for the SARIMA and SARIMAX models on the test dataset, providing a measure of the models' predictive performance.

MSPE for SARIMA model without additional variable: 7558075.43500495

MSPE for SARIMAX model with additional variable: 7556795.88658626

A lower MSPE value indicates better predictive accuracy. There is not much of a difference in this case, but "SARIMAX model with additional variable" performs slightly better than "SARIMA model without additional variable".

```
# MSPE for SARIMA model without additional variables
sarima_model_wo_additional_mspe <- mean((sarima_model_wo_additional_pred$mean - test_data$onediffvalue)
print(paste("Mean Squared Prediction Error (MSPE) for SARIMA model without additional variable:",
                                          sarima_model_wo_additional_mspe))
```

```
## [1] "Mean Squared Prediction Error (MSPE) for SARIMA model without additional variable: 7558075.4350
```

```
# MSPE for SARIMAX model with additional variables
sarimax_model_additional_mspe <- mean((sarimax_model_additional_pred$mean - test_data$onediffvalue)^2)
print(paste("Mean Squared Prediction Error (MSPE) for SARIMAX model with additional variable:",
                                          sarimax_model_additional_mspe))
```

```
## [1] "Mean Squared Prediction Error (MSPE) for SARIMAX model with additional variable: 7556795.886586
```

Facebook Prophet on Google Trends:

Now, we implement the Facebook Prophet model, which is an additive regression model for time series forecasting. The Prophet model is fit to the original Google Trends data, incorporating weekly seasonality. The MSPE value for the Prophet model is calculated on the test dataset, allowing for a comparison with the SARIMA and SARIMAX models. MSPE value for the Prophet model is 6611852.07536799.

```
# Rename columns
prophet_data = merged_data[, c("date", "value")]
names(prophet_data) = c('ds', 'y')

# Make sure 'ds' is a date type
prophet_data$ds <- as.Date(prophet_data$ds)

# Fit Prophet model
prophet_model <- prophet(prophet_data, weekly.seasonality = TRUE, daily.seasonality = FALSE)

# Prepare future dataframe for predictions
future <- make_future_dataframe(prophet_model, periods = nrow(test_data))

# Forecast
prophet_forecast = predict(prophet_model, future)

# Extract the predicted values for the test set dates
forecasted_test_values <- prophet_forecast %>%
  filter(ds > max(train_data$ds)) %>%
  select(ds, yhat)
```

```
## Warning: There was 1 warning in `filter()`.
## i In argument: `ds > max(train_data$ds)`.
## Caused by warning in `max()`:
## ! no non-missing arguments to max; returning -Inf
```

```
# Merge the forecasted values with the actual values for the test set
test_data <- merge(test_data, forecasted_test_values, by.x = "date", by.y = "ds")

# Calculate MSPE on the test set
```

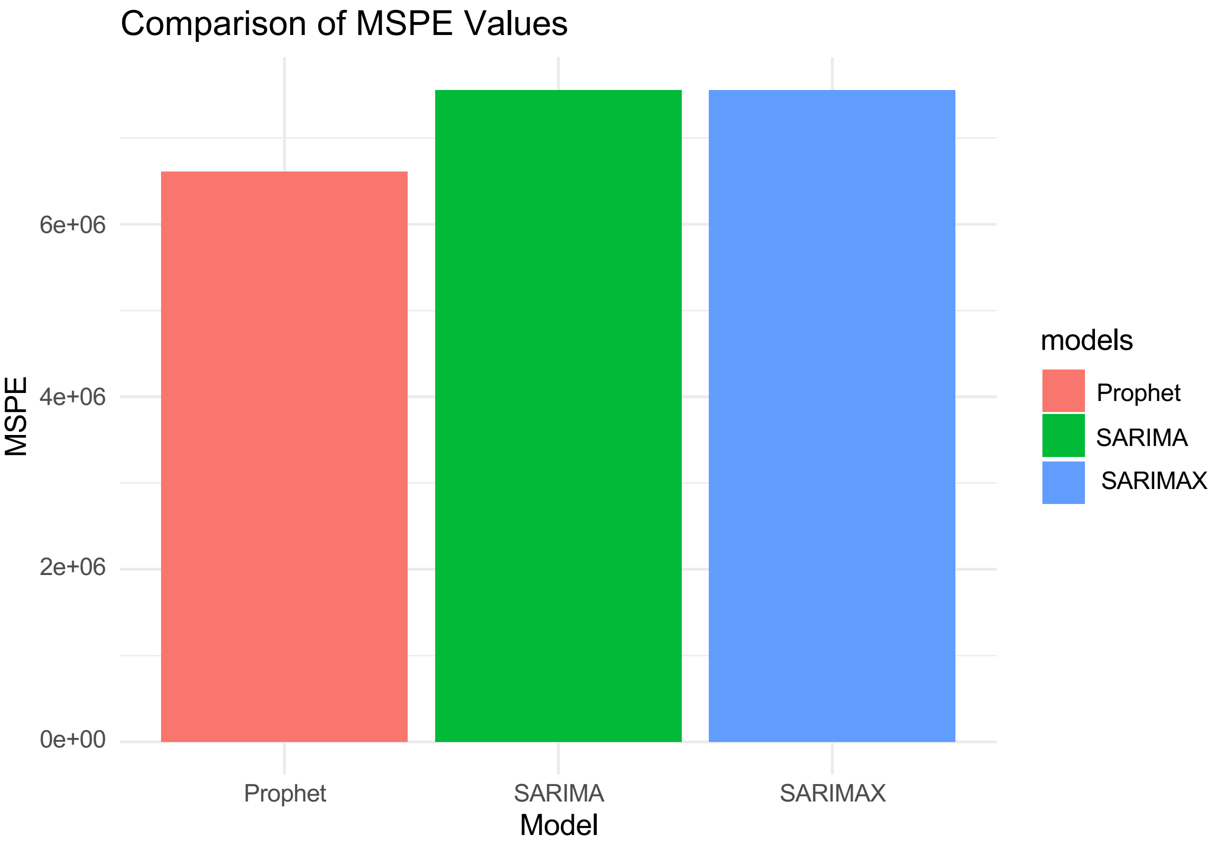
```
prophet_mspe <- mean((test_data$value - test_data$yhat)^2, na.rm = TRUE)
print(paste("Mean Squared Prediction Error (MSPE) for Prophet:", prophet_mspe))
```

```
## [1] "Mean Squared Prediction Error (MSPE) for Prophet: 6611852.07536799"
```

Compare MSPE's for all 3 models:

Prophet model better than SARIMA and SARIMAX since it has the lowest MSPE as compared to the other 2 models.

```
# Create a data frame with the model names and MSPE values
models <- c("SARIMA", "SARIMAX", "Prophet")
mspe <- c(sarima_model_wo_additional_mspe, sarimax_model_additional_mspe, prophet_mspe)
data <- data.frame(models, mspe)

# Create a bar plot
ggplot(data, aes(x = models, y = mspe, fill = models)) +
  geom_bar(stat = "identity") +
  labs(title = "Comparison of MSPE Values",
       x = "Model",
       y = "MSPE") +
  theme_minimal()
```


Comparison of MSPE Values
MSPE
6e+06
4e+06
2e+06
0e+00
Prophet
SARIMA
SARIMAX
Model
models
🟥 Prophet
🟩 SARIMA
🟦 SARIMAX