Predicting traffic flow and volume in Northern Virginia and Washington D.C. capital region

Sarah Liang - sarahliang@ucsb.edu

Abstract

Using traffic flow and volume data, I fitted SARIMA and ARMAX model to the data, with the goal to forecast and predict traffic volume in the next 15 minutes in the locations at which my data was recorded.

Intro

I decided to use traffic flow data. My data set contains 1261 timesteps in the training split (and 840 timesteps in the testing split), involving traffic volume measurements along two major highways in Northern Virginia and Washington D.C. capital region, measured every 15 minutes at 36 sensor locations. The reason I chose traffic volume was that it was something that directly affected people. Prediction of traffic in certain areas can actually improve the livelihoods of people living in those areas. My analysis and prediction using this data can be used to analyze and improve the sustainability of cities.

Methodology

With a SARIMA model, it is easier to apply ARIMA model concepts to real life data that is not automatically stationary, or might have seasonal components that aren't considered in a basic ARIMA model. Often times, dependence on the past can induce seasonality in our data. As an overview, the ARIMA and SARIMA models rely heavily on the interactions of autoregression and moving average. Autoregression is when a time series depends on a linear combination of lagged instances of the times series itself and white noise residual. Moving average is behavior when a time series depends on instances of lagged white noise.

A normal ARIMA model has parameters p, d, and q, which represent autoregressive operator, difference, and moving average operator respectively.

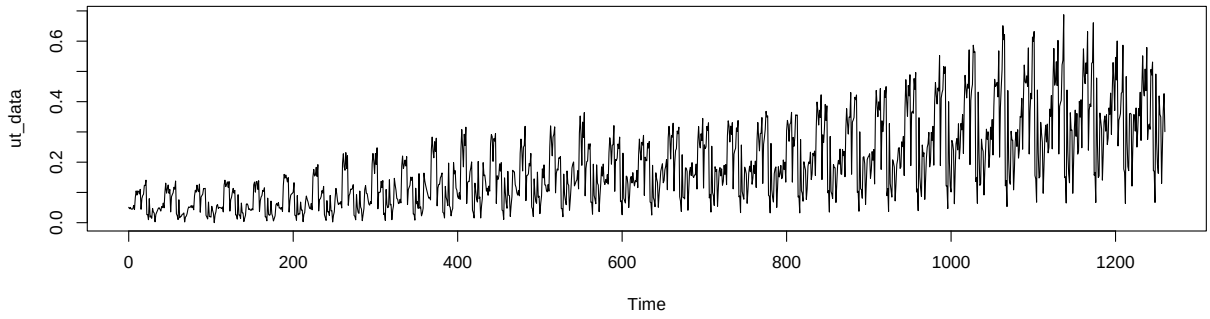
A SARIMA model is an normal ARIMA model with four more parameters that have to do with the seasonal part of the model: P, D, Q, and S which represent seasonal autoregressive operator, seasonal difference, seasonal moving average operator, and seasonal period respectively.

An ARMAX model is an ARMA model with the incorporation of exogeneous variables.

Since my data set also includes other variables, we can explore the dependencies between our traffic data and the exogenous variables using an ARMAX model.

Initial Look at Data

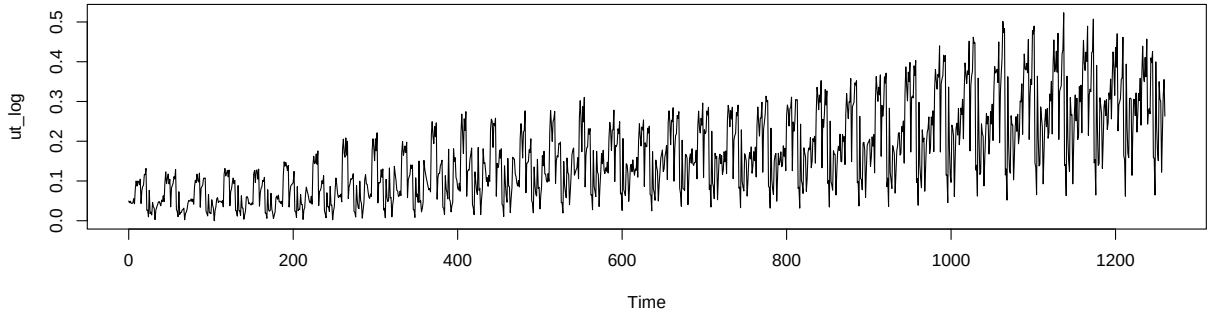
Let's take a look at our data, starting with the training split.

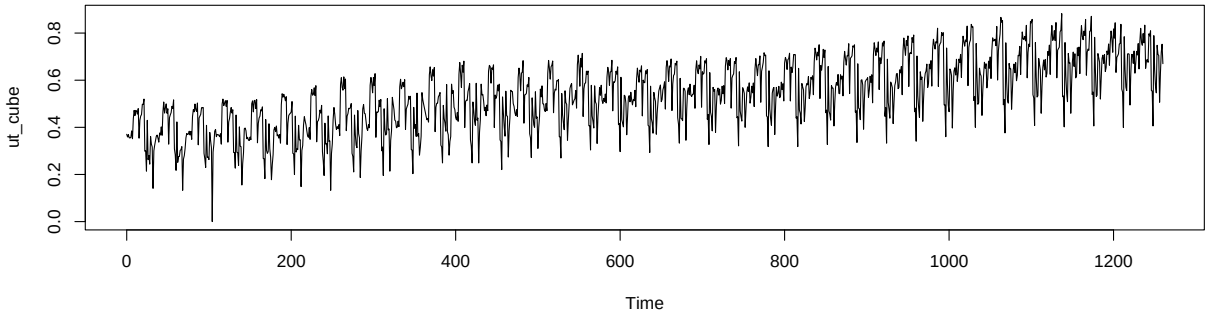


Immediately, the time series does not look stationary. There is a upwards trend overall, but also some very obvious patterns and peaks in the plot that may suggest signal or seasonality in the data. There is also inconsistent, increasing variance.

Transforming Data

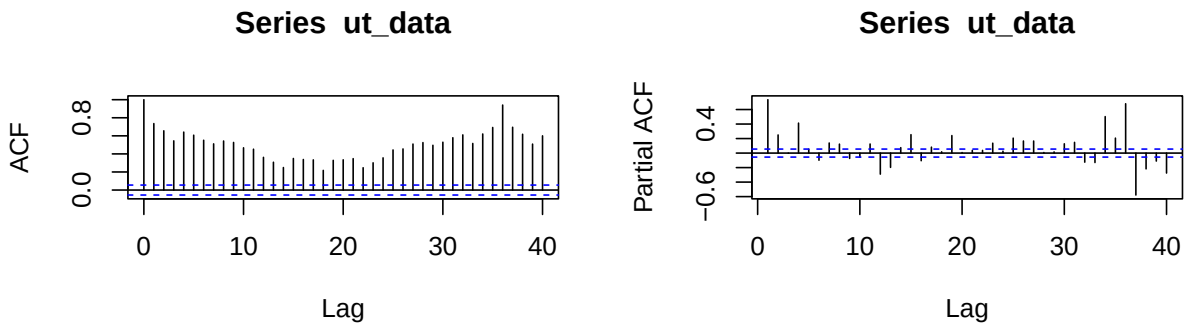
Even though the variance is increasing and inconsistent, I want to avoid log transforming immediately since my data values range from 0 to 1 inclusive. So, I will add the constant 1 to all the data points to avoid taking $log(0)$. I will also try a cube root transform.



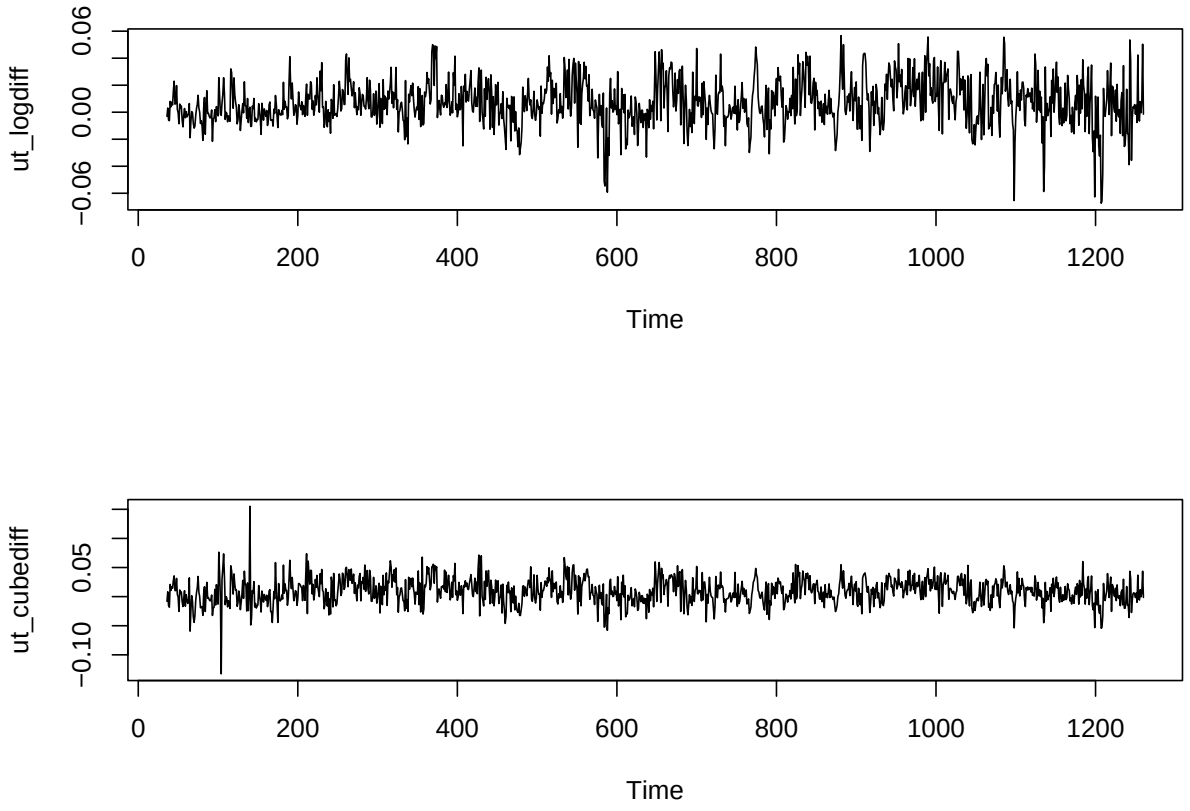

uf_cube
0.0
0.2
0.4
0.6
0.8
0
200
400
600
800
1000
1200
Time

ACF and PACF Interpretation

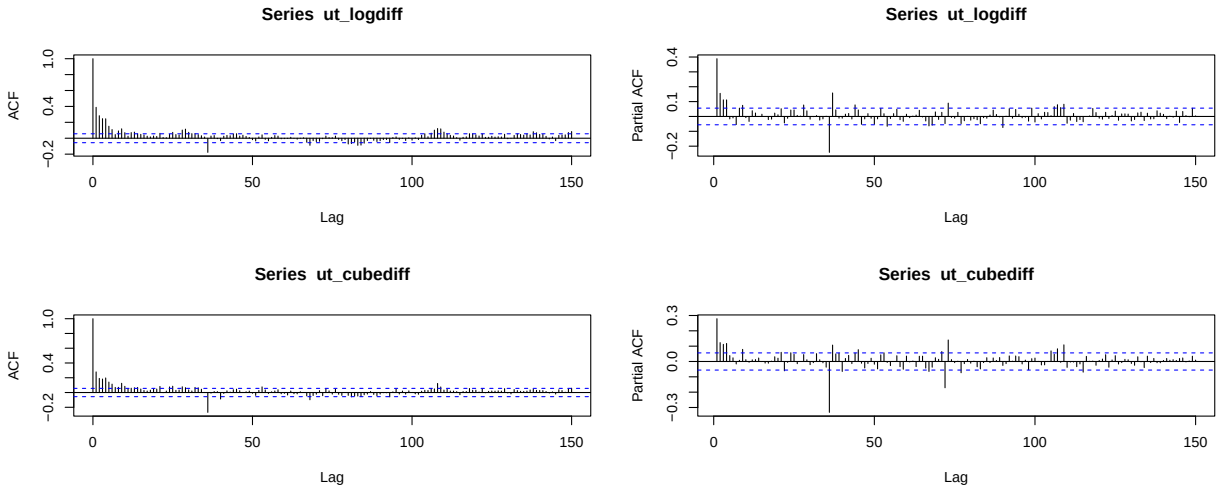
Now, let's look at the ACF and PACF of the untransformed data to see if the series resembles a specific model.



I will try to difference with `lag=36` because there are peaks at `lag=36`. This is due to the data being recorded at 36 different locations sequentially, so this will take care of seasonality in this sense.



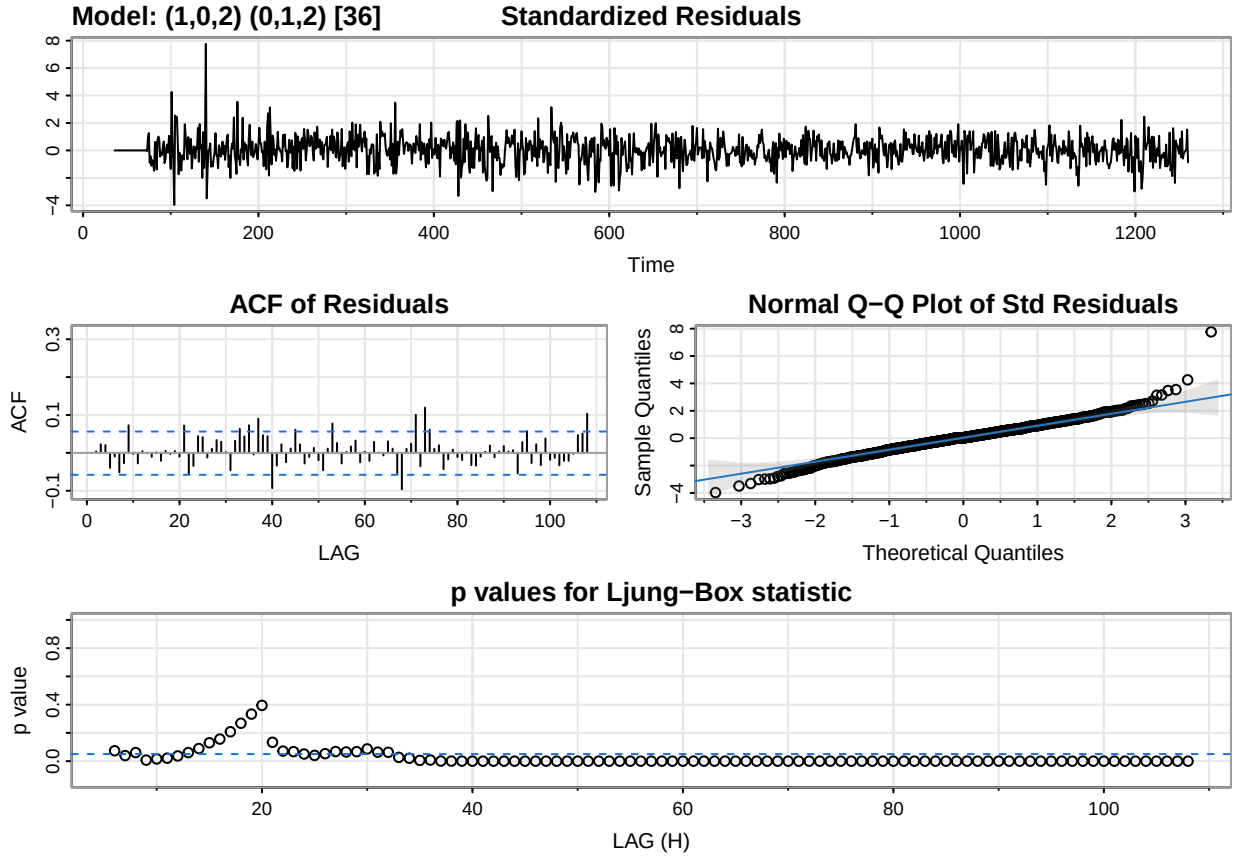
After differencing with `lag=36`, the series now appears with somewhat constant expectation and the variance doesn't vary with any obvious pattern.

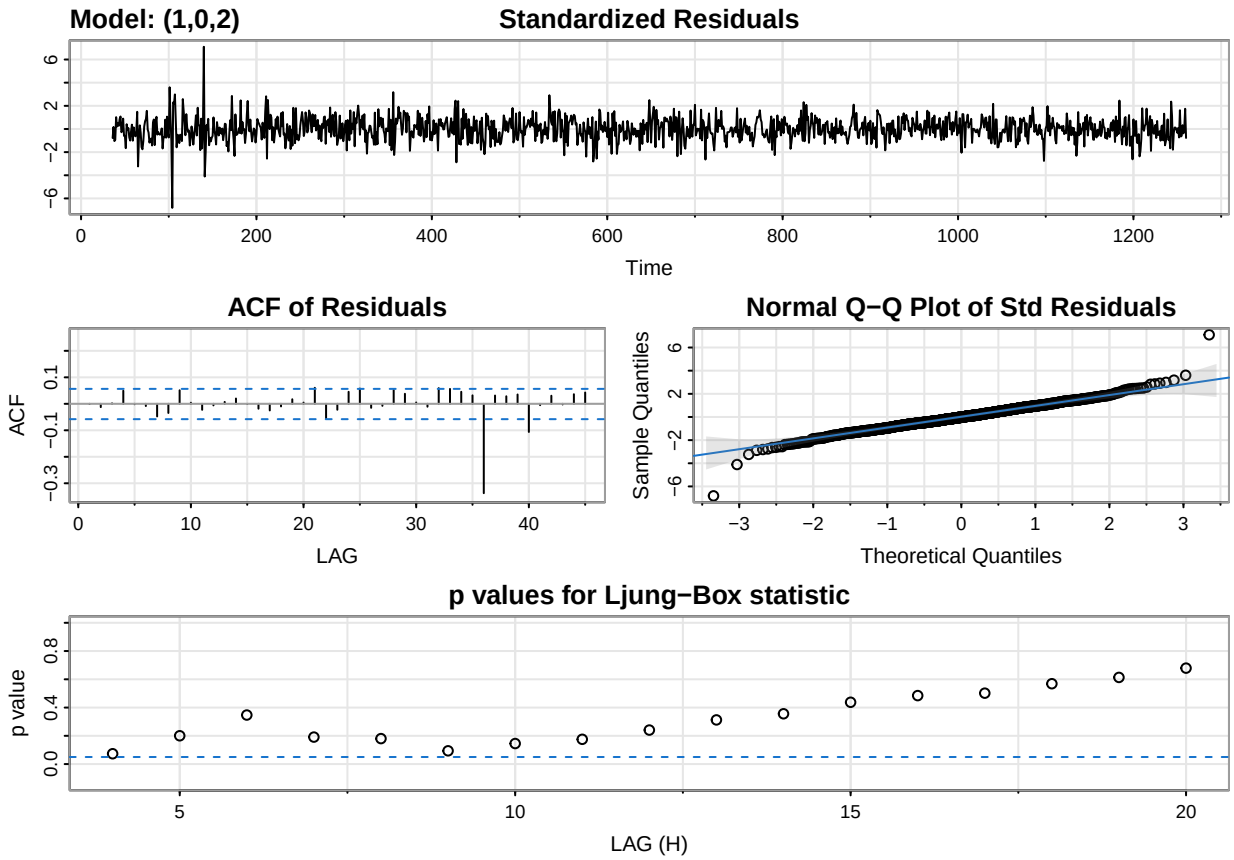


There is some tailing off in both ACF and PACF at lag 1 for both `ut_logdiff` and `ut_cubediff`. This tells me this series is at the very simplest a $SARIMA(1,0,1)(0,1,0)[36]$ model, with D=1 for seasonal difference. Upon closer look, the ACF is tailing off at each season and the PACF is cutting off one lag after each season (noticeable from the strong negative peak indicating the season, then a positive peak, then it cuts off). So, we may consider $SARIMA(1,0,1)(0,1,2)[36]$ model or a $SARIMA(3,0,0)(0,1,2)[36]$ model, since it seems that PACF is cutting off after lag 3 and ACF tails off, which suggest AR(3).

I will use the cube root transformed data from here on since the peaks were more clear in this one, compared to the log transformed.

SARIMA/ARIMA Modelling and Residual Diagnostics and Interpretation



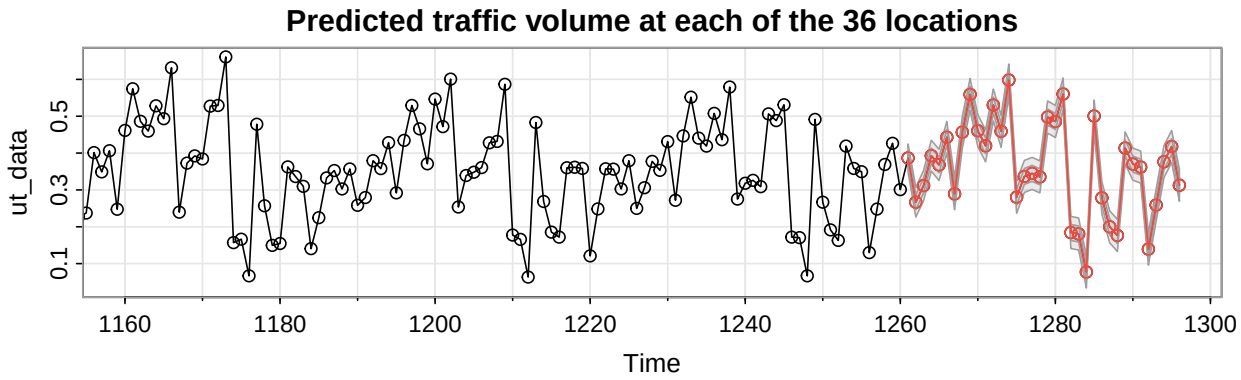


The residual diagnostics of the $SARIMA(1,0,2)(0,1,2)[36]$ model on the seasonal difference of the cube rooted data look overall adequate. The standardized residuals somewhat resemble random white noise in the first plot and the ACF of the residuals look ideal, with not much pattern. The Normal Q-Q Plot of the standardized residuals snakes away from the blue line only minimally at the ends. This suggests the residuals follow normality overall. Though, the p-values for the Ljung-Box statistic are significant as lags increas, which is not a good sign. However, with the AIC very low at -5796.12, I would still use this model even if the Ljung-Box p-values don't look too ideal.

On the other hand, the residual diagnostics of the $ARIMA(1,0,2)$ model on the seasonal difference of the cube rooted data look a bit better compared to the previous model. The standardized residuals in the first plot resemble white noise more consistently than in the previous model. The ACF of the residuals are more consistently within the blue dotted lines close to 0, however, the peak at 36 is still prevalent. This may insinuate that the seasonality is not entirely taken care of by the lagged difference alone. The Normal Q-Q Plot looks much better and the residuals follow the blue line more closely. The biggest difference is the Ljung-Box p-values. They are all not significant, which is desired. This model achieves an AIC of -5972.55, which is still quite low.

I will choose the $SARIMA(1,0,2)(0,1,2)[36]$ model to continue, as it considers the seasonality.

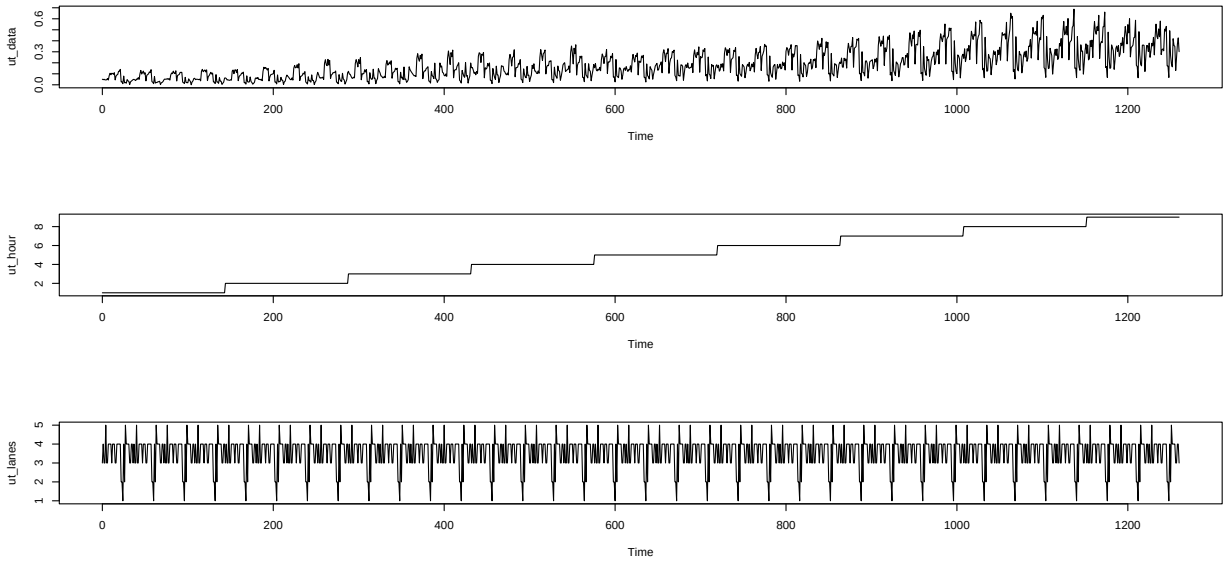
Forecasting



The forecasted data for the next 36 points, i.e. predicted traffic volume at each of the 36 locations 15 minutes later, looks to follow similar pattern to the previous seasons. I consider this $SARIMA$ model adequate for this data as it appears to make successful predictions.

ARMAX Modelling

Since my data set includes other variables, I would like to take a look at some of them. The column names are ambiguous and don't really match up with the description from the UCI archive, so I have a limited selection. The variables that interest me are the hour of day and number of lanes.



It seems the number of lanes has correlation with the traffic volume. The hour of the day seems to increase along with variance, also suggesting some correlation.

Now, let's look at what lag value I should use for the predictors, not including hour of the day because it doesn't match well with it being just a step function and all.

I will use `VARselect()` in the `vars` library to choose an ideal lag for the model.

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10     10     10     10
##
## $criteria
##                1            2            3            4            5
## AIC(n) -5.5195379 -5.573600642 -5.706964601 -5.926007956 -6.038087058
## HQ(n)  -5.5072016 -5.555096097 -5.682291873 -5.895167047 -6.001077967
## SC(n)  -5.4867213 -5.524375717 -5.641331366 -5.843966413 -5.939637206
## FPE(n)  0.0040077  0.003796785  0.003322744  0.002669118  0.002386122
##                6            7            8            9           10
## AIC(n) -6.067477789 -6.106214758 -6.141944034 -6.172987984 -6.320318537
## HQ(n)  -6.024300516 -6.056869304 -6.086430398 -6.111306167 -6.252468538
## SC(n)  -5.952619629 -5.974948289 -5.994269257 -6.008904898 -6.139827143
## FPE(n)  0.002317014  0.002228978  0.002150747  0.002085008  0.001799383
```

It seems the best lag is 10 for the model. Let's fit the $ARMAX$ model with p=10. Let's look at the coefficients using lag 10 in the model.

```
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.305824 -0.039859  0.003147  0.038362  0.256322
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## ut_data.l1    4.852e-01  3.424e-02  14.169  < 2e-16 ***
## ut_lanes.l1   5.463e-03  3.508e-03   1.557 0.119620
## ut_data.l2    2.656e-02  3.666e-02   0.724 0.469000
## ut_lanes.l2   2.355e-02  3.460e-03   6.806 1.56e-11 ***
## ut_data.l3   -2.989e-01  3.612e-02  -8.276 3.29e-16 ***
## ut_lanes.l3  -9.100e-03  3.470e-03  -2.623 0.008833 **
## ut_data.l4    3.044e-01  3.673e-02   8.289 2.96e-16 ***
## ut_lanes.l4   1.711e-02  3.464e-03   4.939 8.95e-07 ***
## ut_data.l5    2.101e-01  3.715e-02   5.656 1.93e-08 ***
## ut_lanes.l5  -7.237e-03  3.574e-03  -2.025 0.043092 *
## ut_data.l6   -2.620e-01  3.812e-02  -6.873 9.97e-12 ***
## ut_lanes.l6   6.566e-03  3.490e-03   1.881 0.060170 .
## ut_data.l7    1.401e-01  3.666e-02   3.821 0.000140 ***
## ut_lanes.l7   1.127e-02  3.351e-03   3.363 0.000795 ***
## ut_data.l8    5.490e-02  3.426e-02   1.602 0.109366
## ut_lanes.l8   6.968e-03  3.206e-03   2.174 0.029923 *
## ut_data.l9    6.674e-03  3.437e-02   0.194 0.846065
## ut_lanes.l9  -2.976e-03  3.216e-03  -0.925 0.354993
## ut_data.l10  -1.657e-01  3.197e-02  -5.182 2.56e-07 ***
## ut_lanes.l10  2.167e-02  3.192e-03   6.788 1.76e-11 ***
## const        -2.474e-01  4.695e-02  -5.269 1.62e-07 ***
## trend         1.314e-04  1.229e-05  10.691  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07355 on 1229 degrees of freedom
## Multiple R-squared:  0.7185, Adjusted R-squared:  0.7137
## F-statistic: 149.4 on 21 and 1229 DF,  p-value: < 2.2e-16
```

There is a significant relationship between traffic volume and the number of lanes at lags 2 through 7 and at lag 10.

I will use these lags in my ARMAX model.

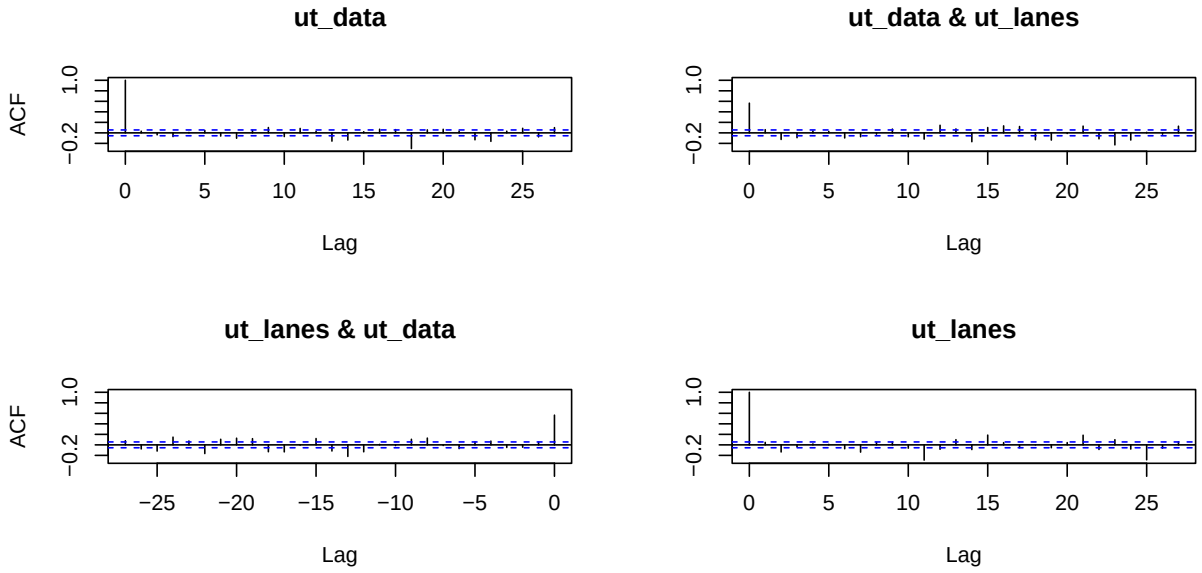
ARMAX Modelling Results

We have the following ARMAX model:

$$\hat{T}_t = -0.247 + 0.00013t + 0.0236L_{t-2} - 0.009L_{t-3} + 0.017L_{t-4} - 0.0072L_{t-5} + 0.00657L_{t-6} + 0.0113L_{t-7} + 0.02167L_{t-10}$$

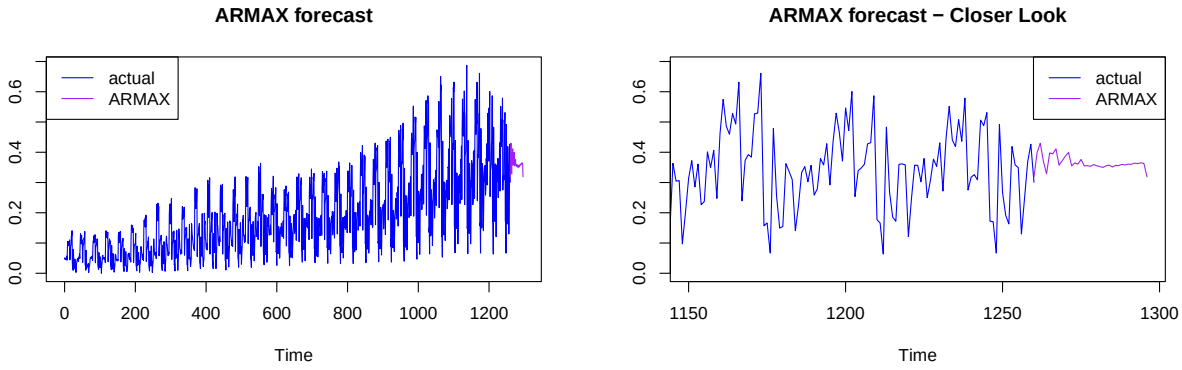
such that T_t and L_t denote traffic flow and number of lanes at time t respectively. We can see that the R^2 is about 0.71, meaning 71% of the variability in traffic volume can be explained by the number of lanes. There

is a very strong relationship between the two variables.




```
##
##  Portmanteau Test (adjusted)
##
## data:  Residuals of VAR object var1
## Chi-squared = 3988.1, df = 104, p-value < 2.2e-16
```

The ACF plots look to follow that of white noise, with the peak at lag 0 and nearly 0 otherwise. Thus, there looks to be no correlation in the residuals. The adjusted Postmanteau test has a p-value less than 0.05, so we can reject the null that the ACF of the residuals are not different from zero.



As you can tell, the ARMAX forecast is rather flat. The ARMAX forecast is nearly constant, settling into a flat line. It seems to capture the average of the points, which is not enough and ignores the complexity of seasonality and variance. It does not capture enough behavior from the original data like the SARIMA model does.

Conclusion

The SARIMA model is very accurate in capturing the seasonal components in the original traffic flow data. Forecasting using the SARIMA model is also impressive and accurate to the actual traffic behavior. On the other hand, the ARMAX model provides a vague, innaccurate forecast result. It is somewhat constant in mean, which does somewhat line up with the actual mean of the actual data. However, the essence of the actual data comes from the seasonality, which is something the ARMAX model fails to represent.

Regardless, fitting these models on actual data and seeing results is still one of my goals here. I achieved a fantastic result from the SARIMA model, so that is something I achieved well. In this project, we discovered how traffic flow varies in these 36 locations within the Northern Virginia and Washington D.C. areas (which results in the seasonality of the data). Though, we now know that the overall increasing variance in the original data is likely correlated with the hour of the day. As the hour of the day grows from midnight to midnight, the variance grows. The number of lanes can also affect traffic flow. Intuitively this makes sense, the more lanes the less congested it is to drive. Overall, the SARIMA model produced a great result, and I can grow further in this project to achieve better results, perhaps in considering other models. Or I can search for other traffic data for better and more variety in variables to use in the ARMAX model.

Resources

Traffic Flow Data Set on Kaggle

Data Set Description on UCI Machine Learning Archive

Time Series Analysis and Its Applications

Appendix

```
# READ IN DATA, CREATE TIME SERIES ORBJECT
set.seed(2536)
library(astsa)
library(dplyr)
ut_train <- read.csv("urbantraffic_train.csv")
ut_test <- read.csv("urbantraffic_test.csv")

ut <- ut_train %>%
  select(timestep, traffic)

ut_data <- ts(ut$traffic, start=0, end=1260)
plot.ts(ut_data)



# TRANSFORM DATA
# goes from about 0.1 to 0.5 now instead of 0.6 (variance is a little less volatile)
ut_log <- log(ut_data+1)
plot.ts(ut_log)

# variance looks much more stable here
ut_cube <- (ut_data)^(1/3)
plot.ts(ut_cube)



# ACF/PACF OF ORIGINAL DATA
par(mfrow=c(1,2))
acf(ut_data, lag.max=40)
pacf(ut_data, lag.max=40)



# SEASONAL DIFFERENCE OF TRANSFORMED DATA
ut_logdiff <- diff(ut_log, lag=36)
plot.ts(ut_logdiff)

ut_cubediff <- diff(ut_cube, lag=36)
plot.ts(ut_cubediff)
```

```r
# ACF/PACF OF DIFFERENCED TRANSFORMED DATA
par(mfrow=c(2,2))

# log transform + differenced
acf(ut_logdiff, lag.max=150)
pacf(ut_logdiff, lag.max=150)

# cube root transform + differenced
acf(ut_cubediff, lag.max=150)
pacf(ut_cubediff, lag.max=150)



# FITTING SARIMA MODELS
sarima(ut_cubediff, p=1,d=0,q=2, P=0,D=1,Q=2,S=36)

# the following ARIMA(1,0,2) model is from auto.arima
sarima(ut_cubediff, p=1,d=0,q=2)



# SARIMA FORECAST
sarima.for(ut_data, n.ahead=36, p=1,d=0,q=2, P=0,D=1,Q=2,S=36,
           main="Predicted traffic volume at each of the 36 locations")



# COMBINING HOUR OF THE DAY DUMMY VARIABLES
library(tidyr)
# combining all hour of the day dummy columns
ut_hourdata <- ut_train %>%
  mutate(hour_2 = replace(hour_2, hour_2==1, 2)) %>%
  mutate(hour_3 = replace(hour_3, hour_3==1, 3)) %>%
  mutate(hour_4 = replace(hour_4, hour_4==1, 4)) %>%
  mutate(hour_5 = replace(hour_5, hour_5==1, 5)) %>%
  mutate(hour_6 = replace(hour_6, hour_6==1, 6)) %>%
  mutate(hour_7 = replace(hour_7, hour_7==1, 7)) %>%
  mutate(hour_8 = replace(hour_8, hour_8==1, 8)) %>%
  mutate(hour_9 = replace(hour_9, hour_9==1, 9)) %>%
  mutate(hour_10 = replace(hour_10, hour_10==1, 10)) %>%
  mutate(hour_11 = replace(hour_11, hour_11==1, 11)) %>%
  mutate(hour_12 = replace(hour_12, hour_12==1, 12)) %>%
  mutate(hour_13 = replace(hour_13, hour_13==1, 13)) %>%
  mutate(hour_14 = replace(hour_14, hour_14==1, 14)) %>%
  mutate(hour_15 = replace(hour_15, hour_15==1, 15)) %>%
  mutate(hour_16 = replace(hour_16, hour_16==1, 16)) %>%
  mutate(hour_17 = replace(hour_17, hour_17==1, 17)) %>%
  mutate(hour_18 = replace(hour_18, hour_18==1, 18)) %>%
  mutate(hour_19 = replace(hour_19, hour_19==1, 19)) %>%
  mutate(hour_20 = replace(hour_20, hour_20==1, 20)) %>%
  mutate(hour_21 = replace(hour_21, hour_21==1, 21)) %>%
  mutate(hour_22 = replace(hour_22, hour_22==1, 22)) %>%
  mutate(hour_23 = replace(hour_23, hour_23==1, 23)) %>%
```

```
  mutate(hour_24 = replace(hour_24, hour_24==1, 24)) %>%
  mutate(across(hour_1:hour_24, na_if, 0))%>%
  unite(hour, hour_1:hour_24, sep="", na.rm=TRUE)%>%
  select(timestep, hour)


# number of lanes
ut_lanesdata <- ut_train %>%
  select(timestep, no_roads)




# CREATING TIME SERIES OBJECTS FOR POTENTIAL PREDICTORS
ut_hour <- ts(ut_hourdata$hour, start=0, end=1260)
ut_lanes <- ts(ut_lanesdata$no_roads, start=0, end=1260)
par(mfrow=c(3,1))
plot.ts(ut_data)
plot.ts(ut_hour)
plot.ts(ut_lanes)




# USING VARSELECT
library(vars)

x <- cbind(ut_data, ut_lanes)
VARselect(x, type="both")

var1 <- VAR(x, p=10, type="both")
summary(var1)$varresult$ut_data




# RESIDUAL DIAGNOSTICS OF ARMAX MODEL
acf(resid(var1))
serial.test(var1, lags.pt=36, type="PT.adjusted")



# ARMAX FORECAST
armax_pred <- predict(var1, n.ahead=36, ci=.95)

armax_forecast <- ts(armax_pred$fcst$ut_data[,1], start=1260, end=1296)

par(mfrow=c(1,2))
ts.plot(ut_data, armax_forecast, col=c("blue", "purple"))
title("ARMAX forecast")
legend("topleft",legend= c("actual", "ARMAX"), col=c("blue","purple"), lty=1)

ts.plot(ut_data, armax_forecast, col=c("blue", "purple"), xlim=c(1150, 1296))
```

```
title("ARMAX forecast - Closer Look")
legend("topleft",legend= c("actual", "ARMAX"), col=c("blue","purple"), lty=1)
```