PSTAT 174 Final Project

Caleb Mazariegos mmazariegos@ucsb.edu

2023-05-23

Abstract: This project utilizes a time series approach to analyze and forecast traffic patterns at four junctions. Through exploratory data analysis, trends, seasonality, and anomalies in traffic volume are identified. Forecasting models like ARIMA and SARIMA are applied to predict future traffic volumes at each junction, considering historical data and seasonal components. Seasonality analysis reveals recurring patterns, aiding in peak hour and season identification. Correlation analysis uncovers dependencies between junctions for better traffic management. Visualizations, such as time series plots and dashboards, effectively communicate findings. The project contributes to capacity planning and resource allocation, considering factors like weather and special events. This research offers insights into traffic management and urban planning, facilitating efficient infrastructure development.

Introduction

Traffic congestion is rising in cities around the world. Contributing factors include expanding urban populations, aging infrastructure, inefficient and uncoordinated traffic signal timing and a lack of real-time data. The purpose of this project is to help understand peak traffic hours, it could also be valuable data for transportation policymakers and urban planners.This project will aim to achieve accurate traffic predictions using the box-jenkins method to find an adequate SARIMA model to forecast traffic. By leveraging historical traffic data, the project aims to develop a robust forecasting model that can provide reliable predictions of future traffic volumes.I chose this dataset because I have lived in Los Angeles most of my life, and traffic is a huge problem especially in Downtown Los Angeles. I think that analyses like this can be valuable in creating solutions to these issues. The findings from this study will contribute to improved traffic management strategies, aiding in efficient resource allocation and proactive decision-making.

Data

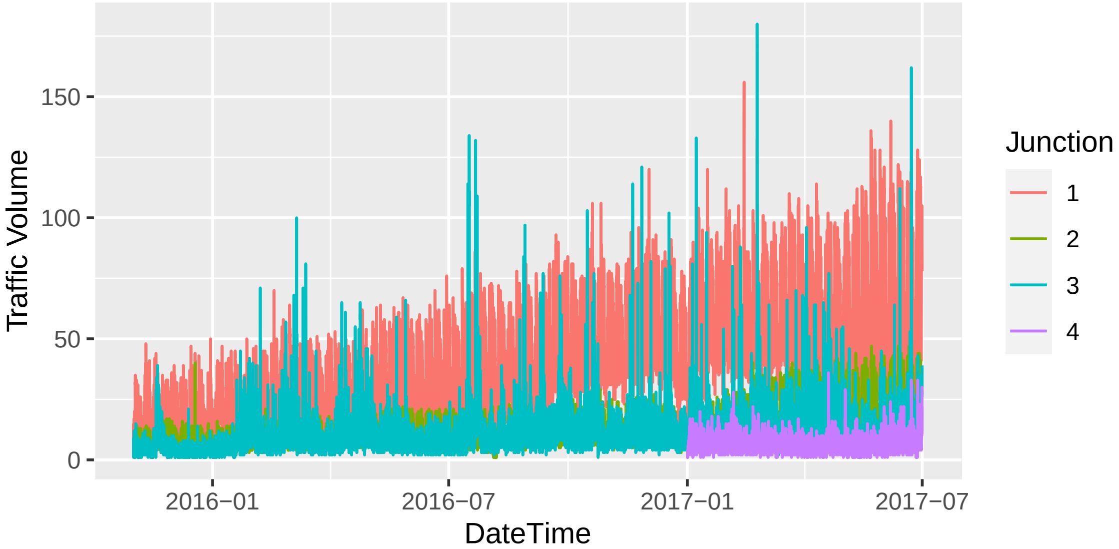
The dataset that I am using contains over 48,000 observations of the number of vehicles each hour in four different junctions. The data was collected using sensors, therefore you will see traffic data from different time periods. The data was collected from 10/31/2015 to 06/30/2017, which is a time period of 1 year and 8 months. The data was collected using sensors and was collected hourly. Each observation has its individual ID.

I will drop the ID column, as we do not need it. I also converted the DateTime column into POSIXct format to make it easier to use.

```
path <- ("/cloud/project/traffic.csv")
traffic_data <- read.csv(path)
# Convert DateTime column to POSIXct format
traffic_data$DateTime <- as.POSIXct(traffic_data$DateTime)
traffic_data$Junction <- as.factor(traffic_data$Junction)

# Selecting which variables to keep
keeps <- c("DateTime","Junction", "Vehicles")
traffic_data = traffic_data[keeps]
head(traffic_data)
```

```
##              DateTime Junction Vehicles
## 1 2015-11-01 00:00:00        1       15
## 2 2015-11-01 01:00:00        1       13
## 3 2015-11-01 02:00:00        1       10
## 4 2015-11-01 03:00:00        1        7
## 5 2015-11-01 04:00:00        1        9
## 6 2015-11-01 05:00:00        1        6
```

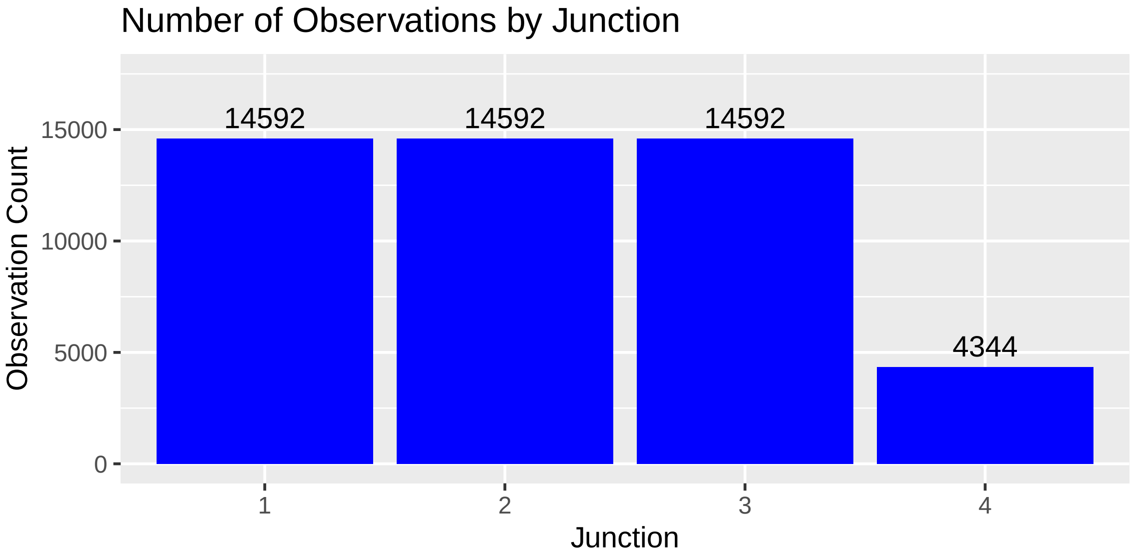
```
# Plot all junctions on one graph
ggplot(traffic_data, aes(x = DateTime, y = Vehicles, color = Junction)) +
  geom_line() +
  labs(x = "DateTime", y = "Traffic Volume") +
  scale_color_discrete(name = "Junction")
```



The graph depicts the traffic volume over time for four different junctions. Each line represents a specific junction, and the color distinguishes between them. The x-axis shows the DateTime values, while the y-axis represents the traffic volume. The graph provides a visual representation of the traffic patterns and allows for easy comparison between the junctions. Based on the graph, we can see that junction 4 did not have any observations until 01/2017. This is important as it can skew the data since it has not been collected as long as the other 3 junctions.

The bar graph below further illustrates the number of observations for each junction in the dataset. From this we can confirm that junction 4 has a reduction of approximately 70.27% from the other junctions.

```r
# Count the number of observations for each junction
junction_counts <- traffic_data %>% group_by(Junction) %>% summarise(Count = n())

# Create the bar graph with count labels
ggplot(junction_counts, aes(x = Junction, y = Count)) +
  geom_bar(stat = "identity", fill = "blue") +
  geom_text(aes(label = Count), vjust = -0.5, color = "black", size = 4) +
  labs(x = "Junction", y = "Observation Count") +
  ggtitle("Number of Observations by Junction") +
  coord_cartesian(ylim = c(0, max(junction_counts$Count) * 1.2))
```



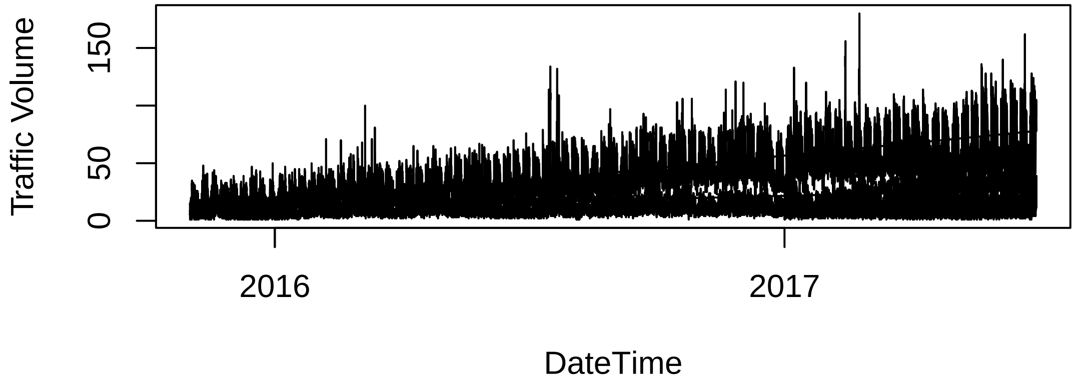
Methodology

The methodology employed in this project involves utilizing the Box-Jenkins method to identify and construct a suitable SARIMA model for traffic forecasting. I will also utilize lagged regression because there is a potential relationship between the current traffic volume and the previous values at different time lags.

Firstly, using the Box-Jenkins method will be employed to determine the appropriate SARIMA model that can effectively capture the temporal and seasonal variations in the traffic data. The chosen model is trained using historical traffic data and validated using suitable evaluation metrics. Finally, the trained SARIMA model is utilized to forecast future traffic volumes, aiding in the understanding of peak traffic hours and facilitating informed decision-making for transportation policymakers and urban planners.

I will begin by checking to make sure that my dataset is seasonal and nonstationary to obtain the most accurate model. The seasonality and nonstationarity will be visualized using a graph of the time series.

```
plot(traffic_data$DateTime, traffic_data$Vehicles, type = "l", xlab = "DateTime", ylab = "Traffic Volume
```



From the graph, and the fact that the data was collected in consistent hourly interval, I can conclude that it is seasonal. Furthermore, from the graph I can conclude that the data is stationary. Since the data is seasonal and stationary, the SARIMA model is appropriate.

SARIMA model

Preparing the data

I start by converting the data into a time series object from the Vehicles column. I set the frequency to 24 because the data was collected hourly

```
ts_data <- ts(traffic_data$Vehicles, frequency = 24)
```

Training and Test split

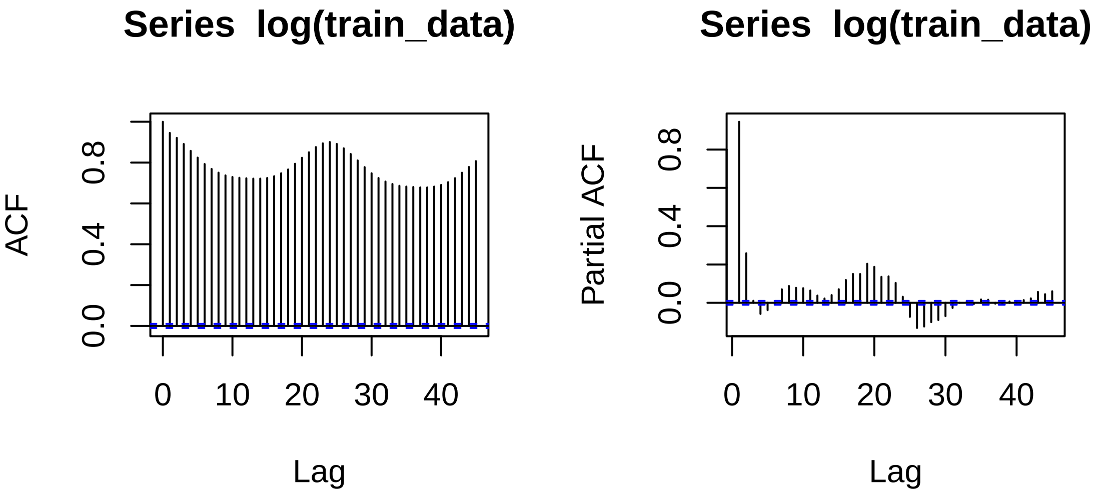
I now split the data into training and testing sets to help evaluate the performance of my model. I split the data by using 80% for training and the other 20% of the data for testing.

```
# Calculate the split index
split_index <- floor(0.8 * length(ts_data))

# Split the data
train_data <- ts_data[1:split_index]
test_data <- ts_data[(split_index + 1):length(ts_data)]
```

ACF and PACF plots

I start out by graphing the ACF and PACF plots to determine the appropriate (p,d,q)x(P,D,Q). Based on the decay seen in the ACF and the single spike seen in the PACF, the desired values are (0,0,0)x(1,0,0)

```
par(mfrow=c(1,2))
# ACF plot
acf(log(train_data))

# PACF plot
pacf(log(train_data))
```



Fitting the model

I fit the SARIMA model using the (p,d,q)x(P,D,Q) values that I found using the ACF and PACF graphs.

```
# Fit the SARIMA model
sarima_model <- arima(train_data, order = c(0, 0, 0), seasonal = c(1, 0, 0),
                xreg = NULL)
```

Generating Forecasts

Forecasting traffic vs. the actual traffic values

```
# Generate predictions
predictions <- predict(sarima_model, n.ahead = length(test_data))
```

Turning the prediction data into a time series object

```
# Time series object for the predicted values
pred_ts <- ts(predictions$pred, start = start(test_data), frequency = frequency(test_data))
```

Spectral Analysis

The next method I chose to apply to my traffic dataset is spectral analysis because it allows me to examine the frequency components present in the data and identify any dominant periodic patterns. By analyzing the spectral density, which represents the distribution of power across different frequencies, I can gain insights into the underlying cyclic behavior of the traffic data.

I did it by utilizing the Fast Fourier Transform (FFT) algorithm, which efficiently computes the spectral density. In R, I used the stats package's spec.pgram() function to perform the spectral analysis on my traffic data. This function takes a time series object as input and returns an estimate of the spectral density.

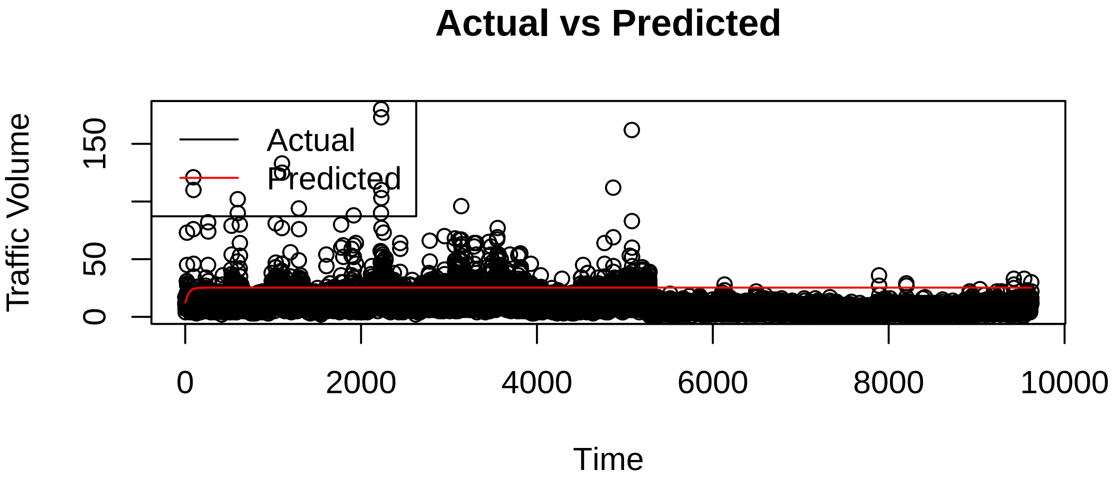
To ensure accurate results, I preprocessed my data by removing any trends or seasonality. This helps to focus the spectral analysis on the underlying cyclic patterns rather than any long-term trends. I used techniques such as differencing or seasonal differencing to achieve stationarity if necessary.

Once the data was prepared, I called the spec.pgram() function and specified the desired arguments such as the window type, the number of frequency points, and any other relevant options. The function computed the periodogram, and provided an estimate of the spectral density.

```
# Perform spectral analysis using spec.pgram()
spec <- spec.pgram(train_data, plot = FALSE)
```

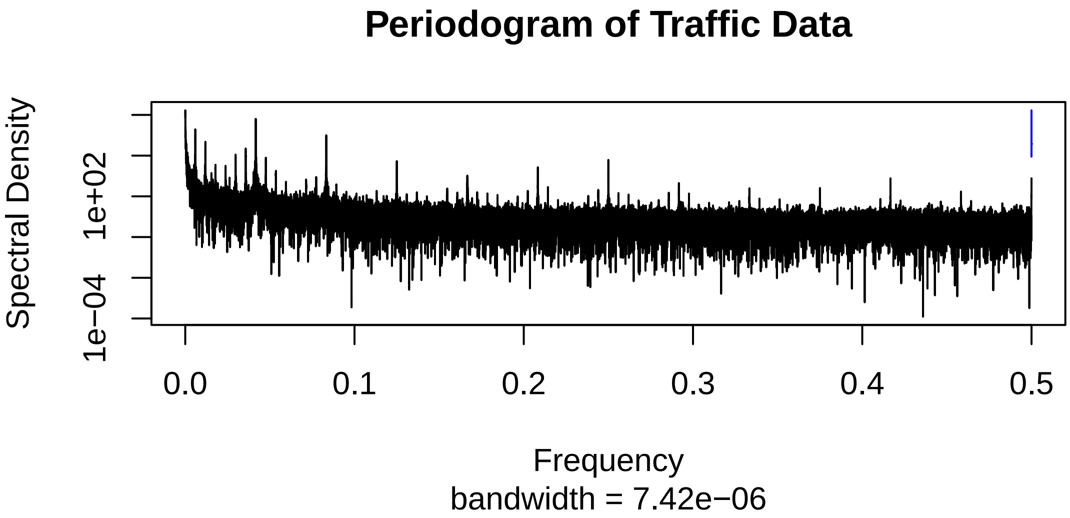
SARIMA plots

```
# Plot the actual and predicted values
plot(test_data, main = "Actual vs Predicted", ylab = "Traffic Volume", xlab = "Time")
lines(pred_ts, col = "red")
legend("topleft", legend = c("Actual", "Predicted"), col = c("black", "red"), lty = 1)
```



Spectral Analysis plots

```
# Plot the periodogram
plot(spec, main = "Periodogram of Traffic Data", xlab = "Frequency", ylab = "Spectral Density")
```


Periodogram of Traffic Data
Spectral Density
1e−04
1e+02
0.0
0.1
0.2
0.3
0.4
0.5
Frequency
bandwidth = 7.42e−06