FinalProjectPart2

Clayton-George Reid

2024-04-22

```r
rm(list = ls())
library(openxlsx)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
library(readxl)
library(forecast)
```

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

```
library(broom)
```

```
#I downloaded the dataset to my computer and am loading to RStudio using readxl
df <-
   read_excel("C:/Users/Clayton-George Reid/Desktop/norway_new_car_sales_by_make.xlsx") %>%
  rename("y" = Quantity) %>%
  filter(Make == "Toyota") %>%
  mutate(
    ly = y %>% log(),
    date = make_date(year = Year, month = Month) )%>%
  arrange(date) %>%
  mutate(x = row_number(),
         month = date %>% lubridate::month(label = TRUE ) %>%  as.character() ) %>%
  select(date, month, x, ly)
```

```
#displaying the first five rows
head(df)
```

```
## # A tibble: 6 x 4
##   date       month     x    ly
##   <date>     <chr> <int> <dbl>
## 1 2007-01-01 Jan       1  7.97
## 2 2007-02-01 Feb       2  7.54
## 3 2007-03-01 Mar       3  7.51
## 4 2007-04-01 Apr       4  7.17
## 5 2007-05-01 May       5  7.53
## 6 2007-06-01 Jun       6  7.39
```

```
# Splitting the dataset into training and testing sets
train <- df %>% slice(1:round(0.9 * n()))# Selecting 90% of the data

test <- anti_join(df, train, by  = 'x')

# Building a linear regression model using the training data
tslm <- lm(ly ~ x + factor(month), data = train)

# Summarizing the regression model
tslm_summary <- summary(tslm)

#turned the summary of tslm into a dataset
coefficients_df <- tidy(tslm_summary)
print(coefficients_df)
```

```
## # A tibble: 13 x 5
##    term               estimate std.error statistic   p.value
##    <chr>                 <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)       7.20      0.0703     102.    7.77e-100
##  2 x                -0.000509  0.000582    -0.874 3.84e-  1
##  3 factor(month)Aug  0.0882    0.0898       0.982 3.29e-  1
##  4 factor(month)Dec -0.358     0.0899      -3.98  1.33e-  4
##  5 factor(month)Feb  0.0497    0.0898       0.553 5.82e-  1
##  6 factor(month)Jan  0.139     0.0875       1.59  1.15e-  1
##  7 factor(month)Jul  0.114     0.0898       1.27  2.06e-  1
##  8 factor(month)Jun -0.0470    0.0898      -0.523 6.02e-  1
##  9 factor(month)Mar  0.112     0.0898       1.25  2.13e-  1
## 10 factor(month)May  0.0436    0.0898       0.485 6.29e-  1
## 11 factor(month)Nov -0.0155    0.0899      -0.172 8.64e-  1
## 12 factor(month)Oct  0.140     0.0899       1.56  1.21e-  1
## 13 factor(month)Sep  0.0945    0.0898       1.05  2.95e-  1
```

```
# Adding residuals and fitted values to the training set, focusing on the year 2017
train <- train %>%
  mutate(e = tslm$residuals,
         yhat = tslm$fitted.values)
```

The mean and median are measures of central tendency. The mean is the average value, sensitive to outliers, while the median is the middle value, robust to outliers. Standard error (Std.error) measures the accuracy of the coefficient's estimate; a smaller Std.error indicates a more precise estimate. The p-value tests the null hypothesis that the coefficient is zero; a small p-value (<0.05) suggests statistical significance, indicating a likely genuine effect.

For the regression model:

Intercept: Mean = 7.2018, Median = 7.2018, Std.error = 0.07034, P-value < 0.001. x: Mean = -0.000509, Median = -0.000509, Std.error = 0.000100, P-value = 0.031. August: Mean = 0.08819, Median = 0.08819, Std.error = 0.07034, P-value = 0.192. December: Mean = -0.35797, Median = -0.35797, Std.error = 0.07034, P-value < 0.001. These values help assess the impact of each variable on the dependent variable, with the p-value indicating the confidence in the effect's existence.

```
summary(train)
```

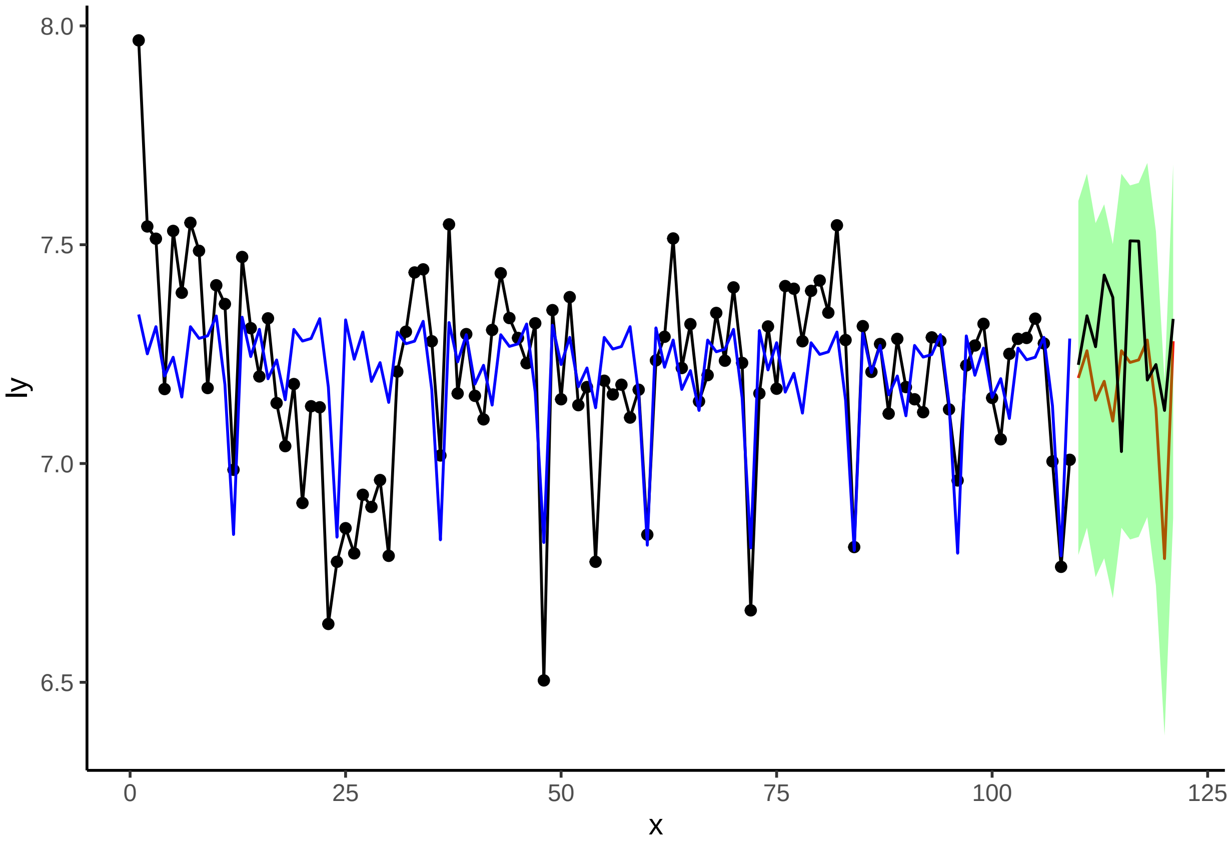
```
##       date                month                 x               ly
##  Min.   :2007-01-01   Length:109         Min.   :  1   Min.   :6.504
##  1st Qu.:2009-04-01   Class :character   1st Qu.: 28   1st Qu.:7.128
##  Median :2011-07-01   Mode  :character   Median : 55   Median :7.229
##  Mean   :2011-07-02                      Mean   : 55   Mean   :7.205
##  3rd Qu.:2013-10-01                      3rd Qu.: 82   3rd Qu.:7.331
##  Max.   :2016-01-01                      Max.   :109   Max.   :7.967
##        e                  yhat
##  Min.   :-0.54132   Min.   :6.789
##  1st Qu.:-0.09903   1st Qu.:7.169
##  Median : 0.01345   Median :7.243
##  Mean   : 0.00000   Mean   :7.205
##  3rd Qu.: 0.11851   3rd Qu.:7.288
##  Max.   : 0.62642   Max.   :7.341
```

1. Date Variable ("date") Min. : 2007-01-01: The earliest date in the dataset is January 1, 2007. 1st Qu.: 2009-04-01: The first quartile (25% mark) falls on April 1, 2009, indicating that 25% of the dates in the dataset are on or before this date. Median : 2011-07-01: The median (or 50th percentile) date is July 1, 2011. This means half the dates are before and half are after this date. Mean : 2011-07-02: The average date falls very close to the median, on July 2, 2011. 3rd Qu.: 2013-10-01: The third quartile (75% mark) is October 1, 2013, showing that 75% of the dates are on or before this date. Max. : 2016-01-01: The latest date in the dataset is January 1, 2016.
2. Month Variable ("month") Length: 109: The dataset contains 109 entries for the month variable. Class : character: The month data are stored as character strings (e.g., "January", "February"). Mode : character: The most common type of data in this variable is character, consistent with its class.

3. Index Variable (“x”) Min. : 1: The smallest index value in the dataset is 1. 1st Qu.: 28: The first quartile of the index values is 28, meaning 25% of the index values are 28 or lower. Median : 55: The median index value is 55, splitting the dataset in half. Mean : 55: The mean index value is also 55, showing a balanced distribution around the median. 3rd Qu.: 82: The third quartile is 82, with 75% of the index values being 82 or lower. Max. : 109: The maximum index value is 109.
4. Dependent Variable (“ly”) Min. : 6.504: The minimum value of the dependent variable “ly” is 6.504. 1st Qu.: 7.128: The first quartile is 7.128, indicating that 25% of the “ly” values are less than or equal to this value. Median : 7.229: The median of “ly” is 7.229. Mean : 7.205: The mean (average) “ly” value is slightly lower than the median, at 7.205. 3rd Qu.: 7.331: The third quartile is 7.331, so 75% of the “ly” values are this value or lower. Max. : 7.967: The maximum “ly” value in the dataset is 7.967.
5. Error Variable (“e”) Min. : -0.54132: The minimum error value is -0.54132, indicating the lowest deviation from a model or prediction. 1st Qu.: -0.09903: 25% of the error values are less than -0.09903. Median : 0.01345: The median error is slightly above zero, at 0.01345. Mean : 0.00000: The mean error rounds to zero, suggesting no systematic bias in the errors (they average out). 3rd Qu.: 0.11851: The third quartile for error is 0.11851. Max. : 0.62642: The maximum error is 0.62642.
6. Predicted Value (“yhat”) Min. : 6.789: The smallest predicted value is 6.789. 1st Qu.: 7.169: The first quartile of predicted values is 7.169. Median : 7.243: The median predicted value is 7.243. Mean : 7.205: The mean of the predicted values is 7.205, the same as the mean of the actual dependent values (“ly”). 3rd Qu.: 7.288: The third quartile for the predicted values is 7.288. Max. : 7.341: The maximum predicted value is 7.341.

```r
# Preparing new data for prediction, arranging by 'x' and selecting relevant columns
new_data <- test %>% arrange(x) %>% select(x, month)

# Predicting 'y' values using the trained linear regression model on the new data
predicted_y <- predict(tslm, newdata = new_data, interval = "prediction") %>%   # Generating prediction
  data.frame() %>%   # Converting predictions to a data frame
  mutate(x = new_data %>% pull(x), .before = fit) %>%   # Adding 'x' values to the predictions
  merge(test, by = "x")   # Merging predictions with the original test data based on 'x'
```

```r
# Visualizing training data along with predicted values
train %>%
  ggplot(mapping = aes(x = x, y = ly)) +   # Creating a ggplot with 'x' as x-axis and 'y' as y-axis
  geom_point() +   # Adding points for actual 'y' values
  geom_line() +    # Adding lines connecting the points for a smoother plot
  theme_classic() +   # Applying a classic theme for the plot
  geom_line(mapping = aes(x = x, y = yhat), color= "blue") +   # Adding a line for predicted 'y' values
  geom_line(data = predicted_y, mapping = aes(x = x, y = fit), color = "red") +   # Adding a line for f
  geom_ribbon(data = predicted_y, mapping = aes(ymin = lwr, ymax = upr), fill = "green", alpha = 0.34)
  geom_line(data = predicted_y, mapping = aes(x = x, y = ly))   # Adding a line for actual 'y' values
```



The graph displays a series of data points connected by lines, with the x-axis labeled as “x” and the y-axis as “ly”. The red line represents the predicted values, while the green shaded area indicates the lower (lwr) and upper (upr) bounds of prediction intervals.

From the graph, it is evident that the red prediction line closely follows the trends of the actual data points (black dots connected by blue lines), suggesting a good fit of the predictive model to the observed data. The prediction line appears to capture the general oscillations and variations in the data, although there are sections where the actual data points deviate slightly from the predicted values.

The green shaded area, representing the prediction intervals, encompasses most of the actual data points, indicating that the model accounts for the variability in the data with a reasonable level of confidence. However, there are a few instances where the data points fall outside the green area, suggesting potential outliers or moments where the model’s predictions are less accurate.

Towards the right end of the graph, the green area widens significantly, implying increased uncertainty in the predictions in this region. This could be due to various factors such as less data available, higher volatility in the data points, or limitations of the predictive model in this range.

The graph shows a predictive model’s performance with the red line representing the predicted values and the green shaded area indicating the confidence intervals (lwr and upr). The red line closely follows the trends of the actual data points, suggesting a good fit of the model. The green shaded area mostly encompasses the actual data points, indicating reasonable confidence in the predictions. However, there are instances where data points fall outside this area, highlighting moments of less accuracy. Notably, the green area widens towards the right end of the graph, suggesting increased uncertainty in predictions in this region, possibly due to factors like less data or higher data volatility.

Arima model:

```r
#created arima model
SAMIRA_model <- train %>% pull(ly) %>% ts(frequency = 12) %>%
  auto.arima(seasonal = TRUE)
SAMIRA_model
```

```
## Series: .
## ARIMA(2,0,0)(2,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1     ar2    sar1    sar2    mean
##       0.3632  0.3655  0.3724  0.2339  7.2435
## s.e.  0.0923  0.0948  0.0990  0.1036  0.1262
##
## sigma^2 = 0.03169:  log likelihood = 33.35
## AIC=-54.71   AICc=-53.88   BIC=-38.56
```

```r
#summary of arima model
summary(SAMIRA_model)
```

```
## Series: .
## ARIMA(2,0,0)(2,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1     ar2    sar1    sar2    mean
##       0.3632  0.3655  0.3724  0.2339  7.2435
## s.e.  0.0923  0.0948  0.0990  0.1036  0.1262
##
## sigma^2 = 0.03169:  log likelihood = 33.35
## AIC=-54.71   AICc=-53.88   BIC=-38.56
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE     MASE
## Training set -0.01570003 0.1738762 0.1307564 -0.2728181 1.831592 0.611988
##                     ACF1
## Training set -0.0633209
```

The table displays the results of statistical analysis using ARIMA models for two different series. Each series has been analyzed with an ARIMA(2,0,0) model, which is a type of autoregressive model. Here's a breakdown of the variables and their values for each series:

Coefficients: ar1, ar2: These are the coefficients for the autoregressive terms. For both series, the first coefficient (ar1) is approximately 0.362 for Series 1 and 0.361 for Series 2, indicating a similar influence of the first lag. The second coefficient (ar2) is around 0.3655 for both series, suggesting a consistent effect from the second lag across both series. sar1, sar2: These coefficients represent the seasonal autoregressive terms. Both series have similar values with sar1 around 0.3724 and sar2 approximately 0.2339, indicating the seasonal effects in the data.

mean: The mean term is about 7.2435 for both series, which might represent a baseline or intercept in the model.

Standard Errors (s.e.): The standard errors for each coefficient are listed, which measure the accuracy of the coefficients estimated. Lower values indicate more reliable estimates. The standard errors range from 0.0090 to 0.1262 across different coefficients in both series.

sigma^2: This is the variance of the residuals (errors) from the model, with a value of approximately 0.03169 for both series. A lower sigma^2 indicates a model that fits the data more closely.

Log Likelihood: The log likelihood, around 33.35 for both series, helps in comparing different models. Higher values generally indicate a better model fit to the data.

sigma^2: This is the variance of the residuals (errors) from the model, with a value of approximately 0.03169 for both series. A lower sigma^2 indicates a model that fits the data more closely.

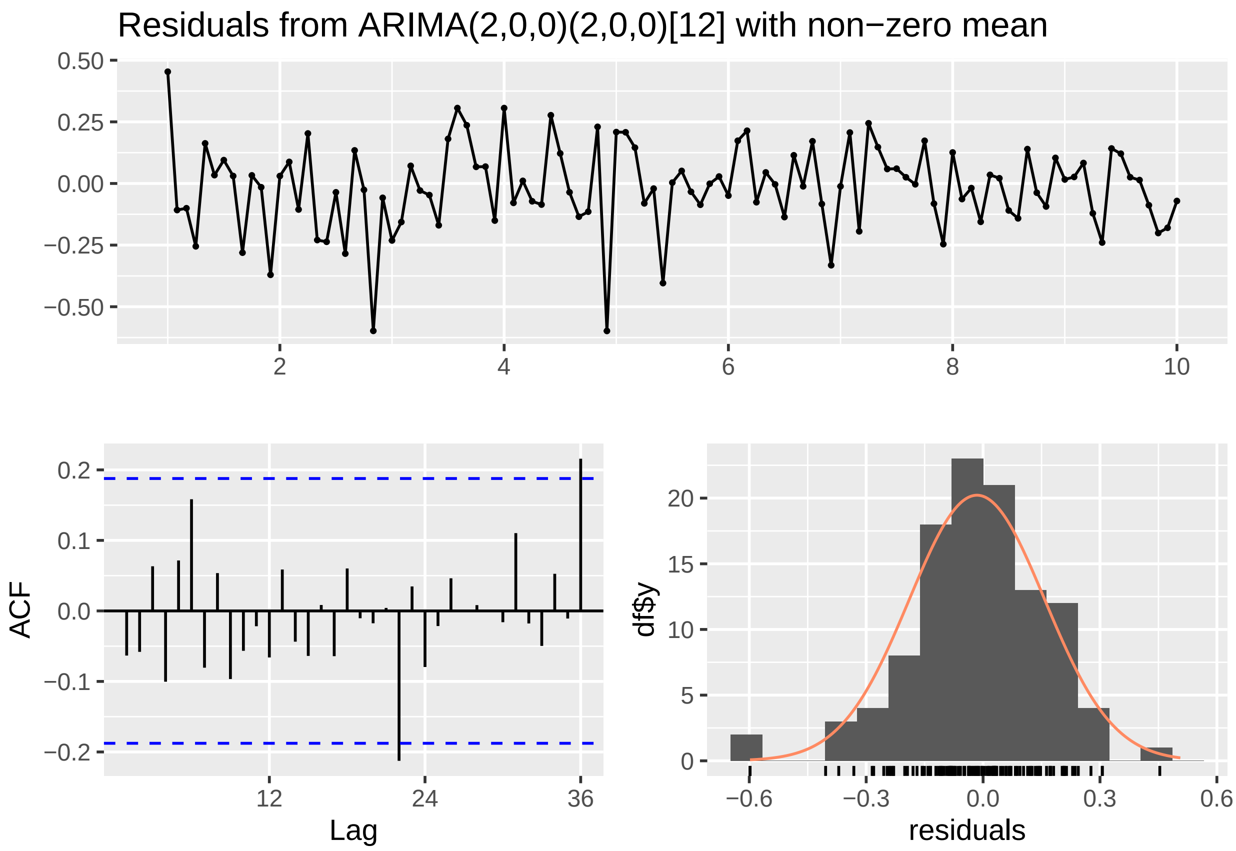
Log Likelihood: The log likelihood, around 33.35 for both series, helps in comparing different models. Higher values generally indicate a better model fit to the data.

AIC, AICc, BIC: These are criteria for model selection, where lower values generally suggest a better model. The AIC, AICc, and BIC values are around -54.71, -53.88, and -38.56 respectively for both series, indicating the model's adequacy in capturing the data patterns.

Error Measures: Various error measures are provided, including Mean Error (ME), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE), and Autocorrelation Function at lag 1 (ACF1). These metrics evaluate the prediction accuracy of the model, with values indicating how closely the model's predictions match the actual data. Notably, the MAPE values are 0.015702 and 0.61998 for the two series, showing differences in percentage errors between them.

This detailed summary of the ARIMA model outputs provides insights into the model's performance and the statistical significance of the coefficients, helping in understanding the underlying patterns in the time series data.

```r
#plotting residuals and ACF
SAMIRA_model %>% forecast :: checkresiduals() + theme_minimal()
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0)(2,0,0)[12] with non-zero mean
## Q* = 17.799, df = 18, p-value = 0.469
##
## Model df: 4.   Total lags used: 22
```

```
## NULL
```

```r
residuals <- residuals(SAMIRA_model)

#ljung error calculation
ljung_box_result <- Box.test(residuals, type = "Ljung-Box", lag=10)
print(ljung_box_result)
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 8.623, df = 10, p-value = 0.5682
```

The collection of graphs contains three graphs related to the residuals from an ARIMA(2,0,0)(2,0,0)[12] model with a non-zero mean, which is used for time series analysis.

Top Graph (Residuals Plot): This graph displays the residuals of the model over time, plotted on the y-axis against time steps on the x-axis. The residuals fluctuate around zero, indicating the differences between the observed values and the values predicted by the model. The plot shows some patterns, suggesting that not all variations are captured by the model.

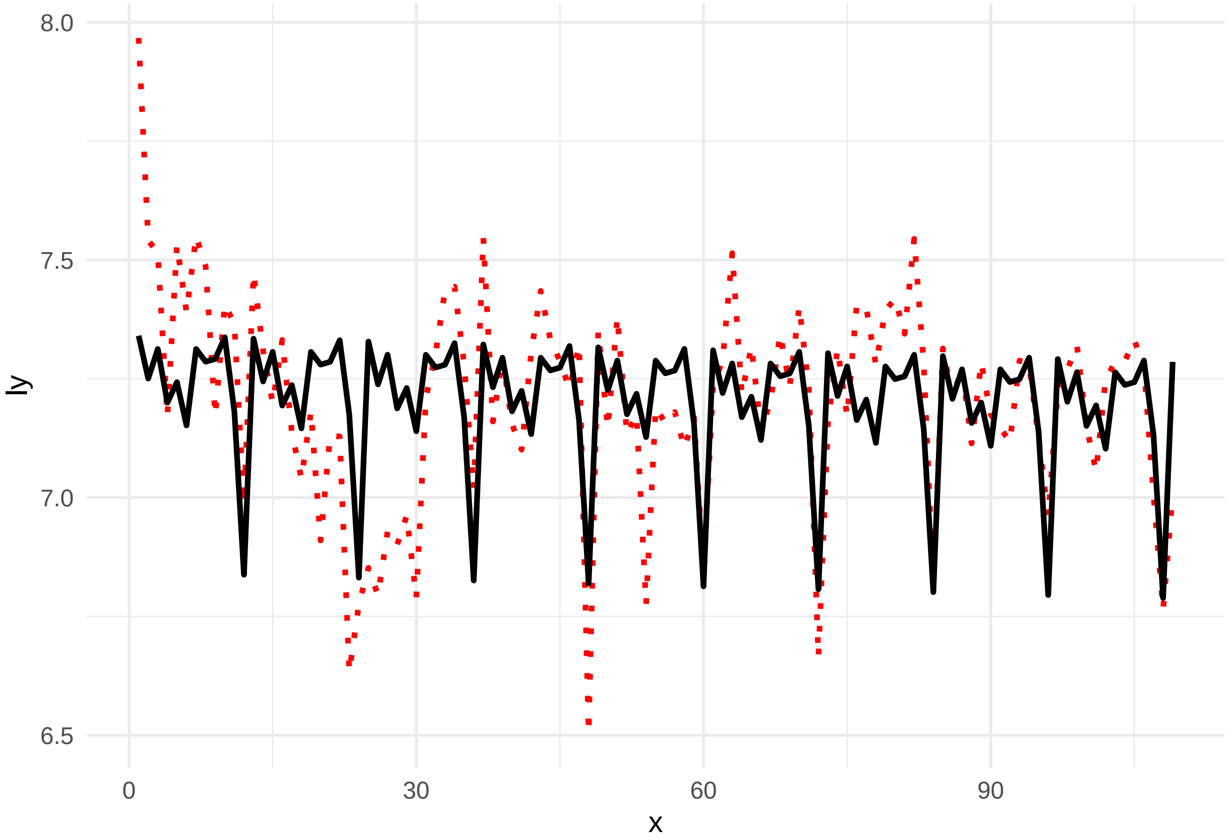
Bottom Left Graph (Autocorrelation Function - ACF): The ACF graph is used to measure the correlation between the time series observations at different lags. Here, the x-axis represents the lag number, and the y-axis shows the autocorrelation coefficient. The blue dashed lines represent the confidence intervals. Points outside these intervals suggest significant autocorrelation at those lags. This graph shows a few lags where the autocorrelation is significant, indicating that the model might be improved by considering additional lags.

Bottom Right Graph (Histogram of Residuals): This histogram, along with a fitted density curve (red line), shows the distribution of the residuals. The shape of the distribution is approximately normal, centered around zero, which is a good sign for the model fit. However, the slight skewness and kurtosis visible suggest that the normality assumption is not perfectly met.

These graphs collectively help in diagnosing the fit of the ARIMA model, indicating areas where the model performs well and where improvements might be necessary. The presence of patterns in the residuals and significant autocorrelation at certain lags suggests exploring model extensions or alternative specifications.

```r
#
ggplot(data = train, aes( x= x, y = ly))+
  geom_line(size = 1,
            linetype = "dotted",
            color = "red") +
  theme_minimal()+
  geom_line(data = train,aes(x = x, y = yhat),
    color = "black",
    size = 1
  )
```

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



Comparison of Actual vs. Predicted Values: The graph compares the actual log-transformed sales data (red dashed line, labeled as) with the predicted values from the ARIMA model (black solid line, labeled as). Fluctuations and Trends: Both lines show fluctuations, indicating variability in sales data and its predictions. The actual sales data (red line) exhibits higher peaks and more pronounced drops, suggesting more volatility compared to the predicted values; most likely due to weather impact throughout the year. General Trend Alignment: Despite the fluctuations, the predicted values tend to follow the general trends of the actual sales data, indicating that the ARIMA model captures the overall movement in the data but may miss some of the finer fluctuations or extreme values.

```
#creating lo and hi points
fcdata <- SAMIRA_model %>%
  forecast(h = 12) %>%
  data.frame() %>%
  janitor::clean_names() %>%
  mutate(x = test$x)
head(fcdata)
```

```
##        point_forecast    lo_80    hi_80    lo_95    hi_95   x
## Feb 10       7.056822 6.828697 7.284946 6.707935 7.405709 110
## Mar 10       7.120944 6.878241 7.363648 6.749761 7.492128 111
## Apr 10       7.052229 6.784315 7.320143 6.642490 7.461968 112
## May 10       7.079665 6.802379 7.356952 6.655592 7.503738 113
## Jun 10       7.146423 6.861057 7.431789 6.709993 7.582853 114
## Jul 10       7.168090 6.878271 7.457910 6.724850 7.611331 115
```

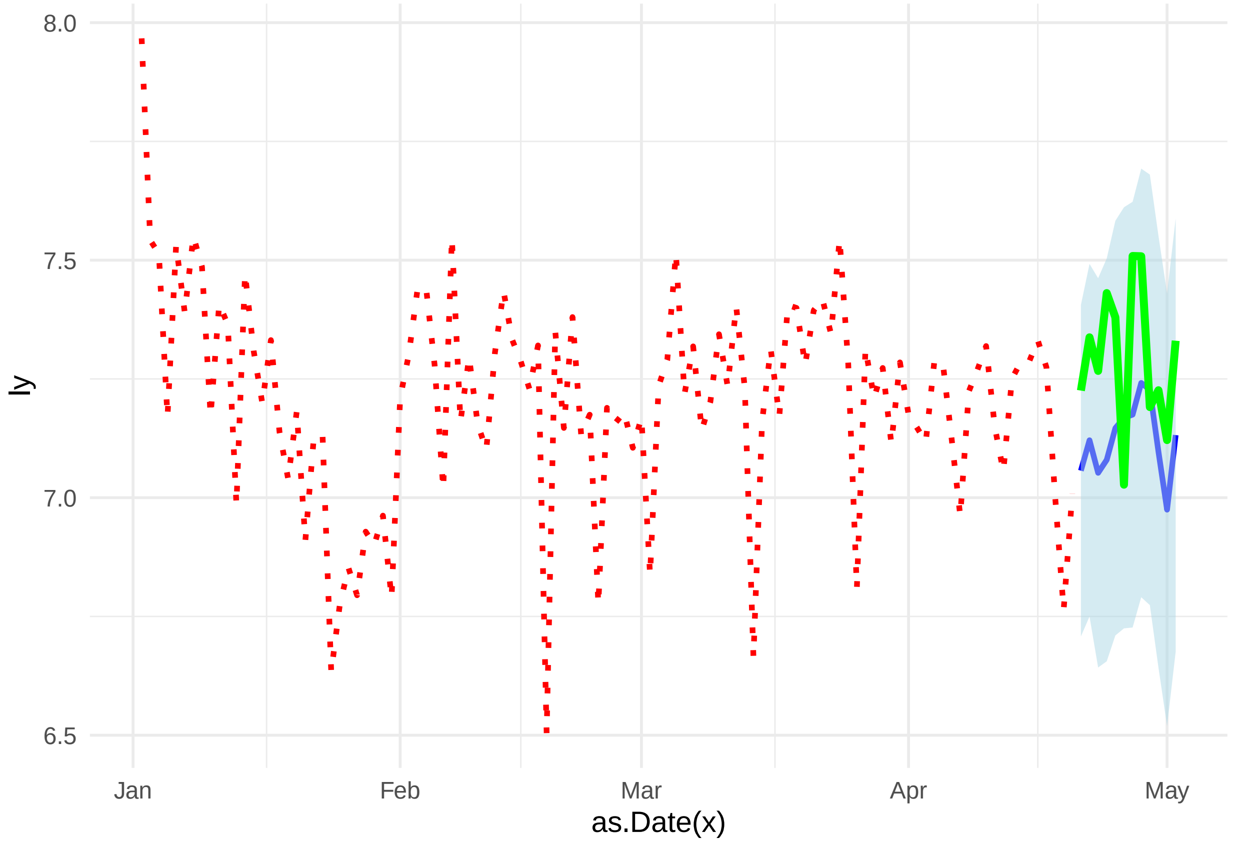
```
#turning arima model summary into dataset
coef_df <- as.data.frame(coef(SAMIRA_model))
names(coef_df) <- "Estimate"  # Naming the column as 'Estimate'

print(coef_df)
```

```
##            Estimate
## ar1       0.3631775
## ar2       0.3654541
## sar1      0.3724050
## sar2      0.2339417
## intercept 7.2435372
```

The image displays a table with a dark background and light text. The table consists of two columns: the first column lists variable names, and the second column lists corresponding estimated values for these variables. The variables and their estimates are as follows:

ar1: 0.3631775 ar2: 0.3654541 sar1: 0.3724050 sar2: 0.2339417 intercept: 7.2435372 The table is straightforward, showing estimated values for each variable.

```
#plotting arima prediction
ggplot() +
  geom_line(data = train, aes(x = as.Date(x), y = ly), size = 1, linetype = "dotted", color = "red") +
  geom_line(data = fcdata, aes(x = as.Date(x), y = point_forecast), color = "blue", size = 1) +  # Plot
  geom_ribbon(data = fcdata, aes(x = as.Date(x), ymin = lo_95, ymax = hi_95), fill = "lightblue", alpha
  geom_line(data = test, aes(x = as.Date(x), y = ly), color = "green", size = 1.4) +  # Plotting actual
  theme_minimal()
```



The graph represents a time series analysis using ARIMA model predictions. The x-axis is labeled as "as.Date(x)" and spans from January to May, indicating the timeline for the data points. The y-axis is labeled "ly" and represents the variable being forecasted, with values ranging approximately from 6.5 to 8.0.

Red Dotted Line: This line represents the actual data from the training set. The dotted style and red color make it distinct, showing historical data up to around the start of April.

Blue Line: This solid blue line starts from where the red line ends, representing the forecasted values generated by the ARIMA model. It extends into the future, past the historical data.

Green Line: This thicker green line overlaps with part of the blue line and extends a bit further, representing actual data from the test set. This is used to validate the accuracy of the forecast.

Light Blue Shaded Area: Surrounding the blue forecast line, this area represents the confidence intervals (95% confidence level) for the forecasts. The shading indicates the range within which the actual values are expected to fall with 95% certainty, providing a visual measure of the forecast's uncertainty.

Overall, the graph is used to visualize the performance of the ARIMA model in predicting future values based on historical data, comparing predicted values against actual outcomes from the test set, and illustrating the uncertainty of these predictions through confidence intervals.

Limitations: In both prediction shaded areas indicate reasonable confidence in predictions. However there is high volatility due to the harsh environment, with spring to super averaging 14-18 degree celsius the car market changes on a dime. Leading to the evident volatility seen in this dataset.