Stat 145 Problem Set 2

Anne Christine Amores

October 8, 2025

1. **Import the datasets into R. Run the codes and diagnostics. You should arrive at 3 competing (S)ARIMA models.**

```
# Importing the dataset and creating ts object
GDP <- read_excel("GDP.xlsx", sheet = "GDP_Dataset") # as df
GDP.ts <- ts(GDP[[2]], frequency = 4, # as ts object
             start=c(2000,1),end=c(2025,2))
diff.GDP.ts <- diff(GDP.ts) # first difference as ts object
```

```
# Model 1: SARIMA(1,1,0) x (0,1,1): Added (1-B^s)
gdp.arima3 <- Arima(GDP.ts,
                    order=c(1,1,0), lambda=0,
                    seasonal=c(0,1,1), include.drift=TRUE) # drift term is a constant
```

```
## Warning in Arima(GDP.ts, order = c(1, 1, 0), lambda = 0, seasonal = c(0, : No
## drift term fitted as the order of difference is 2 or more.
```

```
# Model 2: SARIMA(12,1,0) x (0,0,1): Added up to AR(12)
gdp.arima4 <- Arima(GDP.ts,
                    order = c(12,1,0), lambda = 0, # lambda = 0 for log transform
                    seasonal = c(0,0,1) , include.drift = TRUE) # drift term is a constant
# Model 3: SARIMA(12,1,0) x (0,0,1): Restricted some coeffs to 0
gdp.arima5 <- Arima(GDP.ts,
                    order = c(12,1,0), lambda = NULL,
                    seasonal = c(0,0,1),
                    method = "ML",
                    fixed = c(NA, NA, NA, NA, NA, NA, 0, 0, NA, 0, 0, NA, # AR terms
                              NA, # SMA term
                              NA), # drift
                    include.drift = TRUE) # drift term is a constant
```

2. **In multiplicative (S)ARIMA format, write out the complete specification of the models. Include AR, MA, SAR, SMA, and other relevant terms. Make sure you include transformations (if there are any).**

Note that $Y_t = log(GDP_t)$ since `lambda = 0` in `Arima`.

Model 1: $SARIMA(1,1,0) \times (0,1,1)_4$:

$$(1-\phi_1 B)(1-B)^1(1-B^4)^1 log(GDP_t) = (1+\Theta_1 B^4)\epsilon_t,$$

which has no MA and SAR terms.

Model 2: $SARIMA(12, 1, 0)$Ö$(0, 1, 1)_4$:

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_{12} B^{12})(1 - B)^1 log(GDP_t) = (1 + \Theta_1 B^4)\epsilon_t,$$

which has no seasonal diff, MA, and SAR terms.

Model 3: $SARIMA(12, 1, 0)$Ö$(0, 1, 1)_4$ but with $\phi_7 = \phi_8 = \phi_{10} = \phi_{11} = 0$:

$$(1 - \phi_1 B - \phi_2 B^2 - ... - 0 \cdot B^7 - 0 \cdot B^8 - \phi_9 B^9 - 0 \cdot B^{10} - 0 \cdot B^{11} - \phi_{12} B^{12})(1 - B)^1 log(GDP_t) = (1 + \Theta_1 B^4)\epsilon_t,$$

3. **Evaluate the performance of the (S)ARIMA models with a train-test split. Use varying forecast horizons of 1 quarter, 2 quarters, and 1 year. For the train set(s), take: Akaike Information Criterion (AIC), log-likelihood, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). For the test set(s), take: MAE, MAPE, and RMSE.**

```
# Using same scale as models (lamba = 0 --> log transform)
y <- BoxCox(GDP.ts, lambda = 0)

# Splitting into TRAIN and TEST
n <- length(y)
test_h <- 12L  # last 12 quarters as TEST
split_idx <- n - test_h

y_train <- window(y, end = time(y)[split_idx])
y_test  <- window(y, start = time(y)[split_idx + 1])

# Refitting the 3 models on TRAIN:
F3 <- Arima(y_train, order=c(1,1,0), seasonal=list(order=c(0,1,1), period=4),
            include.drift=FALSE)
F4 <- Arima(y_train, order=c(12,1,0), seasonal=list(order=c(0,0,1), period=4),
            include.drift=TRUE)
F5 <- Arima(y_train, order=c(12,1,0), seasonal=list(order=c(0,0,1), period=4),
            include.drift=TRUE, method="CSS",
            fixed=c(NA, NA, NA, NA, NA, NA, 0, 0, NA, 0, 0, NA,  # 12 ARs
                    NA,  # SMA(1)
                    NA)) # drift

# Metrics
mae  <- function(e) mean(abs(e), na.rm=TRUE)
mape <- function(e, a) mean(abs(e/a), na.rm=TRUE)*100
rmse <- function(e) sqrt(mean(e^2, na.rm=TRUE))

eval_one <- function(fit, name, horizons=c(1,2,4)){
  ins <- tibble(Model=name, Set="Train",
                AIC=AIC(fit),
                BIC=BIC(fit),
                LogLik=as.numeric(logLik(fit)),
                MAE=mae(residuals(fit)), MAPE=NA_real_, RMSE=rmse(residuals(fit)))
  fc  <- forecast(fit, h=length(y_test))
  outs <- map_dfr(horizons, function(h){
```

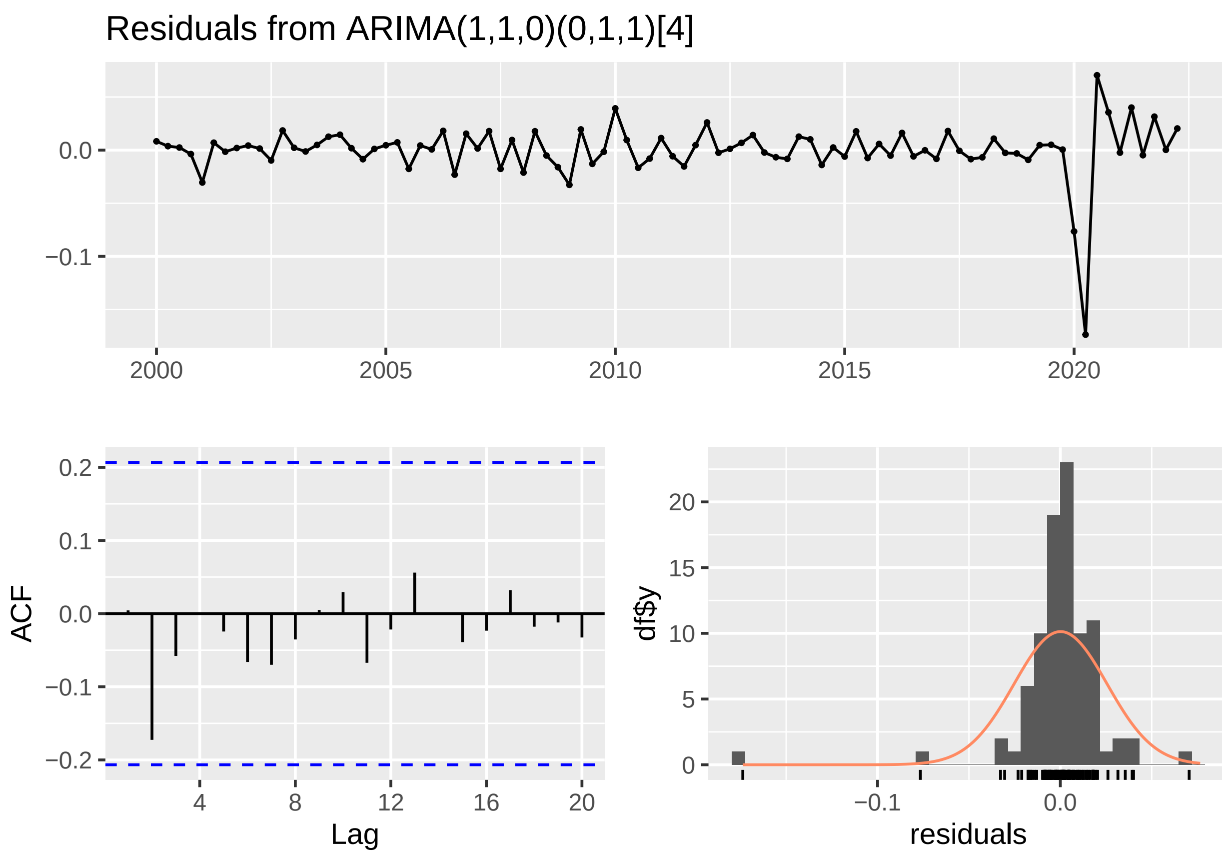
```r
    idx <- 1:h
    e   <- y_test[idx] - fc$mean[idx]   # errors on LOG scale (consistent with fit)
    tibble(Model=name, Set=paste0("Test@h=",h),
           AIC=NA_real_, AICc=NA_real_, BIC=NA_real_, LogLik=NA_real_,
           MAE=mae(e), MAPE=mape(e, y_test[idx]), RMSE=rmse(e))
  })
  bind_rows(ins, outs)
}

tab_perf <- bind_rows(
  eval_one(F3,"Model 3 (1,1,0)(0,1,1)[4]"),
  eval_one(F4,"Model 4 (12,1,0)(0,0,1)[4]"),
  eval_one(F5,"Model 5 (12,1,0)(0,0,1)[4] restricted")
)
```

```
## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): seasonal MA part
## of model is not invertible
```

```r
print(tab_perf, n=Inf)
```

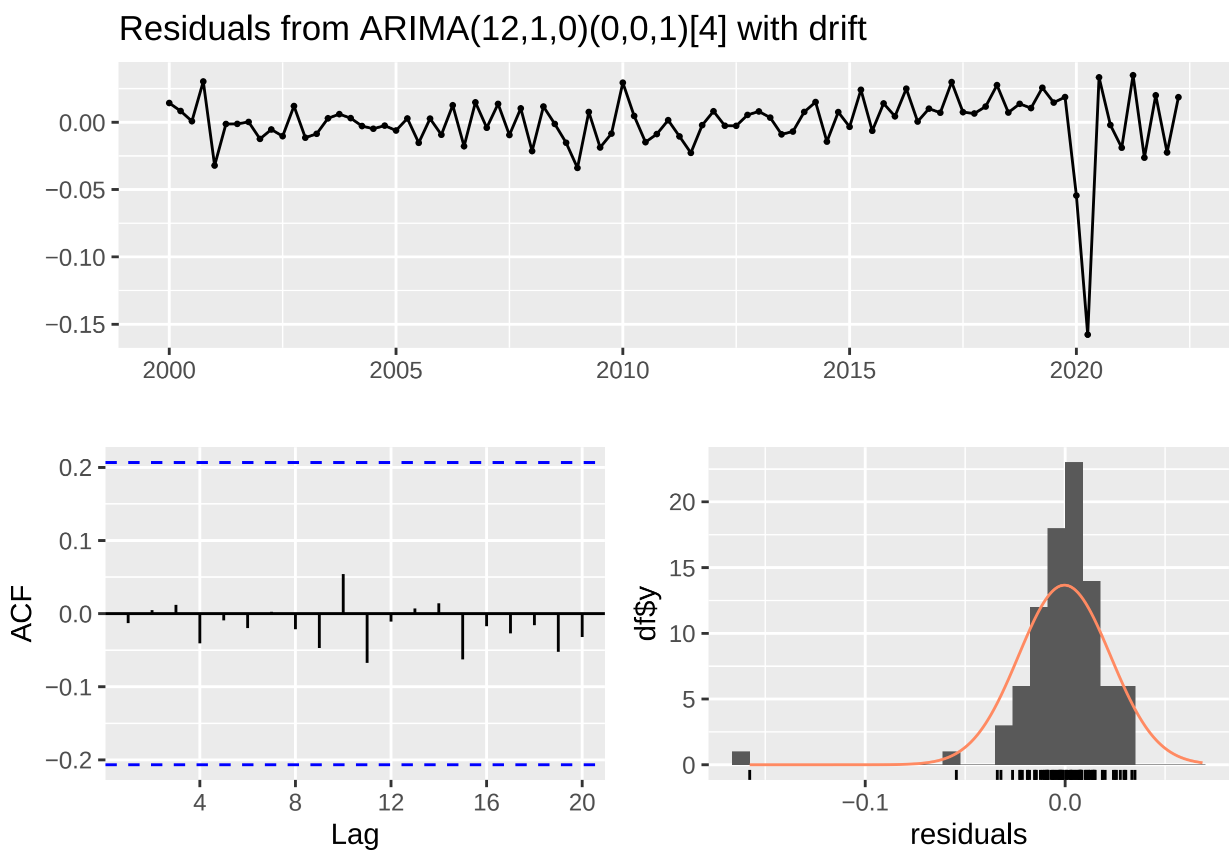
```
## # A tibble: 11 x 9
##    Model                 Set     AIC   BIC LogLik     MAE     MAPE    RMSE  AICc
##    <chr>                 <chr> <dbl> <dbl>  <dbl>   <dbl>    <dbl>   <dbl> <dbl>
##  1 Model 3 (1,1,0)(0,1,~ Train -372. -365.   189. 0.0136  NA       0.0253     NA
##  2 Model 3 (1,1,0)(0,1,~ Test~   NA    NA     NA  0.00125  0.00813 0.00125    NA
##  3 Model 3 (1,1,0)(0,1,~ Test~   NA    NA     NA  0.00734  0.0473  0.00954    NA
##  4 Model 3 (1,1,0)(0,1,~ Test~   NA    NA     NA  0.00554  0.0358  0.00771    NA
##  5 Model 4 (12,1,0)(0,0~ Train -376. -339.   203. 0.0139  NA       0.0229     NA
##  6 Model 4 (12,1,0)(0,0~ Test~   NA    NA     NA  0.0249   0.162   0.0249     NA
##  7 Model 4 (12,1,0)(0,0~ Test~   NA    NA     NA  0.0265   0.172   0.0266     NA
##  8 Model 4 (12,1,0)(0,0~ Test~   NA    NA     NA  0.0439   0.284   0.0479     NA
##  9 Model 5 (12,1,0)(0,0~ Test~   NA    NA     NA  0.0116   0.0754  0.0116     NA
## 10 Model 5 (12,1,0)(0,0~ Test~   NA    NA     NA  0.00966  0.0626  0.00985    NA
## 11 Model 5 (12,1,0)(0,0~ Test~   NA    NA     NA  0.0248   0.161   0.0309     NA
```

```r
# Checking if residuals are white noise (TRAIN) for completeness
checkresiduals(F3)
```



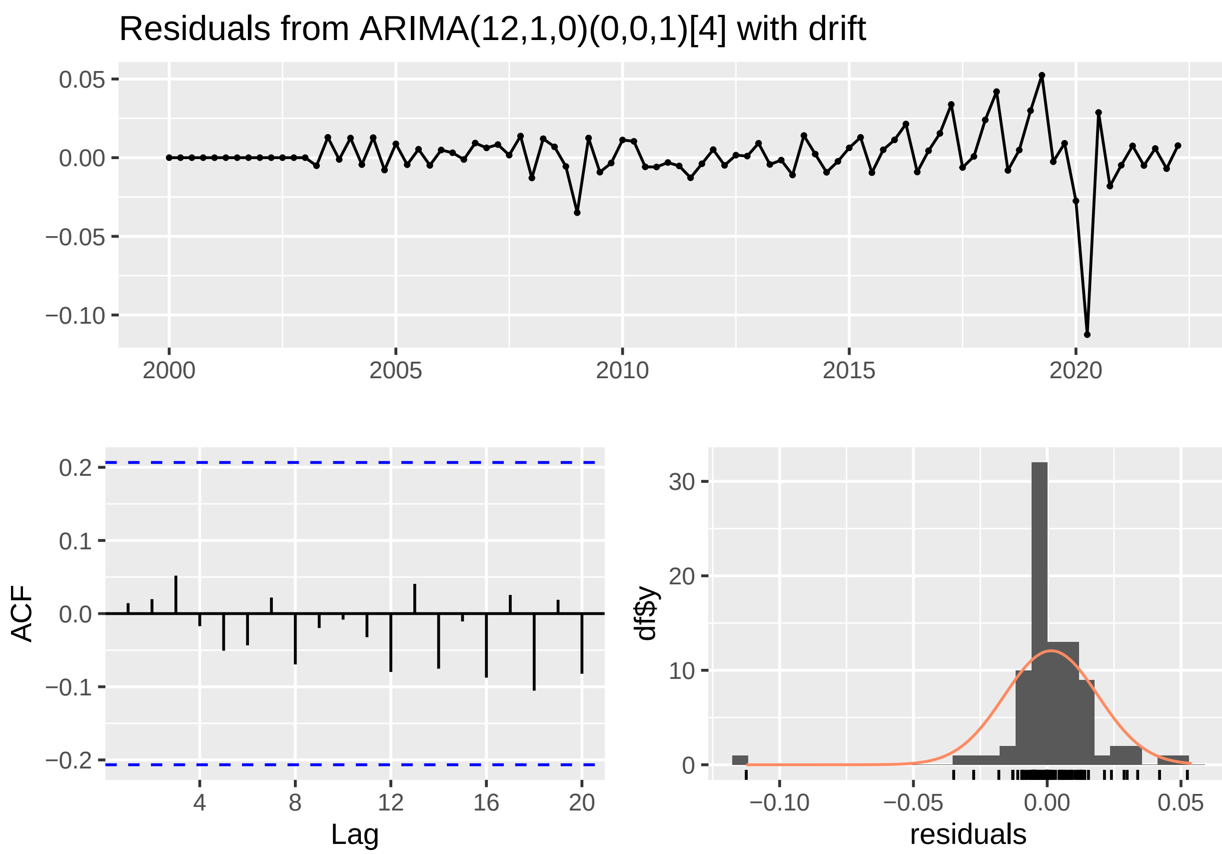

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(0,1,1)[4]
## Q* = 4.2233, df = 6, p-value = 0.6465
##
## Model df: 2.   Total lags used: 8
```

```
checkresiduals(F4)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(12,1,0)(0,0,1)[4] with drift
## Q* = 1.7936, df = 3, p-value = 0.6163
##
## Model df: 13.   Total lags used: 16
```

```
checkresiduals(F5)
```

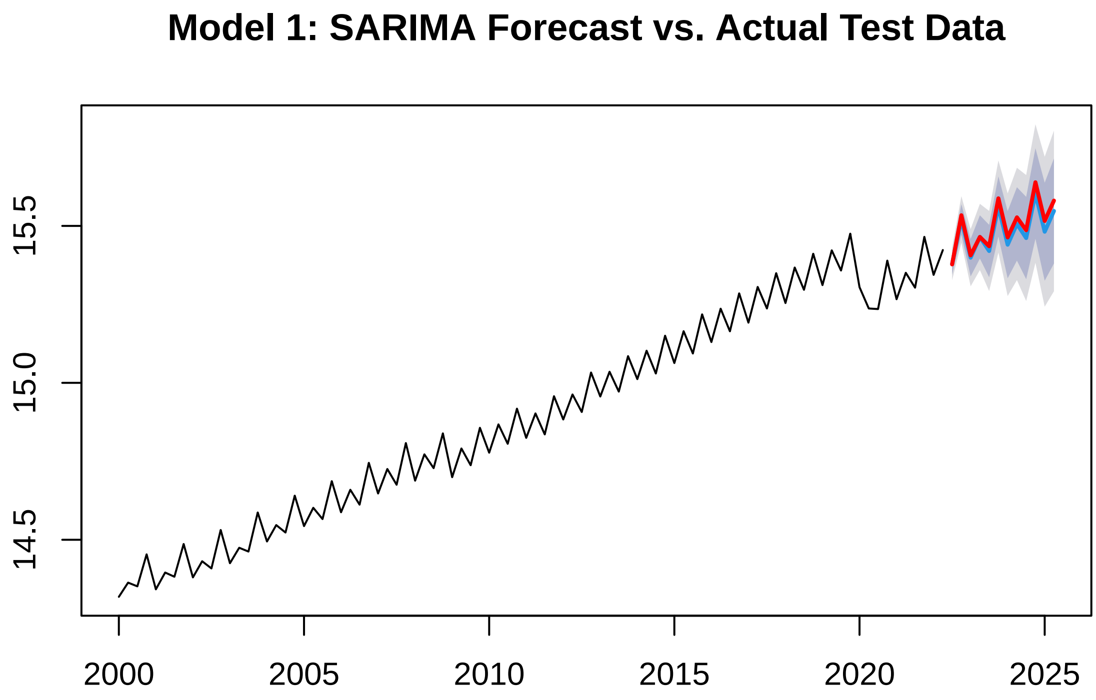



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(12,1,0)(0,0,1)[4] with drift
## Q* = 3.8068, df = 3, p-value = 0.2831
##
## Model df: 13.   Total lags used: 16
```

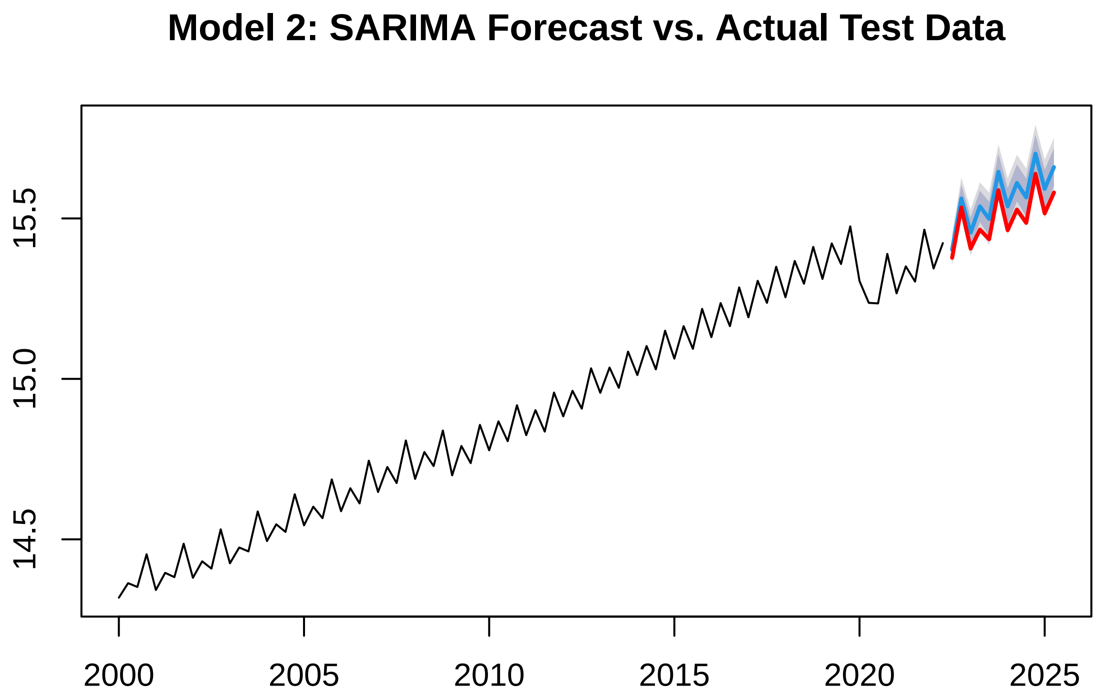
```
forecast_result1 <- forecast(F3, h = test_h)
forecast_result2 <- forecast(F4, h = test_h)
forecast_result3 <- forecast(F5, h = test_h)
```

```
## Warning in predict.Arima(object, n.ahead = h, newxreg = xreg): seasonal MA part
## of model is not invertible
```

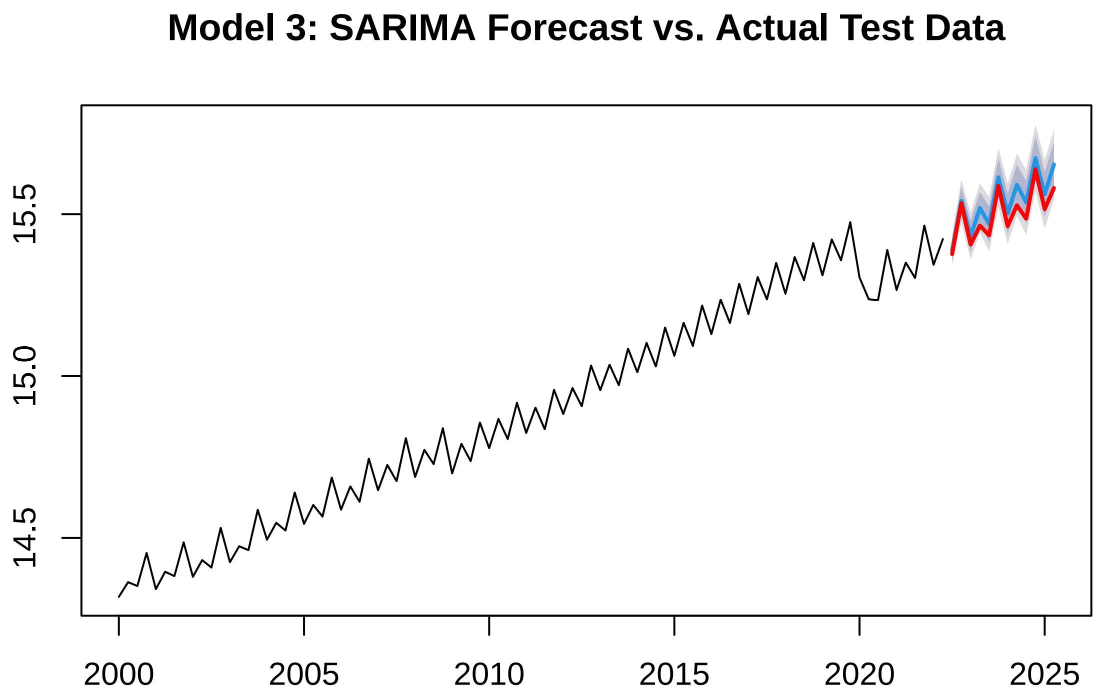
```
plot(forecast_result1, main = "Model 1: SARIMA Forecast vs. Actual Test Data")
lines(y_test, col = "red", lwd = 2)
```




```r
plot(forecast_result2, main = "Model 2: SARIMA Forecast vs. Actual Test Data")
lines(y_test, col = "red", lwd = 2)
```




```r
plot(forecast_result3, main = "Model 3: SARIMA Forecast vs. Actual Test Data")
lines(y_test, col = "red", lwd = 2)
```



4. **Based on the results from item 3, what is the "best model?"**

Based on item 3, the "best model" is $SARIMA(1,1,0) \times (0,1,1)_4$. It has the best out-of-sample accuracy at all required horizons based on its RMSE. The RMSE of the two other competing models are much worse at one year (Model 3: 0.007705, Model 4: 0.047906, Model 5: 0.030875).

Although Model 4 has a slightly lower train AIC (Model 3: -376.43 vs Model 4: -372.34), its test errors are much larger, so we prioritize forecast performance.

TRAIN residuals for Model 3 pass Ljung-Box ($p>0.05$), indicating white-noise errors, so Box-Jenkins diagnostics are satisfied.

Therefore, we select $SARIMA(1,1,0) \times (0,1,1)_4$ for forecasting.