PSTAT 174 Gold Rates Final Project

Rebecca Chang

June 9th, 2023

Executive Summary

This project focuses on total quarterly gold rates per troy ounce (around 31.1 grams) from the years 1985 to 2021 in US dollars. The goal of the project is to create a time series model that accurately forecasts the quarterly gold rates for the year 2021 by comparing it to the actual quarterly gold rates in 2021 from the data. I first began by splitting the original data into training and testing sets, with the last year (last 4 points) as the testing set. Then I performed a log transformation to stabilize the variance of the training data. I also differenced once at lag 1 to remove trend while not over-differencing. From there, I plotted the ACF and PACF graphs of the training data and came up with potential models to test. I fit these models and estimated the parameters of each of these models. Based on the AICc, I compared all the models and chose two models with the lowest AICc. Between these two models, I manually checked if the coefficient estimates were within the 95% confidence interval and changed its value to 0 if it was, since that would mean that it is insignificant. Then I proceeded to model diagnostics for these two models where I checked if the residuals of these models were independent, normal, had nonlinear dependence, and followed a white noise model. If the model passed all the diagnostic checks, I could consider it for forecasting. Since one of my models passed and the other didn't, I used the model that passed diagnostic checking for forecasting. In the forecasting step, I predicted the 4 future values which were the same as the ones in my testing set so I could make a comparison and check the accuracy of my model. The forecast yielded results that were higher than the true values, but the true values were still within the 95% prediction confidence interval. I also predicted another 4 points to see how my model would predict 2 years ahead from my training set, out of curiosity. Ultimately, I learned how to come up with the best model to make forecasts on data sets. Through this project, I also learned that the ability to accurately predict future gold rates can help us learn more about the US economy.

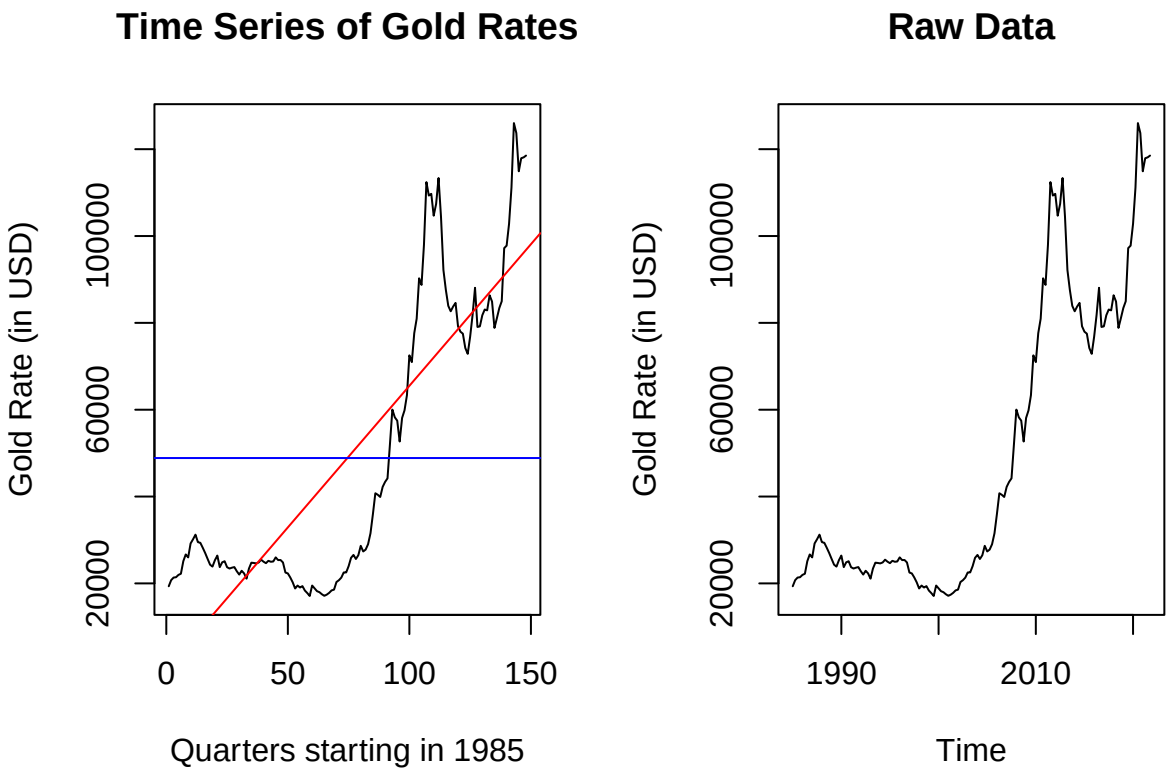
Introduction

The data I chose shows gold rates throughout the years 1985 to 2021. It has a quarterly frequency and is measured by total USD per troy ounce. I obtained this data through the website Kaggle and downloaded the csv file that had daily gold rates. I then used pivot tables in Microsoft Excel to clean and convert my data into quarterly data to ensure a less noisy and smoother-looking time series plot. I also took out columns that had the gold rates in currencies other than USD. Then, I imported this data into RStudio to perform my analyses and forecasts. I found this data interesting because it can give valuable insights on the US economy at specific time periods in history. For example, during the 2008 financial crisis, people were eager to first sell their liquid assets like gold, which caused the value of gold to decrease considerably. However, in 2020 when the COVID-19 pandemic hit, many investors searched for safe havens and a store of value which caused gold rates to rise. These events can be corroborated by the data as seen in the dips in gold rates during the second half of 2008 and the gradual increase in gold rates throughout 2020. Hence, the goal of this project is to find a time series model that can accurately forecast future quarterly gold rates. In order to solve this problem, I first split my original data into a testing set that included the last year of data (last 4 observations) and a training set with the rest of my observations. This way, I could use my training set to determine the best parameters for my model and test it on my testing set to make a comparison of the true

and predicted values. In the forecasting step, I created a time series plot on the training set and predicted 8 steps ahead with 4 for comparison with my testing set and another 4 out of curiosity, all with 95% confidence intervals. The results produced wide confidence intervals which implies that the margin of error could be large. Although the predicted values were higher than the true gold rate values, the actual values were still within my 95% confidence interval. I concluded that my 8 future values followed a positive linear trend, but it likely would not be very accurate if this model was used to predict values further into the future.

Data Importation

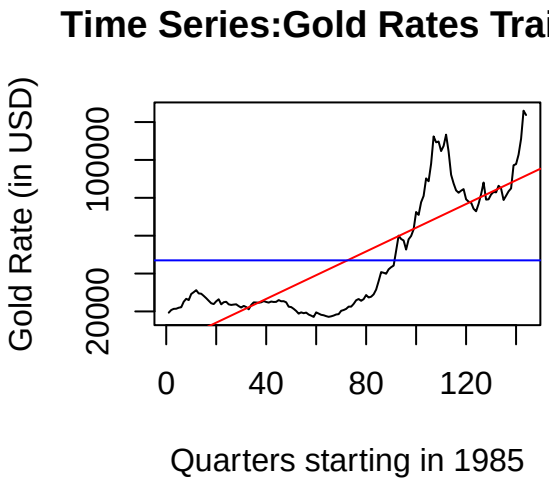
I first read in my data and plotted the quarterly time series plot for gold rates in the years 1985 to 2021. I also added a red line for trend and a blue line for the mean.

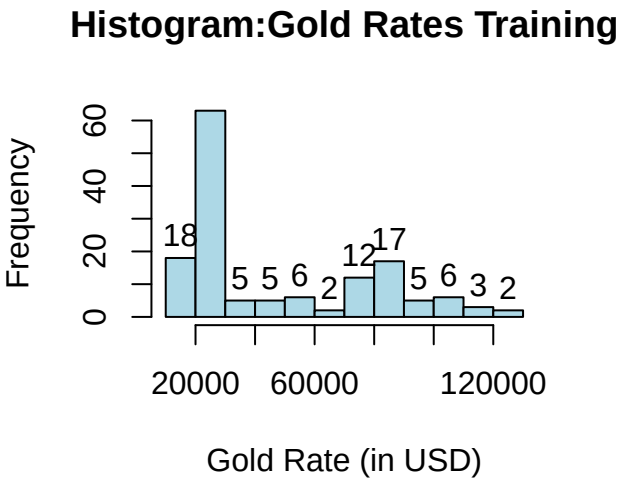


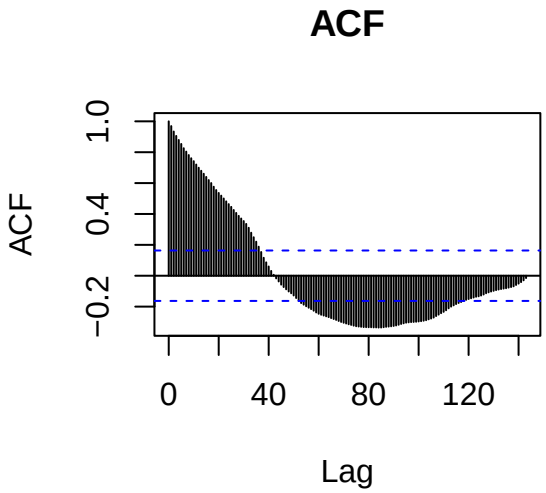
We can immediately observe a positive linear trend as seen by the red line with a constant positive slope. There seems to be no obvious seasonality, like quarterly seasonality in this case, as there is no consistent pattern occurring at equal time intervals. There also does not seem to be any sharp changes in behavior as seen by the absence of sudden spikes. The graph shows a non-constant variance and a mean of around $50,000.

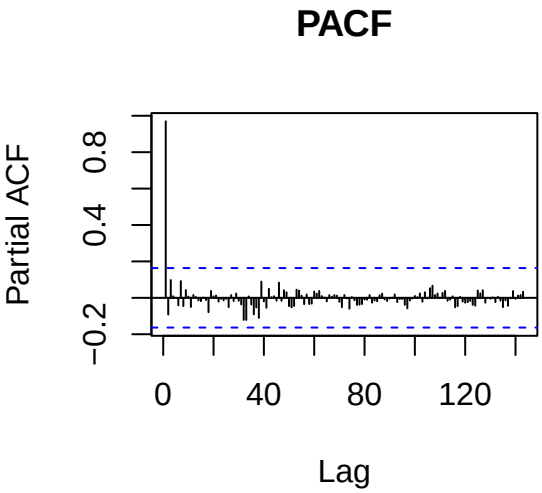
Training/Testing Sets

I then split my data into training and testing sets, with the testing set being the last year of observations. I plotted a time series plot, histogram, ACF (Autocorrelation Function) graph, and PACF (Partial Autocorrelation Function) graph for the training set.







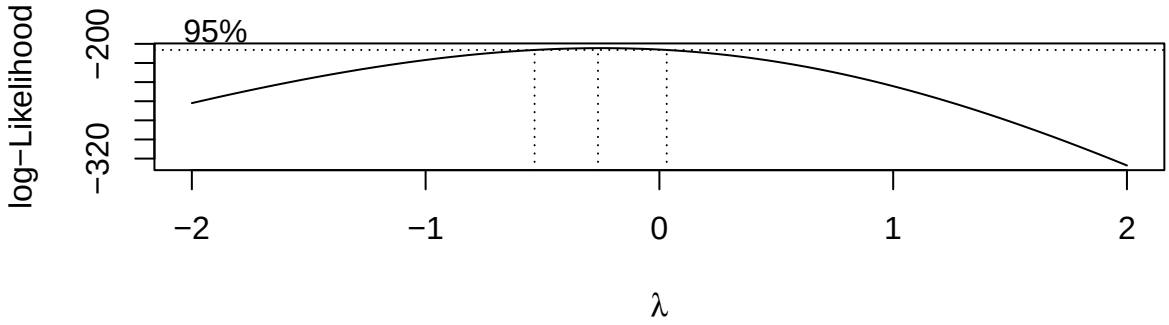


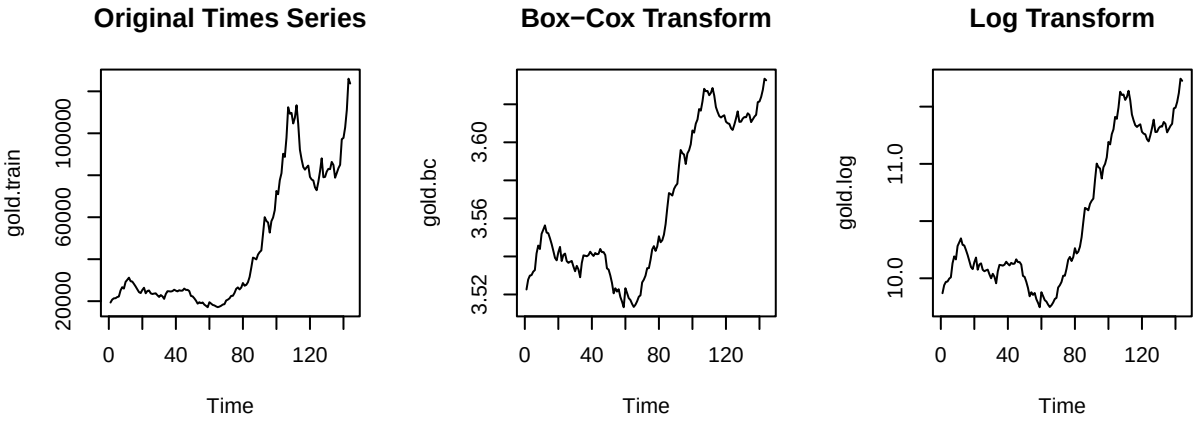
I noticed that the time series plot looks about the same as the original data with a positive linear trend, no seasonality, and no sharp changes. The histogram looks unimodal and skewed to the the right. In addition, the graph of the ACF appears to be very large and the graph of the PACF shows no significant lags other than at lag 0, which will always be 1.

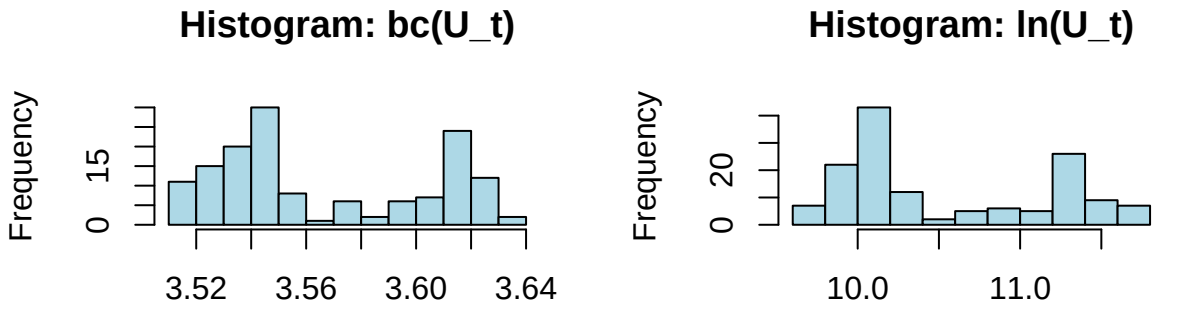
Visualization Techniques

Transformation

Next, I performed necessary transformations to stabilize the variance of my training set since the data seems to have pretty uneven variances throughout the plot.







I used the `boxcox` function to determine the optimal λ value. Since 0 is in the 95% confidence interval, I also considered a log transformation in addition to the Box-Cox transformation. I plotted time series graphs of both the Box-Cox and log transformations and compared them to my original time series data in order to see which was able to stabilize the variance better. I also plotted histograms of both transformations for additional visualizations. Since the time series graphs looked relatively similar, I ended up choosing the log transformation as its histogram looked slightly more symmetric.

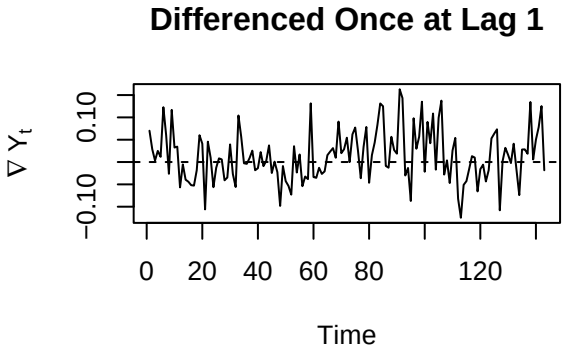
Differencing

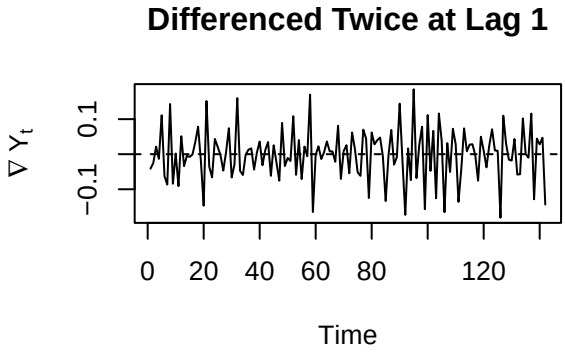
I also want to check how much differencing is needed to remove trend. I first started by differencing my log-transformed data once at lag 1, plotting it, and determining its variance. I then differenced again at lag 1 to see if the variance was lowered. Since the variance actually increased, I stopped differencing at lag 1 and kept d=1 as to not over-difference.

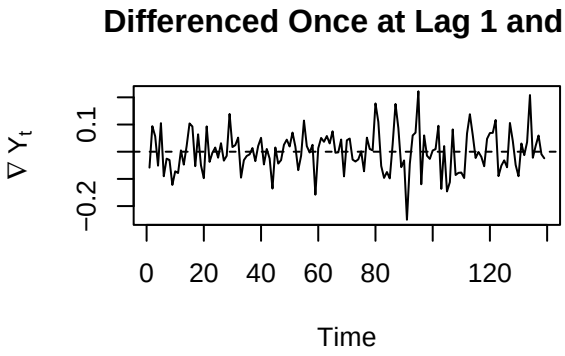
Additionally, I checked if there was any possible seasonality by differencing once at lag 4 (since I have quarterly data) and checking the variance. I once again compared this variance to the variance of my log-transformed data that was differenced once and realized it had increased.

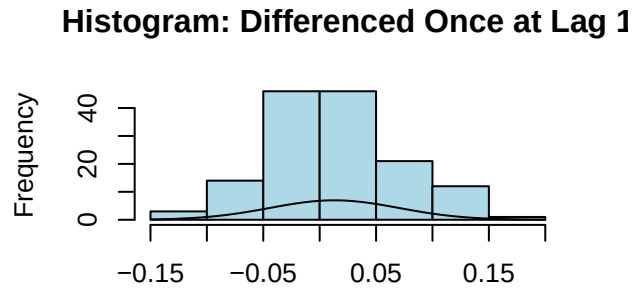
Therefore, I chose to keep my model with d=1 and D=0.

```
## # A tibble: 3 x 2
##   Variance Model
##      <dbl> <chr>
## 1  0.00328 Differenced Once at Lag 1
## 2  0.00523 Differenced Twice at Lag 1
## 3  0.00559 Differenced Once at Lag 1 and Lag 4
```





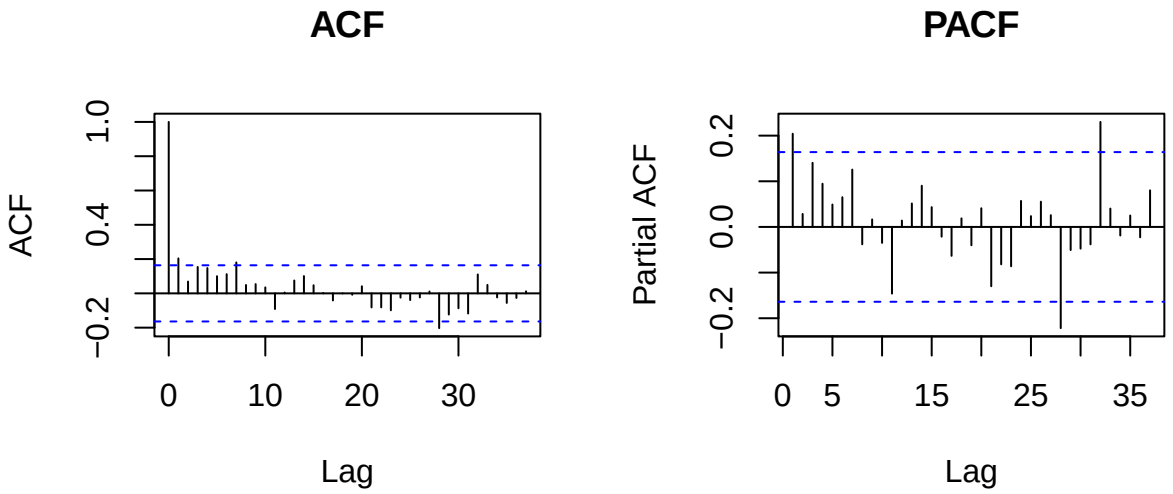




After plotting the time series plot after each try at differencing, I realized the model I chose to keep looked more stationary with a more constant variance. The histogram of my data differenced once at lag 1 also looked much better and more normally distributed compared to before.

Model Identification

Next, I plotted the ACF and PACF for my transformed and differenced data. I set the maximum lag as 37, which was a quarter of my total observations.



Looking at the ACF plot, I noticed significant values at lags 1, 7, and 28. So, I can consider q = 28, but I can also consider q = 1 or 7 for a simpler model. Looking at the PACF plot, I noticed significant values at lags 1, 28, and 32. So, I can consider p = 32, but I can also consider p = 1 or 28.

Since there is no seasonality component in my data, I will be mostly considering ARIMA (Autoregressive Integrated Moving Average) models. I can set d = 1 since I differenced once at lag 1 to remove trend and D = 0 since I did not difference to remove seasonality.

I also considered some SARIMA (Seasonal ARIMA) models since some lags that showed significance were multiples of 4, which means there could still be some slight seasonality. For these models, I kept the nonseasonal AR part as 1 since it had a significant PACF value and the MA part to be 0 or 1 since the ACF showed a value of lag 1 that was only slightly over the 95% confidence interval. I also set the seasonal AR part to be 8 since the PACF has a last significant lag of 32. Just to test out more models, I also included some pure MA (Moving Average) and AR (Autoregressive) models at lower-valued significant lags.

I came up with these potential models by trying out different variations of these parameters.

1) ARIMA(1,1,1)
2) ARIMA(1,1,28)
3) ARIMA(1,1,32)
4) ARIMA(7,1,1)
5) ARIMA(7,1,28)
6) ARIMA(7,1,32)
7) SARIMA (1,1,1) x $(8,0,0)_4$
8) SARIMA (1,1,0) x $(8,0,0)_4$
9) MA(1)
10) MA(7)
11) AR(1)

Model Estimation

After determining some potential models, I created each of these models in order to obtain the model coefficients and compare their AICc (Akaike Information Criterion, Corrected for Bias) values. I made a table of each of the models as well as their corresponding AICc values in ascending order.

```
## # A tibble: 11 x 2
##     AICc Models
##    <dbl> <chr>
##  1 -415. ARIMA(1,1,1)
##  2 -411. SARIMA(1,1,0)x(8,0,0)_4
##  3 -407. SARIMA(1,1,1)x(8,0,0)_4
##  4 -406. ARIMA(7,1,1)
##  5 -397. AR(1)
##  6 -391. ARIMA(1,1,28)
##  7 -381. ARIMA(1,1,32)
##  8 -379. ARIMA(7,1,28)
##  9 -368. ARIMA(7,1,32)
## 10 -292. MA(7)
## 11  101. MA(1)
```

I found that my ARIMA(1,1,1) model and SARIMA(1,1,0)x$(8,0,0)_4$ model both yielded the lowest AICc.

I then calculated the 95% confidence interval for each coefficient in these models to determine if they were significant. If 0 was in the confidence interval, I set that coefficient to 0 and re-evaluated its AICc value.

```
##
## Call:
## arima(x = gold.log, order = c(1, 1, 1), method = "ML")
##
## Coefficients:
##          ar1      ma1
##       0.9168  -0.7792
## s.e.  0.0668   0.0998
##
## sigma^2 estimated as 0.003086:  log likelihood = 210.3,  aic = -414.59

## [1] 0.785872 1.047728

## [1] -0.974808 -0.583592
```

For the ARIMA model, I printed the confidence intervals for both coefficients. Neither contained 0 which means the model did not have any insignificant coefficients, so the AICc remained the same.

```
##
## Call:
## arima(x = gold.log, order = c(1, 1, 0), seasonal = list(order = c(8, 0, 0),
##     period = 4), method = "ML")
##
## Coefficients:
##          ar1    sar1    sar2     sar3    sar4    sar5    sar6     sar7    sar8
##       0.2122  0.1805  0.0338  -0.0286  0.0111  0.1051  0.0170  -0.2403  0.2635
## s.e.  0.0876  0.0835  0.0833   0.0837  0.0778  0.0844  0.0865   0.0848  0.0874
##
## sigma^2 estimated as 0.002781:  log likelihood = 215.93,  aic = -411.85
```

```
##
## Call:
## arima(x = gold.log, order = c(1, 1, 0), seasonal = list(order = c(8, 0, 0),
##     period = 4), fixed = c(NA, NA, 0, 0, 0, 0, 0, NA, NA), method = "ML")
##
## Coefficients:
##          ar1    sar1  sar2  sar3  sar4  sar5  sar6     sar7    sar8
##       0.2155  0.1813     0     0     0     0     0  -0.2270  0.2646
## s.e.  0.0839  0.0826     0     0     0     0     0   0.0843  0.0875
##
## sigma^2 estimated as 0.002824:  log likelihood = 215.01,  aic = -420.01
```

```
## [1] -418.6697
```

However, the SARIMA model had a few insignificant coefficients so I set the new coefficients and obtained a new AICc, which actually became lower than that of the ARIMA model.

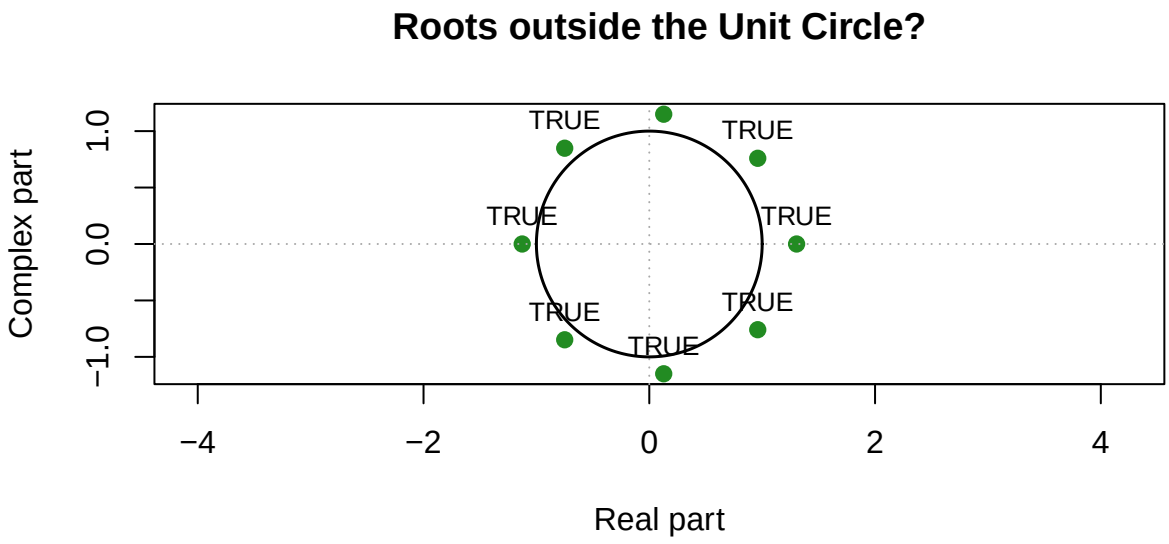
Stationarity and Invertibility

For both of these models, I then checked to make sure that they were both stationary and invertible.

The SARIMA model has the equation: $(1-0.2155B)(1-0.1813B^4+0.227B^{28}-0.2646B^{32})(1-B)X_t = Z_t$

Since the SARIMA model only has an AR part, the model is automatically invertible, so I only have to check for stationarity. Using the coefficients from the equation above, I plotted the roots of the seasonality portion and saw that all the roots were outside the unit circle. The roots of the nonseasonal portion is also outside the unit circle as seen directly from the equation since $|-0.2155| < 1$. Hence, this model is both stationary and invertible.

```
##        real   complex outside
## 1  0.960731  0.759316    TRUE
## 2 -0.749350  0.848606    TRUE
## 3 -0.749350 -0.848606    TRUE
## 4  0.960731 -0.759316    TRUE
## 5  0.127963  1.149683    TRUE
## 6 -1.125930  0.000000    TRUE
## 7  0.127963 -1.149683    TRUE
## 8  1.305142  0.000000    TRUE
## *Results are rounded to 6 digits.
```

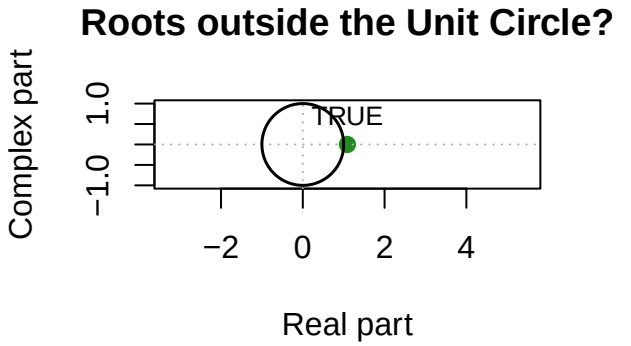


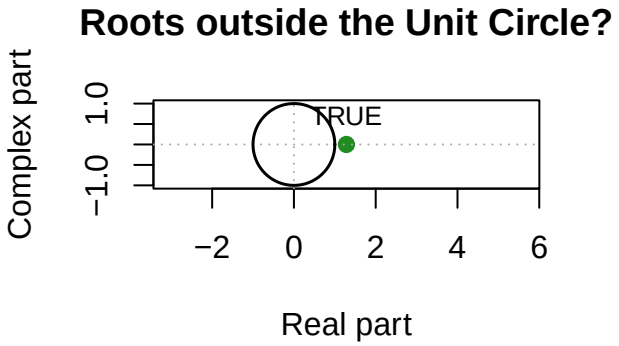
The ARIMA model has the equation: $(1 - 0.9168B)(1 - B)X_t = (1 - 0.7792B)Z_t$.

When I calculated the 95% confidence interval for the AR part of the model, 1 was contained in the interval which means it could be not stationary. However, since it is only slightly over the interval (calculated above) and the roots are still outside the unit circle, I chose to continue with the model. Once again, I checked the MA part of the equation for invertibility and the AR part of the equation for stationarity by plotting the roots on the unit circle. I saw that the roots were outside the unit circle, so the model is stationary and invertible. It is also possible to conclude that this model is both stationary and invertible by looking at the equation since both $|-0.09168| < 1$ and $|-0.7792| < 1$.

```
##      real complex outside
## 1 1.09075       0    TRUE
## *Results are rounded to 6 digits.
```

```
##       real complex outside
## 1 1.283368       0    TRUE
## *Results are rounded to 6 digits.
```

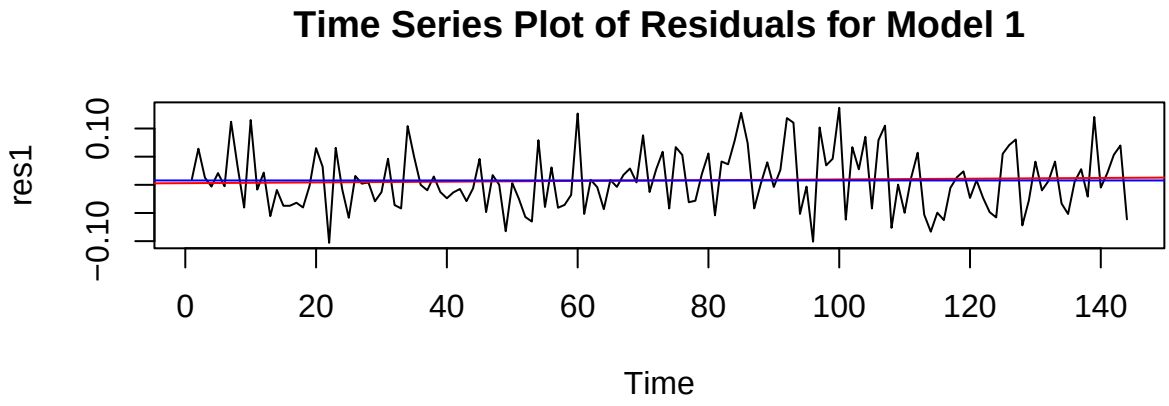


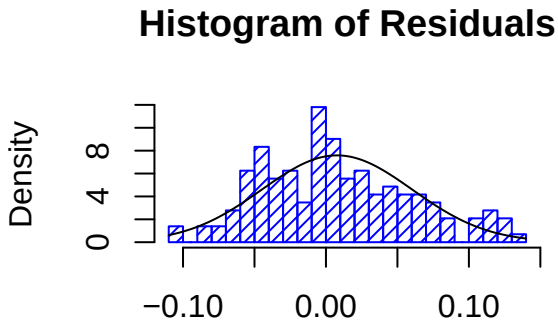


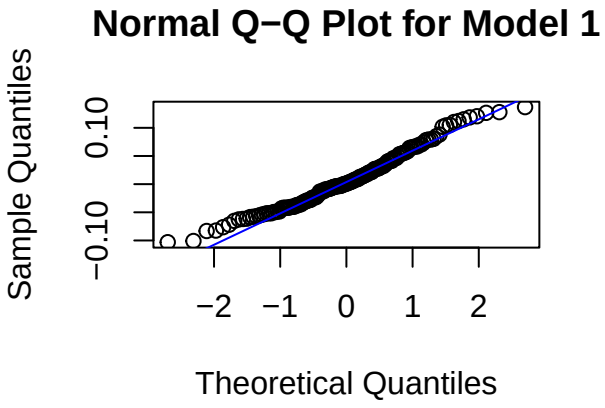
Since the SARIMA model yielded a lower AICc, I will proceed with diagnostic checking for this model first.

Diagnostics Checking for Model 1

To perform the diagnostics checking for both models, I first plotted a time series plot of the residuals of the model. To check normality, I plotted a histogram and a Q-Q plot along with a Q-Q line. I also performed the Shapiro-Wilk test. I plotted the ACF and PACF of the residuals to make sure they generally follow a white noise model and conducted the Ljung-Box test on the residuals to check the white noise hypothesis. I also used the `ar` function to double check the selected order is equal to 0 which corresponds to white noise. I proceeded with the Box-Pierce test to check if there was correlation between residuals and the McLeod-Li test to check non-linear dependence based on the squared residuals. Lastly, I plotted the ACF of squared residuals to see if there were still significant lags.








```
##
##  Shapiro-Wilk normality test
##
## data:  res1
## W = 0.9782, p-value = 0.02133
```

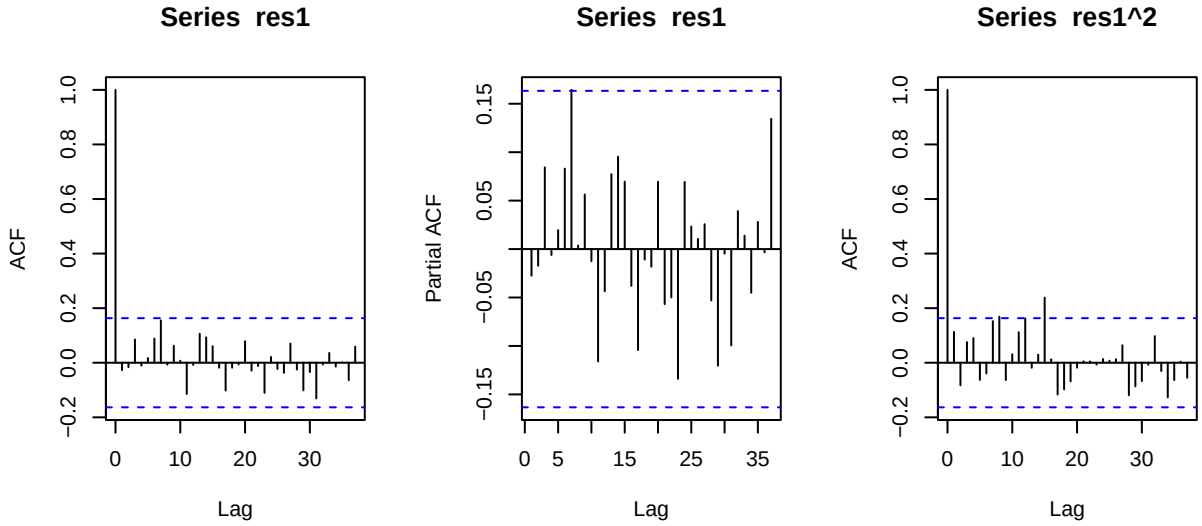
The time series plot resembled a white noise model that had a mean close to 0, constant variance, and no trend. The histogram also looked relatively normally distributed and the Q-Q plot mostly followed the Q-Q line except at the tails, where the residuals deviated from the line. This is also confirmed by the Shapiro-Wilk normality test, which was not passed with a p-value of 0.0213 (<0.05).

```
##
##  Box-Pierce test
##
## data:  res1
## X-squared = 8.3204, df = 8, p-value = 0.4028

##
##  Box-Ljung test
##
## data:  res1
## X-squared = 8.8889, df = 8, p-value = 0.3518

##
##  Box-Ljung test
##
## data:  res1^2
## X-squared = 20.7, df = 12, p-value = 0.05495
```

The Box-Pierce, Box-Ljung, and McLeod-Li tests were all passed since the p-values were greater than 0.05. This means the residuals are independent, follow the white noise model, and there is not non-linear dependence based on the squares of residuals.

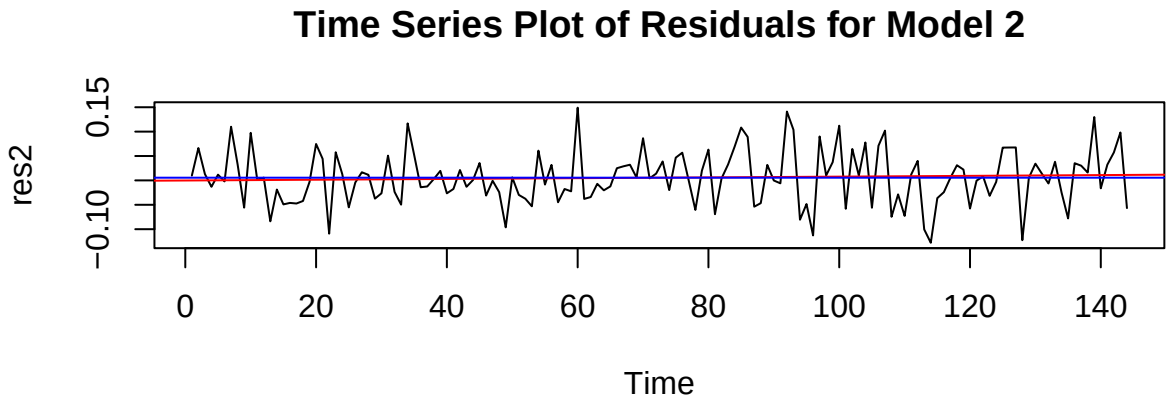


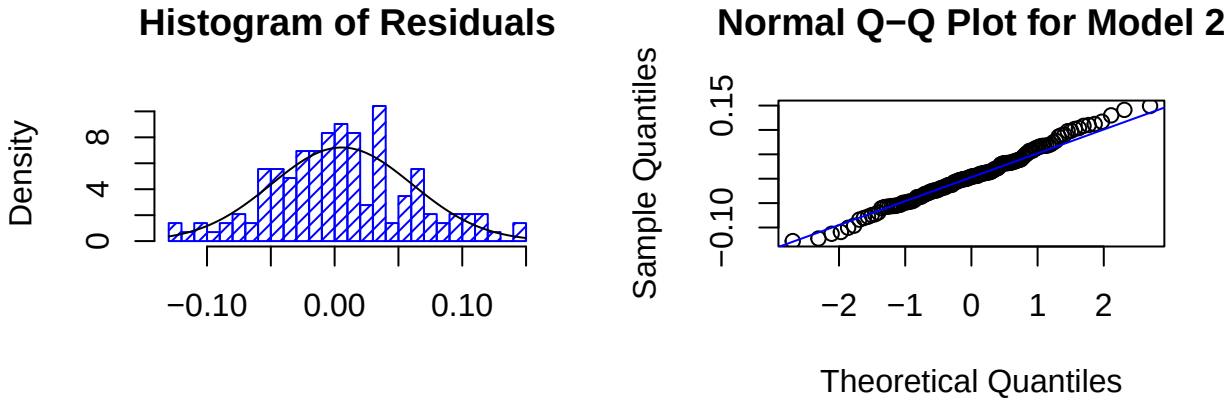

```
##
## Call:
## ar(x = res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.002764
```

Looking at the graphs of the ACF and PACF of the residuals, all the lags were insignificant which means it resembles a white noise model and the order selected was 0 as well. The ACF graph of the squared residuals had one significant lag at 15, but it can be generally ignored since it is only slightly over the 95% confidence interval and is at a relatively large lag value.

Overall, the model passes all diagnostic checks except for the normality test.

Diagnostics Checking for Model 2






```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.9926, p-value = 0.6616
```

For the residuals of the ARIMA model, the time series plot also resembled a white noise model that had a mean close to 0, constant variance, and no trend. The histogram looked normally distributed and the Q-Q plot followed the Q-Q line pretty well, which can be confirmed by the Shapiro-Wilk normality test that passed with a p-value of 0.6616 (>0.05).

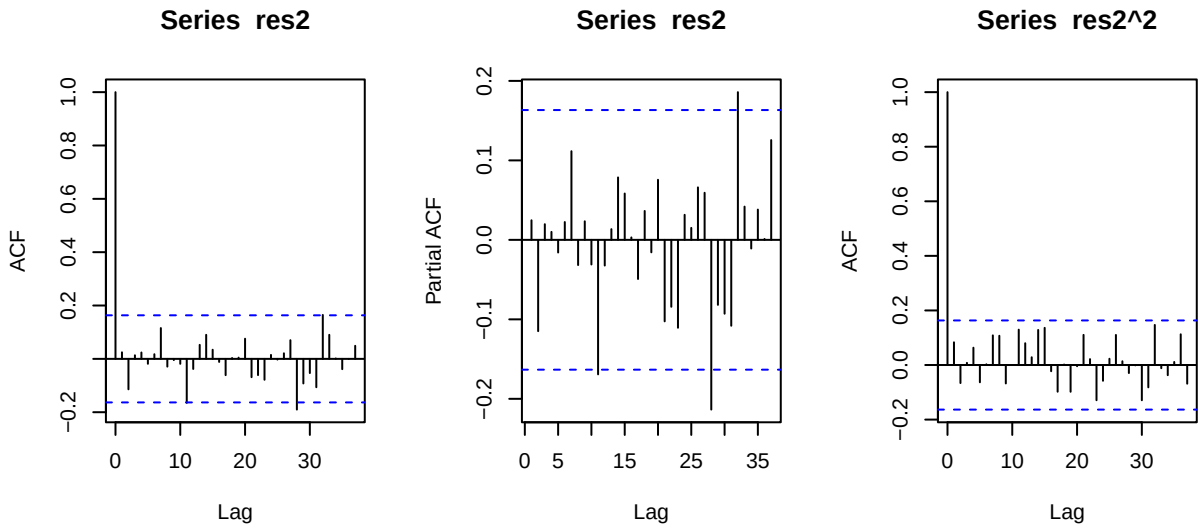
```
##
##  Box-Pierce test
##
## data:  res2
## X-squared = 8.3973, df = 10, p-value = 0.5901

##
```

```
##  Box-Ljung test
##
## data:  res2
## X-squared = 9.007, df = 10, p-value = 0.5314

##
##  Box-Ljung test
##
## data:  res2^2
## X-squared = 10.865, df = 12, p-value = 0.5405
```

The Box-Pierce, Box-Ljung, and McLeod-Li tests were all passed since the p-values were greater than 0.05. This means the residuals follow the white noise model and have uncorrelated residuals and squared residuals.




```
##
## Call:
## ar(x = res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.003056
```

Looking at the graph of the ACF of the residuals, there was only one slightly significant value at lag 28, but Bartlett's Formula allows us to ignore that as it is still very close to the 95% confidence interval. For the PACF of the residuals, there were several significant lags. However, since they are either close to the confidence interval or occur at a large lag, we can ignore them. Additionally, the `ar` function confirms a selected order of 0, so we can consider it as white noise. The ACF of the squared residuals were all within the 95% confidence interval.

Overall, the model passes all diagnostic checks.

Best Model

After performing the diagnostic checks, I decided to use my ARIMA(1,1,1) model since it passed all tests.

My final model will have the equation:

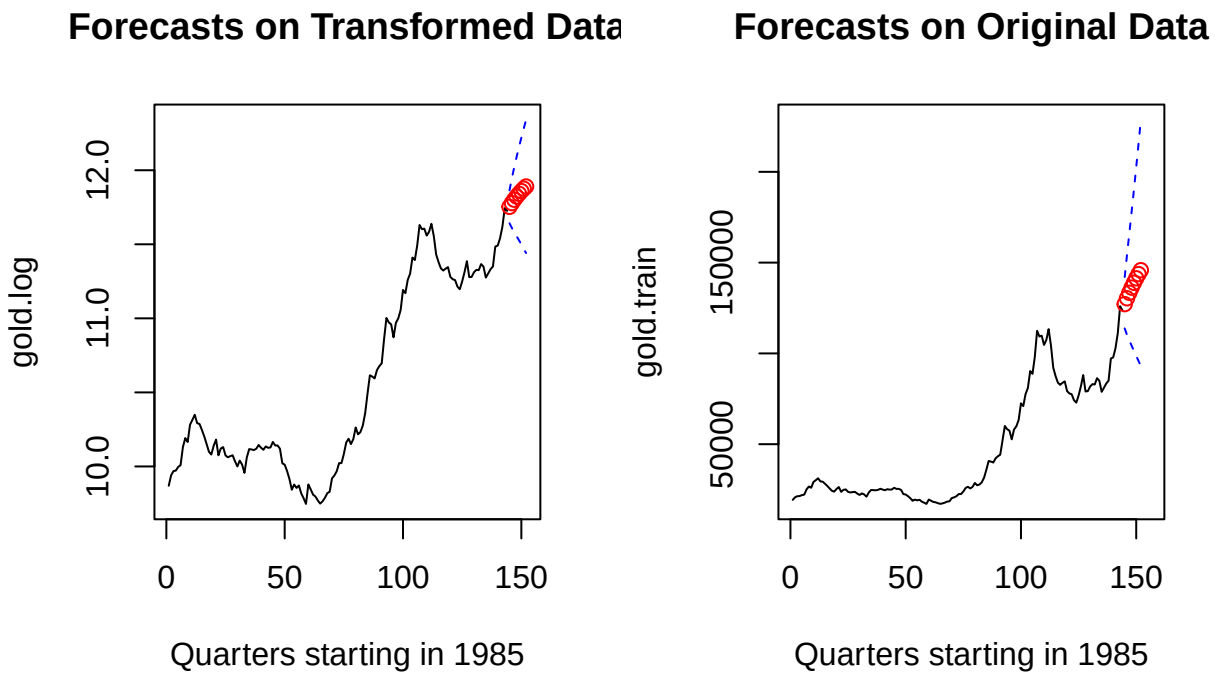
$(1 - 0.9168B)(1 - B)X_t = (1 - 0.7792B)Z_t$, Z_t~WN(0,0.003086).

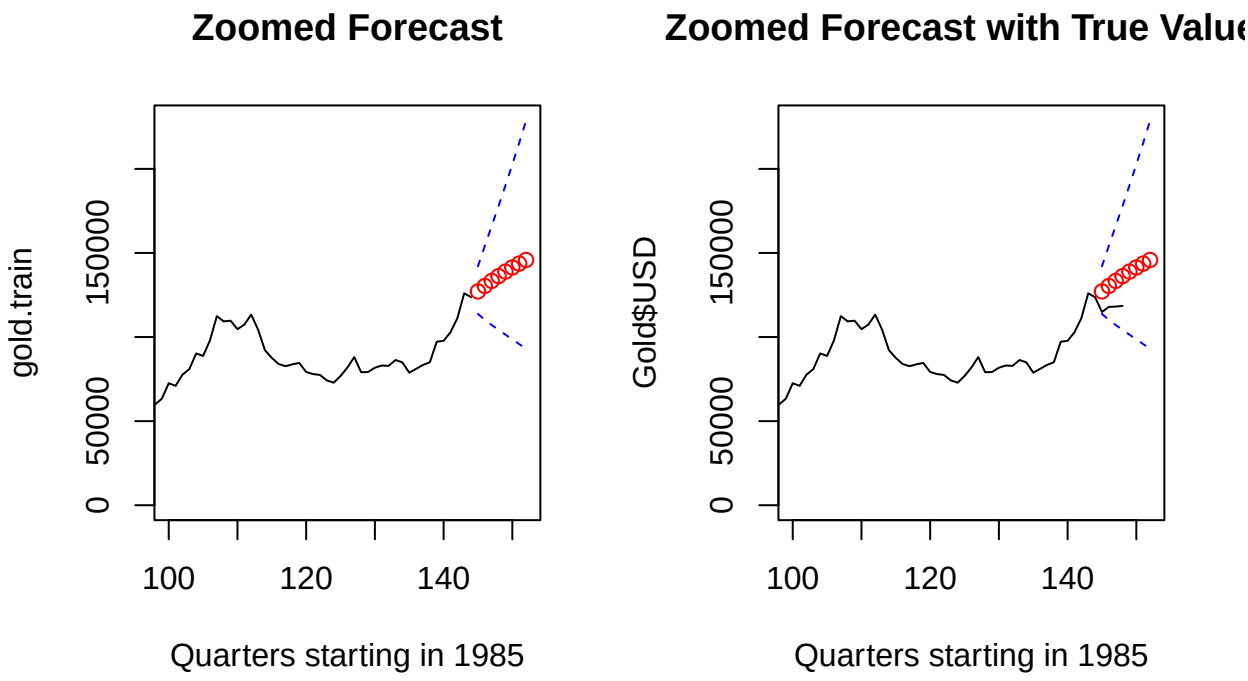
Data Forecasting

For my final step, I used the `forecast` function which prints forecasts for the next 10 observations along with the prediction bounds.

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 145       11.75305 11.68186 11.82424 11.64417 11.86193
## 146       11.77809 11.67026 11.88591 11.61318 11.94299
## 147       11.80104 11.66062 11.94147 11.58628 12.01581
## 148       11.82209 11.65074 11.99344 11.56003 12.08415
## 149       11.84138 11.64003 12.04273 11.53344 12.14932
## 150       11.85907 11.62834 12.08979 11.50621 12.21193
## 151       11.87528 11.61568 12.13488 11.47825 12.27231
## 152       11.89014 11.60210 12.17819 11.44961 12.33067
## 153       11.90377 11.58769 12.21985 11.42037 12.38718
## 154       11.91627 11.57255 12.25998 11.39060 12.44193
```

Then, I plotted 8 future predicted observations (in red) against the true values (in black) with the 95% confidence intervals (in blue).





Conclusion

Ultimately, the model I chose to use for forecasting was not the model with the lowest AICc. However, it had the second-lowest AICc and succesfully passed all diagnostic checks.

I first forecasted 2 years (2021-2022) after my training set on my transformed data. The results showed a steady increase in those quarters. Similarly, I plotted both an unzoomed and zoomed time series plot on my original data for a more helpful visualization. Lastly, I created a time series plot that showed both the entirety of the original dataset as well as the forecasted values for a more direct comparison of the true and predicted values.

My goal of creating a model that could forecast future values of gold rates was achieved. Although overall the model predicted values slightly higher than the actual gold rates, the 95% confidence interval still captured the true values. However, I would still be hesitant to use my predicted model to predict gold rates for quarters in future years given that my predictions deviate further from the true values as time goes on. My final ARIMA(1,1,1) model has the formula : $(1-0.9168B)(1-B)X_t = (1-0.7792B)Z_t$, Z_t~WN(0,0.003086).

I would like to gratefully acknowledge Professor Raya Feldman and Teaching Assistants Lihao Xiao and Thiha Aung for their assistance and contributions to the project. I would like to thank them for putting in the effort and taking the time out of their busy schedules to help me.

References

My original data set was obtained from the website: https://www.kaggle.com/datasets/hemil26/gold-rates-1985-jan-2022?select=daily_gold_rate.csv.

Appendix

```
# Load necessary libraries
library(tidyverse)
library(stats)
library(MASS)
library(ggplot2)
library(ggfortify)
library(qpcR)
library(UnitCircle)
library(tibble)
library(dplyr)
library(forecast)

## Data Importation
# Read in data
Gold <- read.csv("/Users/Rebecca/PSTAT 174 S'23/gold_rate_dataset/daily_gold_rate.csv")

# Plot time series plot of raw data
op= par(mfrow=c(1,2))
ts.plot(Gold$USD, main = "Time Series of Gold Rates", xlab = "Quarters starting in 1985",
        ylab ="Gold Rate (in USD)")
len <- length(Gold$USD)
fit = lm(Gold$USD ~ as.numeric(1:len))
abline(fit, col="red")
abline(h=mean(Gold$USD), col="blue")

# Plot time series plot of raw data with x label as years
gold = ts(Gold[,2], start = c(1985,1), frequency = 4)
ts.plot(gold, ylab = "Gold Rate (in USD)", main = "Raw Data")

## Testing/Training Sets
# Split original data into training and testing set
gold.train = gold[c(1:144)]
gold.test = gold[(c(145:148))]

# Plot time series plot of training set
op= par(mfrow=c(1,2))
plot.ts(gold.train, xlab = "Quarters starting in 1985",
        ylab ="Gold Rate (in USD)",
        main ="Time Series:Gold Rates Training")
t = 1:length(gold.train)
fit = lm(gold.train ~ t)
abline(fit, col="red")
abline(h=mean(gold.train), col="blue")

# Plot histogram of training set
hist(gold.train, label = TRUE, col = "light blue", breaks = "sturges",
  xlab = "Gold Rate (in USD)",
  main = "Histogram:Gold Rates Training")

# Plot ACF and PACF training set
op=par(mfrow=c(1,2))
```

```r
acf(gold.train, lag.max=150, main="ACF")
pacf(gold.train, lag.max=150, main="PACF")

## Visualization Techniques
# Choose lambda value of Box-Cox transformation for training set
t = 1:length(gold.train)
bcTransform = boxcox(gold.train ~ t,plotit = TRUE)
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]

# Perform Box-Cox and Log transformations
gold.bc = (1/lambda)*(gold.train^lambda-1)
gold.log = log(gold.train)

# Time series plot of Original vs. Box-Cox and Log Transformation
op= par(mfrow=c(1,3))
ts.plot(gold.train, main = "Original Times Series")
ts.plot(gold.bc, main = "Box-Cox Transform")
ts.plot(gold.log, main = "Log Transform")

# Plot histogram of Box-Cox and Log Transformation
op= par(mfrow=c(1,2))
hist(gold.bc, col="light blue", xlab="", main="Histogram: bc(U_t)")
hist(gold.log, col="light blue", xlab="", main="Histogram: ln(U_t)")

# De-trend by differencing once at lag 1
y1 = diff(gold.log, 1)
plot.ts(y1, main = "Differenced Once at Lag 1",
        ylab = expression(nabla~Y[t]))
abline(h = 0,lty = 2)

# Check if we need to difference again
y2 = diff(y1, 1)
plot.ts(y2,main = "Differenced Twice at Lag 1",
        ylab = expression(nabla~Y[t]))
abline(h = 0,lty = 2)

# Check if we need to difference at lag = 4 to remove seasonality
y4 = diff(y1, 4)
plot.ts(y4, main = "Differenced Once at Lag 1 and 4",
        ylab = expression(nabla~Y[t]))
abline(h = 0,lty = 2)

# Compare variances of each time differenced and arrange by lowest to highest
vars <- c(var(y1), var(y2), var(y4))
name <- c("Differenced Once at Lag 1",
          "Differenced Twice at Lag 1",
          "Differenced Once at Lag 1 and Lag 4")
result <- tibble(vars=vars, name=name)
result %>%
  arrange(vars)

# Plot histogram of differenced data
hist(y1, col="light blue", xlab="", main="Histogram: Differenced Once at Lag 1")
```

```r
m <- mean(y1)
std <- sqrt(var(y1))
curve(dnorm(x,m,std), add = T)

## Model Identification
# Plot ACF and PACF of transformed and differenced data
op= par(mfrow=c(1,2))
acf(y1, lag.max=37, main="ACF")
pacf(y1, lag.max=37, main="PACF")

## Model Estimation
# Fit ARIMA models
fit_arima1 = arima(gold.log, order = c(1,1,1), method="ML")
fit_arima2 = arima(gold.log, order = c(1,1,28), method = "ML")
fit_arima3 = arima(gold.log, order = c(1,1,32), method = "ML")
fit_arima4 = arima(gold.log, order = c(7,1,1), method = "ML")
fit_arima5 = arima(gold.log, order = c(7,1,28), method = "ML")
fit_arima6 = arima(gold.log, order = c(7,1,32), method = "ML")

# Fit SARIMA models
fit_sarima = arima(gold.log, order = c(1,1,1),
                   seasonal = list(order = c(8,0,0), period=4), method = "ML")
fit_sarima2 = arima(gold.log, order = c(1,1,0),
                    seasonal = list(order = c(8,0,0), period=4), method = "ML")

# Fit MA models
fit_ma1 = arima(gold.log, order = c(0,0,1), method = "ML")
fit_ma7 = arima(gold.log, order = c(0,0,7), method = "ML")

# Fit AR models
fit_ar = arima(gold.log, order = c(1,0,0), method = "ML")

# Order AICc of each model from lowest to highest
AICc <- c(AICc(fit_ma1), AICc(fit_ma7),
          AICc(fit_ar),
          AICc(fit_arima1), AICc(fit_arima2), AICc(fit_arima3),
          AICc(fit_arima4), AICc(fit_arima5), AICc(fit_arima6),
          AICc(fit_sarima), AICc(fit_sarima2)
          )
models <- c("MA(1)", "MA(7)",
            "AR(1)",
            "ARIMA(1,1,1)", "ARIMA(1,1,28)", "ARIMA(1,1,32)",
            "ARIMA(7,1,1)", "ARIMA(7,1,28)", "ARIMA(7,1,32)",
            "SARIMA(1,1,1)x(8,0,0)_4", "SARIMA(1,1,0)x(8,0,0)_4")

results <- tibble(AICc = AICc, Models = models)
results %>%
  arrange(AICc)

# Check if AICc can be lowered for two models with lowest AICc by calculating
# 95% confidence intervals for coefficients
fit_arima1
ar1_ci <- c(0.9168-1.96*0.0668, 0.9168+1.96*0.0668)
```

```
ar1_ci
ma1_ci <- c(-0.7792-1.96*0.0998, -0.7792+1.96*0.0998)
ma1_ci
# all coefficients do not include 0

fit_sarima2
# sar2, sar3, sar4, sar5, sar6, coefficient include 0 in 95% confidence interval
fit_sarima2.1 = arima(gold.log, order = c(1,1,0),
                      seasonal = list(order = c(8,0,0), period=4),
                      fixed = c(NA, NA, 0, 0, 0, 0, 0, NA, NA), method = "ML")
fit_sarima2.1
AICc(fit_sarima2.1)
# AICc went down

# Check for stationarity/invertiblity for SARIMA Model
uc.check(c(1, -0.1813, 0, 0, 0, 0, 0, 0.227, -0.2646))

# Check stationarity/invertibility for ARIMA Model
op= par(mfrow=c(1,2))
# check AR part for stationarity
uc.check(c(1, -0.9168))
# check MA part for invertibility
uc.check(c(1, -0.7792))

## Diagnostic checking of model 1
# Plot residuals
res1 <- residuals(fit_sarima2.1)
plot.ts(res1, main = "Time Series Plot of Residuals for Model 1")
fit <- lm(res1~as.numeric(1:length(res1)))
abline(fit, col="red")
abline(h=mean(res1), col = "blue")

# Plot histogram of residuals
op= par(mfrow=c(1,2))
hist(res1, density = 20, breaks = 20, col ="blue", xlab="", prob=TRUE,
     main = "Histogram of Residuals")
m <- mean(res1)
std <- sqrt(var(res1))
curve(dnorm(x, m, std), add=TRUE)

# Check normality of residuals
qqnorm(res1, main="Normal Q-Q Plot for Model 1")
qqline(res1, col="blue")
shapiro.test(res1)
# did not pass normality test

# Run diagnostic tests for residuals (Box-Pierce/Ljung-Box/McLeod-Li)
Box.test(res1, lag=12, type=c("Box-Pierce"), fitdf = 4)
Box.test(res1, lag=12, type=c("Ljung-Box"), fitdf=4)
Box.test(res1^2, lag=12, type=c("Ljung-Box"), fitdf=0)

# Plot ACF and PACF of residuals
op= par(mfrow=c(1,3))
```

```r
acf(res1, lag.max=37)
pacf(res1, lag.max = 37)
acf(res1^2, lag.max=37)

# Selected order 0 = white noise
ar(res1, aic=TRUE, order.max=NULL, method=c("yule-walker"))

## Diagnostic checking of model 2
# Plot residuals
res2 <- residuals(fit_arima1)
plot.ts(res2, main = "Time Series Plot of Residuals for Model 2")
fit <- lm(res2~as.numeric(1:length(res2)))
abline(fit, col="red")
abline(h=mean(res2), col = "blue")

# Plot histogram of residuals
op= par(mfrow=c(1,2))
hist(res2, density = 20, breaks = 20, col ="blue", xlab="", prob=TRUE,
     main = "Histogram of Residuals")
m <- mean(res2)
std <- sqrt(var(res2))
curve(dnorm(x, m, std), add=TRUE)

# Check normality of residuals
qqnorm(res2, main="Normal Q-Q Plot for Model 2")
qqline(res2, col="blue")
shapiro.test(res2)

# Run diagnostic tests for residuals (Box-Pierce/Ljung-Box)
Box.test(res2, lag=12, type=c("Box-Pierce"), fitdf = 2)
Box.test(res2, lag=12, type=c("Ljung-Box"), fitdf=2)
Box.test(res2^2, lag=12, type=c("Ljung-Box"), fitdf=0)

# Plot ACF and PACF of residuals
op= par(mfrow=c(1,3))
acf(res2, lag.max=37)
pacf(res2, lag.max = 37)
acf(res2^2, lag.max=37)

# Selected order 0 = white noise
ar(res2, aic=TRUE, order.max=NULL, method=c("yule-walker"))

## Forecasting
# prints forecasts with prediction bounds in a table
forecast(fit_arima1)

# Plot 8 steps ahead on transformed data
op= par(mfrow=c(1,2))
pred.tr <- predict(fit_arima1, n.ahead=8)
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr = pred.tr$pred - 2*pred.tr$se
ts.plot(gold.log, xlim=c(1, length(gold.log)+8), ylim = c(min(gold.log), max(U.tr)),
        xlab = "Quarters starting in 1985", main = "Forecasts on Transformed Data")
```

```
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(gold.log)+1):(length(gold.log)+8), pred.tr$pred, col="red")

# Plot 8 steps ahead on original data
pred.orig <- exp(pred.tr$pred)
U = exp(U.tr)
L = exp(L.tr)
ts.plot(gold.train, xlim = c(1, length(gold)+8), ylim=c(min(gold), max(U)),
        xlab = "Quarters starting in 1985", main = "Forecasts on Original Data")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(gold.train)+1):(length(gold.train)+8), pred.orig, col="red")

# Zoomed plot with predictions starting from entry 100
op= par(mfrow=c(1,2))
ts.plot(gold.train, xlim= c(100,length(gold.train)+8), ylim= c(250,max(U)),
        xlab = "Quarters starting in 1985", main = "Zoomed Forecast")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(gold.train)+1):(length(gold.train)+8), pred.orig, col="red")

# Zoomed plot with predictions and true values starting from entry 100
ts.plot(Gold$USD, xlim= c(100,length(gold.train)+8), ylim= c(250,max(U)), col="red",
        xlab = "Quarters starting in 1985", main = "Zoomed Forecast with True Values")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(gold.train)+1):(length(gold.train)+8), pred.orig, col="black")
```