Prediction of PM2.5 Particulate Levels for Beijing

Paul C. Holaway, Albert Li, and Julia Nagel

Department of Statistics, University of Illinois Urbana-Champaign

STAT 429: Time Series Analysis

December 4th, 2022

Abstract

The goal of this project was to train a model to predict PM2.5 particulate levels for cities in China. To do this, we used data from China's recording station Dongsi in Beijing and created two different time series models in RStudio, SARIMA, and ARMA-GARCH. A SARIMA model was constructed due to the cyclical trends present in the data. An ARMA-GARCH model was constructed due to the heteroscedasticity present in the data. Our results showed that while both models perform about the same in terms of our evaluation metric (RMSE), the SARIMA model had a less practical interpretation. It predicted a sharp drop for the beginning of January, while the ARMA-GARCH model had a more realistic gradual decline. The purpose of this project is to aid people in determining how safe the air quality is. We also hope that our model can be used to guide policy-making in China to improve air quality.

Keywords: PM2.5, SARIMA, ARMA-GARCH, Dongsi, Time Series

Introduction

Motivation

The goal of this project is to build prediction models for PM2.5 concentration levels. PM2.5 is particles with an aerodynamic diameter of 2.5 µm or less and is harmful to human health. According to Miller and Xu (2018), “recent literature has confirmed associations between PM2.5 exposure and total mortality, cardiovascular mortality, respiratory mortality, hypertension, lung cancer, influenza, and other adverse health outcomes.” Thus, it is important to have a model that could better predict the level of PM2.5 to help people avoid excess exposure to high-level concentrations.

Goal

This project has two main goals, the analysis of PM2.5 and model building for prediction. For the analysis portion, we attempted to find patterns, similarities, and differences among data recorded at four locations in China. We then tried to explain the reasons for those recordings, which might help to reduce the PM2.5 level in the future. We also aimed to utilize time-series knowledge to build several predictive models, then compare and find which model could best predict the PM2.5 level with the given data. Both model accuracy and complexity are accounted for when deciding which model is the best.

Data

Our data is a collection of readings of PM2.5 in Beijing, China. It was retrieved from Kaggle as a part of a collection of PM2.5 readings for five major Chinese cities (Beijing, Chengdu, Guangzhou, Shanghai, and Shenyang). We decided to use just Beijing to train our models as it is China's capital city. The data collected was between January 1st, 2010, and December 31st, 2015. However, not all recording sites began recording at the same time. There are 946,512 total data inputs and 91,124 NAs out of them, accounting for nearly 10% of the overall data set, mainly because there are no PM2.5 data recorded for all four PM stations until 8 A.M. Mar 5th, 2013. Given the time constraint, we only built multiple models for the Dongsi recording station. A description of the original data is below.

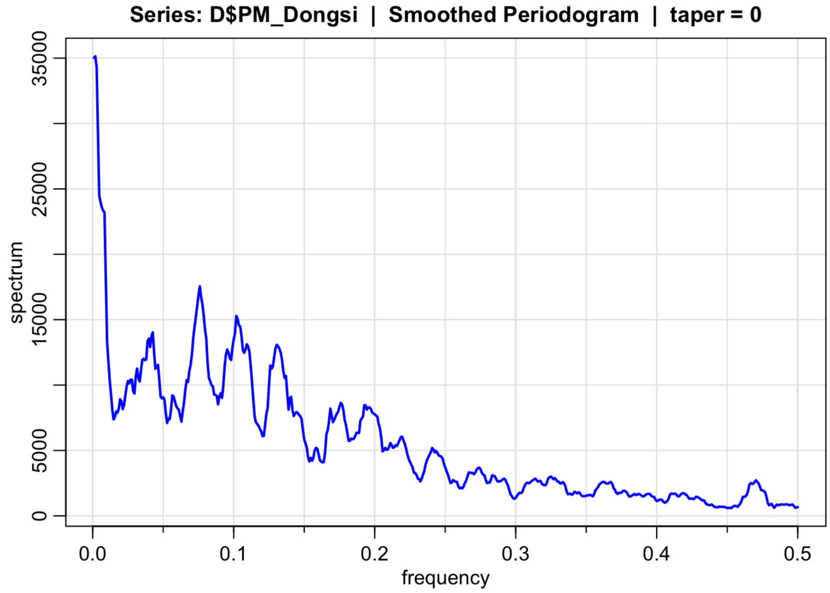
Data Set Variable Descriptions:
- No: Row Number
- year: Year when the data was recorded in the row.
- month: Month when the data was recorded in the row.
- day: Day when the data was recorded in the row.
- hour: Hour when the data was recorded in the row (in 24-hour format).
- season: Season when the data was recorded in the row.
  - 1 = December - February
  - 2 = March - May
  - 3 = June - August
  - 4 = September - November
- PM_Dongsi: PM2.5 concentration (µg/m3) recorded at Dongsi, Beijing (52% NAs).
- PM_Dongsihuan: PM2.5 concentration (µg/m3) recorded at Dongsihuan, Beijing (61% NAs).
- PM_Nongzhanguan: PM2.5 concentration (µg/m3) recorded at Nongzhanguan, Beijing (53% NAs).
- PM_US.Post: PM2.5 concentration (µg/m3) recorded by the US recording station at Dongsi, Beijing (3% NAs).
- DEWP: Dew Point (◦C)
- TEMP: Temperature (◦C)
- HUMI: Humidity (%)
- PRES: Pressure (hPa)
- cbwd: Combined Wind Direction
- Iws: Accumulated wind speed (m/s)
- pressure: Hourly Precipitation (mm)
- Iprec: Accumulated Precipitation (mm)

Subsetting

We then made four sub-datasets (Dongsi, Dongsihuan, Nongzhanguan, and USPost) and for each we:

- Removed all NAs from the PM column corresponding to the location it was recorded in, as PM2.5 is the response variable, and the predictors would not be valuable without this.
- Created one column for each sub-data set that contains all time-related information by combining the year, month, day, and hour columns.

- Created one column that contains the daily average for each station, as hourly input could create too much noise and won't provide much additional value to the user compared with the daily level. This is also the response value used for project building and testing purposes.
- All other ten columns from the main data set are included as well. Those remaining columns are as follows.
- Recorded: The exact hour, day, month, and year when the data was recorded in the row.
- season: Season when the data was recorded in the row.
    - 1 = December - February
    - 2 = March - May
    - 3 = June - August
    - 4 = September - November
- PM_... : PM2.5 concentration (µg/m3) recorded at the specific location in Beijing.
- DEWP: Dew Point (◦C)
- TEMP: Temperature (◦C)
- HUMI: Humidity (%)
- PRES: Pressure (hP a)
- cbwd: Combined Wind Direction
- Iws: Accumulated wind speed (m/s)
- pressure: Hourly Precipitation (mm)
- Iprec: Accumulated Precipitation (mm)

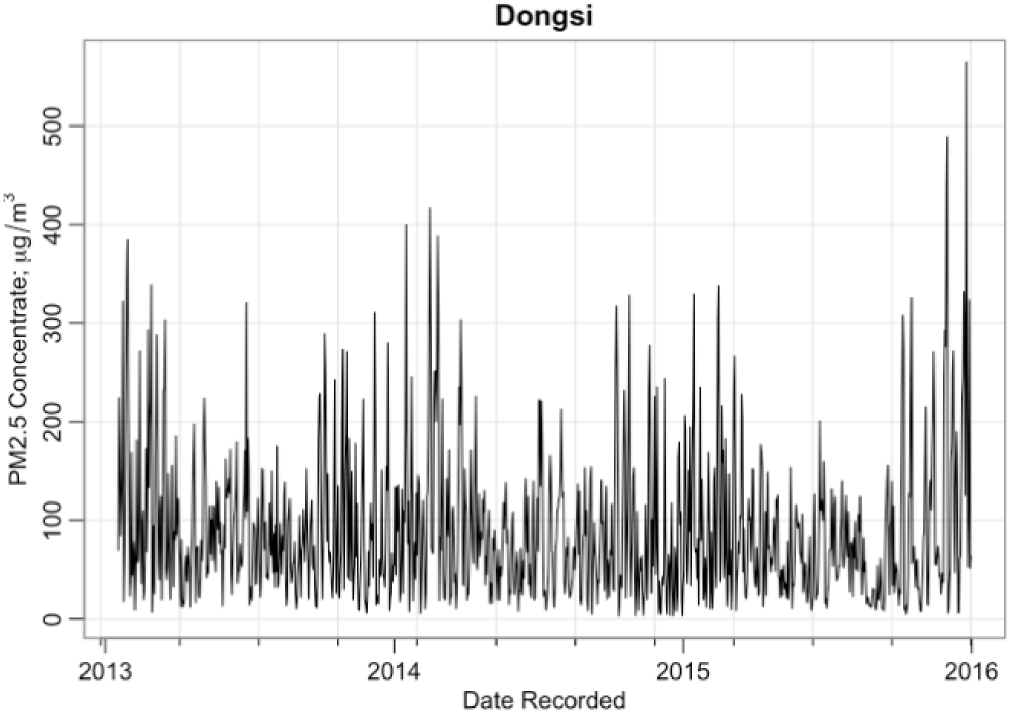


After checking the visualization of all four stations in Beijing, we believed that they all present a similar seasonal trend. However, there are differences in terms of extreme values and frequencies. We decided to focus our time on the Dongsi dataset and attempt to find the comparatively best possible model for this dataset. To decide on the frequency for the SARIMA model, we tried the smoothed periodogram to determine the optimal frequencies and found that a frequency of two weeks (14 days) was optimal for the SARIMA model.

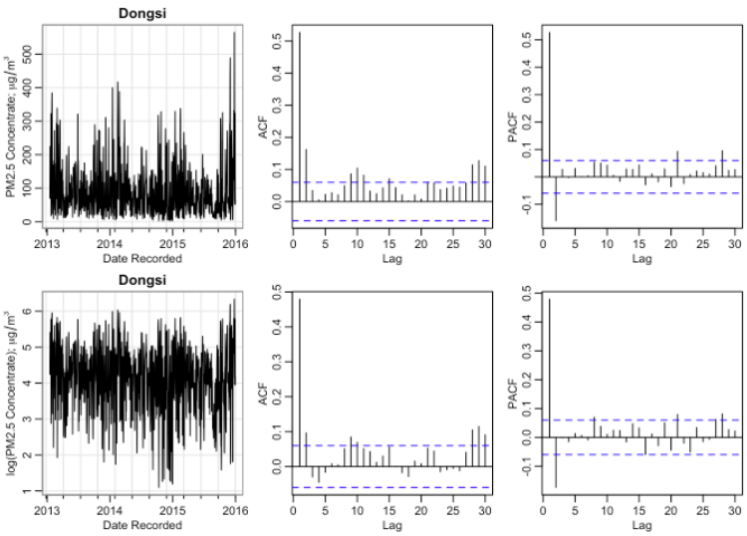
SARIMA Model

Method

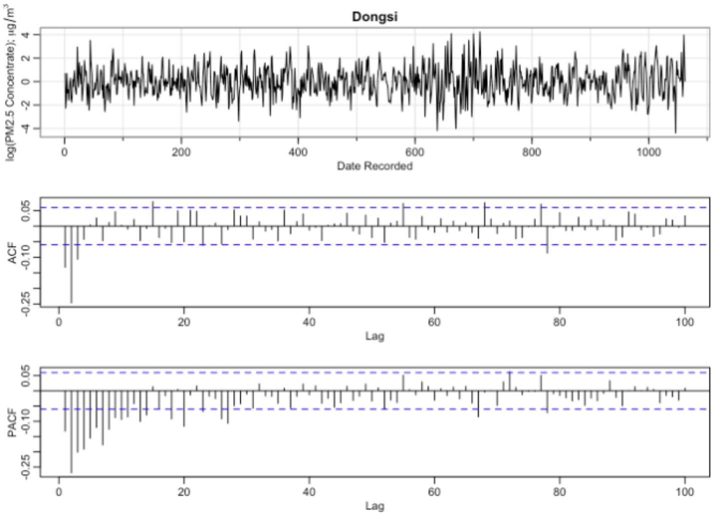
From the initial plotting of Dongsi, it looked like there was likely a seasonal trend, so we looked into SARIMA modeling for the data. There were also differences in variance, which would be an issue for modeling.



We then looked at the ACF and PACF plots of the Dongsi data. From these plots, we could see that there was a seasonal trend. To account for this and the difference in variance, we took a log transformation of the Dongsi data to remove the seasonality from the ACF and PACF plots and help reduce the issue of heteroscedasticity. Even after the log transformation of the data, the ACF and PACF continued to show seasonal trends.



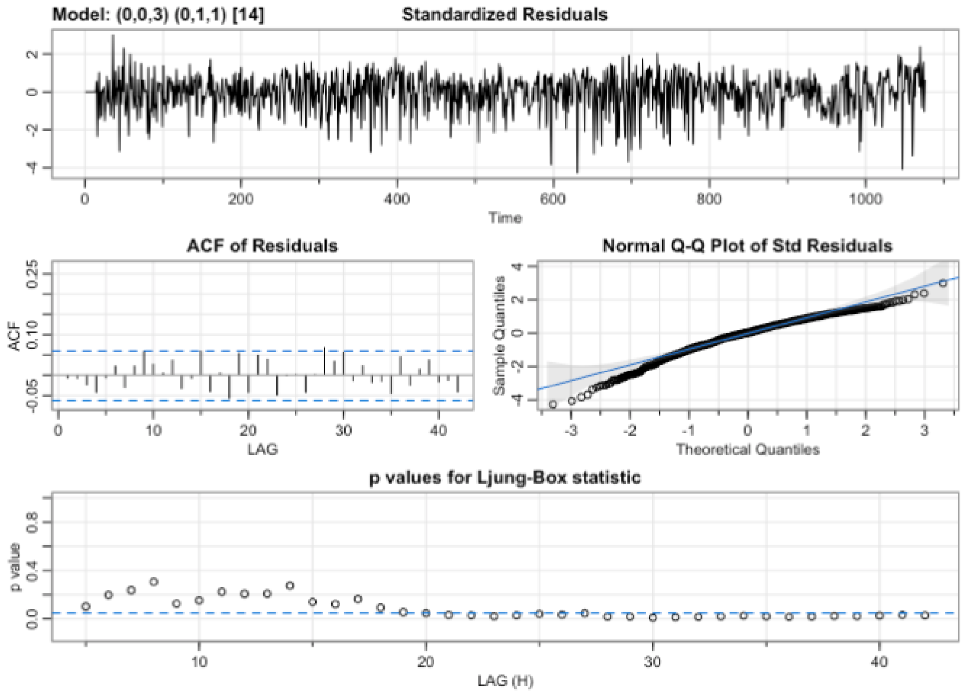
From this point on, we decided to add a two-week difference to the data. From the previous ACF and PACF plots, it looks like the periods were approximately 14 days which is why we decided on that difference. The time series plot for the differenced data looked more stationary, and the ACF and PACF plots improved. From the ACF plots, it looked to cut off around lag 3, and the PACF tails off.



We then decided to check the stationarity of the log-transformed differenced PM2.5 levels using the Augmented Dickey-Fuller Test and the KPSS Test for Stationarity. The

Dickey-Fuller Test resulted in a p-value of 0.01, which means our data was stationary. The KPSS Test for Stationarity also shows that our data is stationary with a p-value of 0.1. From here, we were confident in trying to build SARIMA models.

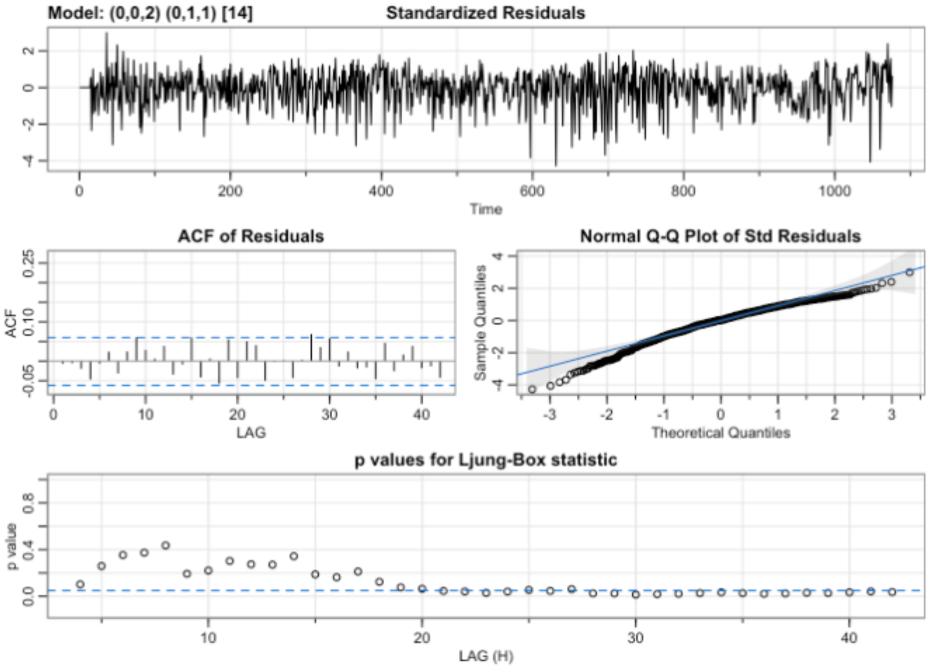
Our original model created was the SARIMA(0,0,3)×(0,1,1) with a seasonal period of 14 days. The results of the original model were overall not bad, but the third AR was not statistically significant, so we decided to try out different models to see if we could get better results.



Results

We then decided to do a SARIMA(0,0,2)×(0,1,1)$_{14}$ to remove the third AR seeing if this would improve our results. In this result, all of the estimates were statistically significant. We tried a SARIMA(0,0,2)×(1,1,1)$_{14}$ to see if that would improve anything, but not all the coefficients were significant. Therefore we stuck with the SARIMA(0,0,2)×(0,1,1)$_{14}$ as our best model for SARIMA. For calculations of the RMSE for our chosen model, we got an RMSE of

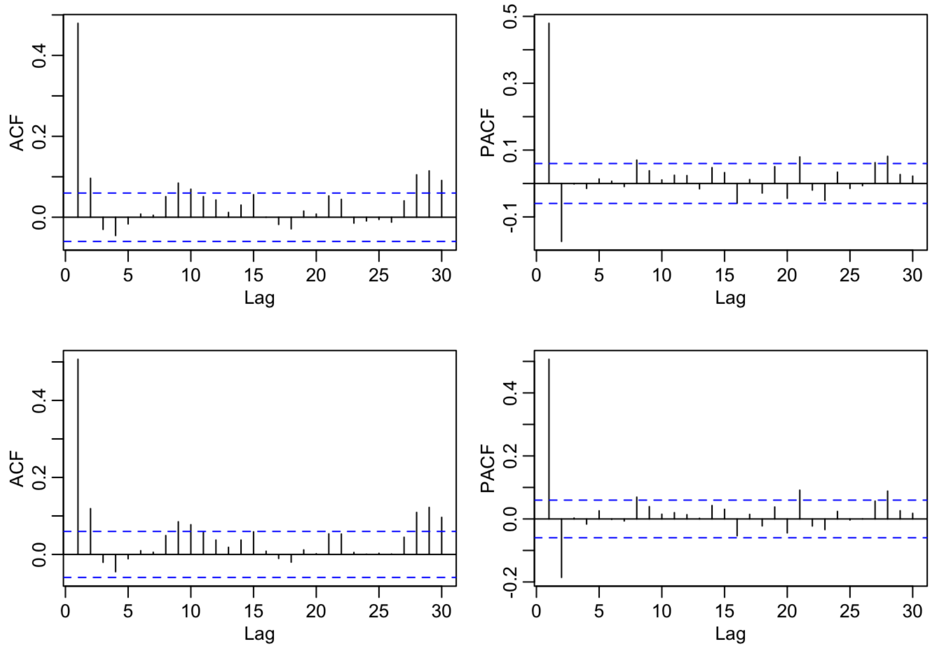
2.79 for the next-month-prediction, and for the first five observations, we got an RMSE of 1.766, which are both a pretty ideal range. Our chosen model passes the Ljung Box Test with a p-value of 0.4886, which means the autocorrelations of the time series are not different from 0.



ARMA-GARCH Model

Methods

As with the SARIMA model, we kept the log-transformed PM2.5 levels for the ARMA-GARCH model. While models using ARCH-GARCH processes are used to deal with issues of heteroscedasticity (R. Kumar, 2020), we kept the transformation for two reasons. The first was that the variance changed too much before transforming, resulting in poor prediction performance, as the range of PM2.5 levels ranged from three to over 500. The second was so it would be easier to compare the SARIMA model to the ARMA-GARCH model. To begin our model creation process, we started by looking at the ACF and PACF of the data and the squared values of the data. The plots on top represent the ACF/PACF for the data, and the plots on the bottom represent the ACF/PACF for the squared values of the data.

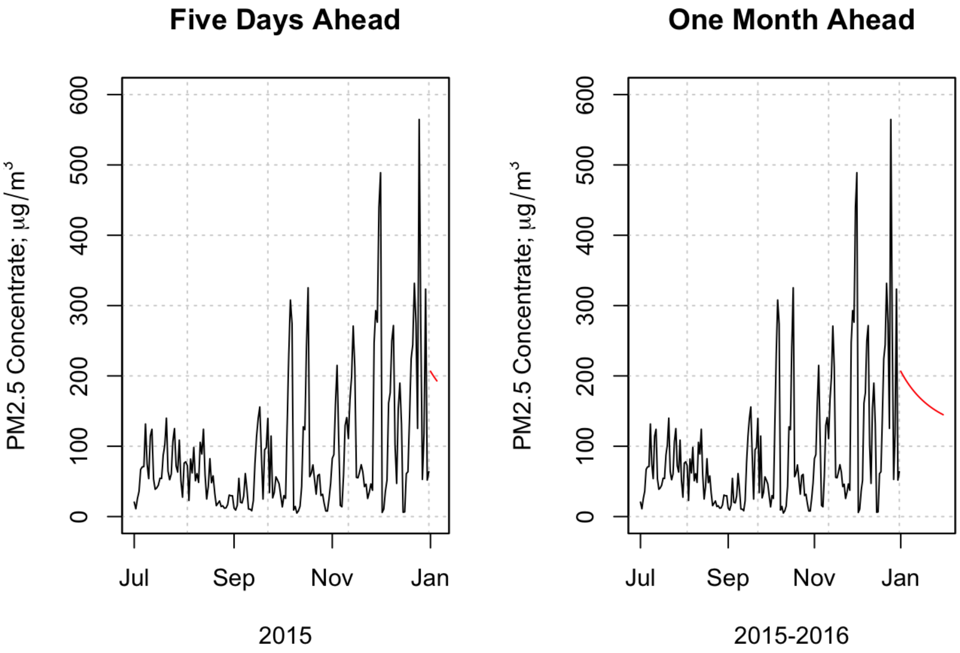


From the ACF and PACF plots, we can derive a starting point for the order of our ARMA-GARCH model. The ACF cuts off at lag 1, while the PACF cuts off at lag 2 for both the regular and squared data. An ARMA(2,2)-ARCH(2) model was the first to be explored as there were autoregressive and moving average components based on the ACF/PACF plots of the data, and the squared data showed potential signs of an autoregressive component (R. Kumar, 2020). While the squared data also showed signs of a moving average component, we started with just the autoregressive, but would incorporate a moving-average component (GARCH) later.

Our first model had issues, such as many insignificant parameters, and autocorrelation issues based on results from the Box Test. The first thing we tried was removing the least significant parameter (MA2) and tried a new model using an ARMA(2,1)-ARCH(2). This model suffered the same fate as the first one, so again we removed the least significant parameter (AR2). The third model we tried was ARMA(1,1)-ARCH(2). While this model did not have any insignificant parameters, there were autocorrelation issues shown via the Box Test. At this point, we decided to include the moving average component for the variance. Our fourth attempt was with an ARMA(1,1)-GARCH(2,1) model. The autocorrelation issues were fixed, but the second ARCH term became insignificant. To remedy this, we removed the insignificant parameter, and would later add a second GARCH term, leading to the final model of ARMA(1,1)-GARCH(1,2).

Results

When calculating the RMSEs for each of the five models (see RMSE table), we noticed that the five-day ahead RMSE values were not significantly different between the SARIMA and the ARMA-GARCH model. The ARMA(1,1)-GARCH(1,2) model had the smallest RMSE over the five-day prediction period. We made two forecasts and like the SARIMA model, did five days and one month ahead. The results were quite different from the predictions made using the SARIMA model. Here, there is a gradual decline in the predictions rather than a sudden drop.



At first, this appears strange given the prediction results from the SARIMA model, as the two are drastically different. However, when we thought about it, the results made sense. A sudden drop off in PM2.5 particulate levels is not realistic, a gradual decline is.

Discussion

Literature Review

From collaborative efforts demonstrated in works like Zhang. Et al(2005) and Lindsay and Xu (2018), we can see there is a clear link between PM2.5 exposure and serious long-term health issues, specifically in the respiratory system. And this is the main motivation and the starting point of this paper. After the exploratory studies and model building, it was certain that seasonality exists, but we needed more solid works to explain such a phenomenon. One of the studies done by Dan. et al. (2004) pointed out that coal heating could significantly increase the PM2.5 concentration level during the cold seasons, which are winter and early spring, as would the long-range transport of dust, which is typical during the winter season for Beijing.

Conclusion

There is a strong seasonality for the PM2.5 level, with winter and early spring the worst (PM2.5 concentration levels are the highest) and summers the best (PM2.5 concentration levels are the lowest). This is consistent with the study done by Dan. et al. (2004) that is mentioned above. These findings help answer the data analysis part of the project goal, as it could give people a better understanding of what the main contributors are to PM2.5 and what some possible ways are to reduce it based on these main factors. For instance, rather than burning coal, Beijing could consider a cleaner way of heating, such as gas heating; or the city could find ways like having more trees to stop the long-range transport of dust from other places during the windy winter season.

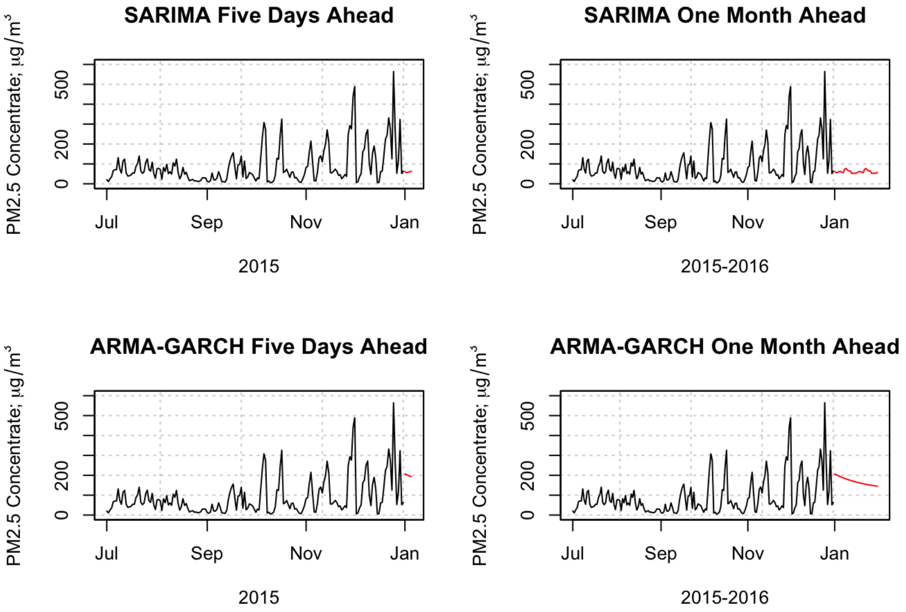
As for the modeling part of the project, the best model we have for SARIMA is SARIMA(0,2,2) x $(0,1,1)_{14}$, and the best ARMA-GARCH model is ARMA(1,1) - GARCH(1,2), which is also the best overall model with the RMSE of 4.8 for the next-month-prediction, and

1.75 for the next five-day prediction. This conclusion might seems odd considering the RMSE table, where the RMSE of next-moth-prediction for SARIMA models is around 2.7. However, since the immediate five-day prediction is the most valuable to users, having the lowest possible RMSE for the five-day prediction is the most crucial factor to consider.

Model	**Five-Day RMSE**	**One-Month RMSE**
SARMA(0,0,2)×(0,1,1)$_{14}$	1.7617	2.7543
ARMA(1,1)-GARCH(1,2)	1.7521	4.8082

(RMSE table)

Additionally, from the prediction plots, we also observe that the ARMA-GARCH model preserves the relatively high starting point, rather than immediately dropping to less than 100 in the SARIMA plots, which is more realistic considering the previous patterns.



We are pleased with the results as they achieve both main goals, education, and prediction. Not only does this project better educates people about what PM2.5 is, what its main contributors are, or the seasonality pattern it is showing with clear plots and numerous pieces of literature, but it is also capable of making relatively accurate 5-day prediction, with RMSE of

1.76 out of a range from 3 to 564. This is accurate enough to give people valuable information and help them to avoid excess exposure by either not going outside or wearing protective gear such as N95 masks.

Future Steps

As for the future, we would investigate why the ARMA-GARCH model has a better 5-day prediction while the 1-month prediction is worse than the SARIMA model, which might in turn improve the 5-day prediction of the ARMA-GARCH model. Additionally, we could utilize more advanced models like MSARIMA (Multiple Seasonal ARIMA), as there are other seasonal predictors in the original dataset (temperature). Also, comparing the model for different stations might be another interesting research topic, as it would highlight the differences among locations, which would not only give people more accurate predictions but also guide more targeted policy makings to improve the air quality in terms of the PM2.5 concentration level.

Citations

Dan, M., Zhuang, G., Li, X., Tao, H., & Zhuang, Y. (2004, July). *The characteristics of carbonaceous species and their sources in PM2.5 in Beijing*. Atmospheric Environment Volume 38, Issue 21. https://www.sciencedirect.com/science/article/pii/S1352231004002432

Kumar, R. K. (2020, January 14). *Time Series model(s) - arch and garch*. Medium. Retrieved November 29, 2022, from https://medium.com/@ranjithkumar.rocking/time-series-model-s-arch-and-garch-2781a982b448

Mei Zheng, Lynn G. Salmon, James J. Schauer, Limin Zeng, C.S. Kiang, Yuanhang Zhang, Glen R. Cass, "Seasonal trends in PM2.5 source contributions in Beijing, China" *Atmospheric Environment,* Volume 39, Issue 22, 2005, Pages 3967-3976, ISSN 1352-2310, https://doi.org/10.1016/j.atmosenv.2005.03.036.

Miller, Lindsay, and Xiaohong Xu. "Ambient PM2.5 Human Health Effects-Findings in China and Research Directions." *MDPI*, Multidisciplinary Digital Publishing Institute, 30 Oct. 2018, https://www.mdpi.com/2073-4433/9/11/424.

Appendix: R code

Custom Functions:

```
#Custom Function for Creating Training and Testing Data
ts.train.test <- function(data, freq, p = 0.75){
total.length = length(data)
#Splitting up the Data
test.length = round(total.length * (1 - p), 0)
train.length = total.length - test.length
data.test = data[train.length:total.length]
data.train = data[1:(train.length - 1)]
#Coercing the data into time series format
data.test = ts(data.test, start = time(data)[train.length], frequency = freq)
data.train = ts(data.train, start = time(data)[1], frequency = freq)
#Returning a list of the training and testing data
x = list(data.train, data.test)
names(x) <- c("train","test")
return(x)
}
#Custom Function for Calculating RMSE
tsRMSE <- function(data, freq, prob = 0.75, p = 0, d = 0, q = 0, P = 0, D = 0, Q = 0, S = 1){
  #Creating the training and testing splits
  train = ts.train.test(data, freq, p = prob)$train
  test = ts.train.test(data, freq, p = prob)$test
  #Forecast modeling
  model = sarima.for(train, p, d, q, P, D, Q, S, no.constant = TRUE,
                     n.ahead = length(test), plot = FALSE)
  #RMSE Calculations
  RMSE = sqrt(mean((model$pred - as.numeric(test))^2))
  RMSE1.5 = sqrt(mean((model$pred[1:5] - as.numeric(test[1:5]))^2))
  #Returning a list of the RMSE values
  x = list(RMSE, RMSE1.5)
  names(x) <- c("RMSE", "RMSE Obs. 1:5")
  return(x)
}
#Custom Function for Calculating RMSE for GARCH
garchRMSE <- function(data, freq, prob = 0.75, p = 0, q = 0, alpha = 0, beta = 0){
  #Creating the training and testing splits
  train = ts.train.test(data, freq, p = prob)$train
  test = ts.train.test(data, freq, p = prob)$test
  #Forecast modeling
  model = garchFit(substitute(~arma(p,q) + garch(alpha, beta)), train)
  pred = predict(model, n.ahead = length(test))
  #RMSE Calculations
  RMSE = sqrt(mean((as.numeric(pred$meanForecast) - as.numeric(test))^2))
  RMSE1.5 = sqrt(mean((as.numeric(pred$meanForecast[1:5]) - as.numeric(test[1:5]))^2))
  #Returning a list of the RMSE values
  x = list(RMSE, RMSE1.5)
  names(x) <- c("RMSE", "RMSE Obs. 1:5")
  return(x)
}
```

Data Cleaning:

```
Beijing <- read.csv("~/Desktop/Courses/STAT429 (UIUC)/Project/Data Sets/BeijingPM20100101_20151231.csv", stringsAsFactors=TRUE)

#Going by each recording station

Dongsi = Beijing %>% drop_na(PM_Dongsi)

Dongsihuan = Beijing %>% drop_na(PM_Dongsihuan)

Nongzhanguan = Beijing %>% drop_na(PM_Nongzhanguan)

USPost = Beijing %>% drop_na(PM_US.Post)


#Looking for Columns with NA

result = data.frame(matrix(data = rep(0,5*ncol(Beijing)), nrow = ncol(Beijing)))

for(i in 1:ncol(Beijing)){

  result[i,1] = any(is.na(Beijing[,i]))

  result[i,2] = any(is.na(Dongsi[,i]))

  result[i,3] = any(is.na(Dongsihuan[,i]))

  result[i,4] = any(is.na(Nongzhanguan[,i]))

  result[i,5] = any(is.na(USPost[,i]))

}

#Outputting Results

#Each row is a column in the data while the columns represent the data sets.
```

```
result = result %>% rename(Beijing = "X1") %>% rename(Dongsi = "X2") %>%
       rename(Dongsihuan = "X3") %>% rename(Nongzhanguan = "X4") %>%
       rename(USPost = "X5") %>%
       mutate(Beijing = ifelse(Beijing == 0, "No", "Yes")) %>%
       mutate(Dongsi = ifelse(Dongsi == 0, "No", "Yes")) %>%
       mutate(Dongsihuan = ifelse(Dongsihuan == 0, "No", "Yes")) %>%
       mutate(Nongzhanguan = ifelse(Nongzhanguan == 0, "No", "Yes")) %>%
       mutate(USPost = ifelse(USPost == 0, "No", "Yes"))
rownames(result) = colnames(Beijing)
result


#Dongsi
Dongsi$Date = as.Date(with(Dongsi,paste(day,month,year,sep = "-")), "%d-%m-%Y")
Dongsi$Recorded = as.POSIXct(paste(Dongsi$Date, Dongsi$hour), format = "%Y-%m-%d %H")
Dongsi = Dongsi %>%
  select(Date, Recorded, season, PM_Dongsi, DEWP, HUMI, PRES, TEMP, cbwd, Iws, precipitation,
       Iprec)
D = aggregate(PM_Dongsi ~ Date, Dongsi, mean)
```

Preliminary Plotting:

```
#Dongsi
tsplot(D$Date, D$PM_Dongsi, type = "l", xlab = "Date Recorded",
    ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"), main = "Dongsi")


#Smoothed Periodogram
smooth = mvspec(D$PM_Dongsi, spans = 15, col = "blue", lwd = 2)
par(mfrow = c(2,3))
```

```
#Dongsi
tsplot(D$Date, D$PM_Dongsi, type = "l", xlab = "Date Recorded",
    ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"), main = "Dongsi")
acf(D$PM_Dongsi, main = "")
pacf(D$PM_Dongsi, ylab = "PACF", main = "")
#Log transformation of original data
tsplot(D$Date, log(D$PM_Dongsi), type = "l", xlab = "Date Recorded",
    ylab = TeX(r"(log(PM2.5 Concentrate); $\mu g/m^3$)"), main = "Dongsi")
acf(log(D$PM_Dongsi), main = "")
pacf(log(D$PM_Dongsi), ylab = "PACF", main = "")
par(mfrow = c(3,1))
#Adding 2 Week Difference
tsplot(diff(log(D$PM_Dongsi), 14),
      type = "l", xlab = "Date Recorded",
      ylab = TeX(r"(log(PM2.5 Concentrate); $\mu g/m^3$)"), main = "Dongsi")
acf(diff(log(D$PM_Dongsi, 14)), main = "", lag.max = 100)
pacf(diff(log(D$PM_Dongsi, 14)), ylab = "PACF", main = "", lag.max = 100)
```

Stationary Testing for SARIMA:

```
adf.test(diff(log(D$PM_Dongsi), 14))
kpss.test(diff(log(D$PM_Dongsi), 14))
```

Model Building for SARIMA:

```
#Original Idea
d1 = sarima(log(D$PM_Dongsi), p = 0, d = 0, q = 3, P = 0, D = 1, Q = 1, S = 14,
          no.constant = TRUE)


#3rd AR NOT significant
```

```
d2 = sarima(log(D$PM_Dongsi), p = 0, d = 0, q = 2, P = 0, D = 1, Q = 1, S = 14,
        no.constant = TRUE)


#Adding P = 1 to see what happens
d3 = sarima(log(D$PM_Dongsi), p = 0, d = 0, q = 2, P = 1, D = 1, Q = 1, S = 14,
        no.constant = TRUE)
```

RMSE Calculations SARIMA:

```
#Original Idea
RMSE1 = tsRMSE(log(D$PM_Dongsi), 365, prob = 0.8, q = 3, D = 1, Q = 1, S = 14)
#3rd AR NOT significant
RMSE2 = tsRMSE(log(D$PM_Dongsi), 365, prob = 0.8, q = 2, D = 1, Q = 1, S = 14)
#Adding P = 1 to see what happens
RMSE3 = tsRMSE(log(D$PM_Dongsi), 365, prob = 0.8, q = 2, P = 1, D = 1, Q = 1, S = 14)
#3rd AR NOT significant
Box.test(d2$fit$residuals, lag = 10, fitdf = 0, type = "Lj")

par(mfrow = c(2,2))
acf(log(D$PM_Dongsi), main = "")
pacf(log(D$PM_Dongsi), ylab = "PACF", main = "")
#Squared Log transformation of original data
acf(log(D$PM_Dongsi)^2, main = "")
pacf(log(D$PM_Dongsi)^2, ylab = "PACF", main = "")
```

ARMA Model Building:

```
#ARMA(2,2)-ARCH(2)
d4 = garchFit(~arma(2,2) + garch(2,0), log(Dongsi$PM_Dongsi))
#ARMA(2,1)-ARCH(2)
d5 = garchFit(~arma(2,1) + garch(2,0), log(Dongsi$PM_Dongsi))
#ARMA(1,1)-ARCH(2)
d6 = garchFit(~arma(1,1) + garch(2,0), log(Dongsi$PM_Dongsi))
```

```
#ARMA(1,1)-GARCH(2,1)

d7 = garchFit(~arma(1,1) + garch(2,1), log(Dongsi$PM_Dongsi))

#ARMA(1,1)-GARCH(1,2)

d8 = garchFit(~arma(1,1) + garch(1,2), log(Dongsi$PM_Dongsi))
```

RMSE Calculations for ARMA:

```
#ARMA(2,2)-ARCH(2)

RMSE4 = garchRMSE(log(Dongsi$PM_Dongsi), freq = 365, p = 2, q = 2, alpha = 2)

#ARMA(2,1)-ARCH(2)

RMSE5 = garchRMSE(log(Dongsi$PM_Dongsi), freq = 365, p = 2, q = 1, alpha = 2)

#ARMA(1,1)-ARCH(2)

RMSE6 = garchRMSE(log(Dongsi$PM_Dongsi), freq = 365, p = 1, q = 1, alpha = 2)

#ARMA(1,1)-GARCH(2,1)

RMSE7 = garchRMSE(log(Dongsi$PM_Dongsi), freq = 365, p = 1, q = 1, alpha = 2, beta = 1)

#ARMA(1,1)-GARCH(1,2)

RMSE8 = garchRMSE(log(Dongsi$PM_Dongsi), freq = 365, p = 1, q = 1, alpha = 1, beta = 2)
```

Predictions:

```
par(mfrow = c(1,2))

#5 Days Ahead

pred3 = predict(d8, n.ahead = 5)

plot(D$Date, rep(0,nrow(D)), col = "white", xlim = c(D$Date[1], as.Date("2016-01-05")),

    ylim = c(0,600), xlab = "Date", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),

    main = "Five Days Ahead")

grid()

box()

lines(D$Date, D$PM_Dongsi)
lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-05"), "day"),
exp(pred3$meanForecast),

     col = "red")

#1 Month Ahead

pred4 = predict(d8, n.ahead = 31)

plot(D$Date, rep(0,nrow(D)), col = "white", xlim = c(D$Date[1], as.Date("2016-01-31")),

    ylim = c(0,600), xlab = "Date", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),

    main = "One Month Ahead")

grid()

box()

lines(D$Date, D$PM_Dongsi)

lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-31"), "day"),
exp(pred4$meanForecast),

     col = "red")

par(mfrow = c(1,2))

#5 Days Ahead

plot(D$Date[893:1076], rep(0,184), col = "white", xlim = c(D$Date[893],
as.Date("2016-01-05")),

    ylim = c(0,600), xlab = "2015", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),

    main = "Five Days Ahead")

grid()

box()

lines(D$Date[893:1076], D$PM_Dongsi[893:1076])

lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-05"), "day"),
exp(pred3$meanForecast),

     col = "red")

#1 Month Ahead

plot(D$Date[893:1076], rep(0,184), col = "white", xlim = c(D$Date[893],
as.Date("2016-01-31")),
    ylim = c(0,600), xlab = "2015-2016", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),
    main = "One Month Ahead")
grid()
box()
lines(D$Date[893:1076], D$PM_Dongsi[893:1076])
lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-31"), "day"),
exp(pred4$meanForecast),
     col = "red")
Model Prediction Comparison
#SARIMA
par(mfrow = c(2,2))
#5 Days Ahead
plot(D$Date[893:1076], rep(0,184), col = "white", xlim = c(D$Date[893],
as.Date("2016-01-05")),
    ylim = c(0,600), xlab = "2015", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),
    main = "SARIMA Five Days Ahead")
grid()
box()
lines(D$Date[893:1076], D$PM_Dongsi[893:1076])
lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-05"), "day"), exp(pred1$pred),
     col = "red")
#1 Month Ahead
plot(D$Date[893:1076], rep(0,184), col = "white", xlim = c(D$Date[893],
as.Date("2016-01-31")),
    ylim = c(0,600), xlab = "2015-2016", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),
    main = "SARIMA One Month Ahead")
grid()
box()
lines(D$Date[893:1076], D$PM_Dongsi[893:1076])
lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-31"), "day"), exp(pred2$pred),
     col = "red")
#ARMA-GARCH
#5 Days Ahead
plot(D$Date[893:1076], rep(0,184), col = "white", xlim = c(D$Date[893], as.Date("2016-01-05")),
    ylim = c(0,600), xlab = "2015", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),
    main = "ARMA-GARCH Five Days Ahead")
grid()
box()
lines(D$Date[893:1076], D$PM_Dongsi[893:1076])
lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-05"), "day"), exp(pred3$meanForecast),
     col = "red")
#1 Month Ahead
plot(D$Date[893:1076], rep(0,184), col = "white", xlim = c(D$Date[893], as.Date("2016-01-31")),
    ylim = c(0,600), xlab = "2015-2016", ylab = TeX(r"(PM2.5 Concentrate; $\mu g/m^3$)"),
    main = "ARMA-GARCH One Month Ahead")
grid()
box()
lines(D$Date[893:1076], D$PM_Dongsi[893:1076])
lines(seq.Date(as.Date("2016-01-01"), as.Date("2016-01-31"), "day"), exp(pred4$meanForecast),
     col = "red")
```