SARIMA Model in Sales and Finance

Week 8 – Advanced Topic 1

- Azimil Gani A.
- Chinmay A. Patwardhan
- M. Tsaqif Wismadi
- Tore Gude

Norwegian University of Science and Technology

Outline

1) Introduction
2) Use cases
3) Keypoints
4) Example of codes
5) Pros & Cons
6) When to use
7) Summary

NTNU

<u>Seasonal</u> <u>Auto-Regressive</u> <u>Integrated</u> <u>Moving Average</u>

Lets break it down

Auto Regressive (AR) Model

Regression based on past data

Helpful to look at PACF chart (Partial Auto Correlation Function)

Moving Average (MA) Model

Based on the error from past predictions

Helpful to look at ACF chart (Auto Correlation Function)

ARMA Model

Based on the regression on past data AND error from past predictions

Helpful to look at PACF and ACF charts

AR-Integrated-MA (ARIMA) Model

Used when the time series is NOT stationary.

“Integrated” refers to taking the **difference** between values at consecutive time steps

Seasonal ARIMA (SARIMA) Model

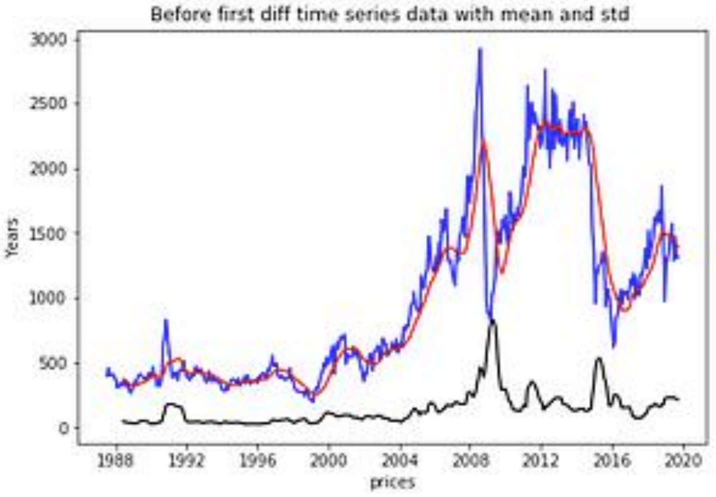
Seasonality refers to a recurring pattern occurring in a duration of time

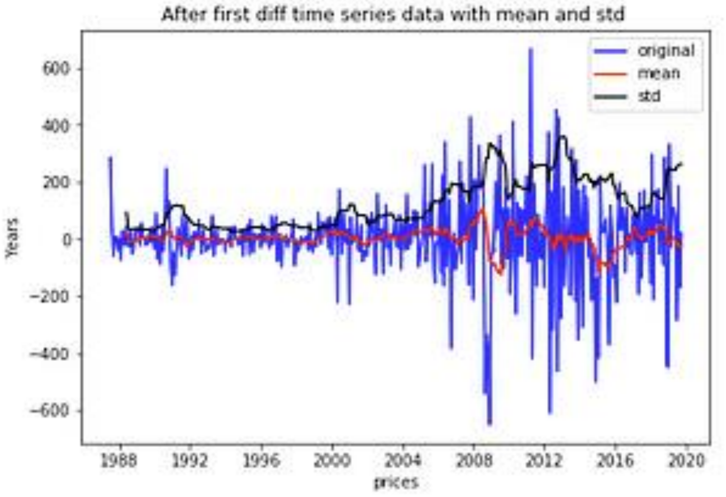
Seasonal Auto-Regressive Integrated Moving Average

- **Auto Regressive (AR)** : the autoregressive part of the **S**easonal component
  - Autocorrelation function (ACF) : correlation between the current and the past values of same variables
  - Partial Autocorrelation (PACF) : direct correlation between past values and current values

- **Integrated** : differencing that has to be applied in order to make the data stationary.
  - (*Dickey-fuller test* : way to check / test stationary)

https://medium.com/@cmukesh8688/why-is-augmented-dickey-fuller-test-adf-test-so-important-in-time-series-analysis-6fc97c6be2f0





Definition

- **Moving Average (MA)** : captures seasonally repeated error patterns: uses past forecast errors rather than past values to forecast future values (in a regression-like model)

Simply,

- ARIMA : AR-Integrated-MA
- SARIMA : *Seasonal* -ARIMA

$$ARIMA \quad (p, d, q) \quad (P, D, Q)_m$$

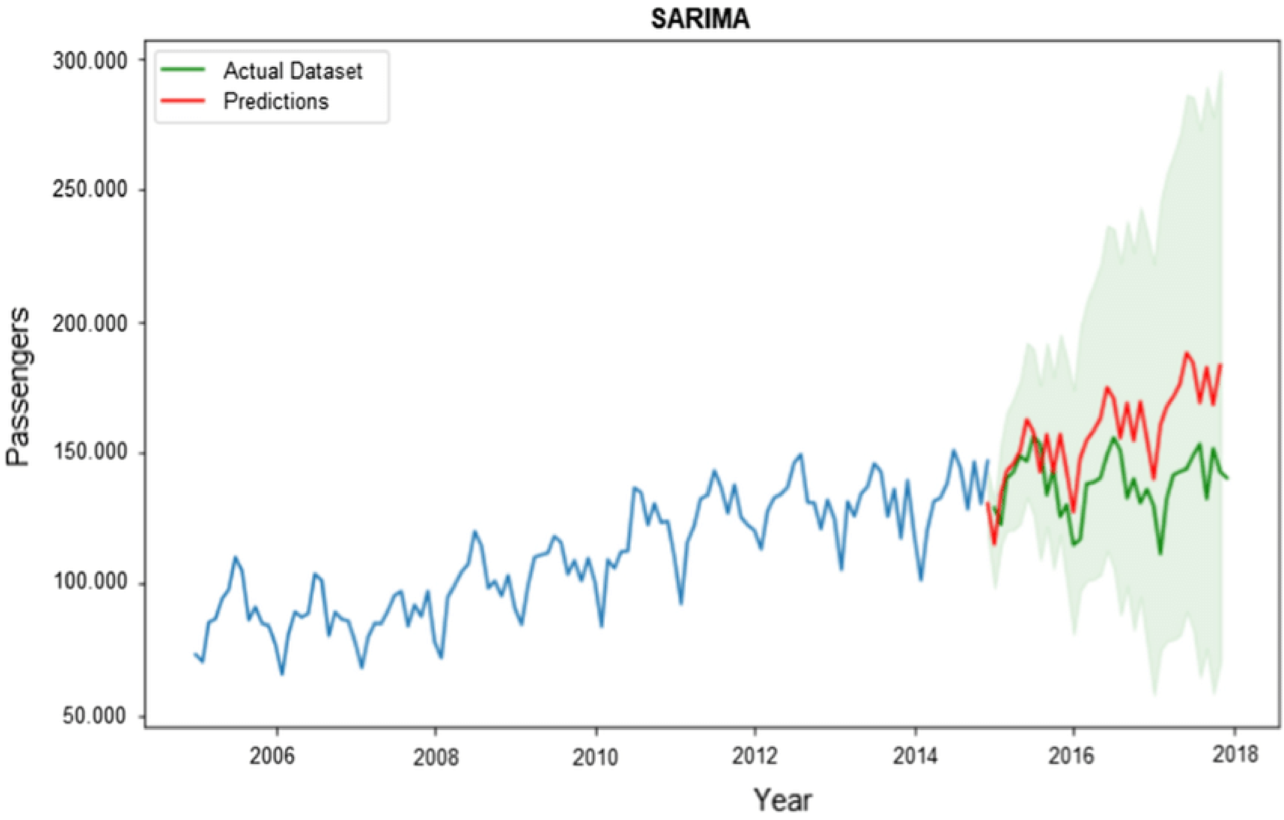
Non-seasonal part of model

Seasonal part of model

Certain numbers as abbreviation:
p : non-seasonal AR terms.
P : seasonal AR terms.
d : differencing steps needed to make the data stationary.
D : seasonal differencing steps.
q : non-seasonal MA terms.
Q : seasonal MA terms.
m : time steps in a seasonal period (e.g., 12 for monthly data).

Use cases - SARIMA

- ***Stock market predictions*** – especially for stocks with seasonal trends.
- ***Sales forecasting*** – particularly in industries with strong seasonal cycles, such as retail and tourism.
- ***Macroeconomic indicators*** – predicting inflation rates or GDP (as usual condition / *no force majeurs*), which often show seasonal variation.
- **Tourism Growth** – predicting growth of tourist number considering seasons



https://link.springer.com/article/10.1007/s00521-021-06232-y

Key points

- SARIMA : an **extension of ARIMA** that adds seasonal components to capture recurring patterns in data.
- Useful **for time series with seasonal cycles** and is widely used in the financial sector for forecasting.
- Quite powerful tool for forecasting time series data that shows periodic behavior, making it highly relevant in financial, retail, and economic applications

Example: Using SARIMA to predict Ford Truck Sales

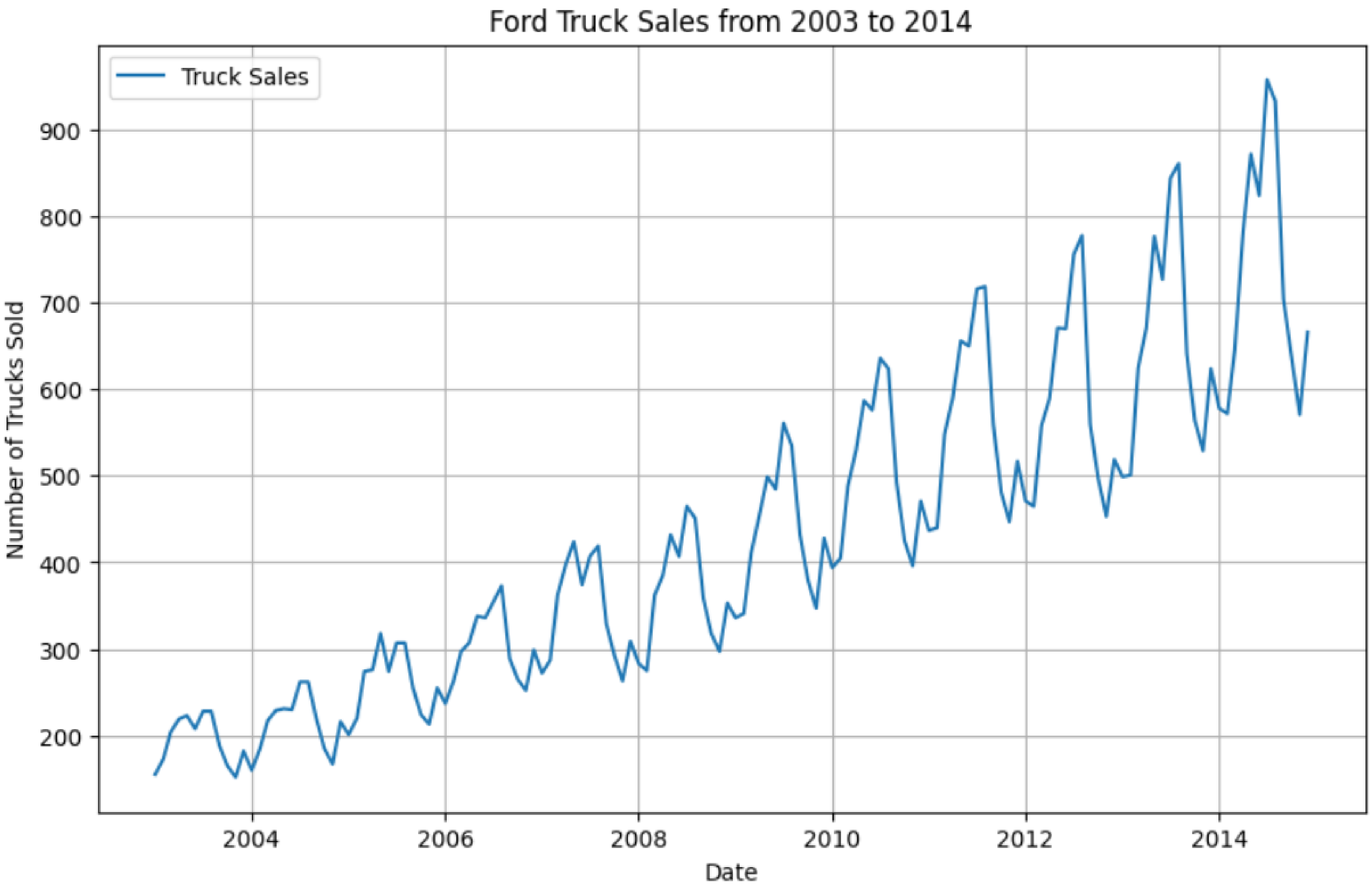
Month-Year	Number_Trucks_Sold
2003-01-01	155
2003-02-01	173
2003-03-01	204
2003-04-01	219
2003-05-01	223

- The dataset includes monthly truck sales data.
- The time period spans from 2003 to 2014.
- The goal is to forecast truck sales for the year 2015.

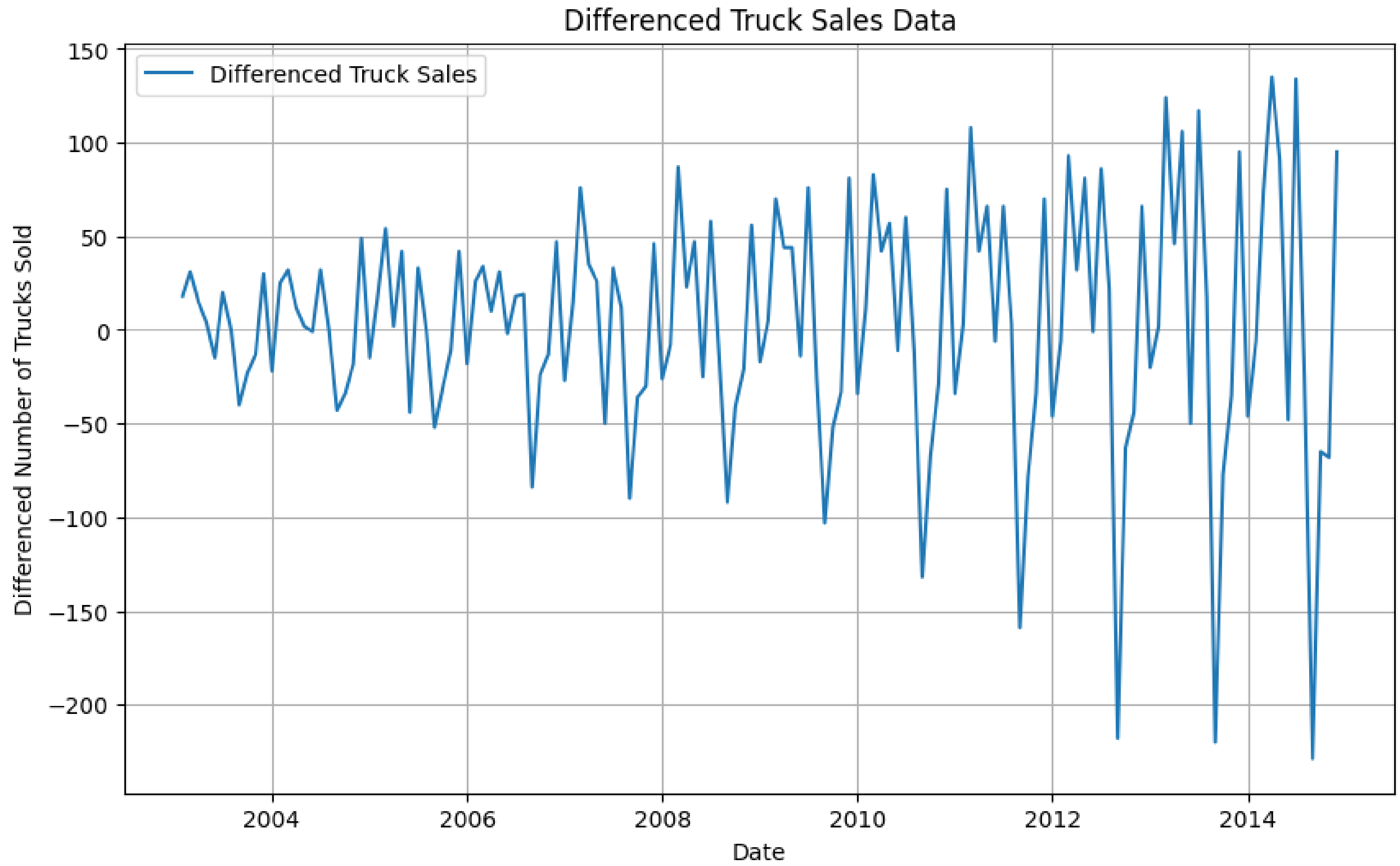
Step-by-step

1. Visualize and check for trend and seasonality
2. SARIMA requires stationary data
   - Check stationary with ADF
   - If dataset is not stationary -> perform differencing
3. Plot ACF and PACF to identify MA (q) and AR (p) parameter
4. Split training-test and fit the SARIMA model
5. Forecast and validate the model
6. Plot the observed VS forecasted values
7. Forecast future values

Visualize and check for trend and seasonality



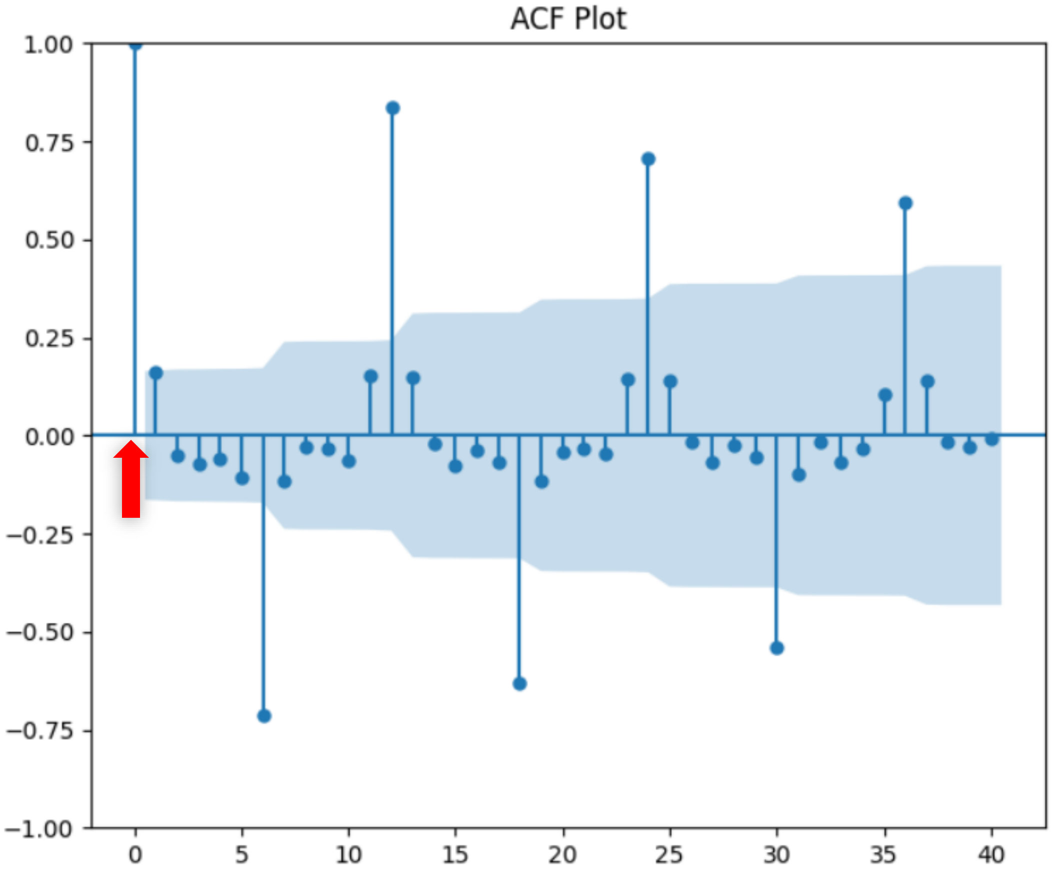
Check stationarity and perform differencing

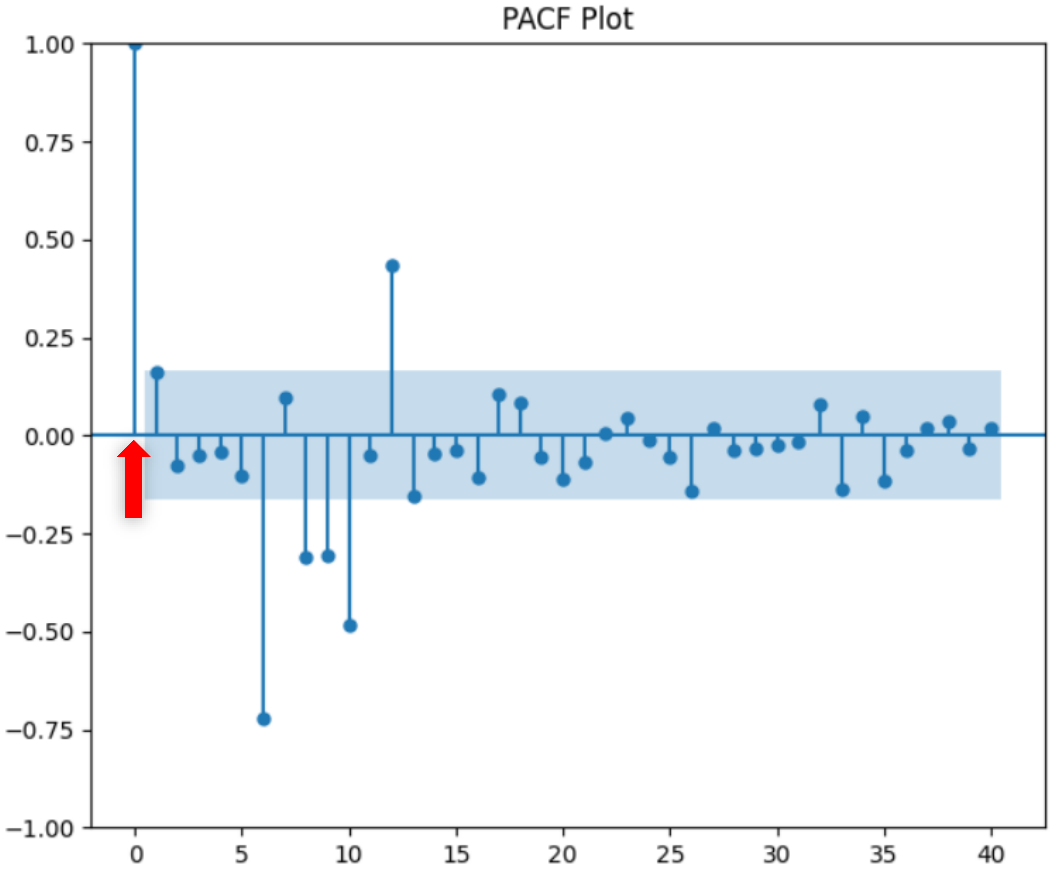



```
After one differencing, p-value: 0.10573354923819939
```

ADF Test: The Augmented Dickey-Fuller (ADF) test checks for stationarity (i.e., whether the time series has a constant mean and variance over time). If the p-value is too high (> 0.15) then we need to perform differencing to make the data stationary.

Plot ACF and PACF to identify MA(q) and AR(p)





How to Interpret the Plots: Look at the ACF plot for significant spikes at certain lags. The spikes drop off quickly after lag 1, it suggests an MA(1) model. Similarly, in the PACF plot, there are significant spikes at lag 1, it suggests an AR(1) model.

Split training-test and fit the SARIMA model

5. Split the training and test dataset

```
[362] train_size = int(len(truck_sales_data) * 0.90)
      train_data = truck_sales_data.iloc[:train_size]
      test_data = truck_sales_data.iloc[train_size:]
```

6. Fit the SARIMA model using the parameter that we got from ACF and PACF analysis

```
[363] sarima_model = SARIMAX(train_data['Number_Trucks_Sold'],
                            order=(1, 1, 1),  # Non-seasonal (p, d, q)
                            seasonal_order=(1, 1, 1, 12),  # Seasonal (P, D, Q, s)
                            enforce_stationarity=False,
                            enforce_invertibility=False)
      sarima_fit = sarima_model.fit(disp=False)
```

Forecast and validate model

```
[364] # Forecast for the test period
      forecast_test = sarima_fit.get_forecast(steps=len(test_data))
      forecast_test_values = forecast_test.predicted_mean

      # Calculate accuracy metrics: MAE and RMSE
      mae = mean_absolute_error(test_data['Number_Trucks_Sold'], forecast_test_values)
      rmse = np.sqrt(mean_squared_error(test_data['Number_Trucks_Sold'], forecast_test_values))

      print(f"Mean Absolute Error (MAE): {mae}")
      print(f"Root Mean Squared Error (RMSE): {rmse}")
```

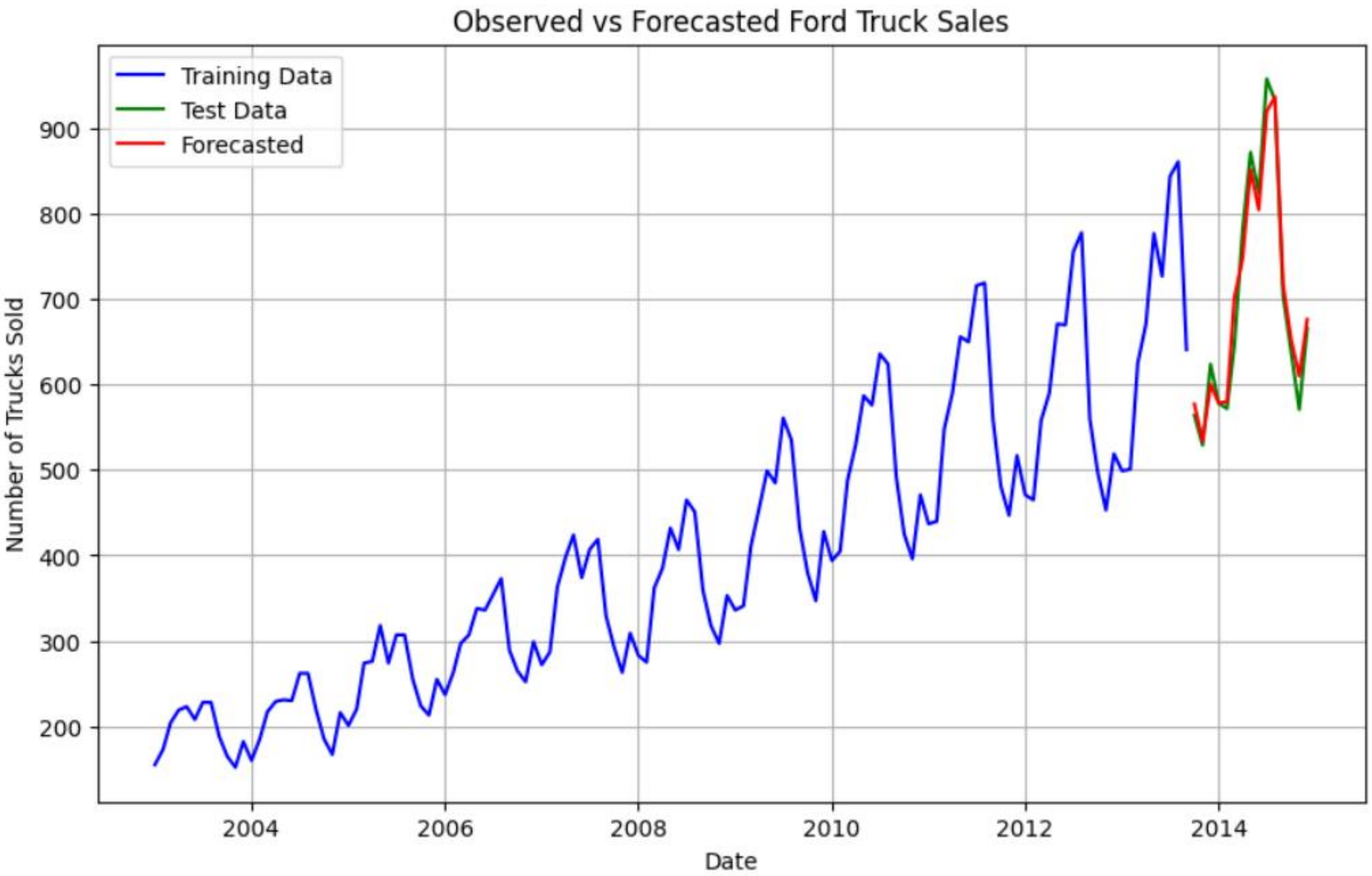
```
Mean Absolute Error (MAE): 19.95885301908104
Root Mean Squared Error (RMSE): 25.045545500431555
```

The model evaluation yielded the following results on the test set:

- **Mean Absolute Error (MAE)**: 19.95
- **Root Mean Squared Error (RMSE)**: 25.04

These metrics suggest that, on average, the model's predictions are off by around 20 units of truck sales. The RMSE, which gives more weight to larger errors, indicates an error magnitude of approximately 25 units.

Plotting observed VS forecasted values



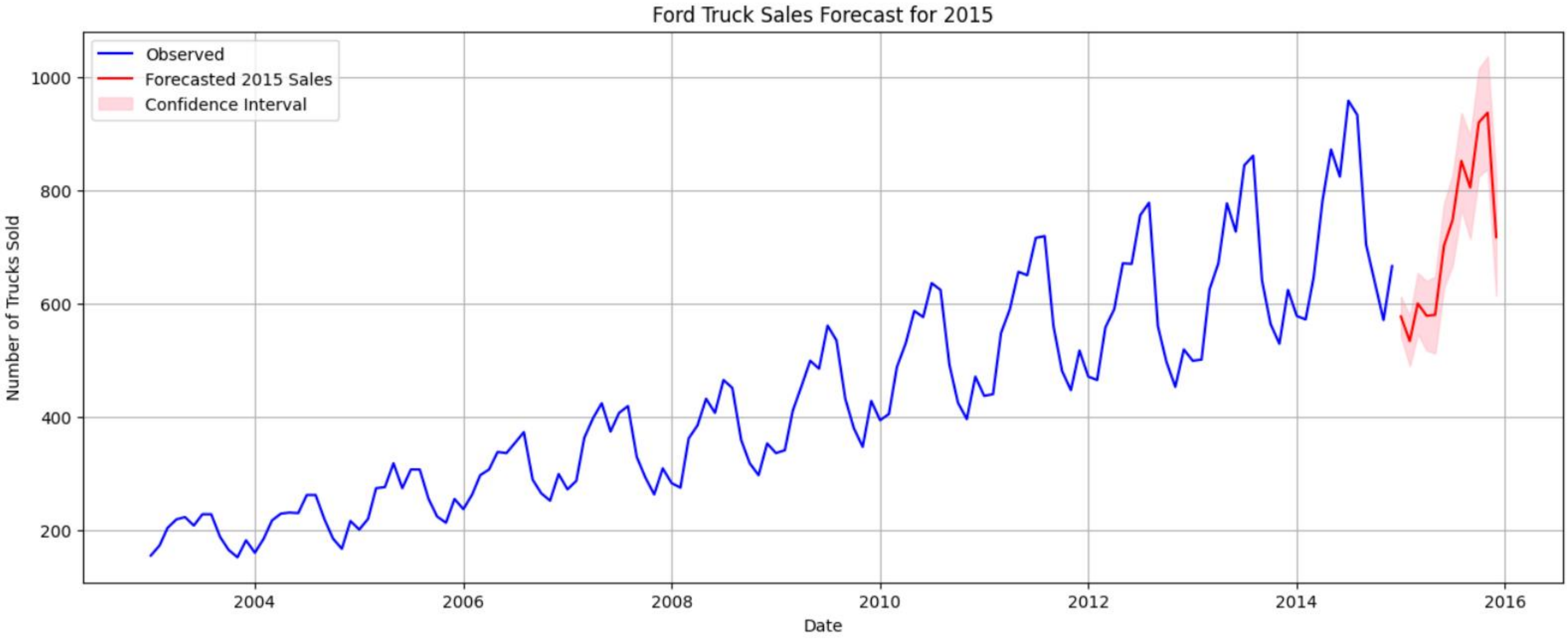
Forecasting future sales in 2015

```python
# Forecast future values for 2015 (next 12 months)
forecast_future = sarima_fit.get_forecast(steps=12)

# Extract predicted mean and confidence intervals
forecast_future_values = forecast_future.predicted_mean
forecast_conf_int = forecast_future.conf_int()

# Generate index for 2015 (12 months ahead)
forecast_index = pd.date_range(start='2015-01-01', periods=12, freq='MS')

# Correct the column names for confidence intervals
# The first column is the lower bound, and the second is the upper bound.
forecast_conf_int.columns = ['Lower_Bound', 'Upper_Bound']

# Create a table of forecasted values for 2015
forecast_2015_df = pd.DataFrame({
    'Month': forecast_index.strftime('%Y-%m'),
    'Predicted_Sales': forecast_future_values,
    'Lower_Bound': forecast_conf_int['Lower_Bound'],
    'Upper_Bound': forecast_conf_int['Upper_Bound']
})
```

Truck Sales for 2015:

Month	Predicted_Sales	Lower_Bound	Upper_Bound
2015-01	577.219725	542.360787	612.078664
2015-02	533.773411	489.830942	577.715880
2015-03	600.088193	546.330978	653.845408
2015-04	578.488289	517.513094	639.463483
2015-05	580.037127	512.119030	647.955223
2015-06	702.147982	628.169968	776.125996
2015-07	747.315724	667.624230	827.007218
2015-08	851.696769	766.730378	936.663160
2015-09	804.733295	714.773652	894.692938
2015-10	919.723804	825.047298	1014.400311
2015-11	936.961225	837.785285	1036.137166
2015-12	717.312945	613.836349	820.789541

Forecasting future sales in 2015



Pros

- Explicity models and incorporates seasonal patterns
- Deals with seasonal and non-seasonal stationarity
- Improved forcasting accuracy for seasonal data
- Well suited for long-term forcasting

Cons

- Complexity: Higher number of parameters to estimate
- Requires long and detailed time series data
- Risk of overfitting seasonal pattern
- Too complex for straightforward forecasting

When to use: ARIMA vs SARIMA

- ARIMA
  - No seasonality or very weak seasonal patterns
  - Limited data
  - Model interpretability is priority
- SARIMA
  - Strong seasonality
  - Large datset
  - Forecast accuracy is priority

Summary

- Great forecasting for seasonal data