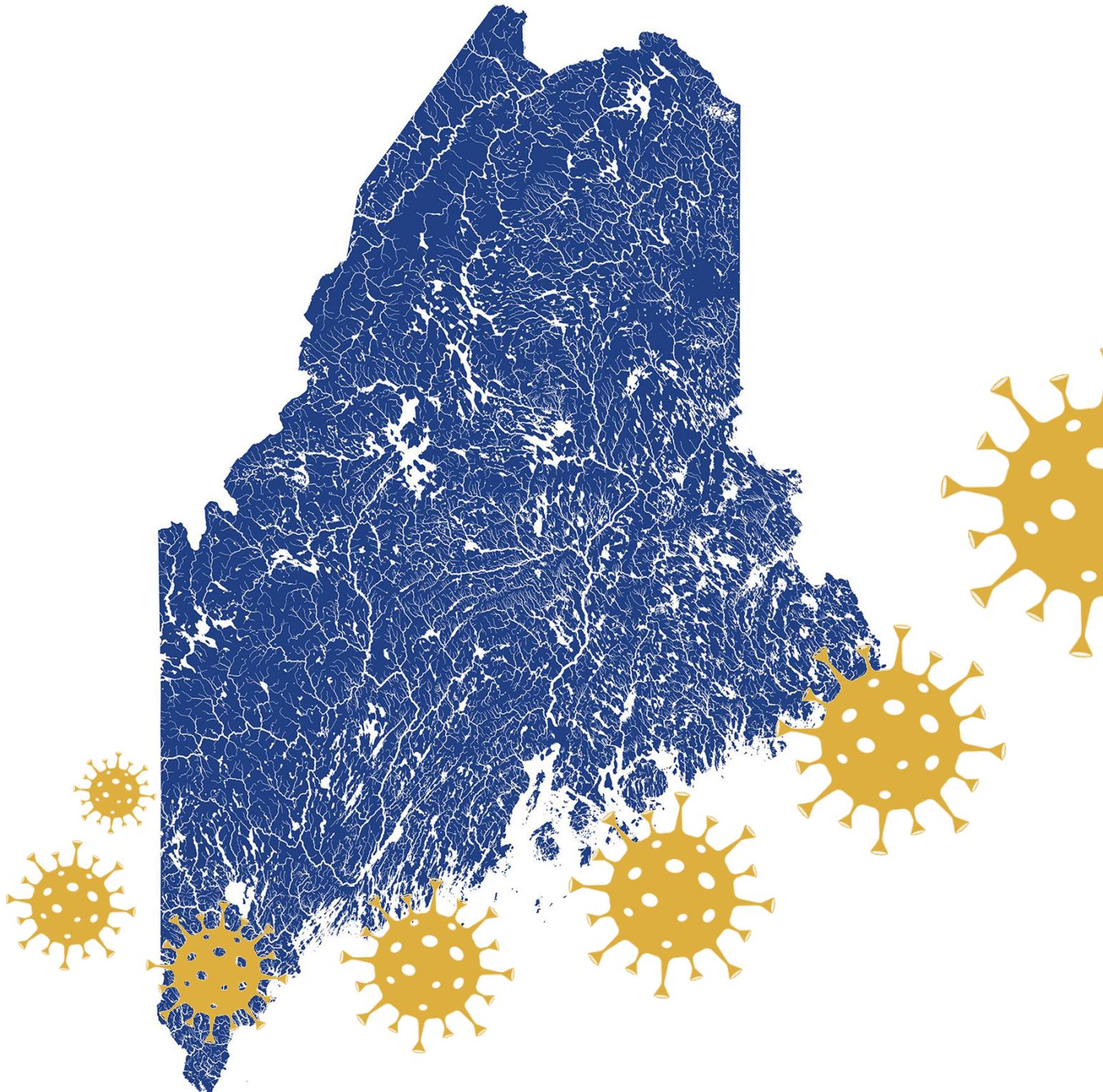


COVID-19 and Maine Gas Tax

Initial Economic Impact Analysis



Antonio Jurlina

Time Series Econometrics
Prof. Thomas Wiesen

Regional Economics: Modeling
Prof. Andrew Crawley

This report covers the attempt to estimate the effect of COVID-19 pandemic on gasoline tax revenue in Maine. Around the world, most economies have responded with lockdowns of varying degrees, in order to curtail the spread of the virus. Following this pattern, the state of Maine has imposed mandatory closures of non-essential businesses and reduced access to public areas. An inevitable outcome of this approach has been a significant reduction in traffic levels. Given that most activities require a certain amount of motor vehicle travel, these closures resulted in less distance being traveled and therefore less gasoline being purchased. For the past decade, the state has relied on a steady revenue stream from the tax on gasoline. This has accounted for nearly 70% percent of the Highway Fund and around 5% of the total tax revenue for a given fiscal* year.

In order to predict the expected losses of decreased traffic volume this reported is broken up into three steps. First, a seasonal ARIMA model is used in order to forecast the general revenue trend the state would have expected to see, had there been no major shock to the economy. Second, the relationship between traffic volume and gasoline tax revenue is estimated through a linear regression model and the expected revenue is predicted using known traffic data. Finally, the entire model is tested within the last fiscal year with available data (which is fiscal year 2019). This allows for comparison of values calculated within steps one and two with actual data.

Step 1: Forecasting

Looking at Figure 1, which shows monthly gasoline tax revenue between 2007 and 2018, several issues need to be addressed before any modelling is to occur. Within the red boxes are monthly values that drastically diverge from the expected trend and the observed mean. This series should be considered a proxy for the real variable of interest. The actual information we could say the state is interested in is the amount of revenue generated each month. However, the state is aware only of the taxes it actually collects, when it collects them. This means that collection processes and lags between the point in time when revenue is generated and when it is collected, need to be taken into account. With this in mind, these points are deemed outliers and reverted to the mean of the remainder of the series.

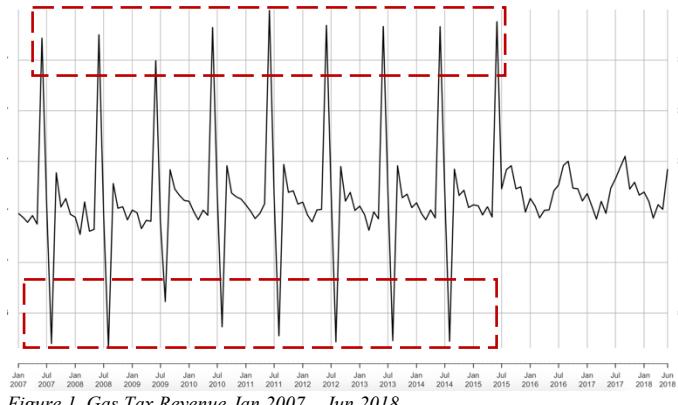


Figure 1. Gas Tax Revenue Jan 2007 – Jun 2018

Furthermore, there is clear seasonality in the data. Each fiscal year resembles the one before. Also, there is a slight upward trend in the data: although barely discernable, it is still worth noting. An augmented Dickey-Fuller (ADF) test, one that

accounts for seasonality with a trend, failed to reject the presence of unit roots with a statistic of -1.04. To detrend the data, remove the seasonal pattern and lose the unit root issue, the series was differenced monthly and seasonally. That is, each month's revenue was subtracted from the one a month away and a year away. Running an ADF test on this series, gives the statistic of -6.79 which strongly indicates no more unit root issues. The final series, which now appears to be stationary, can be seen in Figure 2.

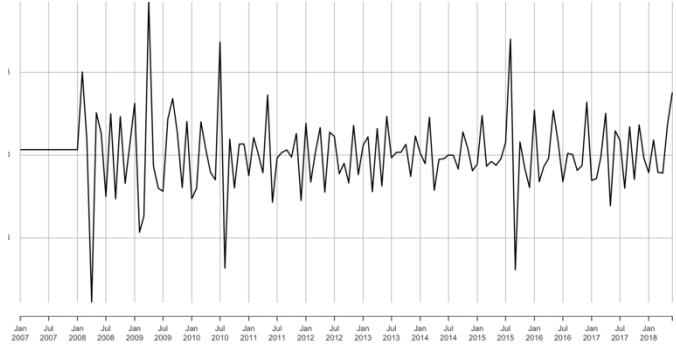


Figure 2. Cleaned-up Gas Tax Revenue Jan 2007 – Jun 2018

Figure 3 shows what the autocorrelations (both partial and complete) look like for the series. Lines colored red represent the seasonal lags (at 12, 24, etc. to account for the monthly nature of the series). Looking at the complete autocorrelation plot (ACF), there is no significant tailing off within a season. However, two red lines above the line of significance suggest that it would be wise to include a seasonal autoregressive term. Looking at the partial autocorrelation plot (PACF), there is some tailing off within a season and no seasonal pattern of significance, suggesting that a moving average term should be added.

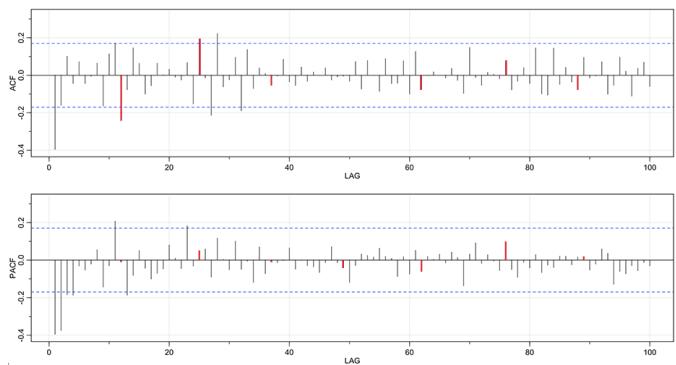


Figure 3. ACF (top) and PACF (bottom), seasonal lags in red

With this in mind (and a few minor alterations around each term), the model with the lowest reported AIC and BIC is SARIMA (0, 1, 1)(1,1,0)₁₂ or

$$(1 - \phi_1 B^{12})(1 - B)(1 - B^{12})Y_t = (1 - \theta_1 B)\epsilon_t$$

where the moving average is 1, seasonal autoregressive lag is 1 and the data has been differenced once monthly and once seasonally. As can be noted by Figure 4, residuals seem to be distributed randomly and independently. Table 1 reports the estimates of the model. Finally, Figure 5 shows what the expected forecast 12 months ahead is. This forecast covers fiscal year 2019, for which data is available.

*in Maine, fiscal year spans July of last year to June of current year. That is, fiscal year 2018 starts in July 2017 and ends in June 2018.

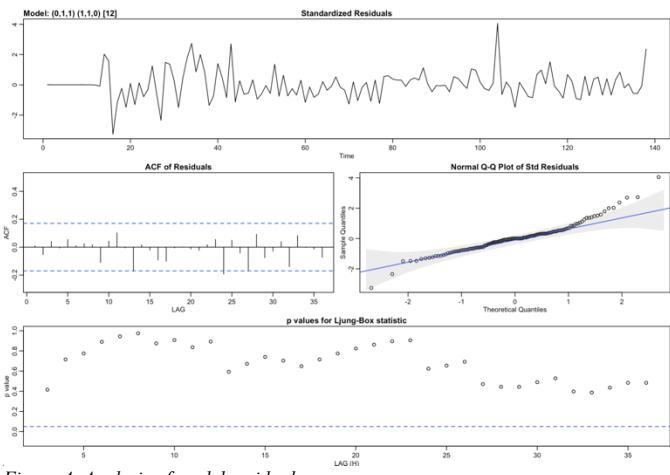


Figure 4. Analysis of model residuals.

	Estimate	SE	t.value	p.value
ma1	-0.8469	0.0557	-15.218	0.0000
sar1	-0.3528	0.0903	-3.9064	0.0002

Table 1. SARIMA estimation results

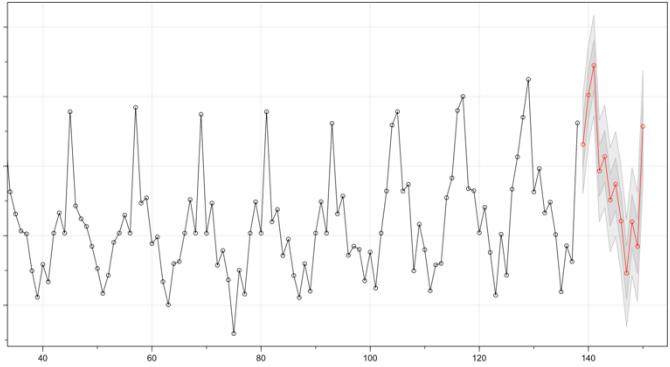


Figure 5. Forecast for 2019 fiscal year

Step 2: Revenue vs. Traffic

Plotting traffic and revenue over time, on the same plot (as can be seen in Figure 7), reveals a few interesting things. They both follow the same seasonal pattern over time (with a month lag between the two) and both have quarterly trends that can be individually considered. First of all, the points are separated into quarterly sets and then shifted over by a month so that the highest and lowest points of each series sync up. This lag is assumed to exist because it must take the state some time (presumably a month) to collect the tax after the traffic levels have been observed.

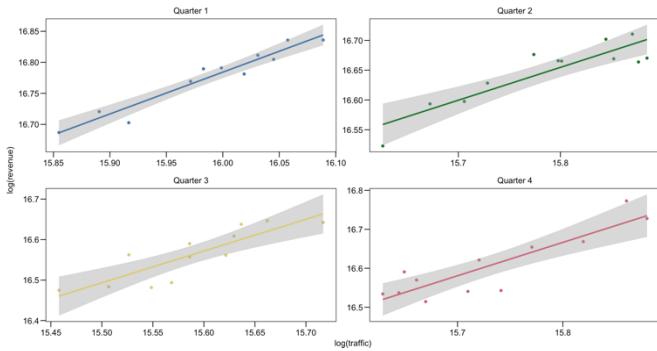


Figure 6. OLS estimates

	Quarter 1	Quarter 2	Quarter 3	Quarter 4
intercept	8.168*** (0.667)	10.616*** (2.63)	2.559* (2.93)	7.744* (3.689)
log(traffic)	0.539*** (0.042)	0.383** (0.167)	0.899*** (0.188)	0.561** (0.234)

Table 2. OLS estimates

Now, as they are synced up, quarterly slope signs of each series are identical and therefore a straightforward linear regression should be all that is necessary. However, to further capture the possible quarterly idiosyncrasy of the data and further reduce variation, regressions are performed within each quarter. Additionally, future data for state traffic levels and tax revenue used for economic impact analysis of COVID-19 closures will concern the last quarter of fiscal year 2020, starting with April. Therefore, it seemed prudent to isolate quarterly effects in order to obtain the most accurate estimates.

Figure 6 and Table 2 show the results of the linear regression. As can be noted, the relationships are fairly straightforward and statistically significant. For each 1% increase in traffic volume, there is 54, 38, 90, and 56 percent increase in revenue for quarters 1 through 4, respectively. This data can now be used to estimate revenue from traffic. In this case, revenue is simply

$$\text{revenue} = \text{intercept} + \text{coefficient} * \text{traffic}$$

and traffic volume is obtained from monthly turnpike data across Maine.

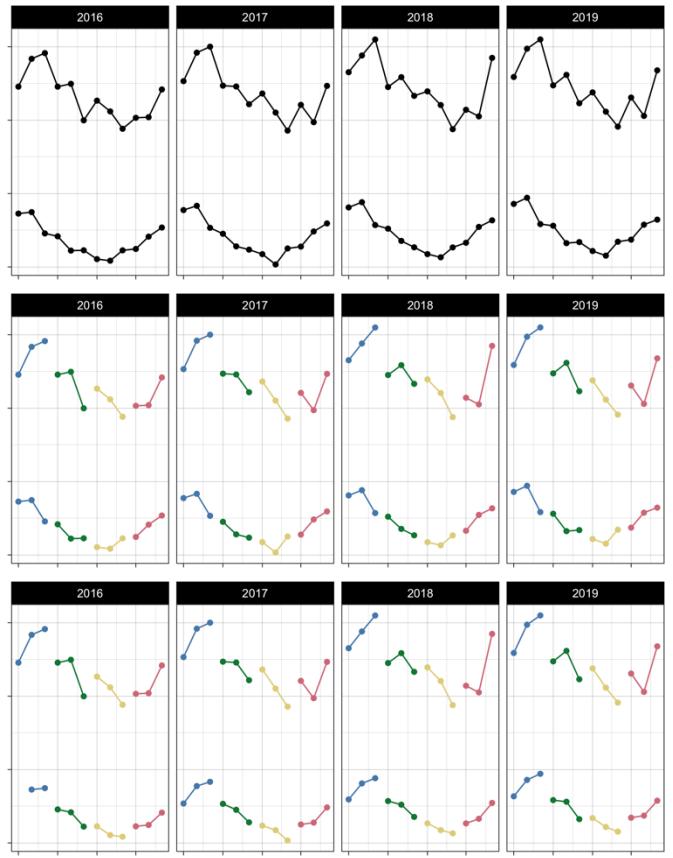


Figure 7. Organizing traffic (bottom) and revenue (top) data.

Step 3: Putting it all together

This one is straightforward, as Figures 8 and 9 illustrate. Fiscal year 2019 contains three lines: actual revenue (yellow), OLS predicted revenue (green) and time series forecast of expected revenue (red). Furthermore, Table 3 shows the difference between expected and predicted revenue. For fiscal year 2019, this doesn't provide new information, because there

were no unexpected shocks, and all the models arrive at the same conclusion (mostly). However, the purpose of running the model across 2019 was only to determine its validity. Matching results were expected. Fiscal year 2020 is where it is actually going to be used to make predictions

	Actual Data			OLS			SARIMA		
	Min	Average	Max	Min	Average	Max	Min	Average	Max
Quarter 1	58,311,032	56,724,305	59,275,087	61,825,870	55,836,036	58,001,488	60,166,940		
Quarter 2	51,606,718	44,014,749	52,863,879	61,713,010	49,853,379	52,084,273	54,315,167		
Quarter 3	47,028,837	40,009,428	49,259,003	58,508,579	45,395,921	47,652,095	49,908,270		
Quarter 4	50,225,926	36,792,179	48,887,326	60,982,473	47,774,515	50,034,520	52,294,526		

Table 3. Tax revenue comparison for fiscal year 2019, in US\$

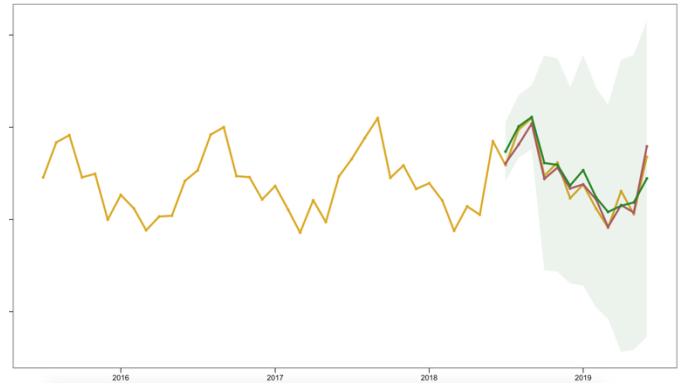


Figure 8. Tax revenue data (yellow), OLS prediction (green), SARIMA forecast (red); shaded region represents 99% CI

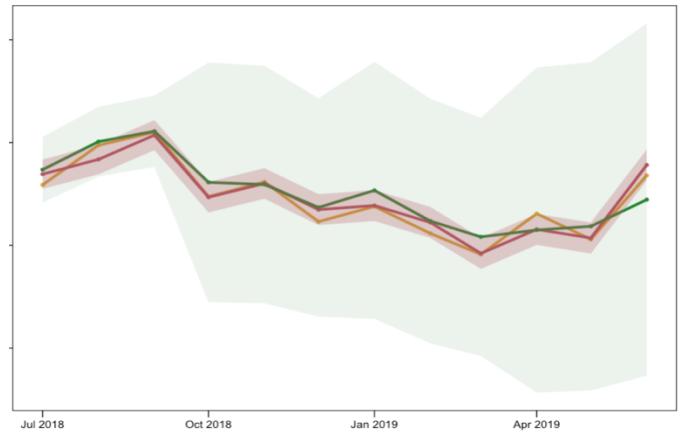


Figure 9. Tax revenue data (yellow), OLS prediction (green), SARIMA forecast (red); shaded region represents 99% CI, (close up of 2019 fiscal year)

Appendix: R Code

```
#----- packages -----
library(tidyverse)
library(ggplot2)
library(lubridate)
library(directlabels)
library(ggrepel)
library(ggthemes)
library(scales)
library(xts)
library(astsa)
library(forecast)
library(tseries)
library(broom)

#----- data and directory -----
directory <- paste0(here::here(), "/data")
setwd(directory)

load("traffic_breakdown.RData")
load("traffic_summary.RData")
load("class.RData")
load("comparison1.RData")
load("comparison2.RData")
load("interchange.RData")
load("tax_revenues.RData")
load("unemployment.RData")

directory <- paste0(here::here(), "/output")
setwd(directory)

options(scipen = 100)

#----- functions -----
linear_reg <- function(data, formula){
  lm(formula, data)
}

#----- analysis -----
gas <- tax_revenues %>%
  filter(revenue != 0, !is.na(revenue)) %>%
  filter(code == "0321") %>%
  filter(date < "2020-01-01", date >= "2007-01-01") %>%
  filter(fiscal_year < 2019) %>%
  group_by(date) %>%
  summarise(revenue = sum(revenue, na.rm = TRUE)) %>%
  ungroup()

gas <- gas %>%
  mutate(revenue = ifelse(revenue > 22000000, NA, revenue),
         revenue = ifelse(revenue < 13000000, NA, revenue),
         revenue = ifelse(is.na(revenue), mean(revenue, na.rm = T), revenue))

gas <- xts(gas$revenue, order.by = gas$date)
lgas = log(gas)
dgas <- na.locf(diff(gas), fromLast = T)
ddgas <- na.locf(diff(dgas, lag = 12), fromLast = T)

data <- ddgas
length(data)

adf.test(data)
plot(data)
acf2(data, max.lag = 100, main = "")

sarima(gas, 0, 1, 1, 1, 1, 0, 12)
sarima.for(gas, n.ahead = 12, 0, 1, 1, 1, 1, 0, 12)

auto.arima(gas, trace = TRUE, ic = "bic", seasonal = TRUE,
           seasonal.test = "ocsb")

ols <- tax_revenues %>%
  filter(code == "0321", fiscal_year %in% c(2016:2019)) %>%
  filter(revenue != 0, !is.na(revenue)) %>%
  filter(date < "2020-04-01") %>%
  left_join(comparison_2 %>%
    group_by(date) %>%
    summarise(traffic = sum(transactions, na.rm = TRUE)) %>%
    mutate(month = month(date) + 1,
          year = year(date),
          year = if_else(month > 12, year + 1, year),
          month = if_else(month > 12, month - 12, month),
          date_shifted = ymd(paste0(year, "-", month, "-01"))),
    by = "date"))

plot <- ols %>%
  ggplot(aes(x = date, y = revenue, color = quarter)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y = lag(traffic, 1))) +
  geom_point(aes(y = lag(traffic, 1))) +
  facet_wrap(~ fiscal_year, scales = "free_x", nrow = 1) +
  scale_color_ptol() +
  theme_linedraw() +
  scale_y_continuous(label = comma) +
  theme(axis.text.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.y = element_blank(),
        axis.title.y = element_blank(),
        legend.position = "none")

ggsave("plot.png", plot, height = 3, width = 7)

ols %>%
  ggplot(aes(x = log(lag(traffic, 1)), y = log(revenue), colour = quarter)) +
  geom_point() +
  stat_smooth(method = "lm") +
  facet_wrap(~ quarter, scales = "free") +
  scale_color_ptol() +
  theme_par() +
  xlab("log(traffic)") +
  ylab("log(revenue)") +
  theme(legend.position = "none")

ols %>%
  ggplot(aes(x = lag(traffic, 1), y = revenue, colour = quarter)) +
  geom_point() +
  stat_smooth(method = "lm") +
  scale_y_continuous(label = comma) +
  scale_x_continuous(label = comma) +
```

```

facet_wrap(~ quarter, scales = "free") +
  scale_color_ptol() +
  theme_par() +
  xlab("traffic") +
  ylab("revenue") +
  theme(legend.position = "none")

ols_output <- map(c("Quarter 1", "Quarter 2", "Quarter 3",
"Quarter 4"), function(x) {
  bind_rows(
    ols %>%
      filter(quarter == x) %>%
      linear_reg(log(revenue) ~ log(lag(traffic, 1))) %>%
      tidy() %>%
      mutate(type = "log",
            quarter = x),
    ols %>%
      filter(quarter == x) %>%
      linear_reg(revenue ~ lag(traffic, 1)) %>%
      tidy() %>%
      mutate(type = "normal",
            quarter = x)
  )
}) %>% data.table::rbindlist()

ols_output_log <- ols_output %>%
  filter(type == "log")
ols_output <- ols_output %>%
  filter(type == "normal")

ols_output <- left_join(
  ols_output %>%
    filter(term == "lag(traffic, 1)") %>%
    rename(slope = estimate,
           slope_se = std.error) %>%
    select(slope, slope_se, quarter),
  ols_output %>%
    filter(term != "lag(traffic, 1)") %>%
    rename(intercept = estimate,
           intercept_se = std.error) %>%
    select(intercept, intercept_se, quarter),
  by = "quarter")

expectation <- sarima.for(gas, n.ahead = 12, 6, 0, 1, 1, 0, 0, 12)
expectation <- tibble(expectation$pred,
  expectation$se,
  tax_revenues %>%
    filter(fiscal_year == 2019) %>%
    select(date) %>% unique())
colnames(expectation) <- c("expectation", "error", "date")

prediction <- tax_revenues %>%
  filter(code == "0321") %>%
  filter(revenue != 0, !is.na(revenue)) %>%
  filter(date < "2020-04-01") %>%
  left_join(comparison_2 %>%
    group_by(date) %>%
    summarise(traffic = sum(transactions, na.rm =
TRUE)) %>%
    mutate(month = month(date) + 1,
          year = year(date),
          year = if_else(month > 12, year + 1, year),
          month = if_else(month > 12, month - 12, month),
          date_shifted = ymd(paste0(year, "-", month, "-01"))),
      by = "date") %>%
  left_join(ols_output,
            by = "quarter") %>%
  left_join(expectation,
            by = "date") %>%
  mutate(traffic = lag(traffic, 1),
        prediction = ifelse(fiscal_year == 2019, intercept +
slope*traffic, NA),
        prediction_upper = ifelse(fiscal_year == 2019, (intercept +
intercept_se) + (slope + slope_se) * traffic, NA),
        prediction_lower = ifelse(fiscal_year == 2019, (intercept -
intercept_se) + (slope - slope_se) * traffic, NA),
        prediction_error = revenue - prediction,
        expectation_upper = expectation + error,
        expectation_lower = expectation - error)

prediction %>%
  filter(fiscal_year == 2019) %>%
  ggplot(aes(x = date)) +
  geom_point(aes(y = revenue), colour = "#DDAA33") +
  geom_line(aes(y = revenue), colour = "#DDAA33", size = 1.5) +
  geom_point(aes(y = expectation), colour = "#BB5566") +
  geom_line(aes(y = expectation), colour = "#BB5566", size = 1.5) +
  geom_point(aes(y = prediction), colour = "#228833") +
  geom_line(aes(y = prediction), colour = "#228833", size = 1.5) +
  geom_ribbon(aes(ymin = prediction_lower, ymax = prediction_upper),
              fill = "#228833", colour = "transparent", alpha = 0.1) +
  geom_ribbon(aes(ymin = expectation_lower, ymax = expectation_upper),
              fill = "#BB5566", colour = "transparent", alpha = 0.3) +
  scale_y_continuous(label = comma) +
  theme_par() +
  theme(axis.title.x = element_blank())

prediction %>%
  filter(fiscal_year == 2019) %>%
  group_by(quarter) %>%
  summarize(revenue = sum(revenue, na.rm = T),
            prediction = sum(prediction, na.rm = T),
            prediction_upper = sum(prediction_upper, na.rm = T),
            prediction_lower = sum(prediction_lower, na.rm = T),
            expectation = sum(expectation, na.rm = T),
            expectation_upper = sum(expectation_upper, na.rm = T),
            expectation_lower = sum(expectation_lower, na.rm = T)) %>%
  View()

```