### **From:** Helena Mabey, Isuri Rajapaksa, Lauren Thelen

**Subject:** I-70 Congestion Case Study Analysis

**Date:** March 9, 2025

Traffic on I-70 through the Rocky Mountains of Colorado plays a critical role in business, tourism, and the overall well-being for Colorado residents. Given its impact, it is essential for the Colorado Department of Transportation and the federal government to conduct regular analyses to review traffic flow and congestion as well as independent factors that potentially impact traffic flow. These insights enable proactive measures to attempt to minimize major disruptions and improve overall highway efficiency.

To support this effort, our team of data analysts have conducted both historical traffic analysis as well as developed forecasting for the remainder of 2017. This report provides key findings, analytical insights, and multiple data-driven recommendations to enhance traffic management strategies including traditional methods as well as potential powerful AI solutions.

### **Executive Summary**

**Major Findings:**

Colorado’s traffic patterns are largely shaped by weather, seasonal changes, gas prices, and peak travel times, with rain and snow leading to lower traffic volumes, while warmer temperatures drive increased congestion, particularly in the summer due to tourism. Traffic is heaviest on weekdays, especially during rush hours, and Saturdays also experience significant volume due to leisure and recreational travel. Infrastructure constraints, particularly on major highways, which may be causing bottlenecks during peak hours, highlighting the need for road expansions, HOV lanes, and dynamic toll pricing. Public transit remains underutilized, suggesting the need for expanded park-and-ride options into mountain locations.

**Analytical Overview:**

The analysis of traffic data reveals clear trends and patterns that provide valuable insights for traffic management and planning. Over time, traffic volume has shown a steady increase, indicating overall growth in roadway usage. However, periodic declines suggest that external factors, such as economic conditions, policy changes, or significant events, may influence traffic fluctuations.

Seasonal variations are a key factor in traffic trends. The highest traffic volumes typically occur during the summer months, particularly in June, July, and August, likely due to increased travel during vacations and holiday periods. Conversely, winter months, such as December, January, and February, consistently see lower traffic volumes, which may be attributed to weather conditions or reduced travel activity. These seasonal trends remain stable over the years, though the intensity of traffic can vary from one year to another.

Traffic distribution also indicates that most daily traffic volumes fall within a moderate range, with extreme spikes being less frequent. However, when they do occur, these peaks can place significant strain on infrastructure and highlight the need for effective congestion management strategies. Additionally, occasional anomalies in traffic counts suggest that external disruptions, such as special events, weather incidents, or policy shifts, can cause temporary but notable changes in traffic flow.

Understanding these trends is essential for optimizing resource allocation, infrastructure development, and congestion control. The consistent seasonal patterns allow for proactive planning to accommodate peak periods, while identifying anomalies can help mitigate unexpected traffic disruptions. These insights can support transportation agencies in making data-driven decisions, improving road efficiency, and enhancing commuter experiences. By recognizing long-term growth and seasonal fluctuations, policymakers can implement targeted strategies to manage increasing traffic demands effectively.

**Recommendations:**

CDOT can enhance road safety and traffic efficiency by integrating AI-powered chatbots in navigation apps like Waze and Google Maps. When drivers route through specific highways, AI chatbots can ask curated questions, collecting real-time data to help CDOT identify traffic issues, monitor road conditions, and take proactive action. This data-driven approach enables CDOT to prevent or prepare for hazardous driving conditions and send real-time alerts about road closures, weather disruptions, or alternative routes. Expanding public transportation routes, increasing park-and-ride options, and incentivizing carpooling can further reduce congestion. Additionally, improving drainage, increasing road salting, and optimizing transit services will help manage seasonal traffic and create a safer, more adaptive transportation system.

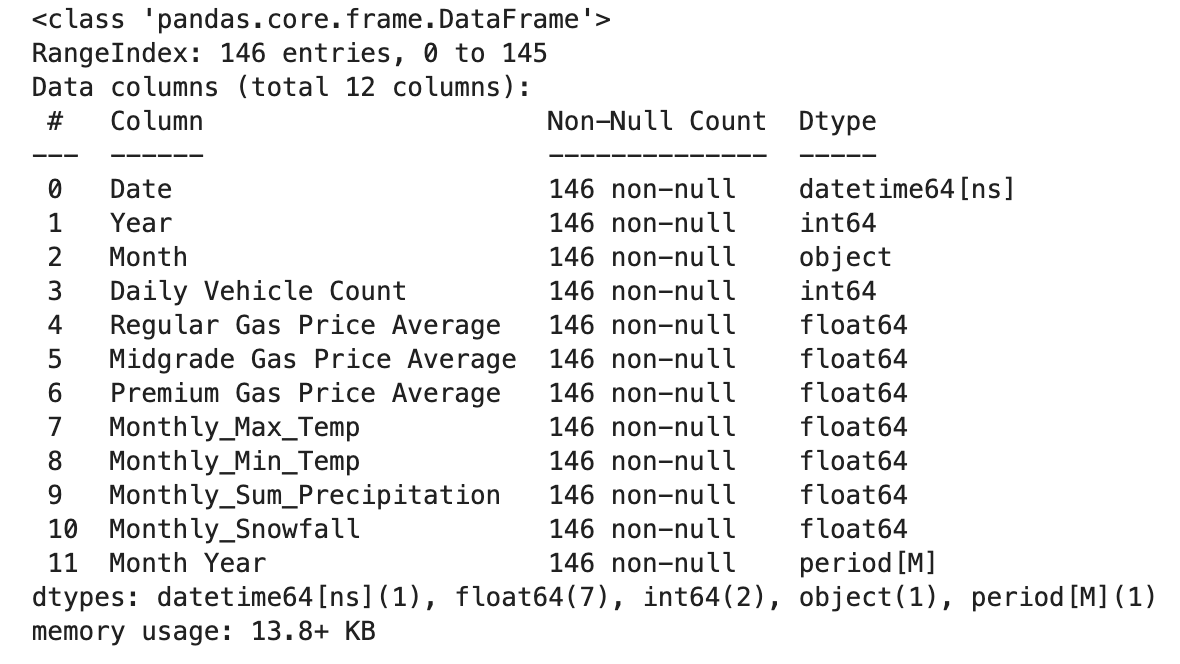
### **Appendix**

**Data Prep, EDA, Regression Modeling, Analysis**

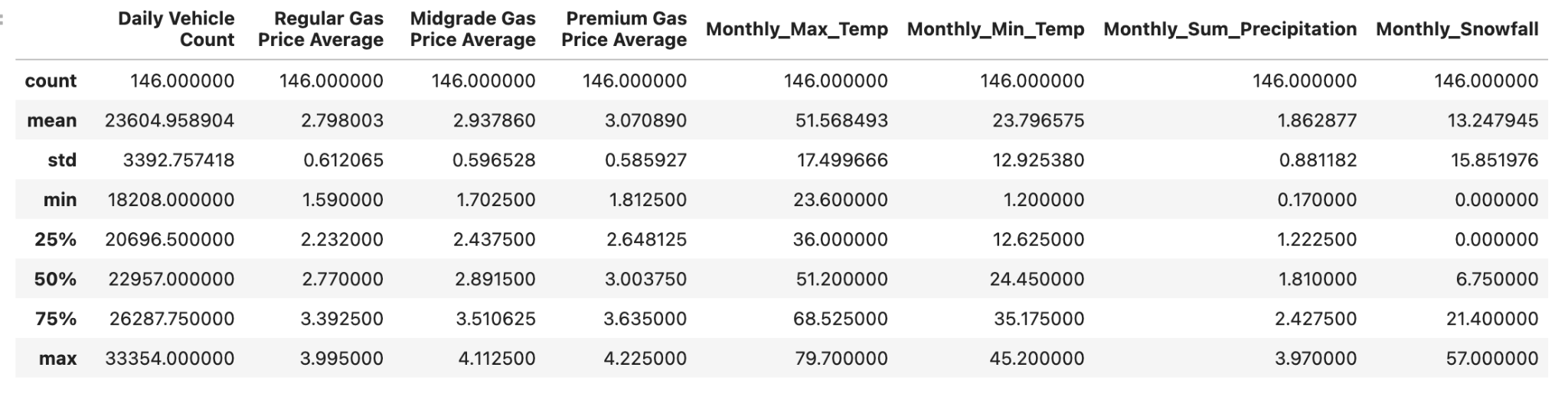
**YouTube Link:** [**https://youtu.be/lkPjRo2zkDo**](https://youtu.be/lkPjRo2zkDo)

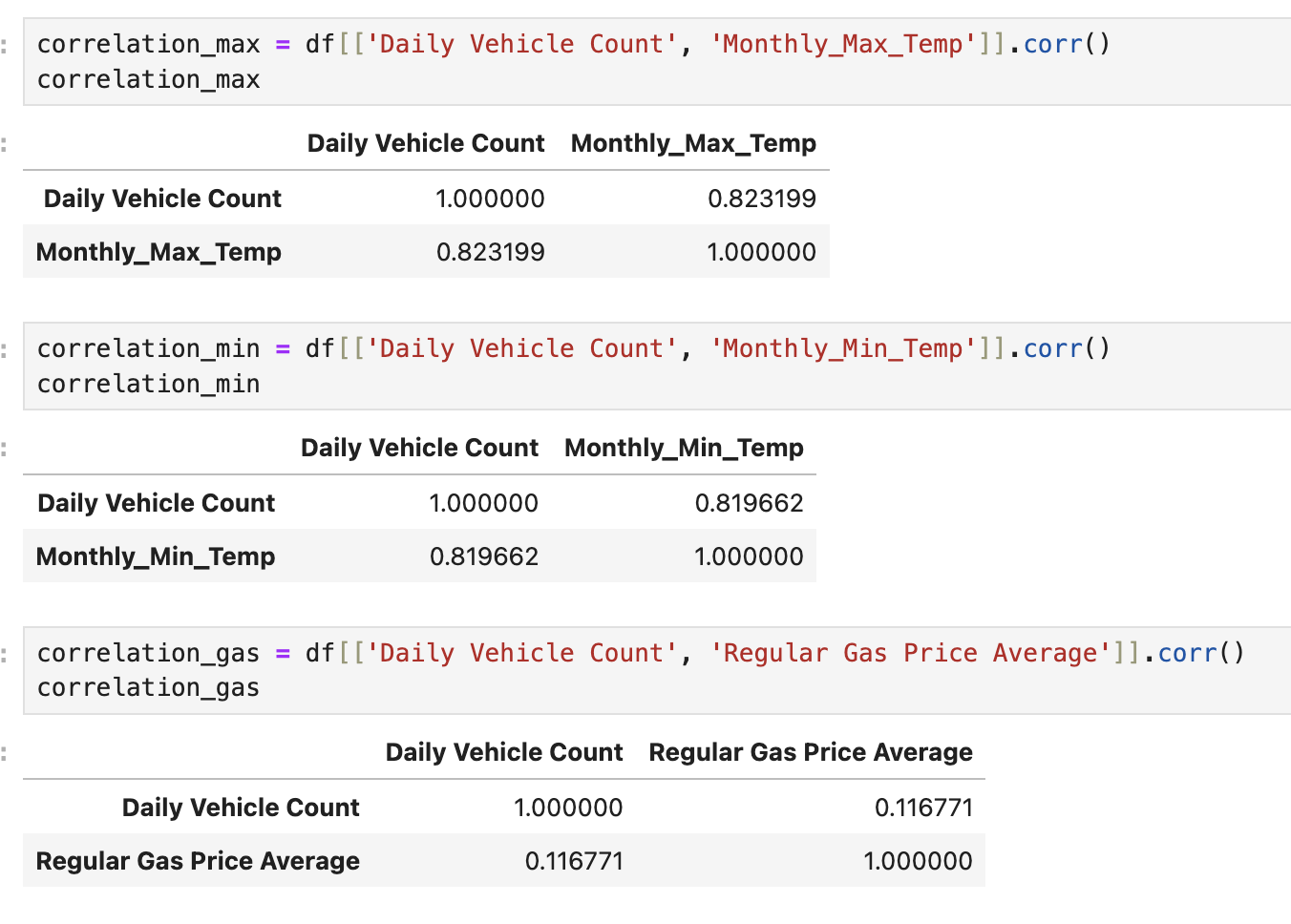
### **1. Data Preparation & Non-Visual Exploration**

* **Reviewed the dataset:** 
  + Found no null values and all months in the data were continuous
  + Reviewed and downloaded maximum temperature, minimum temperature, precipitation, and snowfall from [Colorado Climate Center - Data Access](https://climate.colostate.edu/data_access_new.html).
    - This data is from a weather station in Vail, which is in Eagle county approximately 20 miles east on I-70 of the Wolcott traffic station (between MM 160-161). Vail, being a larger city on I-70, provides a good representation of traffic on I-70 in the general area of Wolcott. There was not a weather station in Wolcott.
    - Reviewed data for missing information. Used mean of values for each month for the missing month temperatures and precipitation.
  + Reviewed and downloaded average weekly gas prices from the [Colorado Information Marketplace](https://data.colorado.gov/Energy/Gasoline-Prices-in-Colorado/8pk9-mh2i/explore/query/SELECT%0A%20%20%60date%60%2C%0A%20%20%60allgradesgasprice%60%2C%0A%20%20%60regulargasprice%60%2C%0A%20%20%60midgradegasprice%60%2C%0A%20%20%60premiumgasprice%60%0AWHERE%0A%20%20%60date%60%0A%20%20%20%20BETWEEN%20%222005-01-01T14%3A11%3A49%22%20%3A%3A%20floating_timestamp%0A%20%20%20%20AND%20%222017-02-28T14%3A11%3A49%22%20%3A%3A%20floating_timestamp/page/aggregate)
    - Calculated monthly averages based on weekly prices for each grade of gas
  + Combined all new data with the original Congestion dataset for ease of ingestion into the notebook. Completed all preliminary data transformation within Excel.
* **Understand key variables:** Once loaded into a notebook, confirmed all data types, reviewed again for null values, and made any transformations needed for further analysis



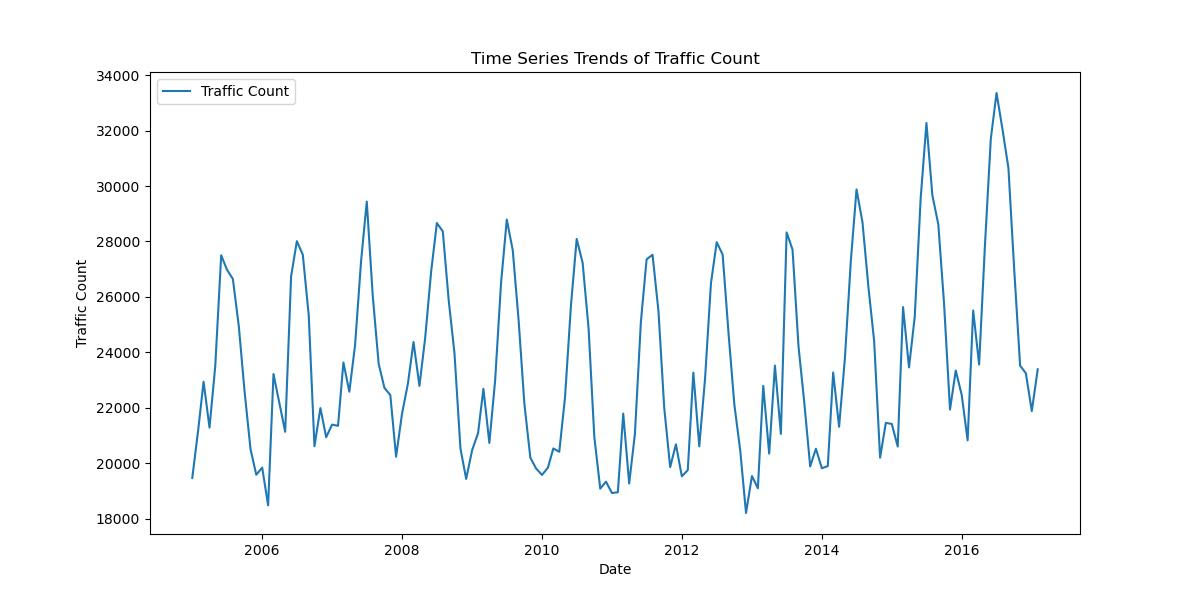
* **Completed Preliminary Non-Visual EDA:** Reviewed all characteristics of each independent variable along with the dependent variable, Daily Vehicle Count. Reviewed skew and kurtosis values of each variable. Reviewed for correlation between Daily Vehicle Count and a sample of the independent variables. Took note of this data for future visual EDA.



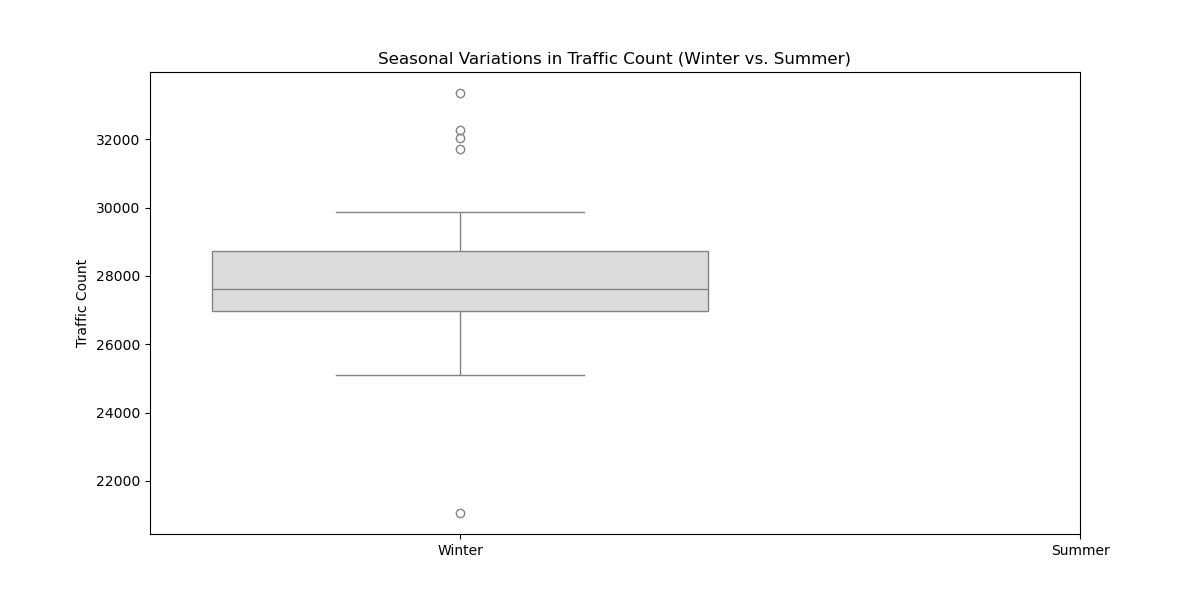


### **2. Descriptive Analysis**

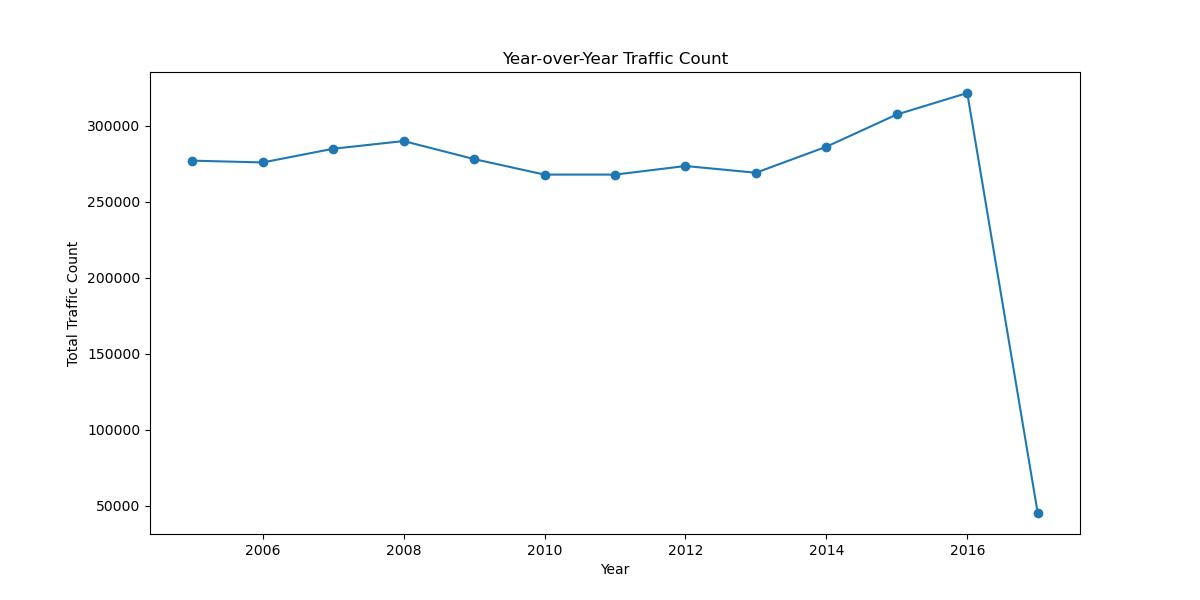
* **Time Series Trends:** Analyze the **Count** variable over time to detect seasonality, trends, and fluctuations.



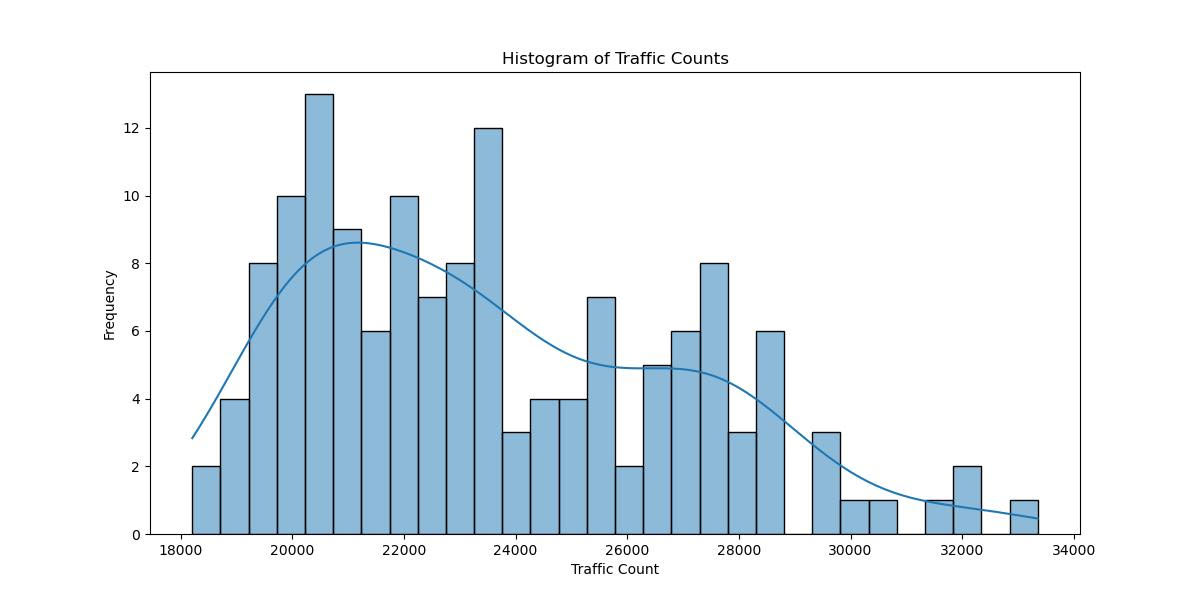
* **Monthly & Seasonal Patterns:**
  + Identify high and low traffic months.
  + Compare seasonal variations (winter vs. summer).



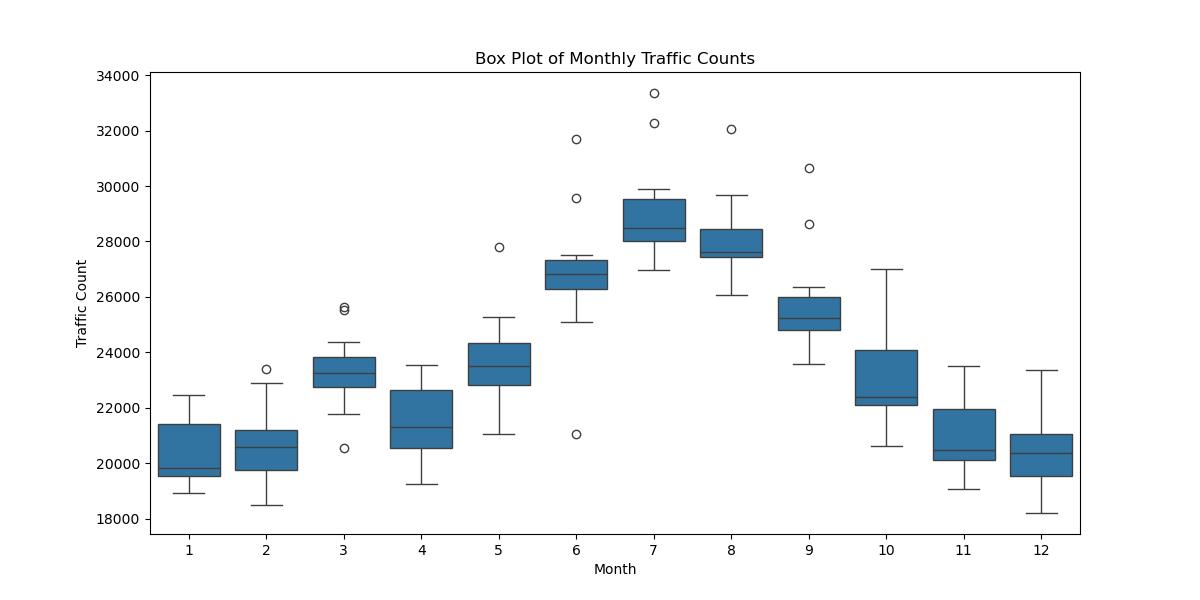
* **Year-over-Year Analysis:** Examine whether traffic has been increasing, decreasing, or remaining stable over multiple years.



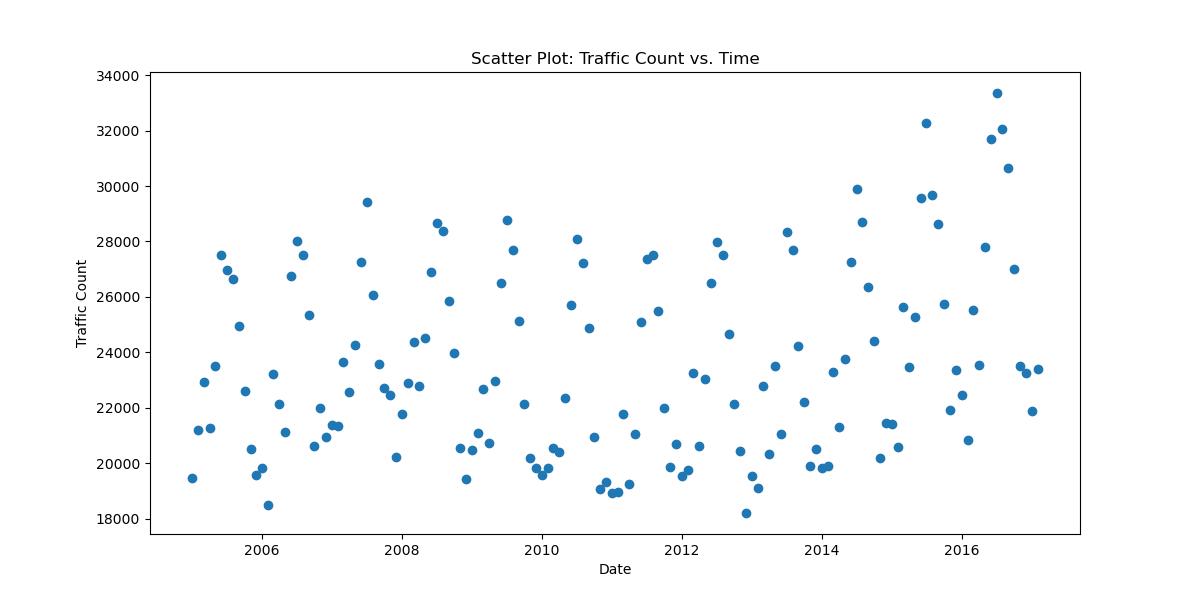
* **Histogram of Traffic Counts**
  + **X-axis:** Traffic count (number of vehicles)
  + **Y-axis:** Frequency
  + **Insight:** Shows how traffic count is distributed and whether the data is skewed.



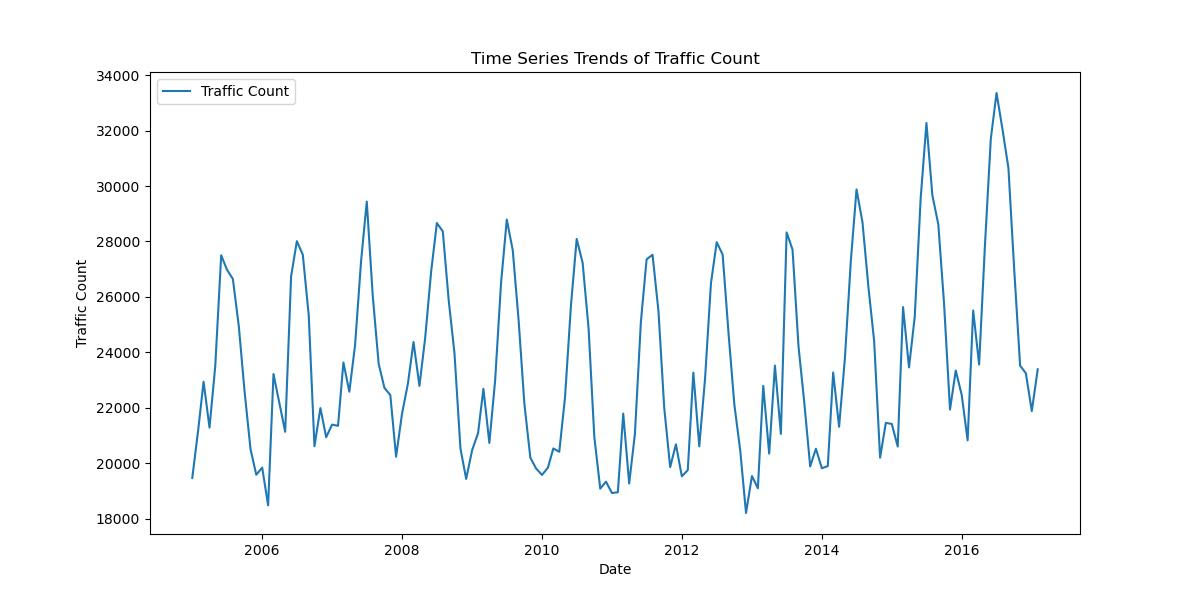
* **Box Plot of Monthly Traffic Counts**
  + **X-axis:** Month
  + **Y-axis:** Traffic count
  + **Insight:** Identifies outliers, median traffic levels, and variability in different months.



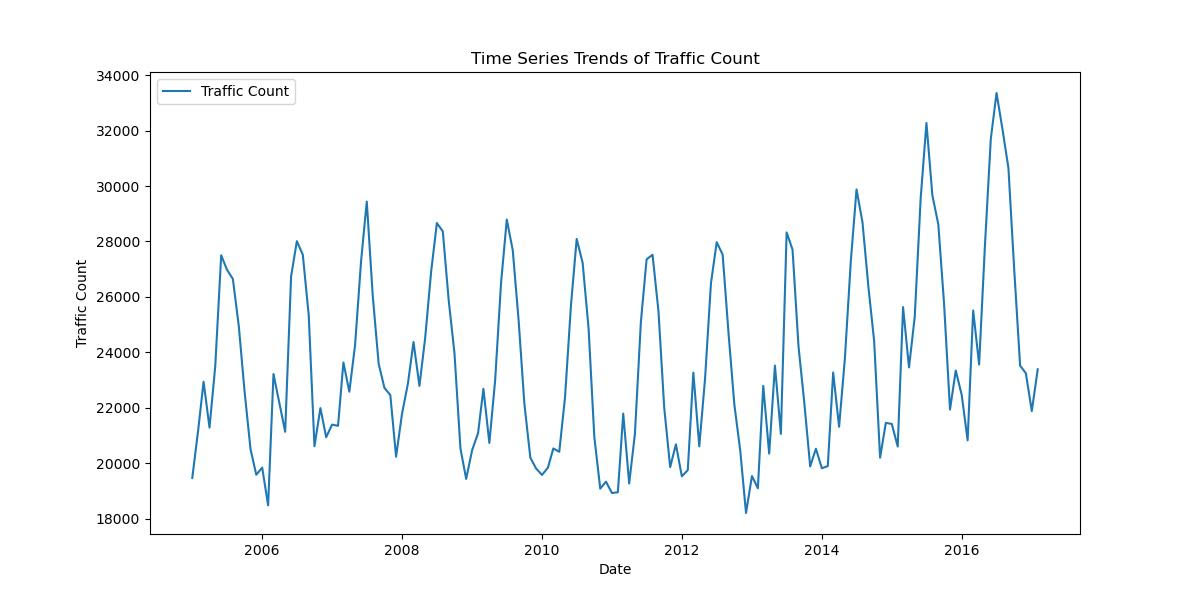
* **Scatter Plot: Traffic Count vs. Time**
  + **X-axis:** Date (months/years)
  + **Y-axis:** Traffic count
  + **Insight:** Highlights trends, seasonality, and any unusual fluctuations over time.



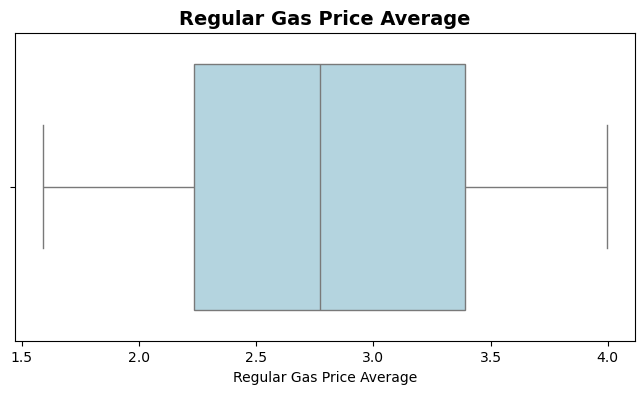
* **Time Series Line Chart of Monthly Traffic**
  + **X-axis:** Time (by month)
  + **Y-axis:** Average daily traffic count
  + **Insight:** Reveals overall trends, such as growth or decline in traffic.



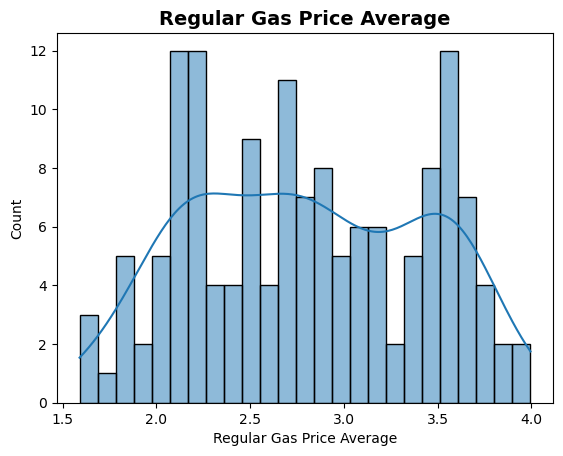
* **Rolling Average (Moving Average) Line Chart**
  + **X-axis:** Time
  + **Y-axis:** Smoothed traffic count (e.g., 3-month or 6-month rolling average)
  + **Insight:** Reduces short-term fluctuations to highlight trends.



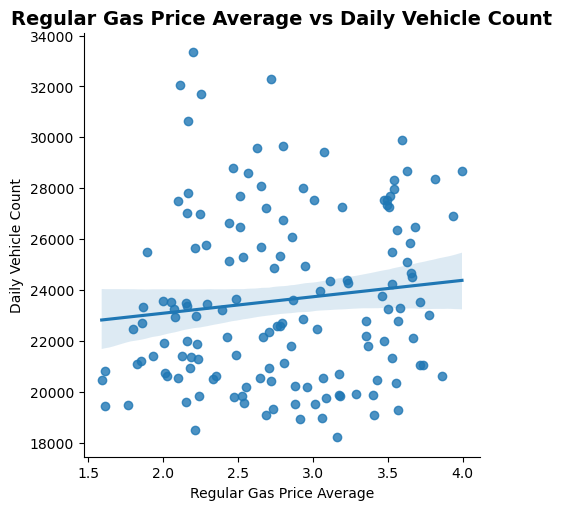
* **Boxplot of Regular Gas Price Average:** This graph shows the distribution of average regular gas prices over the congestion review timeframe.



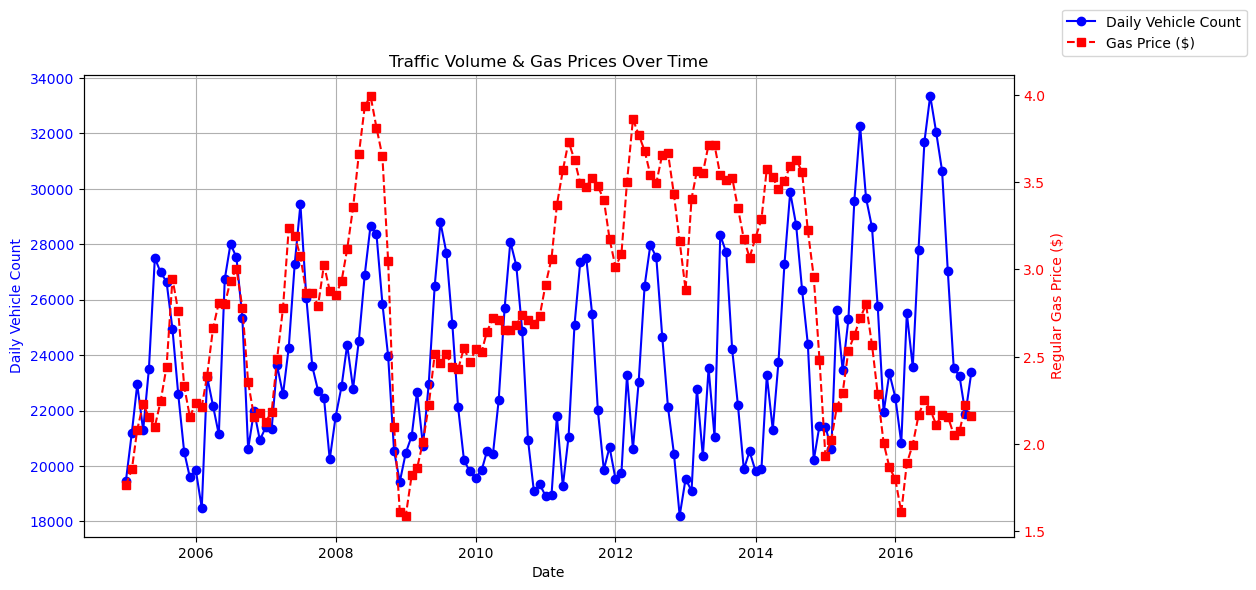
* **Histogram of Regular Gas Price Average**
  + **X-Axis:** Average regular gas prices
  + **Y-Axis:** Count of occurrences of average regular gas prices
  + **Insight:** Shows distribution of regular gas price averages. Few extreme outliers and most values fall within the standard deviation



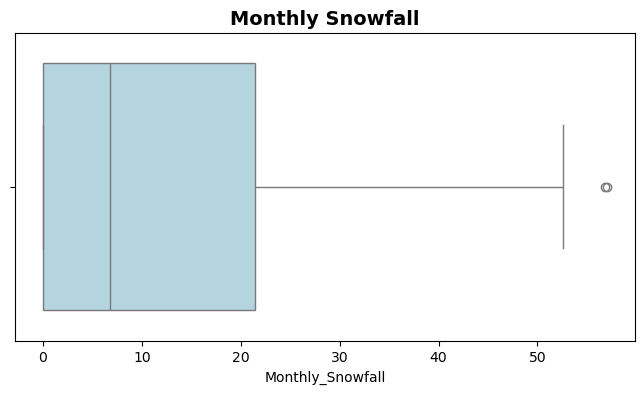
* **Scatterplot of Regular Gas Price Average vs. Daily Vehicle Count**
  + **X-Axis:** Average regular gas prices
  + **Y-Axis:** Daily Vehicle Count
  + **Insight:** Shows impact of gas prices on Daily Vehicle Count. Slight positive correlation between higher average regular gas prices and increased daily vehicle count



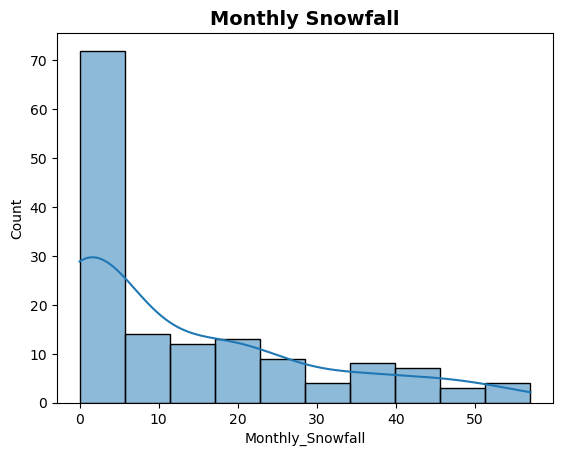
* **Dual Trending Line Chart of Regular Gas Price Average vs. Daily Vehicle Count**
  + **X-Axis:** Time (by year)
  + **Blue Y-Axis:** Daily Vehicle Count
  + **Red Y-Axis:** Regular Gas Price Average
  + **Insight:** Visually show the impact of gas prices on daily vehicle count over time showing clear trends based on increased or decreased average gas price



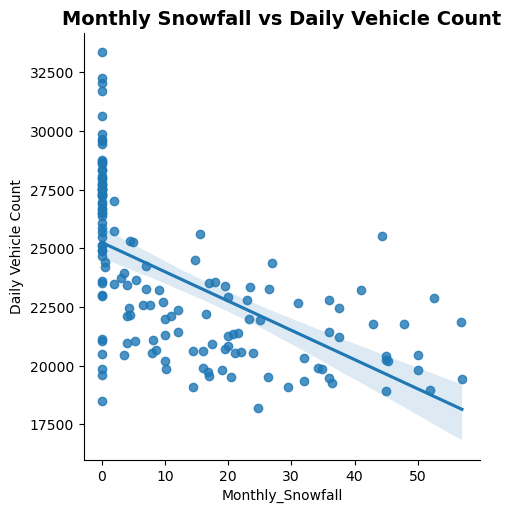
* **Boxplot of Monthly Snowfall:** This graph shows the distribution of monthly snowfall. Clear right-skew distribution with few extreme outliers. Because snowfall is seasonal, this is expected

****

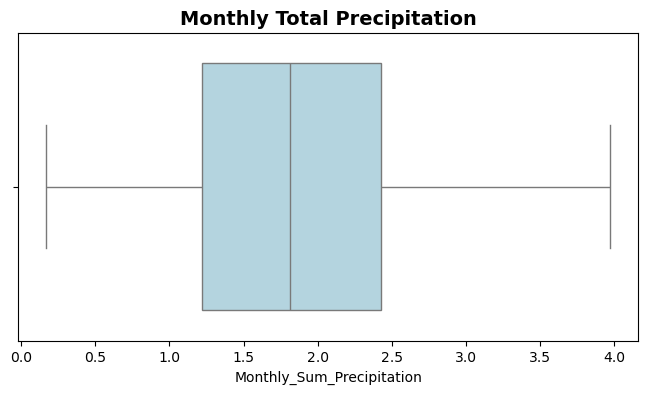
* **Histogram of Monthly Snowfall**
  + **X-Axis:** Monthly snowfall amounts
  + **Y-Axis:** Count of occurrences of monthly snowfall in inches
  + **Insight:** Shows the clearly right-skewed distribution of monthly snowfall amounts. Expected results based on seasonality



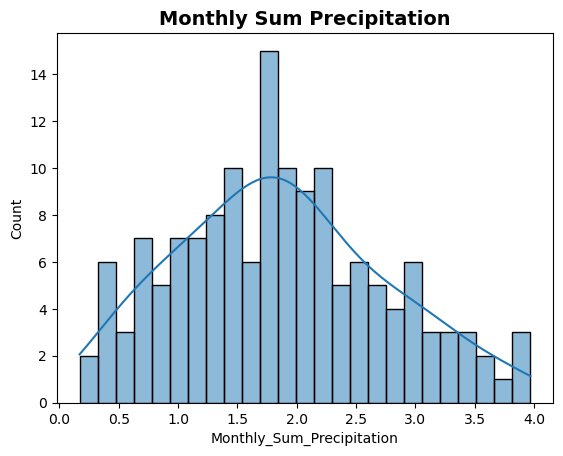
* **Scatterplot of Daily Vehicle Count vs Monthly Snowfall**
  + **X-Axis:** Monthly Snowfall
  + **Y-Axis:** Daily Vehicle Count
  + **Insight:** Shows the high negative correlation between higher snowfall amounts and decreased daily vehicle counts.



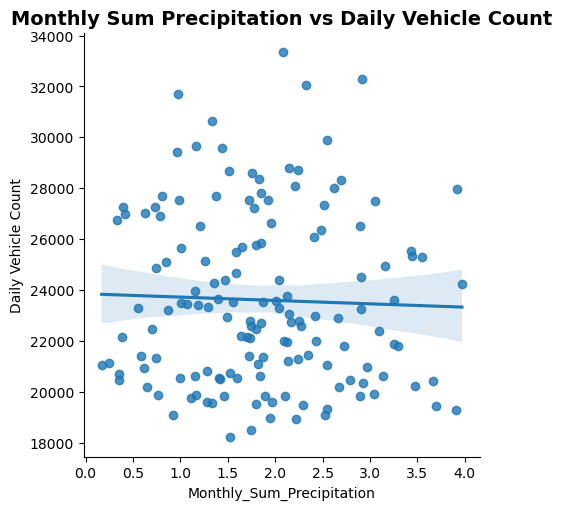
* **Boxplot of Monthly Total Precipitation:** This graph shows the distribution of monthly total precipitation amounts. It has a slight right-skew but is fairly normally distributed.



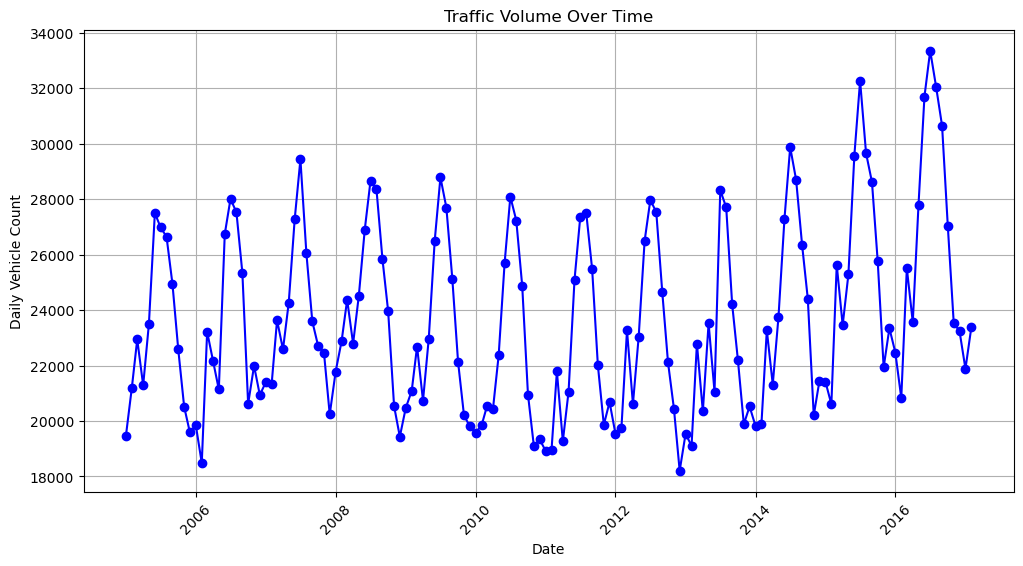
* **Histogram of Monthly Total Precipitation**
  + **X-Axis:** Monthly total precipitation
  + **Y-Axis:** Count of occurrences of monthly precipitation amounts
  + **Insight:** Confirms the very slight right-skew of the distribution with few extreme outliers



* **Scatterplot of Daily Vehicle Count vs Monthly Sum Precipitation**
  + **X-Axis:** Monthly total precipitation
  + **Y-Axis:** Daily Vehicle Count
  + **Insight:** Shows a very slight negative correlation between increased precipitation and lowered total vehicle count

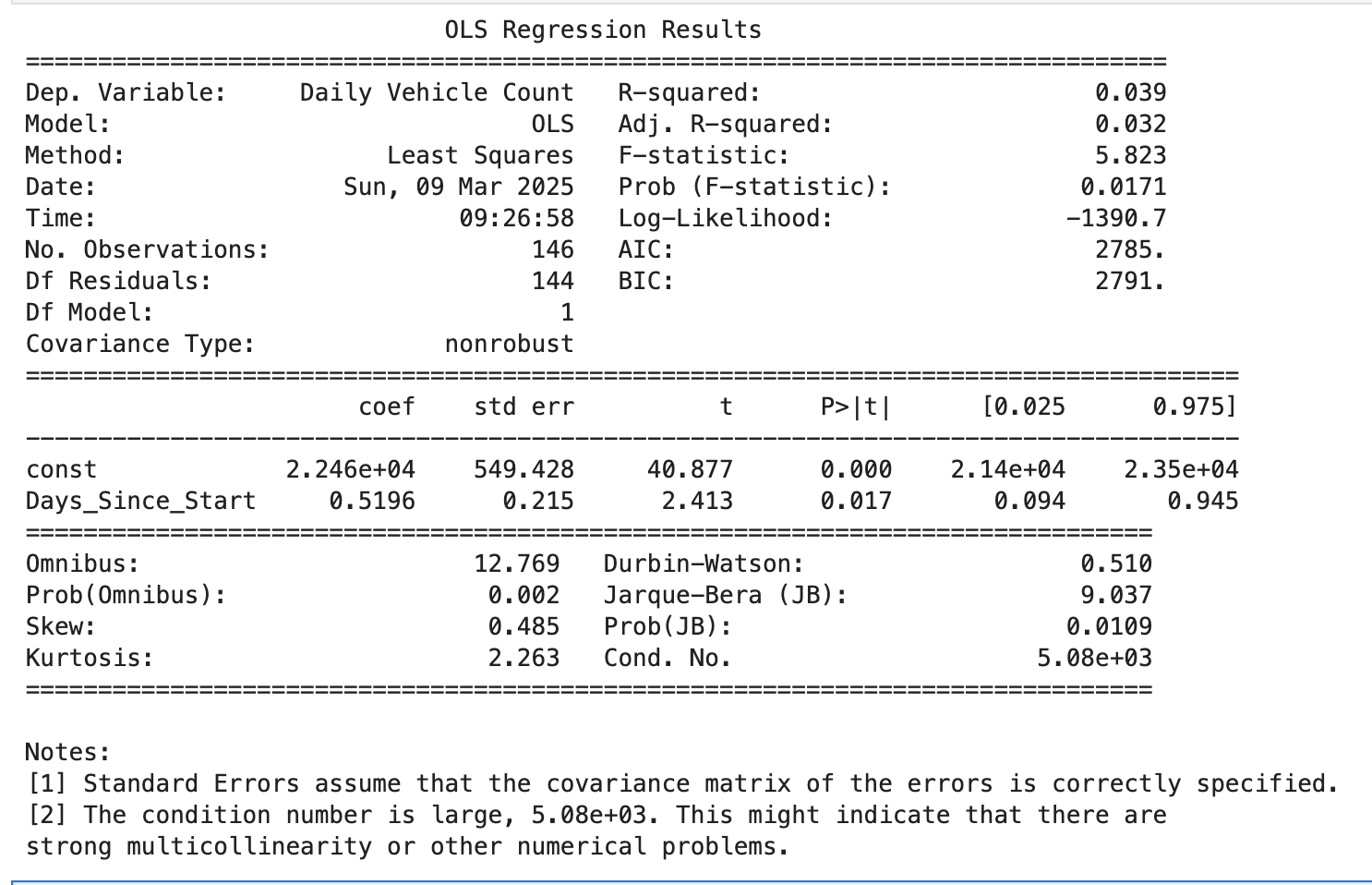


* **Time Series Line Chart of Traffic Over Time**
  + **X-Axis:** Time (by year)
  + **Y-Axis:** Daily Vehicle Count
  + **Insight:** Provides a trending overview of traffic over time showing a gradual increase in traffic as time goes by

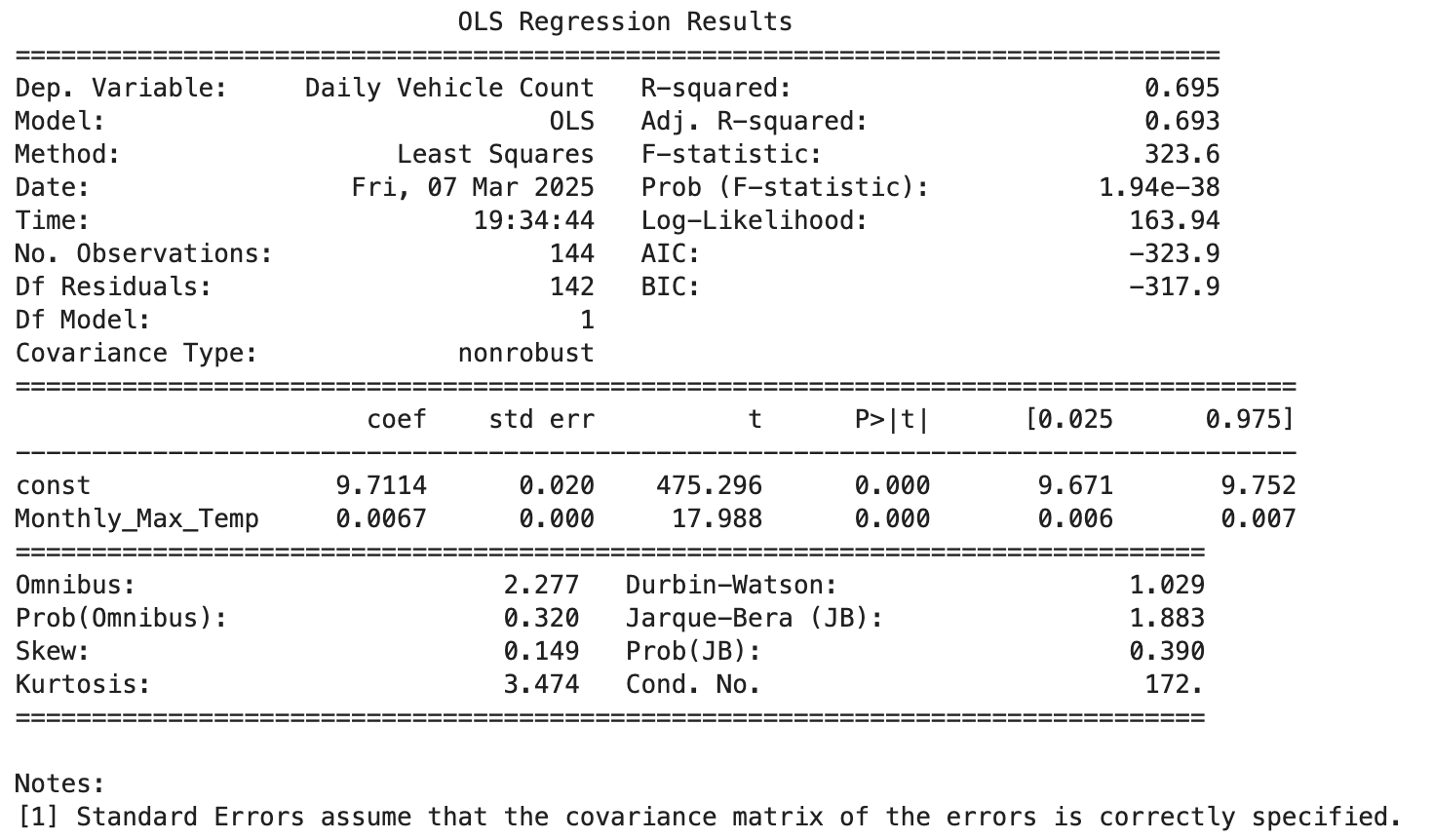


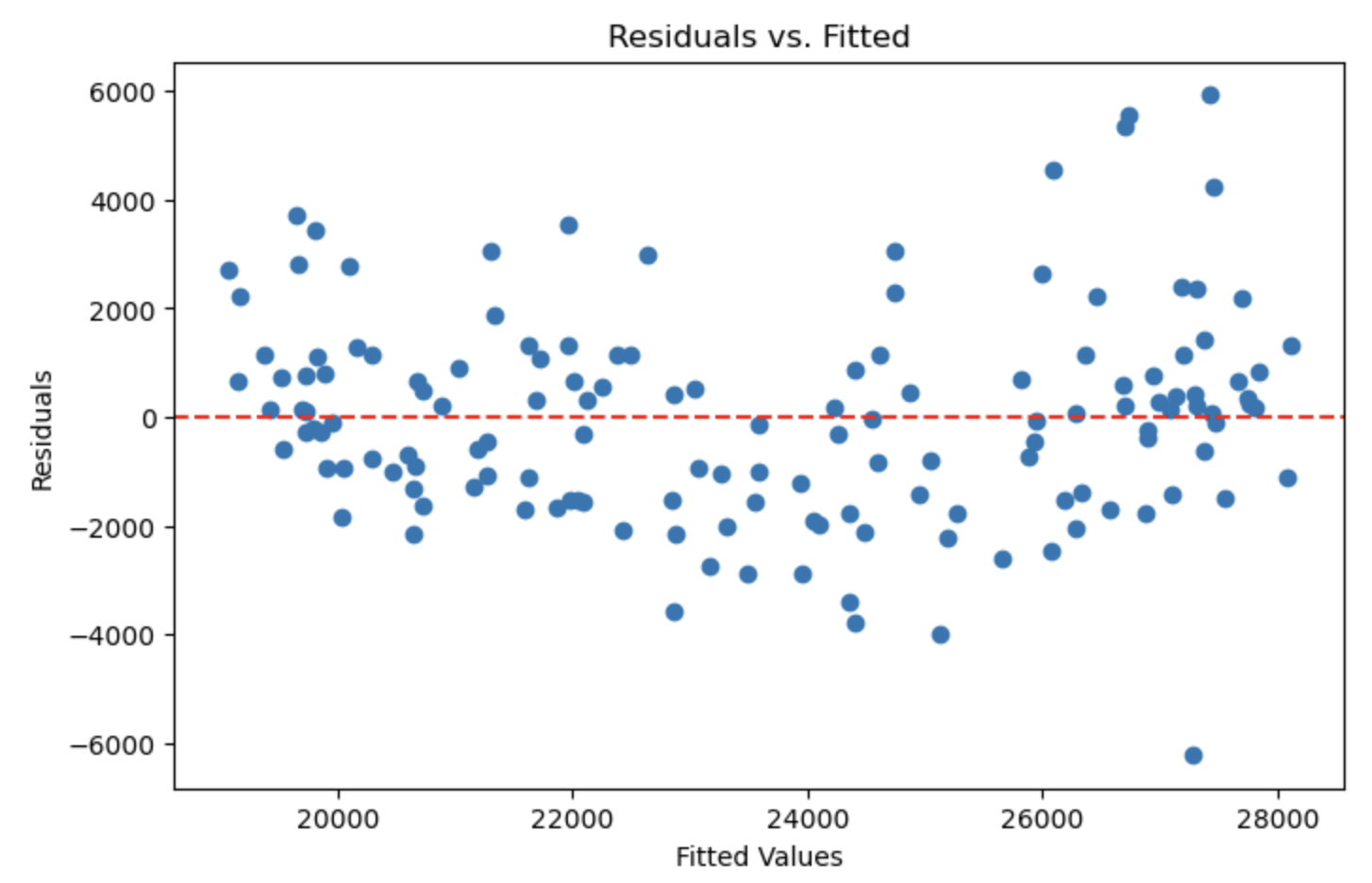
### **3. Statistical & Predictive Analysis**

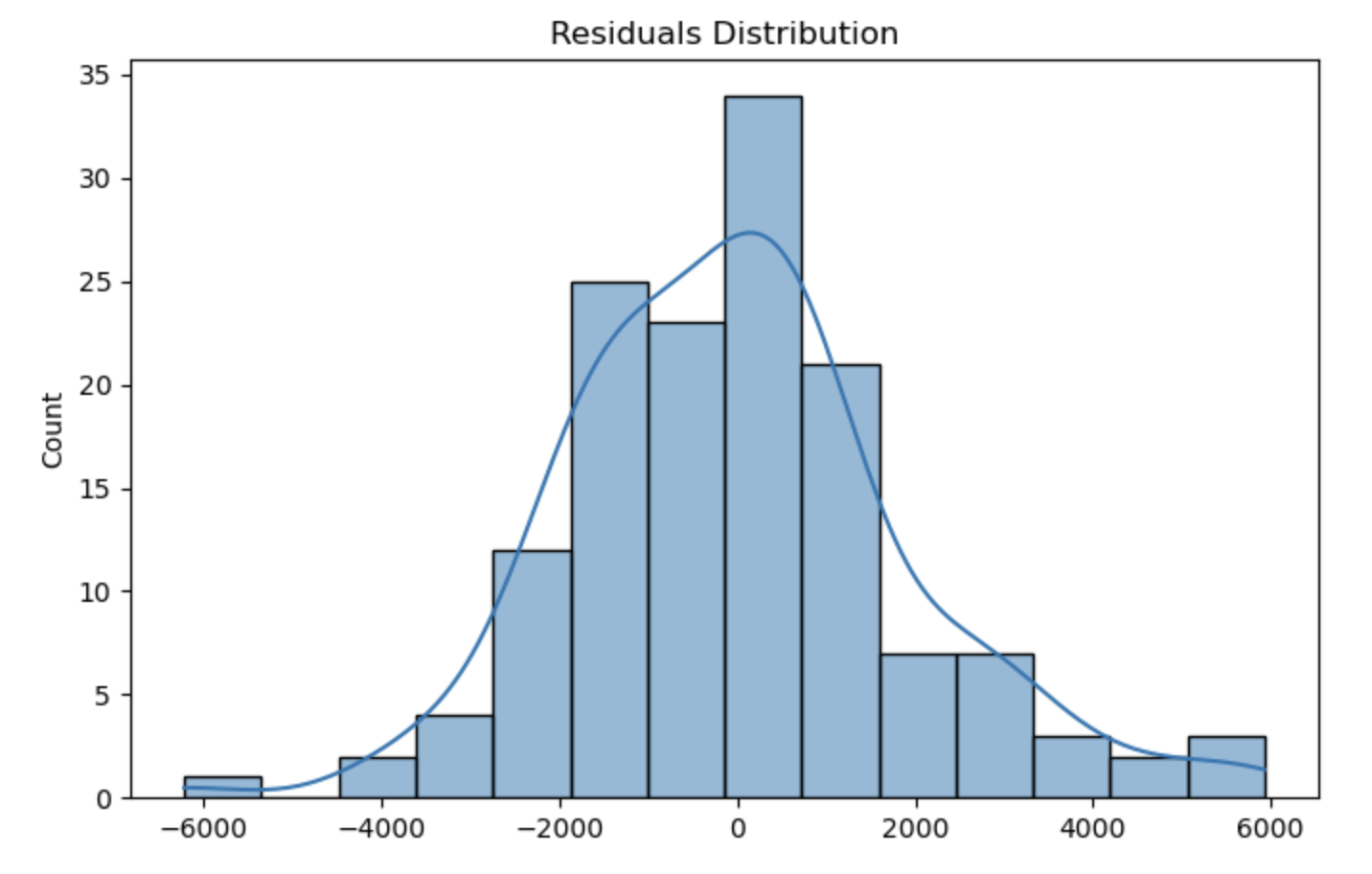
* **Moving Averages & Smoothing Techniques:** Apply techniques like:
  + **Simple Moving Average (SMA)** to smooth fluctuations.
  + **Exponential Smoothing (ETS) models** to emphasize recent trends.
* **Trend Analysis:** Determine if traffic is increasing, decreasing, or cyclical using:
  + **Simple Linear Regression:** We attempted two simple linear regression models: Traffic Over Time and Traffic vs Maximum Temperature
    - In the traffic over time model, it was confirmed that time itself is not a good indicator of traffic volume. With an R^2 score of 3.9%, time itself has only a slight impact. No additional visualizations were completed on this model since the outcome was so poor. This model led us to review a more seasonal approach to time in the multiple regression model.

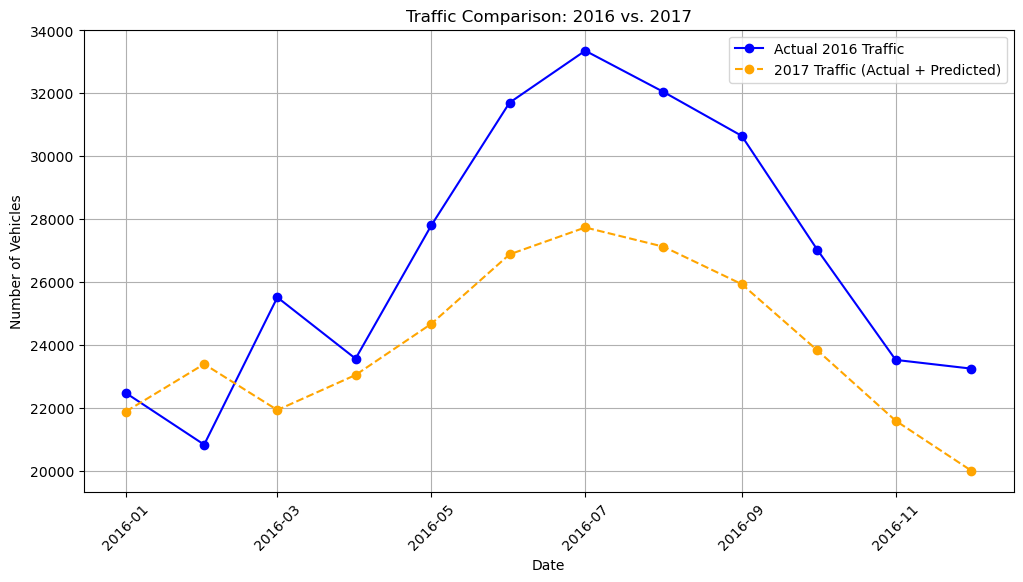


* + - In the Traffic vs Monthly Maximum Temperature model, there was more success. After a log transformation, the adjusted R^2 value for this model was 69.3%, meaning that maximum temperature had a strong impact on the traffic counts. This is understandable as more people generally travel during warmer temperatures. This model had multiple violations of assumptions of linear regression. It showed signs of autocorrelation and multicollinearity. It did indicate a relatively normal distribution. Visualizations of the residuals do show a generally normal distribution with few outliers. A general prediction for the remaining months in 2017 was also performed based on this model and was compared to actual traffic numbers for 2016. It followed a similar pattern but may not be completely accurate as other factors that may impact traffic have not been considered.

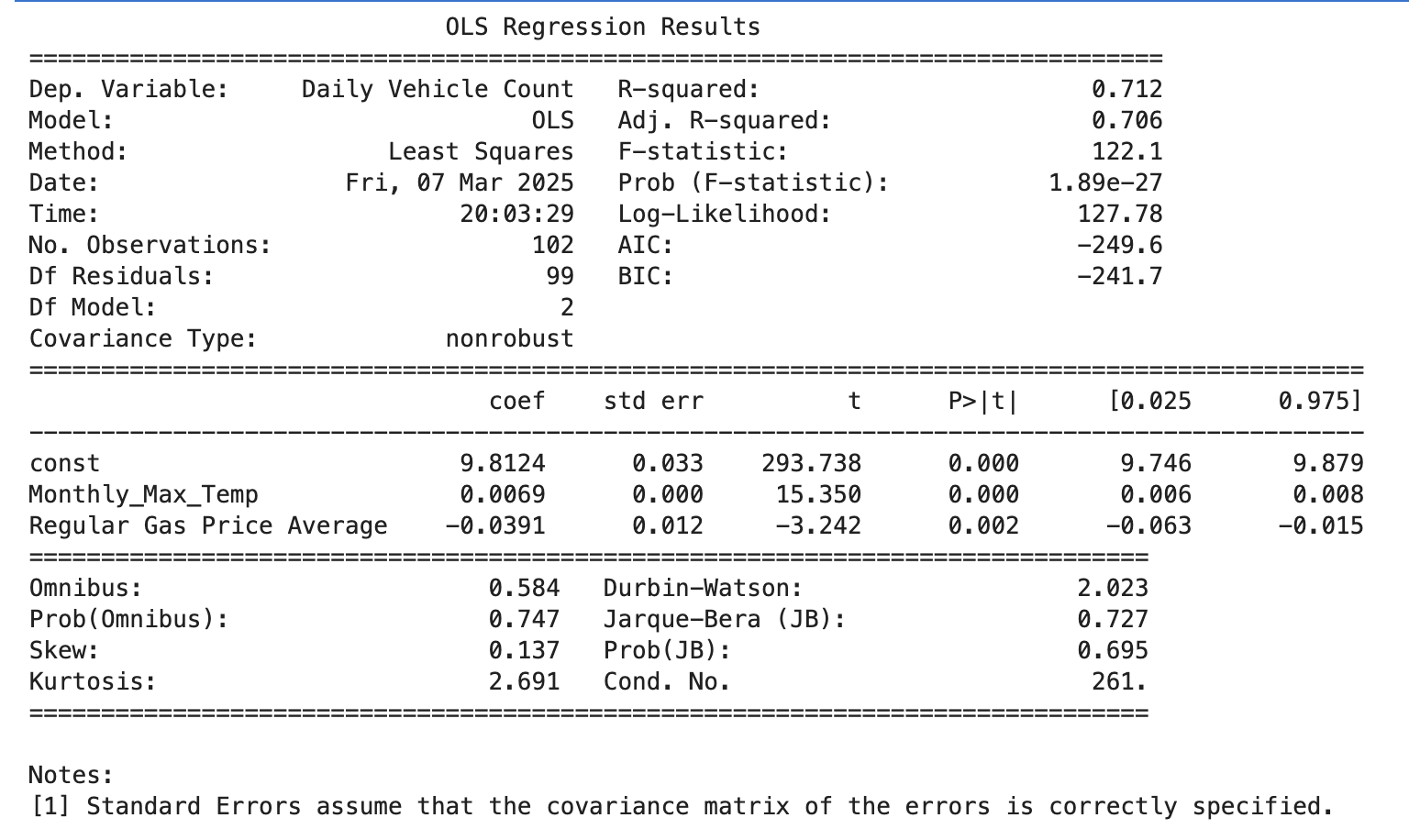


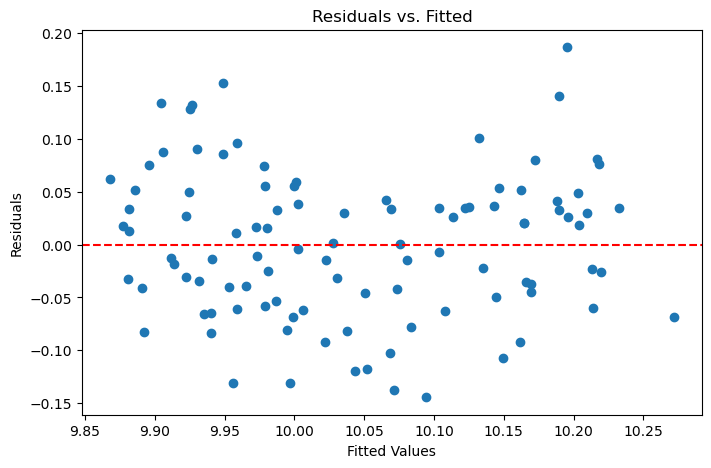


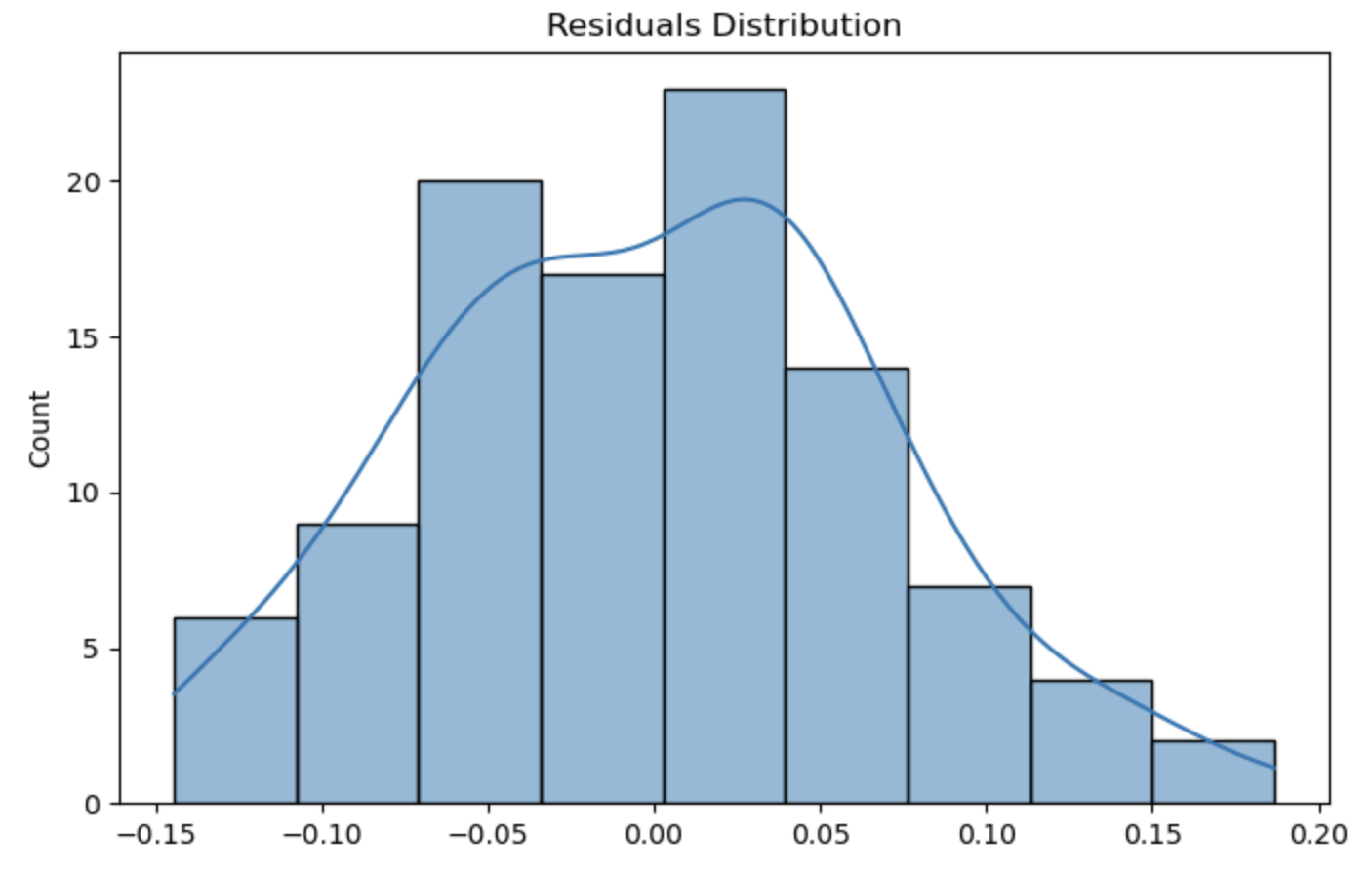


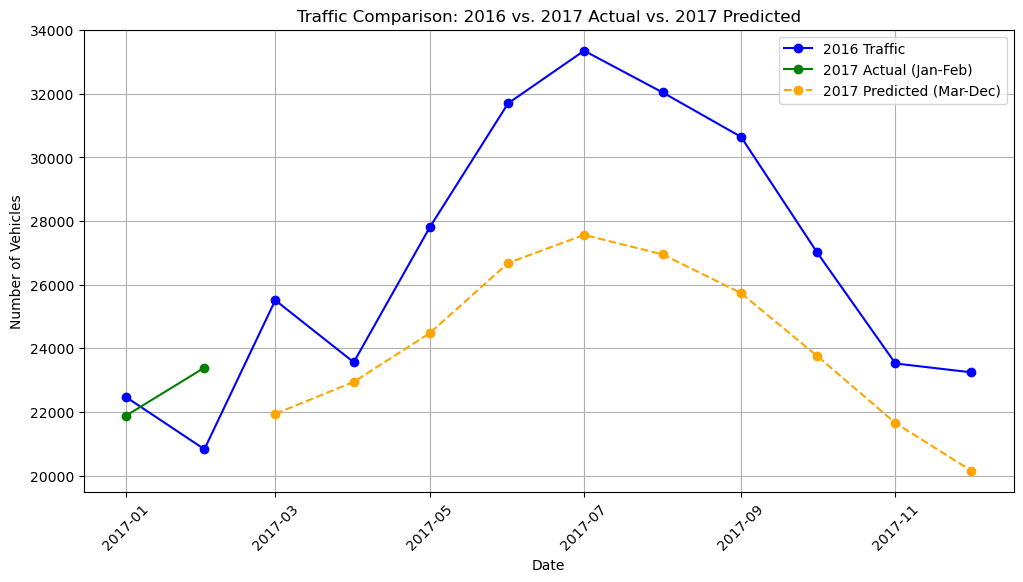


* + **Multiple Linear Regression:** We attempted multiple iterations of multiple linear regression models to see if we could determine what other variables, outside of monthly maximum temperature, impacted traffic counts.
    - Using all of the variables returned a model that had a problem with multicollinearity issues. This was expected since we were using multiple different gas grades as well as minimum and maximum temperature variables, which are innately related. This model was disregarded.
    - The next model attempted was using Monthly Maximum Temperature and Regular Gas Price Average and a log transformation. This model was much more successful with an adjusted R^2 value of 70.6%. Adding in the gas prices increased the ability to account for changes in traffic based on these variables. The residuals in the model were fairly normally distributed. We also attempted a prediction for the remaining months in 2017 and compared them to actual values in 2016. The prediction was more in-line with the actual data from 2016.

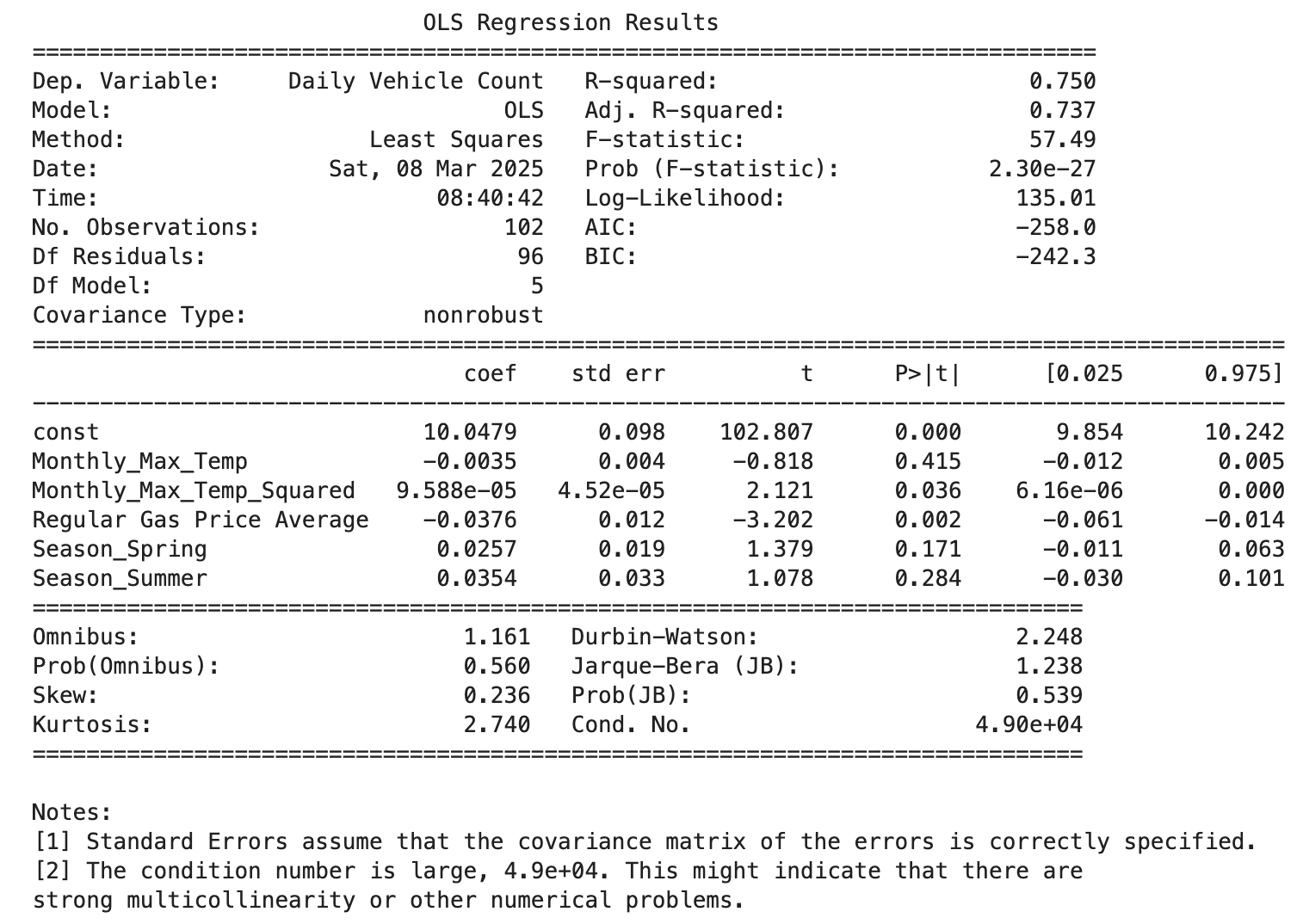


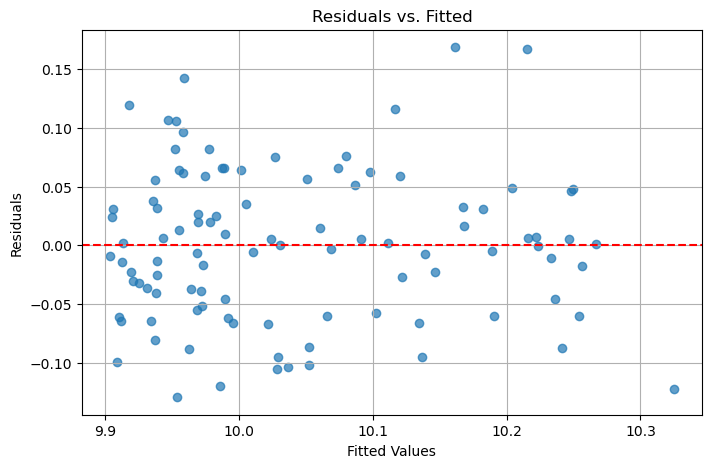


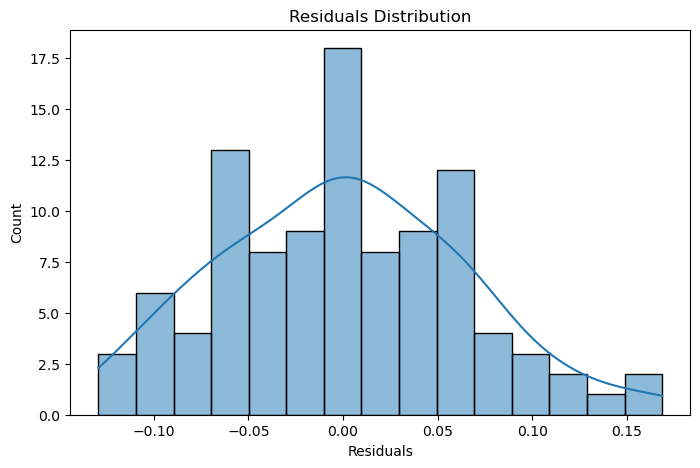


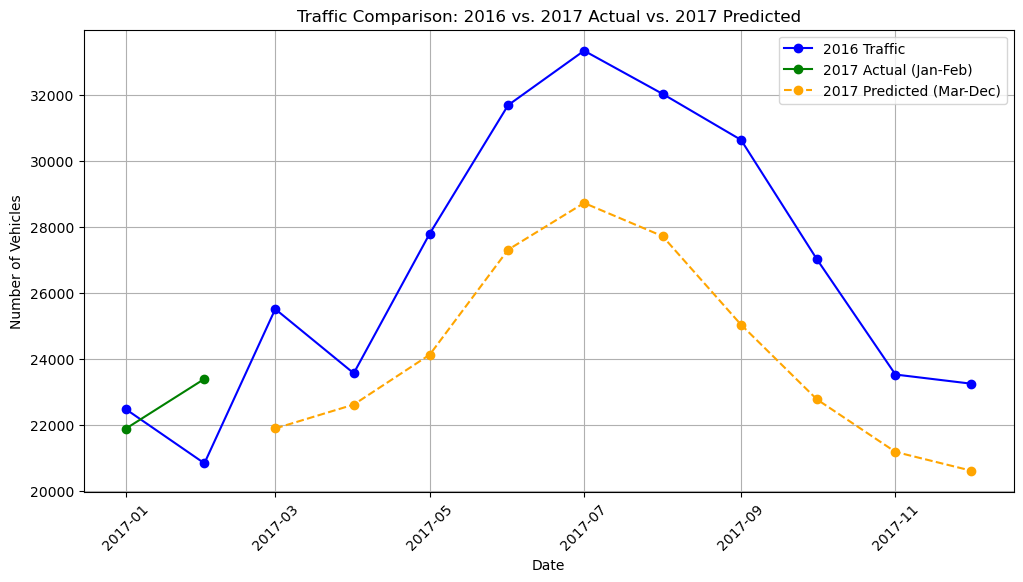


* + - The final and most impactful model was created using the Regular Gas Price Average, Maximum Temperature, a quadratic transformation of Maximum Temperature, and dummy values for seasonality (based on month). This model returned an adjusted R^2 value of 73.7%. Other indicators on violations were well within bounds meaning little autocorrelation, normal distribution, no multicollinearity, and mostly equal variances within the dataset. A prediction was made for the remaining months in 2017 and compared to actual data from 2016. This prediction most similarly follows the curve of the actual known data from the prior year (2016).



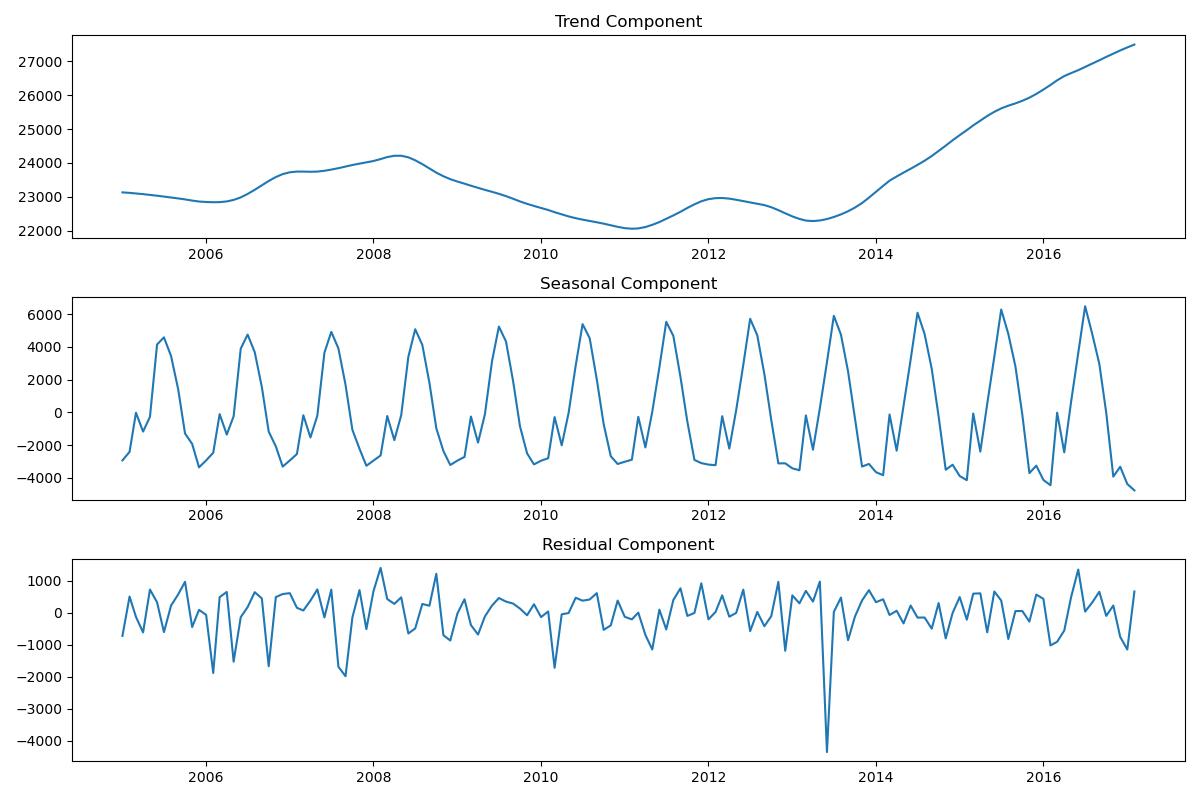




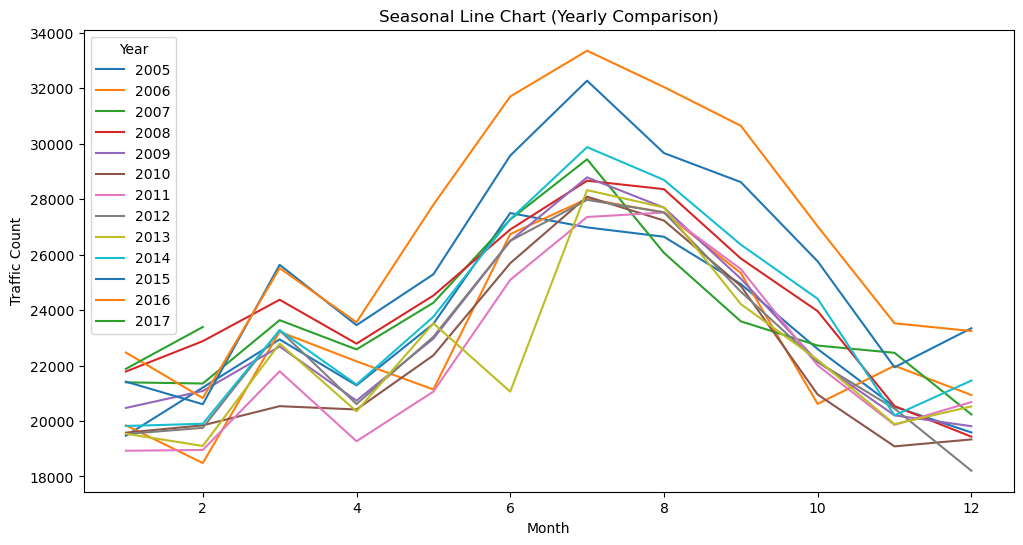


* **Seasonal Decomposition:** Decompose
  + traffic data into **trend by months**
  + **seasonality (snow and precipitation)**

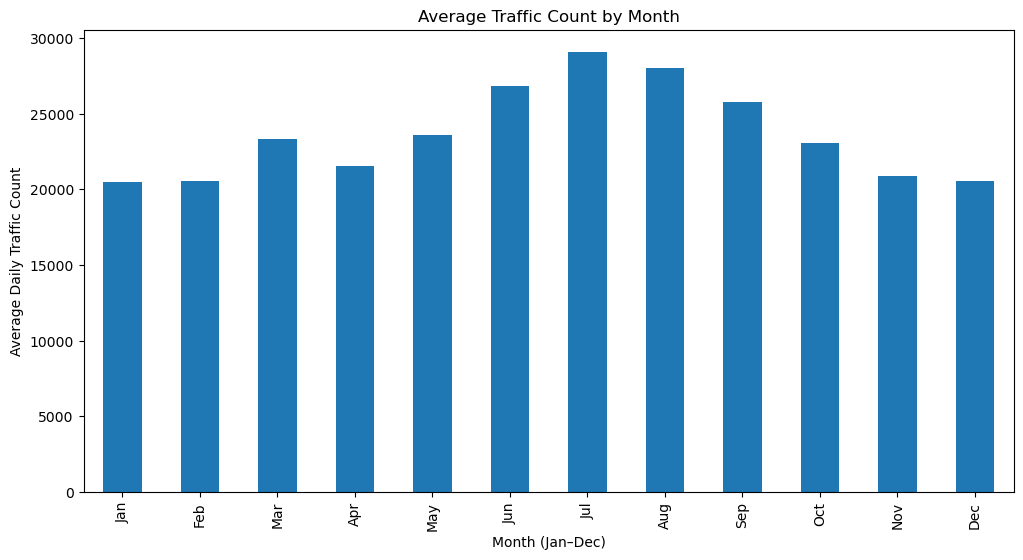
(using methods like STL decomposition).



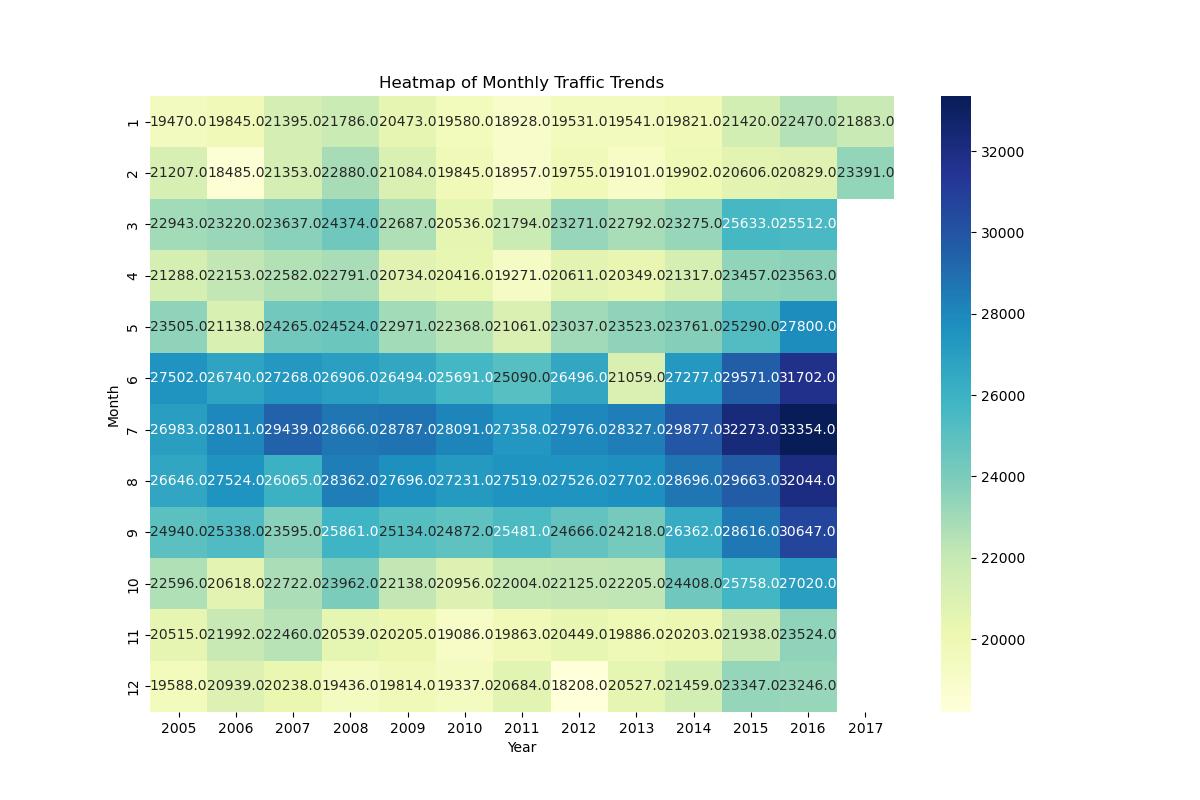
* **Seasonal Line Chart (Yearly Comparison)**
  + **X-axis:** Month
  + **Y-axis:** Traffic count
  + **Different Lines:** Separate years (e.g., 2015, 2016, 2017)
  + **Insight:** Shows if seasonal patterns (winter/summer peaks) are consistent across years.



* **Bar Chart: Average Traffic Count by Month**
  + **X-axis:** Month (Jan–Dec)
  + **Y-axis:** Average daily traffic count
  + **Insight:** Highlights peak travel seasons (e.g., winter for ski traffic or summer for tourism).



* **Heatmap of Monthly Traffic Trends**
  + **X-axis:** Year
  + **Y-axis:** Month
  + **Color Intensity:** Traffic count
  + **Insight:** Visualizes patterns over multiple years to detect periodic trends.



### 

### 

### 

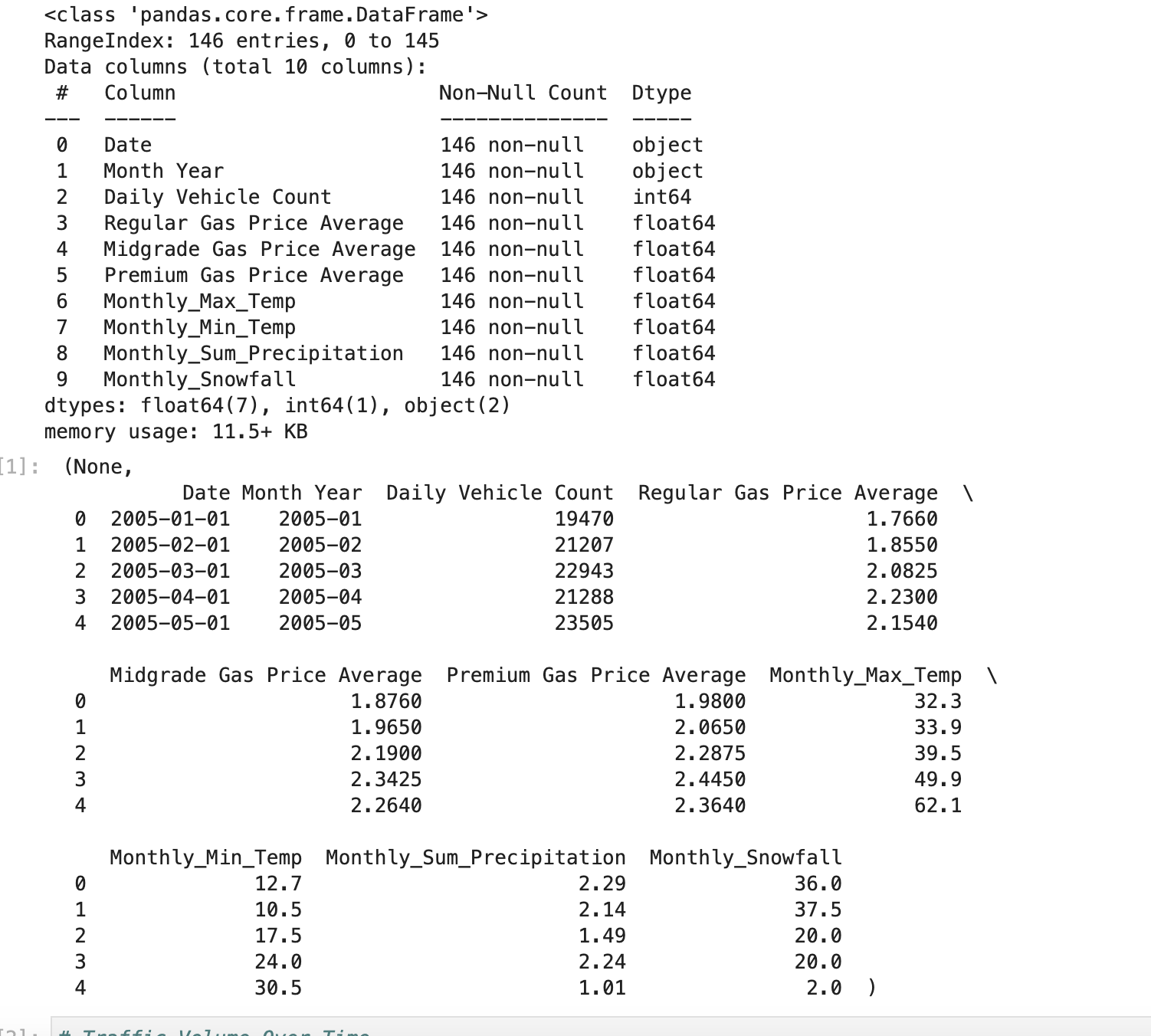
### 

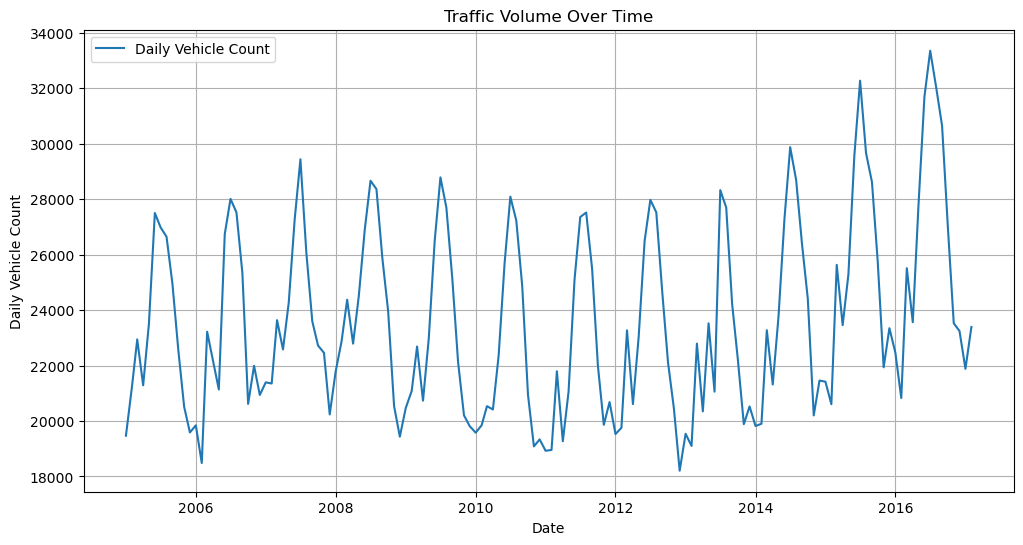
### 

### 

### **4. Forecasting Traffic for the Rest of 2017 (Isuri)**

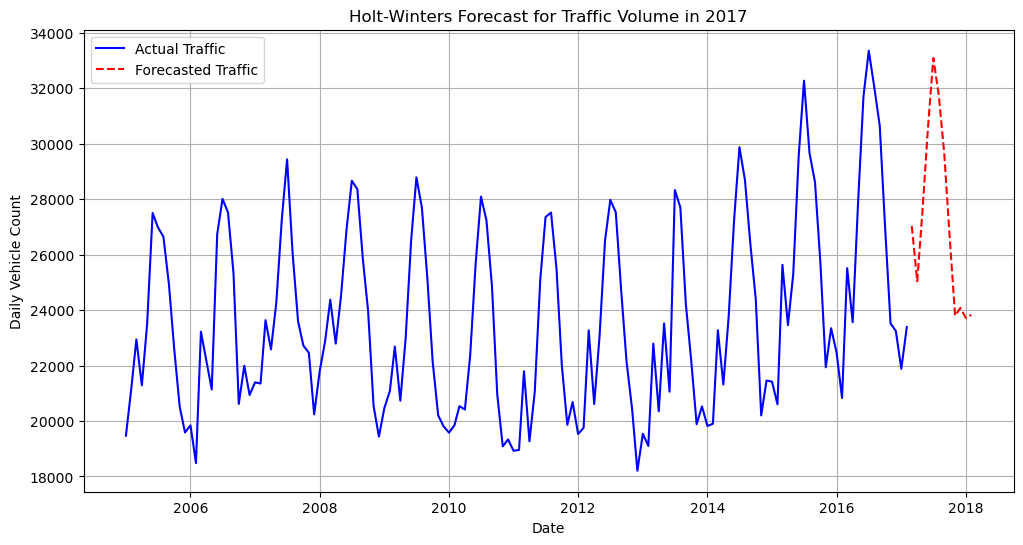
* **Time Series Models:**
  + **Holt-Winters (Exponential Smoothing):** Captures seasonality and trends for shorter-term forecasting.





*Analysis*

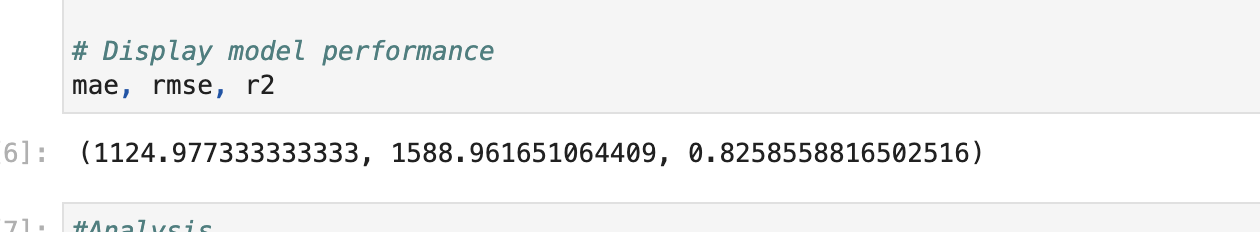
* *Traffic Volume Over Time (Historical Data)*
  + *The initial time-series plot shows fluctuations in daily vehicle counts over time*
  + *There appears to be cyclical patterns, suggesting that traffic volume varies based on seasonal factors* *such as weather conditions, economic cycles, or fuel prices.*
    - *The data appears to have an upward trend, meaning that traffic volume is gradually increasing over time.*

****

***Analysis***

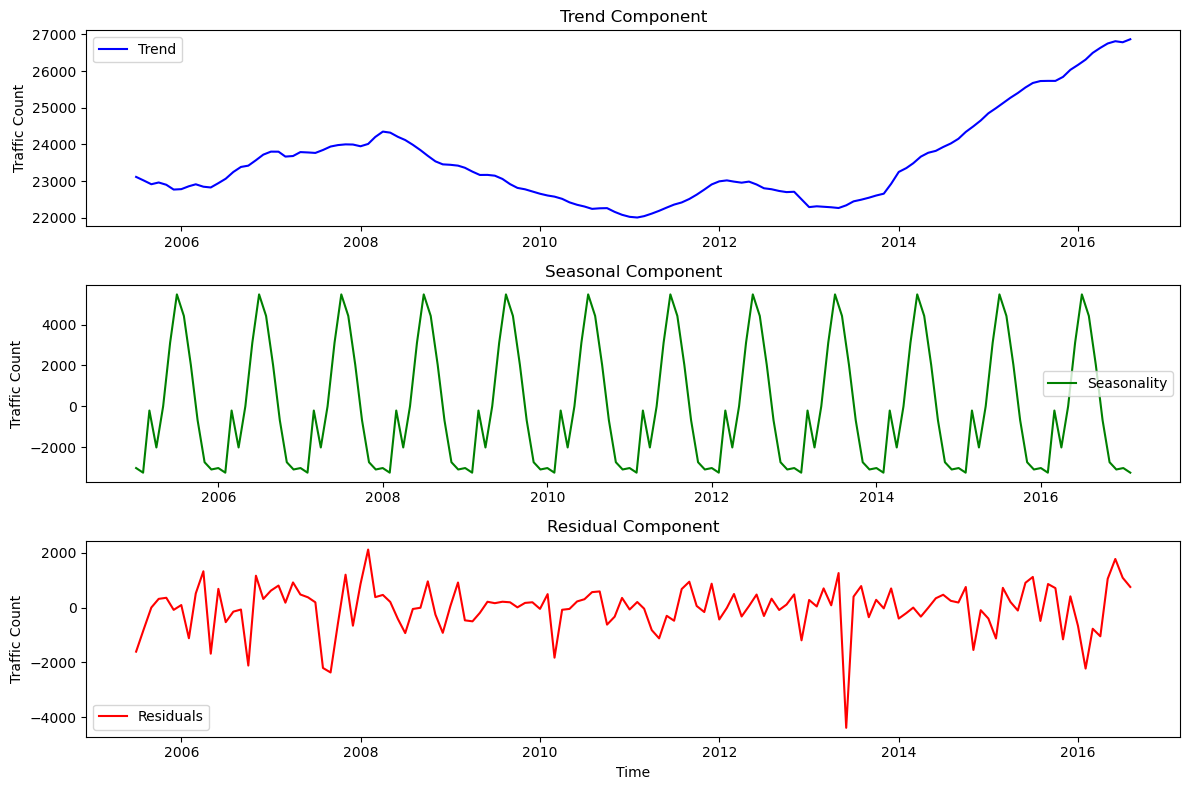
* *Holt-Winters Forecast for 2017*
  + *The Holt-Winters model captures both trend and seasonality,* 
    - *More accurate short-term forecast.*
  + *The forecasted traffic volume (red dashed line) follows the historical pattern*
    - *Traffic volume is expected to continue its seasonal fluctuations while gradually increasing.*
  + *If there were external matters like economic shifts, extreme weather, or policy changes- the actual traffic might deviate from the forecast.*

**Machine Learning Models (if applicable):** Consider Decision Trees or Random Forest if patterns are complex.



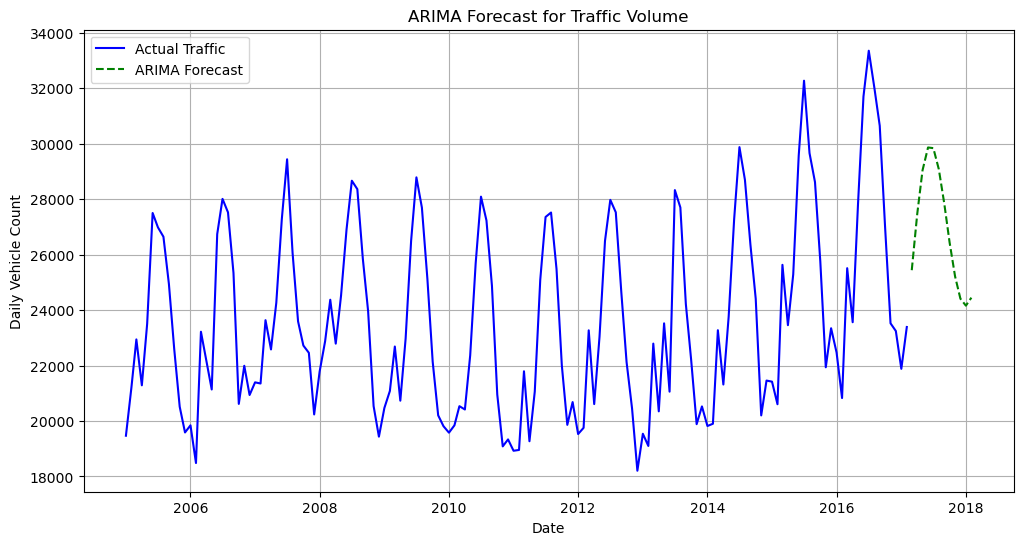
***Analysis***

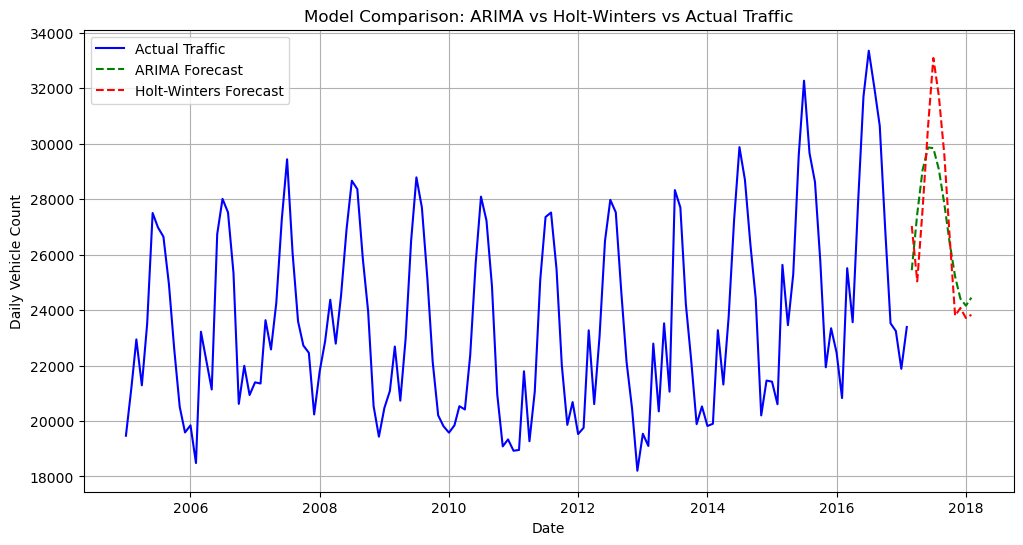
* ***Model Performance:***
  + ***Mean Absolute Error (MAE):*** *1,124.98 On average, the model's predictions deviate from actual values by about 1,125 vehicles.*
  + ***Root Mean Squared Error (RMSE):*** *1,588.96. A slightly higher error metric, indicates some variability in the model’s performance.*
  + ***R-squared (R²)****: 0.826 → The model explains 82.6% of the variance in traffic volume, which is a strong performance.*
  + *The model performs well in capturing patterns in the data, with an R² value above 0.8, indicating it explains most of the variability in traffic.*
  + *The errors (MAE & RMSE) suggest that while predictions are relatively accurate, there can be deviations due* *to unexpected external factors like policy changes, roadwork, or other events.*
* **Decomposition Plot (Trend, Seasonality, Residuals)**



**Analysis**

* Trend Component (Top Graph - Blue Line)
  + The overall trend appears stable with slight fluctuations rather than a strong upward or downward movement.
  + This suggests traffic volume remains relatively consistent over time, with no major long-term increase or decrease.
  + However, some periods show gradual growth or dips, which could be influenced by external factors
* Seasonal Component (Middle Graph - Green Line)
  + The seasonal pattern is well-defined, showing periodic increases and decreases in traffic volume.
  + There are regular peaks and dips, which indicate:
    - Higher traffic in certain months, possibly due to tourism, summer months, or holiday travel.
    - Lower traffic in specific months, likely due to colder weather reducing travel or other seasonal effects.
    - This confirms that traffic is not random but follows a predictable annual cycle.
* Residual Component (Bottom Graph - Red Line)
  + The residuals show fluctuations that are not explained by trend or seasonality.
  + While most values are near zero, some spikes and dips occur, suggesting unexpected anomalies in traffic patterns.
  + Possible reasons for these anomalies:
    - Extreme weather events (e.g., snowstorms, heavy rain reducing traffic).
    - Construction projects or road closures temporarily disrupting normal traffic flow.
    - Major local events
* Seasonality is a dominant factor in traffic volume—planning should account for these regular fluctuations.
* The overall trend is stable, meaning traffic growth is not extreme, but it fluctuates within expected patterns.
* Anomalies in residuals indicate occasional disruptions, requiring further analysis to pinpoint causes (weather, economy, policy changes).
* **Model Comparison Line Chart - ARIMA**



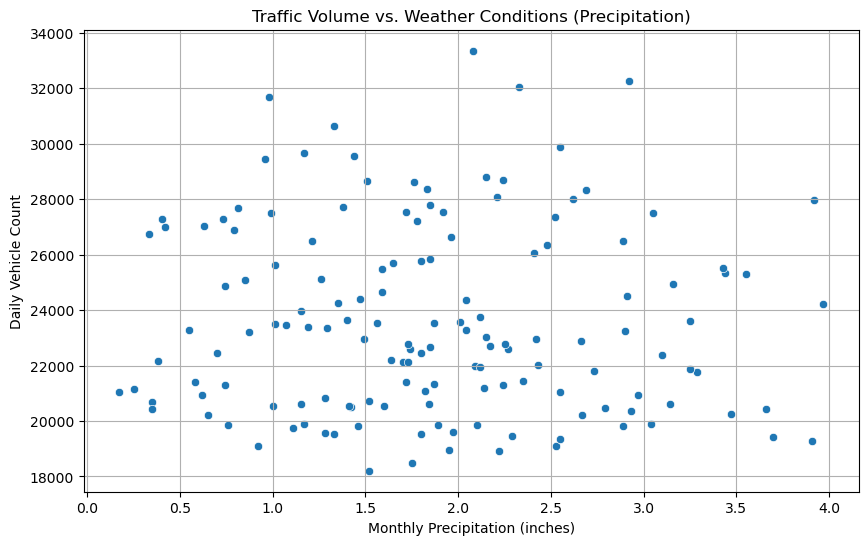


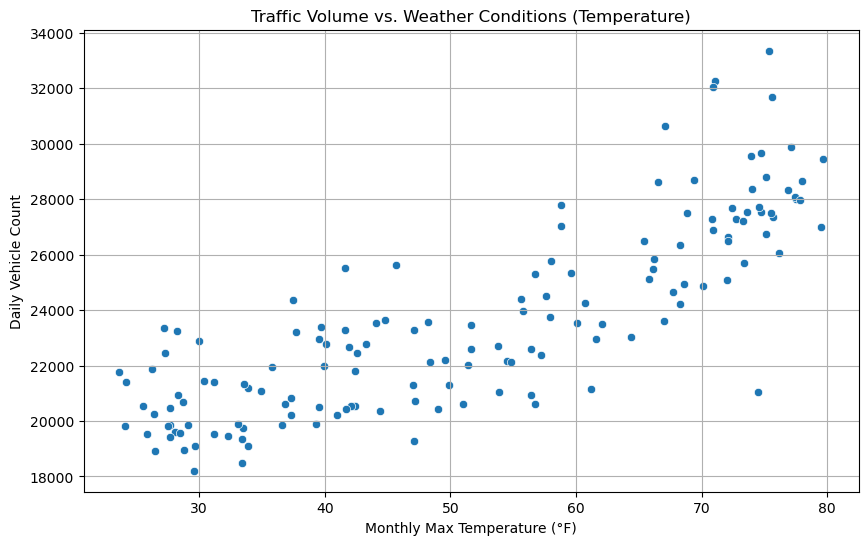
**Analysis**

* **Actual Traffic (Blue Line)**
  + traffic follows a clear seasonal pattern, with noticeable peaks and valleys.
  + The fluctuations suggest that traffic varies significantly across months, possibly influenced by weather, holidays, or external factors.
* **ARIMA Forecast (Green Dashed Line)**
  + It doesn’t fully reflect the recurring peaks and dips seen in the actual data.
  + ARIMA is better for long-term trends but not ideal for short-term, highly seasonal predictions.
* **Holt-Winters Forecast (Red Dashed Line)**
  + The Holt-Winters model mirrors the seasonal trends much more accurately than ARIMA.
  + It follows the ups and downs of actual traffic, making it more responsive to short-term fluctuations.
  + This suggests that Holt-Winters is more suitable for traffic forecasting, especially when seasonality is a key factor.
* Holt-Winters outperforms ARIMA in capturing seasonal fluctuations, making it more accurate for short-term traffic forecasting.
* ARIMA is smoother and better for long-term trends, but it does not account for strong seasonal effects as well.
* Traffic forecasting should prioritize models that incorporate seasonality, as seen with Holt-Winters.

### **5. Recommendations**

* **Actionable Recommendations based on External factors**
* **Traffic vs. Weather Scatter Plot**
  + **X-axis:** Weather condition (e.g., precipitation, temperature)
  + **Y-axis:** Traffic count
  + **Insight:** Determines if adverse weather impacts congestion.
* **Bar Chart: Traffic by Day of the Week**
  + **X-axis:** Day (Monday–Sunday)
  + **Y-axis:** Average traffic count
  + **Insight:** Highlights which days experience the most congestion





**Analysis**

* Precipitation Impact:
  + The scatter plot suggests a negative correlation between precipitation levels and traffic volume.
  + As precipitation increases (rain/snow), daily vehicle count decreases, meaning people avoid driving in poor weather conditions.
* Temperature Impact:
  + The scatter plot suggests a positive correlation between temperature and traffic volume.
  + Warmer temperatures lead to increased traffic, likely due to:
    - More recreational activities, summer travel, and outdoor events.
* Increased tourism in Colorado’s warmer months:
  + Colder months see lower traffic volumes, except for ski season, where mountain regions might see an increase in weekend traffic.
  + Higher temperatures correlate with increased traffic, likely due to summer travels and outdoor activities
  + Lower temperatures may reduce traffic, except for peak commuting periods.
  + CDOT can prepare for higher summer traffic by increasing road maintenance, adjusting speed limits, and improving public transport availability during peak tourism months.

Recommendations:

From these graphs here are some following recommendations

* AI-powered chatbox pop up on Waze, google maps, or other map apps- whenever the driver routes to a destination that uses a specific highway, curated questions can be asked. This Data can help CDOT take necessary action to either prevent or prep for certain situations and driving conditions. And then send necessary alerts to the driver.
  + This is important for safety, time
* AI-powered road condition alerts-CDOT should enhance road drainage systems, increase road salting in winter, and optimize real-time traffic alerts for severe weather.
* CDOT can prepare for higher summer traffic by increasing road maintenance, adjusting speed limits, and improving public transport availability during peak tourism months.
* Improve Public Transit & Carpooling Options

**Appendix Cont.**

**Code**

**Python Code PDF and Original Notebooks:**

**Helena Mabey’s Github link:** [**Group Project 2**](https://github.com/helenamabey/spring_stats_hw2/tree/main)

**Python Codes (Lauren):**

**Time Series Trends of Traffic Count**

plt.figure(figsize=(12, 6))

plt.plot(traffic\_data['Daily Vehicle Count'], label='Traffic Count') plt.title('Time Series Trends of Traffic Count')

plt.xlabel('Date')

plt.ylabel('Traffic Count')

plt.legend()

plt.show()

**Average Monthly Traffic Count**

plt.figure(figsize=(12, 6))

monthly\_traffic.plot(kind='bar')

plt.title('Average Monthly Traffic Count')

plt.xlabel('Month')

plt.ylabel('Average Traffic Count')

plt.show()

**Seasonal Variations in Traffic Count (Winter vs. Summer)**

plt.figure(figsize=(12, 6))

sns.boxplot(data=[winter\_traffic['Daily Vehicle Count'], summer\_traffic['Daily Vehicle Count']], palette="coolwarm") plt.xticks([0, 1], ['Winter', 'Summer'])

plt.title('Seasonal Variations in Traffic Count (Winter vs. Summer)') plt.ylabel('Traffic Count')

plt.show()

**Year-over-Year Traffic Count**

plt.figure(figsize=(12, 6))

yearly\_traffic.plot(kind='line', marker='o')

plt.title('Year-over-Year Traffic Count')

plt.xlabel('Year')

plt.ylabel('Total Traffic Count')

plt.show()

**Histogram of Traffic Counts**

plt.figure(figsize=(12, 6))

sns.histplot(traffic\_data['Daily Vehicle Count'], bins=30, kde=True) plt.title('Histogram of Traffic Counts')

plt.xlabel('Traffic Count')

plt.ylabel('Frequency')

plt.show()

**Box Plot of Monthly Traffic Counts**

plt.figure(figsize=(12, 6))

sns.boxplot(x=traffic\_data['Month'], y=traffic\_data['Daily Vehicle Count']) plt.title('Box Plot of Monthly Traffic Counts')

plt.xlabel('Month')

plt.ylabel('Traffic Count')

plt.show()

**Scatter Plot: Traffic Count vs. Time**

plt.figure(figsize=(12, 6))

plt.scatter(traffic\_data.index, traffic\_data['Daily Vehicle Count']) plt.title('Scatter Plot: Traffic Count vs. Time')

plt.xlabel('Date')

plt.ylabel('Traffic Count')

plt.show()

**Time Series Line Chart of Monthly Traffic**

plt.figure(figsize=(12, 6))

monthly\_avg\_traffic['Daily Vehicle Count'].plot()

plt.title('Time Series Line Chart of Monthly Traffic')

plt.xlabel('Time (by month)')

plt.ylabel('Average Daily Traffic Count')

plt.show()

**Rolling Average (Moving Average) Line Chart**

plt.figure(figsize=(12, 6))

rolling\_avg\_3m.plot(label='3-Month Rolling Average')

rolling\_avg\_6m.plot(label='6-Month Rolling Average')

plt.title('Rolling Average (Moving Average) Line Chart')

plt.xlabel('Time')

plt.ylabel('Smoothed Traffic Count')

plt.legend()

plt.show()

**Seasonal Line Chart (Yearly Comparison)**

import pandas as pd

import matplotlib.pyplot as plt

# Load the data

traffic\_data = pd.read\_csv('Congestion Cleaned.csv')

# Convert the date column to datetime format

traffic\_data['Date'] = pd.to\_datetime(traffic\_data['Date'])

# Extract month and year from the date

traffic\_data['Month'] = traffic\_data['Date'].dt.month

traffic\_data['Year'] = traffic\_data['Date'].dt.year

# Group by year and month to get average traffic count for each month of each year

monthly\_traffic = traffic\_data.groupby(['Year', 'Month'])['Daily Vehicle Count'].mean().unstack(level=0)

# Plot the seasonal line chart (yearly comparison)

plt.figure(figsize=(12, 6))

for year in monthly\_traffic.columns:

plt.plot(monthly\_traffic.index, monthly\_traffic[year], label=str(year))

plt.title('Seasonal Line Chart (Yearly Comparison)')

plt.xlabel('Month')

plt.ylabel('Traffic Count')

plt.legend(title='Year')

plt.savefig('seasonal\_line\_chart\_yearly\_comparison.jpg')

print("Seasonal Line Chart (Yearly Comparison) has been saved as a jpg image.")

**Bar Chart: Average Traffic Count by Month**

import pandas as pd

import matplotlib.pyplot as plt

# Load the data

traffic\_data = pd.read\_csv('Congestion Cleaned.csv')

# Convert the date column to datetime format

traffic\_data['Date'] = pd.to\_datetime(traffic\_data['Date'])

# Extract month from the date

traffic\_data['Month'] = traffic\_data['Date'].dt.month

# Group by month to get average traffic count for each month average\_monthly\_traffic = traffic\_data.groupby('Month')['Daily Vehicle Count'].mean()

# Plot the bar chart: Average Traffic Count by Month

plt.figure(figsize=(12, 6))

average\_monthly\_traffic.plot(kind='bar')

plt.title('Average Traffic Count by Month')

plt.xlabel('Month (Jan–Dec)')

plt.ylabel('Average Daily Traffic Count')

plt.xticks(ticks=range(12), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])

plt.savefig('average\_traffic\_count\_by\_month.jpg')

print("Bar Chart: Average Traffic Count by Month has been saved as a jpg image.")

**Heatmap of Monthly Traffic Trends**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the data

traffic\_data = pd.read\_csv('Congestion Cleaned.csv')

# Convert the date column to datetime format

traffic\_data['Date'] = pd.to\_datetime(traffic\_data['Date'])

# Extract month and year from the date

traffic\_data['Month'] = traffic\_data['Date'].dt.month

traffic\_data['Year'] = traffic\_data['Date'].dt.year

# Group by year and month to get average traffic count for each month of each year

monthly\_traffic = traffic\_data.groupby(['Year', 'Month'])['Daily Vehicle Count'].mean().unstack(level=0)

# Plot the heatmap of monthly traffic trends

plt.figure(figsize=(12, 8))

sns.heatmap(monthly\_traffic, cmap='YlGnBu', annot=True, fmt=".1f") plt.title('Heatmap of Monthly Traffic Trends')

plt.xlabel('Year')

plt.ylabel('Month')

plt.savefig('heatmap\_monthly\_traffic\_trends.jpg')

print("Heatmap of Monthly Traffic Trends has been saved as a jpg image.")

**STL Decomposition of Traffic Data**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import STL

# Load the data

traffic\_data = pd.read\_csv('Congestion Cleaned.csv')

# Convert the date column to datetime format

traffic\_data['Date'] = pd.to\_datetime(traffic\_data['Date'])

# Set the date column as the index

traffic\_data.set\_index('Date', inplace=True)

# Perform STL decomposition

stl = STL(traffic\_data['Daily Vehicle Count'], seasonal=13) result = stl.fit()

# Plot the decomposed components

plt.figure(figsize=(12, 8))

plt.subplot(3, 1, 1)

plt.plot(result.trend)

plt.title('Trend Component')

plt.subplot(3, 1, 2)

plt.plot(result.seasonal)

plt.title('Seasonal Component')

plt.subplot(3, 1, 3)

plt.plot(result.resid)

plt.title('Residual Component')

plt.tight\_layout()

plt.savefig('stl\_decomposition.jpg')

print("STL decomposition has been saved as a jpg image.")

Isuri Rajapaksa Code

**import** pandas **as** pd

file\_path **=** "/Users/isurirajapaksa/Desktop/BANA 6610/group project/Congestion Cleaned.csv"

df **=** pd**.**read\_csv(file\_path)

df**.**info(), df**.**head()

***# Traffic Volume Over Time***

**import** matplotlib.pyplot **as** plt

df['Date'] **=** pd**.**to\_datetime(df['Date'])

*# Plot the Daily Vehicle Count over time*

plt**.**figure(figsize**=**(12, 6))

plt**.**plot(df['Date'], df['Daily Vehicle Count'], label**=**"Daily Vehicle Count")

plt**.**xlabel("Date")

plt**.**ylabel("Daily Vehicle Count")

plt**.**title("Traffic Volume Over Time")

plt**.**legend()

plt**.**grid(**True**)

plt**.**show()

***#Holt-Winters (Exponential Smoothing)***

**from** statsmodels.tsa.holtwinters **import** ExponentialSmoothing

*# Sort data by date*

df **=** df**.**sort\_values(by**=**"Date")

*# Fit the Holt-Winters model*

model **=** ExponentialSmoothing(df['Daily Vehicle Count'], seasonal**=**'add', seasonal\_periods**=**12, trend**=**'add')

hw\_model **=** model**.**fit()

*# Forecast for the remaining months of 2017 (assuming last data is before 2017)*

forecast\_period **=** 12 *# Forecasting for 12 months*

forecast **=** hw\_model**.**forecast(steps**=**forecast\_period)

*# Create a date range for the forecasted period*

last\_date **=** df['Date']**.**iloc[**-**1]

forecast\_dates **=** pd**.**date\_range(start**=**last\_date **+** pd**.**DateOffset(months**=**1), periods**=**forecast\_period, freq**=**'MS')

*# Plot the original data and the forecast*

plt**.**figure(figsize**=**(12, 6))

plt**.**plot(df['Date'], df['Daily Vehicle Count'], label**=**"Actual Traffic", color**=**'blue')

plt**.**plot(forecast\_dates, forecast, label**=**"Forecasted Traffic", color**=**'red', linestyle**=**'dashed')

plt**.**xlabel("Date")

plt**.**ylabel("Daily Vehicle Count")

plt**.**title("Holt-Winters Forecast for Traffic Volume in 2017")

plt**.**legend()

plt**.**grid(**True**)

plt**.**show()

***#Machine Learning Model Analysis (Random Forest)***

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.ensemble **import** RandomForestRegressor

**from** sklearn.metrics **import** mean\_absolute\_error, mean\_squared\_error, r2\_score

**import** numpy **as** np

*# Prepare the data*

df\_ml **=** df**.**copy()

*# Convert categorical 'Month Year' column to datetime and extract features*

df\_ml['Month Year'] **=** pd**.**to\_datetime(df\_ml['Month Year'])

df\_ml['Year'] **=** df\_ml['Month Year']**.**dt**.**year

df\_ml['Month'] **=** df\_ml['Month Year']**.**dt**.**month

*# Select relevant features for ML model*

features **=** ['Regular Gas Price Average', 'Midgrade Gas Price Average', 'Premium Gas Price Average',

'Monthly\_Max\_Temp', 'Monthly\_Min\_Temp', 'Monthly\_Sum\_Precipitation', 'Monthly\_Snowfall',

'Year', 'Month']

target **=** 'Daily Vehicle Count'

*# Define X (features) and y (target)*

X **=** df\_ml[features]

y **=** df\_ml[target]

*# Split data into training and test sets (80% train, 20% test)*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2, random\_state**=**42)

*# Train a Random Forest model*

rf\_model **=** RandomForestRegressor(n\_estimators**=**100, random\_state**=**42)

rf\_model**.**fit(X\_train, y\_train)

*# Make predictions*

y\_pred **=** rf\_model**.**predict(X\_test)

*# Evaluate the model*

mae **=** mean\_absolute\_error(y\_test, y\_pred)

mse **=** mean\_squared\_error(y\_test, y\_pred)

rmse **=** np**.**sqrt(mse)

r2 **=** r2\_score(y\_test, y\_pred)

*# Display model performance*

mae, rmse, r2

***#Decomposition Plot (trend, season, residual)***

*import pandas as pd*

*import matplotlib.pyplot as plt*

*import statsmodels.api as sm*

*# Load the dataset*

*file\_path = "/Users/isurirajapaksa/Desktop/BANA 6610/group project/Congestion Cleaned.csv"*

*df\_new = pd.read\_csv(file\_path)*

*# Convert Date column to datetime format*

*df\_new['Date'] = pd.to\_datetime(df\_new['Date'])*

*# Set Date as index and sort the data*

*df\_new.set\_index("Date", inplace=True)*

*df\_new = df\_new.sort\_index()*

*# Ensure the index has a monthly frequency for proper seasonal decomposition*

*df\_new.index.freq = 'MS'*

*# Perform time series decomposition*

*decomposition = sm.tsa.seasonal\_decompose(df\_new['Daily Vehicle Count'], model='additive', period=12)*

*# Plot the decomposition components*

*plt.figure(figsize=(12, 8))*

*plt.subplot(3, 1, 1)*

*plt.plot(decomposition.trend, label="Trend", color="blue")*

*plt.legend(loc='best')*

*plt.ylabel("Traffic Count")*

*plt.title("Trend Component")*

*plt.subplot(3, 1, 2)*

*plt.plot(decomposition.seasonal, label="Seasonality", color="green")*

*plt.legend(loc='best')*

*plt.ylabel("Traffic Count")*

*plt.title("Seasonal Component")*

*plt.subplot(3, 1, 3)*

*plt.plot(decomposition.resid, label="Residuals", color="red")*

*plt.legend(loc='best')*

*plt.ylabel("Traffic Count")*

*plt.title("Residual Component")*

*plt.xlabel("Time")*

*plt.tight\_layout()*

*plt.show()*

**#ARIMA**

# Load the newly uploaded dataset

file\_path = "/Users/isurirajapaksa/Desktop/BANA 6610/group project/Congestion Cleaned.csv"

df\_arima = pd.read\_csv(file\_path)

# Convert Date column to datetime format

df\_arima['Date'] = pd.to\_datetime(df\_arima['Date'])

# Set Date as index and sort the data

df\_arima.set\_index('Date', inplace=True)

df\_arima = df\_arima.sort\_index()

# Ensure monthly frequency for time series analysis

df\_arima.index.freq = 'MS'

# Fit a new ARIMA model

from statsmodels.tsa.arima.model import ARIMA

# Define ARIMA model order (p, d, q) - can be optimized further

arima\_model = ARIMA(df\_arima['Daily Vehicle Count'], order=(2,1,2))

arima\_fitted = arima\_model.fit()

# Forecast for the next 12 months

future\_dates = pd.date\_range(start=df\_arima.index[-1] + pd.DateOffset(months=1), periods=12, freq='MS')

arima\_forecast = pd.Series(arima\_fitted.forecast(steps=12), index=future\_dates)

# Plot actual vs ARIMA forecast

plt.figure(figsize=(12, 6))

plt.plot(df\_arima.index, df\_arima['Daily Vehicle Count'], label="Actual Traffic", color='blue')

plt.plot(future\_dates, arima\_forecast, label="ARIMA Forecast", color='green', linestyle='dashed')

plt.xlabel("Date")

plt.ylabel("Daily Vehicle Count")

plt.title("ARIMA Forecast for Traffic Volume")

plt.legend()

plt.grid(True)

plt.show()

**import** seaborn **as** sns

***# Scatter Plot: Traffic vs. Weather (Precipitation)***

plt**.**figure(figsize**=**(10, 6))

sns**.**scatterplot(x**=**df\_arima['Monthly\_Sum\_Precipitation'], y**=**df\_arima['Daily Vehicle Count'])

plt**.**xlabel("Monthly Precipitation (inches)")

plt**.**ylabel("Daily Vehicle Count")

plt**.**title("Traffic Volume vs. Weather Conditions (Precipitation)")

plt**.**grid(**True**)

plt**.**show()

***# Scatter Plot: Traffic vs. Weather (Temperature)***

plt**.**figure(figsize**=**(10, 6))

sns**.**scatterplot(x**=**df\_arima['Monthly\_Max\_Temp'], y**=**df\_arima['Daily Vehicle Count'])

plt**.**xlabel("Monthly Max Temperature (°F)")

plt**.**ylabel("Daily Vehicle Count")

plt**.**title("Traffic Volume vs. Weather Conditions (Temperature)")

plt**.**grid(**True**)

plt**.**show()

*# Extract day of the week from the Date index*

df\_arima['Day of Week'] **=** df\_arima**.**index**.**dayofweek *# Monday=0, Sunday=6*

day\_labels **=** ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

*# Group by day of the week and calculate average traffic count*

traffic\_by\_day **=** df\_arima**.**groupby('Day of Week')['Daily Vehicle Count']**.**mean()

*# Bar Chart: Traffic by Day of the Week*

plt**.**figure(figsize**=**(10, 6))

sns**.**barplot(x**=**day\_labels, y**=**traffic\_by\_day**.**values)

plt**.**xlabel("Day of the Week")

plt**.**ylabel("Average Daily Vehicle Count")

plt**.**title("Average Traffic Volume by Day of the Week")

plt**.**xticks(rotation**=**45)

plt**.**grid(axis**=**'y')

plt**.**show()

**AI Statement**

Our team recognizes the importance of utilizing AI as a learning tool. We were able to incorporate it into our project through sample codes and assistance with analysis and recommendations. We worked to create a chatbox through machine learning to help the travelers to pre plan and give preparation time for related authorities to plan and take action. AI was used as a tool for 60-70% of this project as a means for guiding us in our case study.