Assignment 3

Gavin Kunish

2025-07-07

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

Question 1

```r
load("usgdp.rda")

usgdp = ts(usgdp$GDP, freq = 1, start = 1960)

print(start(usgdp))
```

```
## [1] 1960    1
```

```r
print(end(usgdp))
```

```
## [1] 2017    1
```

```r
print(frequency(usgdp))
```

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

```r
train_usgdp <- window(usgdp, start = 1960, end = 2012)

print(start(train_usgdp))
```

```
## [1] 1960    1
```

```r
print(end(train_usgdp))
```

```
## [1] 2012    1
```

```r
test_usgdp <- window(usgdp, start = 2013, end = 2017)

print(start(test_usgdp))
```

```
## [1] 2013    1
```

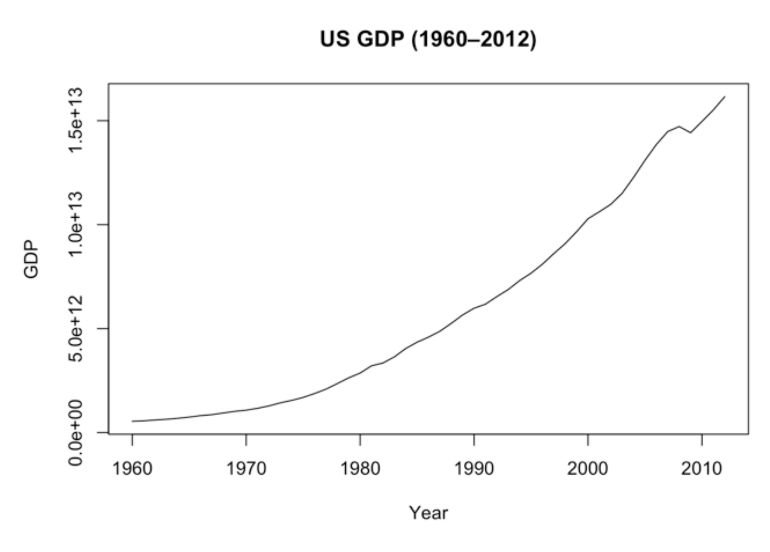
```r
print(end(test_usgdp))
```

```
## [1] 2017    1
```

We start by initalizing the USGDP time series with frequency 1, since it is a yearly GDP series. Then we slice our data into a train set and a test set with the window() function. After each initalization, we check to make sure the start and end date is correct.

Question 2

```r
plot(train_usgdp, main = "US GDP (1960–2012)", ylab = "GDP", xlab = "Year")
```



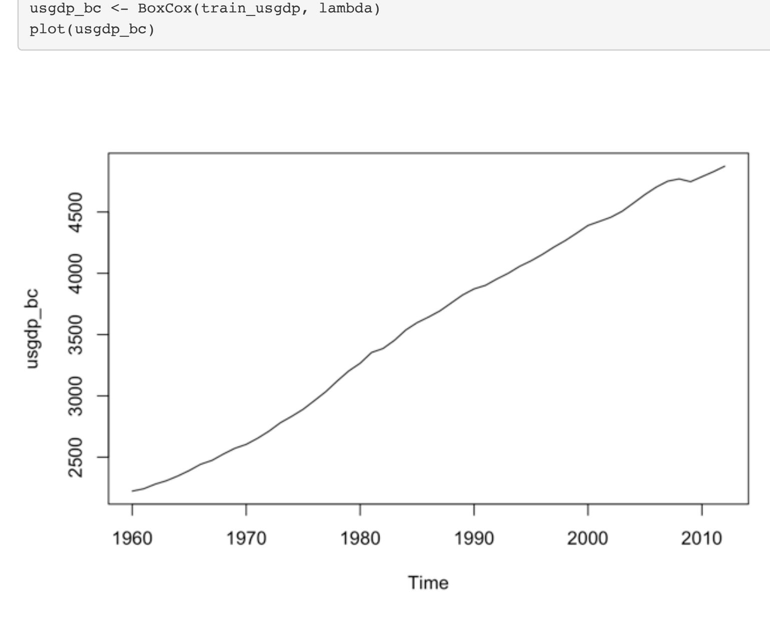

```r
lambda <- BoxCox.lambda(train_usgdp)
lambda
```

```
## [1] 0.2310656
```

Yes a Box-Cox transformation is necessary. In the time series plot we can see that there is an exponential trend, and a BC transformation could fix that. Also we see that the optimal lambda is .231, suggesting that the series is heteroskedastic and non linear.

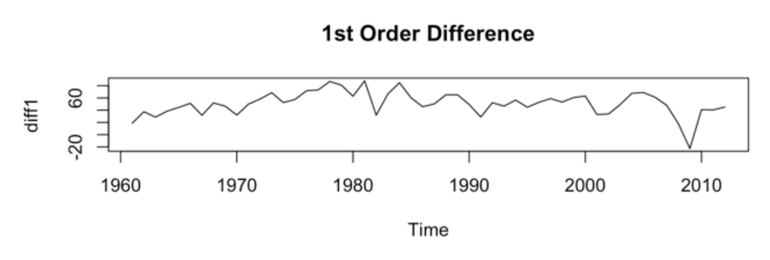
Question 3

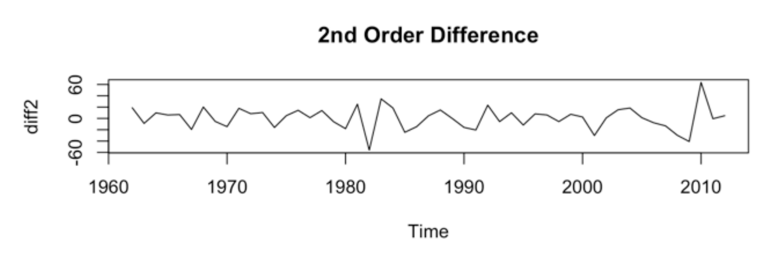
```r
usgdp_bc <- BoxCox(train_usgdp, lambda)
plot(usgdp_bc)
```




```r
diff1 <- diff(usgdp_bc, differences = 1)
diff2 <- diff(usgdp_bc, differences = 2)

par(mfrow = c(2,1))
plot(diff1, main = "1st Order Difference")
plot(diff2, main = "2nd Order Difference")
```






```r
adf.test(diff1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff1
## Dickey-Fuller = -2.7597, Lag order = 3, p-value = 0.2688
## alternative hypothesis: stationary
```

```r
kpss.test(diff1)
```

```
## Warning in kpss.test(diff1): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff1
## KPSS Level = 0.23261, Truncation lag parameter = 3, p-value = 0.1
```

```r
adf.test(diff2)
```

```
## Warning in adf.test(diff2): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff2
## Dickey-Fuller = -5.7003, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(diff2)
```

```
## Warning in kpss.test(diff2): p-value greater than printed p-value
```

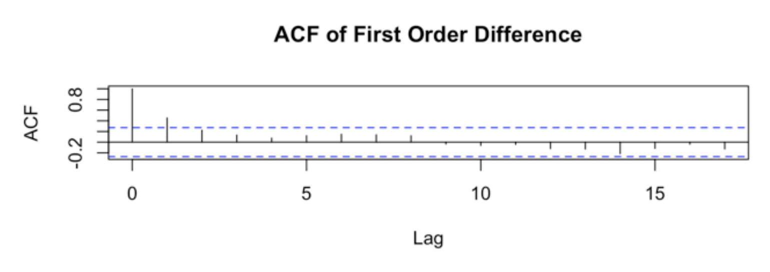
```
##
##  KPSS Test for Level Stationarity
##
## data:  diff2
## KPSS Level = 0.10115, Truncation lag parameter = 3, p-value = 0.1
```

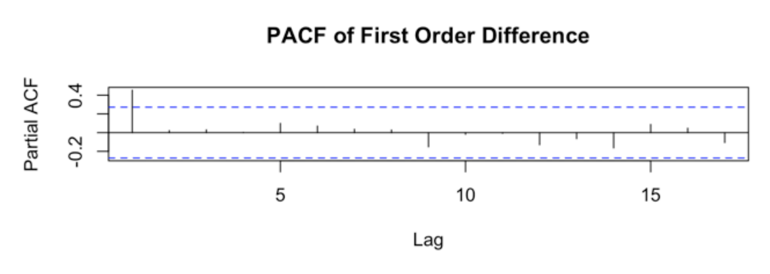
In the ADF test of the first order difference, we have a p-value of .27, thus we Fail to reject the null hypothesis. This suggests the first order difference series is not stationary. For the KPSS test, we have a p-value of >.1, thus we reject the null hypothesis. Contrary to the Dickey Fuller test, this suggests that the first order difference IS stationary.

In the ADF test of the second order difference, we have a p-value of <.01, thus we reject the null hypothesis. This suggests that the second difference series is stationary. In the KPSS test, we have a p-value of >.1, thus we reject the null hypothesis. This suggests that the second order difference series is stationary.

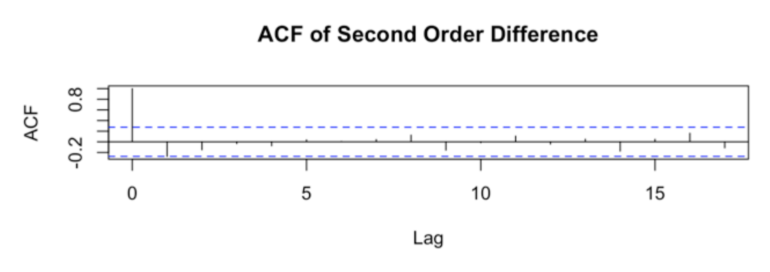
Given the results from the ADF test and the KPSS tests for first and second order differences, there is more evidence to suggest stationality in the second order difference. We can also acknowledge that the first order difference may also be stationary, however we will move foreward with the second order difference as it has a stronger case for stationality.

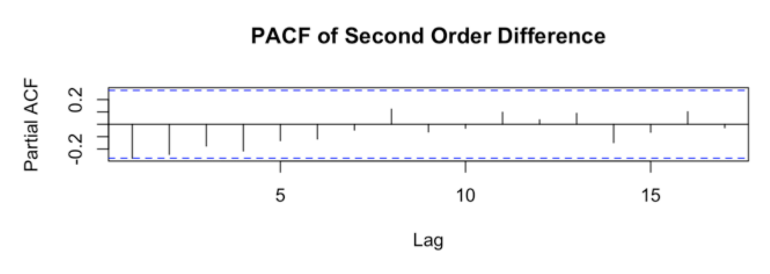
```r
par(mfrow = c(2,1))
acf(diff1, main = "ACF of First Order Difference")
pacf(diff1, main = "PACF of First Order Difference")
```






```r
acf(diff2, main = "ACF of Second Order Difference")
pacf(diff2, main = "PACF of Second Order Difference")
```





In the ACF plot for first order difference, we see the lags decay into a sinusudial pattern. In the PACF we see a spike at lag 1 into a steep drop off. This suggests an AR(1) model.

In the PACF plot for second order difference, we see the lags decay exponentially. We also see the ACF plot have a spike in lag 1 with an immediate drop off after. This suggests an MA(1) model.

Question 4

```r
arima_model <- auto.arima(train_usgdp, seasonal = FALSE, lambda = lambda)
summary(arima_model)
```

```
## Series: train_usgdp
## ARIMA(1,1,0) with drift
## Box Cox transformation: lambda= 0.2310656
##
## Coefficients:
##          ar1    drift
##       0.4728  50.3366
## s.e.  0.1242   4.3713
##
## sigma^2 = 295.7:  log likelihood = -220.81
## AIC=447.62   AICc=448.12   BIC=453.47
##
## Training set error measures:
##                      ME         RMSE         MAE        MPE     MAPE     MASE
## Training set -7063298553 150531242402 83757119701 0.03309517 1.575578 0.268657
##                    ACF1
## Training set 0.07372654
```

The suggested ARIMA model is an ARMIA(1,1,0) with drift. Although in our evaluation of the differencing order, we suggested 2, we also acknowledged that 1 would work since it did pass the KPSS test. In our evaluation of the ACF and PACF we determined that the first order difference was an AR(1) model, which lines up with what the ARIMA model is suggesting.

The first coefficient reported in this arima model is AR(1). It has an estimate of .47 and a standard error of .1242 which we deem statistically significant. The second coefficient reported in this model is drift. It gives an estimate of 50.3366 and a standard error of 4.37 which we also deem to be statistically significant.

Question 5

```r
source("eacf.R")

eacf(diff1)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o o o o o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 o o o o o o o o o o o  o  o  o
## 5 x x o o o o o o o o o  o  o  o
## 6 x x o o o o o o o o o  o  o  o
## 7 o o o x o o o o o o o  o  o  o
```

After visualizing the triangle on the sample eacf, we decide to test out ARMA models (0,2), (1,2), and (0,1).

```r
model_02 <- Arima(train_usgdp, order = c(0,1,2), lambda = lambda)

model_12 <- Arima(train_usgdp, order = c(1,1,2), lambda = lambda)

model_01 <- Arima(train_usgdp, order = c(0,1,1), lambda = lambda)

aicc_02 <- summary(model_02)$aicc
aicc_12 <- summary(model_12)$aicc
aicc_01 <- summary(model_01)$aicc
aicc_auto <-summary(arima_model)$aicc

c("ARIMA(0,1,2)" = aicc_02,
  "ARIMA(1,1,2)" = aicc_12,
  "ARIMA(0,1,1)" = aicc_01,
  "ARIMA(1,1,0)" = aicc_auto)
```

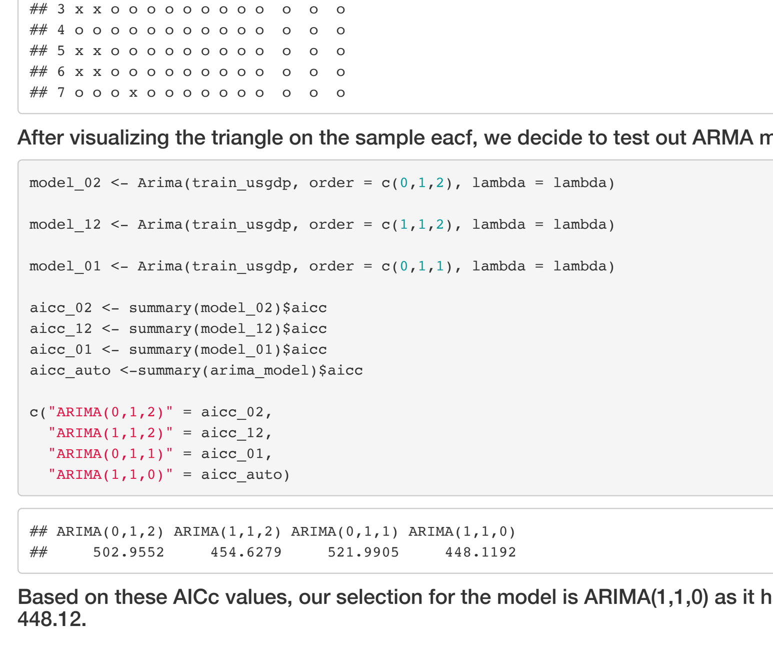
```
## ARIMA(0,1,2) ARIMA(1,1,2) ARIMA(0,1,1) ARIMA(1,1,0)
##     502.9552     454.6279     521.9905     448.1192
```

Based on these AICc values, our selection for the model is ARIMA(1,1,0) as it has the lowest AICc value at 448.12.

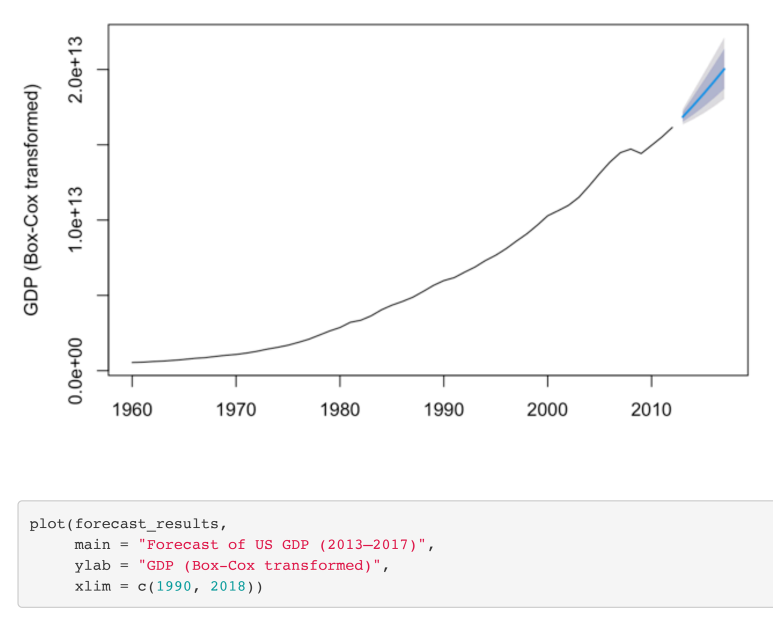
Question 6

```r
forecast_results <- forecast(arima_model, h = 5, level = c(80,95), biasadj = TRUE)

plot(forecast_results, main = "Forecast of US GDP (2013–2017)", ylab = "GDP (Box-Cox transformed)")
```




```r
plot(forecast_results,
     main = "Forecast of US GDP (2013–2017)",
     ylab = "GDP (Box-Cox transformed)",
     xlim = c(1990, 2018))
```



The forecast does seem reasonable. Barring something unexpected, and given what we know about increasing GDP over time, the slight continued upward trend does seem like it fits. We can also see that the confidence interval becomes more uncertain the further out it is.

Question 7

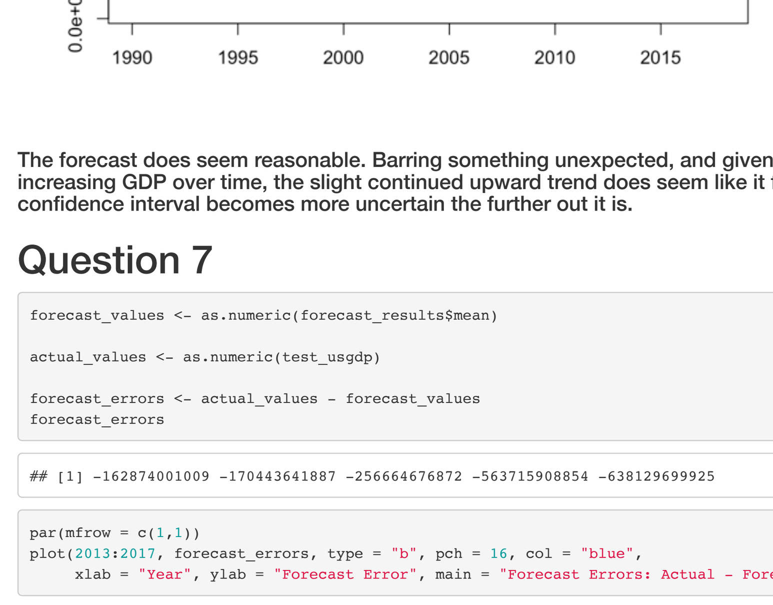
```r
forecast_values <- as.numeric(forecast_results$mean)

actual_values <- as.numeric(test_usgdp)

forecast_errors <- actual_values - forecast_values
forecast_errors
```

```
## [1] -162874001009 -170443641887 -256664676872 -563715908854 -638129699925
```

```r
par(mfrow = c(1,1))
plot(2013:2017, forecast_errors, type = "b", pch = 16, col = "blue",
     xlab = "Year", ylab = "Forecast Error", main = "Forecast Errors: Actual - Forecast")
```



Because the errors are negative, it shows our forecast is underestimating GDP each year. With each year, the error term gets larger, which lines up with the increasing uncertainty in our confidence intervals.

Question 8

```r
sse_arima <- sum(forecast_errors^2)
sse_arima
```

```
## [1] 8.464409e+23
```

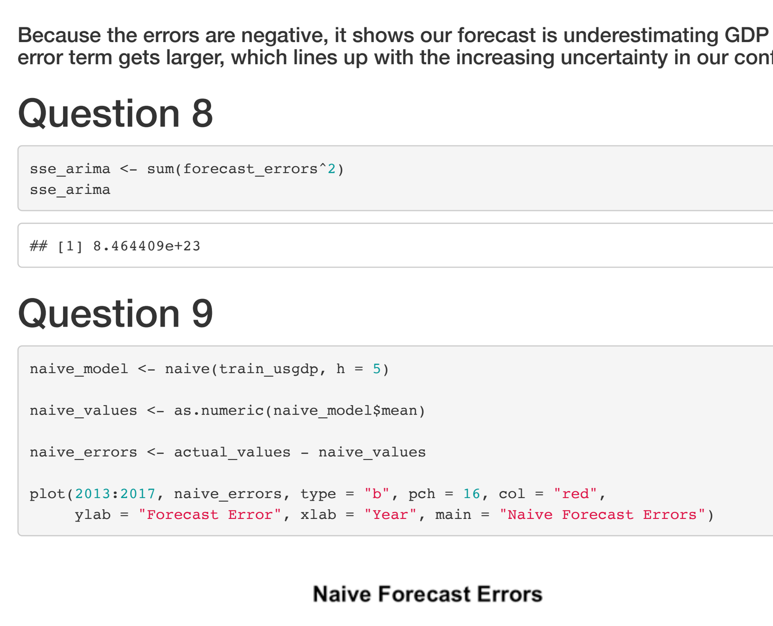
Question 9

```r
naive_model <- naive(train_usgdp, h = 5)

naive_values <- as.numeric(naive_model$mean)

naive_errors <- actual_values - naive_values

plot(2013:2017, naive_errors, type = "b", pch = 16, col = "red",
     ylab = "Forecast Error", xlab = "Year", main = "Naive Forecast Errors")
```




```r
sse_naive <- sum(naive_errors^2)
sse_naive
```

```
## [1] 2.233402e+25
```

The SSE of the naive model is substantially higher than the SSE of the ARIMA model. Because the ARIMA model actually captures the underlying trend and autocorrelation structure as opposed to the naive model which just uses the most recent observation as its prediction, it makes more sense for a time series that sees continuous growth, like the USGDP time series.