ECON 707/807: Econometrics II

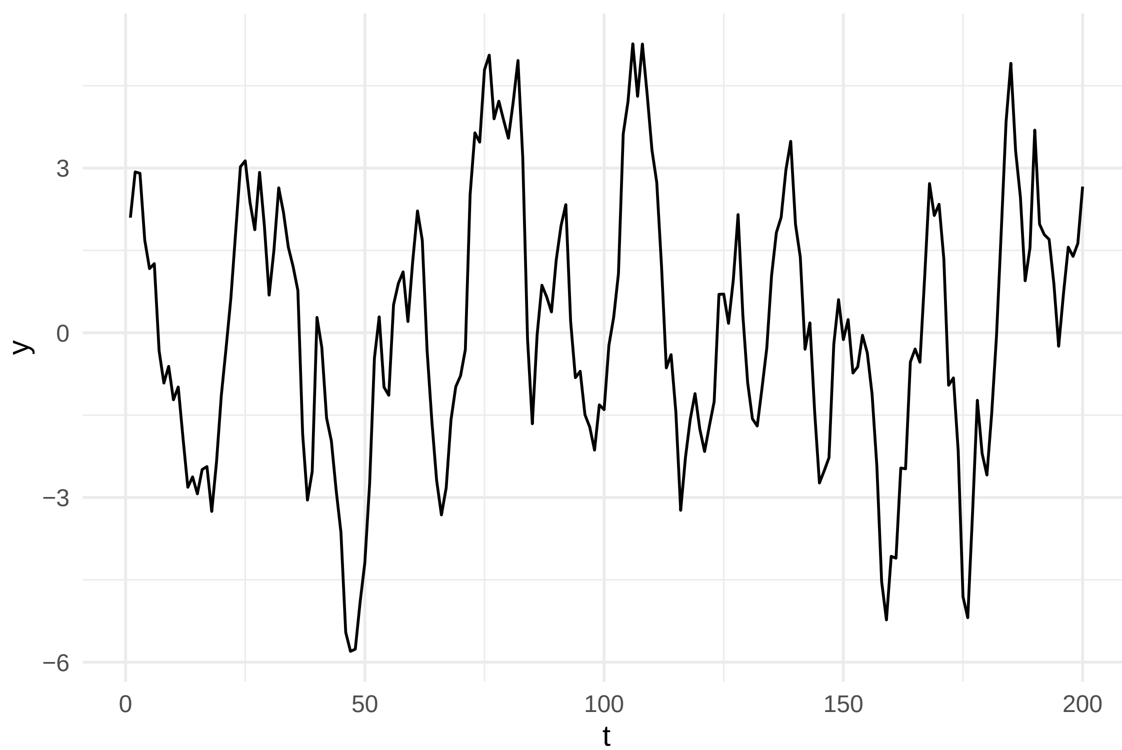
Homework 4: Answers
Fall Semester, 2023
Evie Zhang
s2zhang@odu.edu

ARMA Model

In this question. You are going to practice fitting different types of ARMA model and forecast based on the model you choose. Download the file “sim_arma_data.csv” from Canvas.

1. Fit models where you should consider all combinations of p = 0, 1, 2 and q = 0, 1. Report, in a single table, the following information for each of these models: parameter estimates, standard errors, t-values, p-values, as well as R squares.

```
data <- read.csv("../data/sim_arma_data.csv", header = FALSE) %>%
  rename(y = V1)

data = data %>% mutate(t = c(1:200))

ggplot(data, aes(x = t, y = y)) +
  geom_line() +
  theme_minimal()
```



```r
# There are 5 model combinations ARMA(0,1), ARMA(1,0), ARMA(1,1), ARMA(2,0), ARMA(2,1)


arma01 <- arima(data$y, c(0, 0, 1), include.mean = F)
arma10 <- arima(data$y, c(1, 0, 0), include.mean = F)
arma11 <- arima(data$y, c(1, 0, 1), include.mean = F)
arma20 <- arima(data$y, c(2, 0, 0), include.mean = F)
arma21 <- arima(data$y, c(2, 0, 1), include.mean = F)


stargazer(arma01, arma10, arma11, arma20, arma21, type = "text")
```

```
##
## =================================================================
##                                 Dependent variable:
##                    ----------------------------------------------
##
##                      (1)      (2)      (3)       (4)      (5)
## -----------------------------------------------------------------
## ma1                0.857***          0.450***           0.316*
##                    (0.028)           (0.067)           (0.190)
##
## ar1                         0.893*** 0.818*** 1.241***  0.970***
##                             (0.031)  (0.043)   (0.065)  (0.192)
##
## ar2                                          -0.390***  -0.146
##                                               (0.065)  (0.178)
##
## -----------------------------------------------------------------
## Observations         200      200      200       200      200
## Log Likelihood     -361.083 -301.865 -284.855 -285.435  -284.514
## sigma2              2.152    1.189    1.001     1.007    0.997
## Akaike Inf. Crit. 726.165  607.731  575.710   576.870  577.028
## =================================================================
## Note:                                  *p<0.1; **p<0.05; ***p<0.01
```

2. Which ARMA(p,q) model is the best fitted model?

```r
# Estimate AIC and BIC for the model
aic01 <- AIC(arma01)
bic01 <- BIC(arma01)

aic10 <- AIC(arma10)
bic10 <- BIC(arma10)

aic11 <- AIC(arma11)
bic11 <- BIC(arma11)

aic20 <- AIC(arma20)
bic20 <- BIC(arma20)

aic21 <- AIC(arma21)
bic21 <- BIC(arma21)

# display in the same table
```

```r
results <- tibble(
  Model = c("ARMA(0,1)", "ARMA(1,0)", "ARMA(1,1)", "ARMA(2,0)", "ARMA(2,1)"),
  AIC = c(aic01, aic10, aic11, aic20, aic21),
  BIC = c(bic01, bic10, bic11, bic20, bic21)
)

print(results)
```

```
## # A tibble: 5 x 3
##   Model       AIC   BIC
##   <chr>     <dbl> <dbl>
## 1 ARMA(0,1)  726.  733.
## 2 ARMA(1,0)  608.  614.
## 3 ARMA(1,1)  576.  586.
## 4 ARMA(2,0)  577.  587.
## 5 ARMA(2,1)  577.  590.
```

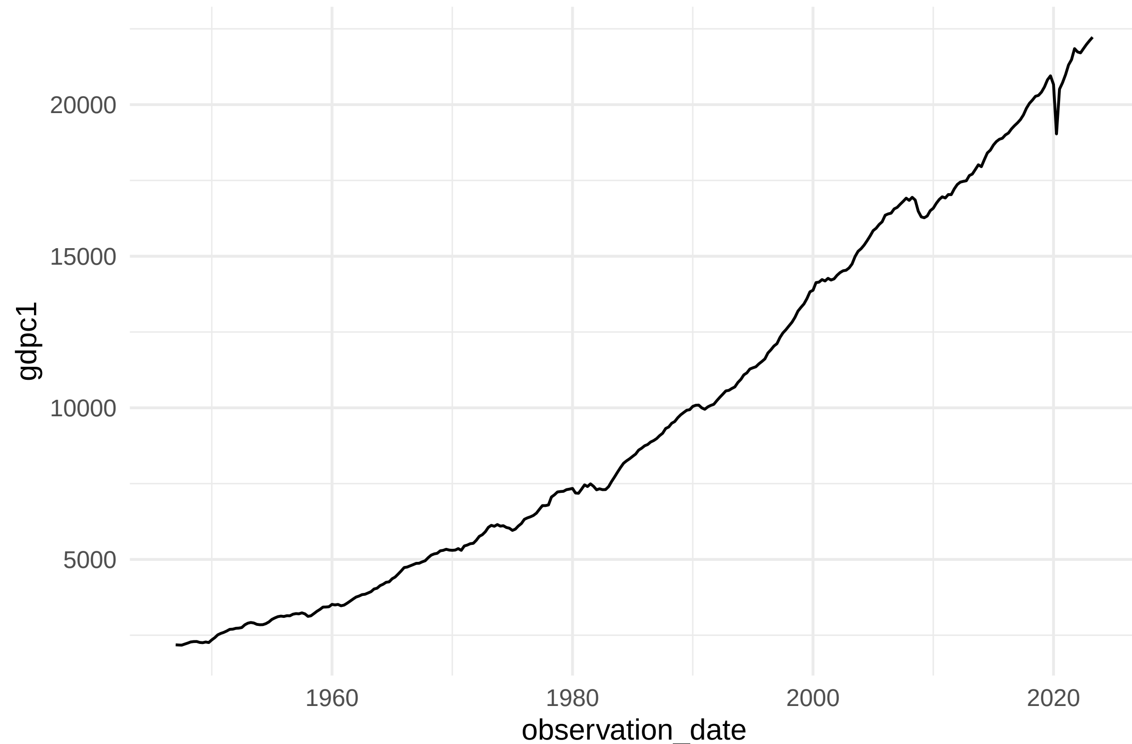
```r
## looks like ARMA(1,1) model has the lowest AIC and BIC. So it is the best fitted model.
```

U.S. real GDP

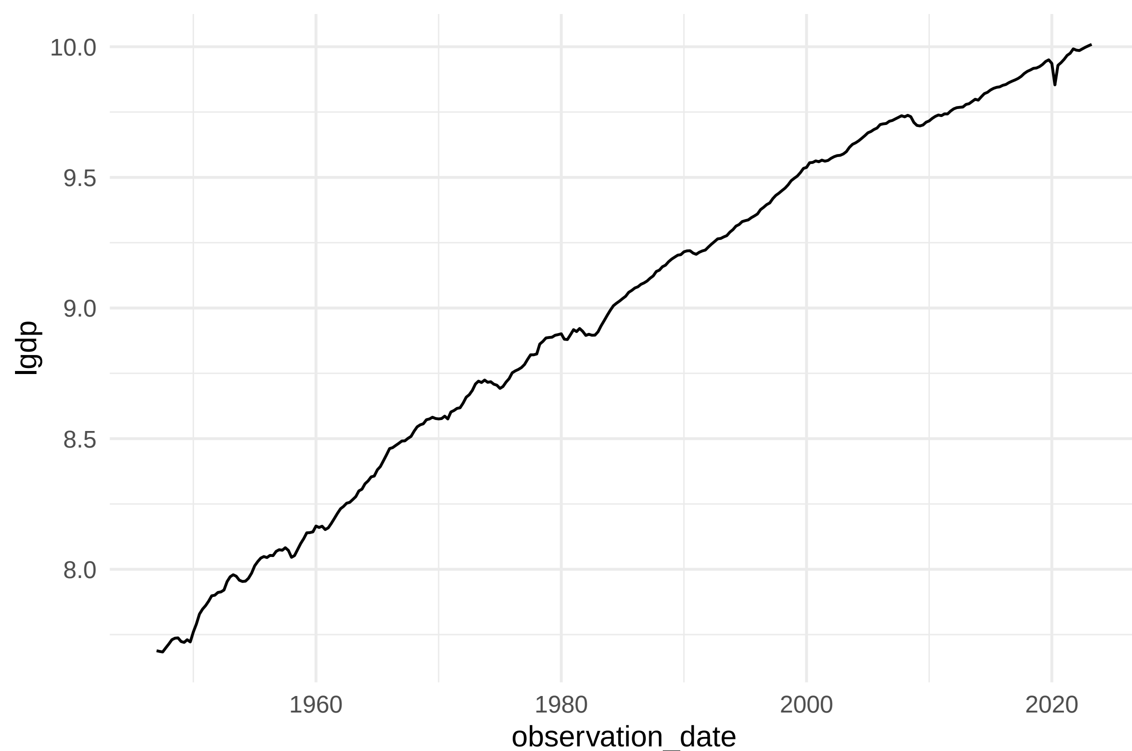
Create a R dataset holding U.S. quarterly real GDP data from 1947:Q1 to the most recent data available (currently 2020:Q2). To get an Excel spreadsheet holding the GDP data, go to the Saint Louis Federal Reserve Bank "FRED" website: https://fred.stlouisfed.org Search on "GDPC1" in the search box at the top of the page. You will be taken to the page for the quarterly real GDP series. Click on "Download" and then "Excel (Data)." Save this file on your computer. The file will be in Microsoft Excel format, and should contain data on U.S. quarterly real GDP from the first quarter of 1947 to the latest quarter available.

1. Plot the real GDP series over the entire sample period. Note that the series has a somewhat exponential shape to it. Transform the series by taking natural logarithms.

```r
gdp = read_excel("../data/GDPC1.xls", skip = 10) %>% clean_names()


ggplot(gdp, aes(x = observation_date, y = gdpc1)) +
  geom_line() +
  theme_minimal()
```



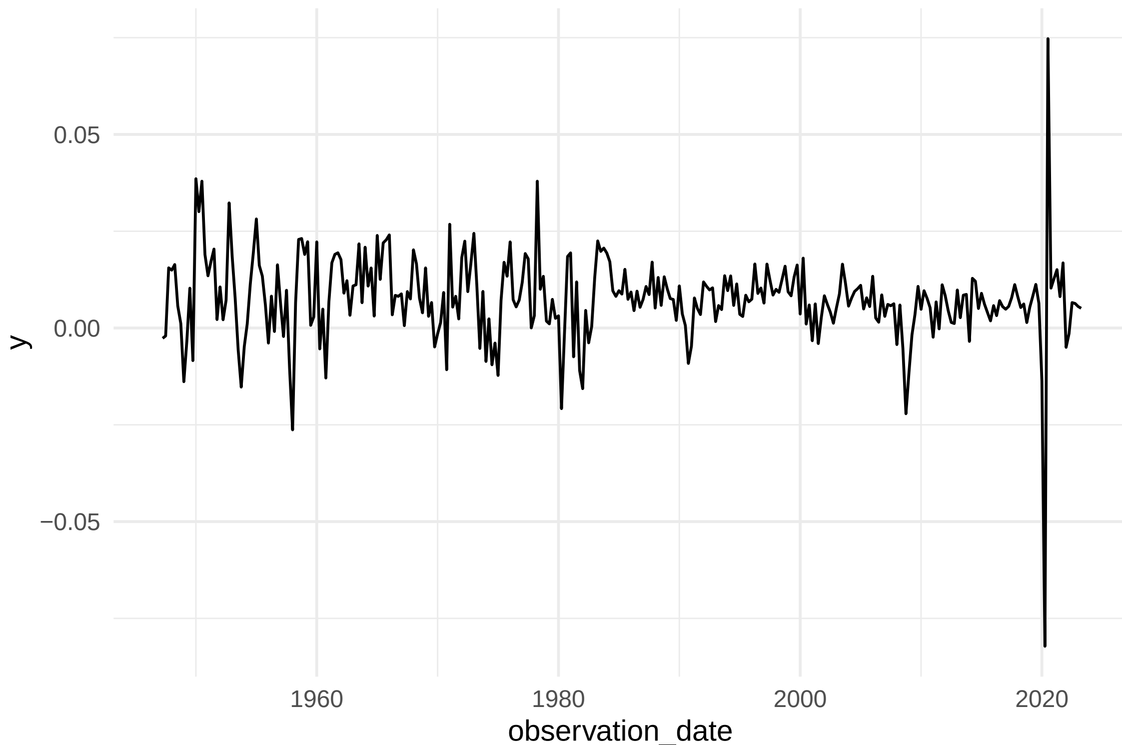

```
gdp = gdp %>% mutate(lgdp = log(gdpc1))

ggplot(gdp, aes(x = observation_date, y = lgdp)) +
  geom_line() +
  theme_minimal()
```



2. Compute the first difference of the log level of real GDP, multiplied by 100, and name this variable y. Note that the first difference of the natural log of a variable is approximately equal to the growth rate of the variable. Plot a line graph of y. Does the mean of this series appear stationary? What about the variance?

```r
# create a lag and take first difference
gdp = gdp %>% mutate(lgdp_lag = lag(lgdp)) %>%
  mutate(y = lgdp - lgdp_lag)

ggplot(gdp, aes(x = observation_date, y = y)) +
  geom_line() +
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```




```r
# After taking the first difference, the mean looks more stationary, the variance looks mostly stationa
```

3. Choose your own model to fit based on raw trend, adf test, acf and pacf. Estimate the model for dlgdp. Report the parameter estimates, standard errors, and AIC.

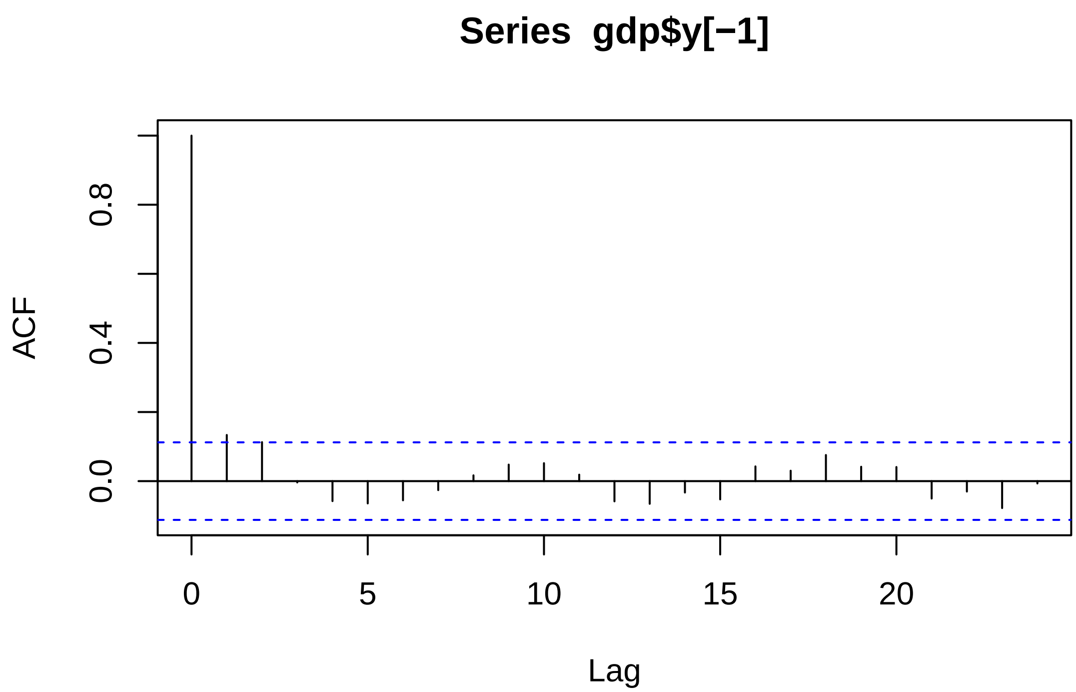
```r
# adf test for unit root
tseries::adf.test(gdp$lgdp)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gdp$lgdp
## Dickey-Fuller = -1.1189, Lag order = 6, p-value = 0.9183
## alternative hypothesis: stationary
```

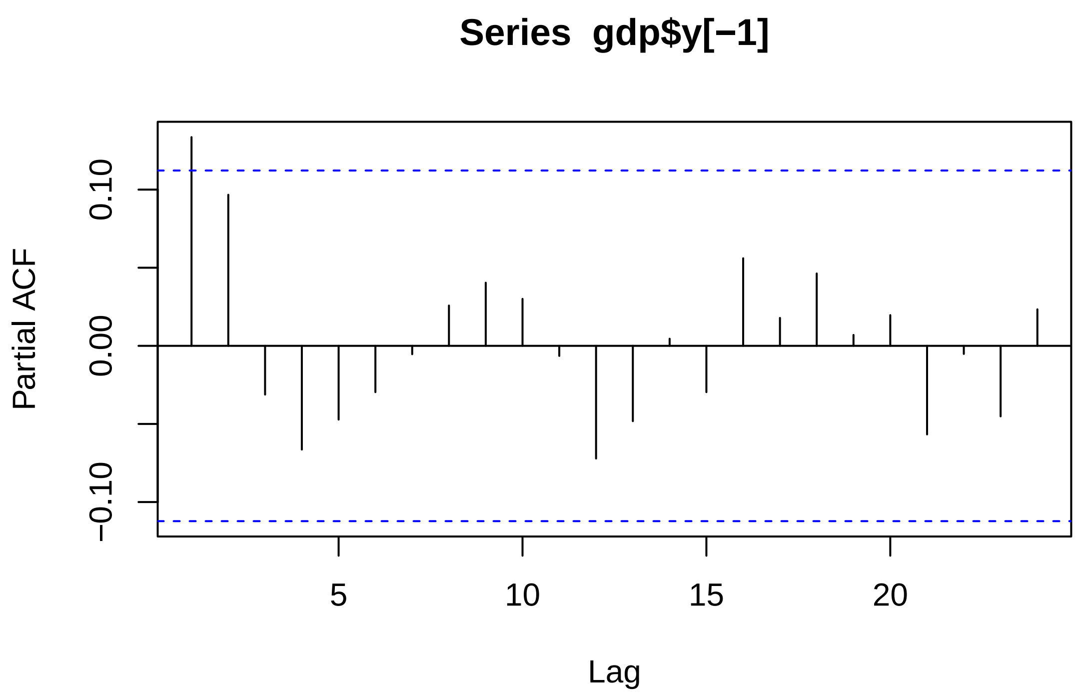
```r
# failed to reject null hypothesis, the time series (log of GDP) is non-stationary

tseries::adf.test(gdp$y[-1])
```

```
## Warning in tseries::adf.test(gdp$y[-1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  gdp$y[-1]
## Dickey-Fuller = -7.3646, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
# reject the null hypothesis, the time series (first difference of log GDP) is sationary

acf(gdp$y[-1])
```




```
pacf(gdp$y[-1])
```




```
# based on acf and pacf, I would choose a AR(1) model

ar1 <- arima(gdp$y, c(1, 0, 0), include.mean = F)
ar1
```

```
##
## Call:
## arima(x = gdp$y, order = c(1, 0, 0), include.mean = F)
##
## Coefficients:
##          ar1
##       0.4059
## s.e.  0.0522
##
## sigma^2 estimated as 0.0001535:  log likelihood = 906.4,  aic = -1808.79
```

4. Use auto.arima() to select the model for dlgdp. Report the parameter estimates, standard errors, AIC, and BIC.

```
reg <- auto.arima(gdp$y)
reg
```

```
## Series: gdp$y
## ARIMA(3,1,2)
##
## Coefficients:
##          ar1     ar2      ar3      ma1     ma2
##       0.8986  0.0074  -0.1334  -1.7745  0.7782
## s.e.  0.1837  0.0797   0.0573   0.1775  0.1762
##
## sigma^2 = 0.0001243:  log likelihood = 936.28
## AIC=-1860.55   AICc=-1860.27   BIC=-1838.25
```

5. Which model, either your selection or the one chosen using auto.arima, demonstrates a superior fit to

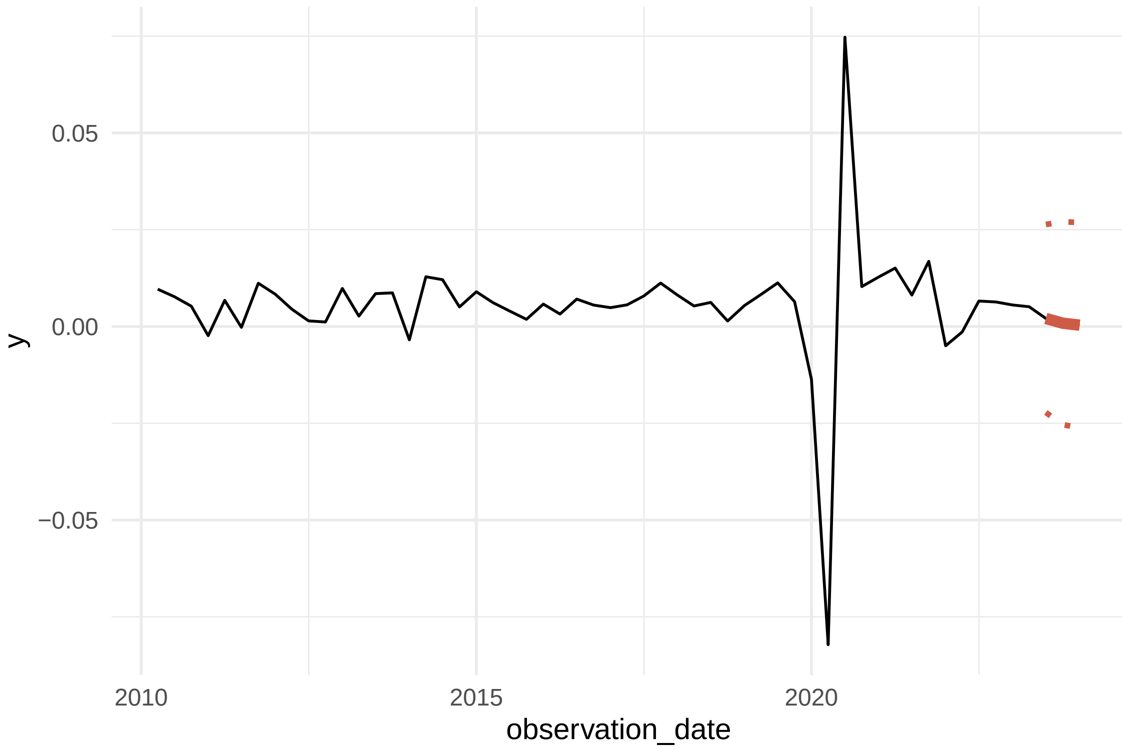
the time series?

Since my own model has the lower AIC value (-1808.79), it is considered to be a better fit for the data compared to auto.arima chosed model. Therefore, AR(1) is the preferred model based on the AIC criterion. Lower AIC values indicate a better balance between goodness of fit and model complexity, so AR(1) is more parsimonious while still providing a good fit to the data.

6. Using the estimated model for dlgdp, produce forecast estimates and prediction intervals of dlgdp for the next three quarters. Show the forecast values along with their corresponding 95% prediction intervals, and the original data (starting from January 1, 2010) within a single graph.

```
# Number of quarters to forecast
n_forecast <- 3

forecast_result <- forecast(ar1, h = n_forecast)

# Forecasted values for the next three quarters
forecast_values <- forecast_result$mean

# Prediction intervals for the next three quarters
lower_bounds <- forecast_result$lower[,2]
upper_bounds <- forecast_result$upper[,2]

forecast = data.frame(
  observation_date = as.Date(c("2023-07-01", "2023-10-01", "2024-01-01")),
  y = as.vector(forecast_result$mean),
  upper_pi = as.vector(forecast_result$upper[,2]),
  lower_pi = as.vector(forecast_result$lower[,2]))

df = gdp[, c(1,5)] %>%
  mutate(upper_pi = NA,
         lower_pi = NA)

df = rbind(df, forecast)

ggplot(data = df %>%
              filter(observation_date > "2010-01-01")) +
  geom_line(aes(x = observation_date, y = y), color = "black")+
  geom_line(data = df %>%
              filter(observation_date > "2023-04-01"),
                     aes(x = observation_date, y = y), color = "coral3", size = 2)+
    geom_line(data = df %>%
              filter(observation_date > "2023-04-01"),
                     aes(x = observation_date, y = upper_pi),
              color = "coral3", size =1, linetype = "dotted")+
    geom_line(data = df %>%
              filter(observation_date > "2023-04-01"),
                     aes(x = observation_date, y = lower_pi),
              color = "coral3", size = 1, linetype = "dotted")+
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


0.05
0.00
−0.05
y
2010
2015
2020
observation_date