Icon

Description automatically generated

**Conestoga College, Doon Campus**

**Predictive Analytics (1498)**

**Statistical Applications for Data Analytics II**

Individual Project: Statistical Forecasting Project 1

**Date: 06/19/2024**

**Guided by:**

Prof. Jonathan Plumbtree

**Table of Contents**

[**ABOUT DATA AND THE DATASET** 3](#_Toc170002651)

[**VISUALIZATION METHODS FOR TIME SERIES ANALYSIS** 5](#_Toc170002652)

[**TRANSFORMATIONS** 8](#_Toc170002653)

[**FORECASTING AND ANALYSIS** 13](#_Toc170002654)

# **ABOUT DATA AND THE DATASET**

* **Raw Data Source**:

<https://in.investing.com/commodities/gold-mini>

* **Dataset link** (CSV file will be attached with the project report and code in the project submission folder):

<https://www.kaggle.com/datasets/nisargchodavadiya/daily-gold-price-20152021-time-series>

* **Details of date dimension and type of data**

1. **The Project is based on analyzing and forecasting time series data of daily gold prices in India.**
2. The **date dimension** itself reflects daily gold prices each day starting from Jan 1st, 2014, to August 5th, 2022.
3. Hence the time dimension is “daily” and below are the key dimension values in the dataset that we wish to forecast for gold prices in CAD:

|  |  |  |
| --- | --- | --- |
| **Dimension Name** | **Dimension type** | **Description** |
| Date | Date dimension  **Type**: Date format  **Unit**: Daily | This mentions the date on which the data for that specific day was recorded for Open, High Low, Volume and Chg% values |
| Price | **Integer**  **Unit**: “INR” (Indian Rupee) | This is the actual price of gold shown daily from 2014-2022 as mentioned above.  Since we wish to forecast in CAD, hence the prices are converted to CAD by dividing by a factor of 62 and that is done in the R code file. |
| Open | **Integer**  **Unit**: “INR” (Indian Rupee) | This is the opening price of gold shown daily from 2014-2022 as mentioned above.  Since we wish to forecast in CAD, hence the prices are converted to CAD by dividing by a factor of 62 and that is done in the R code file. |
| High | **Integer**  **Unit**: “INR” (Indian Rupee) | This is the highest price of gold shown daily from 2014-2022 as mentioned above.  Since we wish to forecast in CAD, hence the  prices are converted to CAD by dividing by a factor of 62 and that is done in the R code file. |
| Low | Integer  Unit: “INR” (Indian Rupee) | This is the lowest price of gold shown daily from 2014-2022 as mentioned above.  Since we wish to forecast in CAD, hence the prices are converted to CAD by dividing by a factor of 62 and that is done in the R code file. |
| Volume | Integer | This is the traded volume for the day. ( Traded here means quantity bought or sold in the market for the day). For example, in the case of stocks, it is the number of stocks bought and sold while trading on a specific day. |
| Chg% | Integer | It denotes the % Change from the previous price of the last day. |

**Inspiration for the data and Forecasting**

Gold has always been one of the precious metals in terms of its price value and the need for people to trade Gold to build their wealth. In terms of long-term investing, gold is considered one of the hedge fund strategies against growing inflation trends that we have seen for the last 10 years.

However, in this project, we want to understand the history of fluctuations in gold prices and chose this topic as one of the forecasting challenges.

* **What to achieve from the time series analysis and forecasting?**

1. We want to present the various time series analysis processes of how to decode the fluctuations in the daily “Open” prices of gold and study the past trends of gold prices. Some of the techniques used would be autocorrelation, seasonal adjustment methods, ACF plots and Time series decomposition.
2. By forecasting gold prices for the next period of 100 days, we want to identify clues of long-term cycles of how gold prices would potentially move in the future. This can help investors correlate these changes will other assets in the market and how they can leverage that to build their long-term investment portfolio.
3. Finally, we aim to showcase various forecasting models and evaluate their performance by showing accuracy results and metrics that calculate which models show better accuracy as compared to others.
4. We wish to study trends of the past 5 years from 2018 to 2022 to make the basis of the analysis and forecasting.

# **VISUALIZATION METHODS FOR TIME SERIES ANALYSIS**

* **Step1: Time Series Plots**

Technical steps followed while plotting time series in R :

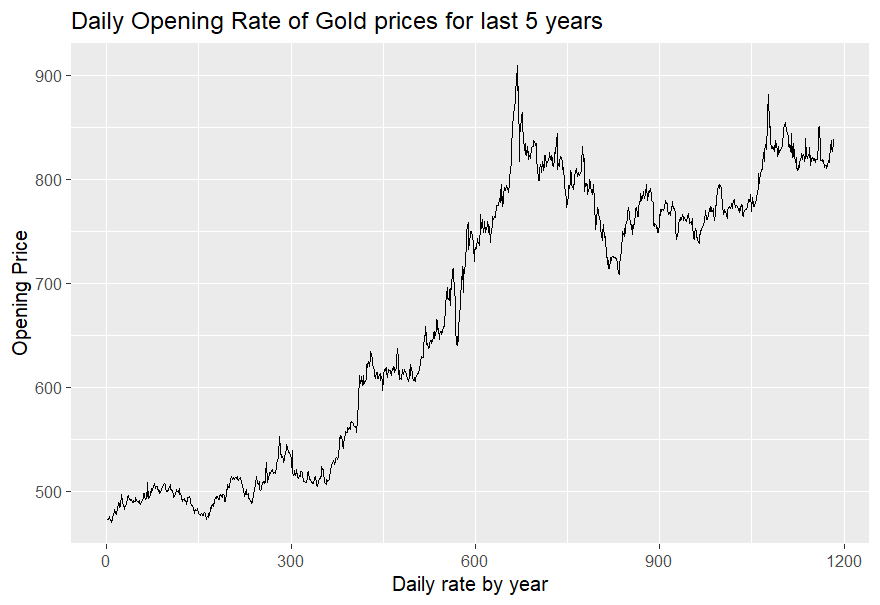
1. We imported the CSV dataset in R and visualized the data summary to check instances of date dimension and pricing variables.
2. Since the date dimension was already in the correct format, datetime conversions were not needed.
3. We found time gaps within the dates like we say missing date of Jan 6 -7 in 2018 and similar trends were seen in all years. This was because these were weekends and since markets are closed at weekends, we do not have data for those days.

Hence, to simplify our plots, we introduced a new column index using **rowid\_to\_column() in R** which is a sequential series of values starting from 1 and filled sequentially for data from Jan 1, 2018 till August 5, 2022.

1. We will use this “**index**” column to act as a time dimension since gaps in the “Date” dimension will show unnecessary horizontal lines in plots which will, result in the inaccuracy of forecasting and unwanted errors in R codes.

**Initials observed trends**:

* **Plot 1**



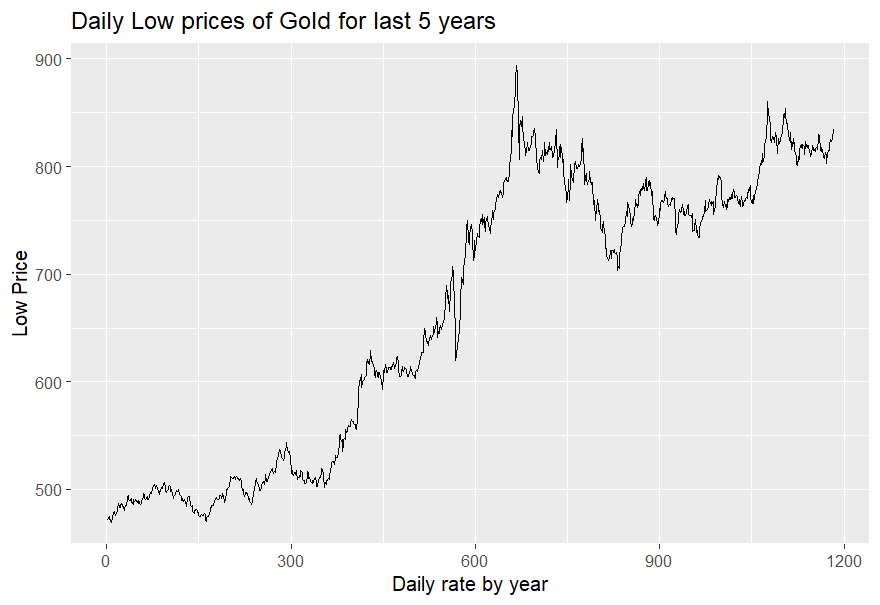
* The above plot shows an overall rising trend of daily Open prices plotted against the daily against each day.

(Note: we can intuitively see a rising trend from 0 to 300 which is the first 300 days of a trend starting from Jan 1, 2018, then 300 to 600 for the next 300 days, and so on and so forth.

* There is a major jump in the Open price of Gold from **CAD 790 to CAD 910 at around the 610th index which is July 8, 2020**. This was around about the time the COVID outbreak happened and gold reached its highest price points due to high inflation that was at its peak.
* Another spike can be seen 1100th and 1200th indexes where **gold open price jumped from around 780 to 880 CAD which was March 7th, 2022.** This was around time when the Ukraine and Russia war caused the spike.
* There are some major drops that can be observed as well:
* Drop from around CAD 710 to around CAD 640 in the week of March 17, 2022
* Another drop point is from around CAD 840 to CAD 710 in the last week of March 2021. This was accounting to rise of interest rates of US dollars that saw some downward trend in gold prices.
* Seasonal trends are highly inconclusive from initial plots and also there is very little we can talk about cyclic trends in data.

A graph of gold prices

Description automatically generated



The above plots have overall similar trends as Open Prices

# **TRANSFORMATIONS**

In practice, usually the variation in data is too random to visualize and we see an abnormal increase or decrease in variation with the actual level of the time series.

Hence in such case, it is often good to use some form of mathematical transformations to reduce this variation. For instance, in our time-series plot, we see that the size of peaks when there are mostly small spikes in open prices are nearly about same (see red circles below). Except for some sudden big spikes as explained earlier in plot 1.

In the same way, few of the small decreases in gold prices throughout the time series have near about similar size(see in green below) except few sudden dipping points in the time series. However, there are quite a few points where decrease is quite sudden.

A graph showing the price of gold

Description automatically generated

Hence to streamline the sudden upward and downward spikes in data, we will run the “Box-Cox” transformation. Box-Cox uses logarithmic transformations based on some value of lambda that satisfies the below mathematical equation.

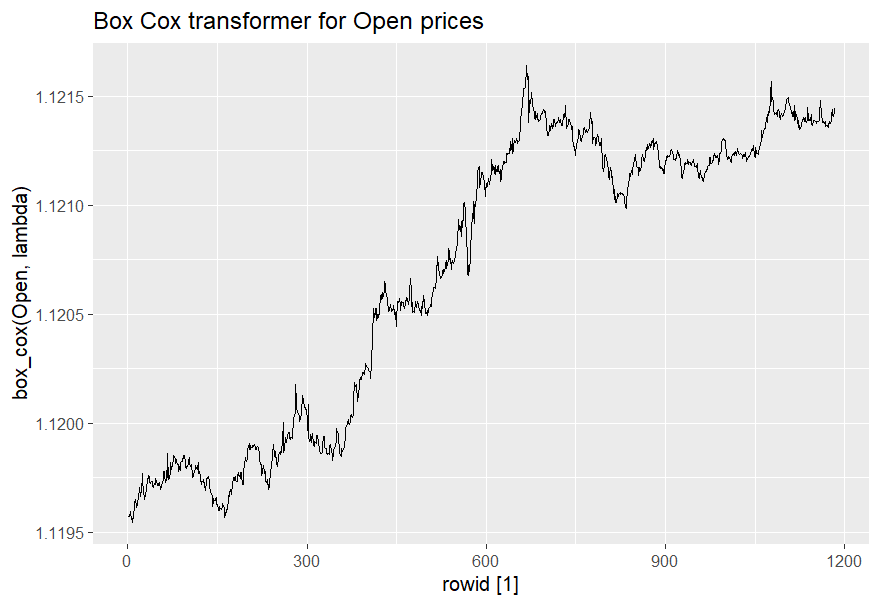
wt= log(yt)if λ=0;

(sign(yt)|yt|λ−1)/λ, otherwise.

In other words, lambda=0 will simply be a logarithmic transformation while a non-zero value of lambda will follow a more aggressive transformation. He,cem determining the value of lambda is crucial in getting the right amount of transformation that suits your time series. A negative or large value of lambda can give inaccurate forecasts with high residual errors.

Below are the steps for the same :

1. First , we use a guerrero feature in R that can be passed in features() and it gives some kind of optimal lambda that would best fit for your timeseries.
2. We can use autoplot() to generate the plot of transformed values of Open prices against time to get below plot.



**ACF PLOT**

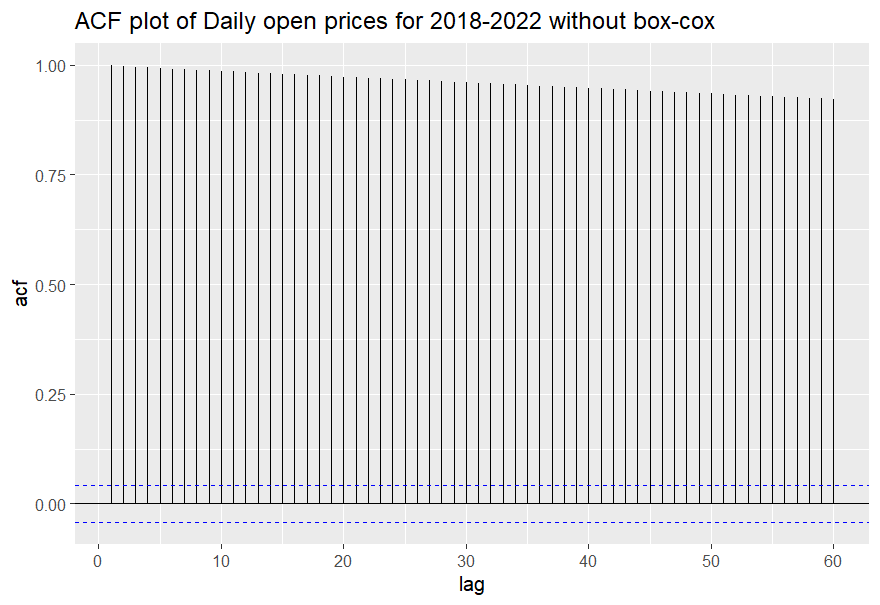
* We need an ACF plot to show autocorrelation of the time series in terms of lags between all pairs of time ( yt, yt-1) in the time series. In other words, we are interested in seeing the series of correlation coefficients(rk) in data at two different consecutive periods in time. Mathematically

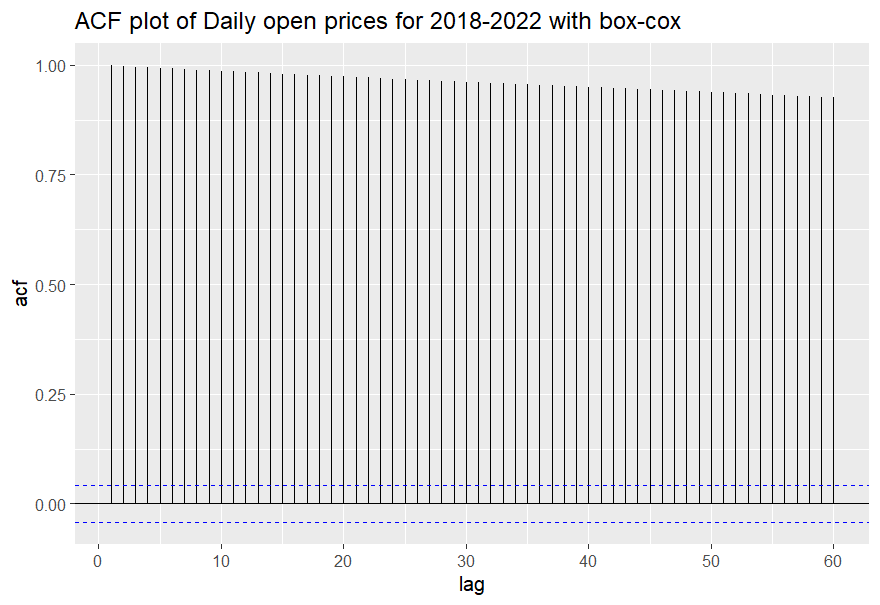
A math equation with numbers and symbols

Description automatically generated with medium confidence

Here, T is the length of the time series and rk represents the sum of all correlation values between yt and yt-k. The value of k represents how many number of lags we want to take to observe the correlations at different time intervals.

Y represents in our case, the correlation coefficient between price values of Open prices of Gold at each of the t and t-k intervals (t is k+1 to T)

****

****

* In the ACF plots above, we generated plots using ACF() that takes a tsibble object (in our case tsibble containing time index and Open gold prices) and lag\_max (in our case we took 60 as lag\_max) which represents value of T, i.e, the length of the time series.
* From the ACF plot for open gold prices, we see a strong positive correlation over different intervals of time since the correlation values are well all above the blue line above 0.
* We also do not see any hints of seasonality in the ACF plot as we can see from above plots.

**Time Series Decomposition**

We use decomposition to categorize our time series in below three components:

**Trend** – this simply shows the smoothened version of the time series by smoothening the jumps in data over the periods.

**Seasonality –** It shows the seasonal effects in different seasons of the time period. This is more evident in case we have a time index that is in quarters or monthly or on basis of any other time dimension.

**Random**( white noise) – Random is just the component in time series that we get after subracting the seasonal and trend patterns and result is all the random pattern that is extracted from the data and is not useful in making any analysis of the series.

In our case, we plotted the decomposition for the Open prices of Gold and we used classical decomposition approach:

We used an additive approach of classical decomposition, which mathematically looks like below:

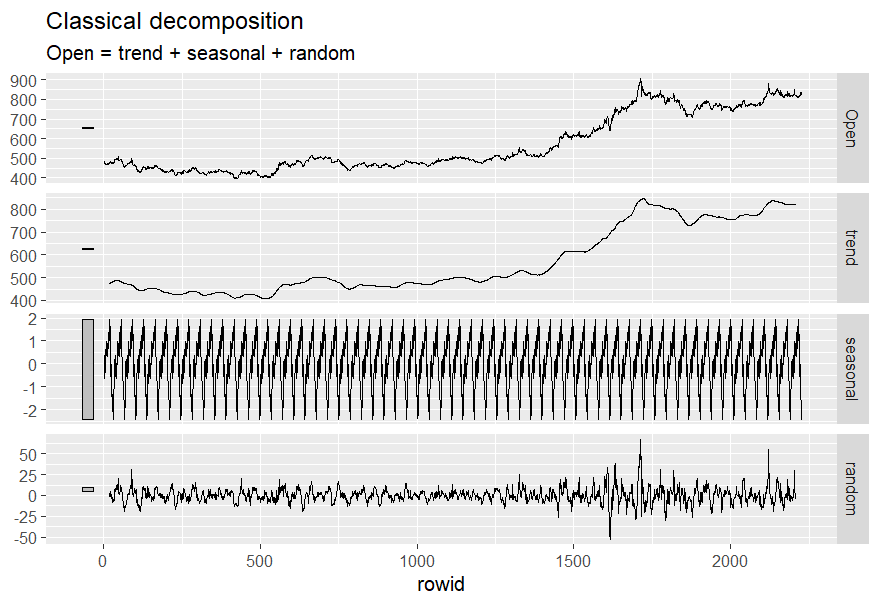
yt=St+Tt+Rt,

Here, **yt** -represents the actual data to decompose

**St**- Seasonal component

**Tt**- Trend component and

**Rt**- random component

****

* From the above plot, we see from the “trend” section of the decomposition that over the last 5 years, the Open prices have had an upward increasing trend starting from CAD 500 in 2018 to CAD 800 in 2022.
* We can see two upward-looking bumps around years 2020 ( COVID) and year 2022 (Ukraine-Russia war) . These are the same results that were evident from Plot 1 we did at the start of the time series analysis. The only difference is that here the decomposition step showed them more prominently.
* Another observation from “season” section of the decomposition is that there is not enough evidence of seasonality in the time series as there is no evident seasonal pattern that might show any reasons for a change in Open prices of Gold at certain time intervals. Hence it is safe to say that gold prices are not muc affected from seasonality.

# **FORECASTING AND ANALYSIS**

There are three basic forecasting models for time series analysis that we are going to consider for this project and below are the considerations for these models:

1. Mean Forecast:

Also known as the average method as the idea behind the mean forecast is that it considers all the future forecasts as the mean of the historical trends over time.

Mathematically we can say, y-bar = (y1 + y2+….+yt)/T, where T is the total length of the time series.

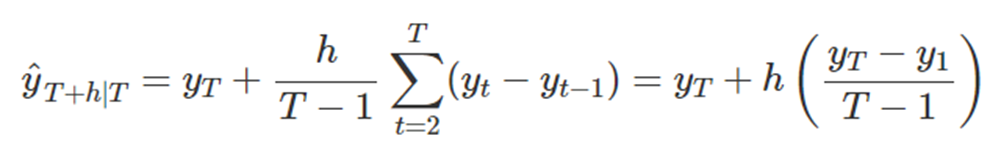
1. Naïve Forecast

This forecast model takes the last observed value of the trend as the next forecast and is more useful for financial and stocks-related data.

1. Drift Forecast

Drift forecast on the other hand, forecasts the future by taking and last observed value and adding it to the average of the rest of the historical trend values(as we do in mean forecast).

Mathematically we can use below equation to visualize the forecast approach:



yt portion above is the last observed value in train data and the additive part represents the mean of the past historical trends in the data.

* **Forecasting steps followed to fit the above models:**

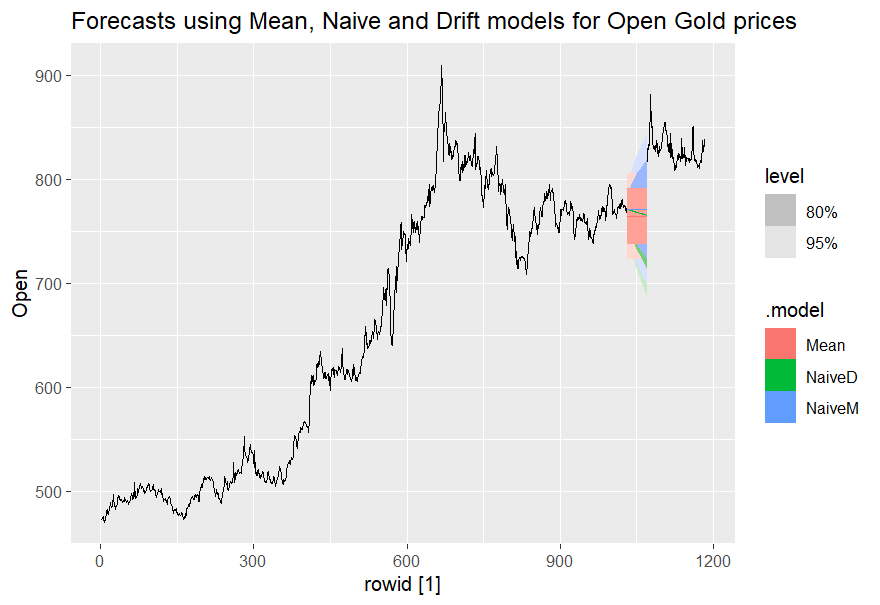
**Step 1**: We created a train test data (gold\_train) and test data(gold\_test) out of the tsibble dataset that we have generated for Open gold prices and hence we will use only the ”Open” column of the tsibble.

In this time series, we are taking all of the data of the year 2021 and for test data we are taking all the Open price values of 2022.

**Step 2**: We fitted the above three forecast models using model() which takes inputs as:

MEAN(Open), NAÏVE(Open), NAÏVE(Open~drift()) for each of the 3 models respectively. Each of these takes the key as an argument that is “Open” for open gold prices. This step generates a mable ( or model table) object that is used to forecast for future gold prices.

**Step 3:** We now run our forecast using the forecast() that the above-fitted model is passed to and in our case, we will forecast for the next 40 days. This gives us a fable object that shows the forecast values for 40 days. The below plot shows the forecast results of the fable returned for the 3 models:



From the above plot we see below results :

* **Mean** forecast shows the next forecast for 40 days that on average the open prices of gold would be around CAD 764. A snap of the R output for mean.

# A fable: 120 x 4 [1]

# Key: .model [3]

.model rowid Open .mean

*<chr>* *<dbl>* *<dist>* *<dbl>*

Mean 1031 N(764, 443) 764.

The **Naïve (**NaïveM in plot above**)** forecast shows CAD 771 to be the forecast for next 40 days. Snap:

*<chr>* *<dbl>* *<dist>* *<dbl>*

NaiveM 1031 N(771, 34) 771.

NaiveM 1032 N(771, 68) 771.

The **Drift** forecast (NaïveD in the plot above) shows a downward trend for the next 40 days from CAD 771 to CAD 765.

* Another observation from the Forecast plot is that the darker-colored shaded portion of the forecast shows an 80% confidence interval of the forecast model while the lighter-colored shaded portion of the forecast shows a 95% confidence interval of the forecast range.

**Step 3**: Visually we see the Naïve method does a decent and a better job as both Drift and Mean forecast models underpredict the forecast value for Open Gold prices as compared to the actual trend in the data.

However, in this step we would evaluate the accuracy using RMSE and MAE metrics.

RMSE shows the mean squared error of the difference of the observed and the forecast values while the MAE gives the absolute mean of the difference of observed and forecast values. Below are the results generated from the accuracy() which takes the fable and our test data of Open gold prices we generated for the year 2022 at **Step1.**

**Below are the results** :

.model RMSE MAE

*<chr>* *<dbl>* *<dbl>*

1 Mean 25.3 19.7

2 NaiveD 23.5 16.6

3 NaiveM 20.5 13.9

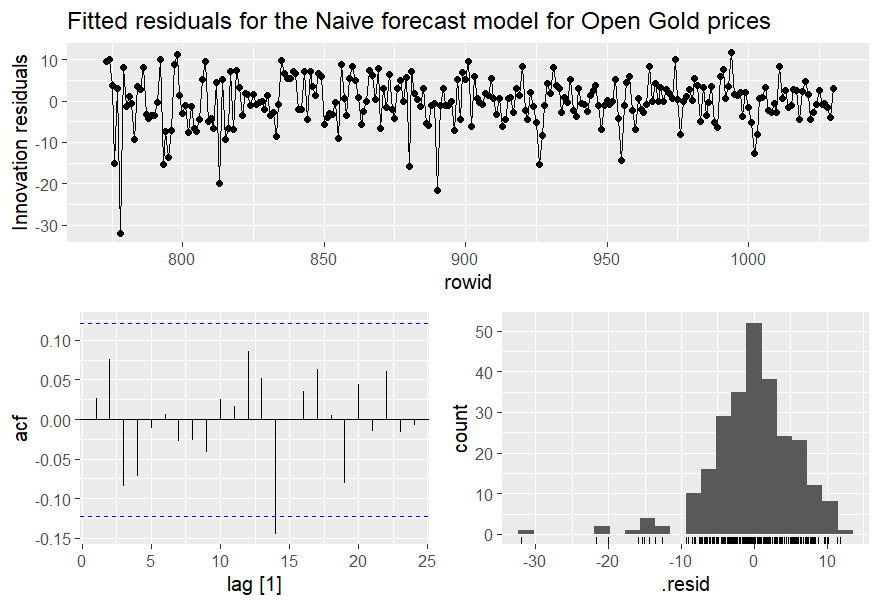
Out of the three models above, we see that the NaïveM has the least RMSE and MAE

Value which means that it is the best model that we can use for our future forecasts.

**Another interpretation of the metrics is that the Naïve method predicts a mean**

**error of CAD 20.5 for the prediction which is the width of the 80% confidence interval as well for the Naïve forecast model which shows an average forecast of CAD 771.**

**Step 4**: Once we have an optimal model above and like in our case is the Naïve forecast model, another approach to check the performance of this model to check its accuracy is the residual plots. Below are the residual plots for the /naïve forecast we generated in R. We used the gg\_tsresiduals(), which takes our fitted model and takes a select() to pass the NaïveM model to the gg\_tsresiduals() which plots :



From the above plot we assess some of the assumptions similar to how we do in linear regression models:

* **Errors should be uncorrelated**:

To check this, we see the innovation residuals plot shows that the residuals are jumping up and down around 0 but have some sort of slight variations looking the height of the peak to the positive above 0 and negative side below 0 that is random and it shows that we have still some really minor accuracies that the Naïve model is missing that can be improved.

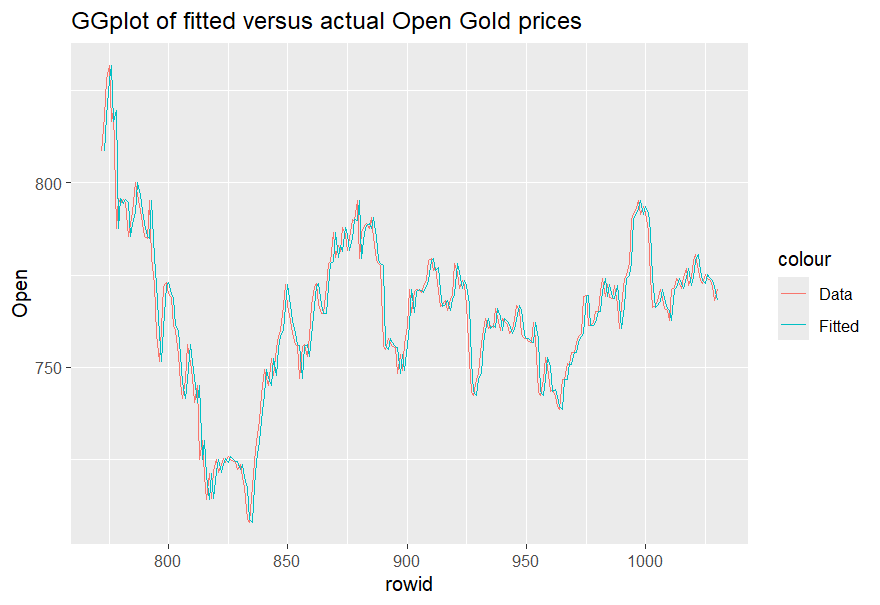
* **Errors are normally distributed and have mean 0**:

Again looking at the residual count plot , we see that it does have decent amount of normality around mean 0. However, there is a presence of some outliers in the count plot above which makes it a bit skewed to the right and that too points to some inaccuracies in the model that should be addressed.

* **Errors should have constant variance**:

This can be looked up from the ACF plot that shows tha autocorrelation plot of the residual errors. Most of the autocorrelations trends is well within the blue line on top and bottom of 0 value and for fair part of plot show some signs of constant variance, however on average it has some of slightly negative values of correlations which makes the variance very slightly negatively autocorrelated. Hence, this also shows some minor improvements that are needed to the model.

Another method to visualize the fitted and the actual data is below :



The above plot looks fairly well fitted and it looks Naïve method has done a decent job and only minor improvements are needed as discussed in the residuals plots above.

**Conclusion on Improvements:**

In our dataset, we see that the different forecast methods, the **Naiive** forecast did a decent job but as future improvement, we can look at some more models like Linear regression by using trend() as one of the predictors or we can also use ARIMA models that take advantage of autoregressive approach using moving averages.