STAT 420 - Project Report 2

Winston Chen, Naidan Ganbold, Daniel Hooks, Erik Janc, Michael Lee

November 11, 2021

1 Introduction

The data-set we are using consists of pollution levels in various cities across the United States. Specifically, it contains mean NO2, SO2, O3, and CO levels measured daily. Also included are details regarding the air quality index (AQI) of each pollutant for each day, the maximum pollutant level recorded, and the hour it was recorded. To narrow the scope of our research objectives, we focus on only the mean pollutant levels. Further, we are only considering New York and Los Angeles for our analysis. To explicitly state our research questions, we hope to analyze trends and patterns over time of the various pollutants in each city. Furthermore, we aim to compare these patterns between the two cities and consider any external factors that may contribute to our findings. For example, we may consider wildfires that have occurred in California that could lead to increases in measured pollutant levels or any environmental laws or regulations passed that could lead to decreases in pollutant levels.

2 Data Preprocessing

A fair amount of data preprocessing was necessary to prepare the data to perform relevant analysis. First, we subset the data on the `City` attribute to consider only records for Los Angeles and New York. In order to make the data more readable, we drop irrelevant columns and keep only the `City`, `Date.Local`, `NO2.Mean`, `O3.Mean`, `SO2.Mean`, `CO.Mean`. Note that we exclude columns that denote each pollutant's units (after verifying that all units for each pollutant in each `City` are consistent). In other words, we take note that all NO2 mean values are given in parts per billion, then drop the `NO2.Units` column. We do the same for each pollutant while noting their units:

- NO2 : parts per billion
- O3 : parts per million
- SO2 : parts per billion
- CO: parts per million

The data does not exactly provide daily data; each day includes four separate rows with differing measurements for each city. In order to simplify our analysis, we average these columns so that there is only one measurement per day. We expect to see annual seasonality in this data, and in order to model this, we would require a lag of 365. However, due to a limitation in R which prevents ARIMA models with lags of more than 350, we aggregate monthly to apply a lag of 12 to model annual seasonality. Our forecasting, therefore, predicts future monthly averages as opposed to future daily values. All relevant code is in section 7.

3 Exploratory Data Analysis

Our initial steps taken for exploratory data analysis consisted of simply plotting each time series. To get a general idea of how the cities compared, we plot them next to each other. The plots can be found in figures 1 through 4 in section 6.

We can see that the O3 levels in both cities are very similar, and there is significant overlap between the two series. For NO2 and CO, we can see that the pollutant levels in Los Angeles are generally higher or more variable, especially in the earlier years from 2000-2005, while the respective levels in New York or more stable. For SO2 levels, we find that there are some significant differences. Namely, the SO2 levels in New York are significantly higher than that in Los Angeles; while New York levels are often in the double-digits (in parts per million), the Los Angeles levels only ever reach around 4 or 5 parts per million. In all cases (across all pollutants and both cities), we see evidence of seasonality as we observe a consistent cycle of ups and downs in pollutant levels on a seemingly yearly basis. Suggesting that our models reflect some seasonal pattern supports our initial hypothesis that the data is seasonal. We also note that most of the series are non-stationary; we observe non-constant means (especially in the Los Angeles data) and non-constant variance over time. Hence a data transformation is needed to remove these trends.

After viewing the initial time series plots, we move on to examining the series after applying log-transformations, then observe their ACF and PACF plots to gain insight into our model selection process and parameter choosing. To remove increasing or decreasing trends, we difference the data with a lag of 1; to remove seasonality, we difference the data by a lag of 12. We re-plot these transformed and differenced series to verify that they appear stationary.

4 Modeling and Forecasting

Since the graphs indicate seasonality for all pollutants of Los Angeles and New York, we first analyzed the ACF and PACF plots of the log-transformed data (Figures 5 to 12). Within the pollutants, we decided to analyze CO and O3. After examining the PACF and ACF graphs for CO and O3 we have evidence of seasonality based on the spikes and lag intervals corresponding to end of the year. From here we decided to test the different ARIMA models. After examining the AIC for each of the models we found that a seasonal $ARIMA(1,1,1) \times (0,1,1)_{12}$ was the best fit the data. We proceeded with forecasting for the next year for both New York and Los Angeles for CO and O3 data (Figures 13 to 16).

5 Tentative Conclusions

Examining the forecast for New York and Los Angeles based on our seasonal $ARIMA(1,1,1) \times (0,1,1)_{12}$ model, we can see the forecasts for both CO and O3 follow pattern trends. Since the data obtained is seasonal and has correlations with previous years, our forecast follows a similar trend. We can see O3 and CO levels rise and fall depending on their respective months in that year. Hence showing that our prediction for the next months is not far off from seasonal trends.

Steps to improve the model are to take a power transformation or find a lambda transformation that can improve the normality of the data. Additionally, we can pursue alternative routes in different cities and forecast other pollutants. In order to verify the accuracy of our prediction, we can preserve 10 percent of the data and use 90 percent to check how well our estimated values match the preserved values in the data set. This method can show the validity of our predictions for future months and years.

6 Appendix A: Figures

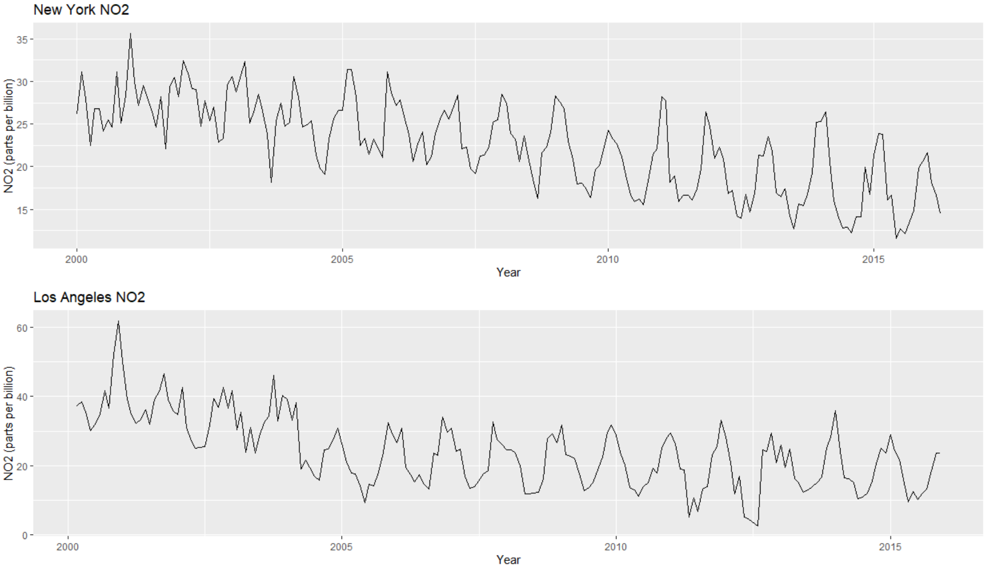

Figure 1: Mean NO2 levels in New York and Los Angeles

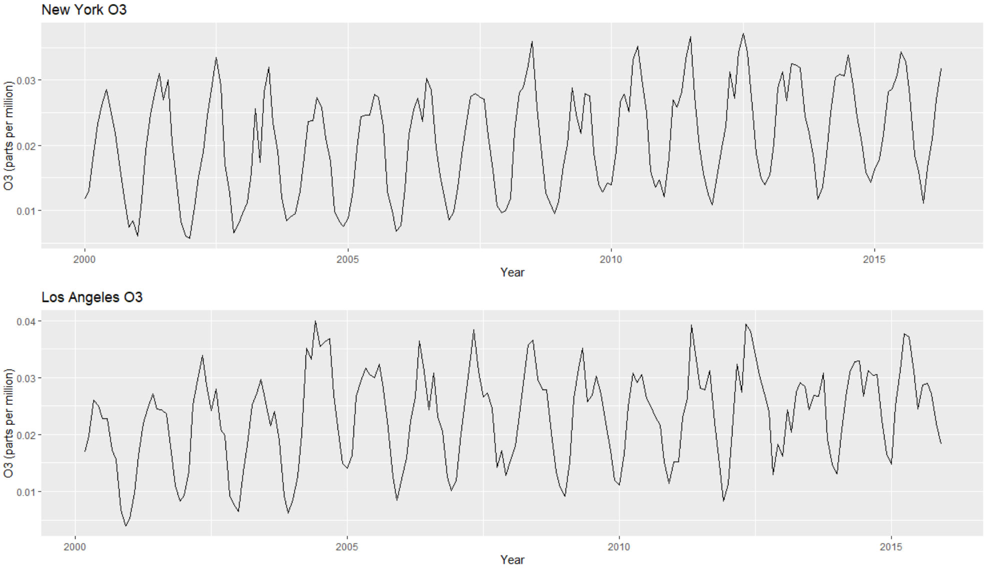

Figure 2: Mean O3 levels in New York and Los Angeles

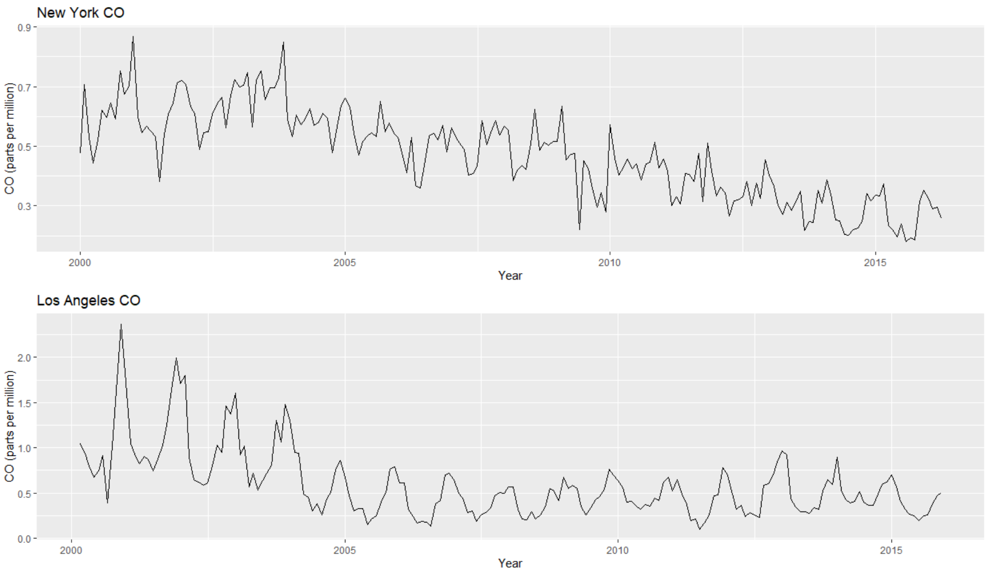

Figure 3: Mean CO levels in New York and Los Angeles

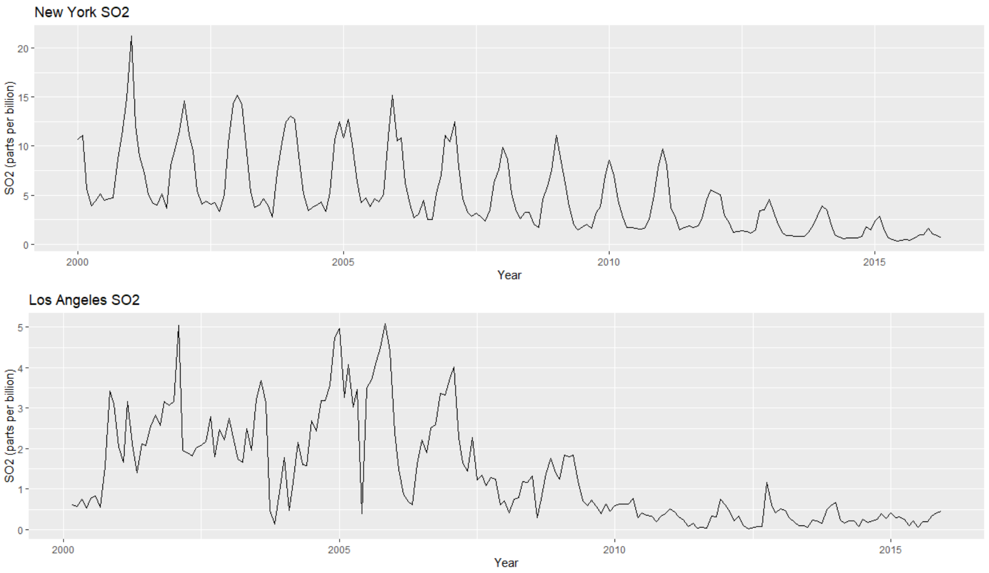

Figure 4: Mean SO2 levels in New York

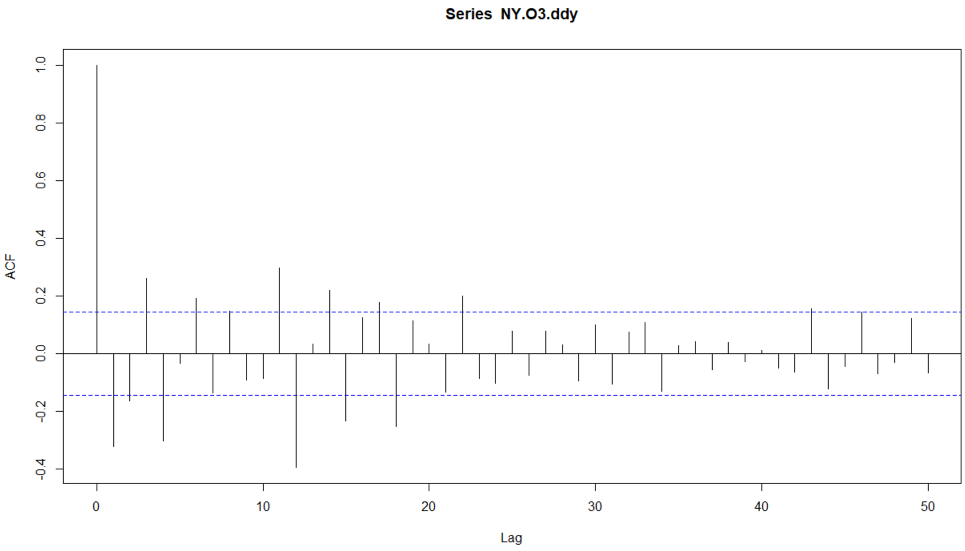

Figure 5: New York ACF for O3

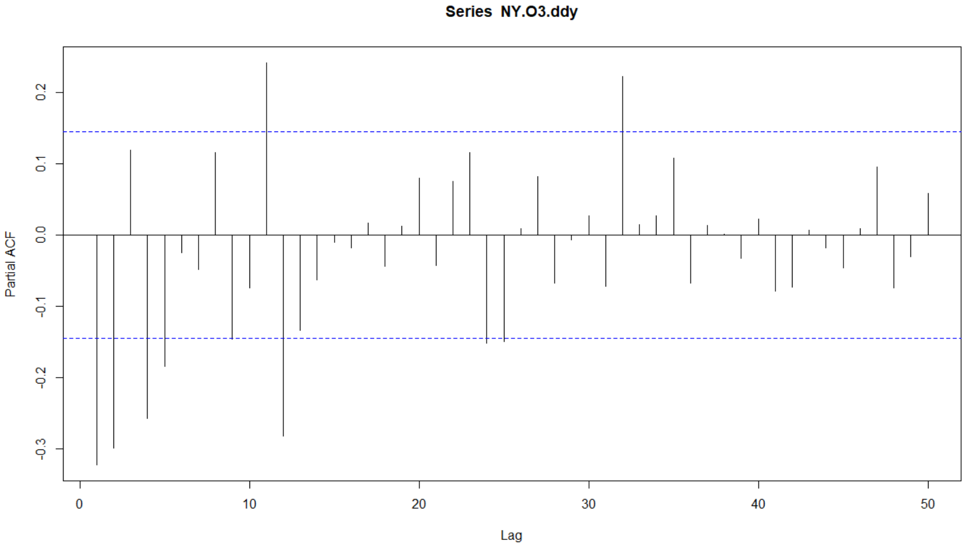

Figure 6: New York PACF for O3

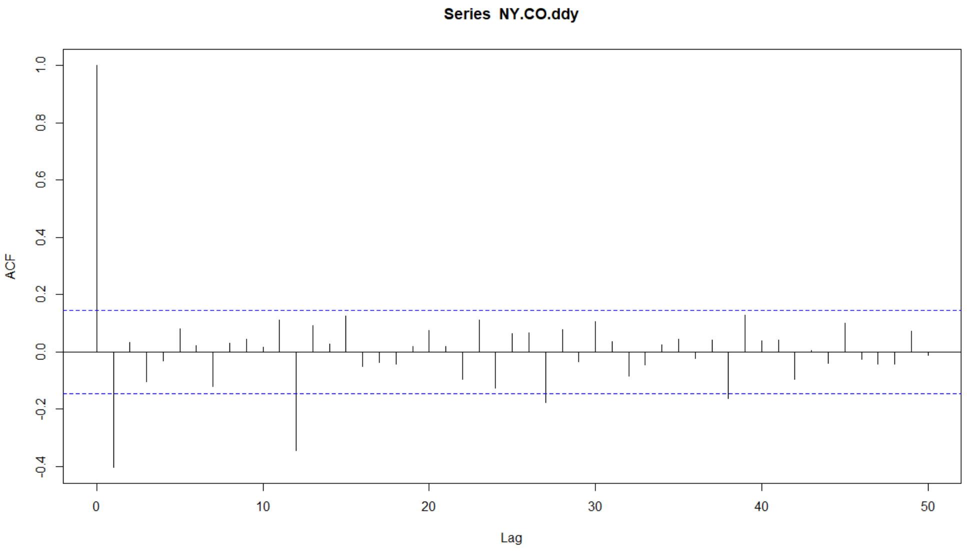

Figure 7: New York ACF for CO

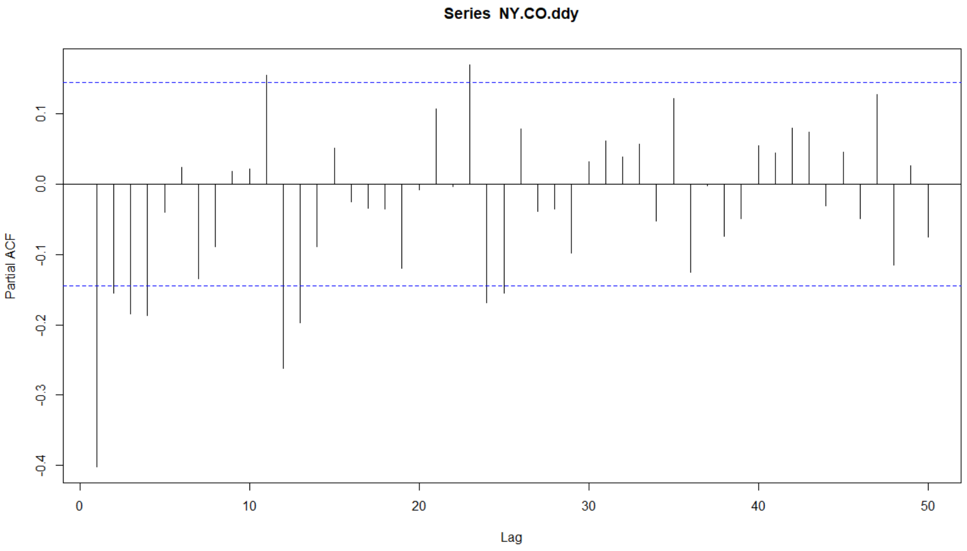

Figure 8: New York PACF for CO

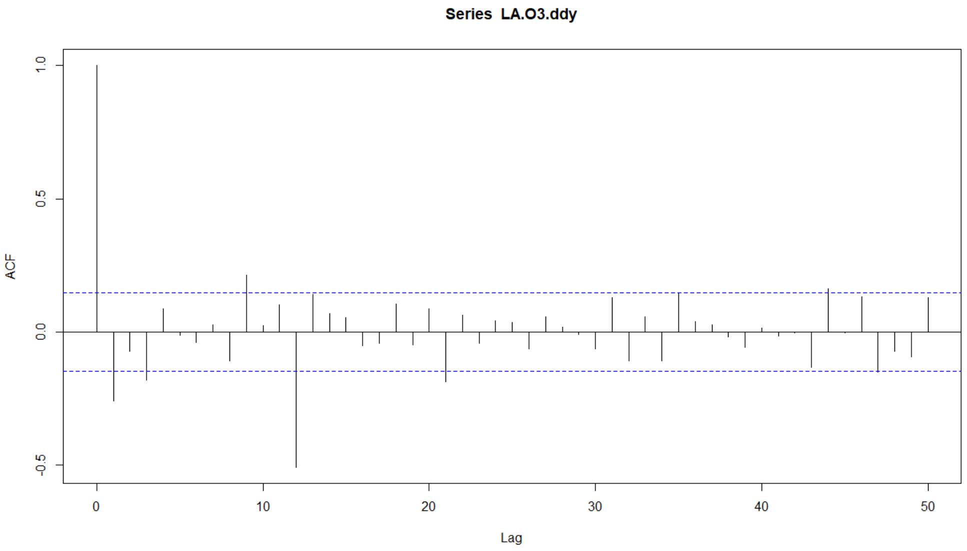

Figure 9: Los Angeles ACF for O3

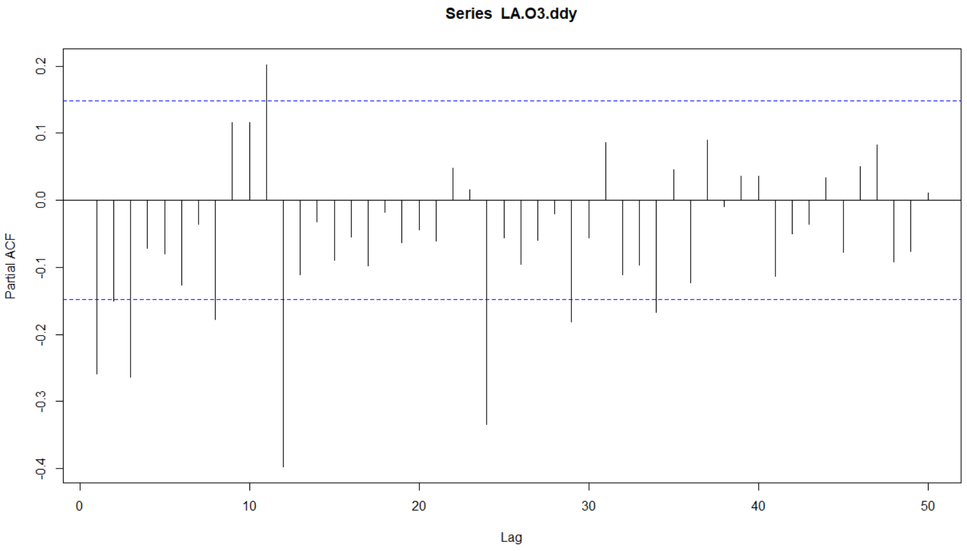

Figure 10: Los Angeles PACF for O3

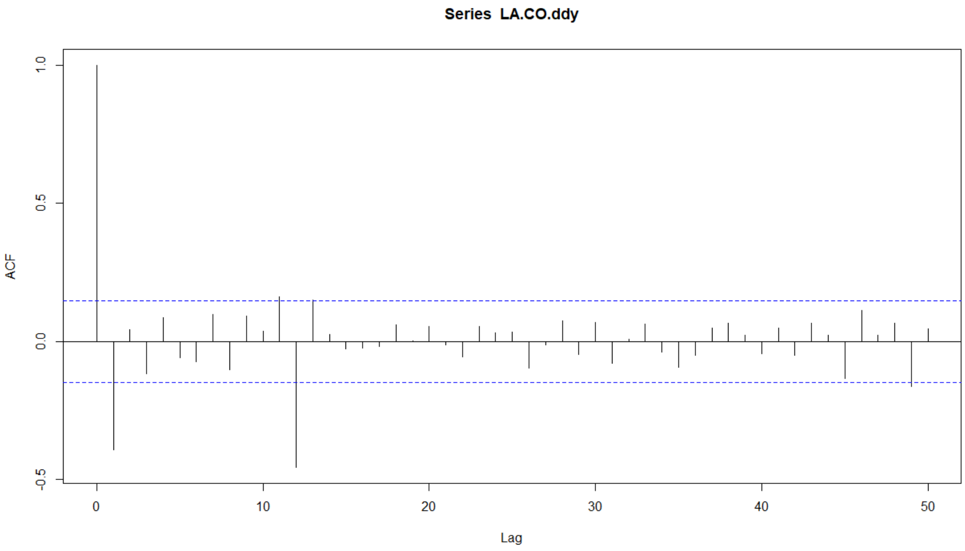

Figure 11: Los Angeles ACF for CO

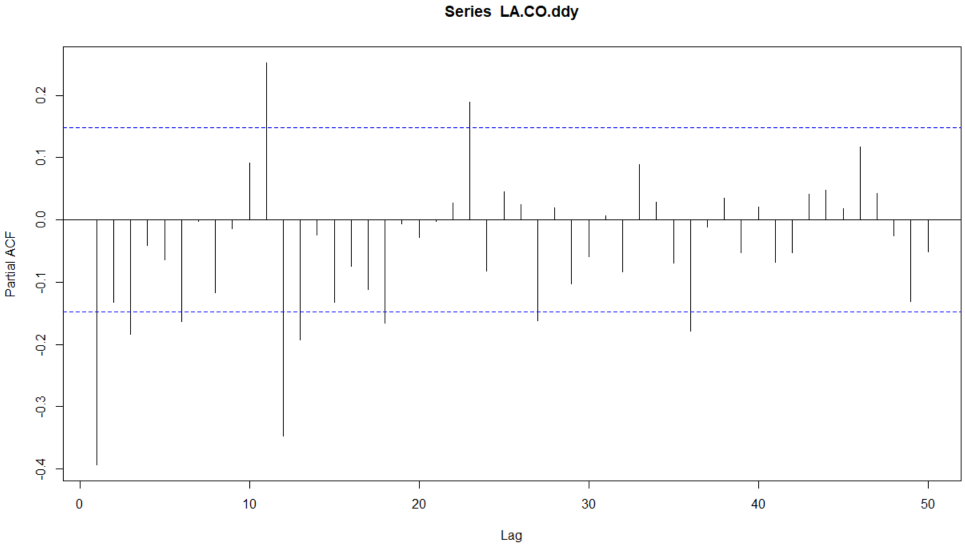

Figure 12: Los Angeles PACF for CO

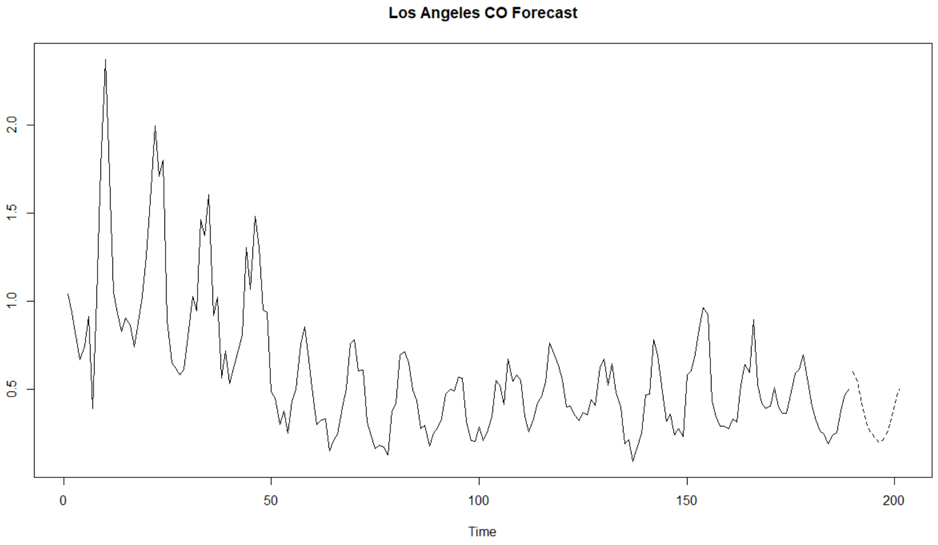

Figure 13: Los Angeles CO forecast

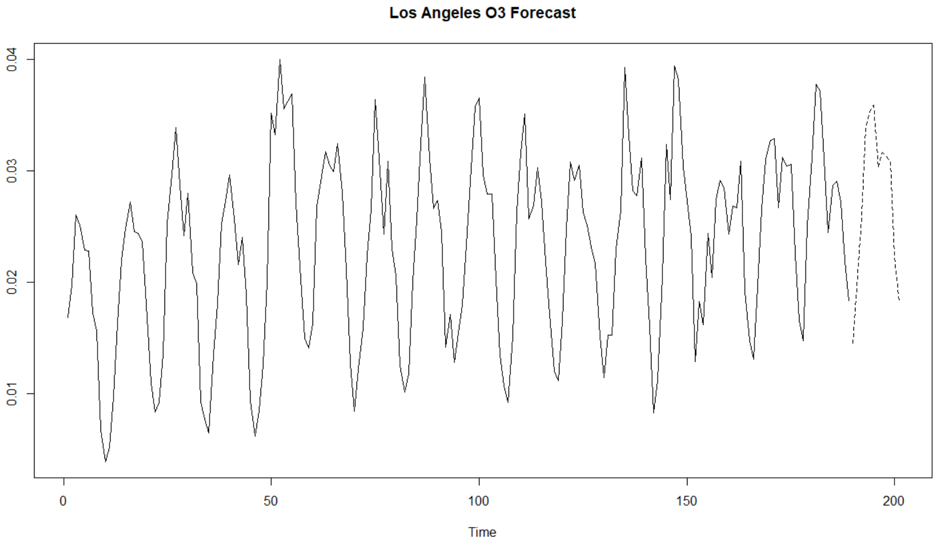

Figure 14: Los Angeles O3 forecast

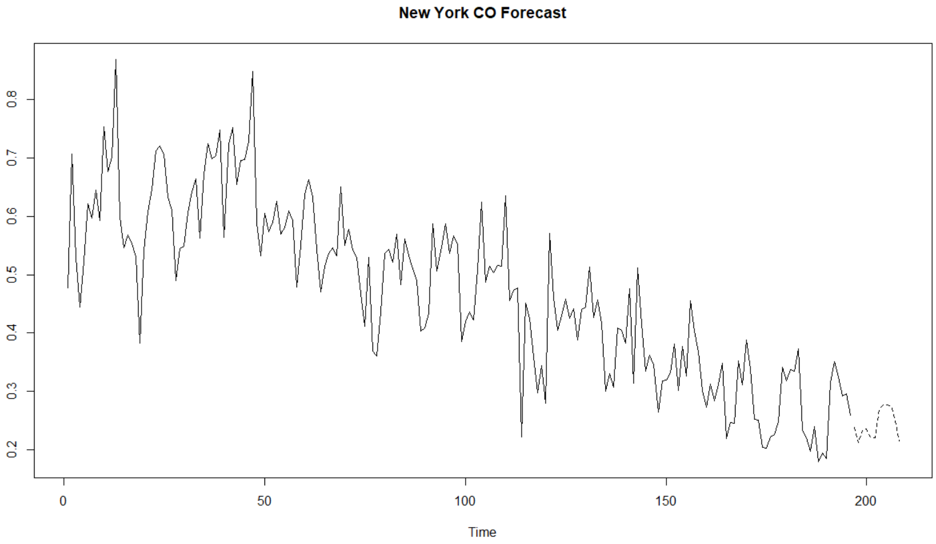

Figure 15: New York CO forecast

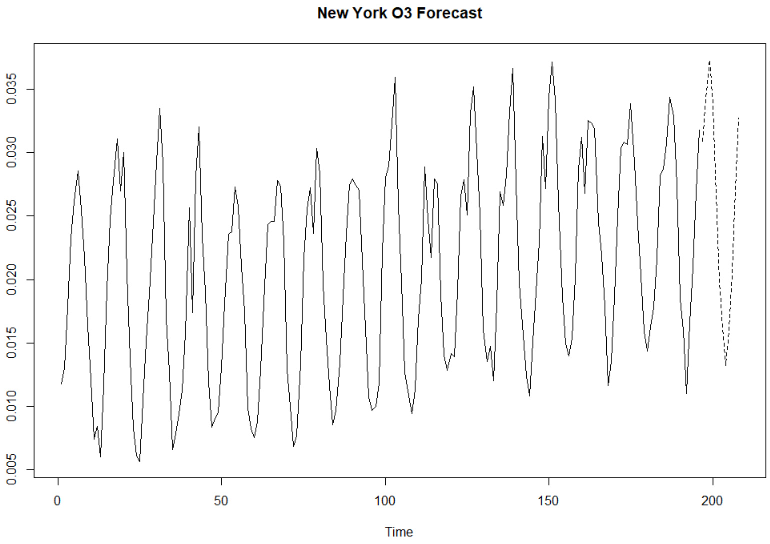

Figure 16: New York O3 forecast

7 Appendix B: Code

7.1 Data Preprocessing

```r
# Subset on relevant cities
cities = c("Los Angeles", "New York")
pollution = subset(
  pollution_raw,
  pollution_raw$City %in% cities,
  drop=TRUE
)

# Subset on relevant columns
cols = c("City", "Date.Local", "NO2.Mean", "O3.Mean", "SO2.Mean", "CO.Mean")
pollution = pollution[cols]
head(pollution)

# Group by city and date, average columns
pollution = pollution %>%
  group_by(City, Date.Local) %>%
  summarize_each(mean)
pollution$Date.Local = as.Date(pollution$Date.Local, format="%Y-%m-%d")
head(pollution)

# Aggregate daily data into monthly data
# Save as separate dataframes for each pollutant in each city
aggregate_monthly = function(city, pollutant_idx) {
  df = pollution[pollution$City==city,]
  df$Month = months(df$Date.Local)
  df$Year = format(df$Date.Local, format="%Y")
  if (pollutant_idx == 1) {
    df = aggregate(NO2.Mean ~ Month + Year, df, mean)
    df$Date.Local = as.Date(paste(df$Year, df$Month, "1"), format="%Y %b %d")
    return(xts(df$NO2.Mean, order.by=df$Date.Local))
  } else if (pollutant_idx == 2) {
    df = aggregate(O3.Mean ~ Month + Year, df, mean)
    df$Date.Local = as.Date(paste(df$Year, df$Month, "1"), format="%Y %b %d")
    return(xts(df$O3.Mean, order.by=df$Date.Local))
  } else if (pollutant_idx == 3) {
    df = aggregate(SO2.Mean ~ Month + Year, df, mean)
    df$Date.Local = as.Date(paste(df$Year, df$Month, "1"), format="%Y %b %d")
    return(xts(df$SO2.Mean, order.by=df$Date.Local))
  } else {
    df = aggregate(CO.Mean ~ Month + Year, df, mean)
    df$Date.Local = as.Date(paste(df$Year, df$Month, "1"), format="%Y %b %d")
    return(xts(df$CO.Mean, order.by=df$Date.Local))
  }
}
```

```

NY.NO2 = aggregate_monthly("New York", 1)
NY.O3 = aggregate_monthly("New York", 2)
NY.SO2 = aggregate_monthly("New York", 3)
NY.CO = aggregate_monthly("New York", 4)
LA.NO2 = aggregate_monthly("Los Angeles", 1)
LA.O3 = aggregate_monthly("Los Angeles", 2)
LA.SO2 = aggregate_monthly("Los Angeles", 3)
LA.CO = aggregate_monthly("Los Angeles", 4)
```

7.2 Exploratory Data Analysis

```
# Plot time series helpers
plot_single = function(ts, city_idx, pollutant_idx) {
  cities = c("New York", "Los Angeles")
  pollutants = c("NO2", "O3", "SO2", "CO")
  units = c("(parts per billion)", "(parts per million)", "(parts per billion)", "(parts per million)")
  autoplot(
    ts
  ) + labs(
    x="Year",
    y=paste(pollutants[pollutant_idx], units[pollutant_idx]),
    title=paste(cities[city_idx], pollutants[pollutant_idx])
  )
}

plot_together = function(ts1, ts2, pollutant_idx) {
  cities = c("New York", "Los Angeles")
  pollutants = c("NO2", "O3", "SO2", "CO")
  units = c("(parts per billion)", "(parts per million)", "(parts per billion)", "(parts per million)")
  p1 = autoplot(
    ts1
  ) + labs(
    x="Year",
    y=paste(pollutants[pollutant_idx], units[pollutant_idx]),
    title=paste(cities[1], pollutants[pollutant_idx])
  )
  p2 = autoplot(
    ts2
  ) + labs(
    x="Year",
    y=paste(pollutants[pollutant_idx], units[pollutant_idx]),
    title=paste(cities[2], pollutants[pollutant_idx])
  )
  gridExtra::grid.arrange(p1, p2, nrow=2, ncol=1)
}
```

7.3 Modeling and Forecasting

```r

## Forecast predictions for New York O3

NY.O3.ddy = diff(diff(log(NY.O3)), 12)[-c(seq(1, 13))]

acf(NY.O3.ddy, lag.max=50)
pacf(NY.O3.ddy, lag.max=50)


# Create various models and compare their AIC
arima(log(NY.O3), order=c(0, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.O3), order=c(0, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.O3), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.O3), order=c(1, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.O3), order=c(1, 1, 1), seasonal=list(order=c(1, 1, 1), period=12))$aic
arima(log(NY.O3), order=c(1, 1, 2), seasonal=list(order=c(1, 1, 1), period=12))$aic


# Select model and perform model diagnostics
NY.O3.model = arima(log(NY.O3), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))
plot(NY.O3.model$residuals, main="New York O3 Residuals")
acf(NY.O3.model$residuals)
pacf(NY.O3.model$residuals)
qqnorm(NY.O3.model$residuals)
qqline(NY.O3.model$residuals)


## Forecasting for New York O3

forecast = predict(NY.O3.model, n.ahead=12)
ts.plot(cbind(ts(NY.O3), exp(forecast$pred)), lty=1:2, main="New York O3 Forecast")


## Forecast predictions for New York CO

NY.CO.ddy = diff(diff(log(NY.CO)), 12)[-c(seq(1, 13))]

acf(NY.CO.ddy, lag.max=50)
pacf(NY.CO.ddy, lag.max=50)


# Create various models and compare their AIC
arima(log(NY.CO), order=c(0, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.CO), order=c(0, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.CO), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.CO), order=c(1, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(NY.CO), order=c(1, 1, 1), seasonal=list(order=c(1, 1, 1), period=12))$aic
arima(log(NY.CO), order=c(1, 1, 2), seasonal=list(order=c(1, 1, 1), period=12))$aic



# Select model and perform model diagnostics
NY.CO.model = arima(log(NY.CO), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))
plot(NY.CO.model$residuals, main="New York CO Residuals")
acf(NY.CO.model$residuals)
pacf(NY.CO.model$residuals)
qqnorm(NY.CO.model$residuals)
qqline(NY.CO.model$residuals)


## Forecasting for New York CO

forecast = predict(NY.CO.model, n.ahead=12)
ts.plot(cbind(ts(NY.CO), exp(forecast$pred)), lty=1:2, main="New York CO Forecast")




## Forecast predictions for Los Angeles O3

LA.O3.ddy = diff(diff(log(LA.O3)), 12)[-c(seq(1, 13))]

acf(LA.O3.ddy, lag.max=50)
pacf(LA.O3.ddy, lag.max=50)


# Create various models and compare their AIC
arima(log(LA.O3), order=c(0, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.O3), order=c(0, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.O3), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.O3), order=c(1, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.O3), order=c(1, 1, 1), seasonal=list(order=c(1, 1, 1), period=12))$aic
arima(log(LA.O3), order=c(1, 1, 2), seasonal=list(order=c(1, 1, 1), period=12))$aic


# Select model and perform model diagnostics
```

```r
LA.O3.model = arima(log(LA.O3), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))
plot(LA.O3.model$residuals, main="Los Angeles O3 Residuals")
acf(LA.O3.model$residuals)
pacf(LA.O3.model$residuals)
qqnorm(LA.O3.model$residuals)
qqline(LA.O3.model$residuals)


## Forecasting for Los Angeles O3

forecast = predict(LA.O3.model, n.ahead=12)
ts.plot(cbind(ts(LA.O3), exp(forecast$pred)), lty=1:2, main="Los Angeles O3 Forecast")


## Forecast predictions for Los Angeles CO

LA.CO.ddy = diff(diff(log(LA.CO)), 12)[-c(seq(1, 13))]

acf(LA.CO.ddy, lag.max=50)
pacf(LA.CO.ddy, lag.max=50)


# Create various models and compare their AIC
arima(log(LA.CO), order=c(0, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.CO), order=c(0, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.CO), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.CO), order=c(1, 1, 2), seasonal=list(order=c(0, 1, 1), period=12))$aic
arima(log(LA.CO), order=c(1, 1, 1), seasonal=list(order=c(1, 1, 1), period=12))$aic
arima(log(LA.CO), order=c(1, 1, 2), seasonal=list(order=c(1, 1, 1), period=12))$aic


# Select model and perform model diagnostics
LA.CO.model = arima(log(LA.CO), order=c(1, 1, 1), seasonal=list(order=c(0, 1, 1), period=12))
plot(LA.CO.model$residuals, main="Los Angeles CO Residuals")
acf(LA.CO.model$residuals)
pacf(LA.CO.model$residuals)
qqnorm(LA.CO.model$residuals)
qqline(LA.CO.model$residuals)


## Forecasting for Los Angeles CO

forecast = predict(LA.CO.model, n.ahead=12)
ts.plot(cbind(ts(LA.CO), exp(forecast$pred)), lty=1:2, main="Los Angeles CO Forecast")
```