Spatial-temporal Modelling for weekly and monthly Vehicle Theft in Downtown Los Angeles from 2010 to 2022 with STARIMA

Introduction

Vehicle theft is one of the major crimes recorded in Los Angeles. It does not only harm public safety but also causes direct financial losses, the opportunity cost to handle the crime, economic and psychological costs to the victims and more (Cozens & Love, 2015). In fact, it has the highest number among all types of crime committed in Los Angeles from 2020 to 2022 (Los Angeles - Open Data Portal, 2023). On the other hand, vehicle theft is considered an opportunistic crime, because vehicle thieves normally target vehicles at specific places, and environments and at opportune timing (Piza et. al, 2017); seasonality was also shown present in this type of crime (Lauritsen et. al, 2014). Therefore, the spatial-temporal aspect of vehicle theft incidents is worth investigating to examine and model current crime patterns, and make predictions. It is also worth mentioning that although COVID-19 has caused changes in the crime pattern in Los Angeles, for example, the containment policies have led to a significant decrease in overall crime, there is no significant effect shown for vehicle theft in Los Angeles (Campedelli et. al, 2021).

With the above explained, this paper aims to examine the spatial-temporal pattern of vehicle theft in Downtown Los Angeles, which has the highest crime rate among all L.A. neighbourhoods in 2022 (Los Angeles crime, n.d.). It hypothesises that as mentioned, the pattern of vehicle theft is not affected by COVID-19. The paper will also compare the performance in two aggregating time scales: weekly and monthly, to examine which time aggregation has better performance with STARIMA. Since aggregation can identify significant clusters quicker than at lower scales; segmentation helps understanding cyclic patterns. (Cheng & Adepeju, 2014). The space-time autoregressive integrated moving average (STARIMA) is a statistical modelling tool that captures and incorporates spatial and lag effects (Pfeifer & Deutsch, 2016); previous studies have also used the STARIMA model to analyse and predict different crime events, for example, property crime in Houston (LeSage et. al, 2009). Lastly, in the spatial aspect, whether it is suitable to aggregate the vehicle theft events into spatial units of L.A. Time’s neighbourhoods will be discussed.

Data Processing

To extract the vehicle theft events, the crime events in L.A. from 2010 to 2022 was extracted from Los Angeles - Open Data Portal (2023). It is then converted to a dataframe with time of occurrence and coordinates in wgs1984, filtered to only with vehicle theft events. The timestamp was then aggregated into temporal units of weeks and month, and performed a spatial join with the L.A. neighbourhoods (114 neighbourhoods within L.A. extracted from the L.A. Times’ neighbourhood map. It is used because it separates different communities and includes average income statistics for possible further investigation (UCLA Library, n.d.).

Data Exploration

If one type of crime is focused, it does not have the drawback that different types of crimes may not have correlations with each other, and therefore, cannot be used to forecast together. Firstly, the yearly, monthly and weekly number vehicle theft counts from 2010 to 2022 are plot. From the curve, the trend of weekly counts shows slight increase throughout the years, while the monthly and yearly counts show more fluctuations. Then, Seasonal-Trend Decomposition using Loess (STL) is used to decompose the time series into seasonal, trend, and residual. It is performed to monthly, weekly and daily aggregation, to show the differences in pattern and seasonality as the data are aggregated.

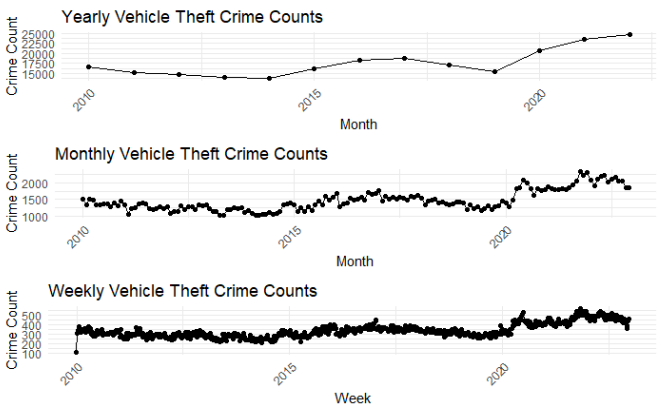

Figure 1. Yearly, Monthly, Weekly aggregated plot from 2010 to 2022

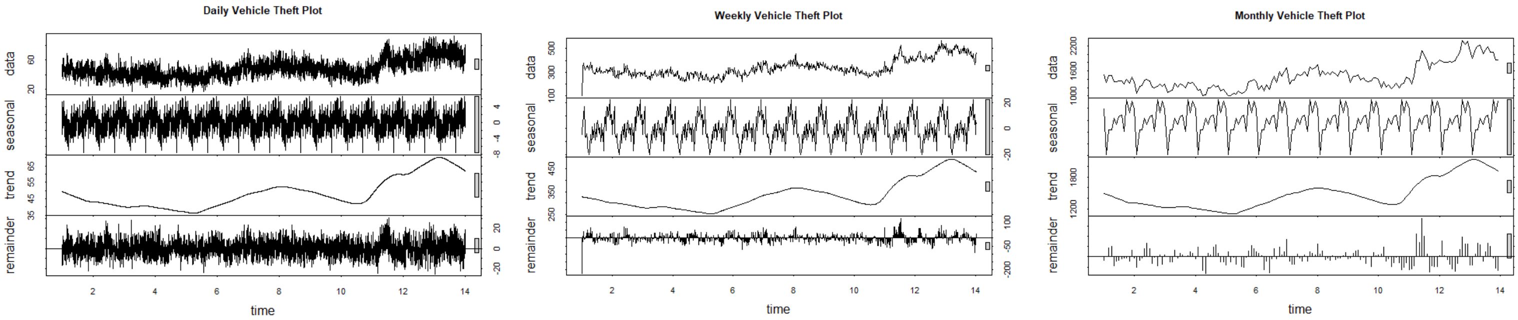

Figure 2. Daily, Monthly, Yearly STL decomposition

From the STL decomposition, all three levels of aggregation show seasonality patterns with repeating patterns - having the same pattern every year that some parts of a year have higher crime rate while other has less. On the other hand, there is an upward trend of vehicle crime throughout the years. And as the scale becomes smaller, there are more noise in the data and residual, but generally, there is no observable patterns for the residuals, indicating that underlying patterns are mostly captured. To conclude the plot, seasonality is observed but in a large time span, such as a quarter of a year.

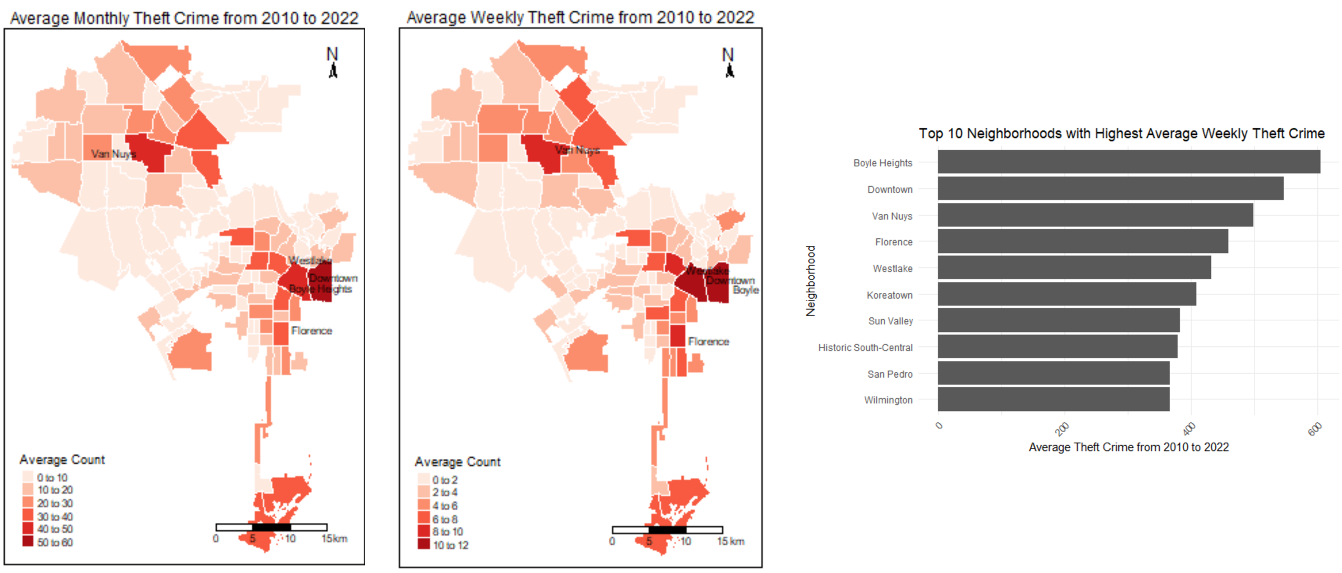

Figure 3. Average Monthly and Weekly Vehicle Theft & top 10 neighbourhoods

From figure 3, the neighbourhoods with the highest average vehicle theft counts identified, same with previous studies, the incident counts for neighbourhoods around Downtown are generally higher. In figure 4 below, the distribution of data in month and week are shown. It shows that most neighbourhoods have low counts, causing non-normal distribution. The plot in figure 3 and the frequency both shows that if we use monthly data, the dispersion of neighborhood counts will increase.

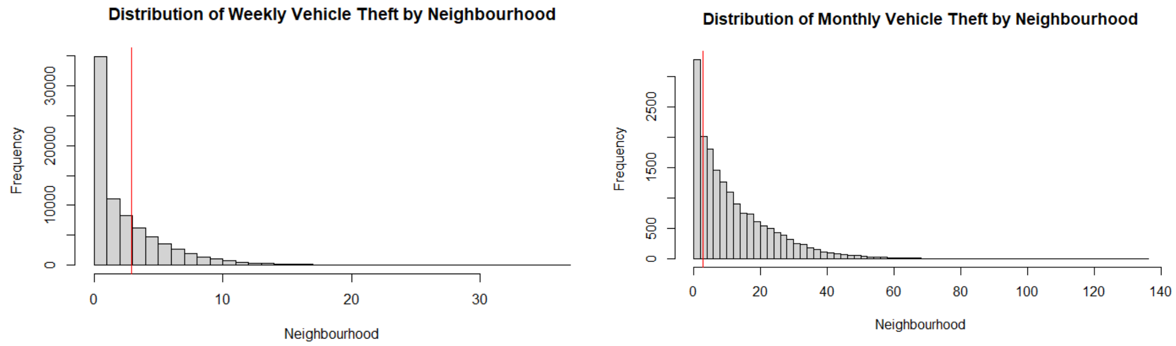

Figure 4. Distribution of Neighbourhood Vehicle Theft counts

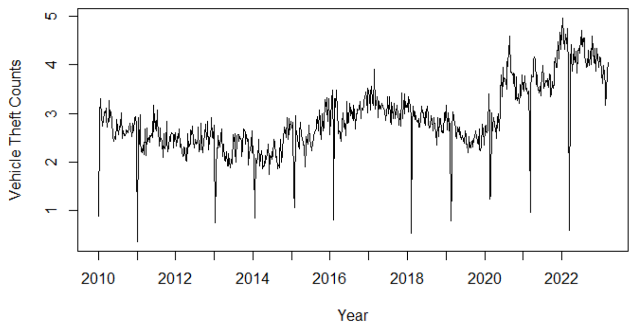

Figure 5. Time series of average weekly counts

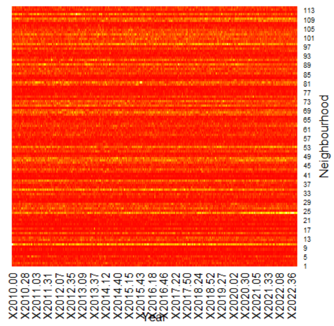

Figure 6. Heatmap for the neighbourhoods counts

In the figure 6 heatmap, the x-axis is the time lag and the y-axis is the neighbourhoods. We can see that the data shows stronger variation in spatial aspect than temporal aspect. Most neighbourhoods show consistent count in the number of weekly vehicle theft counts.

statistic standard deviate	4.9477
p-value	3.754e-07
Moran I statistic	0.301158
Expectation	-0.008850
Variance	0.0039259

Figure 7. Global Moran’s I

Local Moran's I	3.451626
Z-score	-0.083696
P-value	1.182052

Figure 8. Local Moran’s I test on Downtown

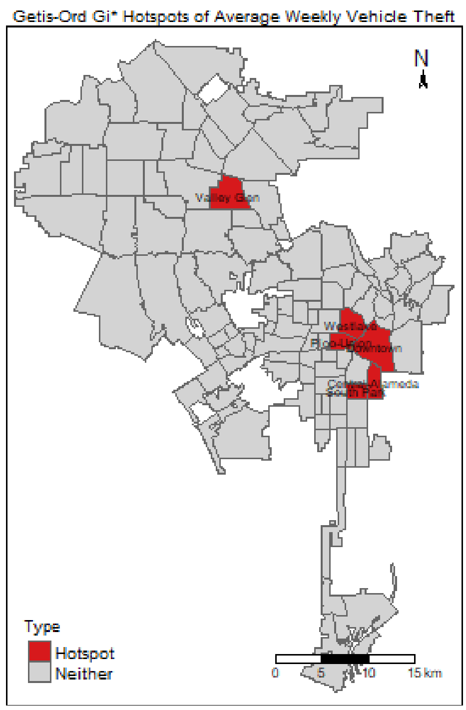

Figure 9. Getis Gi* Hotspots and Coldspot

Next we inspect the spatial dependence factor of the data. Global Moran's I (Figure 7) was used to measure the overall spatial autocorrelation in the data, local Moran's I (Figure 8) was performed for Downtown to indicates if local spatial correlation exists. Lastly, Getis-Ord Gi* statistics were used to identify high values and low values (Figure 9). The global Moran's I has a small p-value (3.754e-07), indicating that spatial autocorrelation is statistically significant but with a small positive spatial autocorrelation (0.301). Getis-Ord Gi* (both weekly and monthly) showed that the hotspots are Central-Alameda, Downtown, Pico-Union, South Park, Valley Glen and Westlake at a 95% confidence level, despite that local Moran's I showed an insignificant p-value. After identifying the spatial autocorrelation, the temporal autocorrelation and spatial-temporal characteristics of the data will be discussed.

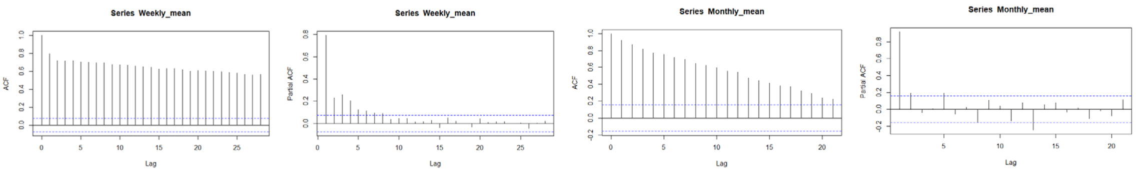

Figure 10. Weekly average ACF, Weekly average PACF, Monthly average ACF, Monthly average PACF

The above figure 10 shows the ACF and PACF plot of the mean values for weekly and monthly vehicle theft count. It is observed both weekly and monthly plot show similar patterns. The ACF slowly decreases but does not fall below the confidence level for long lags. It means that the correlations between the observations reduces as the lag is increased. It may suggest that trend or seasonality exist, in greater time scale. The PACF of weekly average shows the lagged values beyond about 6 are not significantly correlated. Figure 11 shows the result of fitting ARIMA to the weekly mean. The ARIMA fit was (0,1,2), showing no autoregressive terms, 1 lag differencing and 2 moving average terms.

ARIMA	(0,1,2)	
Coefficients	ma1	ma2
	-0.5646	-0.2570
s.e.	0.0372	0.0377
ACF1	0.01913503	
RMSE	0.3737764	
MAE	0.2271361	
sigma^2	0.1403	
log likelihood	-300.17	
AIC	606.34	

Figure 11. ARIMA result

Lastly, a 3D plot of the space-time semivariogram is shown in figure 12. The semivariance measures spatio-temporal dependence between observations. It shows that as time lag increases, the gamma consistently increases. To sum up the data exploration, it shows that the data has significant spatial autocorrelation, and temporal autocorrelation with seasonality in large scale time lag. The findings show that spatial-temporal autocorrelation in the data exists, and can be used for further space-time modelling, however, one drawback of the data is that the distribution is not normal, with many neighbourhoods having very small values. It may affects the statistical modelling.

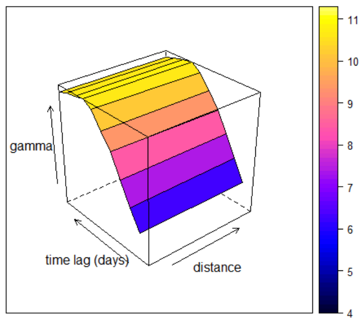

Figure 12. 3D space-time semivariogram

STARIMA Modelling to Downtown L.A.

In the part, this report focuses to model the vehicle theft in Downtown L.A. Using the spatial relationship with other neighbourhoods and ARIMA was first applied to the weekly and monthly Downtown data, and finally the STARIMA. ARIMA was first used to estimate fitting with temporal relationship of itself. The STARIMA as stated in the introduction passes in both the temporal lag and spatial lag into consideration when modelling. While fits the data that both spatial autocorrelation and temporal autocorrelation exist. In STARIMA, the Spatial

Autoregressive part captures the spatial autocorrelation, with the equation of $Y = \rho * W * Y + \varepsilon$. Y is the spatial data, W is the spatial weight matrix for the neighbourhoods, ρ is the spatial autoregressive term, ε is the error. Seondly, the AR - temporal Autoregressive captures the temporal autocorrelation. Thirdly, the MA – temporal moving average models the temporal dependence in error terms. Lastly, the I – integrating part is adopted to make the data stationary, by lag differencing. The equation of the model is $Y'_t = \rho * W * Y'_t + \sum(\varphi_i * Y'_{(t-i)}) + \sum(\theta_j * \varepsilon_{(t-j)}) + \varepsilon_t$**.** The below shows the result for ARIMA for weekly and monthly Downtown vehicle theft counts.

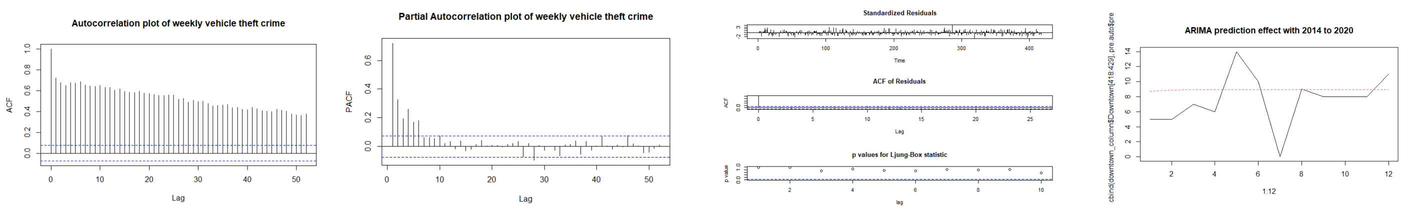

Figure 13. ACF, PACF, Residual, ARIMA forecast for weekly Downtown theft count

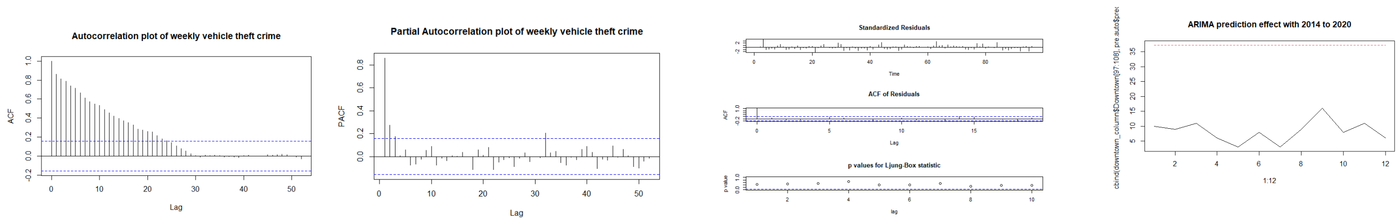

Figure 14.ACF, PACF, Residual, ARIMA forecast for monthly Downtown theft count

The ACF and PACF shows similar results as the mean data in data exploration, that the observation has strong autocorrelation with lagged ones. ARIMA(1,1,1) is fitted for the weekly data and (0,1,1) is fitted for the monthly data. The monthly fit has a lower AIC (655) than the weekly's (2165.61), along with a larger log likelihood (-325.71 than -1079.8). Other than that, the Box-Pierce tests also show no significant evidence in autocorrelation for both weekly and monthly. However, from the result of fitting the data, we can see that both weekly and monthly is not performing well, they are not able to capture the change in data. It can be reflected in the poor NRMSE which measures performance in prediction by using root mean squared error divided by the number of observations. Although lag differencing has already been performed, to eliminate seasonality, it is possible that the temporal data alone cannot be used to forecast the values in Downtown. Next, the STACF, STPACF, STARIMA will be fit.

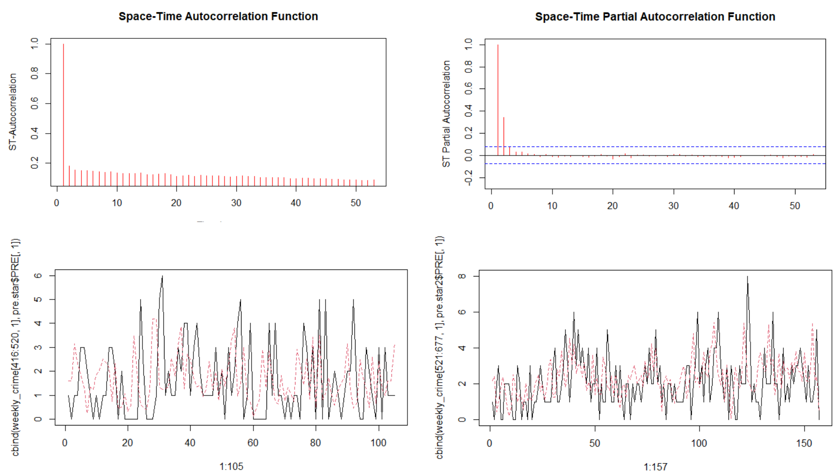

Figure 15. weekly STACF, STPACF, STARIMA forecast with 2010 to 2018, and 2010 to 2020 as training set.

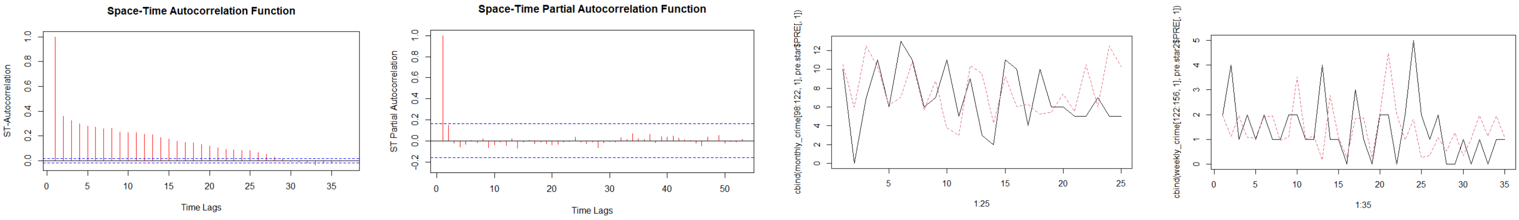

Figure 16. monthly STACF, STPACF, STARIMA forecast with 2010 to 2018, and 2010 to 2020 as training set.

From figure 15 and figure 16, we can see the effect of the adding spatial autocorrelation in the weekly and monthly Downtown data. In STACF, we see less temporal autocorrelation with the lagged observations, although it still exists in the monthly data, it is also reduced. The STPACF also shows those with lag 3 above are no longer related to the current observation in weekly data. For fitting the STARIMA model and forecast the data, (p=1, d=0 and q=2) was used for the weekly fit and (p=1, d=0 and q=3) was used for the monthly data. In the process, two sets are used to examine if the patterns of vehicle theft changed during COVID-19 and avoid it affecting the model fitting. So, firstly, 2010 to 2018 data was trained and predict 2019 to 2020 data. Then. 2010 to 2020 data was trained and predict 2021 and 2022 data. The residuals for the models are normally distributed and the model fitting of STARIMA is much better than fitting with ARIMA. From the plot of 2010 to 2018 testing data fit, we see that generally the prediction can forecast the fluctuations in the testing set, although the time is skewed, meaning that the prediction reacts earlier than the testing data. Moving on to forecasting 2020 to 2022, the weekly data performs better than the monthly data, captures the fluctuations more timely. To sum up, generally weekly data performs better than using monthly data, as it captures the fluctuations and movements better than monthly data. However, both of them have the problem that cannot predict the trend at time, they predict the data earlier than it should be.

Conclusion

To conclude, aims for the investigation are achieved. Firstly, as hypothesized, the pattern for vehicle theft was not affected by COVID-19. It is shown in the report by using the period before COVID-19 as training set first to examine the result, and then also included the COVID-19 period data in forecasting. The result of the forecasting plot shows that the forecasting during the COVID-19 period was not affected much. Moreover, when we included the COVID-19 period data as training set, the prediction showed better fit to the testing data. Secondly, weekly data in this case showed better fitting with the data than monthly data, it can be reasoned that segmented data shows the cycle of data better. However, although STARIMA fits better than ARIMA after adding the spatial autoregressive term, the temporal aspect still does not fit well. It may because of the data itself, that there are too many outliers, and the data is not normally distributed by mostly 0 and small number data. STARIMA also has limited ability to detect non-linear relationships and assumes the data is stationary. But generally, added the spatial aspect to ARIMA caused overall model to be better fitted. Lastly, the result can also be caused by the spatial aggregation. Since the used spatial grid is the 114 neighbourhoods in L.A., bias may occur and it spatial division may not fit the dataset. In this report, the data were aggregated into L.A. Time's Neighbourhoods, they are classified by communities. Further projects may develop on whether using other types of spatial aggregation would improve fitting. Furthermore, it is also worth discussing to try other models in space-time modelling such as deep learning for any non-linear relationships.

References

Campedelli, G.M., Aziani, A. & Favarin, S. (2021). Exploring the Immediate Effects of COVID-19 Containment Policies on Crime: an Empirical Analysis of the Short-Term Aftermath in Los Angeles. Am J Crim Just 46, 704–727 https://doi.org/10.1007/s12103-020-09578-6.

Cheng, T., & Adepeju, M. (2014). Modifiable temporal unit problem (MTUP) and its effect on space-time cluster detection. PLoS One, 9(6), e100465.

Cozens, P., & Love, T. (2015). A Review and Current Status of Crime Prevention through Environmental Design (CPTED). Journal of Planning Literature, 30(4), 393–412. https://doi.org/10.1177/0885412215595440.

Lauritsen L., BJS Visiting Fellow, White N. (2014). Seasonal Patterns in Criminal Victimization Trends. U.S. Department of Justice, Office of Justice Programs, Bureau of Justice Statistics. Available at: https://bjs.ojp.gov/content/pub/pdf/spcvt.pdf. (Accessed: 21 March 2023).

LeSage, James P., Pace, R. Kelley. (2009). Spatial Econometric Models. Handbook of Applied Spatial Analysis, 355–376. Available at: https://link.springer.com/chapter/10.1007/978-3-642-03647-7_18. (Accessed: 21 March 2023).

Los Angeles crime rates: Track the data to find safest neighborhoods. (n.d.). USA Today News. Available at: https://eu.usatoday.com/story/news/2023/03/20/los-angeles-crime-rates-map-safest-neighborhoods/11032761002/#:~:text=Downtown%20L.A.%27s%20crime%20rate,people%20recorded%20downtown%20last%20year. (Accessed: 21 March 2023).

Los Angeles - Open Data Portal. (2023). Available at: https://data.lacity.org/d/2nrs-mtv8/visualization. (Accessed: 21 March 2023).

Piza, E., Feng, S., Kennedy, L., Caplan, J. (2017). Place-based correlates of Motor Vehicle Theft and Recovery: Measuring spatial influence across neighbourhood context. Urban Studies, 54(13), 2998–3021. https://doi.org/10.1177/0042098016664299.

UCLA Library. (n.d.). “Neighborhood Research and Community Analysis”. https://guides.library.ucla.edu/c.php?g=180310&p=1187434. (Accessed: 21 March 2023).

Code Transcript

```
---
title: "Spatial-temporal Modelling for weekly and monthly Vehicle Theft
in Los Angeles Downtown from 2010 to 2022 with STARIMA"
---

```{r include=FALSE}
options(timeout = 1000000)
options(R_MAX_MEMORY_SIZE = "16G")
library(sparr)
library(lubridate)
library(janitor)
library(tibble)
library(readr)
```
```

````
library(tidyr)
library(dplyr)
library(rgdal)
library(sp)
library(sf)
library(stars)
library(spacetime)
library(spdep)
library(tmap)
library(tmaptools)
library(ggplot2)
library(gridExtra)
library(readr)
library(curl)
library(forecast)
library(tidyverse)
library(reshape)
library(lattice)
library(gstat)
source('starima_package.R')
```

## Data Processing

This part is mainly about aggregating the crime point data into neigh-
bourhoods, and turn the neighbourhood observations into time-series data.
If you just want to read the processed files, you can skip to the next
session.

#### Neighbourhood shapefile

```{r include=FALSE}
# LA neighborhood sf objects
neighborhood_sf <- st_read("./neighborhood_reduced/neighborhood_re-
duced.shp")
neighborhood_sf <- subset(neighborhood_sf, select=c(name, geometry))
```

#### Crime events point data

```{r include=FALSE}
# Download and Combine the crime data
crime_data <- rbind(read_csv("https://data.lacity.org/api/views/63jg-
8b9z/rows.csv?accessType=DOWNLOAD"), read_csv("https://data.lac-
ity.org/api/views/2nrs-mtv8/rows.csv?accessType=DOWNLOAD"))
crime_data <- clean_names(crime_data)
crime_data <- subset(crime_data, select= -c(date_rptd, area, area_name,
rpt_dist_no, part_1_2, crm_cd, mocodes, premis_cd, weapon_desc,
weapon_used_cd, status, status_desc, crm_cd_1, crm_cd_2, crm_cd_3,
crm_cd_4, location, cross_street, time_occ, vict_age, vict_sex, vict_de-
scent, premis_desc))

# Add a time stamp column, and remove the data before 2010 and after 2022
crime_data$timestamp <- as.POSIXct(crime_data$date_occ, format =
"%m/%d/%Y %I:%M:%S %p", tz = "America/Los_Angeles")
crime_data <- crime_data %>% filter(timestamp <= as.POSIXct("2022-12-31
23:59:59", tz = "America/Los_Angeles"))
````

````
crime_data <- crime_data %>% filter(timestamp >= as.POSIXct("2010-01-01
00:00:00", tz = "America/Los_Angeles"))
```

#### Filtering Vehicle Theft Events crimes

```{r include=FALSE}
# Filter crime events with only "VEHICLE - STOLEN"
crime_data <- crime_data %>%
  filter(grepl("VEHICLE - STOLEN", crm_cd_desc, fixed = FALSE))

# Export the crime_data to a CSV file for quicker loading
write.csv(crime_data, "./crime_data/crime_data.csv")

crime_data <- read.csv("./crime_data/crime_data.csv")
```

```{r include=FALSE}
# Convert the data into an sf object
coordinates(crime_data) <- ~lon + lat
proj4string(crime_data) <- CRS("+proj=longlat +datum=WGS84")
crime_sf <- crime_data %>% st_as_sf(coords = c("lon", "lat"), crs = 4326)
```

#### Spatial Joining the point events and the neighbourhood shapefile

```{r include=FALSE}
neighborhood_sf <- st_transform(neighborhood_sf, crs = st_crs(crime_sf))
# Spatial joining the tracts and the crime events
crime_neigh <- st_join(neighborhood_sf, crime_sf)
st_write(crime_neigh, "./crime_neigh/crime_neigh.shp")
```

```{r include=FALSE}
# Read the crime_data and crime_neigh (combined data)
crime_data <- read.csv("./crime_data/crime_data.csv")

crime_neigh <- st_read("./crime_neigh/crime_neigh.shp")
crime_neigh <- subset(crime_neigh, select = -c(X_1, X, dr_no, crm_cd_))
```

#### Aggregate Vehicle Theft Events by neighbourhood by each week

```{r include=FALSE}
crime_neigh$date <- as.Date(crime_neigh$timstmp)
crime_neigh$week <- format(crime_neigh$date, format = "%Y-%U")

# Group by name (crime_neigh) and week, and calculate the crime counts
crime_counts_week <- crime_neigh %>%
  group_by(name, week) %>%
  summarise(crime_count = n()) %>%
  ungroup()

# Pivot the data to have census tracts as rows and weeks as columns
crime_counts_week$week <- as.character(crime_counts_week$week)
crime_counts_week <- as.data.frame(crime_counts_week)
````

```
# Pivot the data to have census tracts as rows and weeks as columns
crime_neigh_week <- crime_counts_week %>%
  spread(key = week, value = crime_count)

crime_neigh_week[is.na(crime_neigh_week)] <- 0

st_write(crime_neigh_week, "./crime_neigh_week/crime_neigh_week.shp")
```

```{r include=FALSE}
crime_neigh_week <- st_read("./crime_neigh_week/crime_neigh_week.shp")
```

Aggregate Vehicle Theft Events by neighbourhood by each month

```{r include=FALSE}
crime_neigh$date <- as.Date(crime_neigh$timstmp)
crime_neigh$month <- format(crime_neigh$date, format = "%Y-%m")

# Group by name (crime_neigh) and week, and calculate the crime counts
crime_counts_month <- crime_neigh %>%
  group_by(name, month) %>%
  summarise(crime_count = n()) %>%
  ungroup()

crime_counts_month$month <- as.character(crime_counts_month$month)
crime_counts_month <- as.data.frame(crime_counts_month)

# Pivot the data to have census tracts as rows and weeks as columns
crime_neigh_month <- crime_counts_month %>%
  spread(key = month, value = crime_count)

crime_neigh_month[is.na(crime_neigh_month)] <- 0
crime_neigh_month <- crime_neigh_month[, -ncol(crime_neigh_month)]

st_write(crime_neigh_month, "./crime_neigh_month/crime_neigh_month.shp")
```

```{r include=FALSE}
crime_neigh_month <- st_read("./crime_neigh_month/crime_neigh_month.shp")
```

Data Exploration

```{r include=FALSE}
crime_data <- read.csv("./crime_data/crime_data.csv")
crime_neigh_month <- st_read("./crime_neigh_month/crime_neigh_month.shp")
crime_neigh_week <- st_read("./crime_neigh_week/crime_neigh_week.shp")
```

Detecting trend, seasonality, residual with decompsition

```{r}
# Convert your crime data into a time series object
crime_data$date <- as.Date(crime_data$timestamp)
crime_data$week <- floor_date(crime_data$date, unit = "week")
crime_data$month <- floor_date(crime_data$date, unit = "month")
```

````
crime_data$year <- floor_date(crime_data$date, unit = "year")

# Aggregating crime counts into daily, weekly and monthly
daily_crime <- aggregate(crime_data$date, by = list(crime_data$date), FUN
= length)
weekly_crime <- crime_data %>%
  group_by(week) %>%
  summarise(crime_count = n())
monthly_crime <- crime_data %>%
  group_by(month) %>%
  summarise(crime_count = n())
yearly_crime <- crime_data %>%
  group_by(year) %>%
  summarise(crime_count = n())

# Convert to a time series object
t_daily <- ts(daily_crime$x, frequency = 365)
t_weekly <- ts(weekly_crime$crime_count, frequency = 52)
t_monthly <- ts(monthly_crime$crime_count, frequency = 12)

# Plotting the trend, seasonal, and residual in daily, weekly, monthly
par(mfrow = c(4, 1), mar = c(2, 4, 2, 1))
plot(stl(t_daily, s.window = "periodic"), main = "Daily Vehicle Theft
Plot")
plot(stl(t_weekly, s.window = "periodic"), main = "Weekly Vehicle Theft
Plot")
plot(stl(t_monthly, s.window = "periodic"), main = "Monthly Vehicle Theft
Plot")
```

#### Plot Theft Crimes by Year, Month and Week

```{r}
year_ggplot <- ggplot(yearly_crime, aes(x = year, y = crime_count)) +
  geom_line() +
  geom_point() +
  theme_minimal() +
  labs(title = "Yearly Vehicle Theft Crime Counts",
       x = "Month",
       y = "Crime Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

month_ggplot <- ggplot(monthly_crime, aes(x = month, y = crime_count)) +
  geom_line() +
  geom_point() +
  theme_minimal() +
  labs(title = "Monthly Vehicle Theft Crime Counts",
       x = "Month",
       y = "Crime Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

week_ggplot <- ggplot(weekly_crime, aes(x = week, y = crime_count)) +
  geom_line() +
  geom_point() +
  theme_minimal() +
  labs(title = "Weekly Vehicle Theft Crime Counts",
       x = "Week",
````

```
      y = "Crime Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

grid.arrange(year_ggplot, month_ggplot, week_ggplot, nrow = 3)
```

```
#### Plot the average Vehicle Theft by neighbourhood at different time span

Yearly
```{r}
crime_neigh_month <- as.data.frame(crime_neigh_month)
crime_neigh_month$total_crime <- rowSums(crime_neigh_month[, 2:156])
crime_neigh_month$average_crime <- crime_neigh_month$total_crime / 13

neigh_average_crime <- st_as_sf(crime_neigh_month) %>%
  select(name, geometry, average_crime)

top5_neigh <- neigh_average_crime %>%
  top_n(5, wt = average_crime) %>%
  pull(name)

neigh_average_crime$label <- ifelse(neigh_average_crime$name %in%
top5_neigh, as.character(neigh_average_crime$name), "")

tm_shape(neigh_average_crime) +
  tm_polygons(
    col = "average_crime",
    palette = "Reds",
    border.col = "white",
    lwd = 0.05,
    title = "Average Count",
    text.size = 10
  ) +
  tm_text("label", size = 0.5, col = "black", auto.placement = TRUE) +
  tm_layout(
    main.title = "Average Yearly Theft Crime from 2010 to 2022",
    main.title.position = c("center", "top"),
    main.title.size = 0.8,
    legend.position = c("left", "bottom"),
    legend.title.size = 0.65,
    legend.text.size = 0.5,
  ) +
  tm_scale_bar(position = c("right", "bottom"),
               text.size = 0.5) +
  tm_compass(position = c("right", "top"),
             size = 0.7)
```
```

```
#### List the top 10 neighbourhoods with the highest average weekly count of Theft Crime

```{r}
# Get the top 10 neighbourhoods
ranked_neigh_with_crime <- neigh_average_crime[order(-neigh_aver-
age_crime$average_crime),]
ranked_neigh_with_crime <- ranked_neigh_with_crime[1:10,]
```
```

````
# Plot
ggplot(ranked_neigh_with_crime, aes(x = reorder(name, average_crime), y =
average_crime)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 10 Neighborhoods with Highest Average Weekly Theft
Crime",
       x = "Neighborhood",
       y = "Average Yearly Crime from 2010 to 2022") +
  coord_flip()
```

Monthly
```{r}
crime_neigh_month <- as.data.frame(crime_neigh_month)
crime_neigh_month$total_crime <- rowSums(crime_neigh_month[, 2:156])
crime_neigh_month$average_crime <- rowMeans(crime_neigh_month[, 2:156])

neigh_average_crime <- st_as_sf(crime_neigh_month) %>%
  select(name, geometry, average_crime)

top5_neigh <- neigh_average_crime %>%
  top_n(5, wt = average_crime) %>%
  pull(name)

neigh_average_crime$label <- ifelse(neigh_average_crime$name %in%
top5_neigh, as.character(neigh_average_crime$name), "")

tm_shape(neigh_average_crime) +
  tm_polygons(
    col = "average_crime",
    palette = "Reds",
    border.col = "white",
    lwd = 0.05,
    title = "Average Count",
    text.size = 10
  ) +
  tm_text("label", size = 0.5, col = "black", auto.placement = TRUE) +
  tm_layout(
    main.title = "Average Monthly Theft Crime from 2010 to 2022",
    main.title.position = c("center", "top"),
    main.title.size = 0.76,
    legend.position = c("left", "bottom"),
    legend.title.size = 0.65,
    legend.text.size = 0.5,
  ) +
  tm_scale_bar(position = c("right", "bottom"),
               text.size = 0.5) +
  tm_compass(position = c("right", "top"),
             size = 0.7)
```
#### List the top 10 neighbourhoods with the highest average weekly count
of Theft Crime

```{r}
# Get the top 10 neighbourhoods
````

````
ranked_neigh_with_crime <- neigh_average_crime[order(-neigh_aver-
age_crime$average_crime),]
ranked_neigh_with_crime <- ranked_neigh_with_crime[1:10,]

# Plot
ggplot(ranked_neigh_with_crime, aes(x = reorder(name, average_crime), y =
average_crime)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 10 Neighborhoods with Highest Average Monthly Theft
Crime",
       x = "Neighborhood",
       y = "Average Theft Crime from 2010 to 2022") +
  coord_flip()
```

Weekly
```{r}
crime_neigh_week <- as.data.frame(crime_neigh_week)
crime_neigh_week$average_crime <- rowMeans(crime_neigh_week[, 2:690])

neigh_average_crime <- st_as_sf(crime_neigh_week) %>%
  select(name, geometry, average_crime)
crime_neigh_week <- st_read("./crime_neigh_week/crime_neigh_week.shp")

top5_neigh <- neigh_average_crime %>% top_n(5, wt = average_crime) %>%
pull(name)

neigh_average_crime$label <- ifelse(neigh_average_crime$name %in%
top5_neigh, as.character(neigh_average_crime$name), "")

tm_shape(neigh_average_crime) +
  tm_polygons(
    col = "average_crime",
    palette = "Reds",
    border.col = "white",
    lwd = 0.05,
    title = "Average Count",
    text.size = 10
  ) +
  tm_text("label", size = 0.5, col = "black", auto.placement = TRUE) +
  tm_layout(
    main.title = "Average Weekly Theft Crime from 2010 to 2022",
    main.title.position = c("center", "top"),
    main.title.size = 0.78,
    legend.position = c("left", "bottom"),
    legend.title.size = 0.65,
    legend.text.size = 0.5,
  ) +
  tm_scale_bar(position = c("right", "bottom"),
               text.size = 0.5) +
  tm_compass(position = c("right", "top"),
             size = 0.7)
```
````
````

```
#### List the top 10 neighbourhoods with the highest average weekly count
of Theft Crime

```{r}
# Get the top 10 neighbourhoods
ranked_neigh_with_crime <- neigh_average_crime[order(-neigh_aver-
age_crime$average_crime),]
ranked_neigh_with_crime <- ranked_neigh_with_crime[1:10,]

# Plot
ggplot(ranked_neigh_with_crime, aes(x = reorder(name, average_crime), y =
average_crime)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 10 Neighborhoods with Highest Average Weekly Theft
Crime",
       x = "Neighborhood",
       y = "Average Theft Crime from 2010 to 2022") +
  coord_flip()
```

Plotting the distribution

```{r}
crime_neigh_week_df <- as.data.frame(crime_neigh_week)
week_matrix <- subset(crime_neigh_week_df, select=-c(name, geometry)) %>%
data.matrix()


mean_week = mean(week_matrix)
mean_week

sdev = sd(week_matrix)
sdev

hist(week_matrix, breaks = 50, xlab = "Neighbourhood", main = "Distribu-
tion of Weekly Vehicle Theft by Neighbourhood")
abline(v = mean_week, col = "red")
```

```{r}
crime_neigh_month_df <- as.data.frame(crime_neigh_month)
month_matrix <- subset(crime_neigh_month_df, select=-c(name, geome-
try)) %>% data.matrix()
month_matrix <- month_matrix[, -c((ncol(month_matrix)-1):ncol(month_ma-
trix))]

mean_month = mean(month_matrix)
mean_month

sdev = sd(month_matrix)
sdev

hist(month_matrix, breaks = 50, xlab = "Neighbourhood", main = "Distribu-
tion of Monthly Vehicle Theft by Neighbourhood")
abline(v = mean_week, col = "red")
```
```

````
```

```{r}
plot(colMeans(week_matrix), xlab = "Year", ylab = "Vehicle Theft Counts",
type = "l", xaxt = "n")
axis(1, at = seq(1, as.numeric(difftime(as.Date("2022-12-31"),
as.Date("2010-01-01"), units = "weeks")), by = (as.numeric(dif-
ftime(as.Date("2022-12-31"), as.Date("2010-01-01"), units = "weeks")) /
13)), , main = "Trend of Weekly Vehicle Theft from 2010 to 2022", labels
= seq(2010, 2022, 1))
```

#### Heatmap for all neighbourhoods from 2010 to 2022

```{r}
heatmap(week_matrix,Rowv=NA,Colv=NA, col=heat.colors(256),scale="column",
margins=c(5,3),xlab="Year",ylab="Neighbourhood", cexCol=1.1,y.scale.com-
ponents.subticks(n=10))
```

#### Spatial Autocorrelation

```{r}
crime_neigh_week <- as.data.frame(crime_neigh_week)
crime_neigh_week$total_crime <- rowSums(crime_neigh_week[, c(2:690)])
crime_neigh_week$average_crime <- crime_neigh_week$total_crime / 689
crime_neigh_week <- st_as_sf(crime_neigh_week)

nb <- poly2nb(crime_neigh_week, queen = TRUE, snap = 0.2)
wm <- nb2listw(nb, style = "W")

average_weekly_crime_neighbourhood <- crime_neigh_week$average_crime
moran_result <- moran.test(average_weekly_crime_neighbourhood, wm)
print(moran_result)
```

#### Local Moran's I

Downtown
```{r}
# Find the index of the "Downtown" neighborhood
id <- which(crime_neigh_week$name == "Downtown")
local_moran <- localmoran(crime_neigh_week$average_crime, wm)

# Extract the Local Moran's I statistic, Z-score, and p-value for the
"Downtown" neighborhood
moran <- local_moran[id, 1]
zscore <- local_moran[id, 2]
pvalue <- local_moran[id, 3]

# Display the results
cat("Downtown Local Moran's I: ", moran, "\n")
cat("Downtown Z-score: ", zscore, "\n")
cat("Downtown P-value: ", pvalue)
```

Boyle Heights
````
````

````
```{r}
# Find the index of the "Boyle Heights" neighborhood
id <- which(crime_neigh_week$name == "Boyle Heights")
local_moran <- localmoran(crime_neigh_week$average_crime, wm)

# Extract the Local Moran's I statistic, Z-score, and p-value for the
"Downtown" neighborhood
moran <- local_moran[id, 1]
zscore <- local_moran[id, 2]
pvalue <- local_moran[id, 3]

# Display the results
cat("Boyle Heights Local Moran's I: ", moran, "\n")
cat("Boyle Heights Z-score: ", zscore, "\n")
cat("Boyle Heights P-value: ", pvalue)
```

#### Local Moran's I Hotspot Coldspot

```{r}
lm <- localmoran(crime_neigh_week$average_crime, wm)

hotspots <- which(lm[[2]] >= 1.96)
coldspots <- which(lm[[2]] <= -1.96)

hotspot_names <- crime_neigh_week$name[hotspots]
coldspot_names <- crime_neigh_week$name[coldspots]

crime_neigh_week$spot_type <- "Neither"
crime_neigh_week$spot_type[hotspots] <- "Hotspot"
crime_neigh_week$spot_type[coldspots] <- "Coldspot"

tm_shape(crime_neigh_week) +
  tm_fill(col = "spot_type",
          palette = "Reds",
          title = "Spot Type",
          legend.show = TRUE) +
  tm_borders() +
  tm_layout(
            main.title = "Local Moran's I Hotspots and Coldspots of Aver-
age Weekly Vehicle Theft",
            main.title.position = c("center", "top"),
            main.title.size = 0.7,
            legend.position = c("right", "bottom"),
            legend.title.size = 0.65,
            legend.text.size = 0.65,
            legend.outside = TRUE ) +
  tm_scale_bar(position = c("left", "bottom"),
               text.size = 0.5) +
  tm_compass(position = c("right", "top"),
             size = 0.7)

```

#### Getis-Ord Gi* statistic Hotspot Coldspot - Weekly

"Central-Alameda"
````

```
"Downtown"
"Pico-Union"
"South Park"
"Valley Glen" "Westlake"

```{r}
gi_ord <- localG(crime_neigh_week$average_crime, wm)
crit_value <- qnorm(0.975, mean(gi_ord), sd(gi_ord))

hotspot_indices_gi <- which(gi_ord >= crit_value)
coldspot_indices_gi <- which(gi_ord <= -crit_value)

hotspot_names_gi <- crime_neigh_week$name[hotspot_indices_gi]
coldspot_names_gi <- crime_neigh_week$name[coldspot_indices_gi]

crime_neigh_week$spot_type_gi <- "Neither"
crime_neigh_week$spot_type_gi[hotspot_indices_gi] <- "Hotspot"
crime_neigh_week$spot_type_gi[coldspot_indices_gi] <- "Coldspot"
crime_neigh_week$hotspot_label <- NA
crime_neigh_week$hotspot_label[hotspot_indices_gi] <-
crime_neigh_week$name[hotspot_indices_gi]

print(crime_neigh_week$hotspot_label[hotspot_indices_gi])

tm_shape(crime_neigh_week) +
  tm_fill(col = "spot_type_gi",
          palette = c("Neither" = "#D3D3D3", "Coldspot" = "#2C7BB6",
"Hotspot" = "#D7191C"),
          title = "Type",
          border.col = "white",
          lwd = 0.1,
          legend.show = TRUE) +
  tm_borders() +
  tm_text("hotspot_label", size = 0.45, col = "black") +
  tm_layout(
    main.title = "Getis-Ord Gi* Hotspots of Average Weekly Vehicle
Theft",
    main.title.position = c("center", "top"),
    main.title.size = 0.6,
    legend.position = c("left", "bottom"),
    legend.title.size = 0.65,
    legend.text.size = 0.65,
  ) +
  tm_view(bbox = st_bbox(crime_neigh_week)) +
  tm_scale_bar(position = c("right", "bottom"),
               text.size = 0.5) +
  tm_compass(position = c("right", "top"),
             size = 0.7)
```

Getis-Ord Gi* statistic Hotspot Coldspot - Monthly

"Central-Alameda"
"Downtown"
"Pico-Union"
"South Park"
"Valley Glen"
```

```
"Westlake"

```{r}
gi_ord <- localG(crime_neigh_month$average_crime, wm)
crit_value <- qnorm(0.975, mean(gi_ord), sd(gi_ord))

hotspot_indices_gi <- which(gi_ord >= crit_value)
coldspot_indices_gi <- which(gi_ord <= -crit_value)

hotspot_names_gi <- crime_neigh_month$name[hotspot_indices_gi]
coldspot_names_gi <- crime_neigh_month$name[coldspot_indices_gi]

crime_neigh_month$spot_type_gi <- "Neither"
crime_neigh_month$spot_type_gi[hotspot_indices_gi] <- "Hotspot"
crime_neigh_month$spot_type_gi[coldspot_indices_gi] <- "Coldspot"
crime_neigh_month$hotspot_label <- NA
crime_neigh_month$hotspot_label[hotspot_indices_gi] <-
crime_neigh_month$name[hotspot_indices_gi]

crime_neigh_month <- st_as_sf(crime_neigh_month)

print(crime_neigh_month$hotspot_label[hotspot_indices_gi])

tm_shape(crime_neigh_month) +
  tm_fill(col = "spot_type_gi",
          palette = c("Neither" = "#D3D3D3", "Coldspot" = "#2C7BB6",
"Hotspot" = "#D7191C"),
          title = "Type",
          border.col = "white",
          lwd = 0.1,
          legend.show = TRUE) +
  tm_borders() +
  tm_text("hotspot_label", size = 0.45, col = "black") +
  tm_layout(
    main.title = "Getis-Ord Gi* Hotspots of Average Weekly Vehicle
Theft",
    main.title.position = c("center", "top"),
    main.title.size = 0.6,
    legend.position = c("left", "bottom"),
    legend.title.size = 0.65,
    legend.text.size = 0.65,
  ) +
  tm_view(bbox = st_bbox(crime_neigh_week)) +
  tm_scale_bar(position = c("right", "bottom"),
               text.size = 0.5) +
  tm_compass(position = c("right", "top"),
             size = 0.7)
```

Weekly mean - ACF Plot

```{r}
Weekly_mean <- colMeans(as.data.frame(crime_neigh_week)[,2:690])
acf(Weekly_mean)
```

Monthly mean - ACF Plot
```

````
```{r}
Monthly_mean <- colMeans(as.data.frame(crime_neigh_month)[,2:156])
acf(Monthly_mean)
```

#### Weekly mean - PACF Plot

```{r}
pacf(Weekly_mean)
```

#### Monthly mean - PACF Plot

```{r}
pacf(Monthly_mean)
```

#### Weekly-mean Auto-ARIMA

```{r}
# Use auto.arima() function to find the best ARIMA model
best_model <- auto.arima(Weekly_mean, stepwise = FALSE, approximation =
FALSE)

# Display the best ARIMA model
summary(best_model)

# Residuals' ACF plot
ggAcf(residuals(best_model)) + ggtitle("Residuals' Autocorrelation Func-
tion (ACF) Plot") + theme(plot.title = element_text(hjust = 0.5))

# Residuals' PACF plot
ggPacf(residuals(best_model)) + ggtitle("Residuals' Partial Autocorrela-
tion Function (PACF) Plot") + theme(plot.title = element_text(hjust =
0.5))

# Forecast the next 12 time periods
future_forecast <- forecast(best_model, h = 12)

# Plot the forecast
autoplot(future_forecast) + ggtitle("ARIMA Model Forecast") +
theme(plot.title = element_text(hjust = 0.5))

```

#### Downtown neighbourhood - Weekly

```{r}
# Transpose the DOWNTOWN row
downtown_row <- crime_neigh_week[crime_neigh_week$name == "Downtown", ]
downtown_row <- as.data.frame(downtown_row)
downtown_row <- downtown_row[, -c(ncol(downtown_row)-1, ncol(down-
town_row))]
downtown_row <- as.data.frame(t(downtown_row))
colnames(downtown_row) <- as.character(downtown_row[1, ])
downtown_column <- downtown_row[-1, , drop = FALSE]
````
````

````
downtown_column <- slice(downtown_column, 1:(nrow(downtown_column) - 2))
downtown_column <- downtown_column[1:(nrow(downtown_column) - 2), ]
downtown_column <- as.data.frame(downtown_column)
downtown_column <- gather(downtown_column, key = "Date", value = "Down-
town")
downtown_column <- subset(downtown_column, select = -c(Date))

plot(downtown_column$Downtown, ylab="Weekly Counts", xlab="Time in
Weeks", type="l", main="Weekly vehicle theft crime from 2010 to 2022")
```

The dependence between consecutive observations is linear.

```{r}
str(downtown_column)
downtown_column$Downtown <- as.numeric(downtown_column$Downtown)
lag.plot(downtown_column$Downtown, lags=3, do.lines=FALSE)
```

#### ACF

it indicates that there is significant positive autocorrelation in the
data. This means that there is a strong relationship between the current
observation and the previous observations in the time series, and this
relationship can be used to make predictions about future values of the
time series.

There is no strong cyclic pattern in the autocorrelation for the Downtown
weekly plot.

```{r}
acf(downtown_column$Downtown, lag.max=52, xlab="Lag", ylab="ACF",
main="Autocorrelation plot of weekly vehicle theft crime")

```

After two difference, lag autocorrelation is reduced

```{r}
downtown_column.diff.Downtown <- diff(downtown_column$Downtown, lag=52,
differences=2)

acf(downtown_column.diff.Downtown, lag.max=52, xlab="Lag", ylab="ACF",
main="Autocorrelation plot of weekly vehicle theft crime")
```

#### PACF

Following shows the partial autocorrelation plot of the monthly average
temperature in East Anglia.

```{r}
pacf(downtown_column$Downtown, lag.max=52,xlab="Lag",ylab="PACF",main=
"Partial Autocorrelation plot of weekly vehicle theft crime")
```

#### ARIMA and Seasonality
````
````

````
If the ACF decays to zero and the first 52 lags are above the confidence
level, while the PACF cuts off after lag 4, it suggests that the data
might have a seasonal component along with an autoregressive (AR) compo-
nent.

```{r}
auto_arima <- auto.arima(downtown_column$Downtown[1:417], seasonal =
TRUE)
print(auto_arima)
```

#### NRMSE

```{r}
NRMSE_fit <- NRMSE(res=auto_arima$residuals, obs=downtown_column$Down-
town[1:417])
print(NRMSE_fit)
```

#### Diagnostic Checking

```{r}
tsdiag(auto_arima)
```

In general, if the p-value is less than a predefined significance level
(e.g., 0.05), we reject the null hypothesis and conclude that there is
evidence of autocorrelation in the residuals. However, in your case, the
p-value is 0.9723, which is much greater than 0.05. This means that there
is no evidence to reject the null hypothesis, so we can't conclude that
there is autocorrelation in the residuals of the auto_arima model. This
suggests that the model has captured the temporal dependencies in the
data reasonably well.

```{r}
Box.test(auto_arima$residuals,lag=1)
Box.test(auto_arima$residuals,lag=2)
Box.test(auto_arima$residuals,lag=3)
```

```{r}
pre.auto <-predict(auto_arima, n.ahead=12)
print(pre.auto)

length(downtown_column$Downtown[418:429])
length(pre.auto$pred)
```

```{r}
matplot(1:12,cbind(downtown_column$Down-
town[418:429],pre.auto$pred),type="l",main= "ARIMA prediction effect with
2014 to 2020")
```

#### Downtown neighbourhood - Monthly
````

````
```{r}
# Transpose the DOWNTOWN row
downtown_row_m <- crime_neigh_month[crime_neigh_month$name == "Down-
town", ]
downtown_row_m <- as.data.frame(downtown_row_m)
downtown_row_m <- downtown_row_m[, -c(ncol(downtown_row_m)-1, ncol(down-
town_row_m))]
downtown_row_m <- as.data.frame(t(downtown_row_m))
colnames(downtown_row_m) <- as.character(downtown_row_m[1, ])
downtown_row_m <- downtown_row_m[-1, , drop = FALSE]
downtown_row_m <- slice(downtown_row_m, 1:(nrow(downtown_row_m) - 2))
downtown_row_m <- downtown_row_m[1:(nrow(downtown_row_m) - 2), ]
downtown_row_m <- as.data.frame(downtown_row_m)
downtown_row_m <- gather(downtown_row_m, key = "Date", value = "Down-
town")
downtown_row_m <- subset(downtown_row_m, select = -c(Date))

plot(downtown_row_m$Downtown, ylab="Monthly Counts", xlab="Time in
Months", type="l", main="Monthly vehicle theft crime from 2010 to 2022")
```

The dependence between consecutive observations is linear.

```{r}
str(downtown_row_m)
downtown_row_m$Downtown <- as.numeric(downtown_row_m$Downtown)
lag.plot(downtown_row_m$Downtown, lags=3, do.lines=FALSE)
```

#### ACF

it indicates that there is significant positive autocorrelation in the
data. This means that there is a strong relationship between the current
observation and the previous observations in the time series, and this
relationship can be used to make predictions about future values of the
time series.

There is no strong cyclic pattern in the autocorrelation for the Downtown
weekly plot.

```{r}
acf(downtown_row_m$Downtown, lag.max=52, xlab="Lag", ylab="ACF",
main="Autocorrelation plot of weekly vehicle theft crime")

```

After two difference, lag autocorrelation is reduced

```{r}
downtown_row_m.diff.Downtown <- diff(downtown_row_m$Downtown, lag=52,
differences=1)

acf(downtown_row_m.diff.Downtown, lag.max=52, xlab="Lag", ylab="ACF",
main="Differenced Autocorrelation plot of monthly vehicle theft crime")
```

#### PACF
````

````
Following shows the partial autocorrelation plot of the monthly average
temperature in East Anglia.

```{r}
pacf(downtown_row_m$Downtown, lag.max=52,xlab="Lag",ylab="PACF",main=
"Partial Autocorrelation plot of weekly vehicle theft crime")
```

#### ARIMA and Seasonality

If the ACF decays to zero and the first 52 lags are above the confidence
level, while the PACF cuts off after lag 4, it suggests that the data
might have a seasonal component along with an autoregressive (AR) compo-
nent.

```{r}
auto_arima <- auto.arima(downtown_row_m$Downtown[1:96], seasonal = TRUE)
print(auto_arima)
```

#### NRMSE

```{r}
NRMSE_fit <- NRMSE(res=auto_arima$residuals, obs=downtown_row_m$Down-
town[1:96])
print(NRMSE_fit)
```

#### Diagnostic Checking

```{r}
tsdiag(auto_arima)
```

In general, if the p-value is less than a predefined significance level
(e.g., 0.05), we reject the null hypothesis and conclude that there is
evidence of autocorrelation in the residuals. However, in your case, the
p-value is 0.9723, which is much greater than 0.05. This means that there
is no evidence to reject the null hypothesis, so we can't conclude that
there is autocorrelation in the residuals of the auto_arima model. This
suggests that the model has captured the temporal dependencies in the
data reasonably well.

```{r}
Box.test(auto_arima$residuals,lag=1)
Box.test(auto_arima$residuals,lag=2)
Box.test(auto_arima$residuals,lag=3)
```

```{r}
pre.auto <-predict(auto_arima, n.ahead=12)


length(downtown_row_m$Downtown[97:108])
length(pre.auto$pred)
print(pre.auto)
````
````

~~~~
```

```{r}
matplot(1:12,cbind(downtown_column$Down-
town[97:108],pre.auto$pred),type="l",main= "ARIMA prediction effect with
2014 to 2020")
```
#### Space-time Semivariogram

```{r include=FALSE}

crime_neigh_week2 <- st_read("./crime_neigh_week/crime_neigh_week.shp")

coords_centroid <- st_coordinates(st_centroid(crime_neigh_week2$geome-
try))
coords_centroid <- na.omit(coords_centroid)

pts <- SpatialPoints(coords_centroid[,1:2],
proj4string=CRS("+init=epsg:4326 +proj=longlat +ellps=WGS84 +datum=WGS84
+no_defs +towgs84=0,0,0"))

time <- seq(as.Date("2010-01-01"), length = 689, by = "week")
print(time)

crime_neigh_week2 <- as.data.frame(crime_neigh_week2)
crime_neigh_week2 <- crime_neigh_week2[, -ncol(crime_neigh_week2)]
crime_neigh_week2 <- crime_neigh_week2[, 2:ncol(crime_neigh_week2)]
crime_neigh_week2 <- as.matrix(crime_neigh_week2)

stfdf <- STFDF(pts, time, data.frame(as.vector(t(crime_neigh_week2))))

names(stfdf@data) <- "Crime"
```

```{r}
ChSTVar <- variogram(Crime~1, stfdf, width=100, cutoff=100,tlags=0:10)
```

```{r}
plot(ChSTVar)
```

γ - semivariance measures spatial and spatiotemporal dependence between
observations.
Degree of similarity or dissimilarity between data points as a function
of the distance and/or time lag between them. In other words, gamma quan-
tifies the spatial or spatiotemporal structure of the data.

A semivariogram plot displays gamma values on the vertical axis (z-axis
in a 3D plot) against the distance (x-axis) and time lag (y-axis). The
plot helps to visualize and understand how the spatial or spatiotemporal
correlation in the data changes with increasing distance or time lag.

```{r}
ChSTVar_noNA <- ChSTVar[!is.na(ChSTVar$gamma),]
plot(ChSTVar_noNA, wireframe=T)
```
~~~~

```
## STARIMA - Weekly

```{r include=FALSE}
crime_neigh_month <- st_read("./crime_neigh_month/crime_neigh_month.shp")
crime_neigh_week <- st_read("./crime_neigh_week/crime_neigh_week.shp")

crime_neigh_week_df <- as.data.frame(crime_neigh_week)
crime_neigh_week_df <- crime_neigh_week_df[, -ncol(crime_neigh_week_df)]
transposed_df <- as.data.frame(t(crime_neigh_week_df))
transposed_df <- data.frame(Week = rownames(transposed_df), trans-
posed_df)

colnames(transposed_df) <- transposed_df[1, ]
transposed_df <- transposed_df[-1, ]
transposed_df <- transposed_df[, -1]
transposed_df <- data.frame(lapply(transposed_df, as.numeric))

nb <- poly2nb(crime_neigh_week, queen = TRUE, snap = 0.1)
weight_matrix <- nb2mat(nb, style = "W", zero.policy = TRUE)
weight_matrix <- as.matrix(weight_matrix)
rownames(weight_matrix) <- crime_neigh_week$name
colnames(weight_matrix) <- crime_neigh_week$name

weekly_crime <- transposed_df
```

STACF

```{r}
weekly_crime.mat <- as.matrix(weekly_crime)
stacf(weekly_crime.mat, weight_matrix, 52)
```
Strong autocorrelation with next value, and quickly drop down but all
above the significance level

```{r}
weekly_crime.mat.diff <- diff(weekly_crime.mat,lag=1,differences=1)
stacf(weekly_crime.mat.diff, weight_matrix, 52)
```

STPACF

```{r}
stpacf(weekly_crime.mat, weight_matrix, 52)
```

```{r}
stpacf(weekly_crime.mat.diff, weight_matrix, 52)
```

Fitting STARIMA (2010-2017) predict (2018-2019)

```{r}
W_fit<-list(w1=weight_matrix)
# best parameters
fit.star <- starima_fit(weekly_crime.mat[1:416,],W_fit,p=1,d=0,q=2)
```
```

````
```

```{r}
stacf(fit.star$RES,weight_matrix,52)
```

```{r}
hist(fit.star$RES[,6])
```

predicting 2018-2019

```{r}
pre.star <- starima_pre(weekly_crime.mat[416:520,], model=fit.star)
matplot(1:105,cbind(weekly_crime[416:520, 1],pre.star$PRE[,1]),type="l")
```

```{r}
pre.star$NRMSE
```
fit training set from 2010 to 2019, predict 2020 to 2022 - Weekly

```{r}
fit.star2 <- starima_fit(weekly_crime.mat[1:520,],W_fit,p=1,d=0,q=2)
```

```{r}
stacf(fit.star2$RES,weight_matrix,52)
```

```{r}
hist(fit.star2$RES[,6])
```

predicting 2020 - 2022

No much different for the covid period, it echoes with the hypothesis

```{r}
pre.star2 <- starima_pre(weekly_crime.mat[521:677,], model=fit.star2)
matplot(1:157,cbind(weekly_crime[521:677, 1],pre.star2$PRE[,1]),type="l")
```

STARIMA - Monthly

```{r include=FALSE}
crime_neigh_month <- st_read("./crime_neigh_month/crime_neigh_month.shp")
crime_neigh_week <- st_read("./crime_neigh_week/crime_neigh_week.shp")

crime_neigh_month_df <- as.data.frame(crime_neigh_month)
crime_neigh_month_df <- crime_neigh_month_df[, -
ncol(crime_neigh_month_df)]
transposed_df <- as.data.frame(t(crime_neigh_month_df))
transposed_df <- data.frame(Month = rownames(transposed_df), trans-
posed_df)

colnames(transposed_df) <- transposed_df[1, ]
````

```
transposed_df <- transposed_df[-1, ]
transposed_df <- transposed_df[, -1]
transposed_df <- data.frame(lapply(transposed_df, as.numeric))

nb <- poly2nb(crime_neigh_month, queen = TRUE, snap = 0.1)
weight_matrix <- nb2mat(nb, style = "W", zero.policy = TRUE)
weight_matrix <- as.matrix(weight_matrix)
rownames(weight_matrix) <- crime_neigh_month$name
colnames(weight_matrix) <- crime_neigh_month$name

monthly_crime <- transposed_df
```

STACF

```{r}
monthly_crime.mat <- as.matrix(monthly_crime)
stacf(monthly_crime.mat, weight_matrix, 36)
```
Strong autocorrelation with next value, and quickly drop down but all
above the significance level

```{r}
monthly_crime.mat.diff <- diff(monthly_crime.mat,lag=1,differences=1)
stacf(monthly_crime.mat.diff, weight_matrix, 36)
```

STPACF

```{r}
stpacf(monthly_crime.mat, weight_matrix, 52)
```

```{r}
stpacf(monthly_crime.mat.diff, weight_matrix, 52)
```

Fitting STARIMA (2010-2017) predict (2018-2019) - monthly

```{r}
W_fit<-list(w1=weight_matrix)
# best parameters
fit.star <- starima_fit(monthly_crime.mat[1:97,],W_fit,p=1,d=0,q=3)
```

```{r}
stacf(fit.star$RES,weight_matrix,52)
```

```{r}
hist(fit.star$RES[,6])
```

predicting 2018-2019

```{r}
pre.star <- starima_pre(monthly_crime.mat[98:122,], model=fit.star)
```

```
matplot(1:25,cbind(monthly_crime[98:122, 1],pre.star$PRE[,1]),type="l")
```

```{r}
pre.star$NRMSE
```

fit training set from 2010 to 2019, predict 2020 to 2022 - monthly

```{r}
fit.star2 <- starima_fit(monthly_crime.mat[1:121,],W_fit,p=1,d=0,q=3)
```

```{r}
stacf(fit.star2$RES,weight_matrix,52)
```

```{r}
hist(fit.star2$RES[,6])
```

predicting 2020 - 2022

No much different for the covid period, it echoes with the hypothesis

```{r}
pre.star2 <- starima_pre(weekly_crime.mat[122:156,], model=fit.star2)
matplot(1:35,cbind(weekly_crime[122:156, 1],pre.star2$PRE[,1]),type="l")
```

```{r}
pre.star2$NRMSE
```