Assignment 3: ARIMA

Peter Ye

2024-04-08

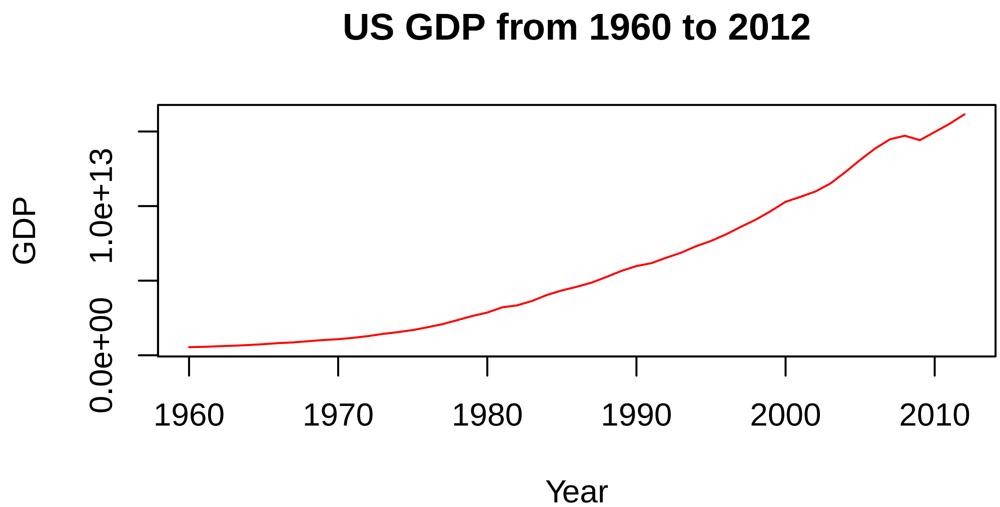
Question 1

```r
# Load usgdp.rda data
load("usgdp.rda")
# usgdp$GDP <- usgdp$GDP / 1e12


# Split the dataset into training and test sets
train_data <- subset(usgdp, Year >= 1960 & Year <= 2012)
test_data <- subset(usgdp, Year >= 2013 & Year <= 2017)
```

Question 2

```r
# Plot the training dataset
plot(train_data$Year, train_data$GDP, type = "l",
     main = "US GDP from 1960 to 2012",
     xlab = "Year", ylab = "GDP", col = "red")
```

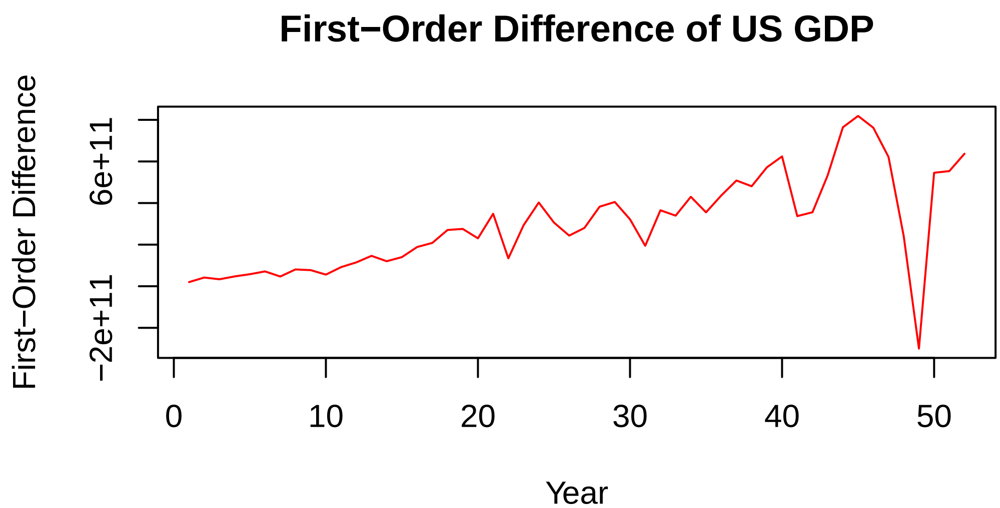


The graph demonstrates the training data is non-stationary and shows an exponential growth trend from 1970 to 2010. This type of trend can lead to non-constant variance in the data. The Box-Cox transformation could potentially be used here to stabilize the variance and make the data more suitable for linear modeling

Question 3

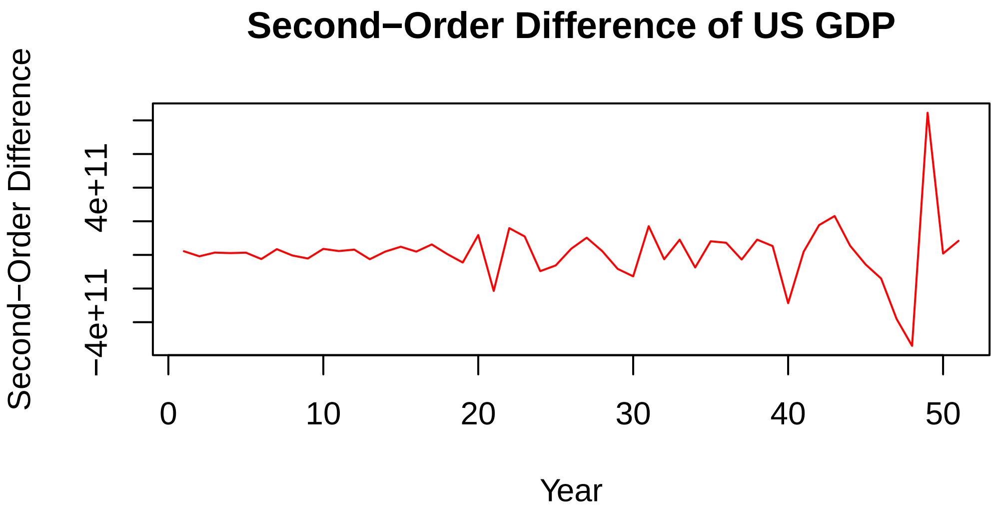
```r
# Calculate first-order difference
first_order_diff <- diff(train_data$GDP, differences = 1)

# Plot first-order difference
plot(first_order_diff, type = "l", main = "First-Order Difference of US GDP",
     xlab = "Year", ylab = "First-Order Difference", col = "red")
```




```r
# Calculate second-order difference
second_order_diff <- diff(train_data$GDP, differences = 2)

# Plot second-order difference
plot(second_order_diff, type = "l", main = "Second-Order Difference of US GDP",
     xlab = "Year", ylab = "Second-Order Difference", col = "red")
```




```r
# Apply KPSS Test for Stationarity on the first-order difference
kpss_test_first_order <- kpss.test(first_order_diff)

# Apply KPSS Test for Stationarity on the second-order difference
kpss_test_second_order <- kpss.test(second_order_diff)

# Print the results
print(kpss_test_first_order)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  first_order_diff
## KPSS Level = 1.1194, Truncation lag parameter = 3, p-value = 0.01
```

```r
print(kpss_test_second_order)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  second_order_diff
## KPSS Level = 0.030686, Truncation lag parameter = 3, p-value = 0.1
```

Based on the test results: the second-order difference of the data results in a stationary dataset, which has a p-value greater than 0.1 and fails to reject the null hypothesis of stationarity. It implies that the dataset after second-order differencing is stationary.

Question 4

```r
# Estimate the Box-Cox transformation parameter lambda
lambda <- BoxCox.lambda(train_data$GDP)
print(lambda)
```

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

A lambda value of 0.2310656, suggests that a Box-Cox transformation could potentially improve the statistical properties of the dataset, because this value is quite far from 1

```r
# Fit the ARIMA model to the transformed data
arima_model <- auto.arima(train_data$GDP, lambda = "auto")
```

```r
# Report the ARIMA model
summary(arima_model)
```

```
## Series: train_data$GDP
## ARIMA(1,1,0) with drift
## Box Cox transformation: lambda= 0.2310592
##
## Coefficients:
##          ar1    drift
##       0.4728  50.3273
## s.e.  0.1242   4.3705
##
## sigma^2 = 295.6:  log likelihood = -220.8
## AIC=447.6   AICc=448.1   BIC=453.45
##
## Training set error measures:
##                       ME         RMSE         MAE        MPE     MAPE      MASE
## Training set -7064027745 150531586247 83757220769 0.03310066 1.575576 0.2686573
##                    ACF1
## Training set 0.07372865
```

The model is ARIMA(1,1,0), which means it has no autoregressive terms (p=1), it's differenced twice (d=1), and has one moving average term (q=0). The coefficient for the AR1 term is 0.4728, with a standard error of 0.1242, indicating the relationship between a given observation and the one preceding it, adjusted for the differencing. The drift coefficient is 50.3273 with a standard error of 4.3705, which shows the average increase per time unit after adjusting for the AR1 effect.

Question 5

```r
# Compute the sample Extended ACF
eacf(train_data$GDP)
```

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

```
## 3 x o o o o o o o o o o o  o  o  o
## 4 x x o o o o o o o o o o  o  o  o
## 5 x o o o o o o o o o o o  o  o  o
## 6 x o o o o o o o o o o o  o  o  o
## 7 x o o o o o o o o o o o  o  o  o
```

```r
# Define the range for p, d, and q
p_range <- 0:2
d_range <- 0:2
q_range <- 0:2

# Initialize a list to store models
model_list <- list()
aic_values <- numeric()

# Loop through possible combinations of p, d, and q
for(p in p_range) {
  for(d in d_range) {
    for(q in q_range) {
      # Define the ARIMA model with the current p, d, q values
      model <- Arima(train_data$GDP, order=c(p, d, q), lambda = "auto")

      # Save the model to the list
      model_id <- paste("ARIMA", p, d, q, sep="_")
      model_list[[model_id]] <- model

      # Save the AICc value
      aic_values[model_id] <- model[["aicc"]]

      # Print the summary of the model
      # cat(paste("ARIMA(", p, ",", d, ",", q, ")\n", sep=""))
      # print(summary(model))
      # cat("\n\n")
    }
  }
}

# Find the model with the smallest AICc
best_model_id <- names(which.min(aic_values))
best_model <- model_list[[best_model_id]]
best_aicc <- aic_values[best_model_id]

cat("The best model is ", best_model_id, " with an AICc of ", best_aicc, "\n", sep="")
```

```
## The best model is ARIMA_0_2_2 with an AICc of 442.2578
```

```r
summary(Arima(train_data$GDP, order=c(0, 2, 2), lambda = "auto"))
```

```
## Series: train_data$GDP
## ARIMA(0,2,2)
## Box Cox transformation: lambda= 0.2310592
##
## Coefficients:
```

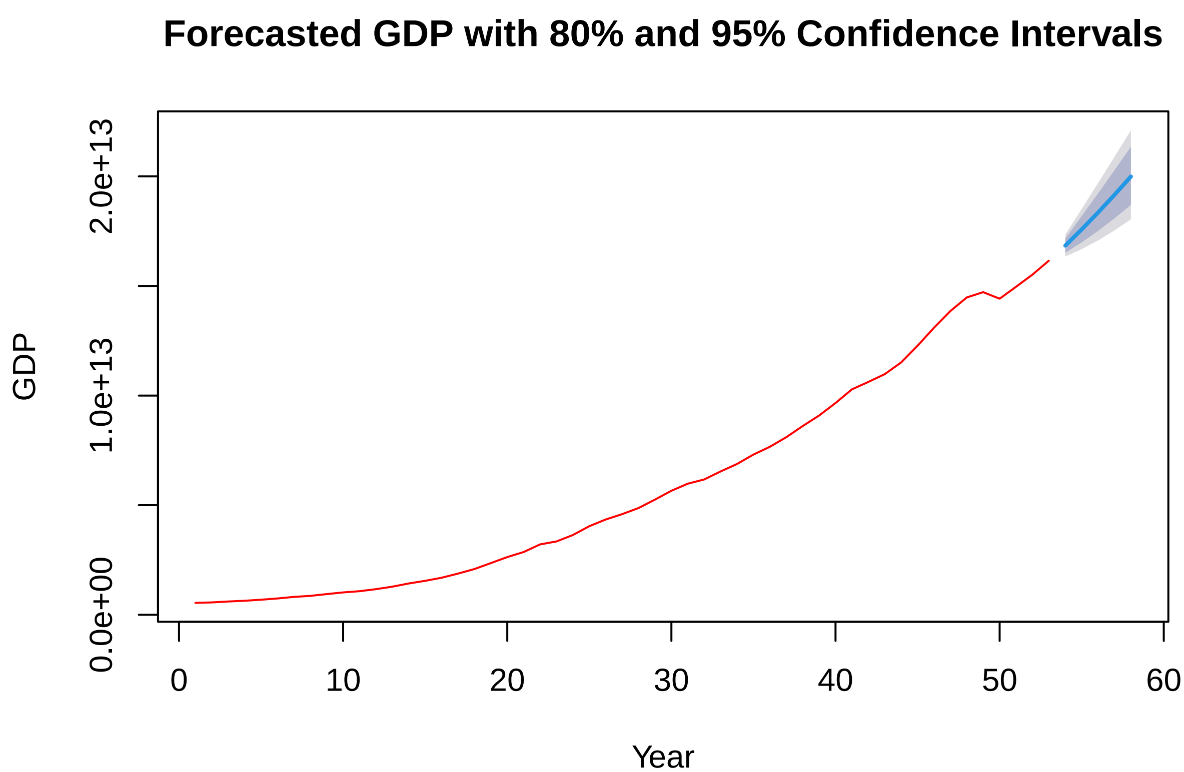
```
##           ma1      ma2
##       -0.4938  -0.2277
## s.e.   0.1379   0.1343
##
## sigma^2 = 309.3:  log likelihood = -217.87
## AIC=441.75   AICc=442.26   BIC=447.54
##
## Training set error measures:
##                        ME         RMSE         MAE      MPE     MAPE      MASE
## Training set -13495961506 158676310471 88037514214 0.224507 1.603328 0.2823866
##                   ACF1
## Training set 0.1097139
```

The Extended Autocorrelation Function (EACF) indicates that potential models could include configurations such as (p=1, q=1), (p=1, q=2), (p=2, q=0), (p=2, q=1), and (p=2, q=2). This selection of d=2 corroborates the findings from Question 3.

Question 6

```
# Extract forecast for the comparison period
forecasted_values <- forecast(arima_model, h=5, lambda = lambda,level=c(80, 95))

# Plot the forecast with 80% and 95% confidence intervals
plot(forecasted_values, main="Forecasted GDP with 80% and 95% Confidence Intervals", xlab="Year", ylab=
```



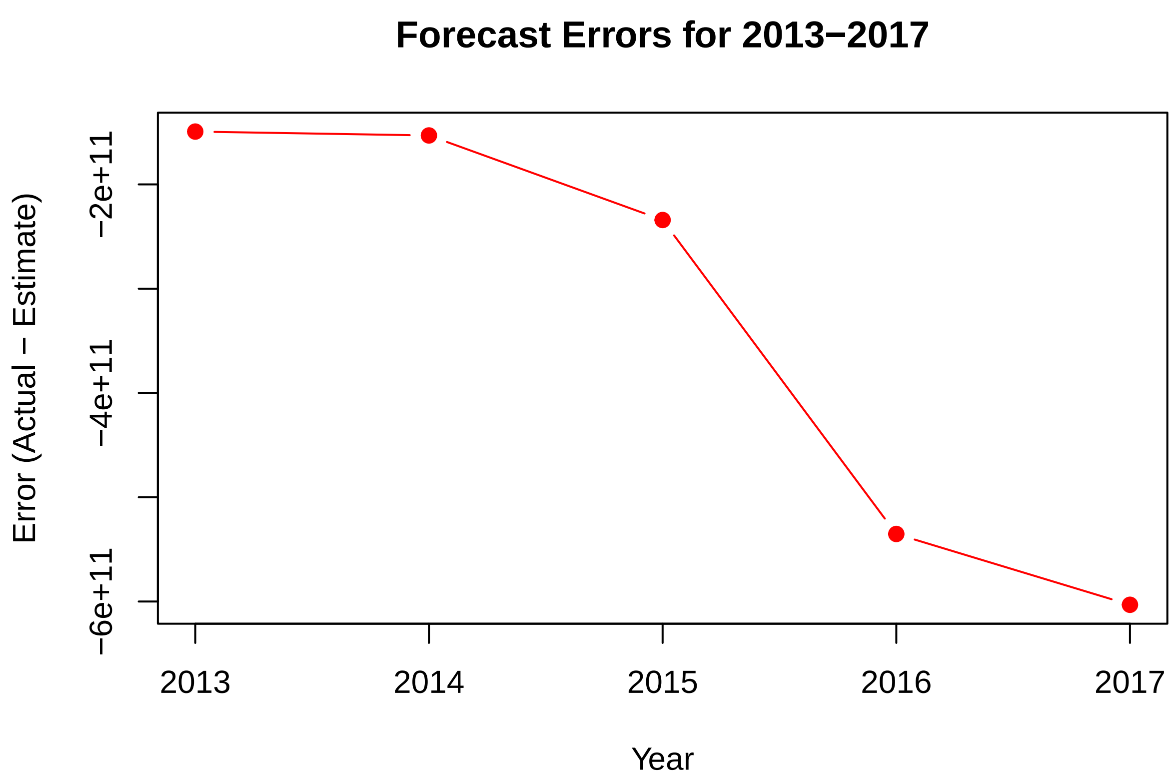
Question 7

```r
# Extract forecast estimates for the comparison period
forecast_estimates <- forecasted_values$mean

# Ensure the order of the 'test_data' matches the forecast period
test_data <- test_data[order(test_data$Year),]

# Calculate the errors
errors <- test_data$GDP - forecast_estimates

# Plot the errors
plot(test_data$Year, errors, type="b", pch=19, col="red",
     main="Forecast Errors for 2013-2017", xlab="Year", ylab="Error (Actual - Estimate)")
```




```r
forecasted_values
```

```
##    Point Forecast        Lo 80        Hi 80        Lo 95        Hi 95
## 54   1.684088e+13 1.651722e+13 1.716939e+13 1.634784e+13 1.734527e+13
## 55   1.758068e+13 1.698851e+13 1.818859e+13 1.668131e+13 1.851688e+13
## 56   1.835487e+13 1.751341e+13 1.922707e+13 1.708015e+13 1.970150e+13
## 57   1.915972e+13 1.808590e+13 2.028190e+13 1.753653e+13 2.089604e+13
## 58   1.999374e+13 1.869999e+13 2.135525e+13 1.804173e+13 2.210424e+13
```

The plot demonstrates that the residuals trend upwards over time, suggesting a decline in prediction accuracy with the progression of time.

Question 8

```r
# Calculate SSE
arima_sse <- sum(errors^2)

arima_sse
```

```
## [1] 7.508302e+23
```

Question 9

```r
naive_forecast <- naive(train_data$GDP, h=5, lambda = "auto")

# Calculate errors for the naive forecast
naive_errors <- test_data$GDP - naive_forecast$mean

# Calculate Sum of Squared Errors (SSE) for the naive forecast
naive_sse <- sum(naive_errors^2)
```

```r
# Compare the SSE of the ARIMA model with the naive forecast
cat("SSE for ARIMA model:", arima_sse, "\n")
```

```
## SSE for ARIMA model: 7.508302e+23
```

```r
cat("SSE for Naive forecast:", naive_sse, "\n")
```

```
## SSE for Naive forecast: 2.233402e+25
```

```r
if(arima_sse < naive_sse) {
  cat("The ARIMA model performed better than the naive approach.\n")
} else if(arima_sse > naive_sse) {
  cat("The naive approach performed better than the ARIMA model.\n")
} else {
  cat("Both the ARIMA model and the naive approach have the same performance.\n")
}
```

```
## The ARIMA model performed better than the naive approach.
```