Connor Martins

Professor Kurt

EC 382

4 May 2023

A Time Series Analysis of American Flight Delays

The main variable of interest in the time series analysis is the difference, in minutes, between a given American plane's scheduled departure time and its actual departure time. The variable can be positive, negative, or zero, and only reflects the delays of domestic American flights. Because so many flights occur in a single day, the data will be grouped by each day in the original dataset, and the observation will reflect the average delay of flights on that particular day. Furthermore, that average will be taken from a random sample of one hundred flights for each day in the dataset to relieve the pressure on the processing power of the computer being used. The data were arranged into comma separated values files and published for download on Kaggle, a popular site for publishing datasets. Its original source is the Bureau of Transportation Statistics, and the links to both sites are included below:

Kaggle Link

OST Link

Methods	Yes/No
Basic plot of your data, comment on its properties	**Y**
Time series decomposition (seasonality, trend, cyclicality)	Y
Smoothing method (SMA) (if your data is very granular like daily, you may want to implement this)	Y

Stationarity analysis: decide whether you should log your data or take first difference of it for your analysis.	
Unit root test (Dickey Fuller)	Y
Autocorrelations, partial autocorrelations, lag plots	Y
Regression analysis	
Post-estimation: Construct Heteroskedasticity and Autocorrelation Consistent Covariance Matrix	
Hypothesis Testing (Linear hypothesis testing or F-test via anova)	
AR-MA models	Y
Jarque Bera Test	
Forecasting with AR-MA models	Y
AIC, BIC Information Criteria, Model Comparisons	Y
VAR - Impulse response functions	

I. Plotting the Time Series

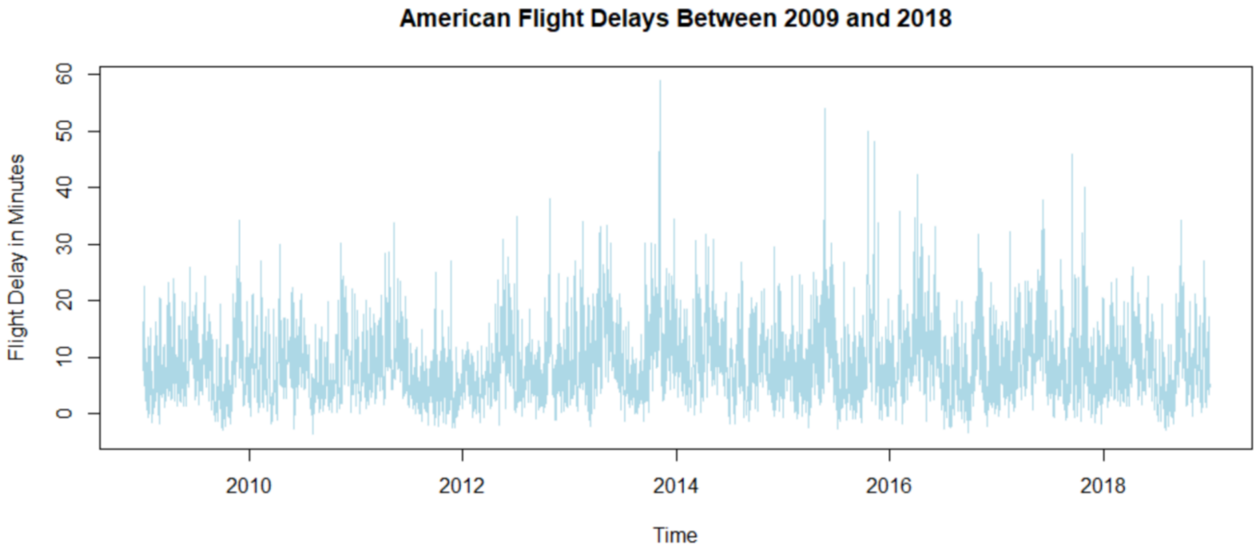

Figure 1: Plot of the Time Series Variable

The time series at first glance is somewhat difficult to interpret because it is so granular. The variable is recorded with a daily frequency. However, there are observable spikes towards the end of years and troughs in the months leading up to that period, suggesting the possibility of seasonality in the series. It is also worth noting that those spikes were higher in later years, though they decreased noticeably in 2018.

II. Conducting a Time Series Decomposition

Examining the plot of the time series revealed evidence of potential seasonality, so a complete decomposition was conducted in R. This process attempts to graphically distinguish between the seasonal and random components of a time series and its actual long-term trend if one exists. The preliminary results are shown in Figure 2.

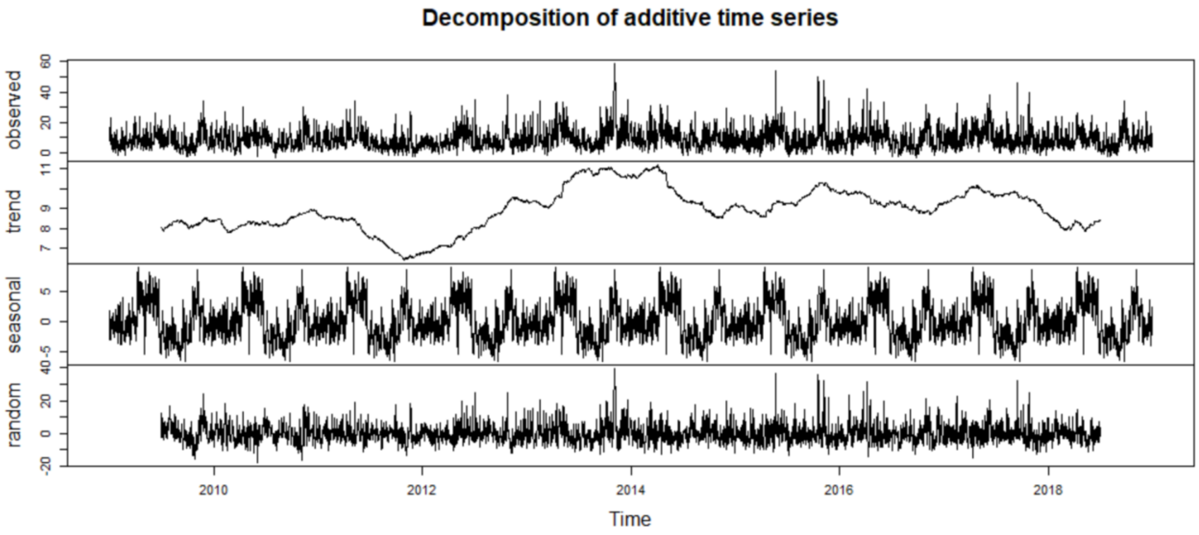

Figure 2: Decomposition of the Flight Delay Time Series

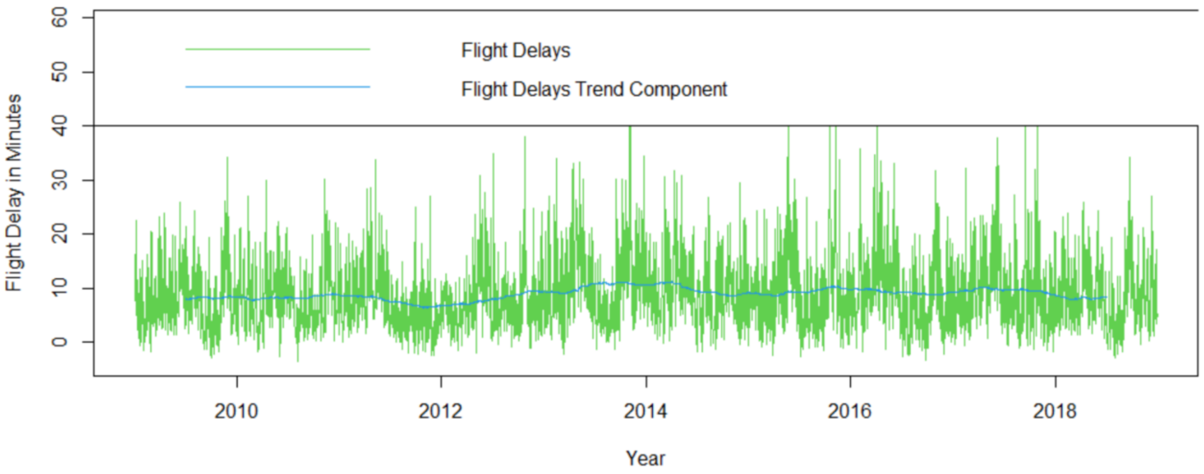

Figure 3: Plot of the Additive Decomposition Trend Component with the Time Series

The time series decomposition shows clear evidence of a seasonal component to the data characterized by spikes in summer months and in December. This is logical since airlines are likely overwhelmed by summer vacation travelers and the travel demand during the holiday season at the end of the year. Additionally, the trend component clearly lacks the significantly high spikes in certain times of the year, as demonstrated by Figure 3. Estimating the seasonal component of the time series allows for the generation of a seasonally adjusted series, which is plotted in Figure 4.

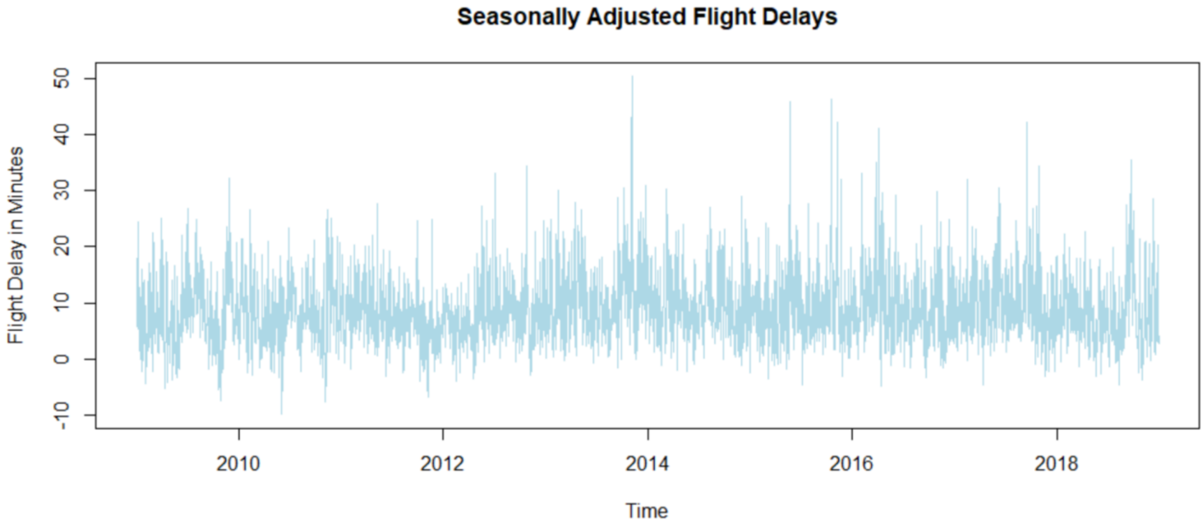

Figure 4: Plot of the Seasonally Adjusted Flight Delays Series

Despite adjusting for seasonality, the plot of the time series is still highly granular and difficult to interpret. One improvement was a reduction in the magnitude of the seasonal spikes observed in the original plot of the time series. However, the overall shape of the series remained similar. The generation of this plot did not allow for any conclusions about a long-term trend to be made. It was concluded that a smooth-moving average plot should be generated to develop a better understanding of the data.

III. Plotting the Smooth-Moving Average of the Time Series

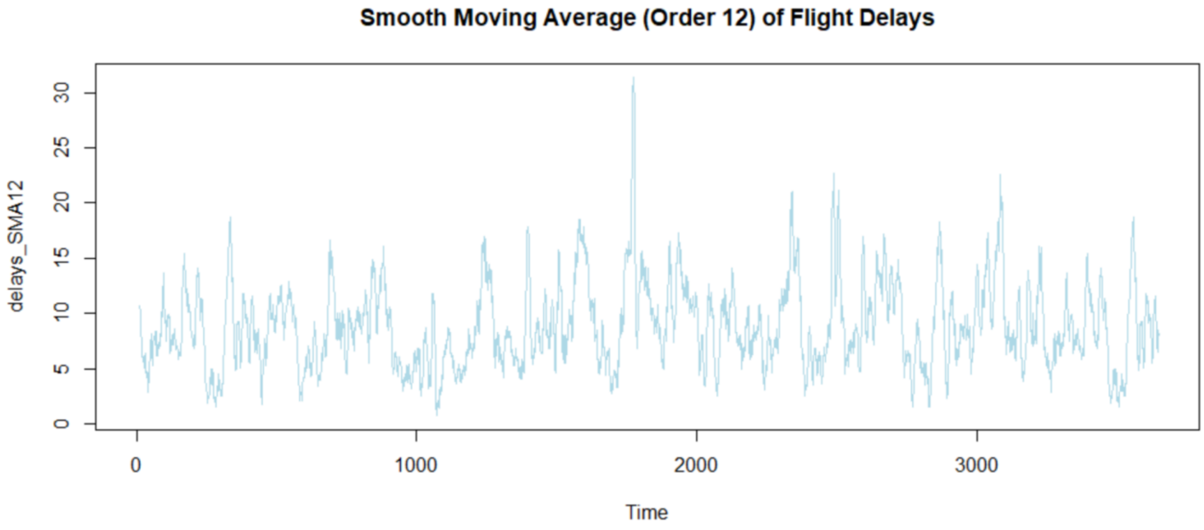

Figure 5: Plot of the SMA (Order 12) of the Time Series

A smooth-moving average (SMA) variable of order 12 was generated in R and plotted in Figure 5. While this technique certainly improves the readability of the data, the series still seems to lack a long-term trend, which was also the conclusion from the additive decomposition analysis. However, the additive decomposition analysis generated a much smoother graph of the series' trend component, making it the more useful technique for this time series.

IV. Conducting an Augmented Dickey Fuller Test for a Unit Root

The flight delays time series was tested for having a unit root, a statistical property that indicates a time series is non-stationary. The unit root test technique was an augmented Dickey-Fuller (ADF) test. Based on the shape of the data and the clear evidence of seasonality within the series, there is suspicion that the time series hovers around the same mean and could potentially be at least weakly stationary. Conducting a unit root test helps to eliminate or not eliminate the null hypothesis that the series is non-stationary. It is important to note that the ADF test was conducted on the first difference of the time series, mainly because it contains many negative values and would not be suitable for taking a natural log, which is the other common technique for conducting an ADF test of a time series. The coefficient of interest is on the first difference of the time series. The coefficient has a test statistic of -65.755 and a p-value of 2e-16, which indicates significance at all levels. Therefore, the null hypothesis of non-stationarity of the first difference of the time series can be rejected. In other words, it can be concluded based on the Augmented Dickey-Fuller test that the time series does not have a unit root and does not seem to have any time-dependent structure or trend.

V. Plotting the Autocorrelation and Partial Autocorrelation Functions of the Time Series

The autocorrelation function and partial autocorrelation function of the first difference of the time series were plotted and the results are shown in Figures 6 and 7.

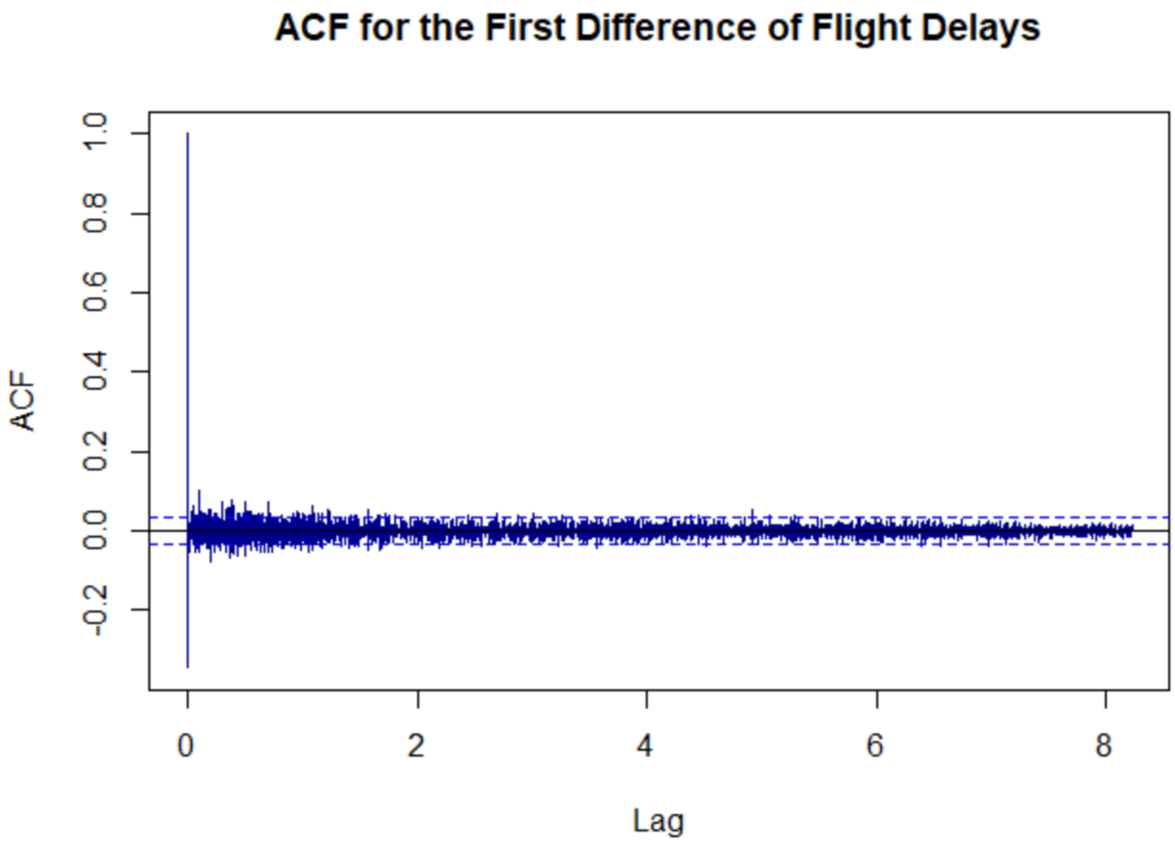

Figure 6: Plot of the ACF of the Time Series' First Difference

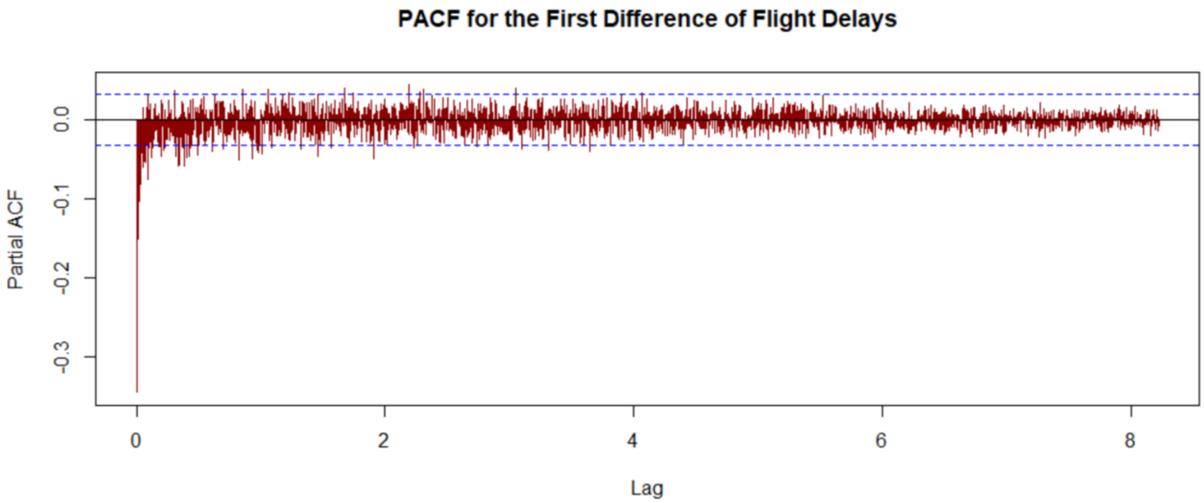

Figure 7: Plot of the PACF of the Time Series' First Difference

As shown in the plots, there are very few significant instances of autocorrelation or partial autocorrelation in the first difference of the time series. That being said, the ACF shocks tend to oscillate between negative and positive, as evidenced by the slightly significant, positive ACF shocks around the first, third, and fifth lags, along with the slightly significant, negative ACF shocks around the second and fourth lags. The PACF follows a similar trend, exhibiting slightly significant positive shocks close to the third lag and slightly significant negative shocks close to the first and second lags. Out of curiosity and because of the granular nature of the daily average, ACF and PACF's were plotted for the first difference of flight delays aggregated by month. The plots are shown in Figures 8 and 9 and are more readable than the daily data plots.

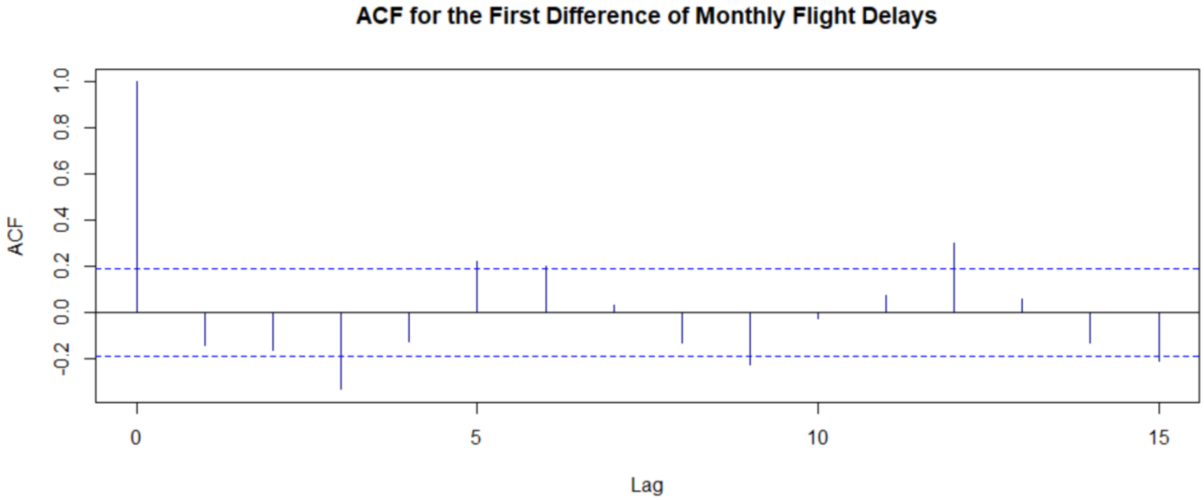

Figure 8: Plot of the ACF of the Monthly Time Series' First Difference

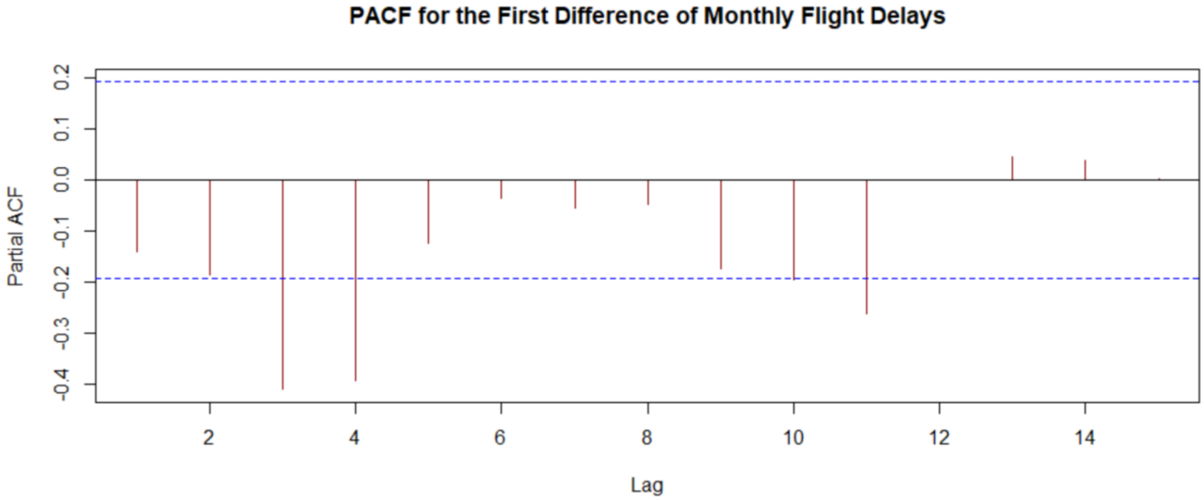

Figure 9: Plot of the PACF of the Monthly Time Series' First Difference

The plot of the monthly series' first difference's ACF shows a significant negative shock close to the fourth lag and a significant positive shock just past the twelfth lag. The PACF shows significant negative shocks at lags three and four, along with another negative shock at lag eleven.

VI. Fitting AR, MA, and ARMA Models to Attempt Forecasting of the Time Series

AR(2), MA(2), and ARMA(15,15) models were fitted to the time series, aggregated by month, for comparison. The monthly frequency was chosen because of the granularity of the daily data and to increase the readability of the forecast plot. To show how well the forecast models fit the series, the five-step-ahead forecasts were plotted along with the observed series. Finally, the models were compared using AIC and BIC. The three forecast plots, with ninety-five percent confidence intervals included, are shown in Figures 10-12.

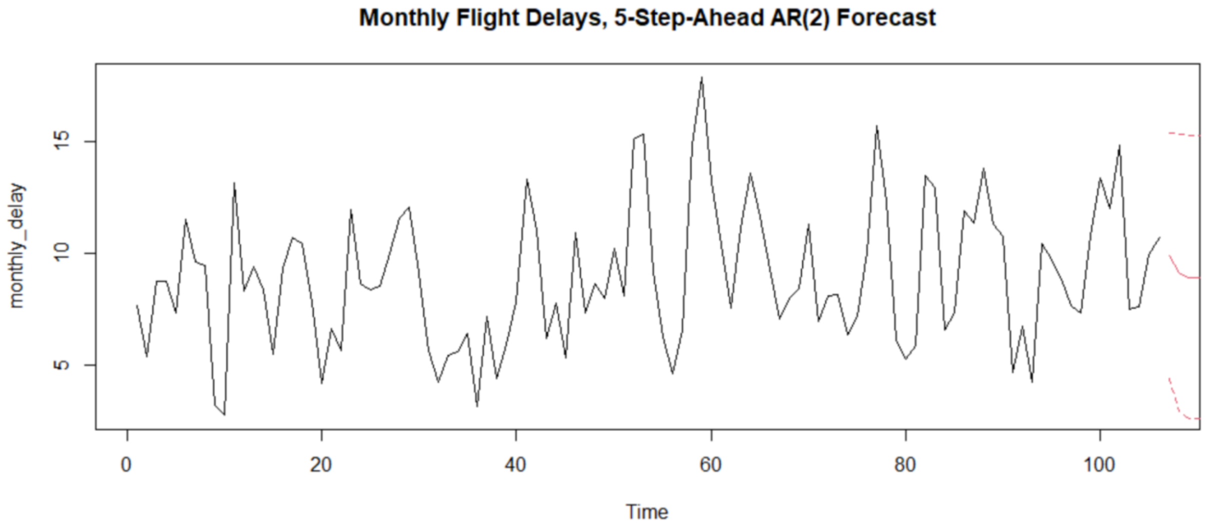

Figure 10: Plot of the AR(2) 5-Step Ahead Forecast and the Time Series

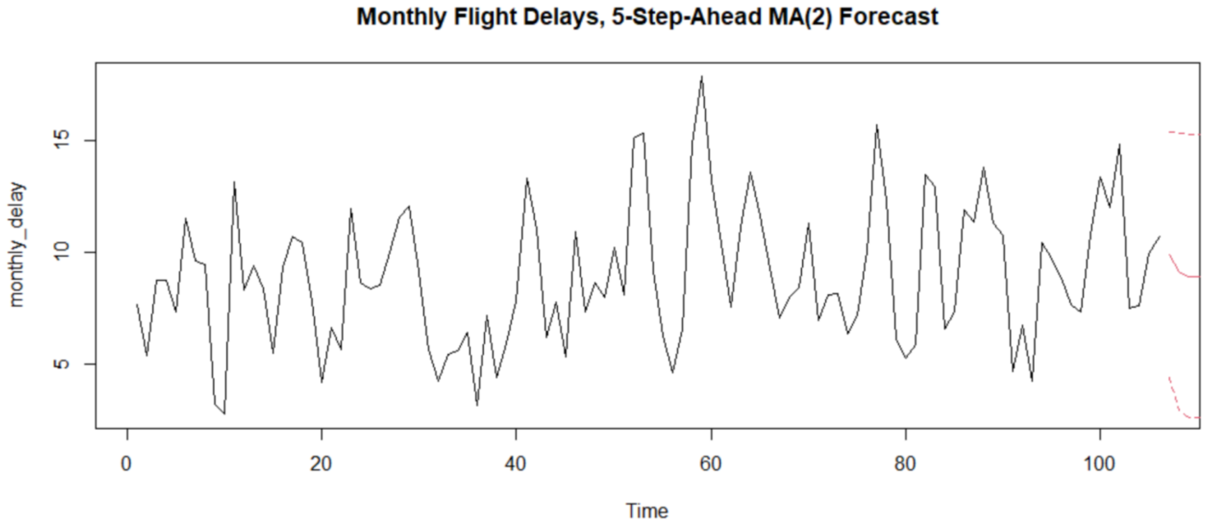

Figure 11: Plot of the MA(2) 5-Step Ahead Forecast and the Time Series

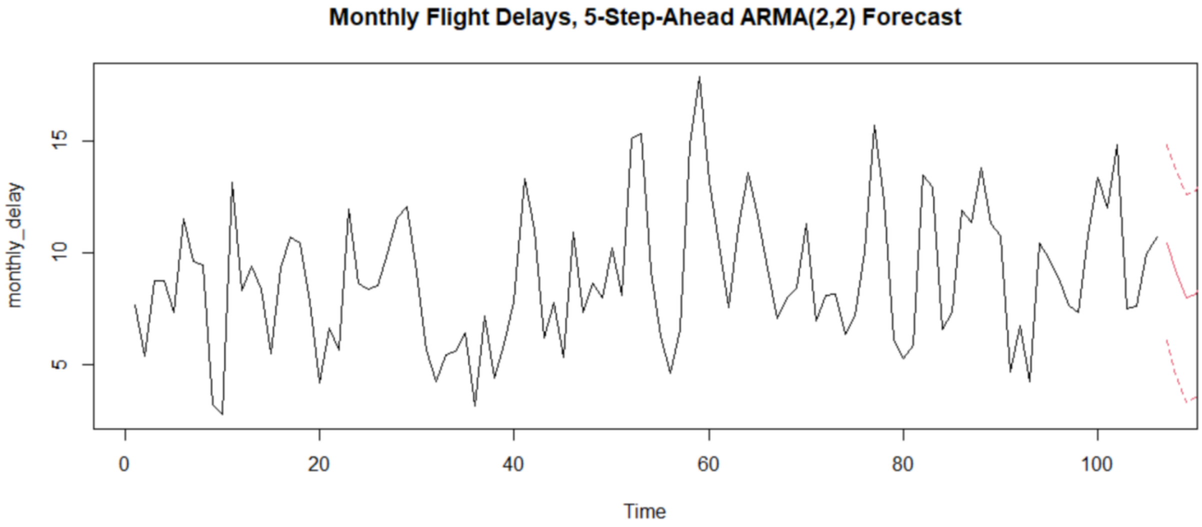

Figure 12: Plot of the ARMA(2,2) 5-Step Ahead Forecast and the Time Series

As shown in the figures, both the AR(2) and MA(2) five-step-ahead forecasts predicted similar monthly flight delay movement with very wide error bands. The ARMA(2,2) model predicted a sharp decline in the average monthly flight delay with a slightly narrower error band. To further compare the three models, AIC and BIC information criteria values were computed for each model. The ARMA(2,2) model had the lowest value for both the AIC and BIC information criteria (517.02 and 532.99 respectively). However, it only marginally outperformed the other two models in the BIC criteria, with a point-value separation of just 0.25 between it and the MA(2) model. The ARMA(2,2) outperformed the AR(2) model by an AIC information criteria point-value separation of 5.57, with the separation only slightly larger between it and the MA(2) model. Therefore, the ARMA(2,2) is the best choice out of the three for the five-step-ahead forecast of the average monthly flight delay.

```r
# connor martins
# ec 382 term project - flight delays time series analysis
# importing and cleaning the data into one dataframe:

library(data.table)

year_files <- list("C:\\Users\\cmart\\Documents\\EC 382\\project\\2009.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2010.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2011.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2012.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2013.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2014.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2016.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2017.csv",
                "C:\\Users\\cmart\\Documents\\EC 382\\project\\2018.csv")

sample_size <- 100


# Define a function to read and sample data from each file
read_and_sample <- function(file) {
  dt <- fread(file)
  dt[, .(DEP_DELAY = sample(DEP_DELAY, min(sample_size, .N), replace = TRUE))
, by = .(FL_DATE)]
}

# Read and sample data from each file
all_results <- rbindlist(lapply(year_files, read_and_sample), use.names = TRU
E, fill = TRUE)

# Convert the flight delay column to numeric
all_results[, DEP_DELAY := as.numeric(DEP_DELAY)]

# Compute the mean delay for each day
daily_mean <- all_results[, .(mean_delay = mean(DEP_DELAY, na.rm = TRUE)), by
= .(FL_DATE)]


# daily_mean will serve as the main dataframe from which the TS variable will
be generated:
delay <- ts(daily_mean$mean_delay, frequency = 365, start = c(2009, 1), end =
c(2018, 365))

# plot the time series:
plot.ts(delay, main = "American Flight Delays Between 2009 and 2018", col = "
lightblue",
        xlab = "Time", ylab = "Flight Delay in Minutes")
```

```r
# time series decomposition due to evidence of seasonality:
delays_composed <- decompose(delay)
plot(delays_composed)

# plot the original series along with its trend component from the decomp:
ts.plot(delay, delays_composed$trend, gpars=list(xlab="Year", ylab="Flight Delay in Minutes", col = 3:4))
legend("topleft", legend = c("Flight Delays", "Flight Delays Trend Component"), col = 3:4, lty = 1)

# plotting the seasonally adjusted TS:
delays_seasonally_adj <- delay - delays_composed$seasonal
plot(delays_seasonally_adj, main = "Seasonally Adjusted Flight Delays",
     col = "lightblue",
     xlab = "Time", ylab = "Flight Delay in Minutes")

# plotting the SMA of the TS:
library("TTR")

delays_SMA3 <- SMA(delay,n=3)
plot.ts(delays_SMA3, main = "Smooth Moving Average (Order 3) of Flight Delays",
     col = "lightblue")

delays_SMA9 <- SMA(delay,n=9)
plot.ts(delays_SMA9, main = "Smooth Moving Average (Order 9) of Flight Delays",
        col = "lightblue")

delays_SMA12 <- SMA(delay,n=12)
plot.ts(delays_SMA12, main = "Smooth Moving Average (Order 12) of Flight Delays",
        col = "lightblue")

# conducting an ADF test for a unit root:
library(tseries)

Log_Delay <- ts(log(delay)) # generates NaNs because you cannot take natural log of negative values

## Warning in log(delay): NaNs produced

FirstDiff_Delay <- diff(delay) # the first difference of the delay TS will be used for the ADF test
adf.test(FirstDiff_Delay)

adf_output <- adf.test(FirstDiff_Delay)

## Warning in adf.test(FirstDiff_Delay): p-value smaller than printed p-value

adf_output$critical[,"1%"]

## NULL
```

```
# adf.test() not returning a critical ADF statistic, so use urca:
library(urca)

adf_urca_output <- ur.df(FirstDiff_Delay, type = "trend", selectlags = "AIC")
summary(adf_urca_output)

## 
## ############################################### 
## # Augmented Dickey-Fuller Test Unit Root Test # 
## ############################################### 
## 
## Test regression trend 
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -24.360  -4.239  -0.422   3.670  37.558 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2.914e-03  2.229e-01  -0.013    0.990    
## z.lag.1     -1.726e+00  2.601e-02 -66.361   <2e-16 ***
## tt          -1.966e-07  1.058e-04  -0.002    0.999    
## z.diff.lag   2.860e-01  1.587e-02  18.017   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.727 on 3643 degrees of freedom
## Multiple R-squared:  0.6979, Adjusted R-squared:  0.6977 
## F-statistic:  2805 on 3 and 3643 DF,  p-value: < 2.2e-16
## 
## 
## Value of test-statistic is: -66.3614 1467.948 2201.921 
## 
## Critical values for test statistics: 
##       1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34

# plotting the ACF and PACF's of the time series' first difference:
acf(na.omit(FirstDiff_Delay), lag.max = 3000, main = "ACF for the First Difference of Flight Delays", col = "darkblue")

pacf(na.omit(FirstDiff_Delay), lag.max = 3000, main = "PACF for the First Difference of Flight Delays", col = "darkred")
```

```
# very granular, convert TS to monthly to see the effects on the ACF and PACF
:
library(xts)

delay_xts <- xts(daily_mean$mean_delay, order.by = as.Date(daily_mean$FL_DATE
))
monthly_delay <- apply.monthly(delay_xts, mean)
diff_monthly_delay <- diff(monthly_delay)
acf(na.omit(diff_monthly_delay), lag.max = 15, main = "ACF for the First Diff
erence of Monthly Flight Delays", col = "darkblue")

pacf(na.omit(diff_monthly_delay), lag.max = 15, main = "PACF for the First Di
fference of Monthly Flight Delays", col = "darkred")

# AR(2) predictions, monthly:
ts.plot(monthly_delay, main = "Monthly Flight Delays, 5-Step-Ahead AR(2) Fore
cast")
AR_2 <- arima(monthly_delay, order = c(2,0,0))
predict_AR2 <- predict(AR_2)
AR2_forecast <- predict(AR_2, n.ahead = 5)$pred
AR2_forecast_se <- predict(AR_2, n.ahead = 5)$se
points(AR2_forecast, type = "l", col = 2)
points(AR2_forecast - 2*AR2_forecast_se, type = "l", col = 2, lty = 2)
points(AR2_forecast + 2*AR2_forecast_se, type = "l", col = 2, lty = 2)

# MA(2) predictions, monthly:
ts.plot(monthly_delay, main = "Monthly Flight Delays, 5-Step-Ahead MA(2) Fore
cast")
MA_2 <- arima(monthly_delay, order = c(0,0,2))
predict_MA2 <- predict(MA_2)
MA2_forecast <- predict(MA_2, n.ahead = 5)$pred
MA2_forecast_se <- predict(MA_2, n.ahead = 5)$se
points(MA2_forecast, type = "l", col = 2)
points(MA2_forecast - 2*MA2_forecast_se, type = "l", col = 2, lty = 2)
points(MA2_forecast + 2*MA2_forecast_se, type = "l", col = 2, lty = 2)

# ARMA(15,15) predictions, monthly:
ts.plot(monthly_delay, main = "Monthly Flight Delays, 5-Step-Ahead ARMA(2,2)
Forecast")
ARMA_22 <- arima(monthly_delay, order = c(2,0,2))
predict_ARMA22 <- predict(ARMA_22)
ARMA22_forecast <- predict(ARMA_22, n.ahead = 5)$pred
ARMA22_forecast_se <- predict(ARMA_22, n.ahead = 5)$se
points(ARMA22_forecast, type = "l", col = 2)
points(ARMA22_forecast - 2*ARMA22_forecast_se, type = "l", col = 2, lty = 2)
points(ARMA22_forecast + 2*ARMA22_forecast_se, type = "l", col = 2, lty = 2)

AIC(AR_2)

## [1] 521.1414

AIC(MA_2)
```

```
## [1] 520.7793
AIC(ARMA_22)
## [1] 517.5043
BIC(AR_2)
## [1] 531.7952
BIC(MA_2)
## [1] 531.433
BIC(ARMA_22)
## [1] 533.4849
```