

History of Military Spending

Joseph Chang

March 11th, 2022

Executive Summary

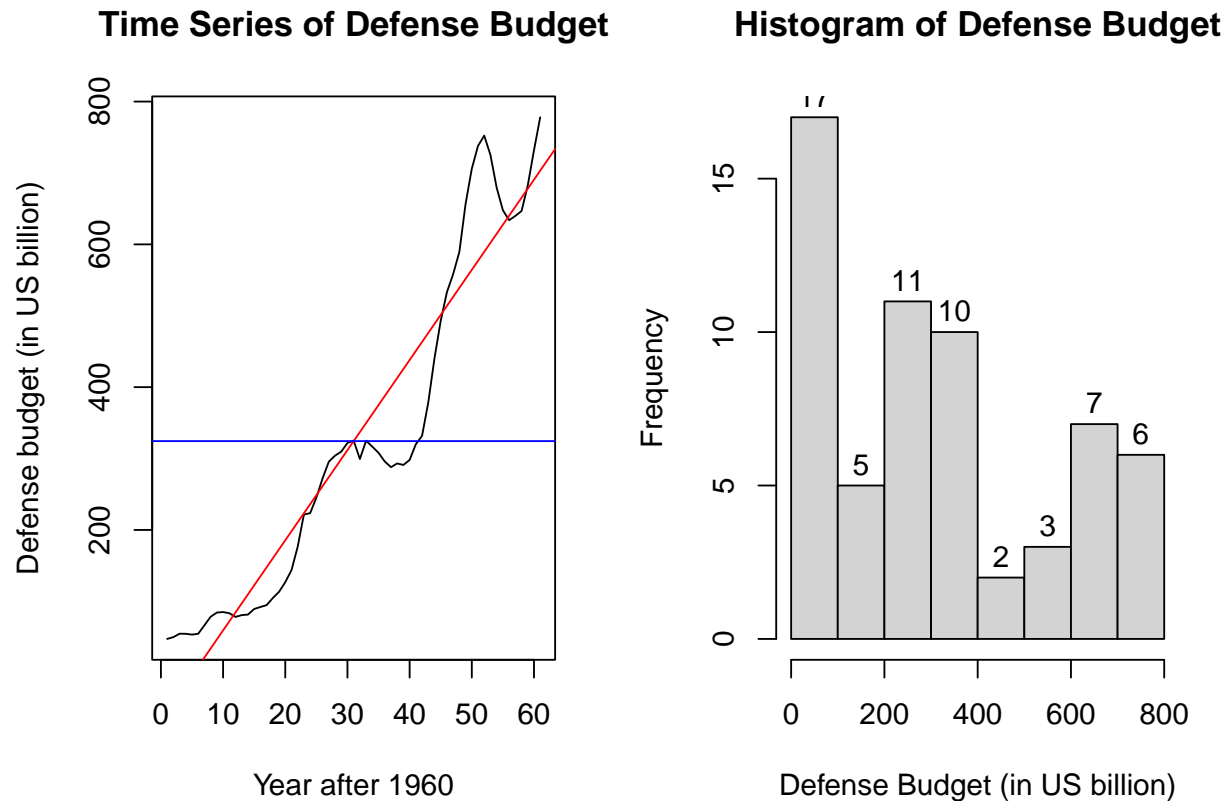
This project focuses on the United States' defense budget from the years 1960 ~ 2021. The goal of this project is to create a time series model that could predict the last 10 years (2012-2021) of the U.S. Defense Budget and compare it with last 10 years (2012-2021) given by the actual data. In order to do this, I began by splitting the original dataset into testing and training sets, with training as the first 51 points and testing as the last 10 points. After training, I made any necessary transformations using Boxcox and discovered that I needed to transform the training data by taking its log. Next, when I differenced and found the smallest variance, I needed to difference the log of training at lag 1 once. From there, I could plot the ACF and PACF of my training data and estimate the parameters of a potential model using a for loop. After model estimation, I considered the two best models from the lowest AICc and began model diagnostics. In this step, I tested if the residuals in my candidate models were independent, had any nonlinear dependence, and were normal. If passed, the model could be considered for forecasting. In the model forecasting, I predicted the 10 future observations from my best model and concluded that my model closely followed the actual model for about the first seven points (years), but then began to diverge from the actual model the rest of the way. As a final step, I predicted 5 extra years from my predicted model onto the original dataset just for my curiosity. From this project, I learned that the U.S government spends a billions of dollars every year on its defense budget.

Introduction

My data shows the United States military defense budget since 1960. It has a yearly frequency and measures budget spending in terms of billions of USD. Using the website Kaggle, I imported a csv file into RStudio and began to analyze the time series data. Although the original dataset has columns dedicated to population and GDP of US for each respective year, I find the defense budget most interesting because it matches closely with the history of the United States and the wars it has participated in. For example, after the fall of the USSR (1992) and ending the Cold War, US defense spending dipped, and stayed constant. But after the September 11th attack (2001), total spending increased exponentially as the government prioritized protecting its own citizens. Both of these historical events are corroborated by the data. Thus, the goal of this project is to create a prediction model that could predict as close to possible of future US defense spending. To solve this problem, I split training and testing sets from the original model, with testing as the last 10 points. Using the training set to predict parameters in my model, I could correctly determine the best model that would represent the training dataset. In the forecasting step, I plotted a time series on the training set and predicted 15 steps ahead: 10 for comparison with testing (marked in red) and 5 just for my pleasure (marked in blue). Along with these points were 95 percent confidence intervals that showed the margin of error for the predicted points. Both the predicted points and confidence intervals are in blue to signify prediction. The results indicated that the confidence intervals were very wide, meaning the margin of error could be large. While the first 7 predicted points were closely related to the first 7 actual points, the rest of the predicted model showed a very different trend compared to the testing set. I can conclude that if my model were to continue predicting beyond the testing set, it would probably not be very accurate.

Data Importation

First, I read in the data and plotted the time series model for the Defense Budget for each of the years, along with a red line for trend and blue line for mean. Additionally, I created a histogram to visualize the entire Defense Budget dataset.

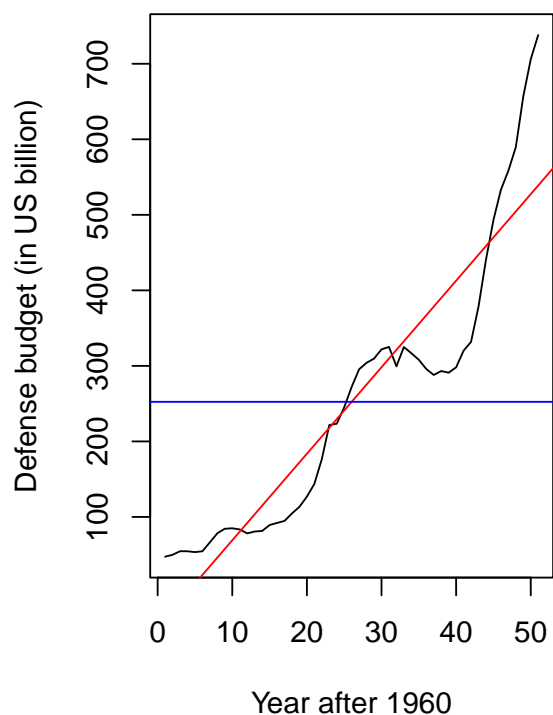


Immediate observations: There seems to be a linear trend that is positive but there is no seasonality and no apparent sharp change in behavior. There is non-constant variance and mean, where mean is 324.4 billion USD. The histogram seems to be skewed right.

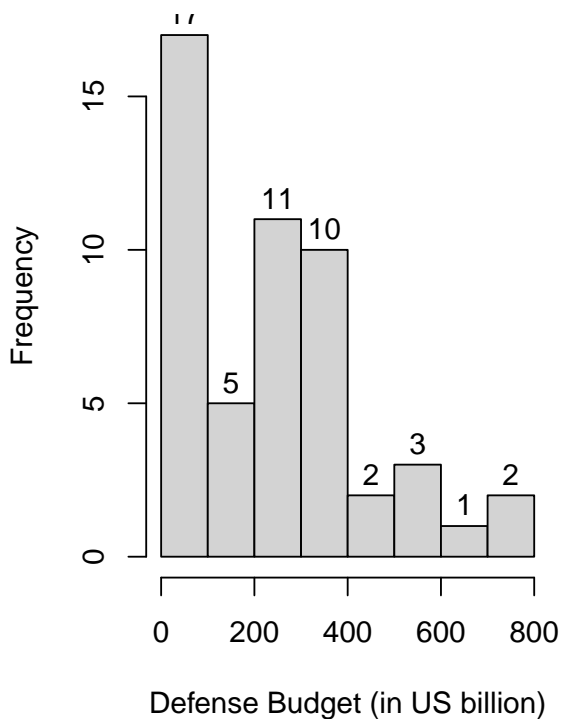
Testing and Training Sets

Next, I created training and testing sets from the original data. For better visualization, I plotted the time series plot for the training set, along with trend in red and mean in blue. In addition, I plotted a histogram of the training set.

Time series of training set



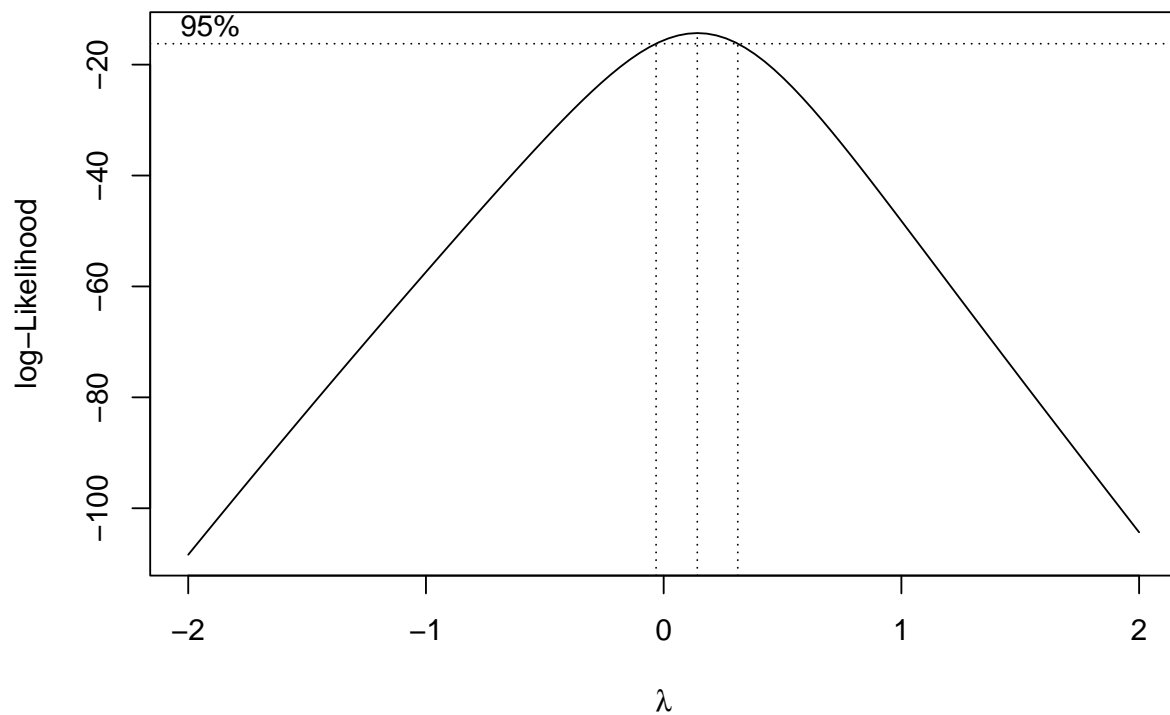
Histogram of training set



Based from the original data and training data, both look about the same and non-stationary. The training data appears to have a linear positive trend with no seasonality or sharpe changes. There is still non-constant variance and mean, where mean is now 252.4 billion USD. The histogram still looks skewed right.

Transformation

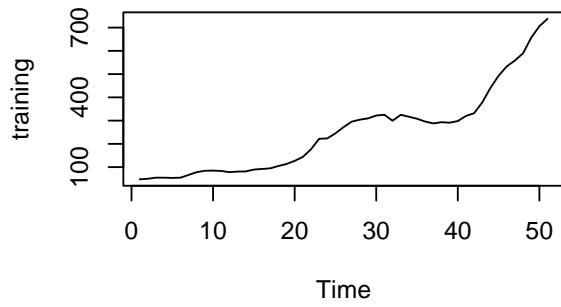
Next, since the original data looked skewed, I will see if I need to make any necessary transformations to make the model stationary and to stabilize the variance.



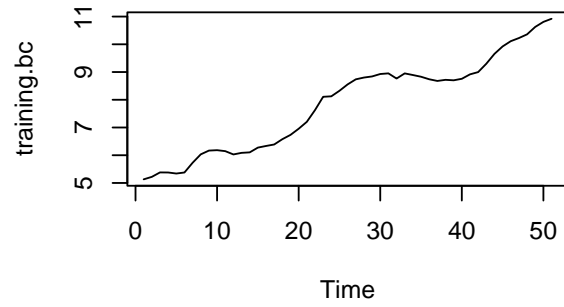
Using the Box-Cox transformation, the BcTransform command gave me the value of lambda to be around 0.1414. Thus, I named the box-cox transformation data as training.bc. One thing I notice is that 0 is inside the confidence interval, so a log transformation is another possibility. As such, I will define it as training.log.

In order to determine which model to use, I will create time series plots for training, box-cox transformation, and log transformation. I will create histograms of each as well.

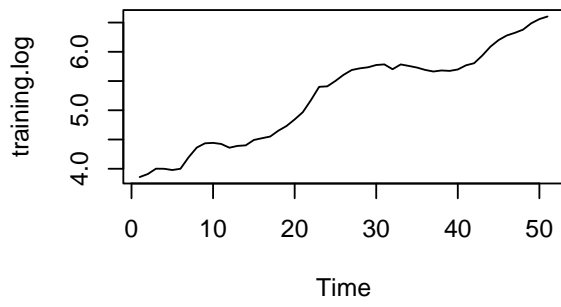
Original Training set



Box-cox Transform



Log Transform





From the time series plot, both the box-cox transform and the log transform look to have a more stable variance and trend than the original. Based from the histogram, the original is still looking skewed right while the box-cox transformation looks more symmetric. Log transformation looks even more symmetric than the box-cox. As a result, the log transformation looks the most appropriate to use based on my judgement.

Differencing

Then, I want to check if training.log needed any differencing to remove trend. To begin, I differenced once the log of training set at lag 1, when $d=2$, and $d=3$. All differencing took place at lag 1. Additionally, I checked the variances of each time I differenced as well.

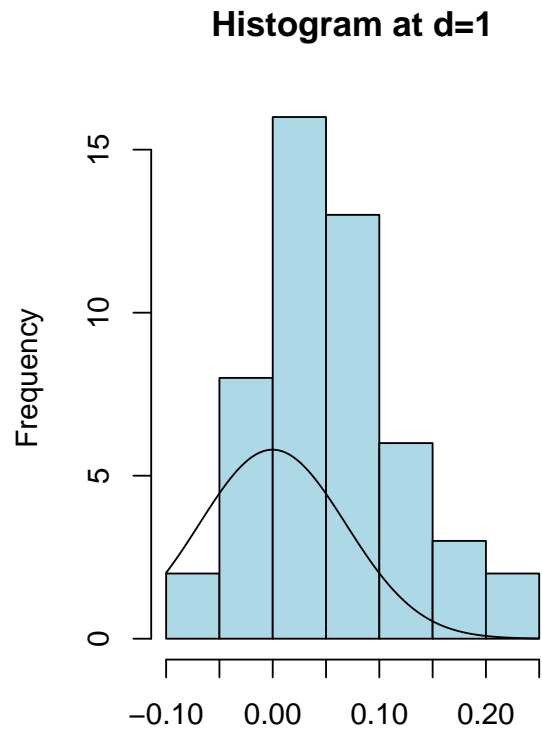
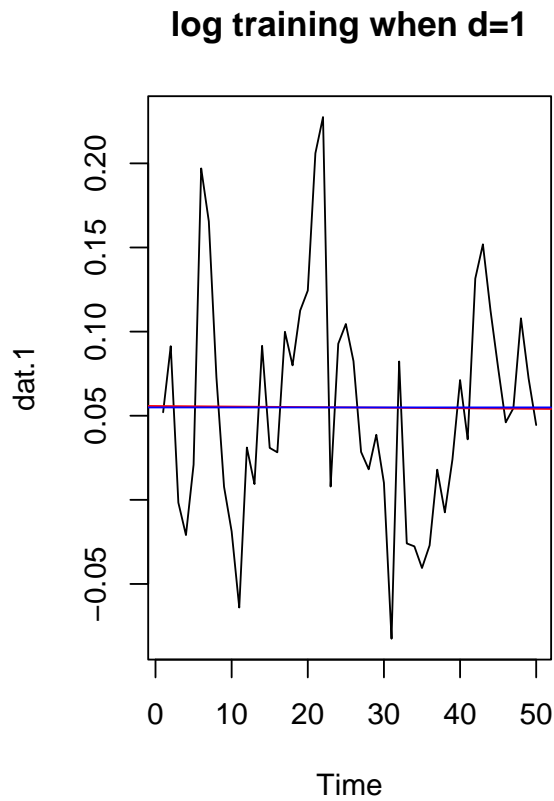
```
## [1] 0.00457
```

```
## [1] 0.004732
```

```
## [1] 0.01196
```

Based from the variances, I only need to difference when $d=1$ because variance is lowest when $d=1$. At $d=2$, variance increased again. Therefore, $d=1$ is most appropriate.

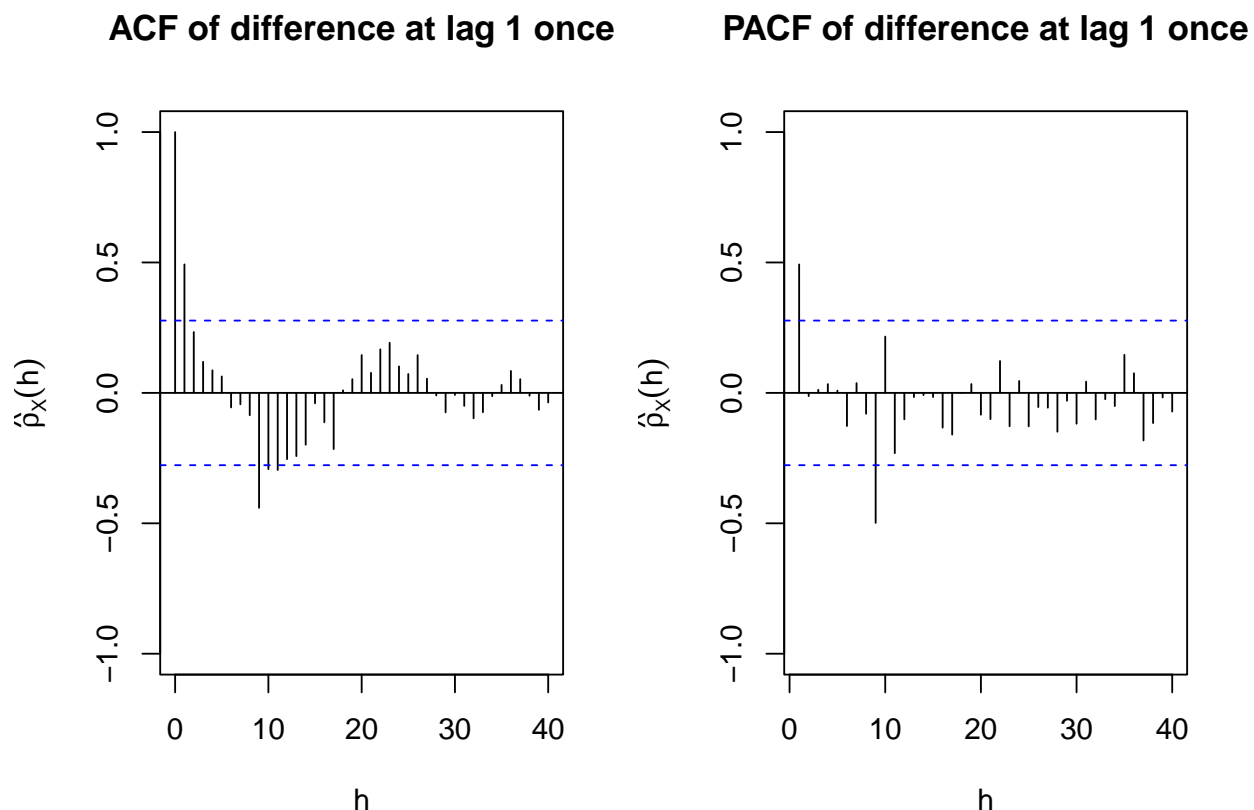
Therefore, I plotted the time series for dat.1 along with its mean colored in blue as well as its histogram.



The time series plot for dat.1 looks somewhat stationary and the histogram looks somewhat gaussian, but overall maybe skewed left.

Model Identification

For comparison, I plotted ACF and PACF for dat.1. I set the maximum lag at 40 because since there is no seasonality in training, I won't need too much data.



From the ACF, I noticed that lags 1,9,10,11 were outside of the confidence interval, with lag 9 being the furthest outside and as the peak. For PACF, both lags 1 and 9 outside confidence interval, with 9 again being the furthest outside and the peak.

Model Estimation

Here, I will estimate the parameters on my potential model. I used a for loop function to find the five models with the lowest AICc to run diagnostics on. Based from the ACF of dat.1, lags 1, 9, 10, 11 were outside the confidence interval and for PACF lags 1 and 9 were outside. Therefore, I suspect there might be some seasonality and the potential use of the SARIMA model, with period at 9 since both ACF and PACF had peak at 9.

Both the ACF and PACF showed peak and outside confidence interval at lag 9 but not at lag 18. Thus, in my for loop, I put values of both P and Q in SARIMA to be either 0 or 1. In addition, p and q will also take on the same values as P and Q (either 0 or 1) because both ACF and PACF also showed lag 1 to be outside confidence interval.

```
##      p q P Q   AICc
## 6    1 0 1 0 -143.0
## 14   1 0 1 1 -140.9
## 8     1 1 1 0 -140.8
## 10    1 0 0 1 -138.9
## 16    1 1 1 1 -138.8
```


Based from the lowest AICc, I will consider the two best models for diagnostics, along with their AICc. Model 1 will be SARIMA (1,1,0),(1,0,0), period at 9. Model 2 will be SARIMA (1,1,0),(1,0,1), period at 9.

```
##
## Call:
## arima(x = training.log, order = c(1, 1, 0), seasonal = list(order = c(1, 0,
##      0), period = 9), method = "ML")
##
## Coefficients:
##      ar1      sar1
##      0.830   -0.566
## s.e.  0.079    0.122
##
## sigma^2 estimated as 0.00271:  log likelihood = 74.61,  aic = -143.2

## [1] -143

##
## Call:
## arima(x = training.log, order = c(1, 1, 0), seasonal = list(order = c(1, 0,
##      1), period = 9), method = "ML")
##
## Coefficients:
##      ar1      sar1      sma1
##      0.844   -0.502   -0.111
## s.e.  0.082    0.191    0.227
##
## sigma^2 estimated as 0.00268:  log likelihood = 74.73,  aic = -141.4

## [1] -140.9
```

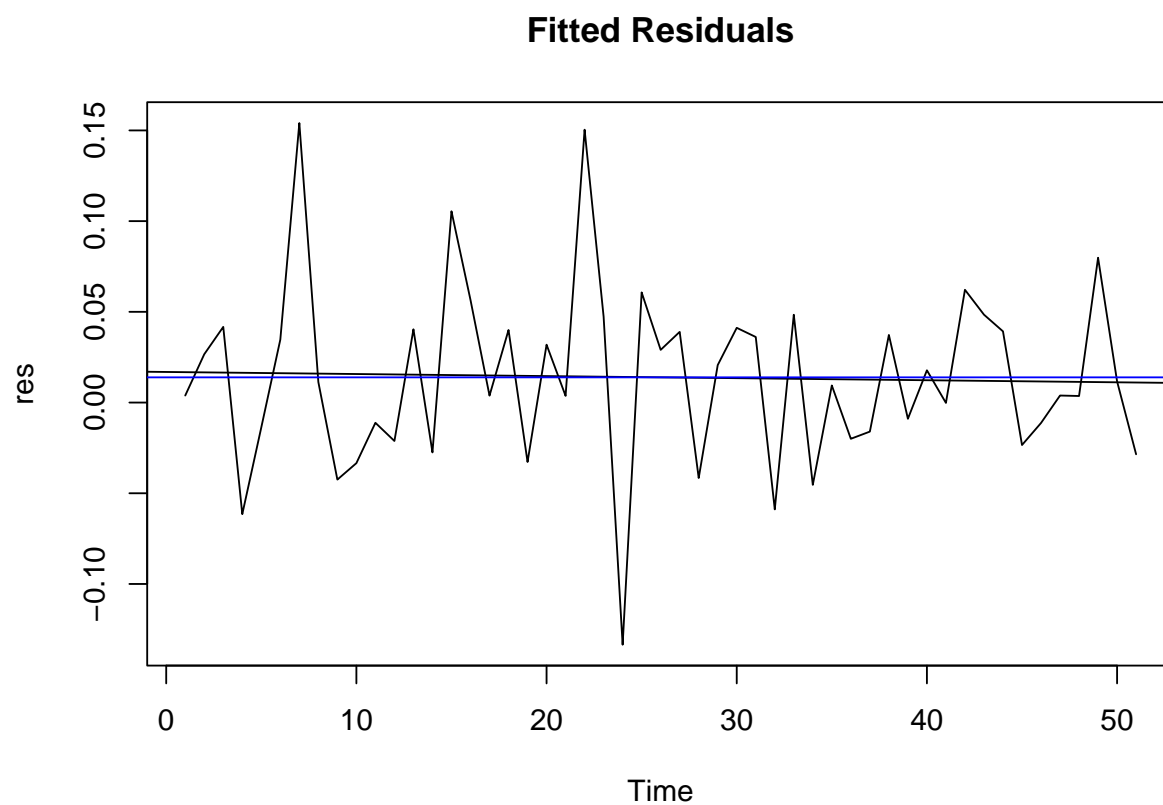
Both Model 1 and 2 are stationary because their phi values are less than the absolute value of 1. Both are invertible because all AR models are invertible. Model 1 has a lower AICc than model 2.

Model Diagnostics for Model 1

This will be model diagnostics for model 1: SARIMA (1,1,0),(1,0,0), period at 9. I found mean then variance of the residuals of model 1. Since the training data has 51 observations, the box.test will have sqrt of 51 or around 7.

```
## [1] 0.0139

## [1] 0.002512
```

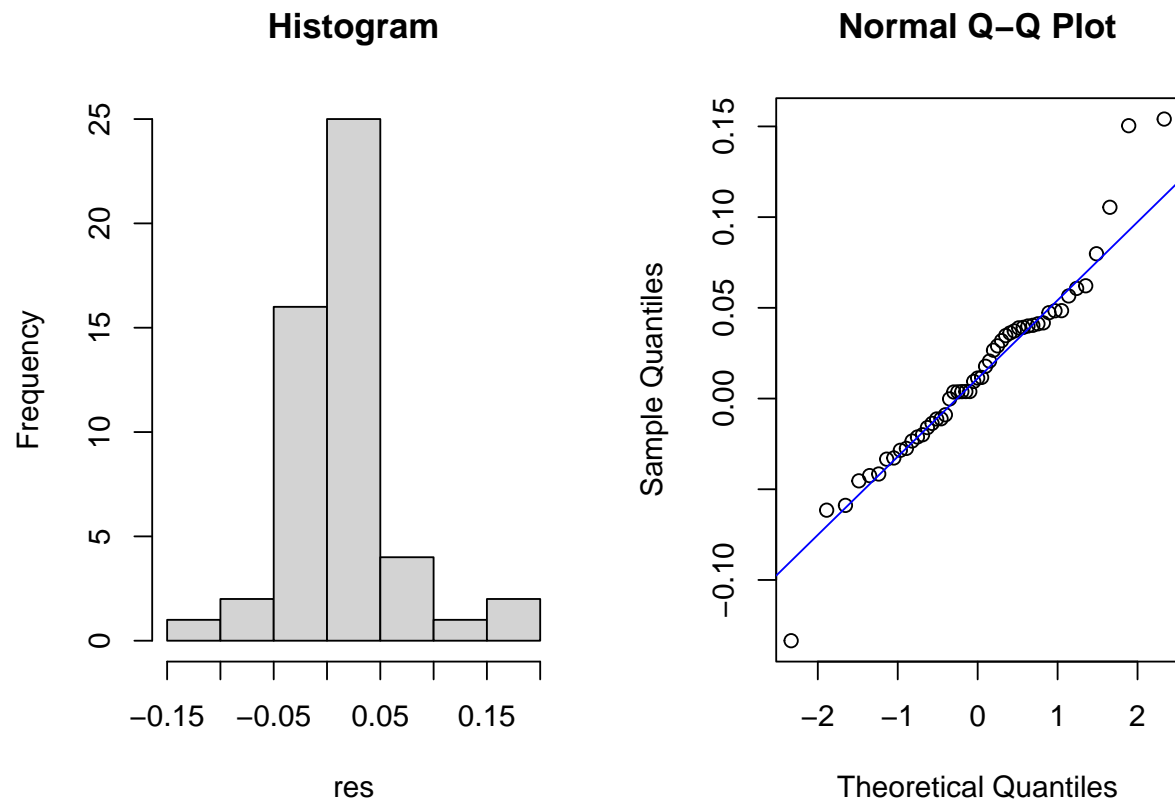


```
##  
## Box-Pierce test  
##  
## data: res  
## X-squared = 6.7, df = 5, p-value = 0.2
```

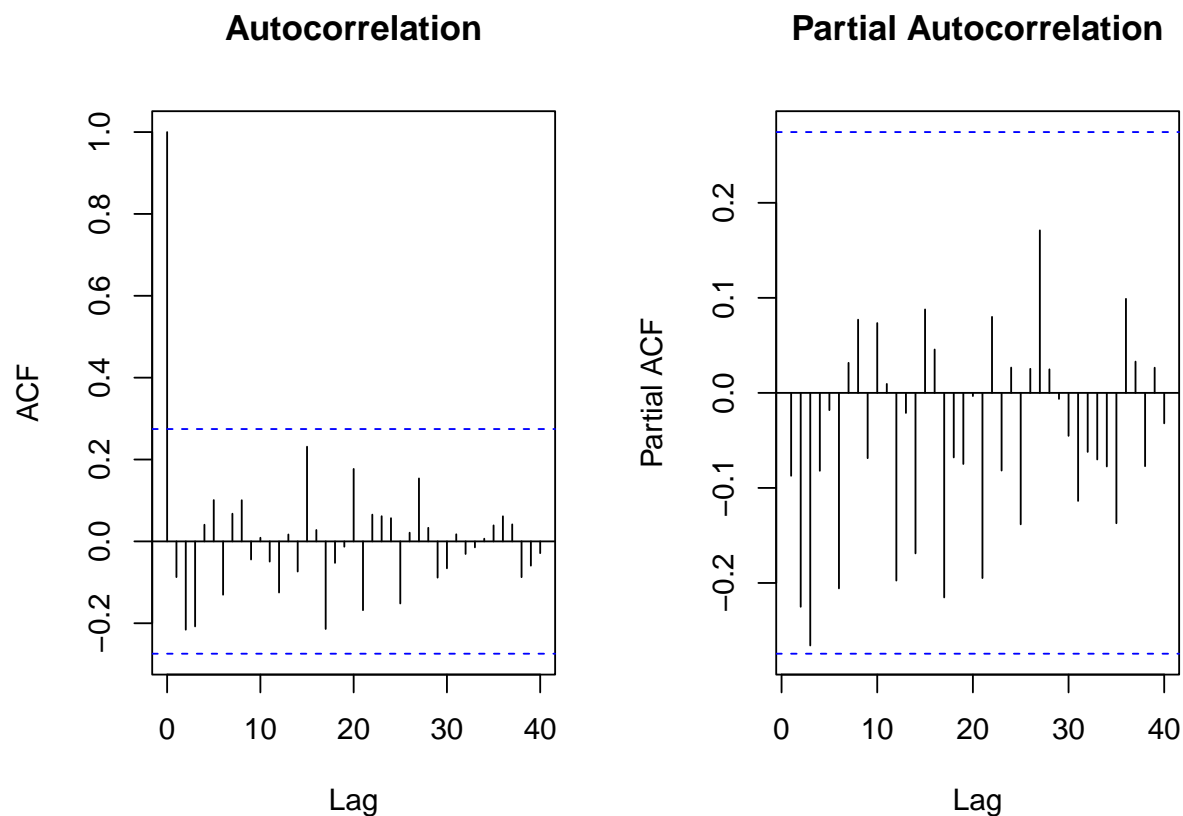
```
##  
## Box-Ljung test  
##  
## data: res  
## X-squared = 7.4, df = 5, p-value = 0.2
```

```
##  
## Box-Ljung test  
##  
## data: res^2  
## X-squared = 3.8, df = 7, p-value = 0.8
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: res  
## W = 0.96, p-value = 0.06
```



Model 1 passes all tests and residuals are normal. Next, I will check ACF and PACF of residuals.



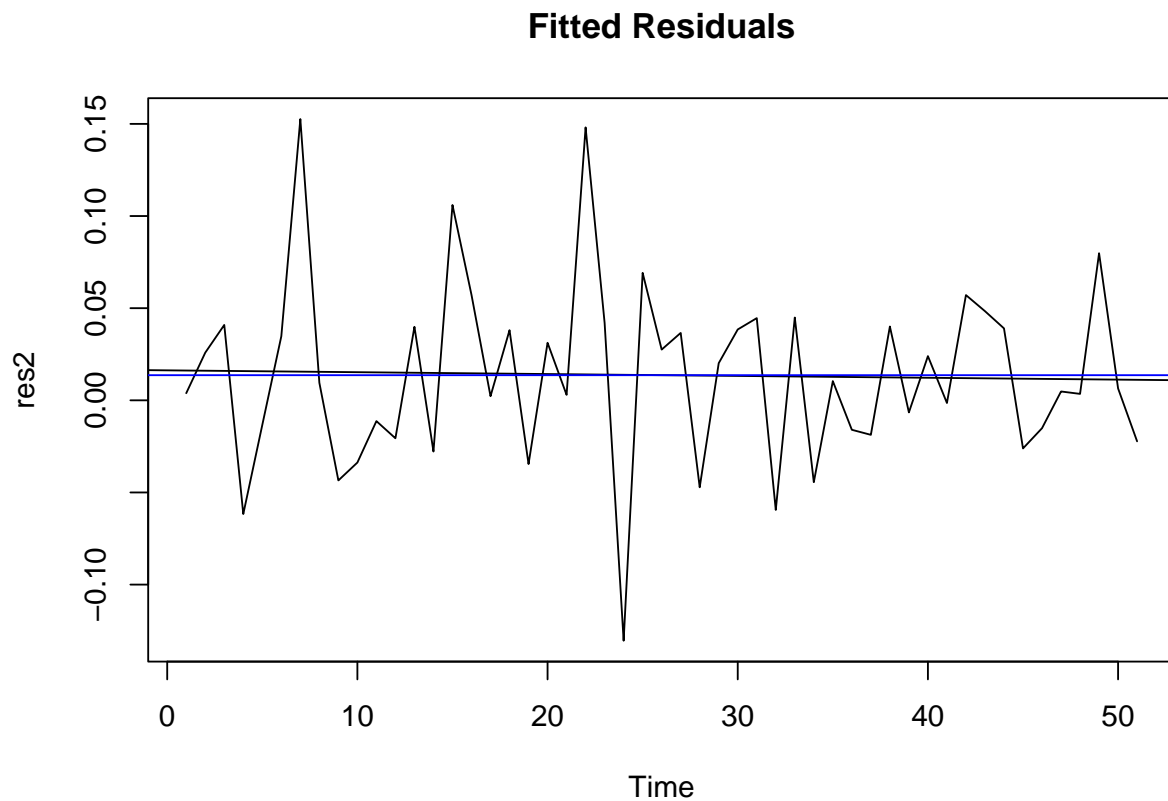
The ACF and PACF of residuals shows no lags outside of confidence interval. The fitted residuals are to $AR(0)$, so it is white noise. This passes diagnostic checking for model 1.

Model Diagnostics for Model 2

This will be model diagnostics for model 2: SARIMA (1,1,0),(1,0,1), period at 9. I found mean then variance of the residuals of model 2. Since the training data has 51 observations, the box.test will have sqrt of 65 or around 7.

```
## [1] 0.01364
```

```
## [1] 0.002493
```

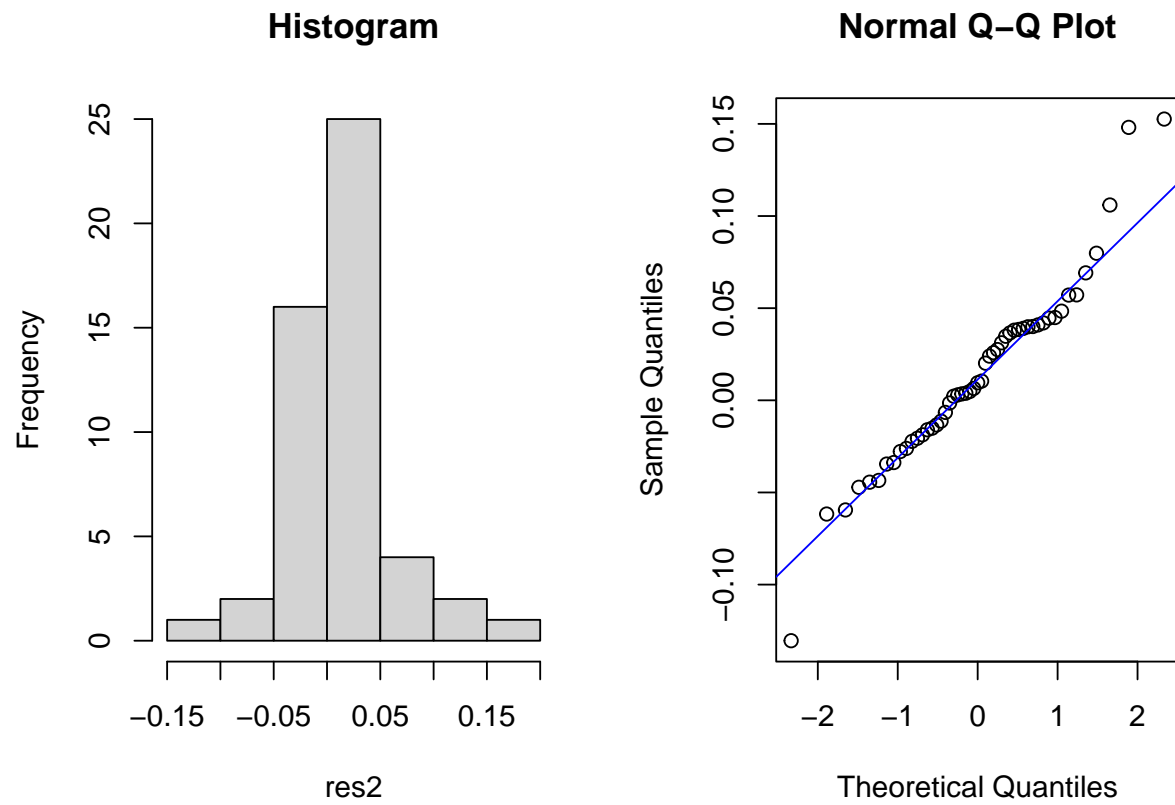


```
##  
## Box-Pierce test  
##  
## data: res2  
## X-squared = 6.8, df = 4, p-value = 0.1
```

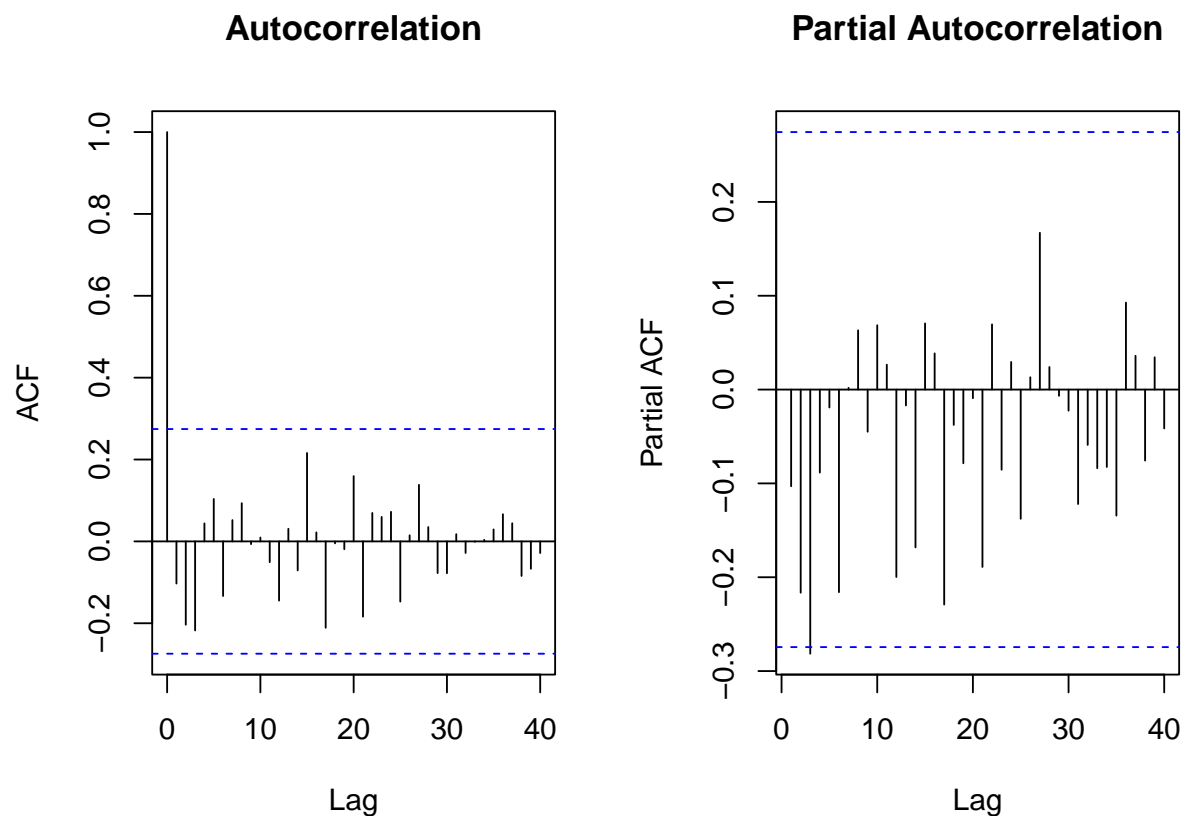
```
##  
## Box-Ljung test  
##  
## data: res2  
## X-squared = 7.5, df = 4, p-value = 0.1
```

```
##  
## Box-Ljung test  
##  
## data: res2^2  
## X-squared = 3.7, df = 7, p-value = 0.8
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: res2  
## W = 0.96, p-value = 0.07
```



Model 2 passes all tests and residuals are normal. Next, I will check ACF and PACF of residuals.



The ACF of residuals shows no lags outside, but PACF does show one lag barely outside of the confidence interval. This results in model 2 failing diagnostic checking.

Best model

Since the analysis of residuals in model 1 was satisfactory and passed diagnostic checking, I will use fit1.

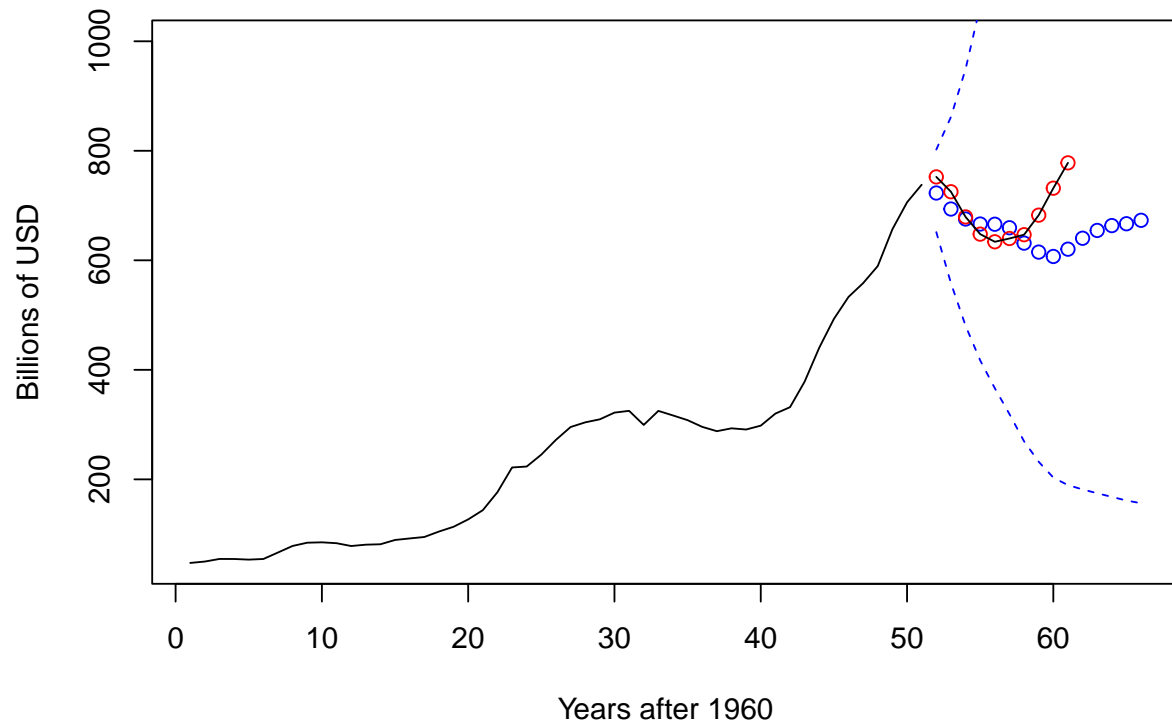
```
##
## Call:
## arima(x = training.log, order = c(1, 1, 0), seasonal = list(order = c(1, 0,
##    0), period = 9), method = "ML")
##
## Coefficients:
##      ar1      sar1
##    0.830  -0.566
## s.e.  0.079   0.122
##
## sigma^2 estimated as 0.00271:  log likelihood = 74.61,  aic = -143.2
```

The final model can be written as $(1 - 0.83B)(1 + 0.566B^9)(1 - B)X_t = Z_t, Z_t \sim WN(0, 0.00271)$

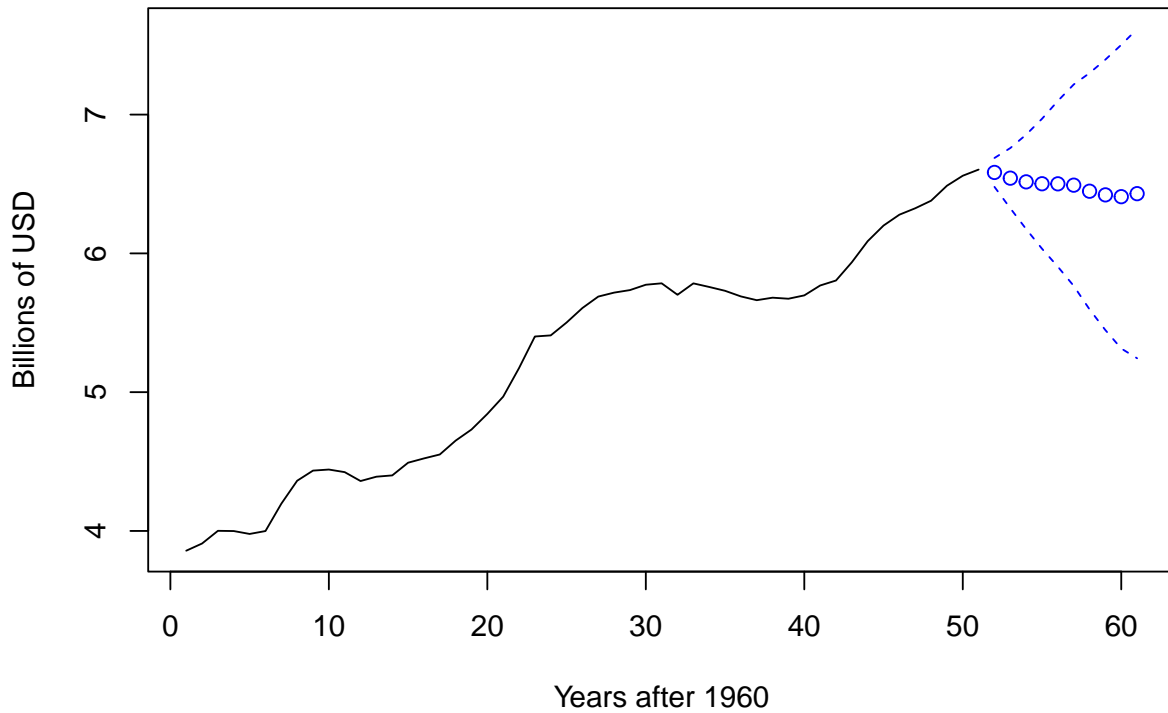
Data Forecasting

In the final step of forecasting, I will predict 10 future observations (colored in blue) and plot it against the testing set of the original data (colored in red and connected with a line) for comparison. Then, I will do the same for the transformed data. For fun, I predicted 5 extra years from my prediction in the original data.

Forecast on Original Time Series



Forecast on Transformed Time Series



Conclusion

Model 1 passed diagnostic checking but Model 2 failed. Ultimately, my predicted model followed closely with the actual model at first, but then predicted in a different direction towards the end of the actual data. My model correctly predicted a seasonality, but the period might have been different. Instead of a period at 9, another period to try could be at 10 or 11, as shown in the ACF and PACF of dat.1. The goal of creating a model that could predict future values of the original model was achieved. Although overall the model did correctly follow the points in the testing set in the short-term future, predicting points in the long-term future was not as accurate. Therefore, I would be hesitant to use my predicted model to predict the defense budget in 10-20 years. My model consisted of this formula : $(1 - 0.83B)(1 + 0.566B^9)(1 - B)X_t = Z_t, Z_t \sim WN(0, 0.00271)$. I would like to acknowledge Raisa Feldman, Youhong Lee, and Sunpeng Duan for their contributions and assistance with this project and thank them for taking the time, effort, and energy out of their busy schedules to help me.

References

I used this site to obtain my original dataset: <https://www.kaggle.com/brandonconrady/us-military-spending-by-year-1960-2020/version/1>

Appendix

```
library(knitr)
library(tidyverse)
library(ISLR)
library(ROCR)
library(dplyr)
library(tinytex)
library(MASS)
#devtools::install_github("FinYang/tsdl")
library(tsd1)
library(forecast)
library(astsa)
library(qpcR)
spending_data <- read.csv("/Users/josephchang/Desktop/MilitarySpending.csv.xls")

# added line for trend
op=par(mfrow=c(1,2))
plot.ts(spending_data$DefenseBudget, ylab = "Defense budget (in US billion)", xlab = "Year after 1960",
nt = length(spending_data$DefenseBudget)
fit <- lm(spending_data$DefenseBudget ~ as.numeric(1:nt))
abline(fit, col = "red")
# added mean
data_mean <- mean(spending_data$DefenseBudget)
abline(h=data_mean, col="blue")

# histogram of original data points
hist(spending_data$DefenseBudget, label=TRUE, main = "Histogram of Defense Budget", breaks="Sturges", xlab = "Defense Budget")

# Define training and testing sets
training = spending_data$DefenseBudget[c(1:51)]
testing = spending_data$DefenseBudget[c(52:61)]

# time series plot in training with trend and mean
op=par(mfrow=c(1,2))
plot.ts(training, ylab = "Defense budget (in US billion)", xlab = "Year after 1960", main = "Time series plot of training set")
fit <- lm(training ~ as.numeric(1:length(training)))
abline(fit, col="red")
abline(h=mean(training), col = "blue")

# histogram of original data points
hist(training, label=TRUE, main = "Histogram of training set", breaks= "Sturges", xlab = "Defense Budget")
t <- 1:length(training)
fit <- lm(training~t)
bcTransform <- boxcox(training ~ t, plotit=TRUE)
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
# define boxcox transformation and log transformation for training set
training.bc = (1/lambda)*(training^lambda-1)
training.log = log(training)
# compare transforms on time series plot
op=par(mfrow=c(2,2))
plot.ts(training, main = "Original Training set")
plot.ts(training.bc, main = "Box-cox Transform")
```

```

plot.ts(training.log, main = "Log Transform")

# compare transforms on histogram
op=par(mfrow=c(2,2))
hist(training, col = "light blue", main = "Histogram of original training set", breaks = "Sturges")
hist(training.bc, col = "light blue", main = "Histogram of Boxcox transformed Data", breaks="Sturges")
hist(training.log, col = "light blue", main = "Histogram of Log transformed Data", breaks="Sturges")
dat.1 <- diff(training.log, 1)
dat.2 <- diff(dat.1, 1)
dat.3 <- diff(dat.2, 1)

# check variances
var(dat.1)
var(dat.2)
var(dat.3)

# time series plot for dat.1
op=par(mfrow=c(1,2))
plot.ts(dat.1, main = "log training when d=1", type= "l")
fit_a <- lm(dat.1 ~ as.numeric(1:length(dat.1)))
abline(fit_a, col = "red")
abline(h=mean(dat.1), col = "blue")

# histogram of dat.1
hist(dat.1, col = "light blue", xlab = "", main = "Histogram at d=1")
m <- mean(dat.2)
std <- sqrt(var(dat.2))
curve(dnorm(x,m,std), add = T)

# ACF and PACF for dat.1
op=par(mfrow=c(1,2))
op <- par(no.readonly=TRUE)
acf(dat.1, lag.max = 40, main="ACF of difference at lag 1 once", ylim=c(-1,1),xlab="h", ylab= expression(
pacf(dat.1, lag.max = 40, main="PACF of difference at lag 1 once", ylim=c(-1,1),xlab="h", ylab= expression(

# range of parameters
df <- expand.grid(p=0:1, q=0:1, P= 0:1, Q=0:1)
df <- cbind(df, AICc=NA)

for (i in 1:nrow(df)) {
  sarima.obj <- NULL
  try(arima.obj <- arima(training.log, order=c(df$p[i],1, df$q[i]),
    seasonal = list(order=c(df$P[i],0,df$Q[i]), period=9), method="ML"))
  if (!is.null(arima.obj)) { df$AICc[i] <- AICc(arima.obj) }
}

# first five models with lowest AICc
head(df[order(df$AICc),], 5)

# Final model 1
fit1 <- arima(training.log, order = c(1,1,0), seasonal = list(order=c(1,0,0), period=9), method = "ML")
fit1
AICc(fit1)

# Final model 2
fit2 <- arima(training.log, order = c(1,1,0), seasonal = list(order=c(1,0,1), period=9), method = "ML")
fit2

```

```

AICc(fit2)
# residual plots
res <- residuals(fit1)
mean(res)
var(res)

# layout
par(mfrow=c(1,1))
ts.plot(res, main = "Fitted Residuals")
t <- 1:length(res)
fit1.res = lm(res~t)
abline(fit1.res)
abline(h=mean(res), col = "blue")

# Testing independence of residuals
Box.test(res, lag = 7, type = c("Box-Pierce"), fitdf = 2)
Box.test(res, lag = 7, type = c("Ljung-Box"), fitdf = 2)
Box.test(res^2, lag = 7, type = c("Ljung-Box"), fitdf = 0)

# Testing normality of residuals
shapiro.test(res)

# Histogram and qq plot of residuals
par(mfrow=c(1,2))
hist(res, main= "Histogram")
qqnorm(res)
qqline(res, col = "blue")
# ACF and PACF of residuals
par(mfrow=c(1,2))
acf(res, main = "Autocorrelation", lag.max = 40)
pacf(res, main = "Partial Autocorrelation", lag.max = 40)
# residual plots
res2 <- residuals(fit2)
mean(res2)
var(res2)

# layout
par(mfrow=c(1,1))
ts.plot(res2, main = "Fitted Residuals")
t <- 1:length(res2)
fit2.res = lm(res2~t)
abline(fit2.res)
abline(h=mean(res2), col = "blue")

# Testing independence of residuals
Box.test(res2, lag = 7, type = c("Box-Pierce"), fitdf = 3)
Box.test(res2, lag = 7, type = c("Ljung-Box"), fitdf = 3)
Box.test(res2^2, lag = 7, type = c("Ljung-Box"), fitdf = 0)

# Testing normality of residuals
shapiro.test(res2)

# Histogram and qq plot of residuals

```

```

par(mfrow=c(1,2))
hist(res2, main= "Histogram")
qqnorm(res2)
qqline(res2, col = "blue")
# ACF and PACF of residuals
par(mfrow=c(1,2))
acf(res2, main = "Autocorrelation", lag.max = 40)
pacf(res2, main = "Partial Autocorrelation", lag.max = 40)
fit1
# Predict 15 future observations on data and plot on original
m <- length(training)
mypred <- predict(fit1, n.ahead=15)

ts.plot(training, xlim=c(1, length(training)+15), ylim=c(min(training), 1000) , xlab = "Years after 1960")

points((m+1):(m+15), col="blue", exp(mypred$pred))
points((m+1):(m+10), col="red", testing)
lines((m+1):(m+10), col="black", testing, lty= "solid")
lines((exp(mypred$pred +2*mypred$se)), col = "blue", lty= "dashed")
lines((exp(mypred$pred -2*mypred$se)), col = "blue", lty= "dashed")

# predict 10 future observations and plot on transformed data
m <- length(training.log)
mypred <- predict(fit1, n.ahead=10)

ts.plot(training.log, xlim=c(1, length(training.log)+10), ylim=c(min(training.log), max(mypred$pred +2*mypred$se)), xlab = "Years after 1960")

points((m+1):(m+10), col="blue", mypred$pred)
lines((mypred$pred +2*mypred$se), col = "blue", lty= "dashed")
lines((mypred$pred -2*mypred$se), col = "blue", lty= "dashed")

```