FDSS Lab 3 Report

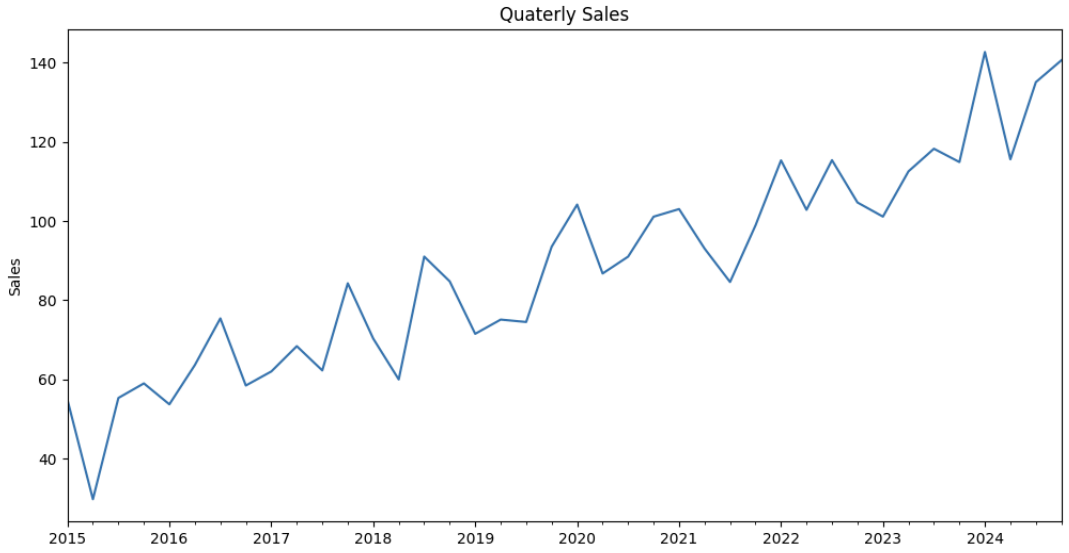
Linear Regression & ARIMA Models Time Series Analysis

Merve Pakcan

1. Linear Regression Model

In this assignment, I generated a synthetic time series using NumPy to build a Linear Regression Model using Python. To make the results reproducible, I set the random seed based on my date of birth (np.random.seed(2704)). The main goal of this part is to see how a simple linear model performs on data with a clear trend, and to evaluate the fitted values and residuals. This helps understand whether a linear approach is suitable for forecasting the generated series.

1.1 - Quarterly Sales Plot



From the plot, it's clear that the generated quarterly sales data shows an overall upward trend over time. Even though the series fluctuates from quarter to quarter due to the added randomness, the general direction is increasing. This suggests that the underlying pattern in the data is mainly driven by a positive linear trend rather than seasonal effects or other complex structures.

OLS Regression Results

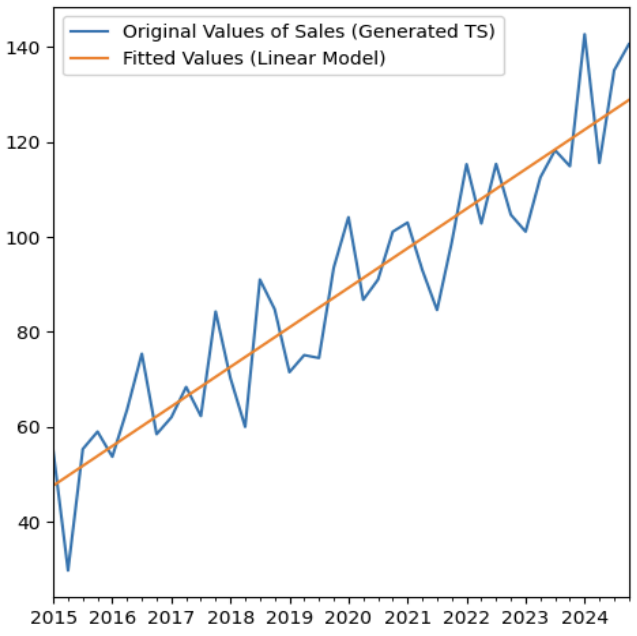
Dep. Variable:	sales	R-squared:	0.867
Model:	OLS	Adj. R-squared:	0.863
Method:	Least Squares	F-statistic:	246.9
Date:	Sat, 15 Nov 2025	Prob (F-statistic):	3.28e-18
Time:	22:07:16	Log-Likelihood:	-146.52
No. Observations:	40	AIC:	297.0
Df Residuals:	38	BIC:	300.4
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	45.5236	3.118	14.600	0.000	39.212	51.836
t	2.0823	0.133	15.712	0.000	1.814	2.351

Omnibus:	0.637	Durbin-Watson:	2.195
Prob(Omnibus):	0.727	Jarque-Bera (JB):	0.688
Skew:	0.053	Prob(JB):	0.709
Kurtosis:	2.367	Cond. No.	48.0

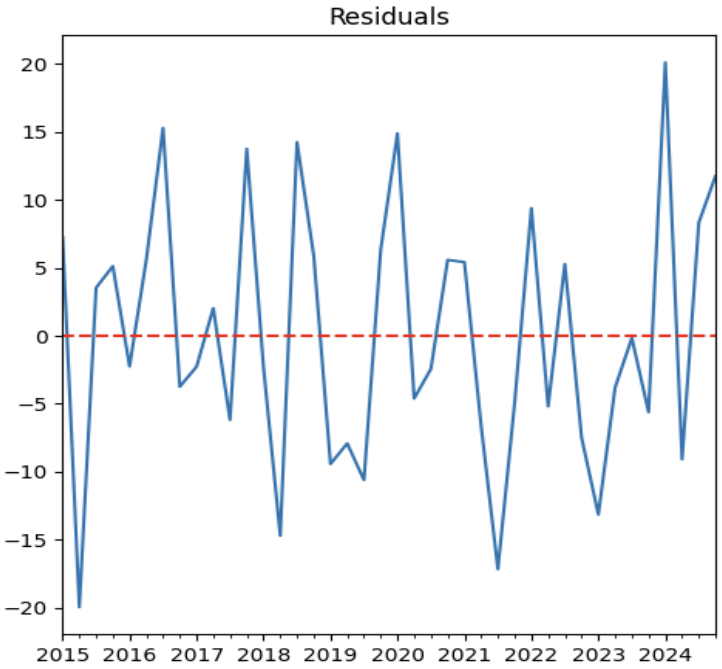
The regression summary shows that the model fits the data well, with an R-squared of 0.867. The time variable is highly significant ($p < 0.001$), which confirms the strong positive trend observed in the series. The Durbin-Watson value is close to 2, indicating no major autocorrelation issues in the residuals.

1.2 - Original vs Fitted Plot



The fitted line captures the overall upward trend in the original series quite well. Although the actual values fluctuate around the line due to the random variation in the generated data, the model still follows the main direction of the trend. This indicates that a simple linear regression is able to represent the underlying pattern in this synthetic time series.

1.3 - Residual Plot



The residuals appear to be scattered randomly around zero without showing any obvious pattern or structure. This suggests that the linear model fits the data reasonably well, and there is no clear indication of model misspecification. The randomness of the residuals supports the idea that the main trend has been successfully captured by the regression model.

```
t_forecast = np.arange(41,49)
x_forecast = sm.add_constant(t_forecast)
forecast = model.predict(x_forecast)
forecast_index=pd.period_range(start="2016Q1", periods=8, freq="Q")
df_forecast = pd.DataFrame({"forecast":forecast}, index=forecast_index)
print(df_forecast)
```

```
         forecast
2016Q1  130.899903
2016Q2  132.982252
2016Q3  135.064602
2016Q4  137.146951
2017Q1  139.229300
2017Q2  141.311650
2017Q3  143.393999
2017Q4  145.476348
```

The forecast values show a steady increase, which matches the positive trend in the generated data. While the model doesn't capture fluctuations, it gives a reasonable simple estimate for upcoming quarters.

2. ARIMA Model

In this part of the assignment, I used the U.S. Airline Traffic dataset from Kaggle(https://www.kaggle.com/datasets/yyxian/u-s-airline-traffic-data) to build a seasonal ARIMA model. This dataset includes monthly passenger counts from 2003 to 2023, which makes it suitable for analyzing trend and seasonality. The goal here is to transform the data, fit the best SARIMA model using auto_arima, and generate forecasts to see how well the model captures the real traffic pattern.

2.1 – Monthly Airline Traffic Plot

```
[13] ✓ 5s
!pip install pmdarima
```

```
Requirement already satisfied: pmdarima in /usr/local/lib/python3.12/dist-packages (2.1.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (1.5.2)
Requirement already satisfied: Cython!=0.29.18,!=0.29.31,>=0.29 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (3.0.12)
Requirement already satisfied: numpy>=1.21.6 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (2.0.2)
Requirement already satisfied: pandas>=0.19 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (2.2.2)
Requirement already satisfied: scikit-learn>=0.22 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (1.6.1)
Requirement already satisfied: scipy>=1.13.0 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (1.16.3)
Requirement already satisfied: statsmodels>=0.14.5 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (0.14.5)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (2.5.0)
Requirement already satisfied: setuptools!=50.0.0,>=42 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (75.2.0)
Requirement already satisfied: packaging>=17.1 in /usr/local/lib/python3.12/dist-packages (from pmdarima) (25.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=0.19->pmdarima) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=0.19->pmdarima) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=0.19->pmdarima) (2025.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn>=0.22->pmdarima) (3.6.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dist-packages (from statsmodels>=0.14.5->pmdarima) (1.0.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas>=0.19->pmdarima) (1.17.0)
```

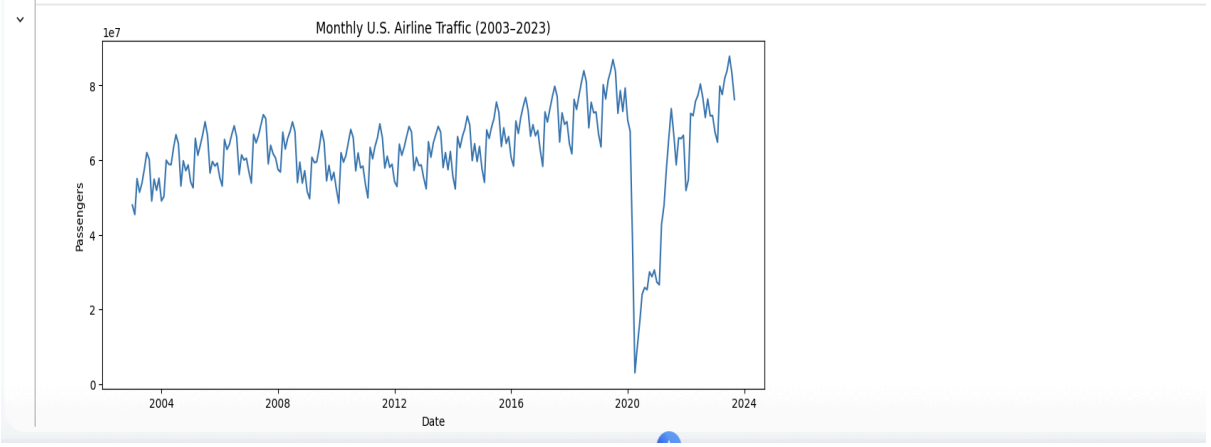
```
[14] ✓ 0s
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pmdarima as pm
```

```
[15] ✓ 0s
df = pd.read_csv("air traffic.csv")

df["Date"] = pd.to_datetime(df["Year"].astype(str) + "-" + df["Month"].astype(str))
df = df.set_index("Date")

s = df["Pax"].str.replace(",", "").astype(int)
```

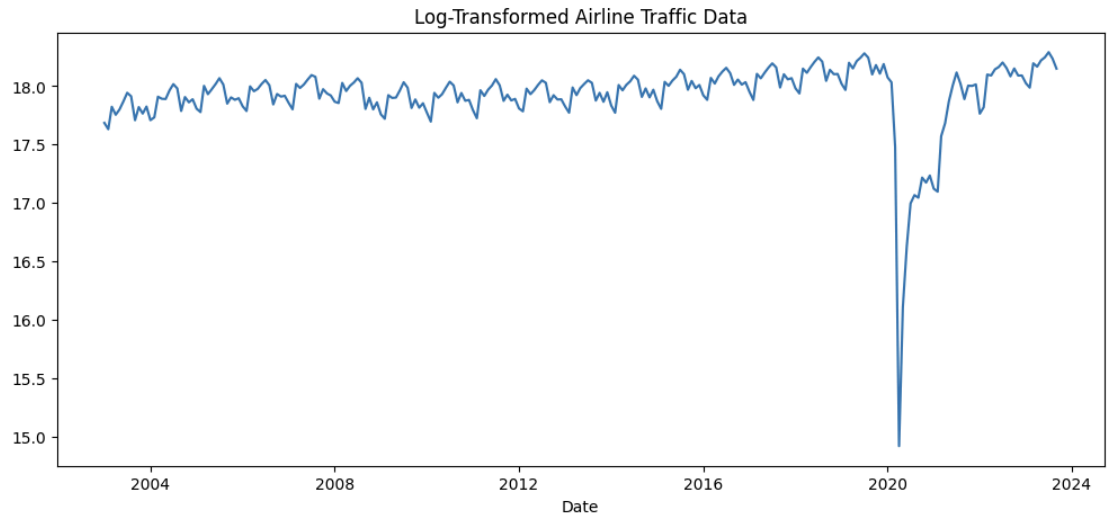
```
[16] ✓ 0s
plt.figure(figsize=(12,5))
plt.plot(s)
plt.title("Monthly U.S. Airline Traffic (2003–2023)")
plt.ylabel("Passengers")
plt.xlabel("Date")
plt.show()
```



In this step, the dataset was loaded from Kaggle and the date-related columns were converted into a proper datetime index. After cleaning the passenger counts and organizing the series, the monthly airline traffic data was plotted to observe its overall behavior. The plot shows a clear upward trend and strong seasonality throughout the years, along with a

sharp drop in early 2020 due to the impact of COVID-19. Apart from this temporary shock, the series follows a stable seasonal pattern, making it suitable for modeling with SARIMA.

2.2 - Log-Transformed Series Plot



The log-transformed series shows more stable variance across time, making the seasonal pattern easier to observe. The long-term upward trend remains visible, while the sharp drop in 2020 appears even clearer. Overall, the log transform stabilizes the series and prepares it better for SARIMA modeling.

2.3 - Auto-ARIMA Search & Model Summary

```
# AUTO ARIMA (SARIMA)
model = pm.auto_arima(
    log_s,
    start_p=1, start_q=1,
    max_p=3, max_q=3,
    m=12,  # monthly seasonality
    start_P=0,
    seasonal=True,
    test='adf',
    d=None,
    D=None,
    trace=True,
    error_action='ignore',
    suppress_warnings=True,
    stepwise=True
)
```

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,1)[12] intercept   : AIC=-86.039, Time=2.61 sec
 ARIMA(0,1,0)(0,0,0)[12] intercept   : AIC=-62.592, Time=0.20 sec
 ARIMA(1,1,0)(1,0,0)[12] intercept   : AIC=-66.717, Time=0.59 sec
 ARIMA(0,1,1)(0,0,1)[12] intercept   : AIC=-66.590, Time=0.98 sec
 ARIMA(0,1,0)(0,0,0)[12]             : AIC=-64.572, Time=0.07 sec
 ARIMA(1,1,1)(0,0,0)[12] intercept   : AIC=-80.942, Time=0.80 sec
 ARIMA(1,1,1)(1,0,1)[12] intercept   : AIC=-95.117, Time=5.00 sec
 ARIMA(1,1,1)(1,0,0)[12] intercept   : AIC=-87.090, Time=1.11 sec
 ARIMA(1,1,1)(2,0,1)[12] intercept   : AIC=-89.279, Time=2.68 sec
 ARIMA(1,1,1)(1,0,2)[12] intercept   : AIC=-95.746, Time=4.48 sec
 ARIMA(1,1,1)(0,0,2)[12] intercept   : AIC=-86.715, Time=3.43 sec
 ARIMA(1,1,1)(2,0,2)[12] intercept   : AIC=inf, Time=2.52 sec
 ARIMA(0,1,1)(1,0,2)[12] intercept   : AIC=inf, Time=4.65 sec
 ARIMA(1,1,0)(1,0,2)[12] intercept   : AIC=inf, Time=2.71 sec
 ARIMA(2,1,1)(1,0,2)[12] intercept   : AIC=-96.877, Time=3.99 sec
 ARIMA(2,1,1)(0,0,2)[12] intercept   : AIC=-85.615, Time=3.84 sec
 ARIMA(2,1,1)(1,0,1)[12] intercept   : AIC=inf, Time=3.23 sec
 ARIMA(2,1,1)(2,0,2)[12] intercept   : AIC=inf, Time=4.55 sec
 ARIMA(2,1,1)(0,0,1)[12] intercept   : AIC=inf, Time=1.46 sec
 ARIMA(2,1,1)(2,0,1)[12] intercept   : AIC=-92.484, Time=4.13 sec
 ARIMA(2,1,0)(1,0,2)[12] intercept   : AIC=inf, Time=4.69 sec
 ARIMA(3,1,1)(1,0,2)[12] intercept   : AIC=-95.010, Time=4.66 sec
 ARIMA(2,1,2)(1,0,2)[12] intercept   : AIC=-96.194, Time=6.65 sec
 ARIMA(1,1,2)(1,0,2)[12] intercept   : AIC=-94.162, Time=3.99 sec
 ARIMA(3,1,0)(1,0,2)[12] intercept   : AIC=inf, Time=3.92 sec
 ARIMA(3,1,2)(1,0,2)[12] intercept   : AIC=-95.823, Time=6.66 sec
 ARIMA(2,1,1)(1,0,2)[12]             : AIC=inf, Time=4.41 sec

Best model:  ARIMA(2,1,1)(1,0,2)[12] intercept
Total fit time: 88.084 seconds
```

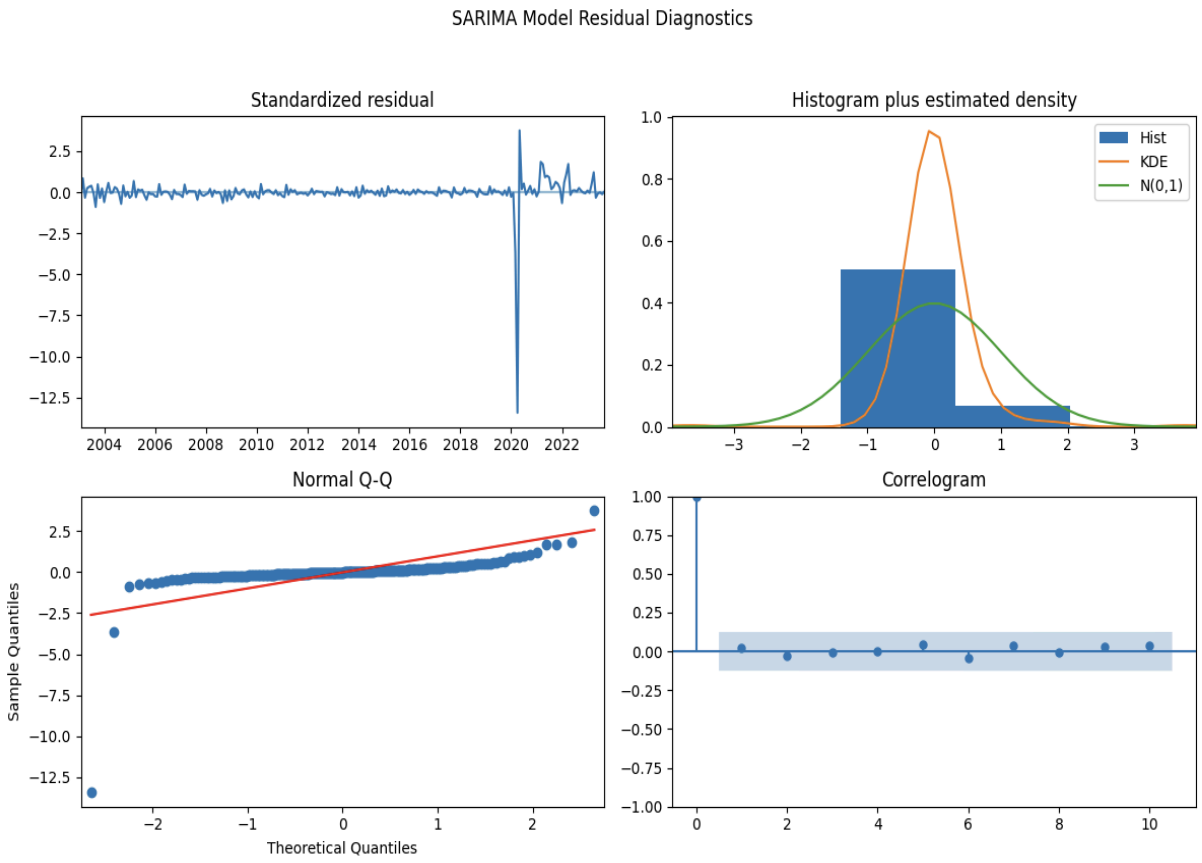
```
print(model.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                   y   No. Observations:                  249
Model:             SARIMAX(2, 1, 1)x(1, 0, [1, 2], 12)   Log Likelihood                56.438
Date:                                 Sat, 15 Nov 2025   AIC                            -96.877
Time:                                         23:48:23   BIC                            -68.770
Sample:                                     01-01-2003   HQIC                           -85.562
                                          - 09-01-2023
Covariance Type:                                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.0001      0.001      0.164      0.869      -0.001       0.001
ar.L1          0.7508      0.185      4.055      0.000       0.388       1.114
ar.L2         -0.0863      0.172     -0.502      0.616      -0.423       0.251
ma.L1         -0.8633      0.187     -4.614      0.000      -1.230      -0.497
ar.S.L12       0.9572      0.068     14.004      0.000       0.823       1.091
ma.S.L12      -0.8963      0.098     -9.139      0.000      -1.089      -0.704
ma.S.L24       0.0555      0.069      0.803      0.422      -0.080       0.191
sigma2         0.0380      0.001     34.381      0.000       0.036       0.040
==========================================================================================
Ljung-Box (L1) (Q):                   0.12   Jarque-Bera (JB):            214197.81
Prob(Q):                              0.73   Prob(JB):                         0.00
Heteroskedasticity (H):              28.98   Skew:                            -10.33
Prob(H) (two-sided):                  0.00   Kurtosis:                        145.49
==========================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Auto-ARIMA selected **SARIMA(2,1,1)(1,0,2)[12]** as the best model according to AIC. Most coefficients are significant, especially the seasonal terms, confirming the strong yearly seasonality in the dataset. Despite the COVID-19 shock, the model captures the main trend and seasonal structure well. The selected model is therefore appropriate for forecasting.

2.4 - Plot Diagnostics



The diagnostic plots show that the residuals mostly behave like white noise. In the standardized residual plot, values stay around zero except for the sharp drop in 2020, which reflects the pandemic shock. The histogram and Q-Q plot suggest some deviation from normality, again mainly due to this outlier. In the correlogram, almost all autocorrelations fall within the confidence bands, indicating that the SARIMA model has captured the main structure of the series. Overall, the residuals do not show any major pattern, so the model appears to be a reasonable fit.

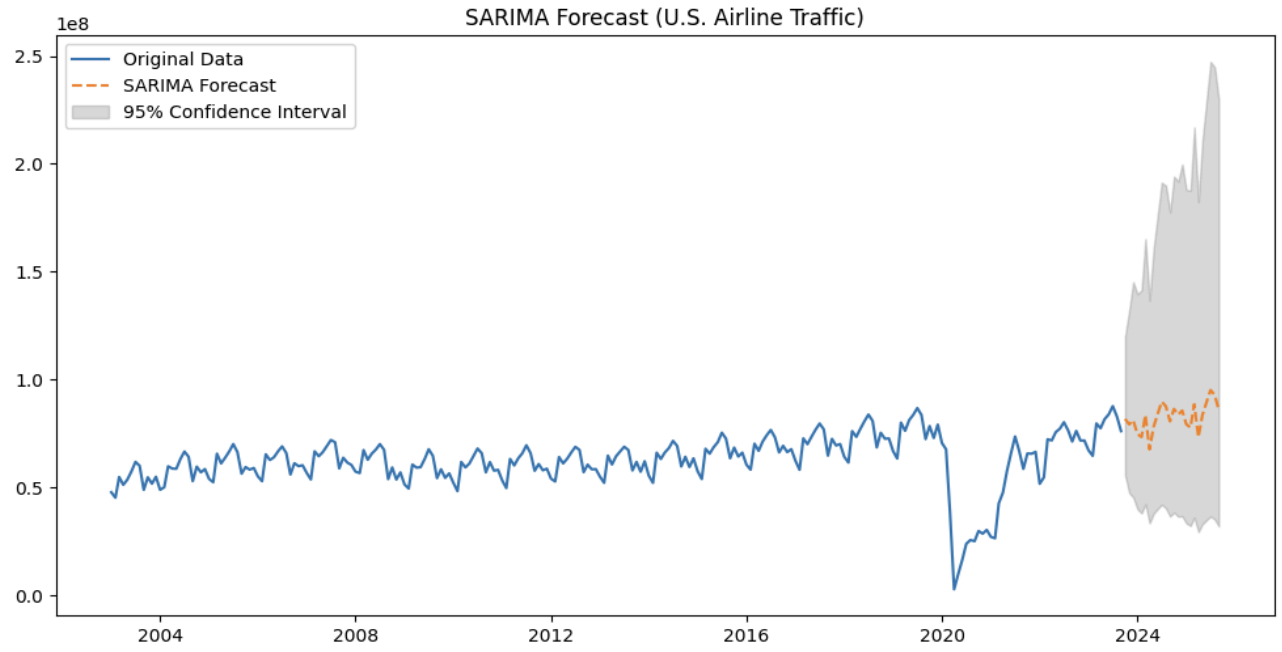
2.5 - Forecasting

```
# Forecast
n_forecast = 24
forecast, conf_int = model.predict(
    n_periods=n_forecast,
    return_conf_int=True,
    alpha=0.05
)

forecast_index = pd.date_range(
    start=s.index[-1] + pd.DateOffset(months=1),
    periods=n_forecast,
    freq="MS"
)
```

```
# Transform back from log scale
forecast_orig = np.exp(forecast)
conf_int_orig = np.exp(conf_int)
```

```
# Fopecast Plot
plt.figure(figsize=(12,6))
plt.plot(s, label="Original Data")
plt.plot(forecast_index, forecast_orig, label="SARIMA Forecast", linestyle="--")
plt.fill_between(
    forecast_index,
    conf_int_orig[:, 0],
    conf_int_orig[:, 1],
    color="gray",
    alpha=0.3,
    label="95% Confidence Interval"
)
plt.title("SARIMA Forecast (U.S. Airline Traffic)")
plt.legend()
plt.show()
```



The SARIMA model provides a reasonable 24-month forecast, continuing the upward trend that was present before the pandemic drop. The predicted values follow a stable seasonal pattern, and the confidence interval widens over time as expected. Although the model does not fully capture the sharp decline in 2020, the forecast quickly aligns back with the long-term trend. Overall, the results are consistent with the historical behavior of the series and offer a realistic outlook for future passenger volumes.