A Time Series Analysis: Predicting Monthly Business Applications in the U.S. Agriculture Industry

Jessie Zhou

2023-06-12

Contents

Abstract

In this project, I perform a time series analysis using the Box-Jenkins methodology to analyze the total number of monthly business applications in the agriculture industry across the U.S. The main questions I will address are what the underlying trends are in the data, as well as what the monthly number of business applications would be in the years 2020-2021 absent of the Covid-19 pandemic. Forecasts for before the pandemic are accurate with original data inside the prediction interval; however, predictions are quite inaccurate during the years of the pandemic.

Introduction

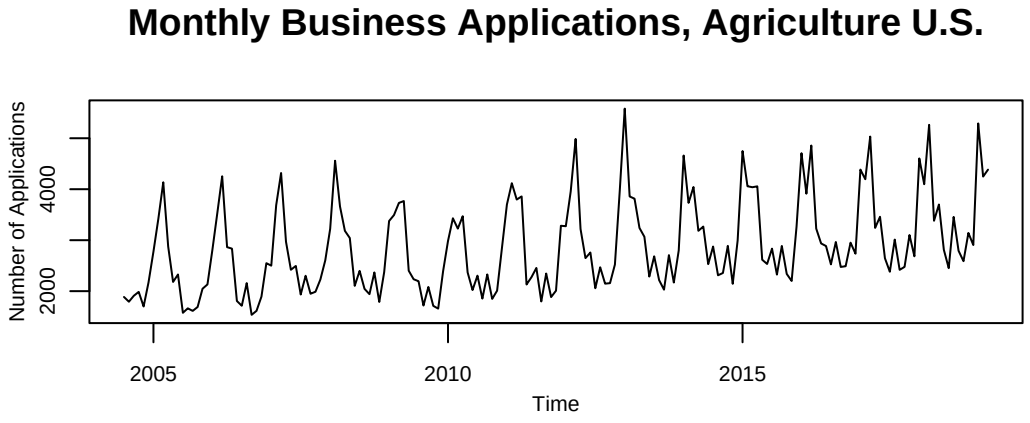
The number of business applications in the agriculture industry can serve as an economic indicator, providing insights into the level of entrepreneurial activity and investment in the sector. This can be useful for policymakers, researchers, and economists studying the industry's performance. I decided to subset the data from 2004 ending at March 2020, which allows the model to be fit on data undisturbed by the pandemic; I would like to predict what would have happened in the year 2021 had Covid-19 not emerged.

In my analysis, I performed differencing and transformation to create a stationary time series, and assessed autocorrelation/partial autocorrelation plots to determine the best model parameters. I then used the `arima()` function to test different models, and performed diagnostic checking on each model's residuals. Despite my best model not passing every test, the test set of my original data fits within the prediction intervals.

EDA

The first step I took before assessing the data was to split it into a testing and training set in order to compare the forecasts with original data. I decided to take out the last 12 observations for testing as I am only predicting a year in advance[1].

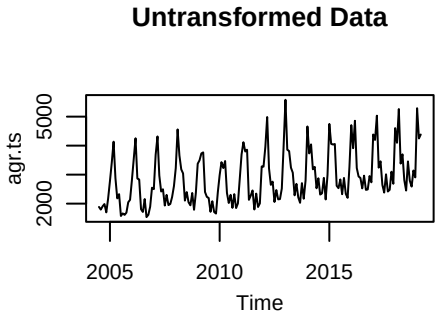
After the split, I plotted the training data and noticed both trend and seasonality. There is a steady increase of applications over time, with seasonal cycles of about a year: Business applications seem to spike at the beginning of the years, peaking after the first few months and decreasing throughout the rest. A decomposition shown later on after transformation reveals these distinct patterns. There seems to be a decrease of applications around 2010 and an increase in 2014.

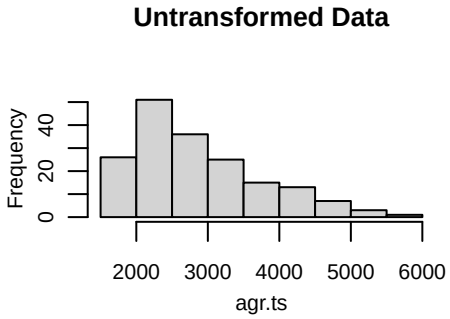


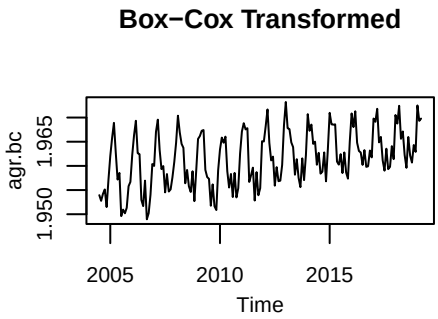
[1]While I predict 24 months ahead for the pandemic years, I gathered an additional 12 months later on for a new testing set. Therefore, I only remove 12 months for the first prediction.

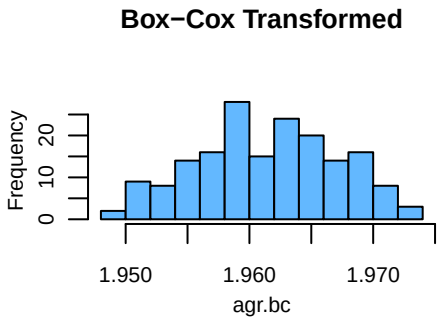
Transformation

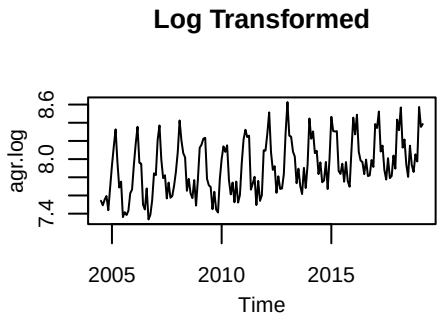
Since the seasonal cycles vary slightly, I decided to test three different methods of transformation: Box-Cox, natural log, and square root. To decide on the best transformation method, I assessed the plots and histograms of transformed data to check for a more stable variance and normality. From these, the Box-Cox transformed data seems the most normally distributed. We will proceed with this transformation.

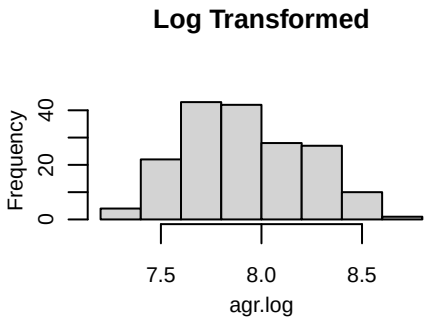


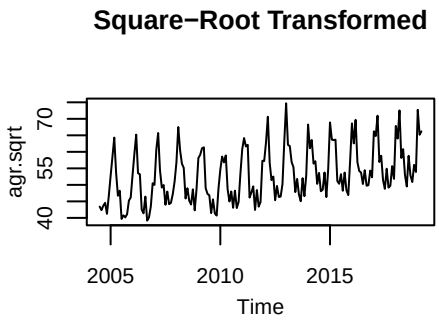


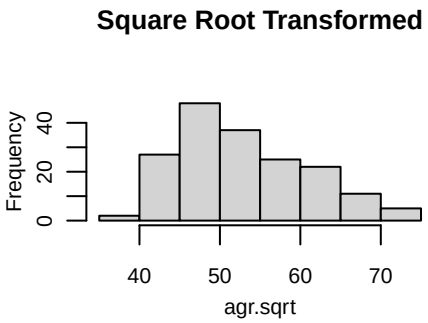




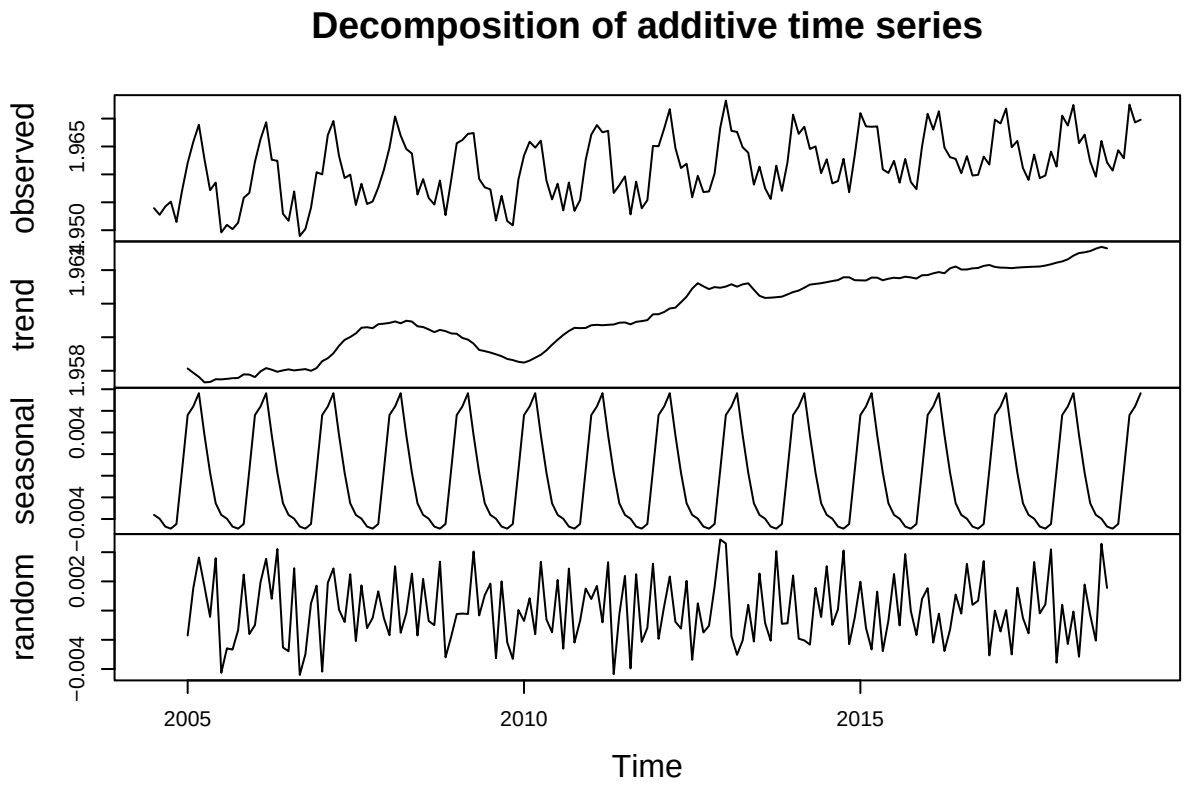






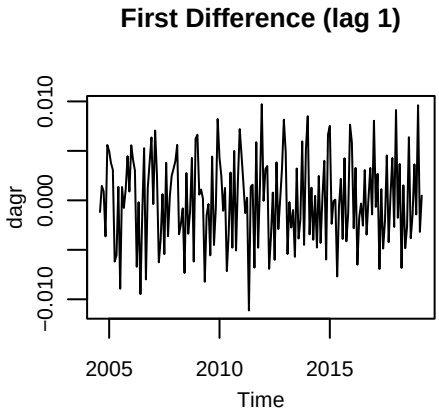


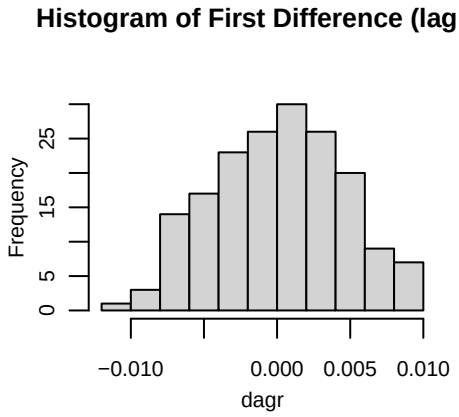
Now, we can plot the decomposition of our final transformed data to get a better look at the trend and seasons.

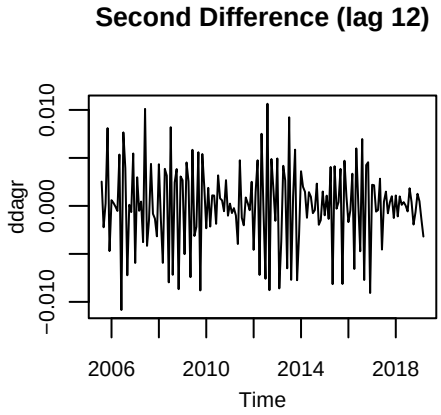


Differencing

Since the data has clear trend and seasonality, we can perform differencing to remove each of these components. I first difference at lag 1 to remove trend, and at lag 12 (as it is monthly data) to remove seasonality. From the histograms, we can see that both differences resulted in more normally distributed data. In addition, we note that the variances decrease (Table 1) with each difference so we proceed with these operations to the data.







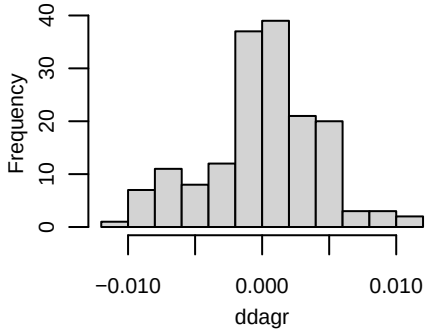
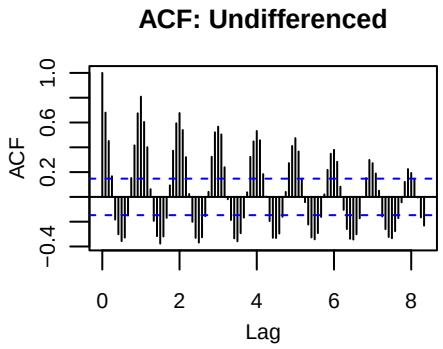


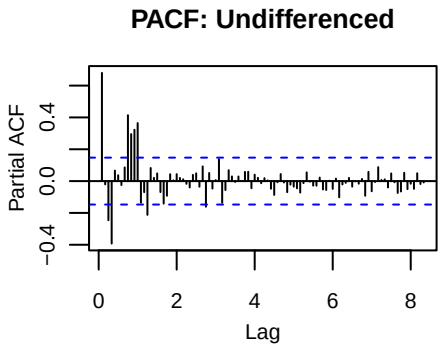
Table 1: Variances after Differencing

	Undifferenced Data	First Difference (lag 1)	Second Difference (lag 12)
Variance	3.2e-05	1.99e-05	1.78e-05

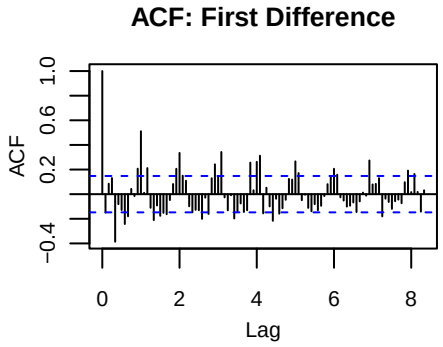
P/ACF analysis

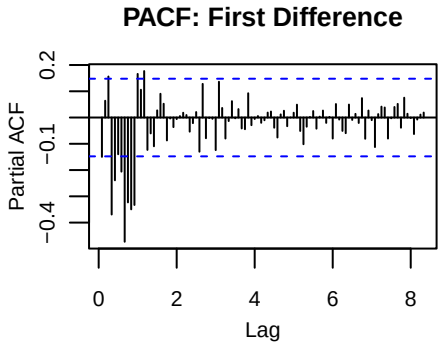
After differencing the data, I plotted the ACF and PACF plots to help me identify the model. Below are the ACF and PACF plots of the original undifferenced time series, and we can observe strong seasonality in the ACF plot.



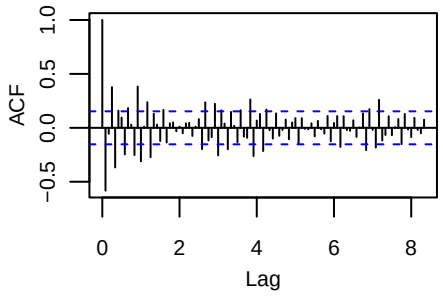


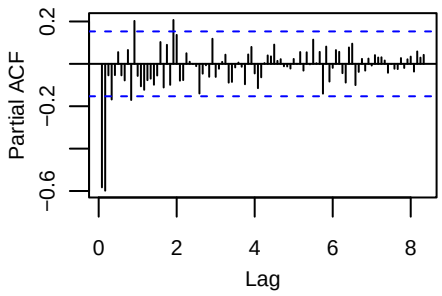
Next, I checked the plots after differencing at lag 1. We can note a significantly shrunken ACF plot; however, it still looks to be seasonal. The PACF plot shows more negative values than the orginal.





Finally, I checked the plots for the final model differenced at both lag 1 and lag 12. The ACF is reduced even more than the last, and there is no apparent seasonality. More PACF values have become insignificant, and we can begin to suggest model parameters. Based on the PACF plot, the data seem to fit a model with autoregressive and seasonal autoregressive components; I will begin with $P = 2$ and $p = 2, 4$.





Model Specification

Round 1: AR components only

To begin model specification, I started with only autoregressive components. From the PACF, the autogressive component looks to be either order 2 or 4, and the seasonal autoregressive component looks to be order 2. I checked SARIMA$(2,1,0)(2,1,0)_{12}$ and SARIMA$(4,1,0)(2,1,0)_{12}$, and both seem to work with none of the parameters having 0 in the confidence interval. Since the AICC values for both are very similar, I decided to move on with p = 2 to reduce the number of parameters.

Table 2: AICC Values of SARIMA(p,1,0)(2,1,0), s=12

	p = 2	p = 4
AICC value	-1484.038	-1484.595

Table 3: Coefficient Estimates for SARIMA(2,1,0)(2,1,0), s = 12

	Coefficient Estimate	Standard Error
ar1	-0.9632616	0.0624664
ar2	-0.6172798	0.0613196
sar1	-0.3699024	0.0788172
sar2	-0.2440051	0.0823035

Table 4: Coefficient Estimates for SARIMA(4,1,0)(2,1,0), s = 12

	Coefficient Estimate	Standard Error
ar1	-1.0014214	0.0773782
ar2	-0.7786154	0.1100704
ar3	-0.2218131	0.1120710
ar4	-0.1678101	0.0784439
sar1	-0.3649720	0.0788812
sar2	-0.2119629	0.0847603

Round 2: Testing MA components

After deciding on the parameters $p = 2$ and $P = 2$, I decided to add in moving average components as there are multiple significant ACF values. Since the ACF is statistically significant at lag 12, I set $Q = 1$[2]. Then, I checked moving average orders $q = 0$ to 4, with order 4 being too complex for R to fit. After testing these, I noticed that the confidence intervals for the sesasonal autoregressive components all included zero.

Table 5: AICC Values of SARIMA(2,1,q)(2,1,1), s=12

	q = 0	q = 1	q = 2	q = 3
AICC value	-1490.413	-1488.934	-1493.664	-1494.958

[2]While ACF is statistically significant at lag 36 as well, setting $Q = 3$ was too complex for R to fit the model. Thus, I proceeded with $Q = 1$ only.

Table 6: Coefficient Estimates for SARIMA(2,1,0)(2,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-0.9409356	0.0633053
ar2	-0.5935751	0.0629612
sar1	0.2086367	0.1616859
sar2	-0.0925046	0.1093572
sma1	-0.6645761	0.1546639

Table 7: Coefficient Estimates for SARIMA(2,1,1)(2,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-0.8192727	0.1694066
ar2	-0.5207484	0.1207788
ma1	-0.1858994	0.2272704
sar1	0.2123839	0.1632493
sar2	-0.0716223	0.1126263
sma1	-0.6643491	0.1542993

Table 8: Coefficient Estimates for SARIMA(2,1,2)(2,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-1.0948928	0.1275543
ar2	-0.4807382	0.0911034
ma1	0.0678176	0.1389393
ma2	-0.4924721	0.1742138
sar1	0.1571366	0.1630474
sar2	-0.0566907	0.1094583
sma1	-0.6379171	0.1487927

Table 9: Coefficient Estimates for SARIMA(2,1,3)(2,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-1.0971589	0.1007969
ar2	-0.8133758	0.1021336
ma1	0.0553061	0.1402148
ma2	-0.1164159	0.1494011
ma3	-0.4524387	0.1658548
sar1	0.1855493	0.1706232
sar2	-0.0524938	0.1174390
sma1	-0.6817482	0.1562262

Round 3

As mentioned abov, and evident in the tables, the seasonal autoregressive coefficients all become statistically insignificant when moving average components are introduced; given the estimates and standard errors, all SAR components have 0 in the confidence interval. Thus, I decided to set $P = 0$.

Table 10: AICC Values of SARIMA(2,1,q)(0,1,1), s = 12

	q = 0	q = 1	q = 2	q = 3
AICC value	-1490.413	-1488.934	-1493.664	-1494.958

Table 11: Coefficient Estimates for SARIMA(2,1,0)(0,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-0.9372661	0.0619740
ar2	-0.5987120	0.0623796
sma1	-0.5422082	0.0855990

Table 12: Coefficient Estimates for SARIMA(2,1,1)(0,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-0.7519476	0.1703416
ar2	-0.4841717	0.1281276
ma1	-0.2862356	0.2255288
sma1	-0.5221025	0.0860677

Table 13: Coefficient Estimates for SARIMA(2,1,2)(0,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-1.1084707	0.1108845
ar2	-0.4727463	0.0859532
ma1	0.0752974	0.1199912
ma2	-0.5395726	0.1329035
sma1	-0.5464175	0.0786695

Table 14: Coefficient Estimates for SARIMA(2,1,3)(0,1,1), s = 12

	Coefficient Estimate	Standard Error
ar1	-1.0738708	0.0834400
ar2	-0.8523651	0.0899738
ma1	0.0311135	0.1265682
ma2	-0.0501956	0.1128073
ma3	-0.5188783	0.1458548
sma1	-0.5817023	0.0818884

After all three rounds of model fitting, I decided to proceed with the following three candidates, given that $X_t = bc(U_t)$, the Box-Cox transformed data:

- **Model 1:** SARIMA$(2,1,0)(2,1,0)_{12}$, to test a model with only AR components
- **Model 2:** SARIMA$(2,1,0)(0,1,1)_{12}$, because it has the least coefficients of the models with MA components
- **Model 3:** SARIMA$(2,1,2)(0,1,1)_{12}$, because it has a lower AICC than model 2

Model Equations (1,2, and 3, respectively):

$(1 + 0.9634_{(0.0625)}B + 0.6173_{(0.0613)}B^2)(1 + 0.3699_{(0.0788)}B^{12} + 0.244_{(0.0823)}B^{24})\nabla^1\nabla^{12}X_t = Z_t$

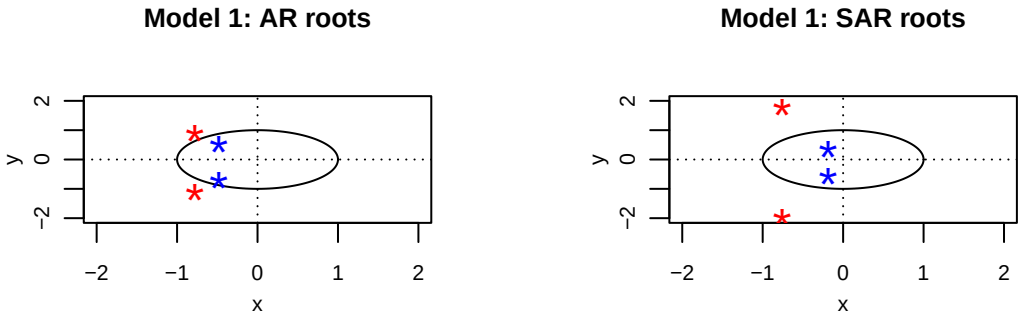
$(1 + 0.9373_{(0.0620)}B + 0.5987_{(0.0624)}B^2)\nabla^1\nabla^{12}X_t = (1 - 0.5422_{(0.0546)}B)Z_t$

$(1 + 1.1089_{(0.1109)}B + 0.4727_{(0.0860)}B^2)\nabla^1\nabla^{12}X_t = (1 - 0.5396_{(0.1329)}B^2)(1 - 0.5464_{(0.0787)}B^{12})Z_t$

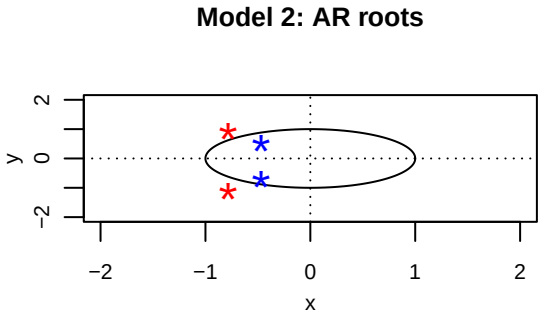
- Note that θ_1 coefficient is assumed to be 0, due to 0 being in the confidence interval.

Stationarity/Invertibility

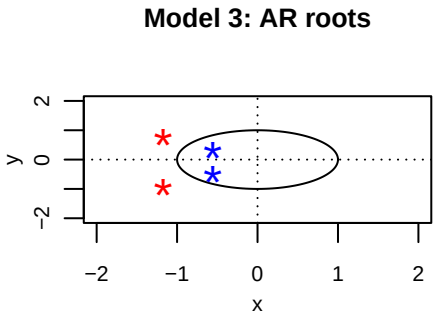
After finalizing three models, I checked each one for stationarity and invertibility by plotting the roots of each characteristic polynomial. For model 1, I looked at the polynomials: $\phi(z) = 1 + 0.9634z + 0.6173z^2$ and $\Phi(z) = 1 + 0.3699z + 0.244z^2$ to check for stationarity. After looking at the plots, we note that all solutions are outside of the unit circle and the model is stationary. The model is invertible as it only has autoregressive components.

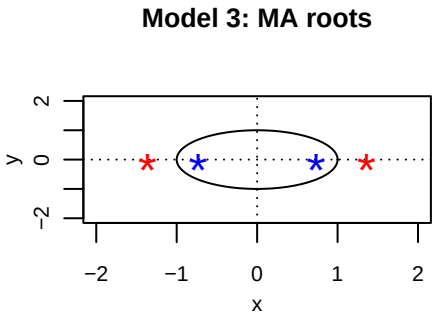


For model 2, I only plot the roots of the characteristic polynomial $\phi(z) = 1 + 0.9373z + 0.5987z^2$, as $\Theta(z) = 1 - 0.5422z$ is of order 1, and since $|\Theta_1| = 0.5422 < 1$ we know the solution is outside of the unit circle. Therefore, it is invertible. After looking at the plot, we see that all solutions are outside of the unit circle and the model is stationary as well.



For model 3, we will look at the characteristic polynomials $\phi(z) = 1 + 1.1085z + 0.4727z^2$ and $\theta(z) = 1 - 0.5396z^2$. Since $\Theta(z) = 1 - 0.5464z$ is of order 1 and $|\Theta_1| = 0.5464 < 1$, we know the solution is outside of the unit circle. After looking at the plot, we see that all solutions are outside of the unit circle and the model is stationary and invertible.



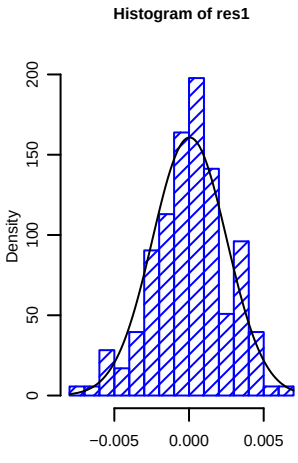


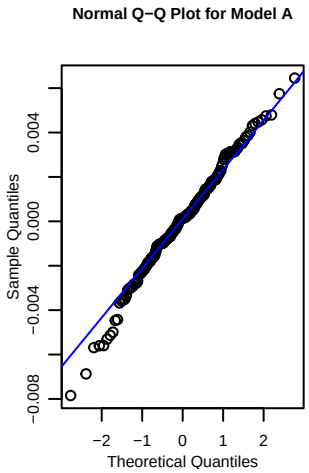
Diagnostic Checking

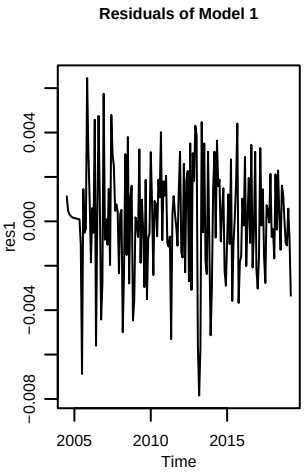
After verifying that all three models are stationary and invertible, I proceeded with diagnostic checking.

Model 1: SARIMA$(2,1,0)(2,1,0)_{12}$

I began by checking the residuals for normality. The qqnorm plot, histogram, and Shapiro-Wilk test suggest the residuals are normally distributed, and the plot of the residuals resemble white noise, which is a good start.

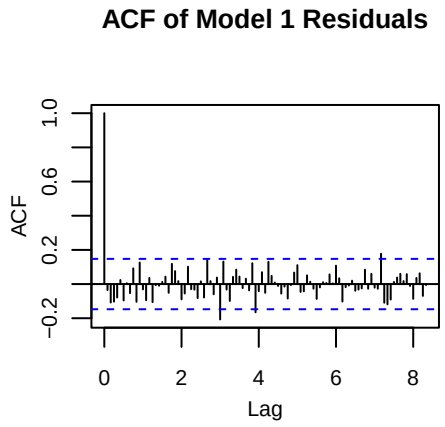


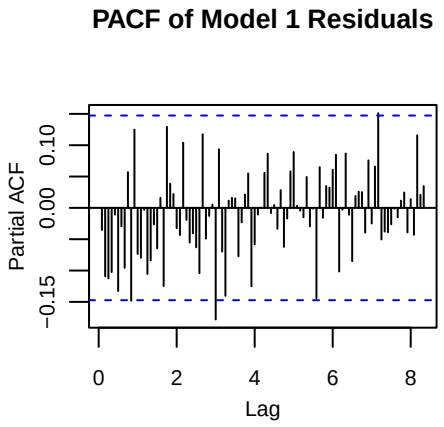





```
##
##  Shapiro-Wilk normality test
##
## data:  res1
## W = 0.99106, p-value = 0.3385
```

Next, I checked the ACF and PACF plots. I noticed that ACF is significant at lags 36 and 48, and PACF is significant at lag 36, which is a cause for concern. I could update the model to have $P = 5$ and $Q = 3$ or 4, but when I attempted to do so the model was too complex for R to handle. However, since the PACF/ACFs are just barely outside of the confidence interval, I will proceed with my diagnostic checking.



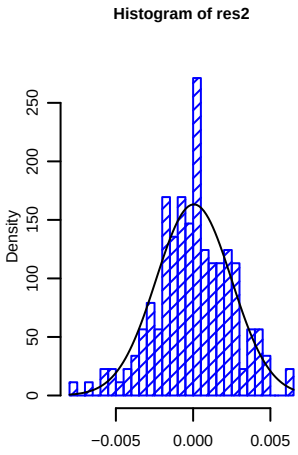


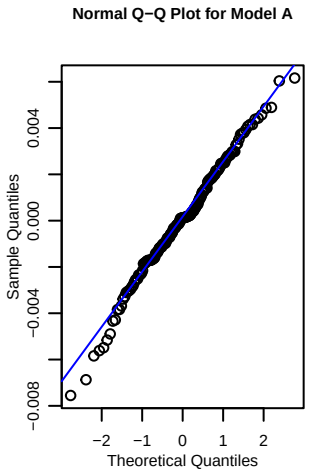
Moving on to Portmanteau tests, I will be testing at the 95% significance level, setting `lag` $= \sqrt{n} = 13$, where n = 177. In addition, `fitdf` $= 4$ as I have 4 estimated coefficients. The residuals passed both tests for linear correlation, and fits into an AR(0) model. However, it fails the McLeod-Li test of squared residuals suggesting some non-linear correlation between the residuals. This will most likely not be our final model, and we move to the next.

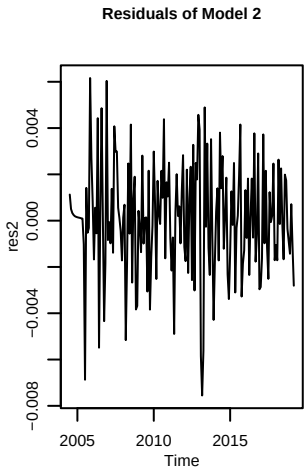
```
##
##  Box-Pierce test
##
## data:  res1
## X-squared = 15.44, df = 9, p-value = 0.07953

##
##  Box-Ljung test
##
## data:  res1
## X-squared = 16.307, df = 9, p-value = 0.06074

##
##  Box-Ljung test
##
## data:  res1^2
## X-squared = 26.998, df = 13, p-value = 0.01245

##
## Call:
## ar(x = res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  6.156e-06
```

Model 2: SARIMA$(2,1,0)(0,1,1)_{12}$

I followed the same steps as above to assess the residuals of model 2. From the qqplot, histogram, Shapiro-Wilk test, and plot, they seem to be normally distributed and resemble white noise.

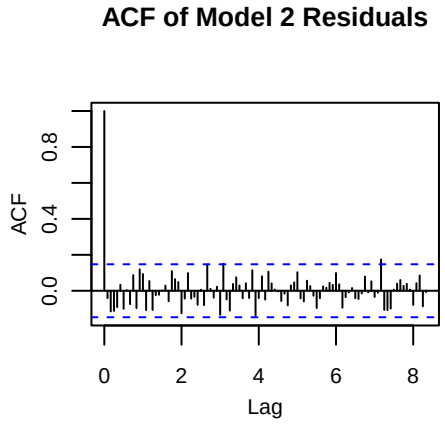


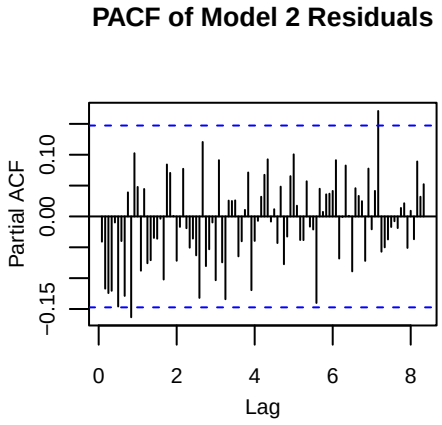





```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.99166, p-value = 0.3987
```

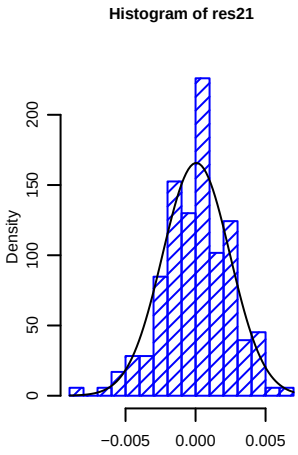
From the ACF and PACF, the PACF is significant at lag 12 and 72. I decided to consider the PACF at lag 72 as within the confidence interval due to Bartlett's formula. Due to this, I decided to update my model to have $P = 1$ for SARIMA$(2,1,0)(1,1,1)_{12}$

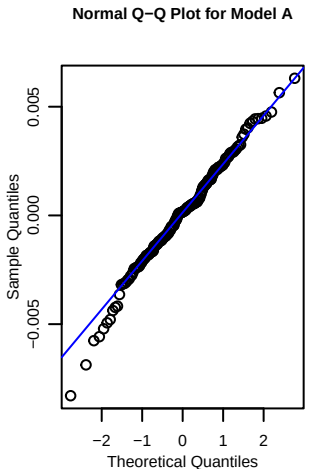


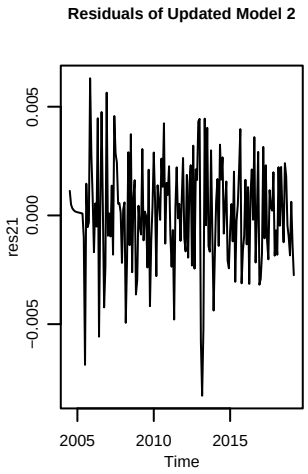


Model 2.1

Based on the PACF of model 1 , I tried updating model 2 to SARIMA$(2,1,0)(1,1,1)_{12}$. I tested and confirmed stationarity and invertibility by plotting characteristic polynomial roots, then proceeded to check the residuals for normality. From all three plots and the Shapiro-Wilk test, we can assume normality.

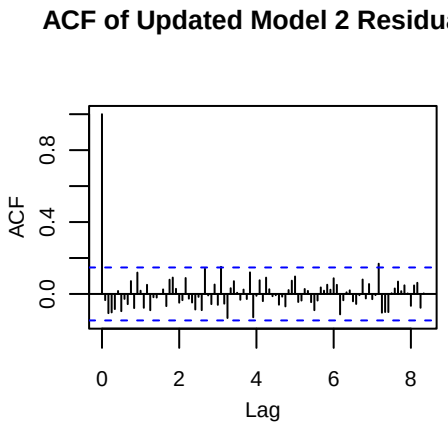


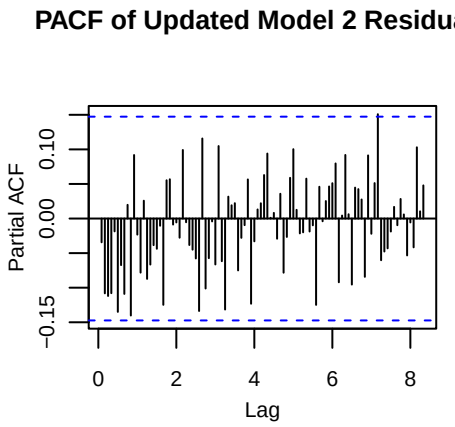





```
##
##  Shapiro-Wilk normality test
##
## data:  res21
## W = 0.99013, p-value = 0.2602
```

The ACF and PACF plots now no longer have values greater than 0 at lags $k > 0$. Thus, we can continue to Portmanteau tests, with `fitdf` $= 4$. However, even after updating model 2, the residuals once again do not pass the McLeod-Li test, suggesting there is still some non-linear correlation even though it fits an AR(0) model. We will now proceed to model 3.



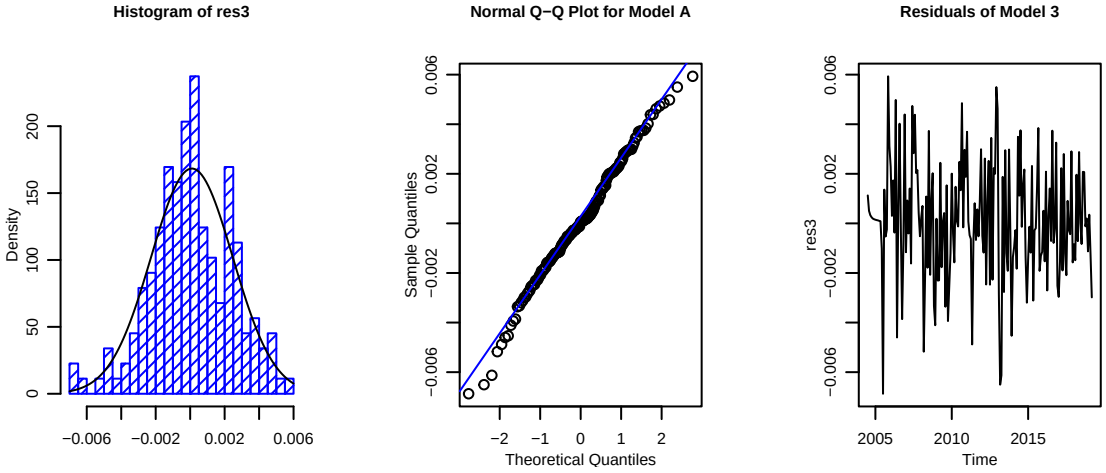



```
##
##  Box-Pierce test
##
## data:  res21
## X-squared = 13.332, df = 9, p-value = 0.1481

##
##  Box-Ljung test
##
## data:  res21
## X-squared = 14.032, df = 9, p-value = 0.1212
```

```
##
##  Box-Ljung test
##
## data:  res21^2
## X-squared = 27.772, df = 13, p-value = 0.009735

##
## Call:
## ar(x = res21, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  5.795e-06
```

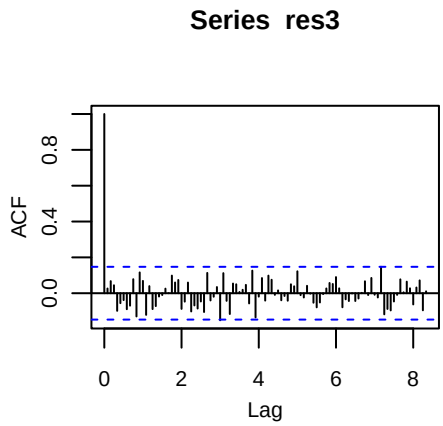
Model 3: SARIMA$(2,1,2)(0,1,1)_{12}$

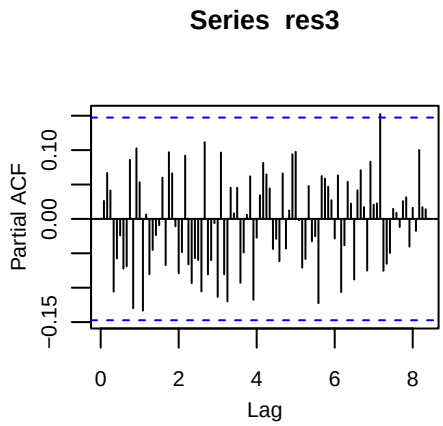
The same steps are repeated. From the three plots and Shapiro-Wilk test, the residuals are assumed to be Gaussian white noise.




```
##
##  Shapiro-Wilk normality test
##
## data:  res3
## W = 0.99195, p-value = 0.4305
```

From the ACF and PACF plots, we can see there are significant values at all lags $k > 0$.





Proceeding with Portmanteau tests with `fitdf` = 4, the residuals pass only one of two tests for linear correlation, but passes the test for squared residuals. In addition, when fit into an AR model, order 0 is suggested meaning our residuals resemble white noise. Because it passes one test for linear correlation, barely failing the other with a p-value extremely close to 0.05, this is my best model and I will use it for model forecasting.

```
##
##  Box-Pierce test
##
## data:  res3
## X-squared = 15.94, df = 9, p-value = 0.06815
```

```
##
##  Box-Ljung test
##
## data:  res3
## X-squared = 16.969, df = 9, p-value = 0.04921
```

```
##
##  Box-Ljung test
##
## data:  res3^2
## X-squared = 12.887, df = 13, p-value = 0.4565
```

Fit into AR:

```
##
## Call:
## ar(x = res3, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  5.615e-06
```

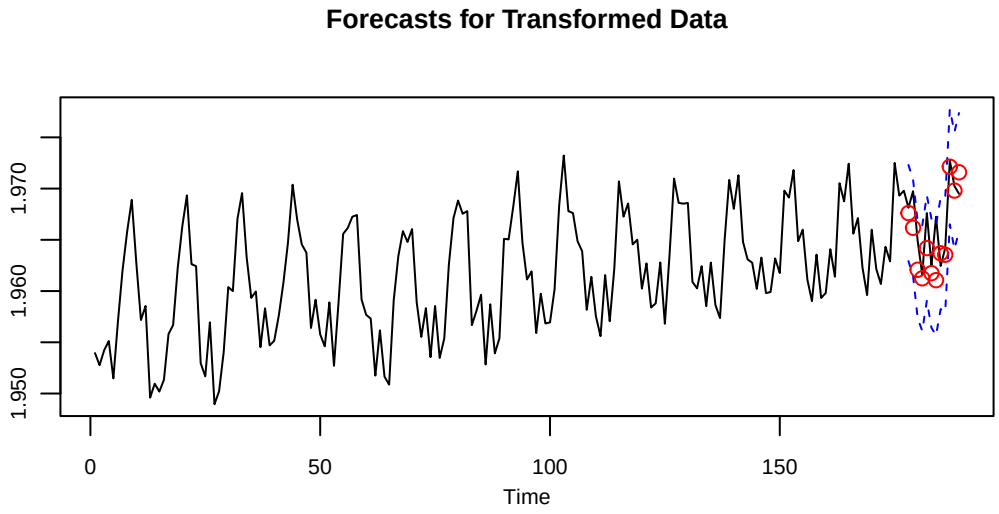
Order 0 selected!

Model Forecasting

Forecasting on Transformed Data

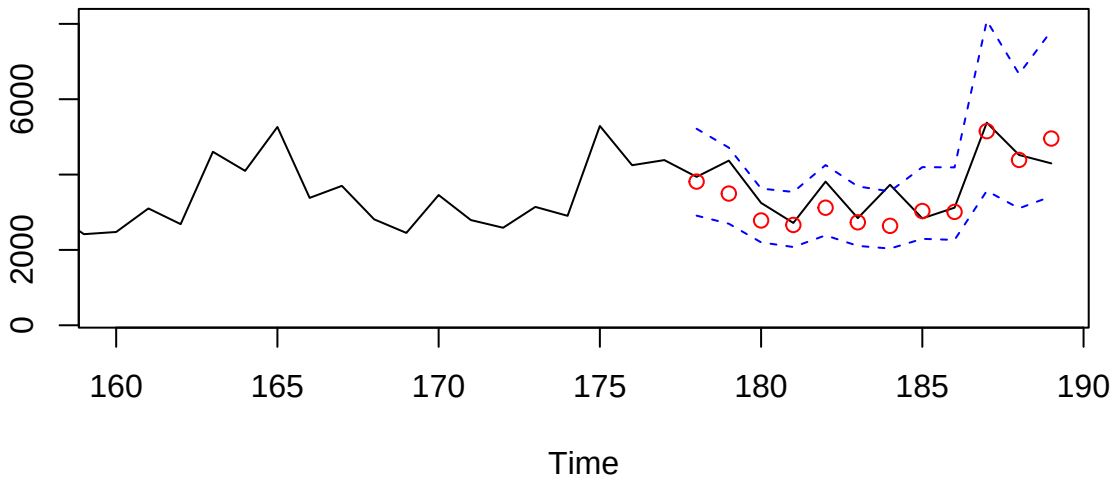
I decided on model 3 as my best fit. While it was not the model that was suggested by my PACF plot (as there is no seasonal AR component), it had the lowest AICc score out of my three candidates.

Using this final model I first forecasted one year ahead on my Box-Cox transformed data, which fits within the blue prediction intervals. Next, I checked the forecasts on my original data.



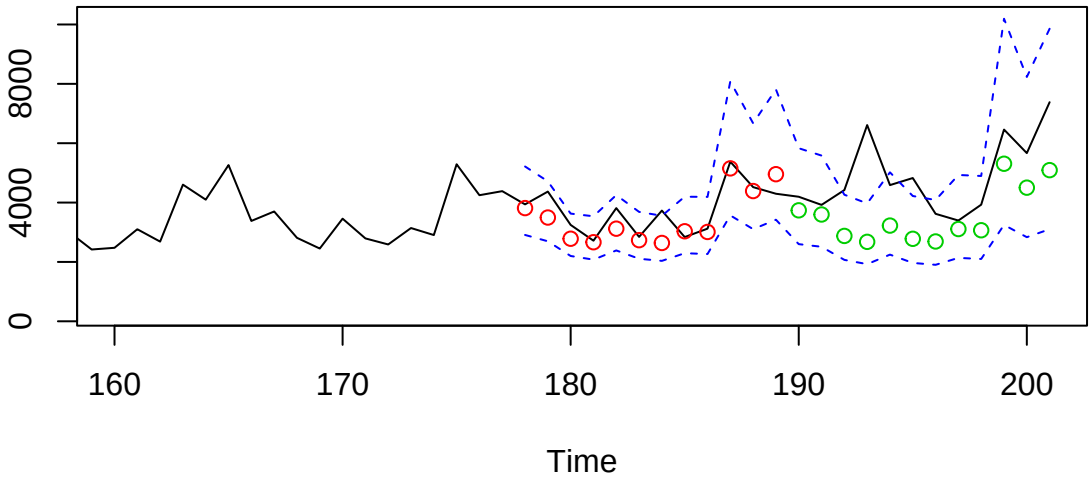
Forecasting on Original Data

To show my predictions on the original data, I created an inverse Box-Cox function and applied it to my one-year-ahead forecasts and prediction intervals. Looking at a zoomed window, the plot of the original data's testing set fits within the prediction interval for the most part, barely lying outside at 7 months ahead. Seeing that my model adequately fits the data, I then moved on to two years ahead.



Two Years Ahead

As mentioned in the introduction, my goal is to forecast what would have happened in the year 2021, absent of the pandemic. To do this, I gathered an additional year of census data, and plotted it against two-year-ahead forecasts. As depicted by the green points on the plot, the forecasts for 2021 are dramatically different for the first 6 months, but align relatively well for the latter half.



Conclusion

After thorough model parameter selection and diagnostic checking, I selected model 3, SARIMA$(2,1,2)(0,1,1)_{12}$ or $(\mathbf{1+1.1089_{(0.1109)}B+0.4727_{(0.0860)}B^2})\nabla^{\mathbf{1}}\nabla^{\mathbf{12}}\mathbf{X_t}=(\mathbf{1-0.5396_{(0.1329)}B^2})(\mathbf{1-0.5464_{(0.0787)}B^{12}})\mathbf{Z_t}$, where X_t is the Box-Cox transformed data. Since it had the most coefficients and did not seem to match the parameters my PACF plot suggested, I was surprised that this was my final model. However, it did have the lowest AICc score and we can see the trade-off between number of parameters and a lower AICc. While it failed to pass the Ljung-Box test for linear correlation, it passes the Box-Pierce test. More importantly, the testing set fits within the prediction interval of the model forecasts, suggesting an adequate model fit. However, since it does not pass all diagnostic checking tests, perhaps more complex models can be explored.

The other goal of this project was to predict the number of business applications in the years 2020-2021 absent of the pandemic. From our model, we can in fact see that the predictions deviate from the real data, suggesting the pandemic may have caused a surge in business applications.

References/Acknowledgements

With help from:

Professor Raya Feldman

Teaching Assistants Thiha Aung and LiHao Xiao

Student Stella Jia

References:

Data sourced from the U.S. Census Bereau

174 Lecture Notes

Appendix

All code is shown below. Some chunks are omitted to reduce redundancy; for example, I fit 10 different models using `arima()` but only included the code for one as it is the exact same code but with different model parameters for all 10.

```
# Libraries:
library(kableExtra)
library(dplyr)
library(tidyr)
library(lubridate)
library(forecast)
library(MASS)
library(ggplot2)
library(ggfortify)
library(MuMIn)
library(tfplot)
library(tframe)
library(dse)
library(forecast)
require(graphics)

# Importing/Cleaning/EDA
agr <- read.csv("agr.csv")
agr$time <- paste0("01-", agr$time)
agr$time <- as.Date(agr$time, format = "%d-%b-%Y")
agr$value <- as.numeric(gsub(",", "", agr$value))
agr.test <- agr[178:189,]
agr.train <- agr[1:177,]
agr.ts <- ts(agr.train$value, start = c(2004,7,1), frequency = 12)
ts.plot(agr.ts, main = "Monthly Business Applications, Agriculture U.S.", ylab = "Number of Applications

# transformation/plotting
t <- 1:length(agr.ts)
bcTransform <- boxcox(agr.ts ~ t, plotit = F)
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
agr.bc = (1/lambda)*(agr.ts^lambda-1)
agr.log = log(agr.ts)
agr.sqrt = sqrt(agr.ts)
plot(agr.ts, main = "Untransformed Data",cex.main = 0.8)
hist(agr.ts, main = "Untransformed Data",cex.main = 0.8)
plot(agr.bc, main = "Box-Cox Transformed",cex.main = 0.8)
hist(agr.bc, col = "steelblue1", main = "Box-Cox Transformed",cex.main = 0.8)
plot(agr.log, main = "Log Transformed",cex.main = 0.8)
hist(agr.log, main = "Log Transformed",cex.main = 0.8)
plot(agr.sqrt, main = "Square-Root Transformed",cex.main = 0.8)
hist(agr.sqrt, main = "Square Root Transformed",cex.main = 0.8)
plot(decompose(agr.bc))

# Differencing
dagr <- diff(agr.bc, 1)
ddagr <- diff(dagr, 12)
plot(dagr, main = "First Difference (lag 1)",cex.main = 0.8)
hist(dagr, main = "Histogram of First Difference (lag 1)",cex.main = 0.8)
```

```r
plot(ddagr, main = "Second Difference (lag 12)",cex.main = 0.8)
hist(ddagr, main = "Histogram of Second Difference (lag 12)", cex.main = 0.8)
vars <- data.frame(c(var(agr.bc)), c(var(dagr)), c(var(ddagr)))
colnames(vars) <- c("Undifferenced Data", "First Difference (lag 1)", "Second Difference (lag 12)")
rownames(vars) <- c("Variance")
kable(vars, caption = "Variances after Differencing") %>%
  kable_styling(latex_options = "HOLD_position")

# P/ACF Plots, same code for all three plots, only one is shown
acf(agr.bc, lag.max = 100, main = "")
title(main = "ACF: Undifferenced", line = 1)
pacf(agr.bc, lag.max = 100, main = "")
title(main = "PACF: Undifferenced", line = 1)

# Finding AIC and creating tables. This format is used for each round of testing,
# only one is shown for the sake of reducing repetition
ar <- data.frame(c(AICc(arima(agr.bc, order=c(2,1,0), seasonal = list(order = c(2,1,0), period = 12), m
colnames(ar) <- c("p = 2", "p = 4")
rownames(ar) <- c("AICC value")
kable(ar, caption = "AICC Values of SARIMA(p,1,0)(2,1,0), s=12") %>%
  kable_styling(latex_options = "HOLD_position")

# Model fittng and creating tables of coefficients/se. Format is the same for each model,
# only one model is shown for the sake of reducing repetition
m1<- arima(agr.bc, order=c(2,1,0), seasonal = list(order = c(2,1,0), period = 12), method="ML")
round1.model1 <- data.frame(m1$coef, sqrt(diag(vcov(m1))))
colnames(round1.model1) <- c("Coefficient Estimate", "Standard Error")
kable(round1.model1, caption = "Coefficient Estimates for SARIMA(2,1,0)(2,1,0), s = 12") %>%
  kable_styling(latex_options = "HOLD_position")

# Plotting unit circles
plot.roots(NULL,polyroot(c(1, 0.9634,0.6173)), main="Model 1: AR roots")
plot.roots(NULL,polyroot(c(1, 0.3699,0.244)), main="Model 1: SAR roots")
plot.roots(NULL,polyroot(c(1, 0.9373, 0.5987)), main="Model 2: AR roots")
plot.roots(NULL,polyroot(c(1, 1.1085, 0.4727)), main="Model 3: AR roots")
plot.roots(NULL,polyroot(c(1, 0, -0.5396)), main="Model 3: MA roots")

# Getting residuals for diagnostic checking
model1 <- arima(agr.bc, order=c(2,1,0), seasonal = list(order = c(2,1,0), period = 12),
                method="ML")
res1 <- residuals(model1)
model2 <- arima(agr.bc, order=c(2,1,0), seasonal = list(order = c(0,1,1), period = 12), method="ML")
res2 <- residuals(model2)
model_21 <- arima(agr.bc, order=c(2,1,0), seasonal = list(order = c(1,1,1), period = 12), method="ML")
res21 <- residuals(model_21)
model3 <- arima(agr.bc, order=c(2,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML", fi
res3 <- residuals(model3)

# Diagnostic checking
# Normality checking (only one of three models is shown, same code for all 3)
hist(res1,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m1 <- mean(res1)
std1 <- sqrt(var(res1))
```

```r
curve(dnorm(x,m1,std1), add=TRUE )
qqnorm(res1,main= "Normal Q-Q Plot for Model A")
qqline(res1,col="blue")
plot(res1, main = "Residuals of Model 1")
shapiro.test(res1)

#ACF/PACF of residuals (same code for all 3)
acf(res1, lag.max = 100, main = "ACF of Model 1 Residuals")
pacf(res1, lag.max = 100, main = "PACF of Model 1 Residuals")

# Checking updated model 2:
arima(agr.bc, order=c(2,1,0), seasonal = list(order = c(1,1,1), period = 12), method="ML")
plot.roots(NULL,polyroot(c(1, 0.9327, 0.5897)), main="Model 2.1: AR roots")

# Portmanteau test for model 1:
Box.test(res1, lag = 13,  type = c("Box-Pierce"), fitdf = 4)
Box.test(res1, lag = 13,  type = c("Ljung-Box"), fitdf = 4)
Box.test(res1^2, lag = 13,  type = c("Ljung-Box"), fitdf = 0)
# Model 2.1:
Box.test(res21, lag = 13,  type = c("Box-Pierce"), fitdf = 4)
Box.test(res21, lag = 13,  type = c("Ljung-Box"), fitdf = 4)
Box.test(res21^2, lag = 13,  type = c("Ljung-Box"), fitdf = 0)
# Model 3:
Box.test(res3, lag = 13,  type = c("Box-Pierce"), fitdf = 4)
Box.test(res3, lag = 13,  type = c("Ljung-Box"), fitdf = 4)
Box.test(res3^2, lag = 13,  type = c("Ljung-Box"), fitdf = 0)

 ar(res3, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# Forecasting on transformed data
pred.tr <- predict(model3, n.ahead = 12)
u.tr= as.vector(pred.tr$pred + 1.96*pred.tr$se)
l.tr= as.vector(pred.tr$pred - 1.96*pred.tr$se)
agr.bc.num <- as.vector(agr.bc)
agr.bc.og <- agr
agr.bc.og$value <- (1/lambda)*(agr.bc.og$value^lambda-1)
agr.bc.og.num <- as.vector(agr.bc.og)
par(mgp = c(1.5, 0.7, 0), cex.axis = 0.7, cex.lab = 0.7,oma = c(0, 0, 0, 2), cex.main = 0.8)
ts.plot(agr.bc.og, xlim=c(1,length(agr.bc.num)+12), ylim = c(min(agr.bc.num),max(u.tr)), main = "Foreca
lines((length(agr.bc.num)+1):(length(agr.bc.num)+12),u.tr, col="blue", lty="dashed")
lines((length(agr.bc.num)+1):(length(agr.bc.num)+12),l.tr, col="blue", lty="dashed")
points((length(agr.bc.num)+1):(length(agr.bc.num)+12), pred.tr$pred, col="red")

# Forecasting on original data
undo <- function(x){
  (x*lambda+1)^(1/lambda)
}
pred.og <- undo(pred.tr$pred)
u.og <- undo(u.tr)
l.og <- undo(l.tr)
ts.plot(agr, xlim = c(160,length(agr.ts)+12), ylim = c(250,max(u.og)))
lines((length(agr.ts)+1):(length(agr.ts)+12),u.og, col="blue", lty="dashed")
lines((length(agr.ts)+1):(length(agr.ts)+12),l.og, col="blue", lty="dashed")
```

```
points((length(agr.ts)+1):(length(agr.ts)+12), pred.og, col="red")

# Forecasting two years ahead
pred.tr2 <- predict(model3, n.ahead = 24)
u.tr2= pred.tr2$pred + 1.96*pred.tr2$se
l.tr2= pred.tr2$pred - 1.96*pred.tr2$se
pred.og2 <- undo(pred.tr2$pred)
u.og2 <- undo(u.tr2)
l.og2 <- undo(l.tr2)
ts.plot(agr2, xlim = c(160, length(agr.ts)+24), ylim = c(250,max(u.og2)))
lines((length(agr.ts)+1):(length(agr.ts)+24),u.og2, col="blue", lty="dashed")
lines((length(agr.ts)+1):(length(agr.ts)+24),l.og2, col="blue", lty="dashed")
  points((length(agr.ts)+1):(length(agr.ts)+12), pred.og2[1:12], col="red")
points((length(agr.ts)+13):(length(agr.ts)+24), pred.og2[13:24], col="green3")
```