Time Series Analysis of U.S. Population Data

Jack McColm-de Jong

2023-06-09

Abstract

This report analyzes United States population data provided by the U.S. Census Bureau, with the goal of forecasting future population growth. A variety of techniques were implemented in modeling the data, including model decomposition, analysis of ACFs and PACFs, seasonal differencing, AICc comparison, and parameter estimation. In conclusion, the U.S. population is expected to continue growing, as evidenced by the derived model. As a society we must prioritize comprehending and preparing for the consequences of a growing US population to achieve well-rounded economic progress, societal cohesion, and ecological sustainability.

Introduction

Understanding the U.S. population is a crucial statistic for several reasons. First and foremost, population statistics provide valuable insights into the demographic composition of the country. By knowing the size, growth rate, and distribution of the population, policymakers, government agencies, and businesses can make informed decisions on resource allocation, infrastructure planning, and social policies. Additionally, population statistics help identify trends and patterns in population dynamics, which is essential for addressing social and economic challenges. Moreover, population data serves as a foundation for conducting research, policy analysis, and forecasting future trends. It enables researchers and analysts to study various aspects of society, including public health, education, labor market dynamics, and voting behavior. Overall, a comprehensive understanding of the US population statistic provides a vital framework for policymakers, researchers, and organizations to make informed decisions, develop effective policies, and address the needs and demands of the nation's diverse population.

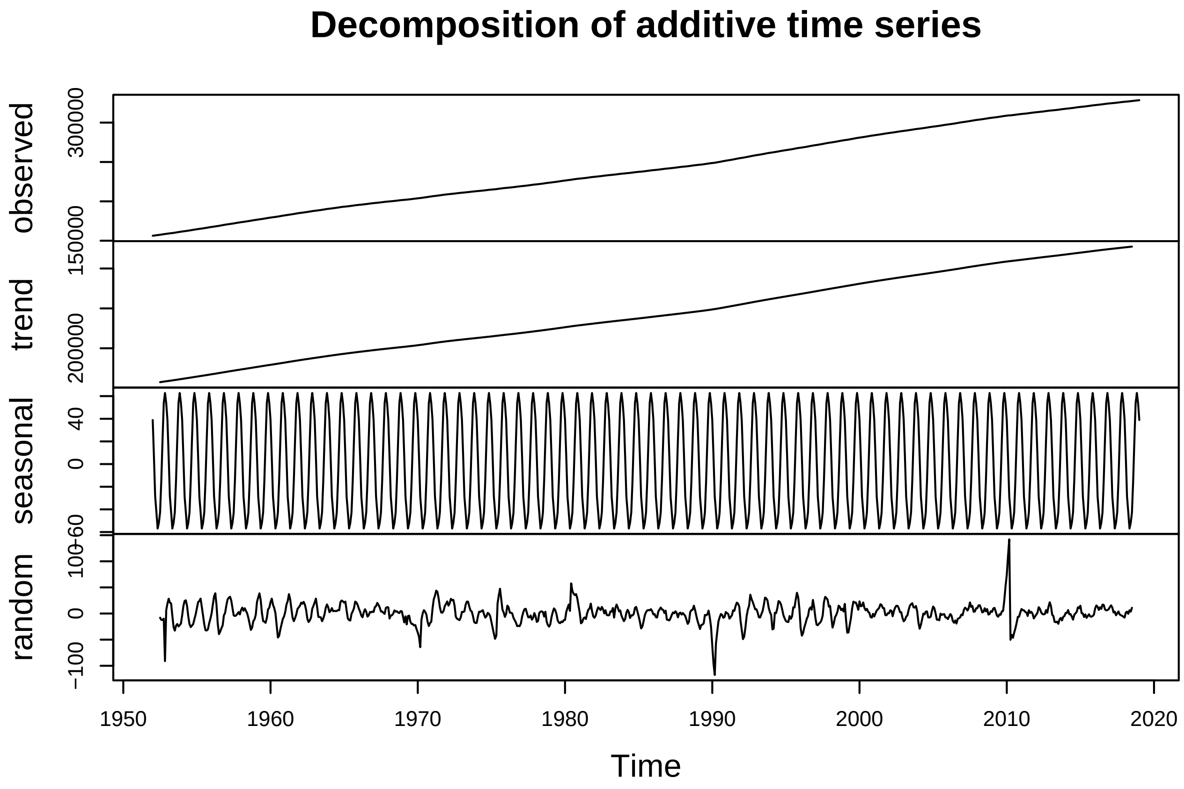
In this project, I planned to model the growth of the U.S. population using time series analysis techniques learned through PSTAT 174, including model decomposition, analysis of ACFs and PACFs, seasonal differencing, AICc comparison, and parameter estimation using R software. My intent was forecasting future population growth. The results are unsurprising: according to my model, the population is set to continue increasing for the foreseeable future. The implications of a growing US population are wide-ranging and multifaceted. Firstly, a growing population can contribute to economic growth by expanding the labor force, fostering innovation, and driving consumer demand. More people can lead to increased productivity, entrepreneurship, and a larger market for goods and services. However, population growth also puts pressure on resources and infrastructure, such as housing, transportation, and healthcare. Furthermore, a growing population impacts environmental sustainability, requiring efforts to manage natural resources, reduce carbon emissions, and preserve ecosystems. Therefore, understanding and planning for the implications of a growing US population are essential for policymakers to ensure balanced economic development, social cohesion, and environmental sustainability.

The data used for this analysis was provided by the US Census Bureau, and made publicly available on Kaggle. It contains population data between the years 1952-2019.

Analysis

Initial Analysis and Decomposion

```
setwd("/home/jovyan/") # set the working directory
pop.csv <- read.table("POP.csv", sep=",", header=FALSE, skip=1, nrows=816) # read in the data
pop <- ts(pop.csv[,2], start = c(1952,1), frequency = 12) # create the time series object
train <- ts(pop[1:805], start = c(1952,1), frequency = 12) # segment the training data
test <- pop[806:816] # set aside some test data for forecasting
plot(decompose(train)) # plot the decomposition of the model
```

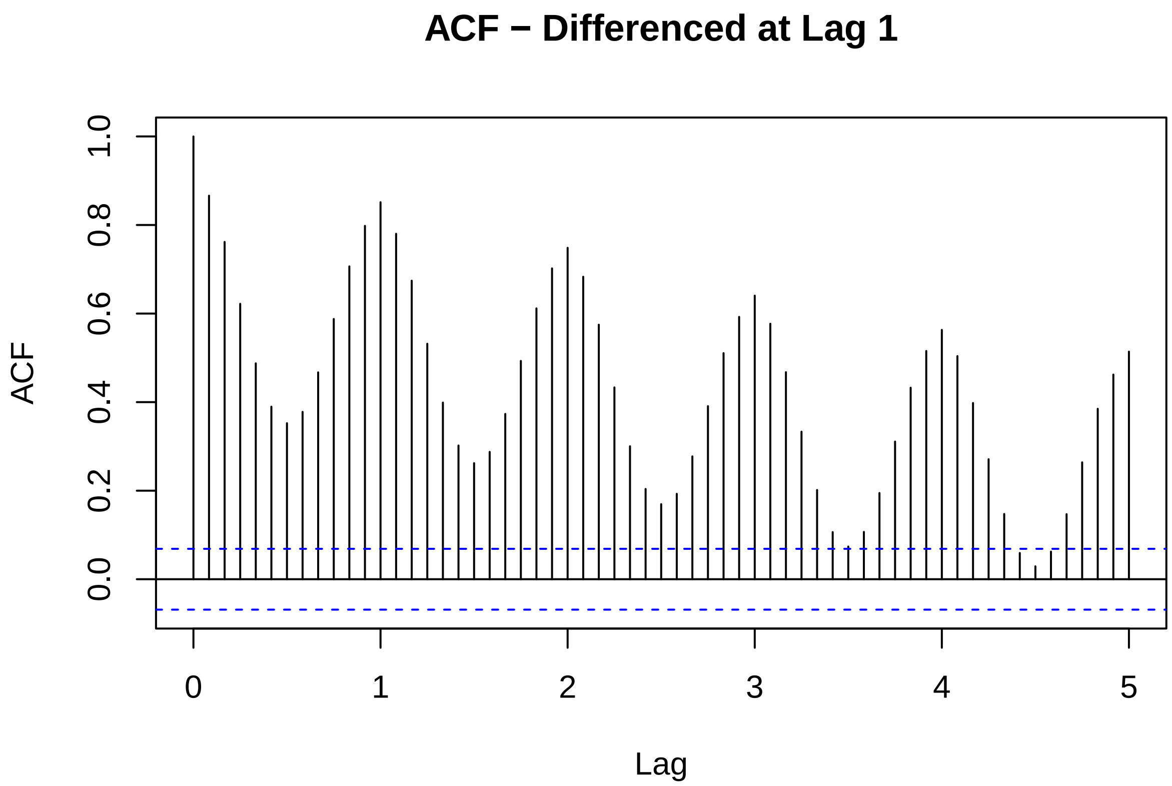


Based on the model decomposition, there is a clear positive and linear trend in the data. While the seasonal component is not immediately obvious, it is clearly shown in the seasonal aspect of the time series decomposition. There are no apparent sharp changes in behavior in the observed data, except for a couple spikes shown in the random component of the time series.

Transformations

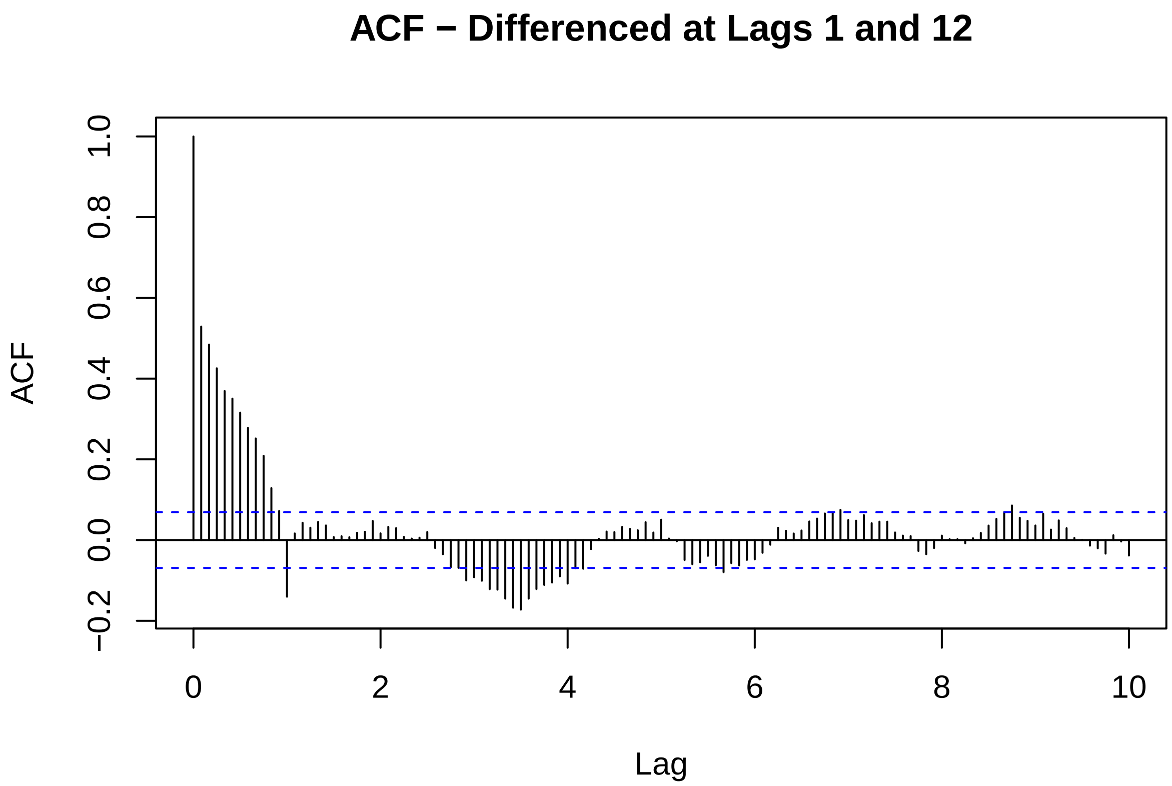
My intention is to obtain a stationary time series. I begin by applying a difference at lag 1 and analyze the resulting ACF

```
pop1 = diff(pop, 1) # difference at lag 1
acf(pop1, lag.max=60, main="ACF - Differenced at Lag 1") # plot the acf, differenced at lag 1
```



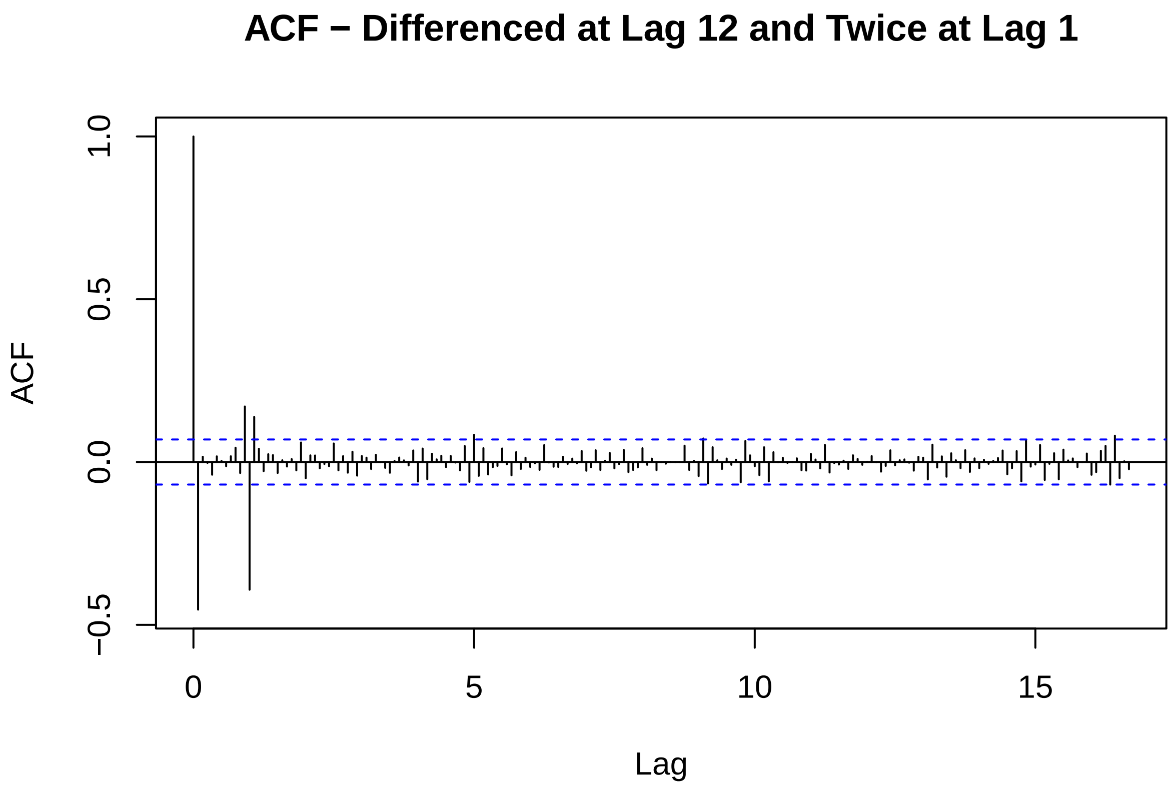
Based on the plot of my ACF, the model still needs transformation as there is a seasonal component that must be accounted for. Since the data are given in monthly lags, I difference at lag 12.

```
pop2 = diff(pop1, 12) # difference at lag 12
acf(pop2, lag.max = 120, main="ACF - Differenced at Lags 1 and 12") # plot the acf, differenced at lags
```

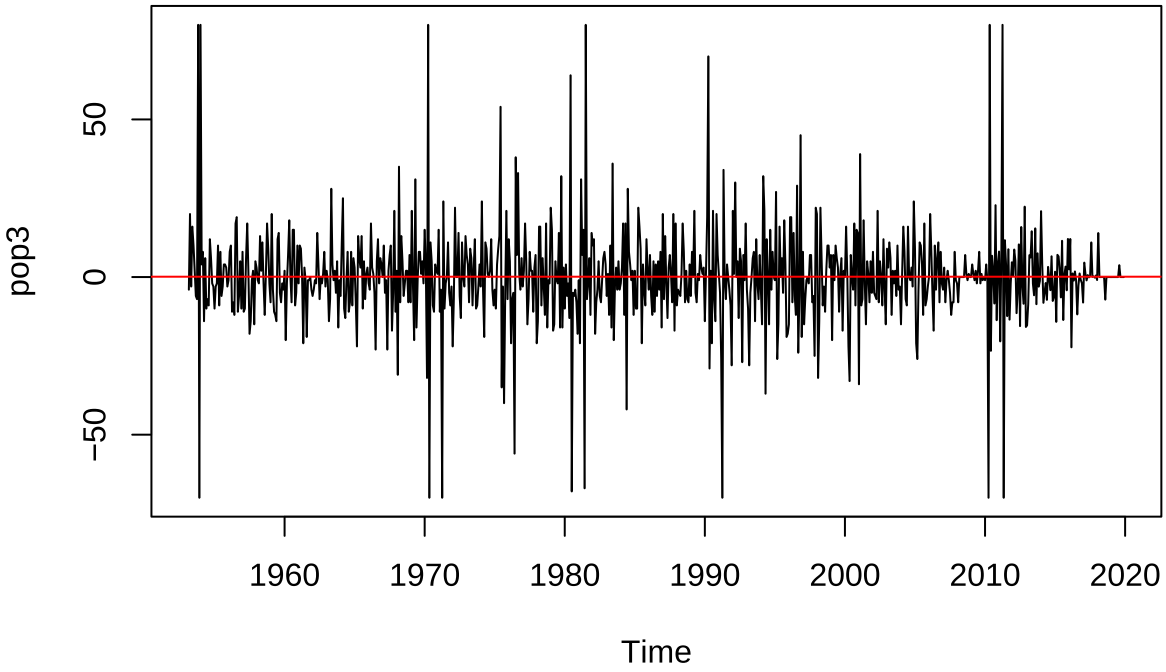


The data has now been differenced at lags 1 and 12, but the ACF does not resemble a stationary process. Therefore I difference at lag 1 once more.

```
pop3 = diff(pop2) # difference at lag 1 again
acf(pop3, lag.max = 200, main="ACF - Differenced at Lag 12 and Twice at Lag 1") # plot the acf, differe
```



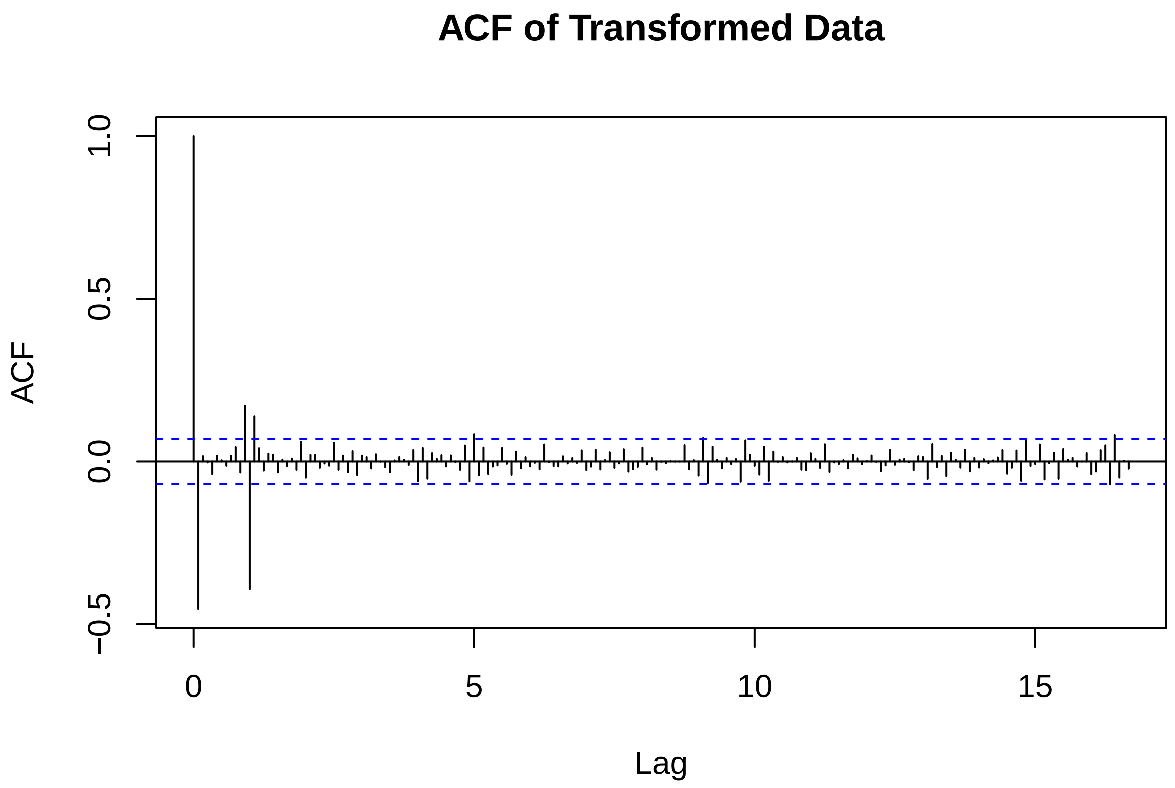
```
ts.plot(pop3) # plot the transformed time series
abline(h=mean(pop3), col="red") # add a line through the mean
```



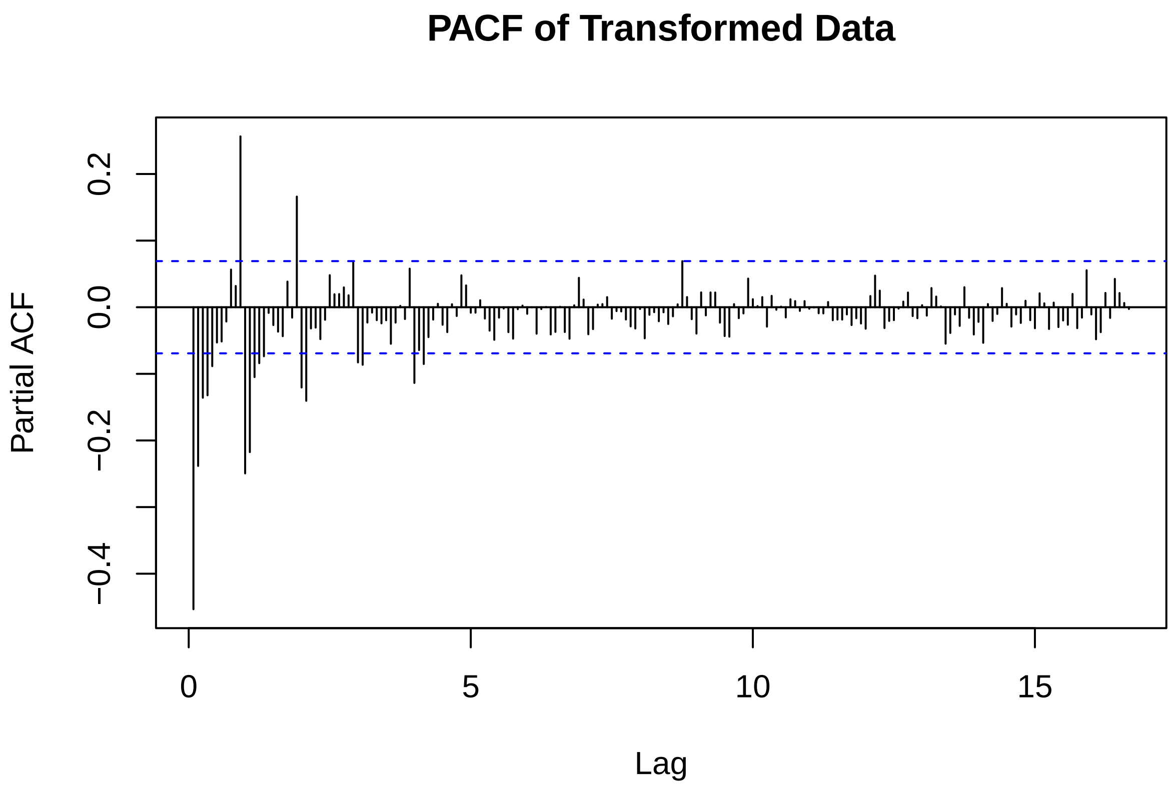
Now the ACF and time series resemble a stationary process, therefore I move on to model identification.

Analysis of ACF and PACF

```
acf(pop3, lag.max = 200, main="ACF of Transformed Data") # plot the ACF of the transformed data
```



```
pacf(pop3, lag.max = 200, main="PACF of Transformed Data") # plot the PACF of the transformed data
```



Modeling the seasonal part (P, D, Q)

I applied one seasonal difference: D = 1 at lag s = 12.
The ACF shows a strong peak at h = 1s.
A good choice for the MA part could be Q = 1.
The PACF shows two strong peaks at h = 1s and 2s, with smaller spikes at h = 3s and 4s.
A good choice for the AR part could be P = 2, 3, or 4.

Modeling the non-seasonal part (p , d, q)

I applied two differences to remove the trend: d = 2
The ACF seems to cut off at lag 1s.
A good choice for the MA part could be q = 1.
The PACF seems to tail off after lag h = 4s. There is one small spike around h = 10s
A good choice for the AR part could be p = 2, 3, 4, or 10.

Based on this information, I have come up with some models evaluate and compare their AICcs.

Possible models to try:

$SARIMA(1, 2, 1)(1, 1, 1)_{12}$

$SARIMA(1, 2, 1)(2, 1, 1)_{12}$

$SARIMA(2,2,1)(2,1,1)_{12}$

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

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

$SARIMA(4,2,1)(1,1,1)_{12}$

$SARIMA(4,2,1)(2,1,1)_{12}$

$SARIMA(10,2,1)(2,1,1)_{12}$

Model Fitting and Comparison

```
AICc(arima(pop3, order=c(1,2,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")) # Evalua
```

```
## [1] 7449.142
```

```
AICc(arima(pop3, order=c(1,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML"))
```

```
## [1] 7387.097
```

```
AICc(arima(pop3, order=c(2,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML"))
```

```
## [1] 7174.643
```

```
AICc(arima(pop3, order=c(3,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML"))
```

```
## [1] 7074.526
```

```
AICc(arima(pop3, order=c(3,2,1), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
```

```
## [1] 7134.668
```

```
AICc(arima(pop3, order=c(4,2,1), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
```

```
## [1] 7056.122
```

```
AICc(arima(pop3, order=c(4,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML")) # 2nd lo
```

```
## [1] 6994.644
```

```
AICc(arima(pop3, order=c(10,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML")) # lowes
```

```
## [1] 6813.41
```

After evaluating the models and their corresponding AICcs, I select the models with the lowest AICc values. The model with the lowest AICc is $SARIMA(10,2,1)(2,1,1)_{12}$

The algebraic form of this model is:
$(1-\phi_1 B-\phi_2 B^2-\phi_3 B^3-\phi_4 B^4-\phi_5 B^5-\phi_6 B^6-\phi_7 B^7-\phi_8 B^8-\phi_9 B^9-\phi_{10} B^{10})(1-\Phi_1 B^{12}-\Phi_2 B^{24})(1-B^{12})(1-B)^2 X_t=(1+\theta_1 B)(1+\Theta_1 B^{12}) Z_t$

Model Diagnostics for $SARIMA(10,2,1)(2,1,1)_{12}$:

```
model <- arima(pop3, order=c(10,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML") # Fi

res <- residuals(model) # save the residuals of the model for analysis
```

After fitting the model, I can access the fitted coefficients and complete my model.

Model Coefficients:

```
model$coef # extract the model coefficients
```

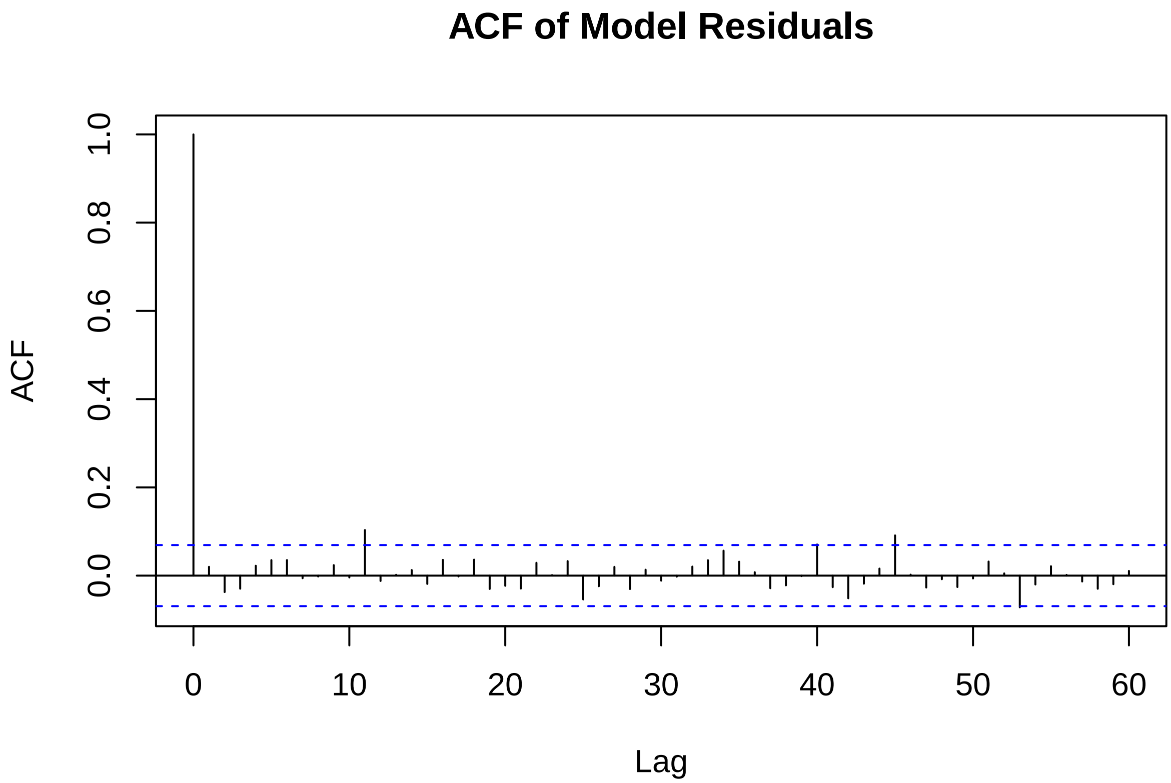
```
##        ar1        ar2        ar3        ar4        ar5        ar6        ar7
## -1.5451936 -1.7748440 -1.8085732 -1.7409019 -1.5596563 -1.3227061 -1.0528028
##        ar8        ar9       ar10        ma1       sar1       sar2       sma1
## -0.7554594 -0.4248468 -0.1903684 -0.9999899 -0.5033037 -0.2643029 -0.9998722
```

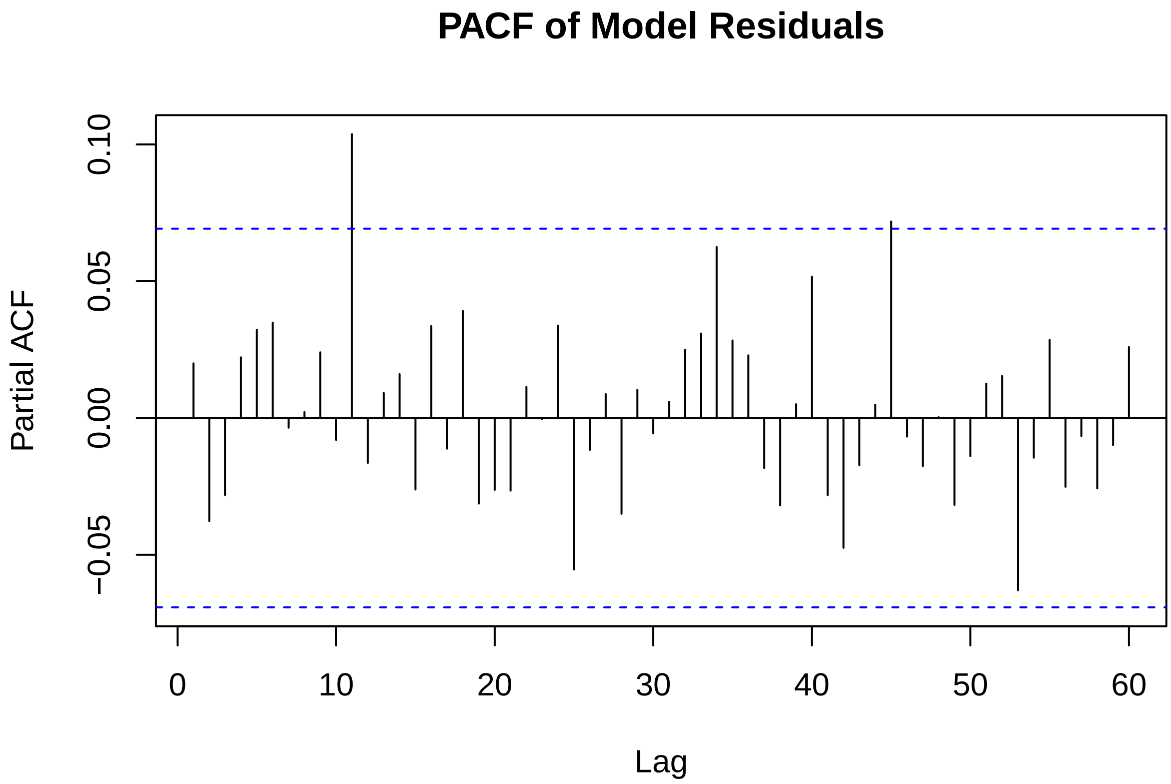
Algebraic Form of Model with Coefficients:

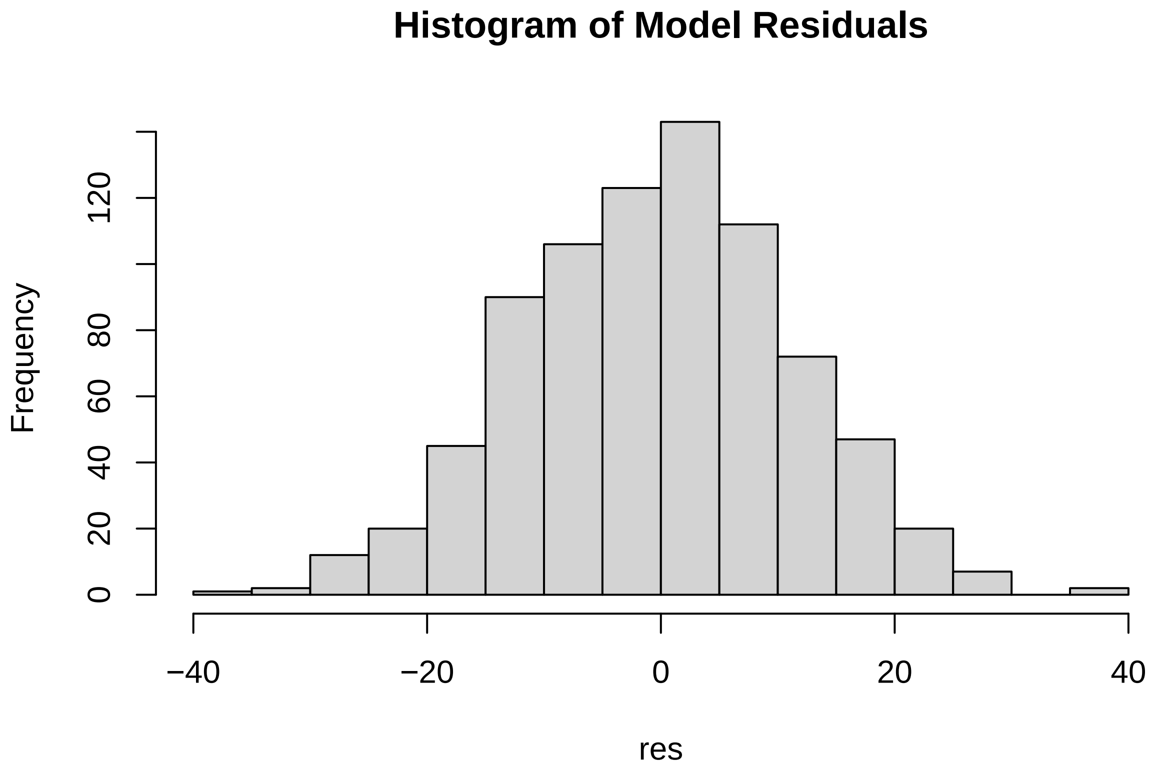
$(1+1.5451_{(0.0351)}B+1.7748_{(0.0640)}B^2+1.8085_{(0.0869)}B^3+1.7409_{(0.1030)}B^4+1.5596_{(0.1128)}B^5+1.3227_{(0.1143)}B^6+1.0528_{(0.1077)}B^7+0.7554_{(0.0938)}B^8+0.4248_{(0.0730)}B^9+0.1903_{(0.0423)}B^{10})(1+0.5033_{(0.0417)}B^{12}+0.2643_{(0.0363)}B^{24})(1+B^{12})(1+B)^2X_t = (1-0.9999_{(0.0053)}B)(1-0.9998_{(0.0155)}B^{12})Z_t$, $\sigma_Z^2 = 282.4$

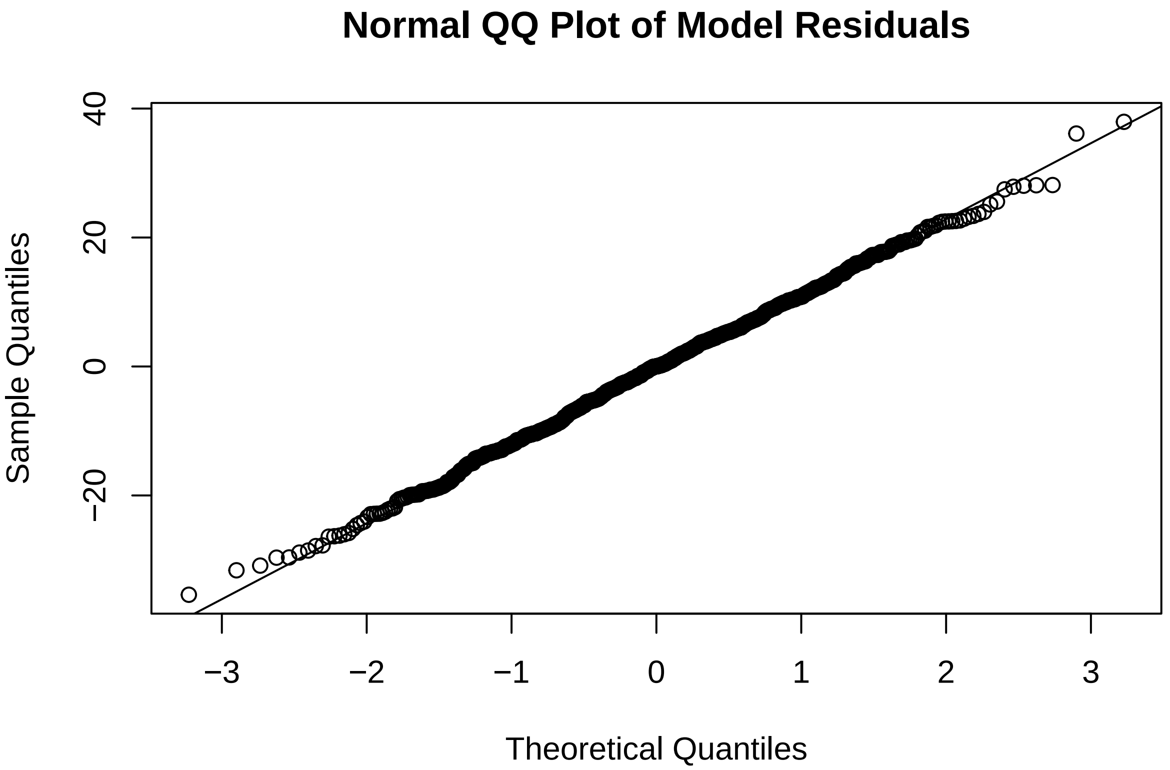
Now I can evaluate the model by running diagnostic tests on the residuals.

Analysis of Residuals:






Histogram of Model Residuals
Frequency
0
20
40
60
80
120
−40
−20
0
20
40
res



```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99867, p-value = 0.8227
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 13.702, df = 12, p-value = 0.3201
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 13.884, df = 12, p-value = 0.3082
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 6.8403, df = 12, p-value = 0.868
```

This model passes the diagnostic tests, and the residuals appear to be normally distributed. However, I will compare it to the model with the 2nd lowest AICc to ensure I have the best model.

Model Diagnostics for $SARIMA(4,2,1)(2,1,1)_{12}$:

The algebraic form of this model is:
$(1-\phi_1 B-\phi_2 B^2-\phi_3 B^3-\phi_4 B^4)(1-\Phi_1 B^{12}-\Phi_2 B^{24})(1-B^{12})(1-B)^2 X_t=(1+\theta_1 B)(1+\Theta_1 B^{12})Z_t$

```
model2 <- arima(pop3, order=c(4,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML") # fi

res2 <- residuals(model2) # save the residuals of the model for analysis
```

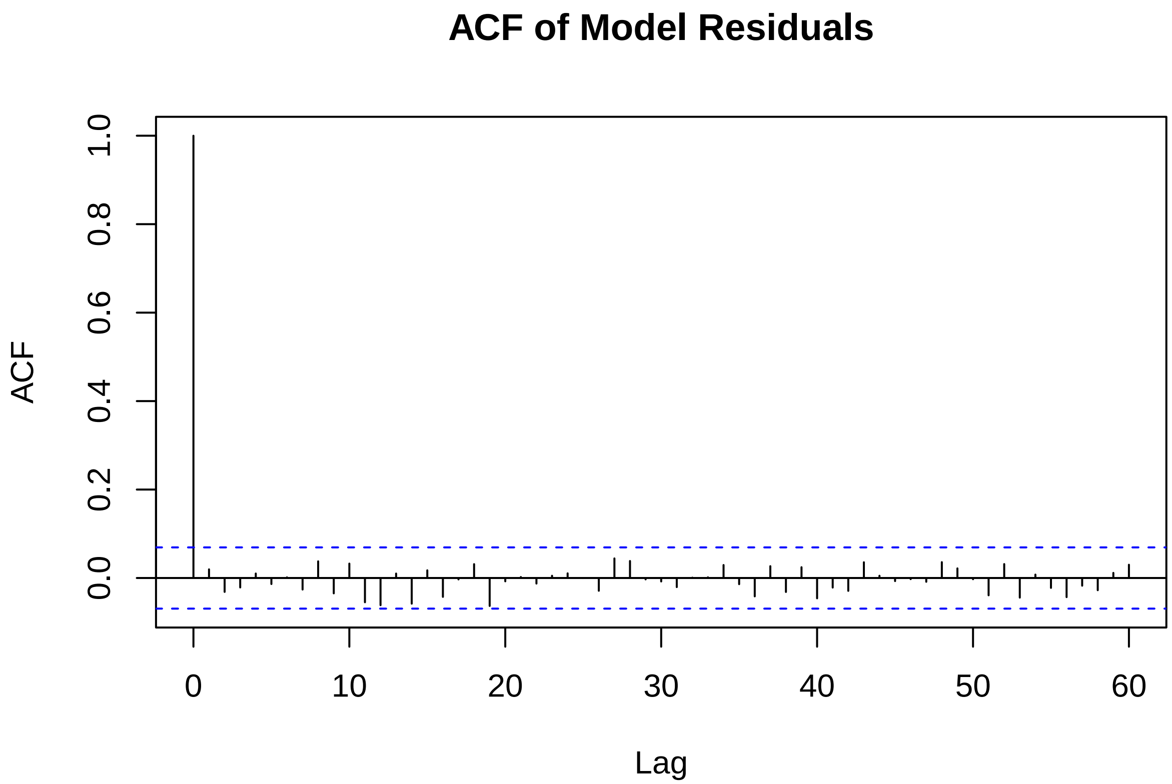
Model Coefficients:

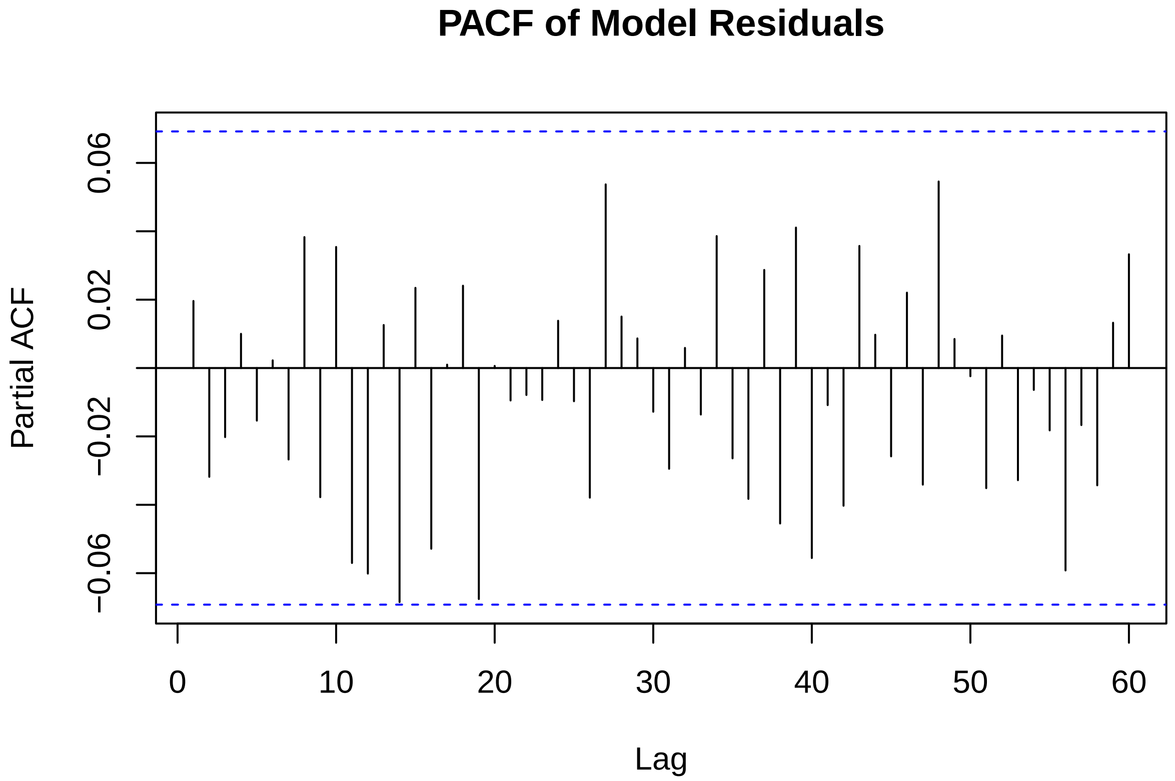
```
model$coef # extract the model coefficients
```

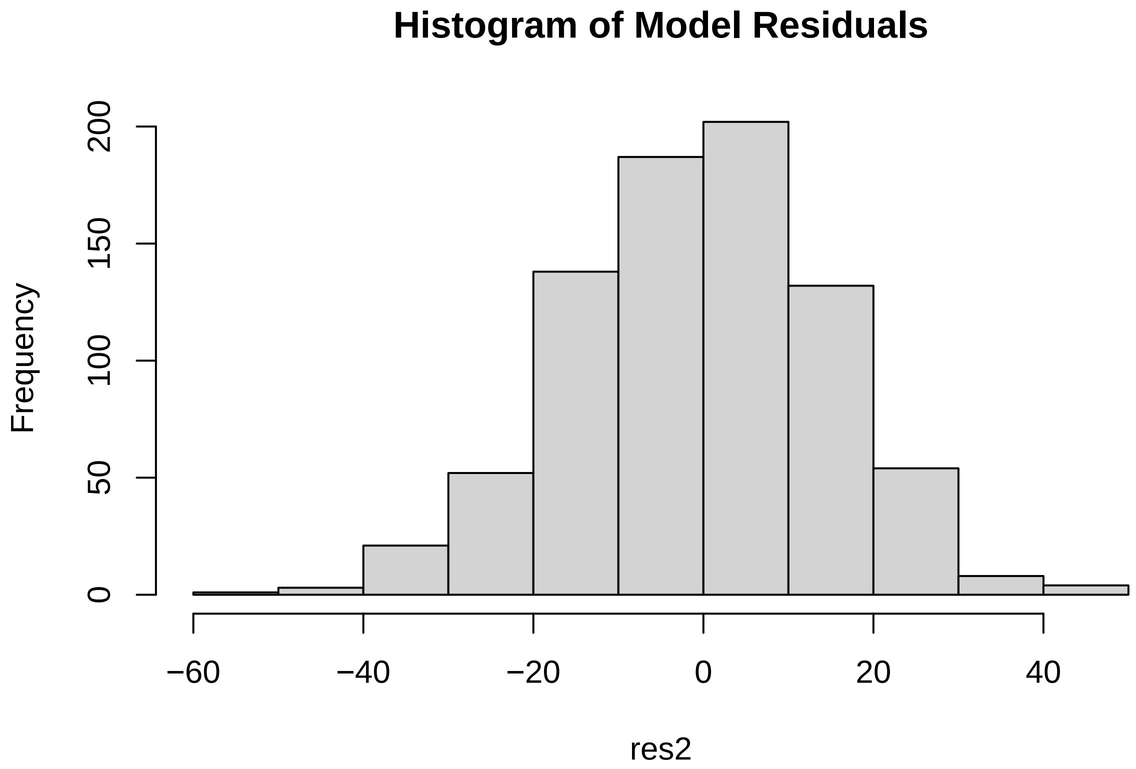
```
##        ar1        ar2        ar3        ar4        ar5        ar6        ar7 
## -1.5451936 -1.7748440 -1.8085732 -1.7409019 -1.5596563 -1.3227061 -1.0528028 
##        ar8        ar9       ar10        ma1       sar1       sar2       sma1 
## -0.7554594 -0.4248468 -0.1903684 -0.9999899 -0.5033037 -0.2643029 -0.9998722
```

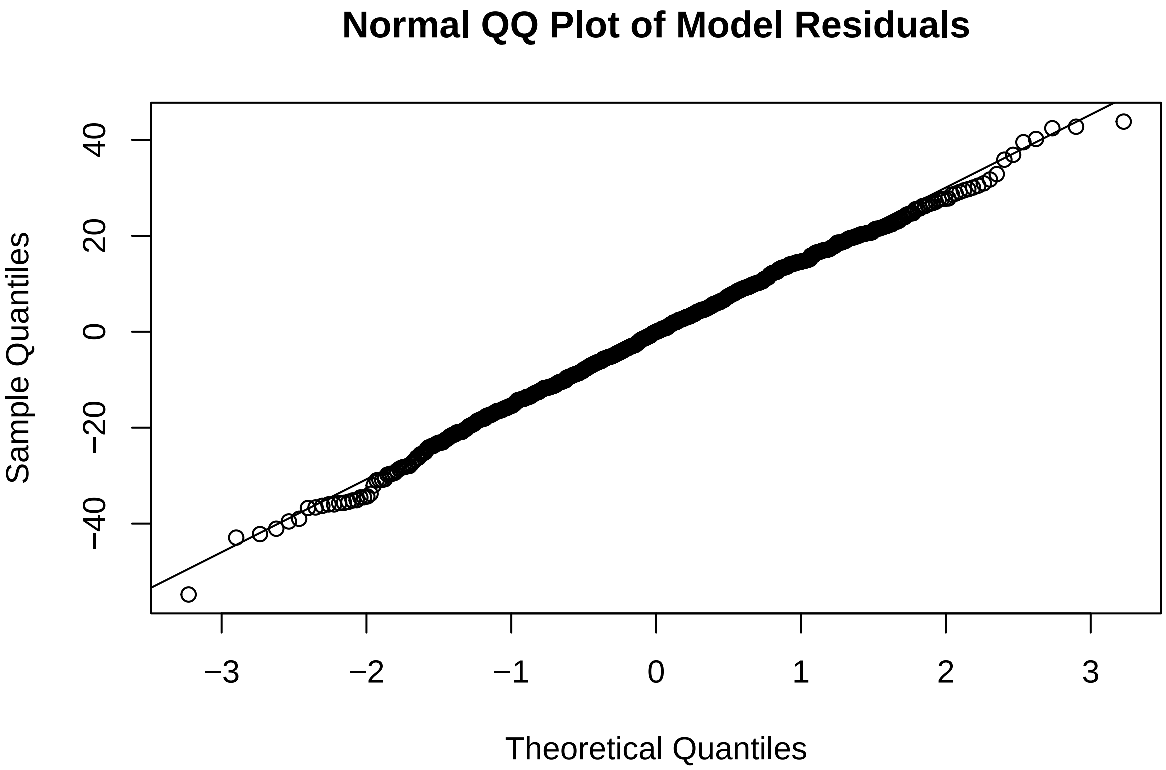
Algebraic Form of Model with Coefficients:

$(1+1.2766_{(0.0339)}B+1.1025_{(0.0509)}B^2+0.7178_{(0.0509)}B^3+0.3153_{(0.0339)}B^4)(1+0.6068_{(0.0361)}B^{12}+0.2922_{(0.0356)}B^{24})(1+B^{12})(1+B)^2X_t=(1-0.9999_{(0.0051)}B)(1-0.9998_{(0.0149)}B^{12})Z_t$, $\sigma_Z^2=363.4$




PACF of Model Residuals
Partial ACF
0.06
0.02
−0.02
−0.06
0
10
20
30
40
50
60
Lag


Histogram of Model Residuals
Frequency
0
50
100
150
200
−60
−40
−20
0
20
40
res2



```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.99772, p-value = 0.3446
```

```
##
##  Box-Pierce test
##
## data:  res2
## X-squared = 10.669, df = 12, p-value = 0.5575
```

```
##
##  Box-Ljung test
##
## data:  res2
## X-squared = 10.818, df = 12, p-value = 0.5445
```

```
##
##  Box-Ljung test
##
## data:  res2^2
## X-squared = 16.172, df = 12, p-value = 0.1835
```

Based on the testing of the 2nd model, I can conclude that the residuals pass all the necessary tests. However, I will choose the first model because it has the lowest AICc and passes the diagnostic tests. The first was also suggested as a potential model by my initial analysis of the ACF and PACF. Now I move on to forecasting with the model.

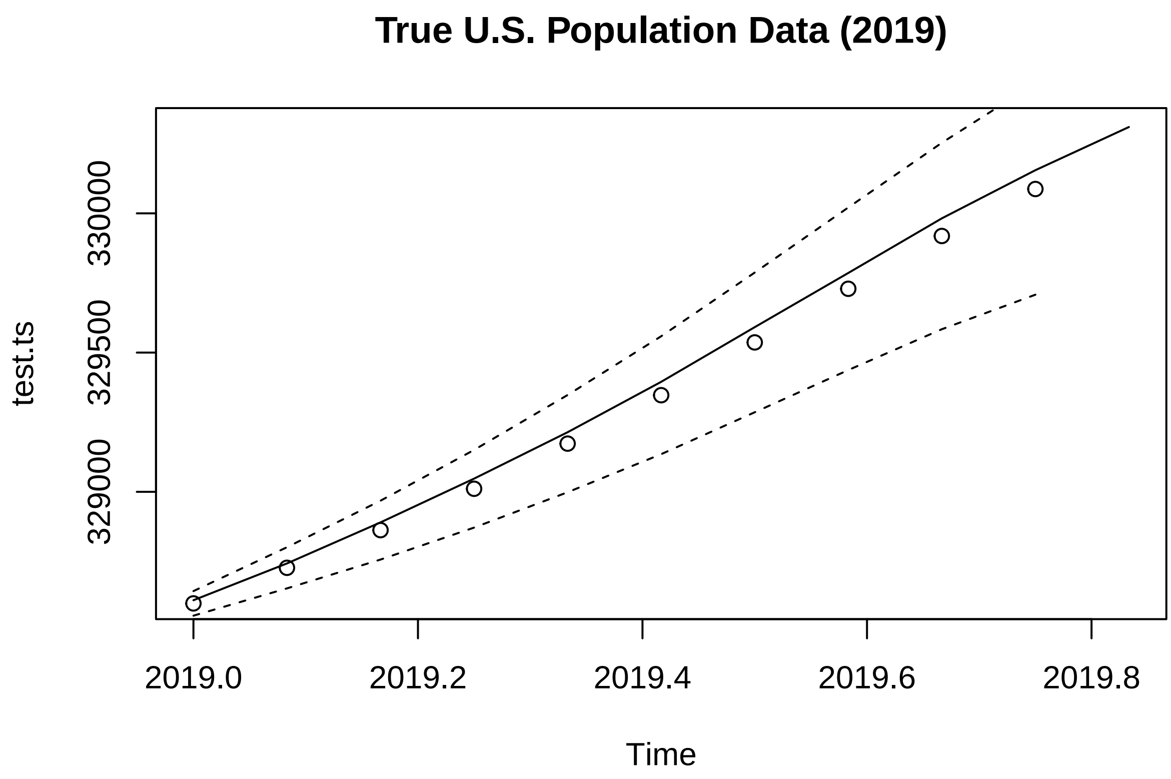
Forecasting

In order to forecast the data, I will use the sarima.for() function to generate 10 values from the training set and compare them to the test values, which I extracted from the data earlier. I also add a 95% confidence interval to confirm that my forecasts are accurate and within the interval.

```
forecast <- sarima.for(train, n.ahead=10, 10,2,1,P=2,D=1,Q=1,S=12) # forecast 10 values ahead using my
```

```
test.ts <- ts(test, start = 2019, frequency = 12) # new ts object to make sure x vals (dates) are corre
test.vec <- ts(as.vector(forecast$pred), start = 2019, frequency = 12) # ts object of predicted values
test.se <- as.vector(forecast$se) # extract the se from forecasts for confidence interval

ts.plot(test.ts, main="True U.S. Population Data (2019)") # plot the true values
points(test.vec) # plot the predicted values
lines(test.vec+1.96*test.se,lty=2) # add in confidence interval lines
lines(test.vec-1.96*test.se,lty=2)
```



The forecasted data are within the confidence interval and very close to the true values given by the original data. Therefore I can conclude that my model is accurate.

Conclusion

After analyzing and modeling the U.S. population data between 1952 and 2019, I have concluded that the population is expected to continue increasing for the foreseeable future. My model, given by $(1+1.5451_{(0.0351)}B+1.7748_{(0.0640)}B^2+1.8085_{(0.0869)}B^3+1.7409_{(0.1030)}B^4+1.5596_{(0.1128)}B^5+1.3227_{(0.1143)}B^6+1.0528_{(0.1077)}B^7+0.7554_{(0.0938)}B^8+0.4248_{(0.0730)}B^9+0.1903_{(0.0423)}B^{10})(1+0.5033_{(0.0417)}B^{12}+0.2643_{(0.0363)}B^{24})(1+B^{12})(1+B)^2X_t = (1-0.9999_{(0.0053)}B)(1-0.9998_{(0.0155)}B^{12})Z_t$, $\sigma_Z^2 = 282.4$, was able to accurately forecast the population in 2019. Therefore my initial goals have been achieved. The U.S. population is an incredibly important statistic to model and understand. Firstly, population modeling helps policymakers, governments, and organizations make informed decisions related to resource allocation, infrastructure planning, and social programs. Additionally, population models can aid in predicting future demands for essential services like healthcare, education, housing, and transportation. Moreover, population modeling plays a vital role in studying and projecting societal trends, identifying potential challenges, and assessing the impact of various factors, such as economic conditions or policy changes, on the population. By accurately modeling the total U.S. population, policymakers and researchers can develop more effective strategies and policies to address current and future needs, promote sustainable development, and improve the overall well-being of the nation.

References

US CENSUS BUREAU. December 2019. "Population Time Series Data", Retrieved 06/05/2023 from https://www.kaggle.com/datasets/census/population-time-series-data?resource=download&select=POP.csv

Appendix

```
setwd("/home/jovyan/") # set the working directory
pop.csv <- read.table("POP.csv", sep=",", header=FALSE, skip=1, nrows=816) # read in the data
pop <- ts(pop.csv[,2], start = c(1952,1), frequency = 12) # create the time series object
train <- ts(pop[1:805], start = c(1952,1), frequency = 12) # segment the training data
test <- pop[806:816] # set aside some test data for forecasting
plot(decompose(train)) # plot the decomposition of the model
```

```
pop1 = diff(pop, 1) # difference at lag 1
acf(pop1, lag.max=60, main="ACF - Differenced at Lag 1") # plot the acf, differenced at lag 1
```

```
pop2 = diff(pop1, 12) # difference at lag 12
acf(pop2, lag.max = 120, main="ACF - Differenced at Lags 1 and 12") # plot the acf, differenced at lags
```

```
pop3 = diff(pop2) # difference at lag 1 again
acf(pop3, lag.max = 200, main="ACF - Differenced at Lag 12 and Twice at Lag 1") # plot the acf, differe
```

```
ts.plot(pop3) # plot the transformed time series
abline(h=mean(pop3), col="red") # add a line through the mean
```

```
acf(pop3, lag.max = 200, main="ACF of Transformed Data") # plot the ACF of the transformed data
pacf(pop3, lag.max = 200, main="PACF of Transformed Data") # plot the PACF of the transformed data
```

```
AICc(arima(pop3, order=c(1,2,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")) # Evalua
AICc(arima(pop3, order=c(1,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML"))
AICc(arima(pop3, order=c(2,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML"))
AICc(arima(pop3, order=c(3,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML"))
AICc(arima(pop3, order=c(3,2,1), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
AICc(arima(pop3, order=c(4,2,1), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
AICc(arima(pop3, order=c(4,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML")) # 2nd lo
AICc(arima(pop3, order=c(10,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML")) # lowes
```

```
model <- arima(pop3, order=c(10,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML") # Fi

res <- residuals(model) # save the residuals of the model for analysis
```

```
model$coef # extract the model coefficients
```

```
acf(res, lag.max = 60, main="ACF of Model Residuals") # plot the ACF of the residuals
pacf(res, lag.max = 60, main="PACF of Model Residuals") # plot the PACF of the residuals
hist(res, main="Histogram of Model Residuals") # plot a histogram of the residuals
```

```r
qqnorm(res, main="Normal QQ Plot of Model Residuals") # normal qq plot of residuals to analyze normality
qqline(res) # add a line through the qq plot
shapiro.test(res) # test residuals for normality
Box.test(res, lag = 12, type = c("Box-Pierce")) # test for autocorrelation in the residuals
Box.test(res, lag = 12, type = c("Ljung-Box")) # test for independence in residuals
Box.test(res^2, lag = 12, type = c("Ljung-Box")) # test for autocorrelation in the squared residuals
```

```r
model2 <- arima(pop3, order=c(4,2,1), seasonal = list(order = c(2,1,1), period = 12), method="ML") # fi

res2 <- residuals(model2) # save the residuals of the model for analysis
```

```r
model$coef # extract the model coefficients
```

```r
acf(res2, lag.max = 60, main="ACF of Model Residuals") # plot the ACF of the residuals
pacf(res2, lag.max = 60, main="PACF of Model Residuals") # plot the PACF of the residuals
hist(res2, main="Histogram of Model Residuals") # plot a histogram of the residuals
qqnorm(res2, main="Normal QQ Plot of Model Residuals") # normal qq plot of residuals to analyze normali
qqline(res2) # add a line through the qq plot
shapiro.test(res2) # test residuals for normality
Box.test(res2, lag = 12, type = c("Box-Pierce")) # test for autocorrelation in the residuals
Box.test(res2, lag = 12, type = c("Ljung-Box")) # test for independence in residuals
Box.test(res2^2, lag = 12, type = c("Ljung-Box")) # test for autocorrelation in the squared residuals
```

```r
forecast <- sarima.for(train, n.ahead=10, 10,2,1,P=2,D=1,Q=1,S=12) # forecast 10 values ahead using my
```

```r
test.ts <- ts(test, start = 2019, frequency = 12) # new ts object to make sure x vals (dates) are corre
test.vec <- ts(as.vector(forecast$pred), start = 2019, frequency = 12) # ts object of predicted values
test.se <- as.vector(forecast$se) # extract the se from forecasts for confidence interval

ts.plot(test.ts, main="True U.S. Population Data (2019)") # plot the true values
points(test.vec) # plot the predicted values
lines(test.vec+1.96*test.se,lty=2) # add in confidence interval lines
lines(test.vec-1.96*test.se,lty=2)
```