INTRODUCTION TO TIME SERIES
MATH 546

STOCK PRICE ANALYSIS & FORECASTING

AKSHAY SINGH
USAMAH ABDULAZEEZ
ZAINAB HASNAIN

The Team

AKSHAY SINGH
asingh149@hawk.iit.edu

USAMAH ABDULAZEEZ
uabdulazeez@hawk.iit.edu

ZAINAB HASNAIN
zhasnain1@hawk.iit.edu

Problem Statement

- FAANG stocks **(Facebook, Amazon, Apple, Netflix, and Google)** are highly sought-after investments due to their technological advancements that shape our daily lives.
- Time series analysis can help investors gain insight into **future price** movements of FAANG stocks by analyzing **historical data patterns and trends**.
- This project will explore key concepts of time series analysis, such as **seasonality, trend detection,** volatility analysis, and **forecasting techniques,** to analyze FAANG stock prices.
- The goal is to provide valuable insights and recommendations for investors and analysts by using historical trends and patterns present in the data to **make informed predictions** about future stock prices.

Data Description

Source: We are analyzing the FAANG dataset, containing time series data of stock prices for the companies: Facebook, Amazon, Apple, Netflix, and Google.

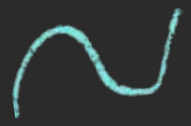
https://www.kaggle.com/datasets/aayushmishra1512/faang-complete-stock-data?select=Apple.csv

Columns in the Dataset

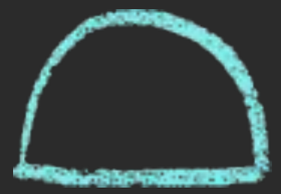
Name	Description
Date	Date of the recorded (Datatype: Date)
Open	Opening price of the stock (Datatype: Double/Float)
High	Max price of the stock for the day (Datatype: Double/Float)
Low	Min price of the stock for the day (Datatype: Double/Float)
Close	Closing price of stock for the day (Datatype: Double/Float)
Adj Close	Data is adjusted using appropriate split and dividend multipliers for the closing price for the day. (Datatype: Double/Float)
Volume	Volume is the physical number of shares traded of that stock on a particular day (Datatype: Integer)

Dimension and range of years for each company

Company	Dimension	Range
Facebook	2076 x 7	2012 - 2020
Amazon	5852 x 7	1997 - 2020
Apple	10016 x 7	1980 - 2020
Netflix	4851 x 7	2002 - 2020
Google	4041 x 7	2004 - 2020




Approach

1. Data Cleaning & EDA

- Checking and aggregating missing values
- Visualising Data

2. Data Decomposition

- Fitting a Loess Curve
- Extracting Trend & Seasonality
- ACF and PACF

3. Stationarity Check

- Augmented Dickey Fuller (ADF) Test
- Differencing the series

4. Model Creation

- ARIMA Model
- SARIMA Model
- Holt Winters Model

5. Model Evaluation

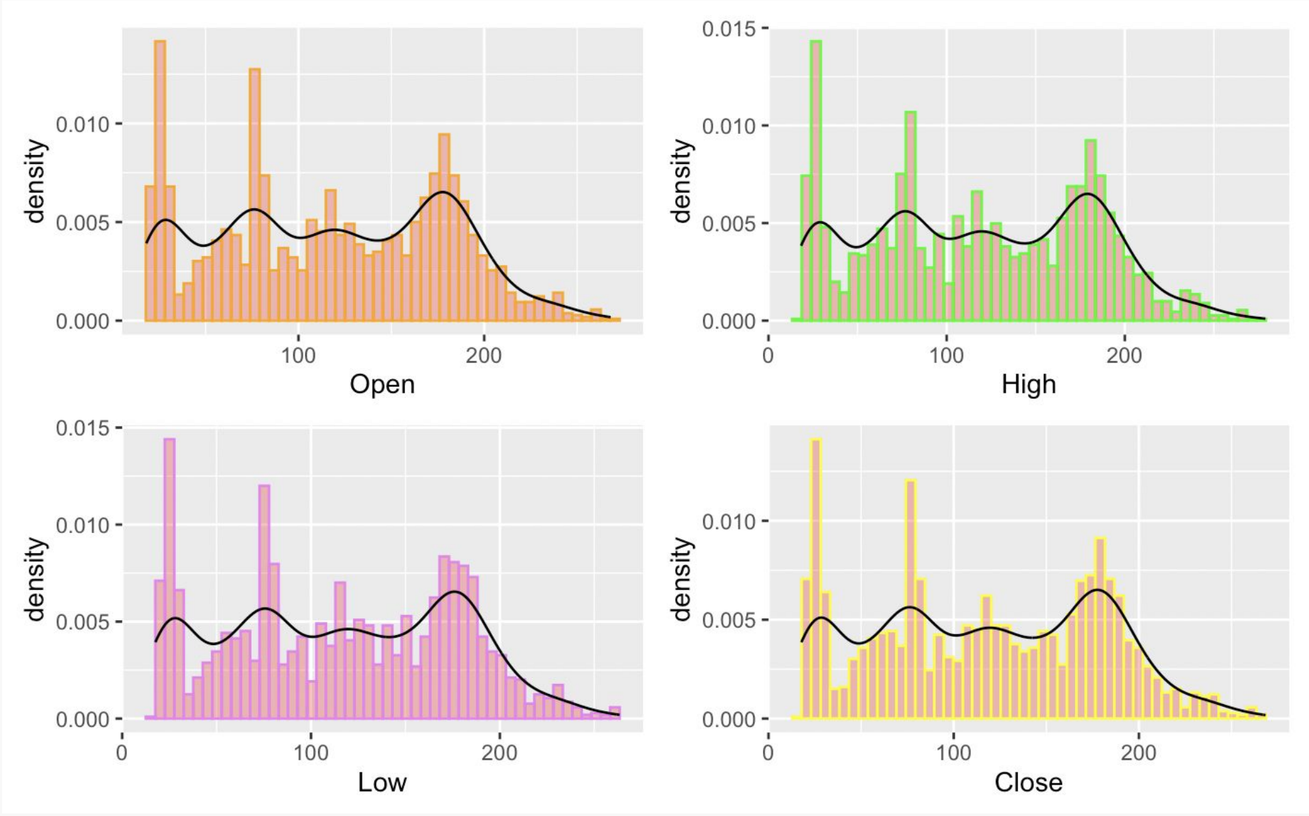
- Residuals Normality
- Ljung Box Test
- Accuracy Scores

6. Conclusion

- Summary of our Findings
- Future Scope

Data Cleaning & EDA - Facebook

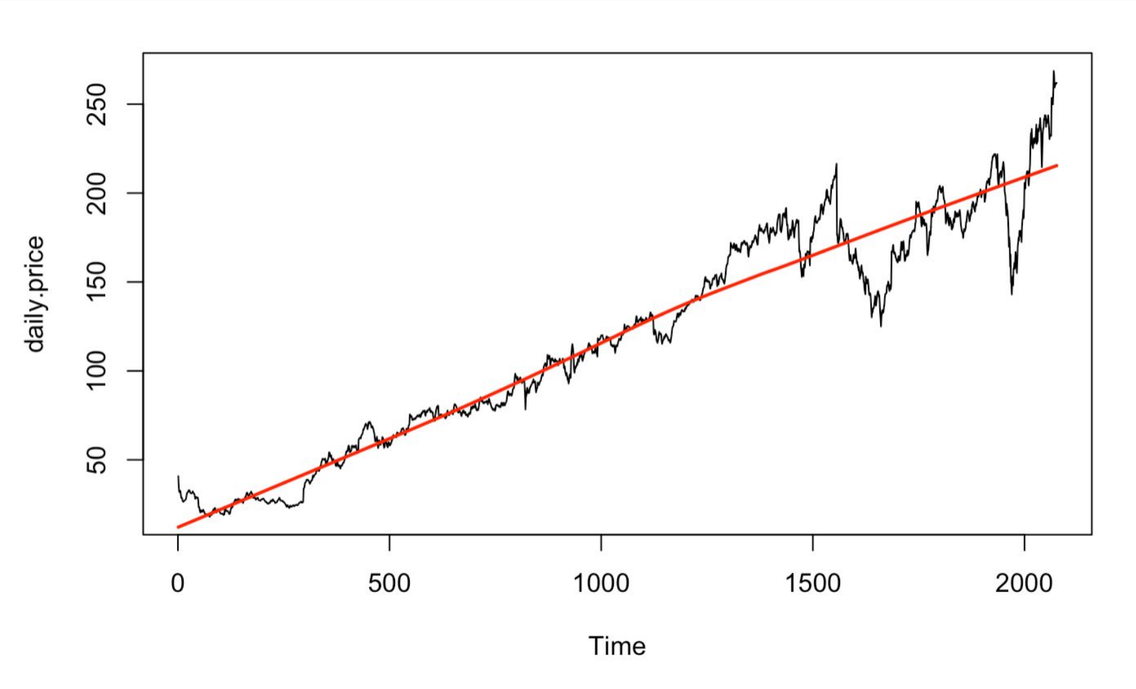
- We have decided to **drop the 'Volume' column**
  because it's not relevant to our problem statement and we want to **perform univariate analysis.**
- Since we are not concerned about the hourly fluctuations in the price of the stock, we can find **an average price for the day** for analysis.
- Since original data has daily frequency, we can use it as it is for daily time series.



Back to **Agenda Page**

Fitting a Lowess Curve

- Fitting a LOWESS curve to daily stock data can provide valuable insights into the **underlying trends and patterns** in stock prices, helping investors and analysts make more informed decisions.
- By smoothing out short-term fluctuations and noise, the LOWESS curve **highlights the overall direction and potential turning points** in the stock price.
- This can be particularly useful for identifying trends, detecting anomalies, and understanding the market's behavior, enabling better risk management and strategic investment planning.
- We are using lowess() function from Kendall library to do this in R.



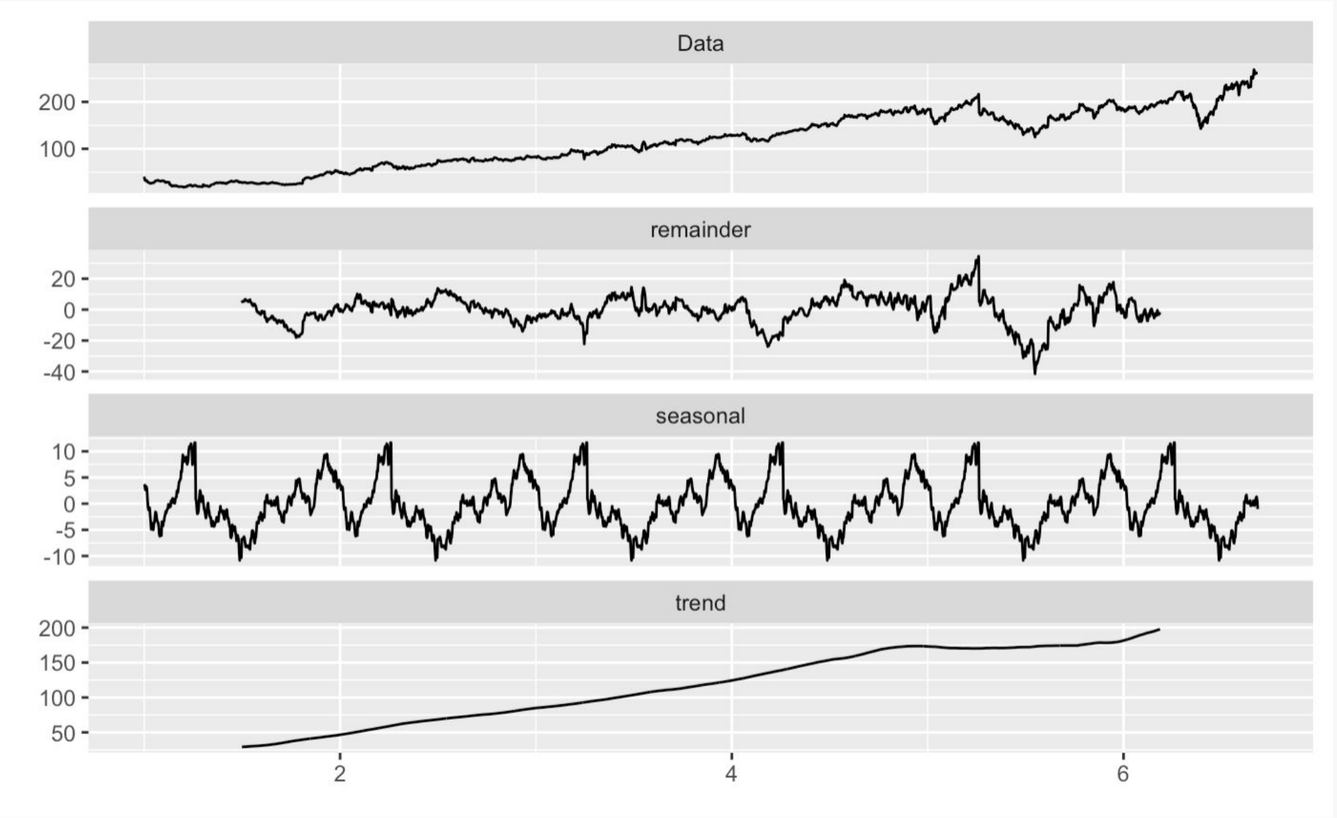
Data Decomposition

- We do this to uncover the underlying components of stock price data to better understand market behavior and trends.
- Decomposition Components:

  Trend: The long-term movement of the stock prices.

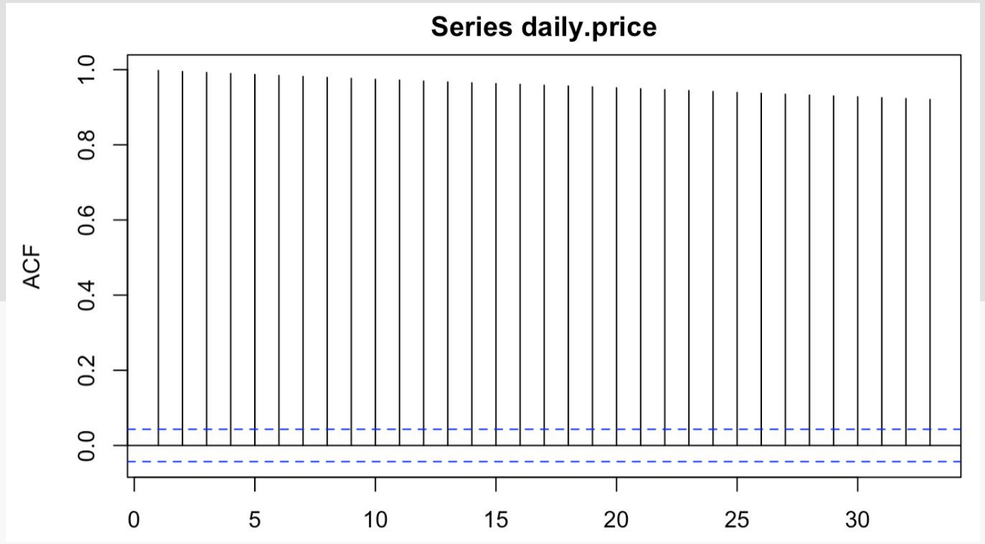
  Seasonal: Periodic fluctuations that occur regularly, such as weekly or monthly patterns.

  Remainder (Residual): Unpredictable variations not explained by the trend or seasonal components.
- We have used **autoplot** from the ggfortify library to display the decomposed components.



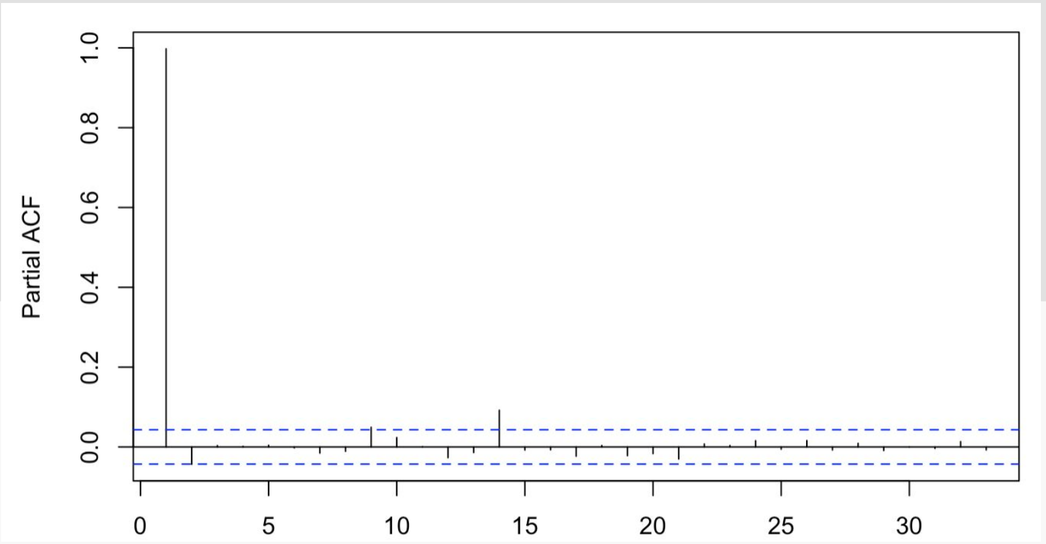
ACF (Autocorrelation Function)

ACF measures the correlation between a time series and its lagged values, i.e., the values from previous time periods. It helps identify the presence of linear dependence between the observations at different time lags.



PACF (Partial Autocorrelation Function)

PACF, on the other hand, measures the correlation between a time series and its lagged values after accounting for the correlations at shorter lags. In other words, PACF represents the direct effect of a previous observation on the current observation, with the influence of the intervening observations removed.



Stationarity Check

Augmented Dickey-Fuller Test

- Determine if the daily stock price data is stationary.
- Apply the Augmented Dickey-Fuller (ADF) test using the R adf.test function from the tseries library.
- p-value is 0.07, which is greater than 0.05. We fail to reject null hypothesis, this series **is not stationary.**

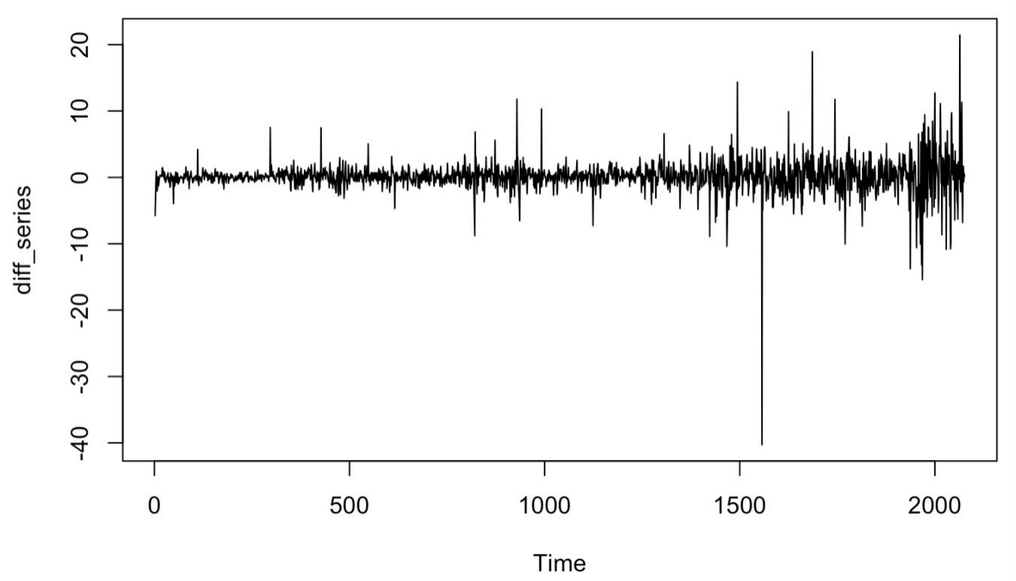
First Order Differencing - To Make it Stationary

- To transform non-stationary daily stock price data into stationary data to ensure reliable time series modeling and forecasting.

Follow Up ADF Test

- p-value is 0.01. We reject null hypothesis, this newly created series from first order difference **is stationary.**

```
        Augmented Dickey-Fuller Test

data:  daily.price
Dickey-Fuller = -3, Lag order = 12, p-value = 0.07
alternative hypothesis: stationary
```




```
        Augmented Dickey-Fuller Test

data:  diff_series
Dickey-Fuller = -13, Lag order = 12, p-value = 0.01
alternative hypothesis: stationary
```

Model Fitting & Forecasting

Model Creation - ARIMA Model

For Differenced Time Series (Facebook)

- Using auto.arima() in R, we find the order with the lowest AIC, AICC, and BIC values.
- It recommends ARIMA(2, 0, 0) as the order of our arima model when we feed it the differenced series which is stationary.

$p =$	order of the autoregressive part;
$d =$	degree of first differencing involved;
$q =$	order of the moving average part.

```
Series: diff_series
ARIMA(2,0,0) with non-zero mean

Coefficients:
         ar1      ar2   mean
      0.104  -0.035  0.106
s.e.  0.022   0.022  0.056

sigma^2 = 5.57:  log likelihood = -4724
AIC=9457   AICc=9457   BIC=9479
```

For Original Time Series (Facebook)

- It recommends ARIMA(2, 1, 0) as the order of our arima model when we feed the non-stationary series into it.
- The '1' in the middle indicates that the series needs to be differenced once to become stationary.
- For better visualization, we will use the original non-stationary time series for the ARIMA model directly, rather than the differenced series.

```
Series: daily.price
ARIMA(2,1,0) with drift

Coefficients:
         ar1      ar2  drift
      0.104  -0.035  0.106
s.e.  0.022   0.022  0.056

sigma^2 = 5.57:  log likelihood = -4724
AIC=9457   AICc=9457   BIC=9479
```

Similarly, we do ARIMA order selection for all other data sets…

```
Series: daily.price
ARIMA(5,2,0)

Coefficients:
         ar1     ar2     ar3     ar4     ar5
      -0.699  -0.574  -0.472  -0.322  -0.129
s.e.   0.013   0.015   0.016   0.015   0.013

sigma^2 = 165:  log likelihood = -23242
AIC=46497   AICc=46497   BIC=46537
```

Amazon

```
Series: daily.price
ARIMA(2,2,3)

Coefficients:
         ar1      ar2      ma1     ma2      ma3
      0.1057  -0.5658  -0.9804  0.5095  -0.4871
s.e.  0.0670   0.0616   0.0693  0.0882   0.0642

sigma^2 = 0.1033:  log likelihood = -2842.7
AIC=5697.41   AICc=5697.42   BIC=5740.68
```

Apple

```
Series: daily.price
ARIMA(0,1,1) with drift

Coefficients:
         ma1   drift
      0.1380  0.3812
s.e.  0.0156  0.1621

sigma^2 = 81.97:  log likelihood = -14632.41
AIC=29270.83   AICc=29270.83   BIC=29289.74
```

Google

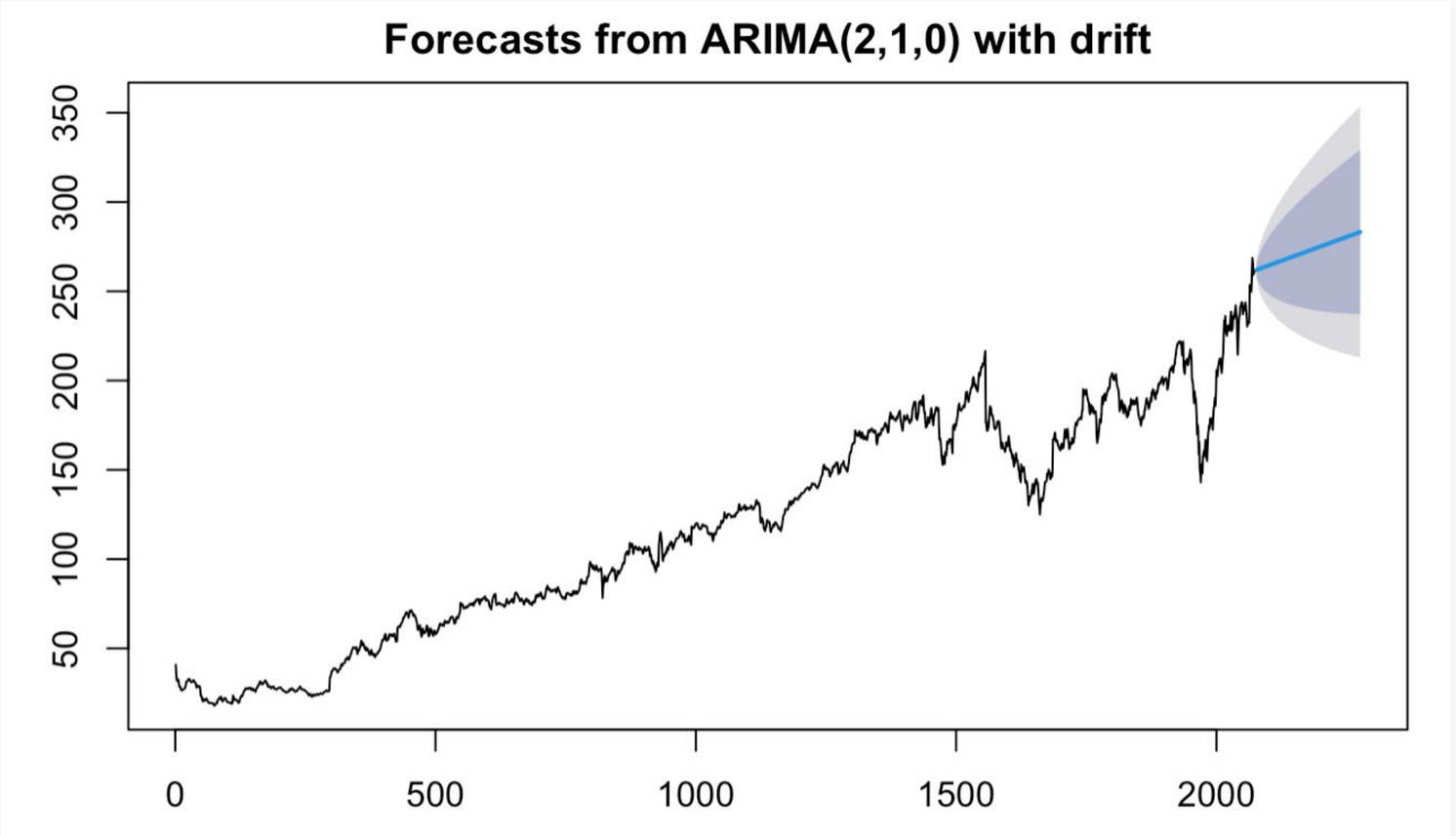
```
Series: daily.price
ARIMA(5,2,0)

Coefficients:
         ar1     ar2     ar3     ar4     ar5
      -0.623  -0.537  -0.326  -0.219  -0.143
s.e.   0.015   0.017   0.018   0.017   0.015

sigma^2 = 10.4:  log likelihood = -11850
AIC=23711   AICc=23711   BIC=23750
```

Netflix

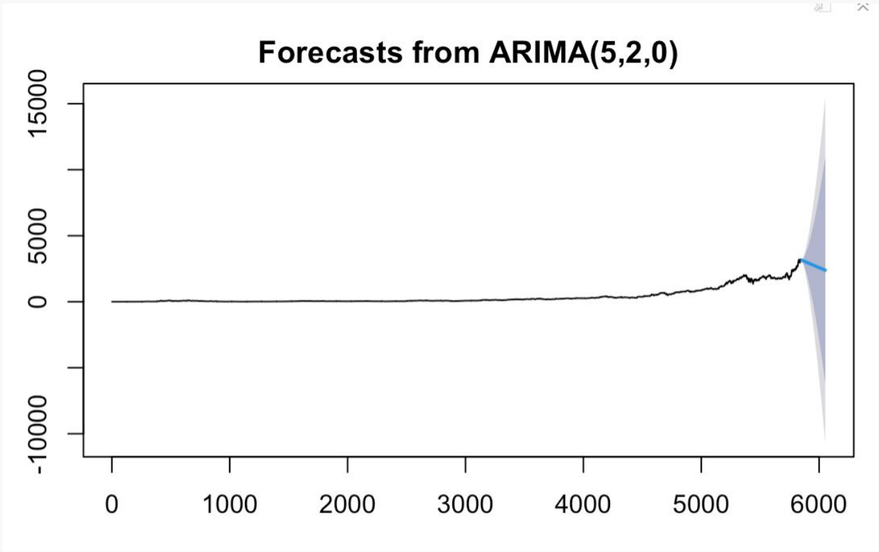
ARIMA - Forecasting

For Facebook's Dataset

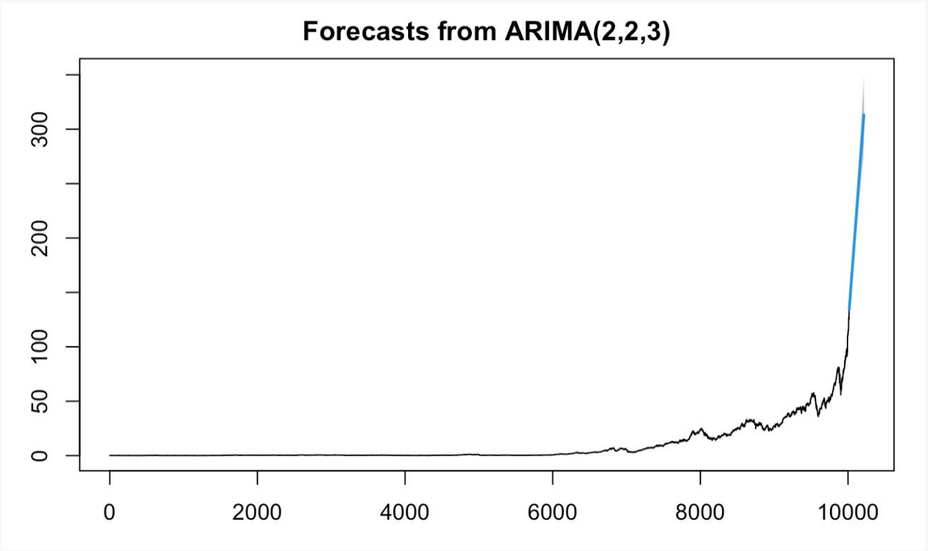
- 'auto.arima()' function for automatic model selection
- 'forecast()' function for generating forecasts
- term = 200: Setting forecast horizon to 200 periods



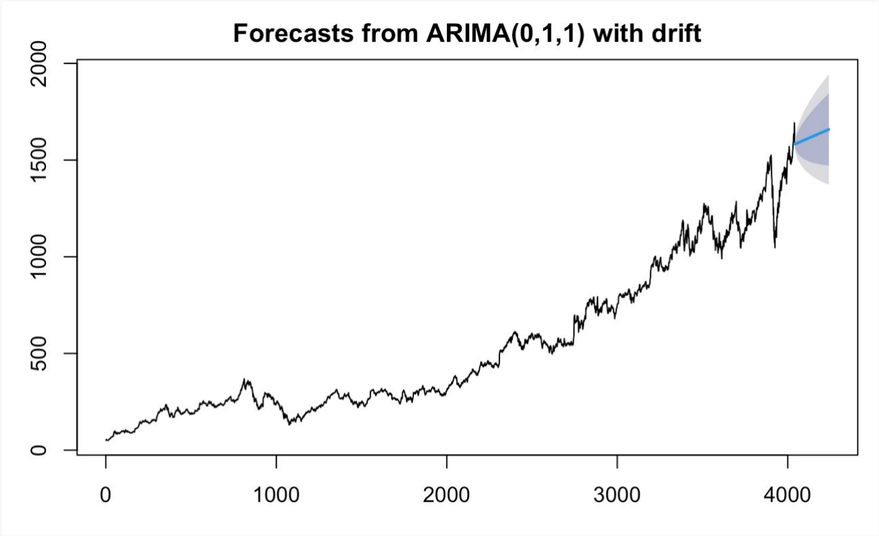
Similarly, for other datasets...



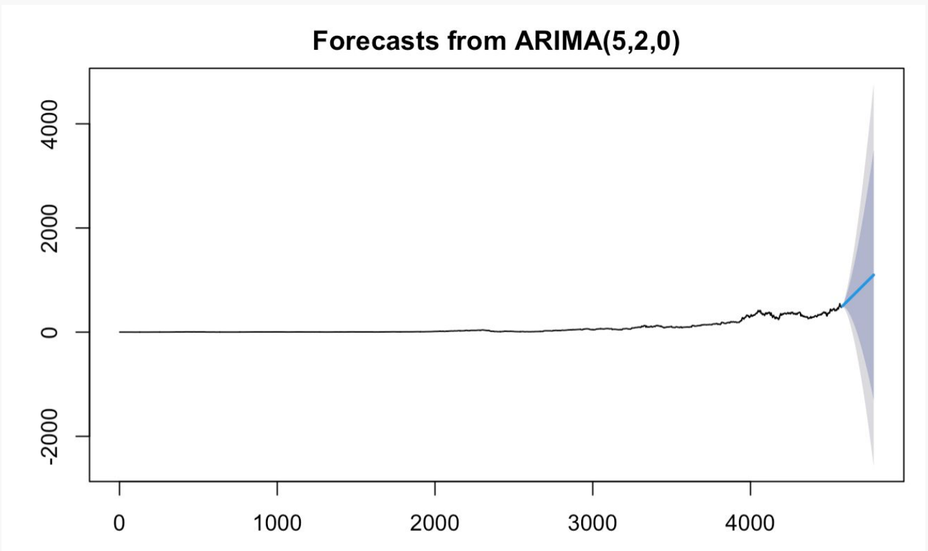
Amazon



Apple



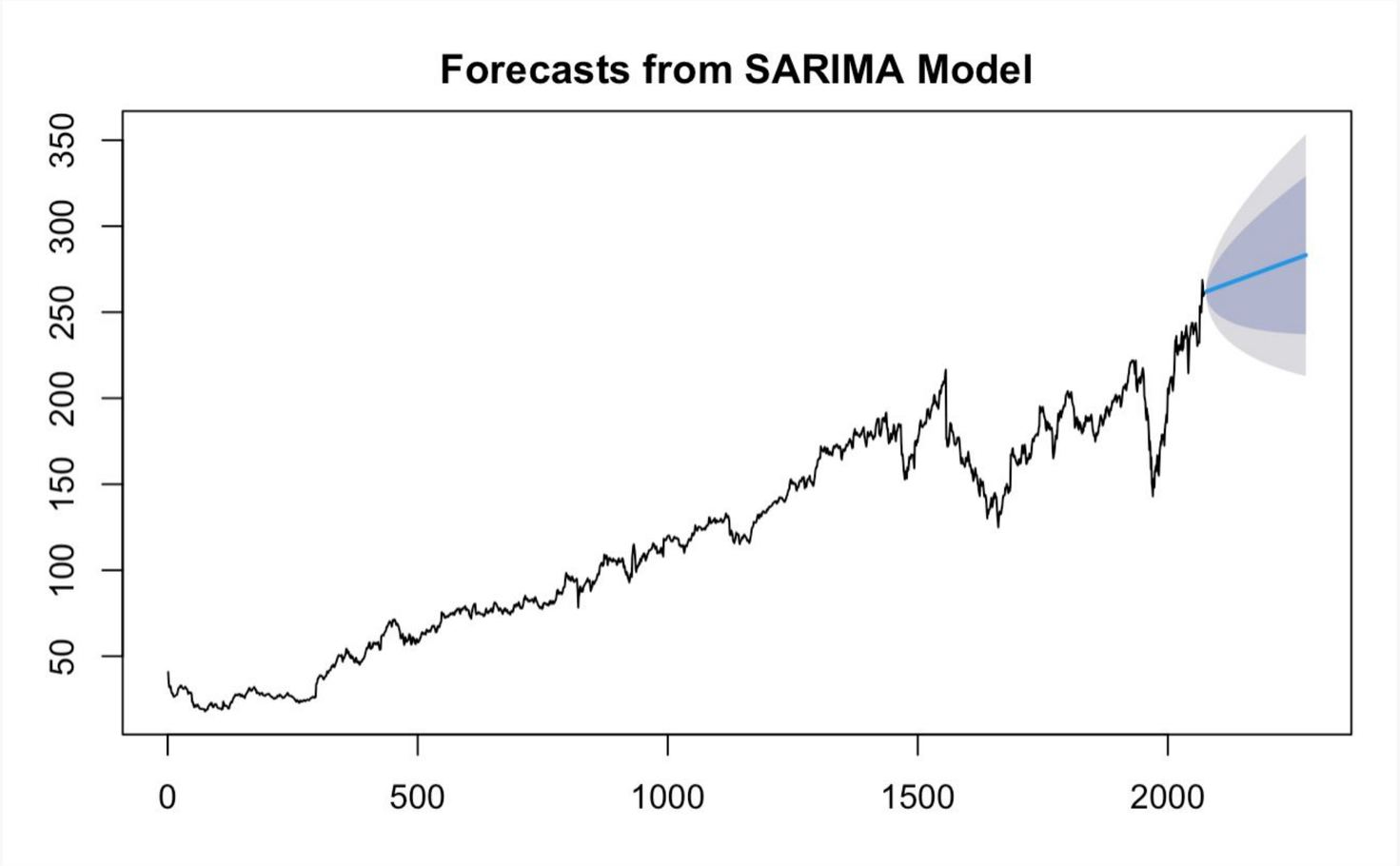
Google



Netflix

SARIMA - Forecasting

The SARIMA model (Seasonal Autoregressive Integrated Moving Average) is an extension of the ARIMA model, specifically designed to handle time series data with seasonal patterns. It combines ARIMA components with seasonal differencing, autoregressive, and moving average terms to capture and forecast seasonal variations effectively.

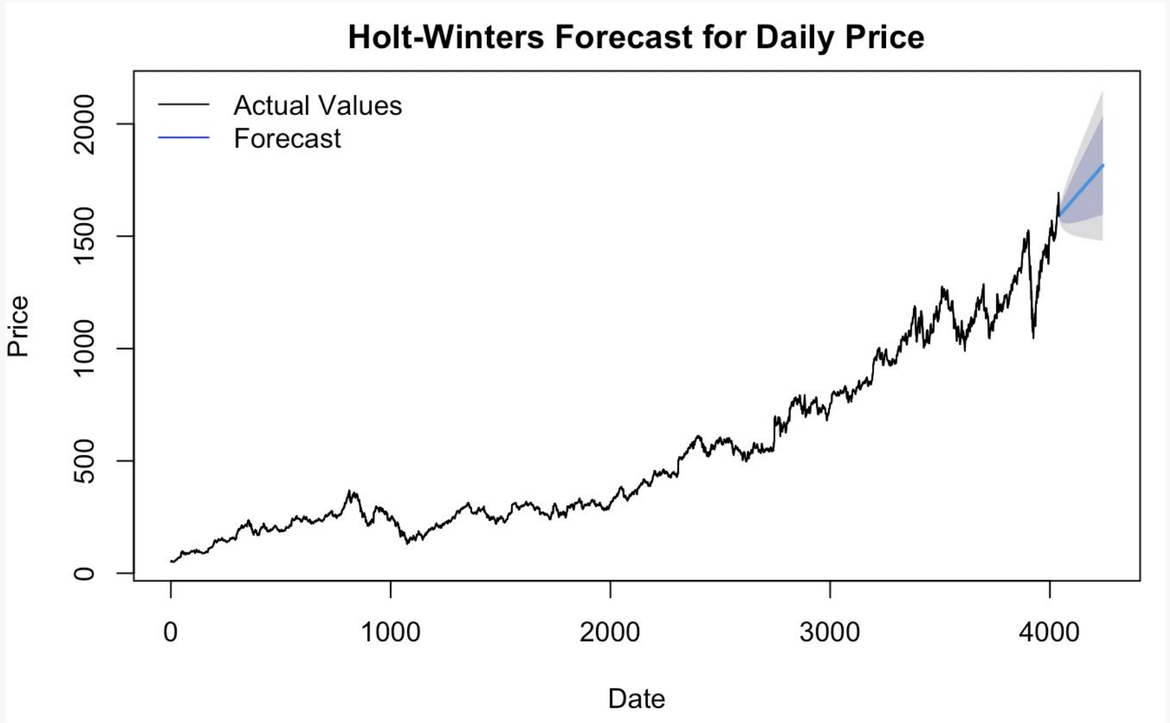


//since results aren't any different, we won't go in depth

Holt-Winters Model

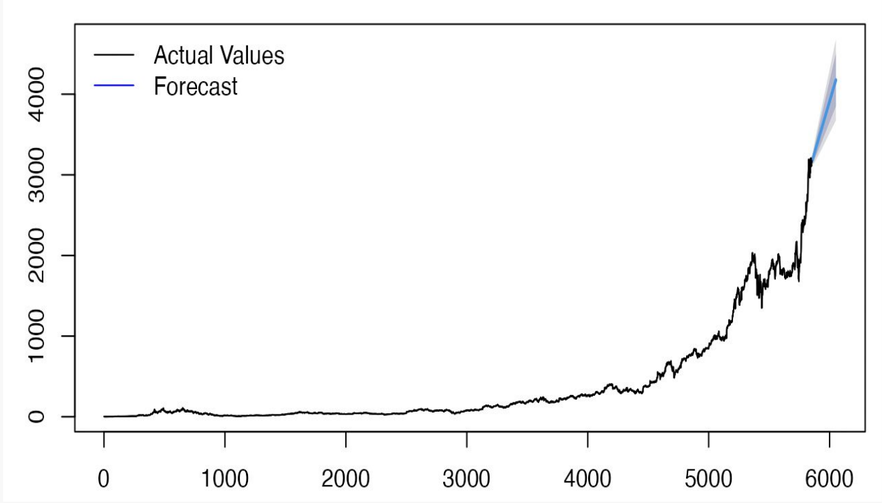
Holt-Winters Forecast, also known as Exponential Smoothing State Space Model, is a popular forecasting method for time series data with seasonal patterns. It incorporates trend and seasonality components to make predictions. The method consists of three variations:

1. Single Exponential Smoothing (level)
2. Double Exponential Smoothing (level and trend)
3. Triple Exponential Smoothing (level, trend, and seasonality)

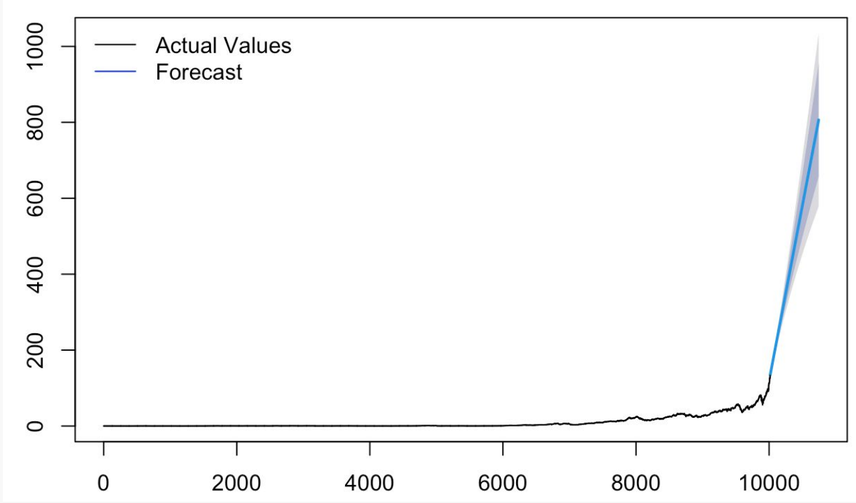


For Facebook Dataset

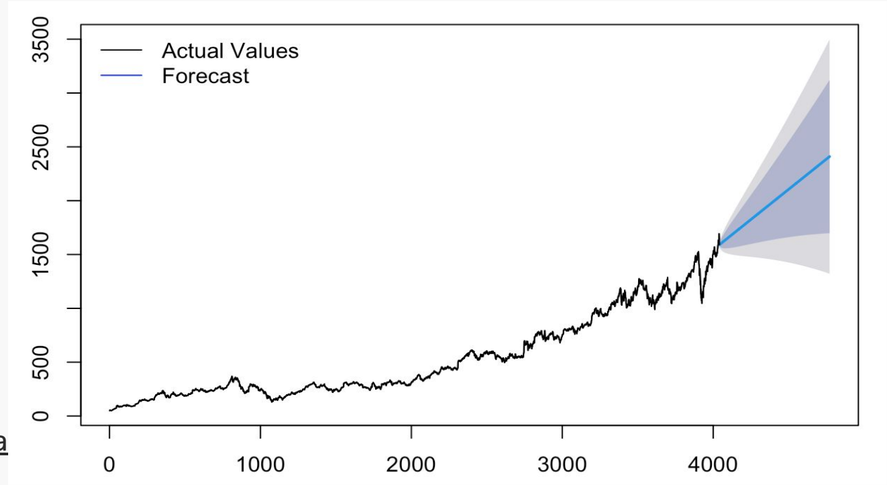
Similarly, we apply the Holt-Winter forecast to all other datasets…



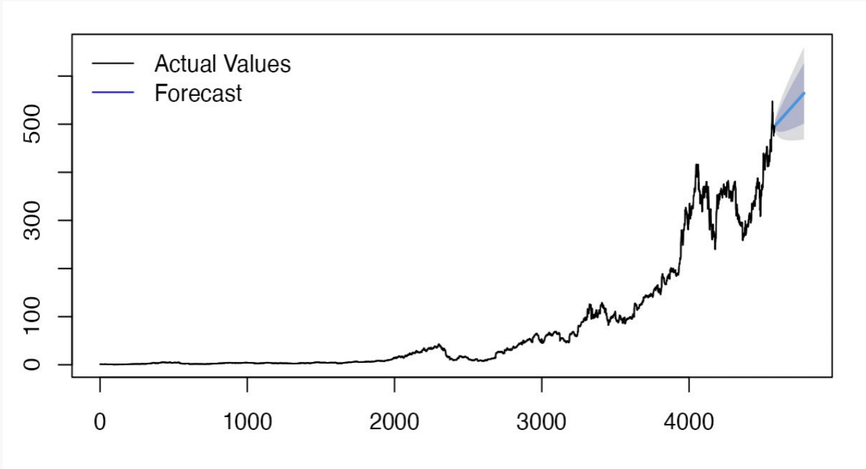
Amazon



Apple



Google

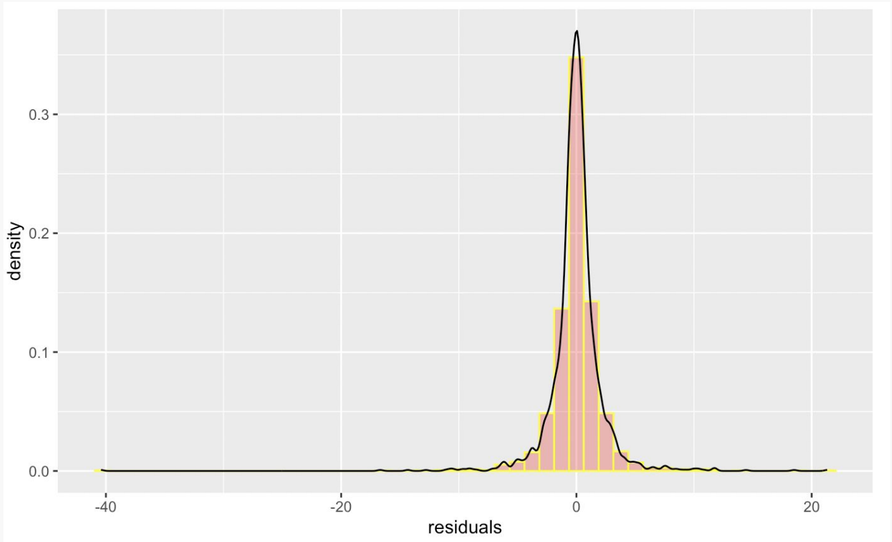


Netflix

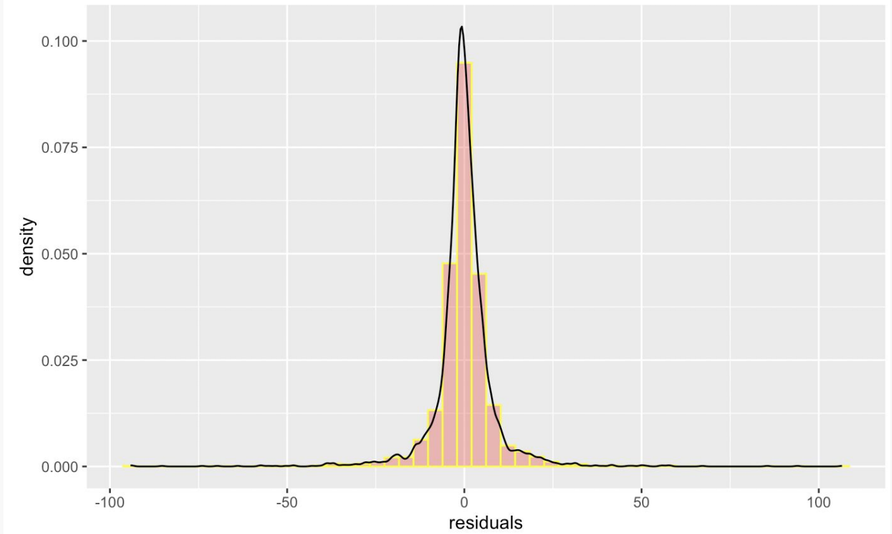
Model Evaluation

Normality of Residuals - Facebook Data

- We fit a histogram of residuals using the ggplot() function in R
- A normal distribution of residuals is desirable, indicating that the forecast errors are randomly distributed around zero.
- We check whether the residuals are centered around mean 0 and have a constant standard deviation
- If the residuals are not normally distributed, it may suggest that the model is not accounting for some important patterns or features in the time series.



Residuals of ARMA model



Residuals of Holt-Winters

Box-Ljung Test - Facebook Data

- Statistical test used to check for the presence of autocorrelation between residuals in a time series
- Null hypothesis = no significant autocorrelation
- The Ljung-Box test produces a test statistic and a p-value
- If the p-value is less than the significance level (e.g., 0.05), we reject the null hypothesis
- conclude that there is evidence of autocorrelation in the time series.
- p-value > 0.05, accept null hypothesis

```
        Box-Ljung test

data:  resid(arima_model)
X-squared = 11.534, df = 10, p-value = 0.3174
```

```
        Box-Ljung test

data:  resid(hw_model)
X-squared = 106.95, df = 10, p-value < 2.2e-16
```

Accuracy Scores

- ME (Mean Error): The average of forecast errors, indicating the overall bias in the model.
- RMSE (Root Mean Squared Error): The square root of the average squared forecast errors, emphasizing larger errors.
- MAE (Mean Absolute Error): The average of absolute forecast errors, reflecting the average magnitude of errors.
- MPE (Mean Percentage Error): The average percentage of forecast errors, indicating the model's percentage bias.
- MAPE (Mean Absolute Percentage Error): The average percentage of absolute forecast errors, showing the average relative magnitude of errors.
- MASE (Mean Absolute Scaled Error): The average of absolute forecast errors, scaled by a benchmark model's error, providing a relative accuracy measure.
- ACF1 (Autocorrelation at Lag 1): The correlation between a time series and its lagged version, assessing the residual autocorrelation in the model.

Facebook Data Results

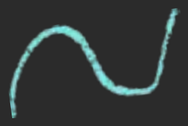
Top: ARMA Model Accuracy Score
Bottom: Holt Winters Accuracy Score

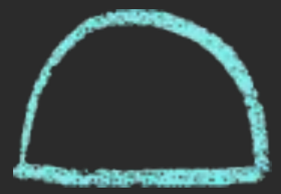
	ME	RMSE	MAE
Training set	0.0001708322	2.35751	1.324617
Holt-Winters:	-0.09445041	9.15458	5.094777

	MPE	MAPE	MASE	ACF1
	-0.06931335	1.225756	0.9816849	-0.0007825155
	-0.1359185	1.079564	1.004808	0.1340055

Model Comparison & Summary

Higher Overall Accuracy for		ARMA	HoltWinters	Comments
1	Facebook	✓		• Only ARMA model passes Box-Ljung Test • Residuals seem to be normally distributed but ARMA has unequal variance • HW residuals have higher variance
2	Amazon	✓		• None of the models pass the Box-Ljung test • Residuals seem to be centered around mean with low variance • Relative difference in accuracy scores is small
3	Apple	✓		• None of the models pass the Box-Ljung test • Residuals seem to be centered around mean with high variance • Relative difference in accuracy scores is small
4	Netflix		✓	• None of the models pass the Box-Ljung test • Residuals seem to be centered around mean but with HW model has lower variance • 4 out of 7 errors are significantly lower for HW
5	Google	✓		• None of the models pass the Box-Ljung test • Residuals seem to be centered around mean but with high variance • Relative difference in accuracy scores is small



Future Study

- Explore the ***volume of stocks sold*** feature
- It seems to be seasonal, find ARIMA and SARIMA orders and compare the models
- Check for autocorrelations between volume and mean daily price
- Make a multivariate time series and evaluate its performance
- **Comparison between univariate and multivariate time series** performance
- Use **GARCH model** to take predict the volatility of the stock prices

Thank You!