ECON 475 Final Project

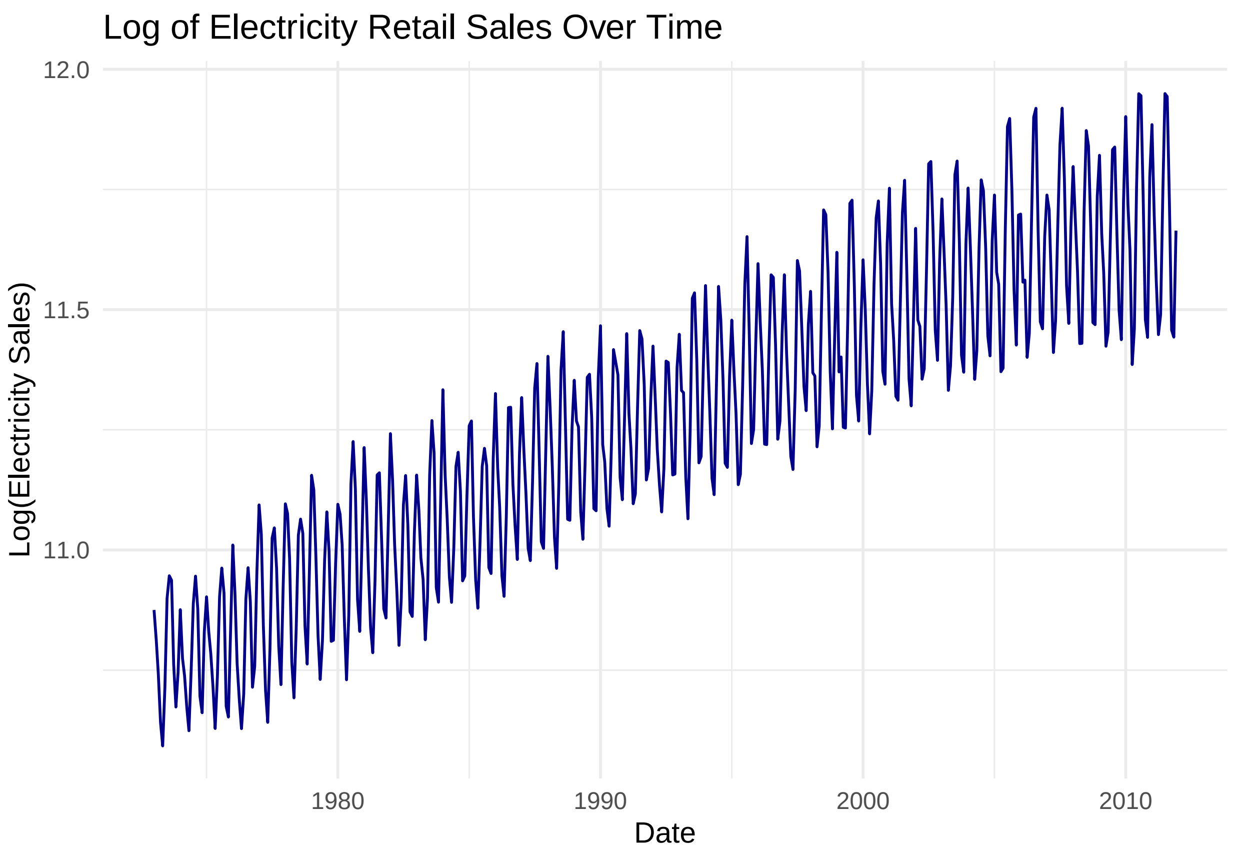
Ramsey EL Lethy

2025-05-07

Question 1. Use the dataset elec.csv that is available on Canvas. The variable elec is electricity retail sales to the residential sector in the United States in million of kilowatt hours. The sample is at monthly frequency and covers the period from January 1973 and December 2011. Define the training sample as all information available up to December 2010.

(a) Let yt denote the log of electricity retail sales. Plot yt over time. Is there a trend?

```r
library(ggplot2)
library(dplyr)
library(readr)

elec_data <- read_csv("elec.csv")

elec_data$log_elec <- log(elec_data$elec)

elec_data <- elec_data %>%
  mutate(
    date = as.Date(paste0(substr(date, 1, 4), "-", substr(date, 6, 7), "-01"))
  )


View(elec_data)

ggplot(elec_data, aes(x = date, y = log_elec)) +
  geom_line(color = "darkblue", group = 1) +
  labs(
    title = "Log of Electricity Retail Sales Over Time",
    x = "Date",
    y = "Log(Electricity Sales)"
  ) +
  theme_minimal()
```



It looks like the data has an upward trend over time.

(b) Estimate a model with a linear and a quadratic trend. Which one would you choose? From now on, use the trend model you chose in part (b)

```r
elec_data <- elec_data %>%
  mutate(
    t = 1:n(),                   # time trend
    t2 = t^2                    # quadratic term
  )

# Linear trend model
linear_model <- lm(log_elec ~ t, data = elec_data)

# Quadratic trend model
quad_model <- lm(log_elec ~ t + t2, data = elec_data)
```

```r
# Compare models
summary(linear_model)
```

```
##
## Call:
## lm(formula = log_elec ~ t, data = elec_data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32867 -0.13043  0.00826  0.11619  0.30463
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.078e+01  1.364e-02  790.33   <2e-16 ***
## t           2.086e-03  5.040e-05   41.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1473 on 466 degrees of freedom
## Multiple R-squared:  0.7861, Adjusted R-squared:  0.7856
## F-statistic:  1712 on 1 and 466 DF,  p-value: < 2.2e-16
```

```
summary(quad_model)
```

```
##
## Call:
## lm(formula = log_elec ~ t + t2, data = elec_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.293908 -0.138007  0.004567  0.117660  0.309623
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.073e+01  2.033e-02 527.934  < 2e-16 ***
## t            2.682e-03  2.002e-04  13.398  < 2e-16 ***
## t2          -1.272e-06  4.134e-07  -3.078  0.00221 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.146 on 465 degrees of freedom
## Multiple R-squared:  0.7904, Adjusted R-squared:  0.7895
## F-statistic: 876.5 on 2 and 465 DF,  p-value: < 2.2e-16
```

```
# AIC comparison
(AIC(linear_model, quad_model))
```

```
##              df       AIC
## linear_model  3 -460.5804
## quad_model    4 -468.0167
```

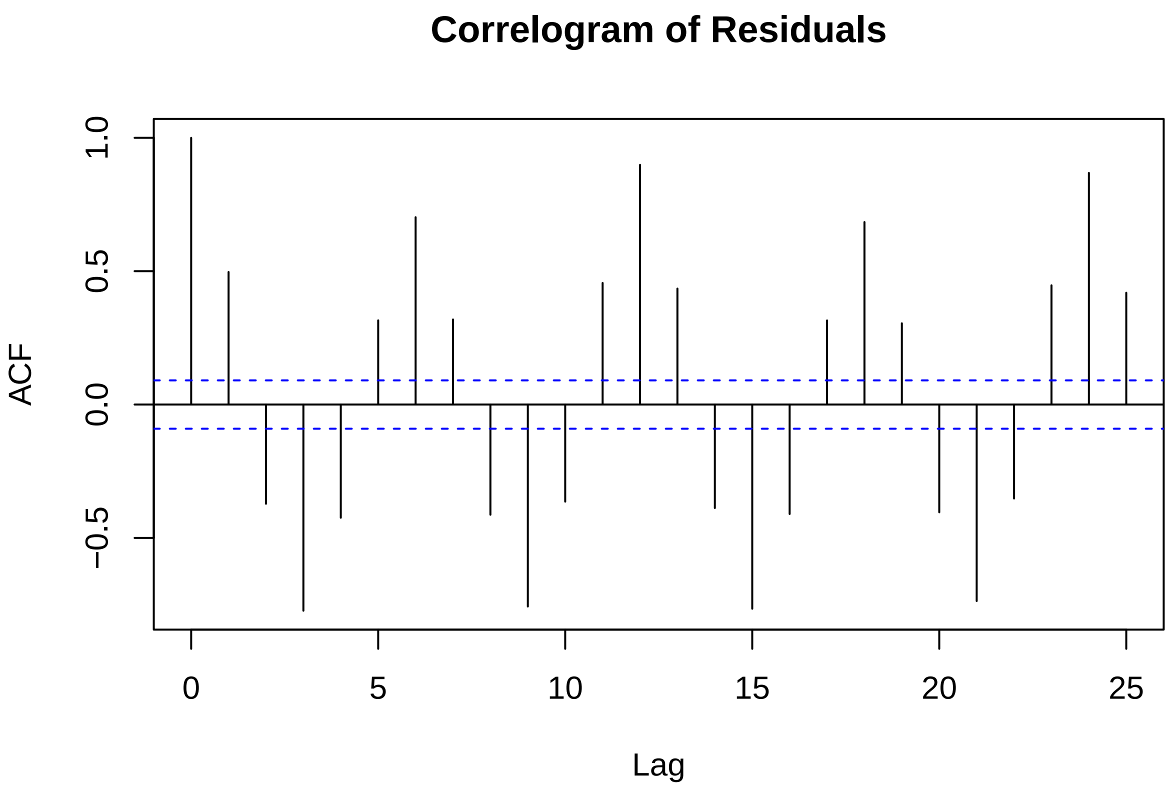
```
(BIC(linear_model, quad_model))
```

```
##              df       BIC
## linear_model  3 -448.1350
## quad_model    4 -451.4229
```

We go with the quad model

(c) Provide a plot of the correlogram (up to 25 lags) of the residuals of the model you chose in part (b). Is there any evidence of seasonal patterns in the correlogram? Explain your answer.

```r
resid_trend <- residuals(quad_model)  # or linear_model

# Plot ACF up to lag 25
acf(resid_trend, lag.max = 25, main = "Correlogram of Residuals")
```



There is definitely seasonality, notice how the positive and negatively related lags occur in even intervals, that the same relationship between even interval lags indicates seasonality in the data.

(d) Estimate a model with a trend and a full set of dummy variables, and report the results.

```r
#will create a column indicating each month
elec_data$month <- factor(format(elec_data$date, "%m"))

# Create time trend variable
elec_data$trend <- 1:nrow(elec_data)

# Estimate model: log(elec) ~ trend + monthly dummies
seasonal_model <- lm(log(elec) ~ trend + month, data = elec_data)

# View summary results
summary(seasonal_model)
```

```
##
## Call:
## lm(formula = log(elec) ~ trend + month, data = elec_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.167425 -0.033858 -0.000269  0.037078  0.137952
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 10.9558755  0.0100012 1095.458  < 2e-16 ***
## trend        0.0020890  0.0000192  108.799  < 2e-16 ***
## month02     -0.1234758  0.0127036   -9.720  < 2e-16 ***
## month03     -0.2122879  0.0127037  -16.711  < 2e-16 ***
## month04     -0.3482388  0.0127037  -27.412  < 2e-16 ***
## month05     -0.3724428  0.0127038  -29.317  < 2e-16 ***
## month06     -0.2013440  0.0127040  -15.849  < 2e-16 ***
## month07     -0.0021043  0.0127041   -0.166   0.8685
## month08      0.0218075  0.0127043    1.717   0.0867 .
## month09     -0.1060599  0.0127046   -8.348  8.4e-16 ***
## month10     -0.3046616  0.0127048  -23.980  < 2e-16 ***
## month11     -0.3374595  0.0127051  -26.561  < 2e-16 ***
## month12     -0.1345384  0.0127054  -10.589  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0561 on 455 degrees of freedom
## Multiple R-squared:  0.9697, Adjusted R-squared:  0.9689
## F-statistic:  1214 on 12 and 455 DF,  p-value: < 2.2e-16
```

```r
AIC(seasonal_model)
```

```
## [1] -1353.356
```

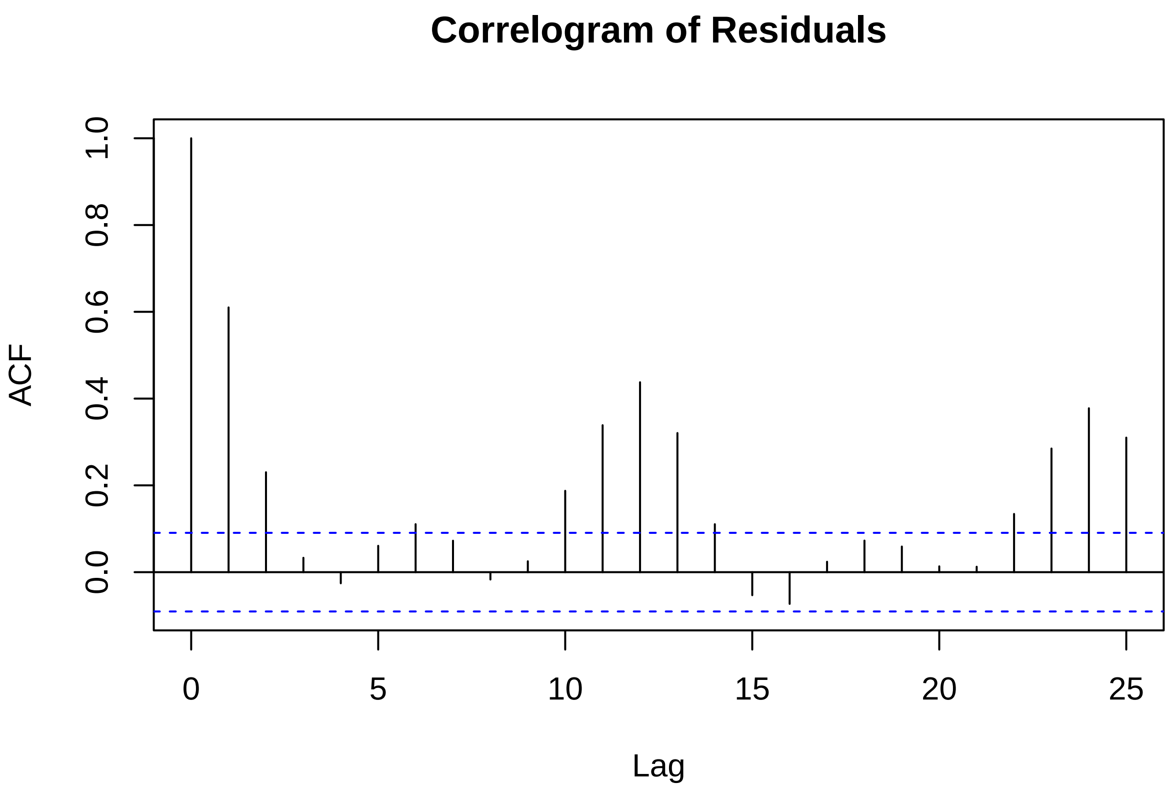
```r
BIC(seasonal_model)
```

```
## [1] -1295.278
```

Our model is much more fit to the data with a 96% model fit, it looks like most of that is apart of seasonality, the only non significant factor in our model is month 7 and 8.

(e) Provide a plot of the correlogram (up to 25 lags) of the residuals of the model you chose in part (d). Is there any evidence of cyclical component or serial correlation in the residuals? Explain your answer.

```r
resid_trend_seasonal <- residuals(seasonal_model)

acf(resid_trend_seasonal, lag.max = 25, main = "Correlogram of Residuals")
```



it looks like after plotting seasonality and trends, there is still some correlated lags, ideally, if we captured all the effects of the model, the plot of the residuals should just be a white noise process. But this correlogram still has a pretty significant relationship between the lags, which tells us that there is some cyclical component to be included in our model.

(f) Estimate an ARMA(p,q) model with p = 0, 1, 2, 3 and q = 0, 1, 2, 3, except for p = q = 0. Report the values of the BICs. Based on the correlogram and values of the BIC, which model would you choose? Explain your answer.

```
log_ts <- ts(elec_data$log_elec, start = c(1973, 1), frequency = 12)


bic_results <- data.frame(p = integer(), q = integer(), BIC = numeric())

bic_results
```

```
## [1] p   q   BIC
## <0 rows> (or 0-length row.names)
```

```
# Fit ARMA models and store BIC
for (p in 0:3) {
  for (q in 0:3) {
    if (p == 0 & q == 0) next
    model <- tryCatch(
      arima(log_ts, order = c(p, 0, q)),
      error = function(e) NULL
    )
    if (!is.null(model)) {
      #if we got a non null value for arma(p,0,q), store its bic into the df
      bic_results <- rbind(bic_results, data.frame(p = p, q = q, BIC = BIC(model)))
    }
  }
}
bic_results
```

```
##    p q        BIC
## 1  0 1  -231.4827
## 2  0 2  -524.0337
## 3  0 3  -613.0865
## 4  1 0  -477.9596
## 5  1 1  -666.3832
## 6  1 2  -683.0051
## 7  1 3  -822.2391
## 8  2 0  -572.7006
## 9  2 1  -670.3474
## 10 2 2  -677.6677
## 11 2 3  -694.4306
## 12 3 0  -823.4755
## 13 3 1 -1044.4332
## 14 3 2 -1024.7070
## 15 3 3 -1126.2405
```

```
# sort the results
bic_results <- bic_results[order(bic_results$BIC), ]
print(bic_results)
```

```
##    p q        BIC
## 15 3 3 -1126.2405
## 13 3 1 -1044.4332
## 14 3 2 -1024.7070
## 12 3 0  -823.4755
## 7  1 3  -822.2391
## 11 2 3  -694.4306
## 6  1 2  -683.0051
## 10 2 2  -677.6677
## 9  2 1  -670.3474
## 5  1 1  -666.3832
## 3  0 3  -613.0865
## 8  2 0  -572.7006
## 2  0 2  -524.0337
## 4  1 0  -477.9596
## 1  0 1  -231.4827
```

It looks like the Arima with the lowest BIC would be with 3 lags of yt and 3 lags of the error, so we will choose an ARMA(3,0,3) process for the rest of this section

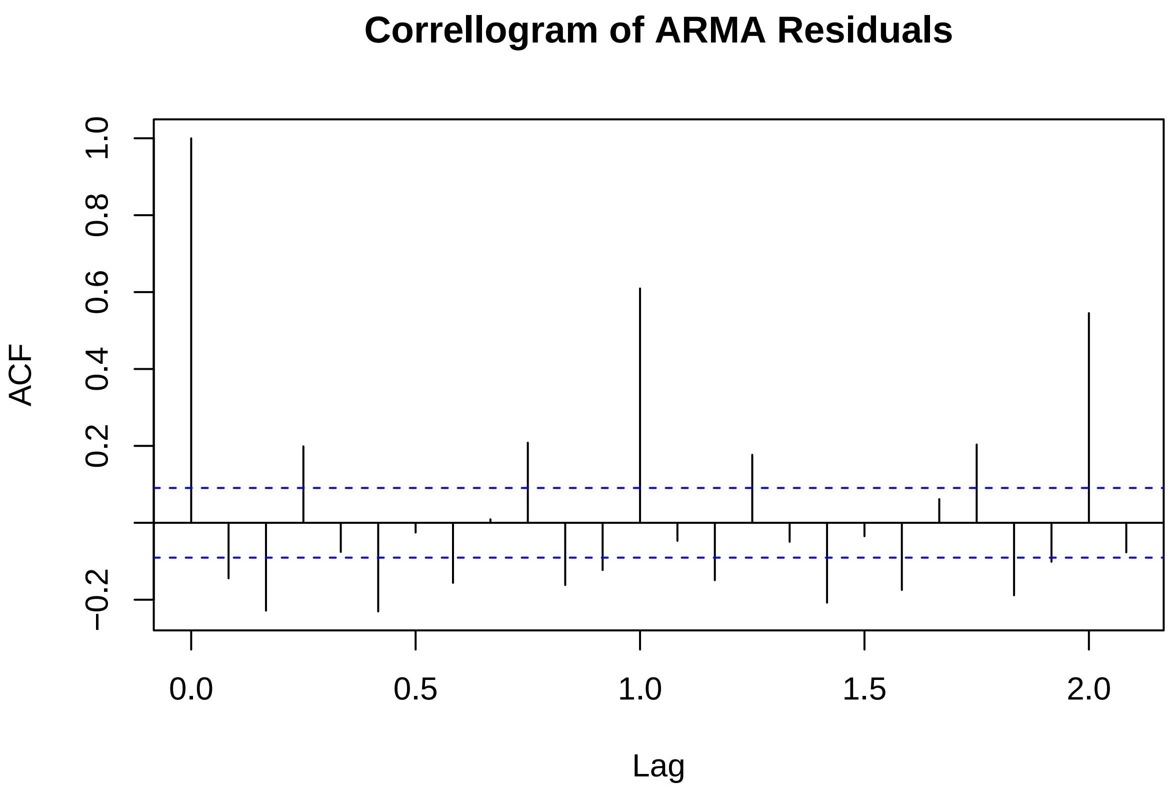
(g) Report the estimation results of the model you choose in part (f). In addition, provide a plot and the correlogram (up to 25 lags) of the residuals of the model. Is there any evidence of cyclical component or serial correlation? Explain your answer

```r
library(stats)
log_ts <- ts(elec_data$log_elec, start = c(1973, 1), frequency = 12)

# Fit ARMA(3,0,3) model
arma_model <- arima(log_ts, order = c(3, 0, 3))

arma_model
```

```
##
## Call:
## arima(x = log_ts, order = c(3, 0, 3))
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2     ma3  intercept
##       1.9769  -1.9733  0.9763  -0.9262  0.7773  0.1392    11.2648
## s.e.  0.0128   0.0144  0.0132   0.0845  0.1173  0.0971     0.1419
##
## sigma^2 estimated as 0.004656:  log likelihood = 587.71,  aic = -1159.43
```

here is our very beautiful ARMA(3,3) summary on the log of electricity sales, it looks like each auto regressive lag is alternating in sign effect, but absolute magnitude of effect is decreasing, while there isn't a particular trend in the direction of the effect, but it also looks like the absolute magnitude is decreasing.

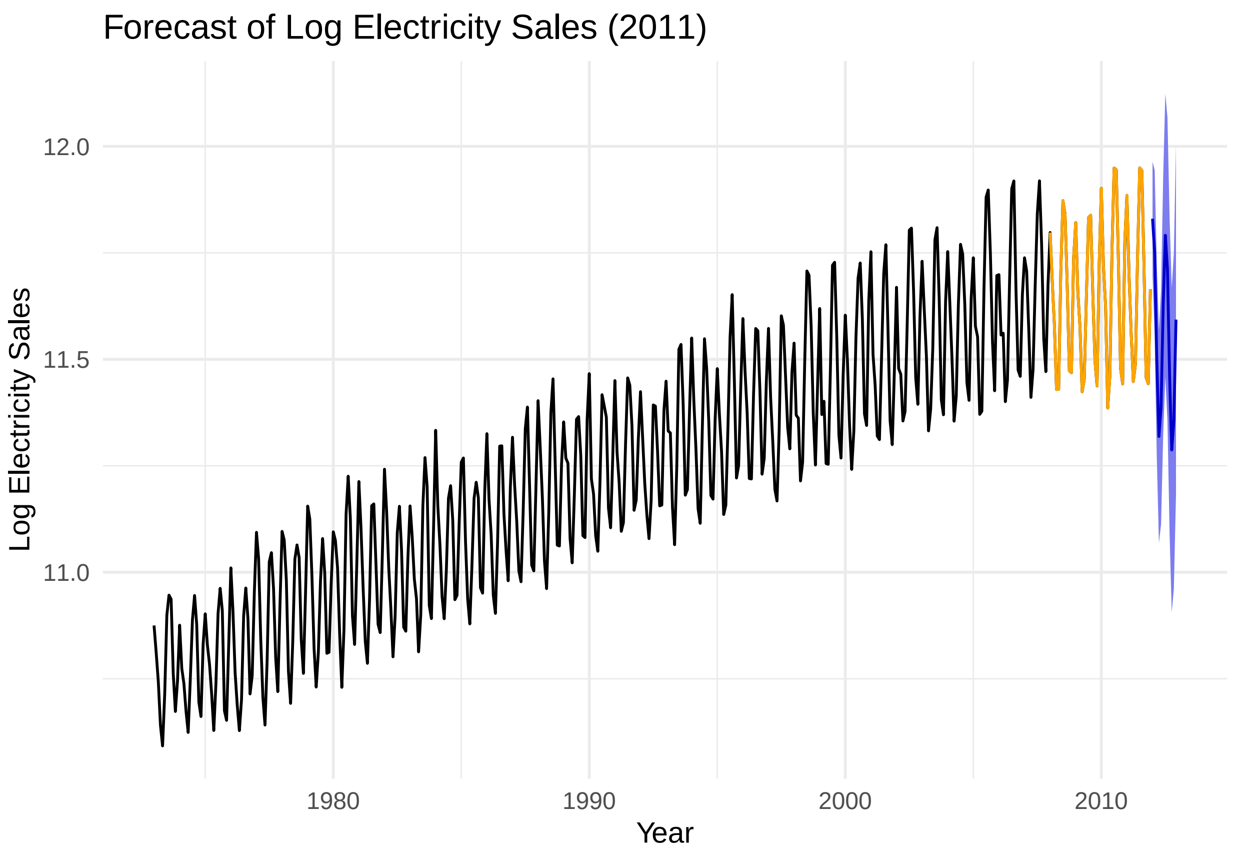
```r
residual_arma <- residuals(arma_model)

acf <- acf(residual_arma, lag.max = 25, main = "Correllogram of ARMA Residuals")
```



It looks like we've done a better job at making our residuals follow a white noise process, there is some high correlation at lag 12 and lag 24, but most of the residuals do fall above the threshold of a white noise process, so it does suggest that the residuals could be serially correlated.

(h) Use the model you chose in part (f) to forecast the log of the electricity retail sales for the year of 2011, and compute its 95% forecast interval as well. Plot your point and interval forecasts together with the actual data for the period from 2008 to 2011

```r
library(ggplot2)
library(forecast)
# Forecast for 12 months ahead (2011)
arma_forecast <- forecast(arma_model, h = 12, level = 95)

actual_ts <- window(log_ts, start = c(2008, 1), end = c(2011, 12))

# forecast with confidence intervals and actual data
autoplot(arma_forecast) +
  autolayer(actual_ts, series = "Actual", color = "orange") +
  labs(
    title = "Forecast of Log Electricity Sales (2011)",
    x = "Year",
    y = "Log Electricity Sales"
  ) +
  theme_minimal() +
  # <-- adjust this range as needed
  guides(colour = guide_legend(title = "Series"))
```



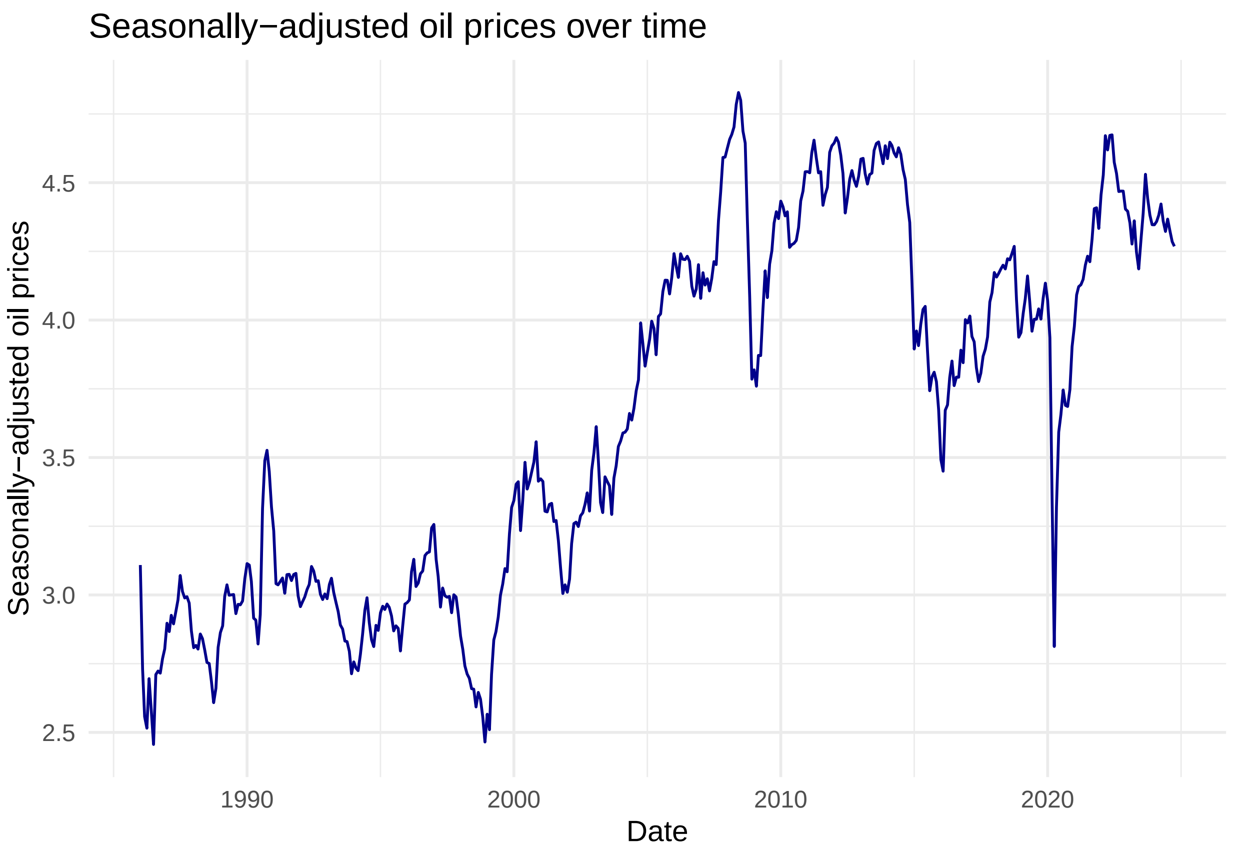
Question 2. Use the dataset wti_oil_price.csv that is available on Canvas. The variable oil_price is the seasonally-adjusted crude oil prices and is on dollars per barrel. The sample is at monthly frequency and covers the period from January 1986 to October 2024. Define the training sample as all information available up to 2022.

(a) Let yt denote the seasonally-adjusted oil prices. Plot yt over time. Does the data look stationary?

```
oil_data <- read_csv("wti_oil_price.csv")

oil_data
```

```
## # A tibble: 466 x 3
##      ...1 date       oil_price
##     <dbl> <date>         <dbl>
##  1      1 1986-01-01      22.4
##  2      2 1986-02-01      15.3
##  3      3 1986-03-01      12.9
##  4      4 1986-04-01      12.4
## 5       5 1986-05-01       14.8
## 6       6 1986-06-01       13.2
## 7       7 1986-07-01       11.7
## 8       8 1986-08-01       15.0
## 9       9 1986-09-01       15.2
## 10     10 1986-10-01       15.1
## # i 456 more rows
```

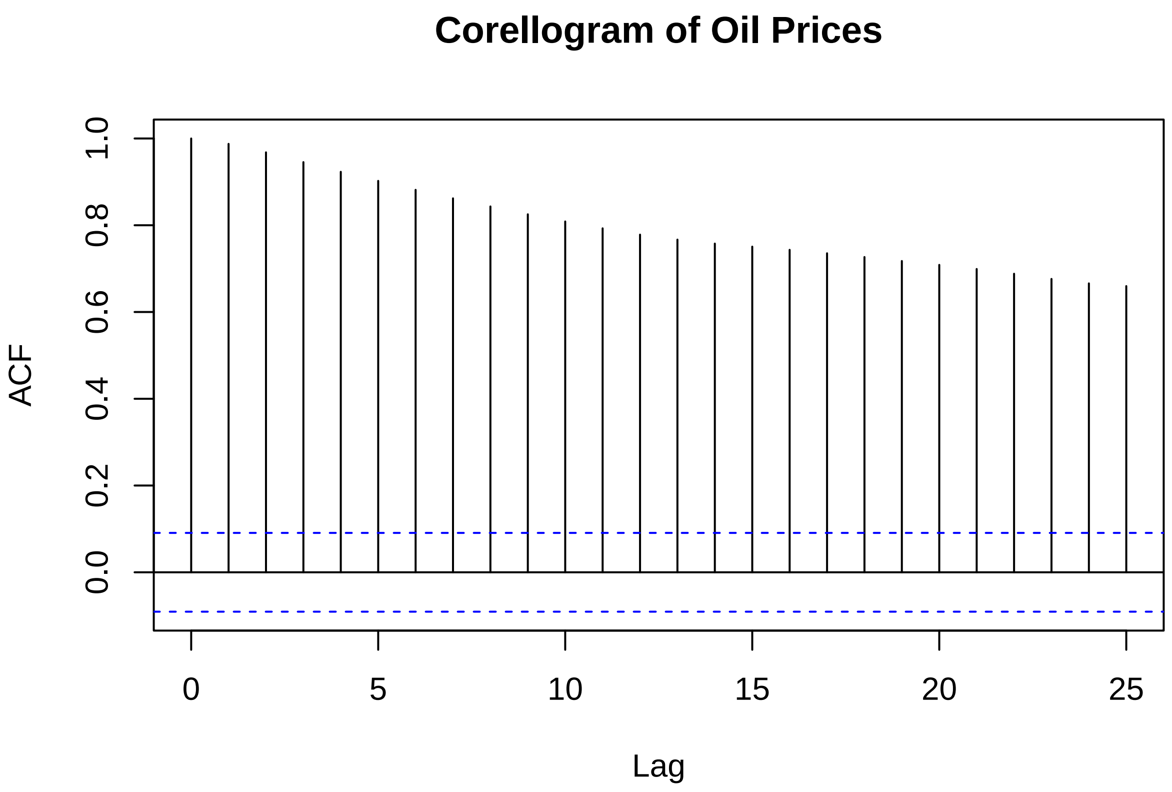
```
ggplot(oil_data, aes(x = date, y = log(oil_price))) +
  geom_line(color = "darkblue", group = 1) +
  labs(
    title = "Seasonally-adjusted oil prices over time",
    x = "Date",
    y = "Seasonally-adjusted oil prices"
  ) +
  theme_minimal()
```



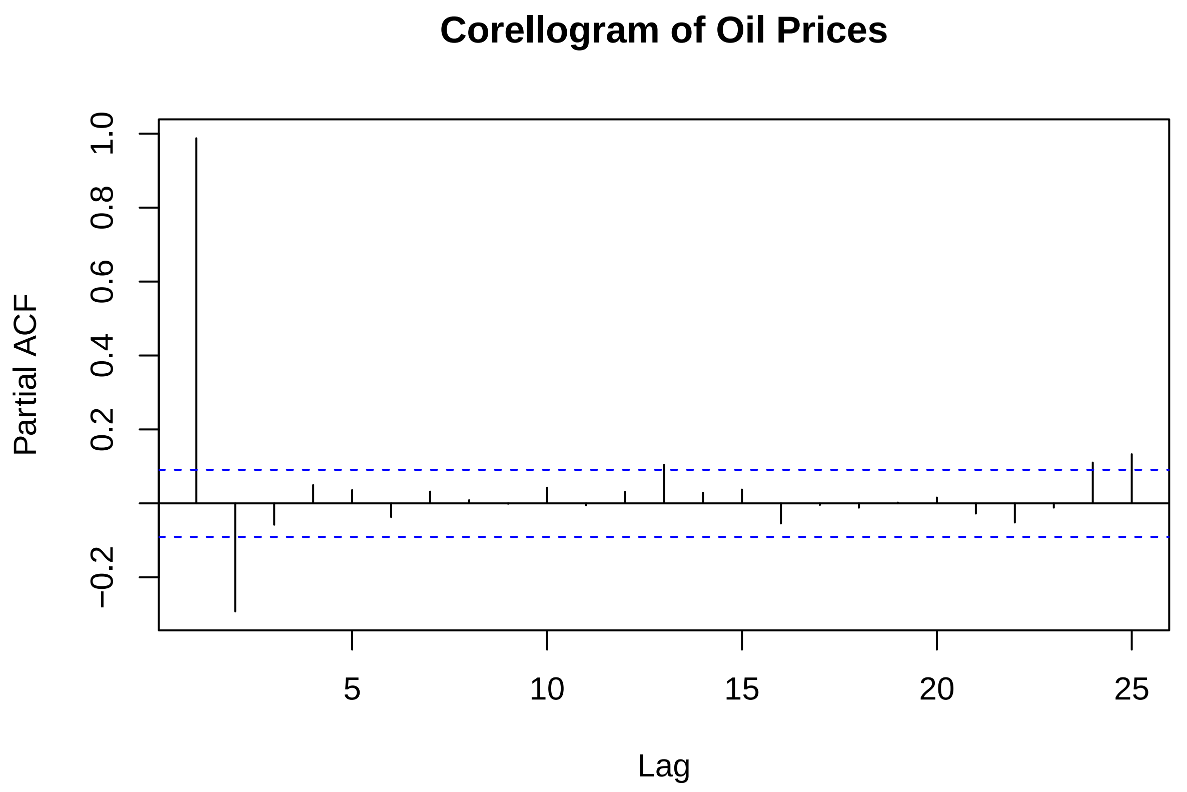
At a first glance, it looks like this process isn't really stationary, its good that there doesnt seem to be a consistent linear trend, but the variance shocks seem to be very random. Maybe this suggest that this process doesn't have very strong components, or there exists a unit root

(b) Plot both the ACF and the PACF of yt. What does the correlogram suggest about yt?

```r
acf <- acf(oil_data$oil_price, lag.max = 25, main = "Corellogram of Oil Prices")
```



```r
pacf <- pacf(oil_data$oil_price, lag.max = 25, main = "Corellogram of Oil Prices")
```



So the ACF function seems to be gradually decreasing, while the pacf cuts off immediately, after lag(1), this suggest that this process follows an Autoregressive process of lag(1) or AR(1). This is because for this time series process, the effect of all lags past lag 1 is completely indirect, so its captured in the ACF, but in teh PACF which measures the direct effect of all lags individually, we see that every lag past lag 1 has no direct effect.

(c) To support your answer in the previous question, perform a Dickey-Fuller test for the presence of a unit root. More specifically, run a regression for each of the following specifications. For each specification, compute the Dickey-Fuller test statistic.

```r
oil_data$log_oil = log(oil_data$oil_price)
attach(oil_data)
oil_data$log_oil = log(oil_price)
oil_data$diff_oil = c(NA, diff(log_oil))
lag_log_oil = lag(log_oil)
t = 1:nrow(oil_data)


df_model1 <- lm(diff_oil ~ 0 + lag_log_oil, data = oil_data)
summary(df_model1)
```

```
##
## Call:
## lm(formula = diff_oil ~ 0 + lag_log_oil, data = oil_data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56363 -0.04375  0.00401  0.05101  0.50519
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## lag_log_oil 0.0003628  0.0011460   0.317    0.752
##
## Residual standard error: 0.09173 on 464 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.000216,   Adjusted R-squared:  -0.001939
## F-statistic: 0.1002 on 1 and 464 DF,  p-value: 0.7517
```

```
df_model2 <- lm(diff_oil ~ lag_log_oil, data = oil_data)
summary(df_model2)
```

```
##
## Call:
## lm(formula = diff_oil ~ lag_log_oil, data = oil_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56636 -0.04484  0.00433  0.04828  0.49573
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.037175   0.023960   1.552    0.121
## lag_log_oil -0.009493   0.006455  -1.471    0.142
##
## Residual standard error: 0.09159 on 463 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.00465,    Adjusted R-squared:  0.0025
## F-statistic: 2.163 on 1 and 463 DF,  p-value: 0.142
```

```
df_model3 <- lm(diff_oil ~ lag_log_oil + t, data = oil_data)
summary(df_model3)
```

```
##
## Call:
## lm(formula = diff_oil ~ lag_log_oil + t, data = oil_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59354 -0.04443  0.00485  0.04996  0.45709
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.265e-02  3.152e-02   2.622  0.00904 **
## lag_log_oil -2.969e-02  1.118e-02  -2.655  0.00820 **
## t            1.210e-04  5.482e-05   2.207  0.02779 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.09121 on 462 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.01504,    Adjusted R-squared:  0.01077
## F-statistic: 3.527 on 2 and 462 DF,  p-value: 0.0302
```

In both models 1 and 2, we see that the dicky fuller test gave us an insignificant outcome, meaning that there is not ennough evidence to reject a unit root.

the last model has very slim margins of significance which probably isnt enough to reject the null hypothesis that there is a unit root too.

(d) Use the adf.test in R to perform the Dickey-Fuller test and compare your results with part (c). Do you obtain the same test statistic? Based on the p-values, do you have evidence to reject the null hypothesis of a unit root? From now on, consider the data in differences given by yt(in-differences) = yt - yt-1.

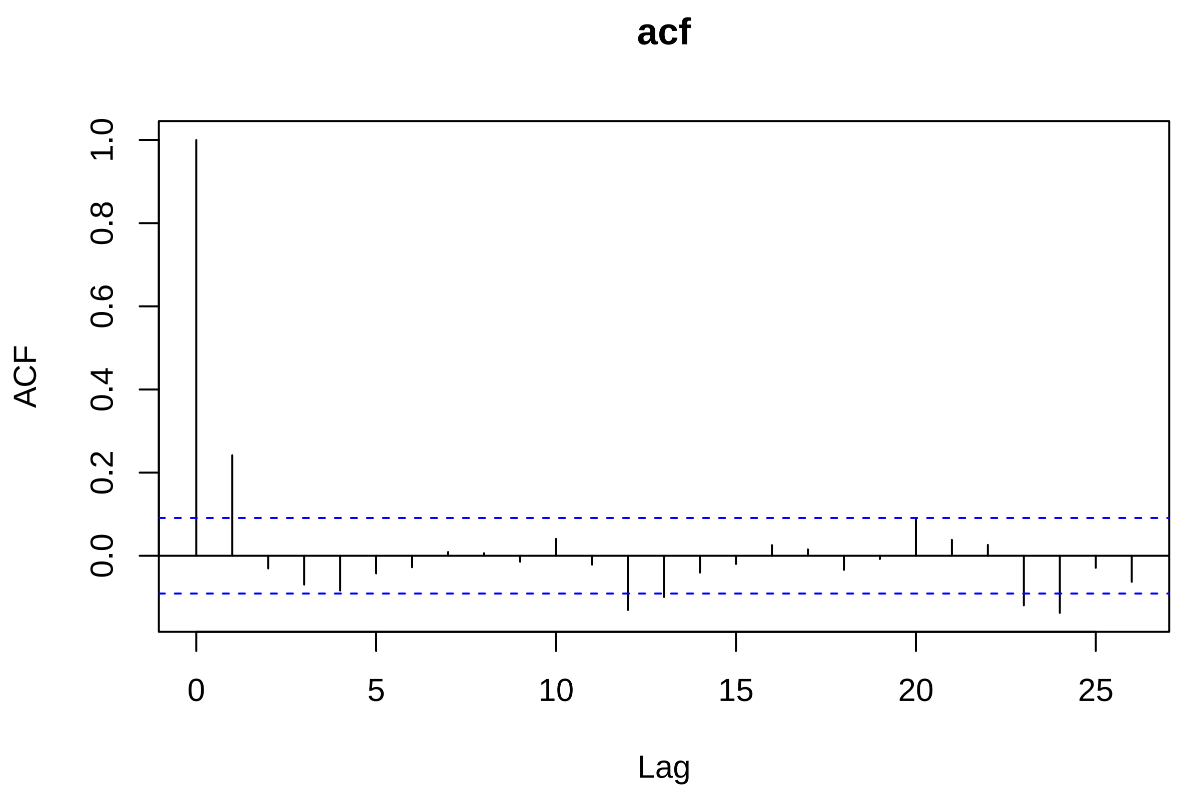
```r
library(tseries)
dicky_fuller <- adf.test(log_oil, k = 0)
print(dicky_fuller)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  log_oil
## Dickey-Fuller = -2.6551, Lag order = 0, p-value = 0.3009
## alternative hypothesis: stationary
```

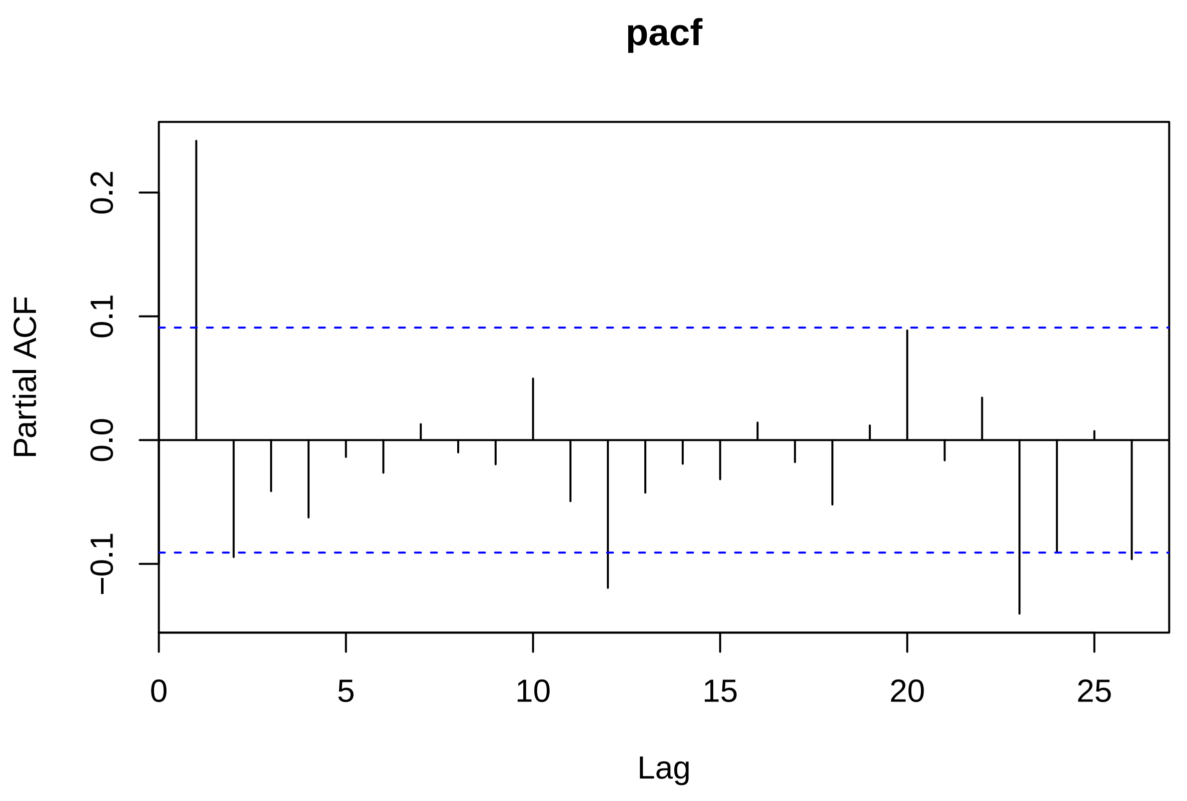
adf gave us a p-value of 0.3009 which is a lot greater than the rejection baseline of 0.05 so we fail to reject the null hypothesis that the log of oil prices has a unit root (not stationary).

##(e) Compute both the correlogram and the Augmented Dickey-Fuller test for the data in differences. What does the results suggest about yt(indifferncces)?

```r
attach(oil_data)
diff_series <- na.omit(diff_oil)
acf <- acf(diff_series, main = "acf")
```



```
pact <- pacf(diff_series, main = "pacf")
```



Now there looks like a sharp cutoff in the ACF after lag 1, as well as in the PACF, this suggests that there is some sort of stationarity, potentially yt in differences follows an AR(1) process.

(f) Estimate an ARMA(p,q) model with p = 0, 1, 2, 3 and q = 0, 1, 2, 3 for yt. Report the values of the BICs. Based on the correlogram and values of the BICs, which model would you choose? Explain your answer.

```r
library(forecast)

arma_bic_results <- data.frame(p = integer(), q = integer(), BIC = numeric())

for (p in 0:3) {
  for (q in 0:3) {
    if (p == 0 && q == 0) next  # Skip (0,0)

    model <- tryCatch(
      Arima(diff_series, order = c(p, 0, q), include.mean = FALSE),
      error = function(e) NULL
    )

    if (!is.null(model)) {
      bic_val <- BIC(model)
      arma_bic_results <- rbind(
        arma_bic_results,
        data.frame(p = p, q = q, BIC = bic_val)
      )
    }
  }
}

# sort by BIC
arma_bic_results <- arma_bic_results[order(arma_bic_results$BIC), ]
print(arma_bic_results)
```

```
##    p q       BIC
## 1  0 1 -923.1649
## 4  1 0 -919.9593
## 8  2 0 -917.8952
## 2  0 2 -917.0425
## 5  1 1 -917.0379
## 6  1 2 -916.5218
## 9  2 1 -915.9977
## 12 3 0 -912.5393
## 3  0 3 -911.3407
## 13 3 1 -910.9595
## 10 2 2 -910.9422
## 7  1 3 -910.8966
## 14 3 2 -904.9376
## 11 2 3 -904.7868
## 15 3 3 -898.8573
```

It looks like this process has a lowest BIC with an AR of 0 and a moving average process of 1. This aligned with the intuition of the Corellogram where only lag 1 has any sort of significant impact.

(g) Use the model you chose in part (f) to forecast changes in oil prices for the years of 2023 and 2024, and compute its 95% forecast interval as well. Plot your point and interval forecasts together with the actual data for the period from 2016 to 2024.

```r
diff_series <- ts(diff_series, start = c(1986, 2), frequency = 12)
train_diff <- window(diff_series, end = c(2022, 12))  # training data

start(diff_series)
```

```
## [1] 1986    2
```

```r
end(diff_series)
```

```
## [1] 2024   10
```

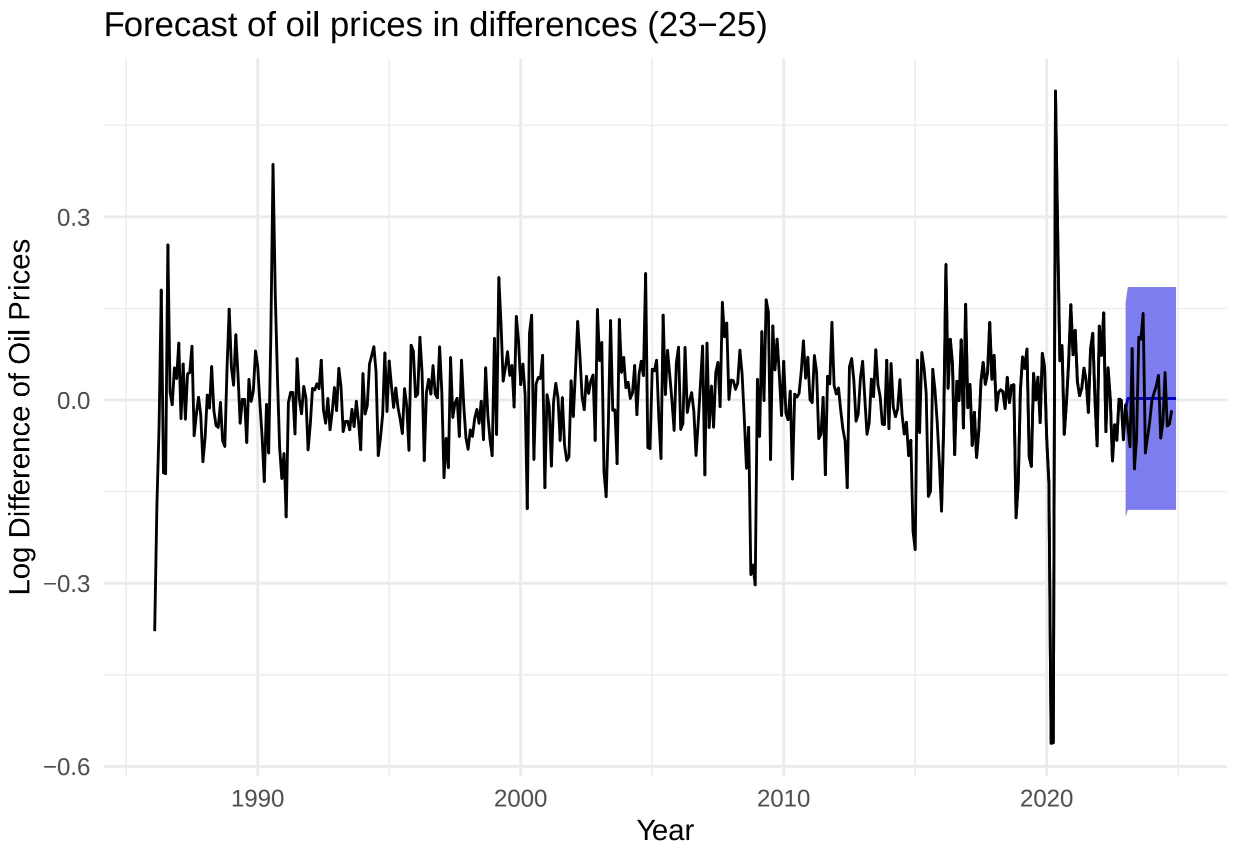
```r
ma_model <- Arima(train_diff, order = c(0, 0, 1))

ma_model
```

```
## Series: train_diff
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1    mean
##       0.2770  0.0026
## s.e.  0.0462  0.0054
##
## sigma^2 = 0.008028:  log likelihood = 441.07
## AIC=-876.14   AICc=-876.09   BIC=-863.86
```

```r
#  24 months ahead
ma_forecast <- forecast(ma_model, h = 24, level = 95)

ma_forecast
```

```
##          Point Forecast      Lo 95     Hi 95
## Jan 2023   -0.015640563 -0.1912508 0.1599696
## Feb 2023    0.002636933 -0.1795856 0.1848594
## Mar 2023    0.002636933 -0.1795856 0.1848594
## Apr 2023    0.002636933 -0.1795856 0.1848594
## May 2023    0.002636933 -0.1795856 0.1848594
## Jun 2023    0.002636933 -0.1795856 0.1848594
## Jul 2023    0.002636933 -0.1795856 0.1848594
## Aug 2023    0.002636933 -0.1795856 0.1848594
## Sep 2023    0.002636933 -0.1795856 0.1848594
## Oct 2023    0.002636933 -0.1795856 0.1848594
## Nov 2023    0.002636933 -0.1795856 0.1848594
## Dec 2023    0.002636933 -0.1795856 0.1848594
## Jan 2024    0.002636933 -0.1795856 0.1848594
## Feb 2024    0.002636933 -0.1795856 0.1848594
## Mar 2024    0.002636933 -0.1795856 0.1848594
## Apr 2024    0.002636933 -0.1795856 0.1848594
## May 2024    0.002636933 -0.1795856 0.1848594
## Jun 2024    0.002636933 -0.1795856 0.1848594
## Jul 2024    0.002636933 -0.1795856 0.1848594
## Aug 2024    0.002636933 -0.1795856 0.1848594
## Sep 2024    0.002636933 -0.1795856 0.1848594
## Oct 2024    0.002636933 -0.1795856 0.1848594
## Nov 2024    0.002636933 -0.1795856 0.1848594
## Dec 2024    0.002636933 -0.1795856 0.1848594
```

```
diff_series_window <- window(diff_series, start = c(2016, 1))

autoplot(ma_forecast) +
  autolayer(diff_series_window, series = "Actual Changes", color = "black") +
  labs(title = "Forecast of oil prices in differences (23-25)", x = "Year",
    y = "Log Difference of Oil Prices"
  ) +
  theme_minimal() +
  guides(color = guide_legend(title = "Series"))
```

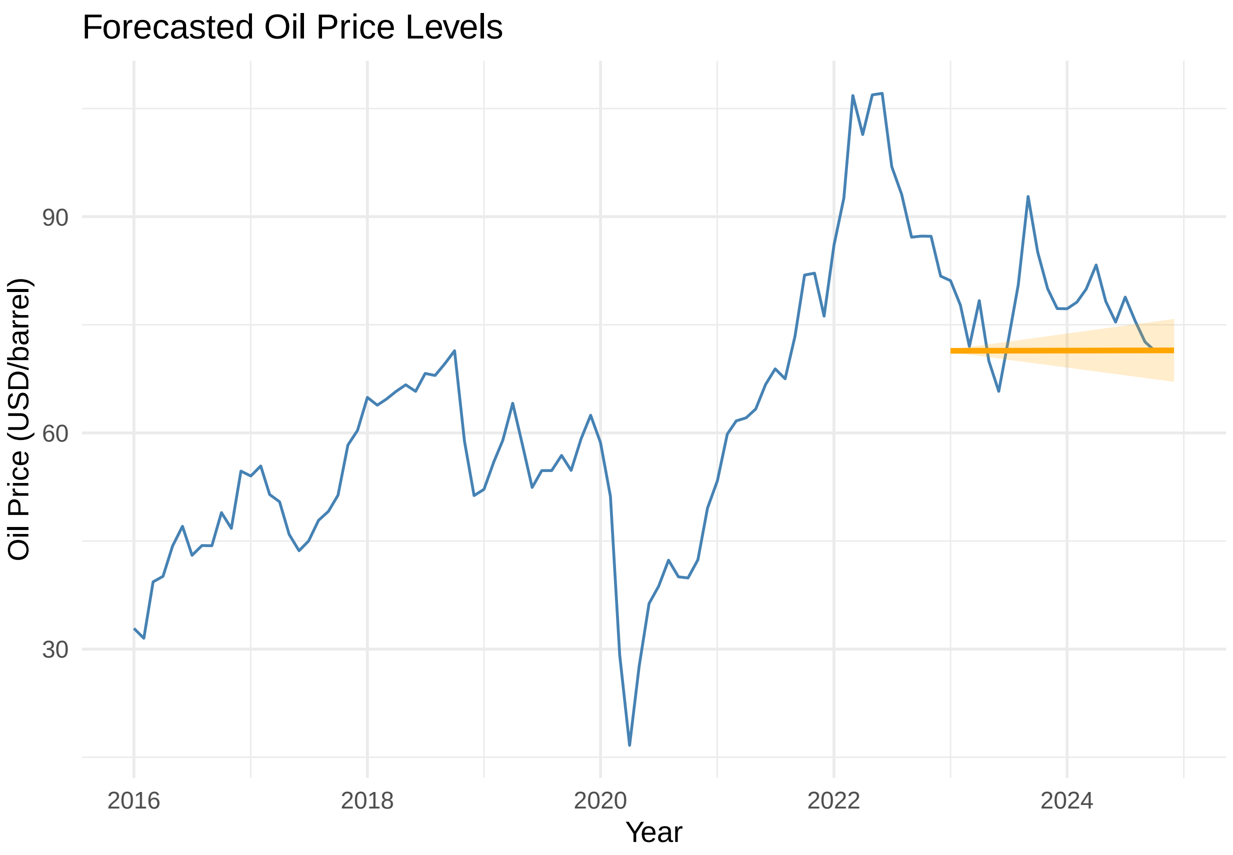


(h) Use the model you chose in part (f) to forecast the level of oil prices for the years of 2023 and 2024, and compute its 95% forecast interval as well. Plot your point and interval forecasts together with the actual data for the period from 2016 to 2024.

```r
library(dplyr)

last_level <- tail(oil_data$oil_price, 1)  # oil price as of dec 2022

# Recover level forecast from difference forecast
level_forecast <- cumsum(ma_forecast$mean) + last_level


lower_level <- cumsum(ma_forecast$lower[,1]) + last_level
upper_level <- cumsum(ma_forecast$upper[,1]) + last_level


dates_forecast <- seq(as.Date("2023-01-01"), by = "month", length.out = 24)

# forecast sample
oil_level_forecast <- data.frame(
  date = dates_forecast,
  point_forecast = level_forecast,
  lower_95 = lower_level,
  upper_95 = upper_level
)


oil_level_forecast
```

```
##          date point_forecast lower_95 upper_95
## 1  2023-01-01       71.38960 71.21399 71.56521
## 2  2023-02-01       71.39223 71.03440 71.75007
## 3  2023-03-01       71.39487 70.85481 71.93492
## 4  2023-04-01       71.39751 70.67523 72.11978
## 5  2023-05-01       71.40014 70.49564 72.30464
## 6  2023-06-01       71.40278 70.31606 72.48950
## 7  2023-07-01       71.40542 70.13647 72.67436
## 8  2023-08-01       71.40805 69.95689 72.85922
## 9  2023-09-01       71.41069 69.77730 73.04408
## 10 2023-10-01       71.41333 69.59772 73.22894
## 11 2023-11-01       71.41596 69.41813 73.41380
## 12 2023-12-01       71.41860 69.23854 73.59866
## 13 2024-01-01       71.42124 69.05896 73.78352
## 14 2024-02-01       71.42388 68.87937 73.96838
## 15 2024-03-01       71.42651 68.69979 74.15324
## 16 2024-04-01       71.42915 68.52020 74.33810
## 17 2024-05-01       71.43179 68.34062 74.52296
## 18 2024-06-01       71.43442 68.16103 74.70782
## 19 2024-07-01       71.43706 67.98144 74.89268
## 20 2024-08-01       71.43970 67.80186 75.07753
## 21 2024-09-01       71.44233 67.62227 75.26239
## 22 2024-10-01       71.44497 67.44269 75.44725
## 23 2024-11-01       71.44761 67.26310 75.63211
## 24 2024-12-01       71.45024 67.08352 75.81697
```

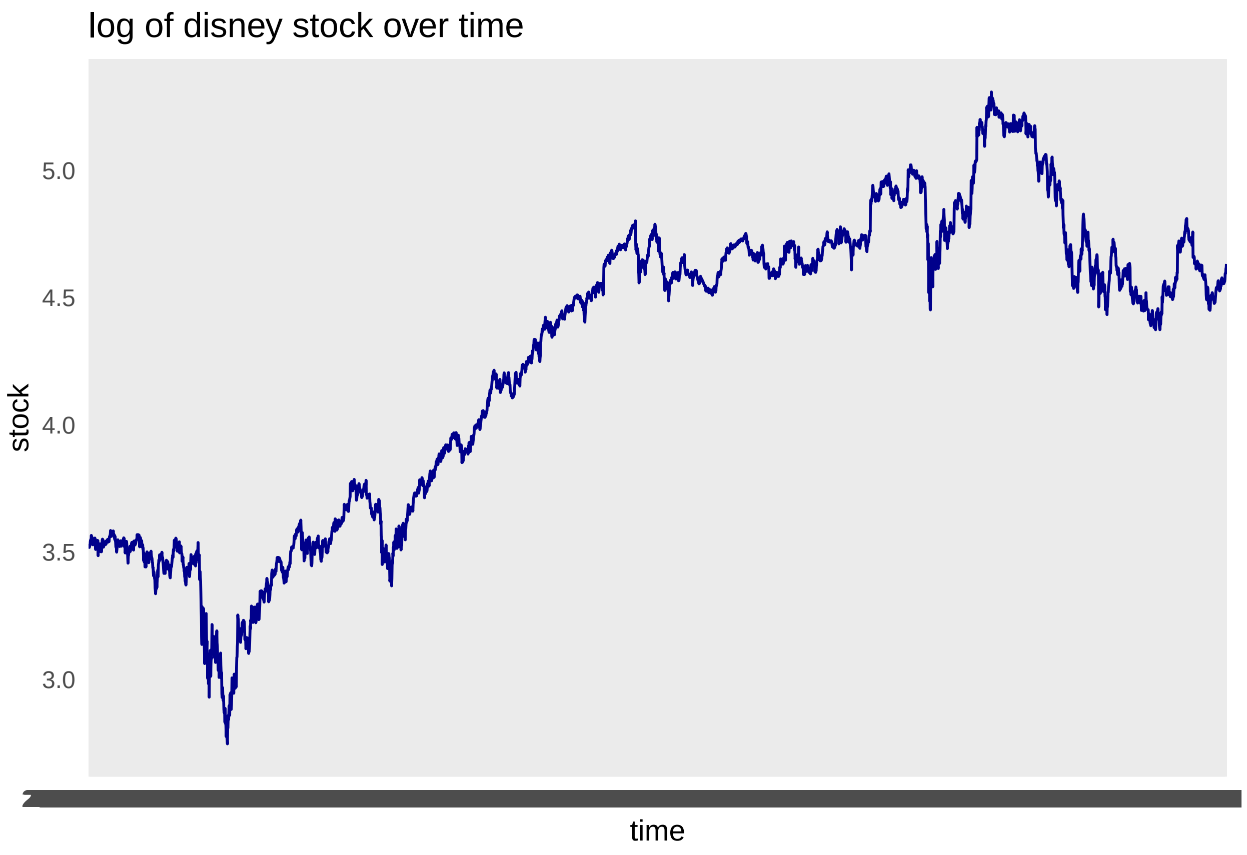
```
plot_data <- oil_data[oil_data$date >= as.Date("2016-01-01"), c("date", "oil_price")]

ggplot() +
  geom_line(data = plot_data, aes(x = date, y = oil_price), color = "steelblue") +
  geom_line(data = oil_level_forecast, aes(x = date, y = point_forecast), color = "orange", linewidth =
  geom_ribbon(data = oil_level_forecast, aes(x = date, ymin = lower_95, ymax = upper_95), alpha = 0.2,
  labs(title = "Forecasted Oil Price Levels",
       x = "Year", y = "Oil Price (USD/barrel)") +
  theme_minimal()
```



Question 3. Use the dataset disney__stock__price.csv that is available on Canvas. The variable dis- ney__stock is the closing price of Disney stock in the day. The sample is at daily frequency and covers the period from January 3, 2007 to November 13, 2024. Define the training sample as all information available up to September 2024.

(a) Let Pt denote the close price of the Disney stock. Plot the log of Pt over time, together with the correlogram of the data. Is there any evidence that the data is non-stationary? Explain your answer.

```r
library(dplyr)
disney <- read.csv("disney_stock_price.csv")
disney$X <- NULL

ggplot(disney, aes(x = date, y = log(disney_stock))) +
  geom_line(color = "darkblue", group = 1) +
  labs(title = "log of disney stock over time", x = "time",
       y = "stock") +
  theme_minimal()
```



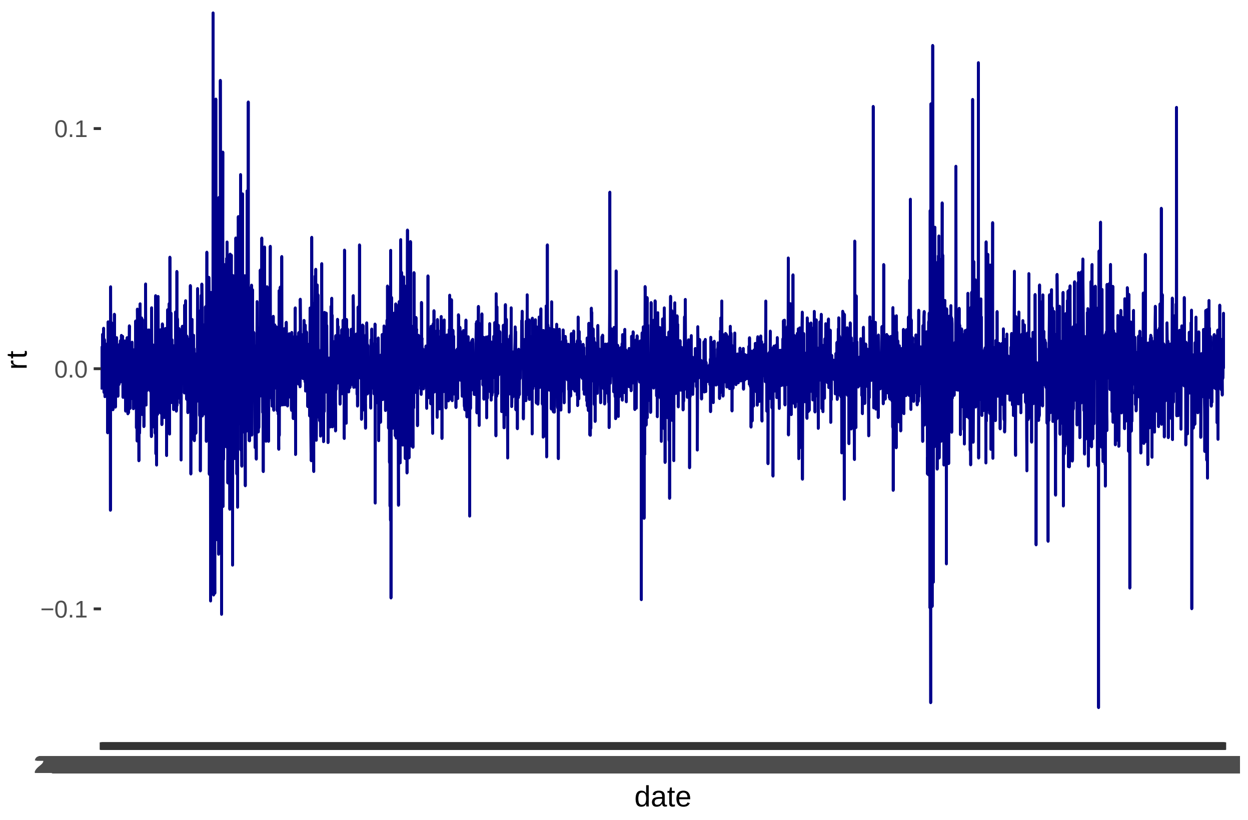
it looks like the averaege of this process is increasing over time, as well as a potential non constant variance is causing pretty weird spikes around some certain areas.

(b) From now on, let rt denote the the first difference of the log price, rt = differences log(Pt). Plot rt over time. Do you observe any volatility clustering? In which periods the volatility is higher

```r
disney$rt <- c(NA, diff(log(disney$disney_stock)))
length(disney$disney_stock)
```

```
## [1] 4498
```

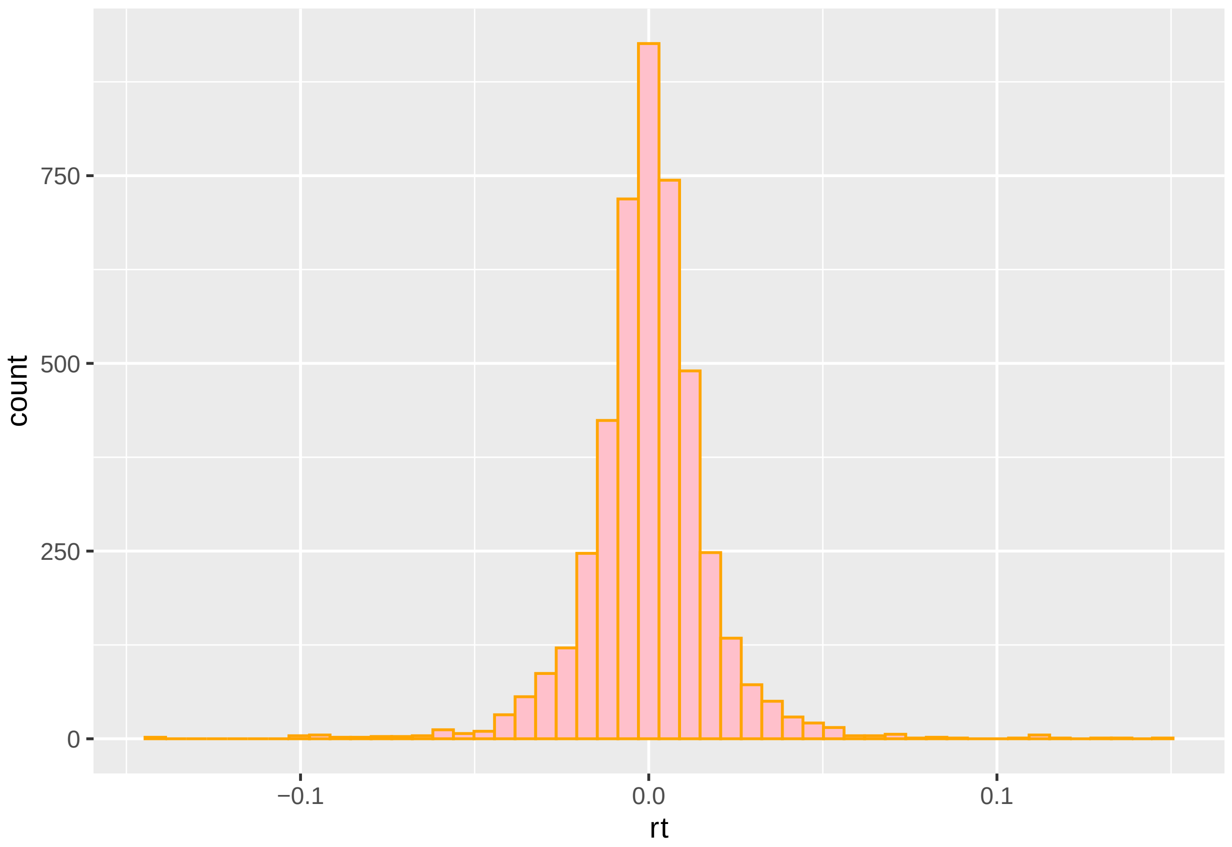
```r
ggplot(disney, aes(x = date, y = rt)) +
  geom_line(color = "darkblue", group = 1)
```



there is definitely some volatility clustering

(c) Plot the histogram and compute the descriptive statistics of rt (mean, median, standard deviation, skewness and kurtosis). Is there any evidence that the data is leptokurtic?

```r
library(moments)
ggplot(disney, aes(x = rt)) +
  geom_histogram(bins = 50, fill = "pink", color = "orange")
```



```
skewness(disney$rt, na.rm = TRUE)
```

```
## [1] 0.09202811
```

```
kurtosis(disney$rt, na.rm = TRUE)
```

```
## [1] 12.37177
```

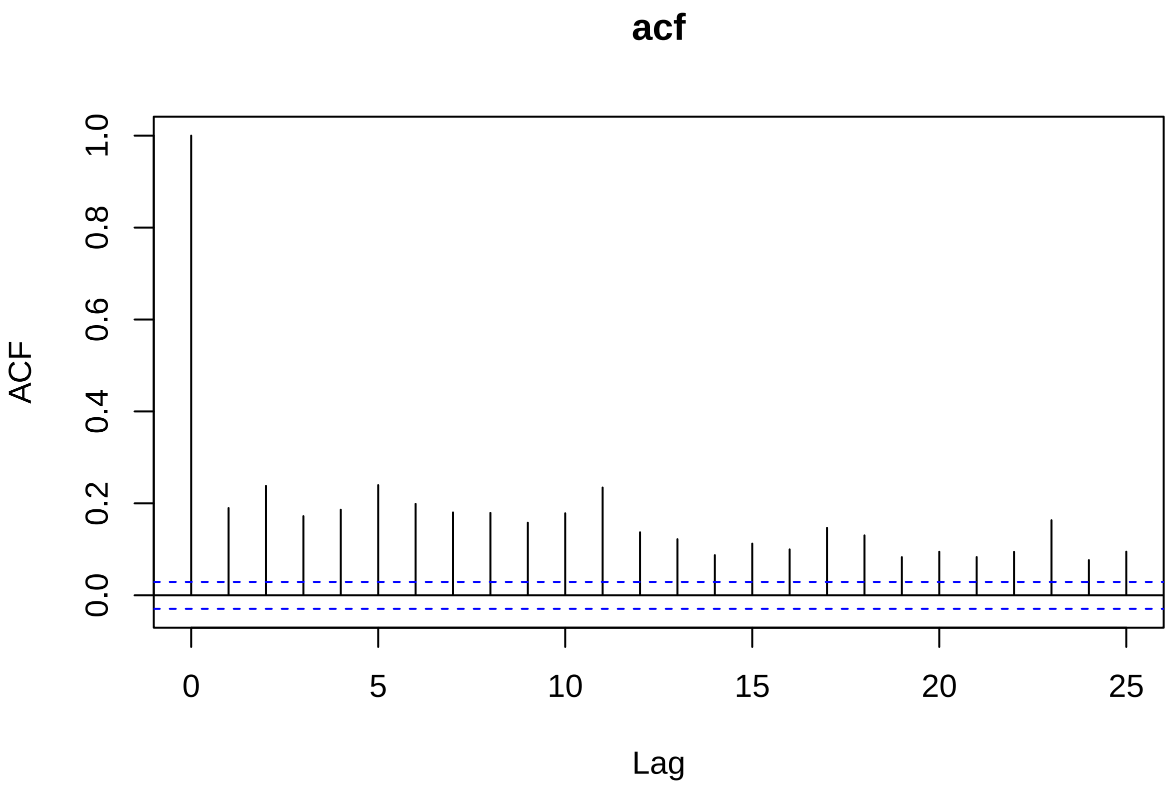
```
summary(disney$rt)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.       NA's
## -0.1411393 -0.0076871  0.0003780  0.0002476  0.0084710  0.1481804          1
```

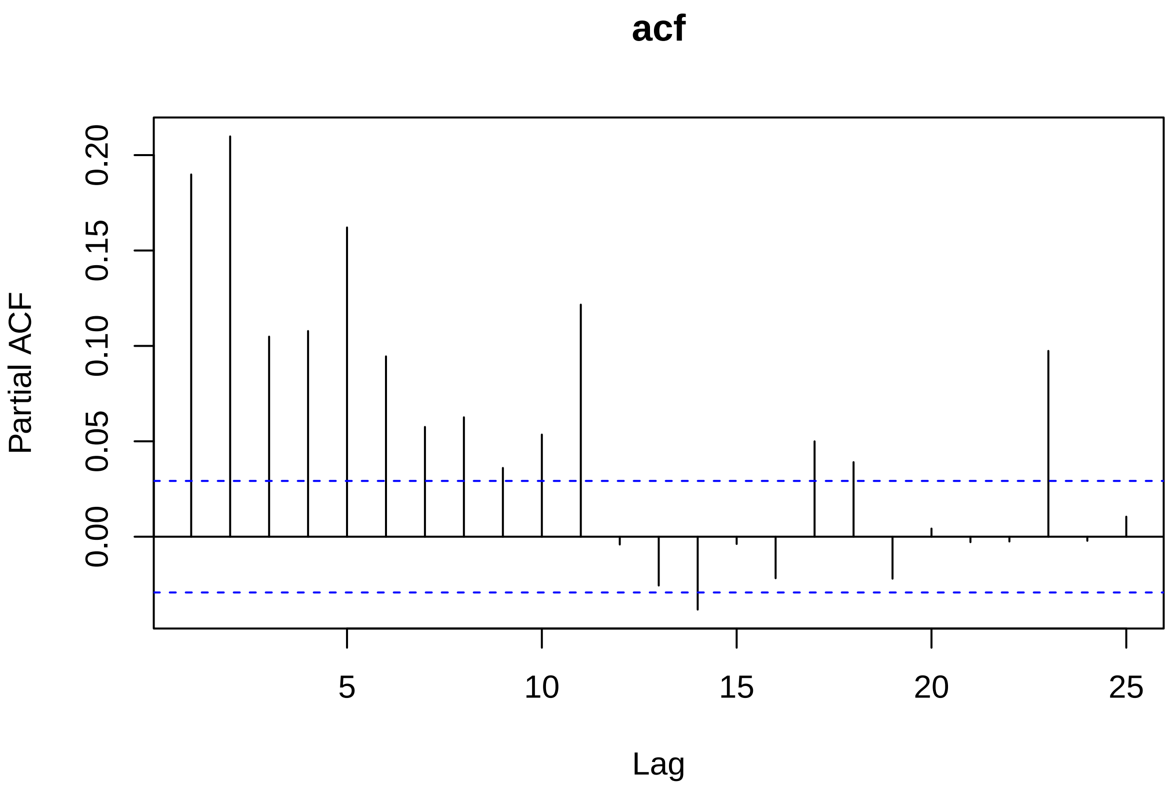
lets be honest, a fourth central moment around the mean of 12 is WILDLY leptokurtic

the data is centered around the mean but we've got the occasional VERY EXTREME outliar thats shooting our kurtosis values up ## (d) Compute the correlogram of the squared returns, that is, r2. Is there any evidence of serial correlation?

```
squared_returns <- disney$rt^2

squared_returns <- na.omit(squared_returns)
acf_disney <- acf(squared_returns, lag.max = 25, main = "acf")
```



```r
pacf_disney <- pacf(squared_returns, lag.max = 25, main = "acf")
```



the volatility of returns is definetly correlated, we see that all of the squared differences of returns are above the min threshold, meaning that the magnitude of returns are serially corrrelated.

(e) Estimate an AR(1) model for the squared returns. Is the AR(1) coefficient significant? What does that mean for rt?

```r
library(forecast)

returns_model <- arima(squared_returns, order = c(1,0,0))

returns_model
```

```
##
## Call:
## arima(x = squared_returns, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.1898      3e-04
## s.e.  0.0146      0e+00
##
## sigma^2 estimated as 1.14e-06:  log likelihood = 24387.87,  aic = -48769.74
```

The AR(1) coefficient is 0.1898, with a standard error of 0.0146. If we run a t-test to see if our lag is statistically significant we get, 12.97. This is highly significant, it suggests that there is significant autocorrelation in the variance.

(f) Estimate an ARCH(1) and a GARCH(1,1) model for which one fits the data better (based on SIC)?

```r
library(rugarch)
library(fGarch)

rt <- na.omit(disney$rt)

model_arch <- garchFit(~ garch(1,0), data = rt, trace = FALSE)
model_garch <- garchFit(~ garch(1, 1), data = rt, trace = FALSE)
summary(model_garch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = rt, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x135301350>
##  [data = rt]
##
## Conditional Distribution:
```

```
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 3.3206e-04  6.0629e-06  7.6897e-02  9.0506e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     3.321e-04   2.069e-04    1.605    0.108
## omega  6.063e-06   1.019e-06    5.950 2.68e-09 ***
## alpha1 7.690e-02   8.895e-03    8.645  < 2e-16 ***
## beta1  9.051e-01   1.028e-02   88.052  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  12362.38    normalized:  2.749027
##
## Description:
##  Wed May  7 21:50:37 2025 by user:
##
##
## Standardised Residuals Tests:
##                                    Statistic   p-Value
##  Jarque-Bera Test   R    Chi^2  1.215795e+04 0.0000000
##  Shapiro-Wilk Test  R    W      9.415346e-01 0.0000000
##  Ljung-Box Test     R    Q(10)  6.786412e+00 0.7454434
##  Ljung-Box Test     R    Q(15)  1.413135e+01 0.5155913
##  Ljung-Box Test     R    Q(20)  1.669564e+01 0.6726391
##  Ljung-Box Test     R^2  Q(10)  2.581260e+00 0.9896379
##  Ljung-Box Test     R^2  Q(15)  4.872409e+00 0.9931527
##  Ljung-Box Test     R^2  Q(20)  5.769536e+00 0.9991749
##  LM Arch Test       R    TR^2   3.400136e+00 0.9919991
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.496276 -5.490573 -5.496277 -5.494266
```

```r
summary(model_arch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 0), data = rt, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 0)
## <environment: 0x140721078>
##  [data = rt]
```

```
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1
## 0.00041491  0.00022968  0.30813926
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     4.149e-04   2.390e-04    1.736   0.0825 .
## omega  2.297e-04   6.164e-06   37.261   <2e-16 ***
## alpha1 3.081e-01   2.639e-02   11.678   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  11932.03    normalized:  2.653332
##
## Description:
##  Wed May  7 21:50:37 2025 by user:
##
##
## Standardised Residuals Tests:
##                                  Statistic   p-Value
##  Jarque-Bera Test   R    Chi^2  1.730384e+04 0.0000000
##  Shapiro-Wilk Test  R    W      9.144026e-01 0.0000000
##  Ljung-Box Test     R    Q(10)  4.794783e+00 0.9044582
##  Ljung-Box Test     R    Q(15)  1.232568e+01 0.6542293
##  Ljung-Box Test     R    Q(20)  1.492669e+01 0.7805871
##  Ljung-Box Test     R^2  Q(10)  1.898566e+02 0.0000000
##  Ljung-Box Test     R^2  Q(15)  2.988620e+02 0.0000000
##  Ljung-Box Test     R^2  Q(20)  3.472614e+02 0.0000000
##  LM Arch Test       R    TR^2   1.964248e+02 0.0000000
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.305330 -5.301053 -5.305331 -5.303823
```

looks like GARCH(1,1) has a smaller AIC, and out performed in the sense of model fit

(g) Estimate an AR(1)-ARCH(1) and a AR(1)-GARCH(1,1)

```r
arch_ar1 <- garchFit(formula = ~ arma(1,0) + garch(1,0), data = rt,
                     trace = FALSE)
garch_ar1 <- garchFit(formula = ~ + arma(1,0) + garch(1,1),
                      data = rt, trace = FALSE)

summary(garch_ar1)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~+arma(1, 0) + garch(1, 1), data = rt, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ +arma(1, 0) + garch(1, 1)
## <environment: 0x14751d748>
##  [data = rt]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1        omega       alpha1        beta1
##  3.4483e-04  -3.8339e-02   5.9991e-06   7.6873e-02   9.0533e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      3.448e-04   2.073e-04    1.663   0.0962 .
## ar1    -3.834e-02   1.654e-02   -2.319   0.0204 *
## omega   5.999e-06   1.016e-06    5.905 3.53e-09 ***
## alpha1  7.687e-02   8.949e-03    8.590  < 2e-16 ***
## beta1   9.053e-01   1.032e-02   87.719  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  12365.19    normalized:  2.749654
##
## Description:
##  Wed May  7 21:50:37 2025 by user:
##
##
## Standardised Residuals Tests:
##                                    Statistic   p-Value
##  Jarque-Bera Test   R    Chi^2  1.202746e+04 0.0000000
##  Shapiro-Wilk Test  R    W      9.416881e-01 0.0000000
##  Ljung-Box Test     R    Q(10)  4.565545e+00 0.9182504
##  Ljung-Box Test     R    Q(15)  1.215103e+01 0.6675631
##  Ljung-Box Test     R    Q(20)  1.485697e+01 0.7845323
##  Ljung-Box Test     R^2  Q(10)  2.549560e+00 0.9901337
##  Ljung-Box Test     R^2  Q(15)  4.887587e+00 0.9930361
##  Ljung-Box Test     R^2  Q(20)  5.866261e+00 0.9990664
##  LM Arch Test       R    TR^2   3.365363e+00 0.9923687
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.497085 -5.489956 -5.497087 -5.494573
```

```r
summary(arch_ar1)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(1, 0) + garch(1, 0), data = rt, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(1, 0) + garch(1, 0)
## <environment: 0x1416b4848>
##  [data = rt]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1        omega       alpha1
##  0.00043035  -0.06320544   0.00022620   0.32829340
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      4.304e-04   2.407e-04    1.788  0.07380 .
## ar1    -6.321e-02   1.982e-02   -3.189  0.00143 **
## omega   2.262e-04   6.313e-06   35.833  < 2e-16 ***
## alpha1  3.283e-01   2.914e-02   11.266  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  11936.71    normalized:  2.654372
##
## Description:
##  Wed May  7 21:50:37 2025 by user:
##
##
## Standardised Residuals Tests:
##                                  Statistic   p-Value
##  Jarque-Bera Test   R    Chi^2  1.656626e+04 0.0000000
##  Shapiro-Wilk Test  R    W      9.156439e-01 0.0000000
##  Ljung-Box Test     R    Q(10)  4.868581e+00 0.8997821
##  Ljung-Box Test     R    Q(15)  1.185790e+01 0.6897517
##  Ljung-Box Test     R    Q(20)  1.396534e+01 0.8322490
##  Ljung-Box Test     R^2  Q(10)  1.833207e+02 0.0000000
##  Ljung-Box Test     R^2  Q(15)  2.890884e+02 0.0000000
##  Ljung-Box Test     R^2  Q(20)  3.363230e+02 0.0000000
##  LM Arch Test       R    TR^2   1.937637e+02 0.0000000
##
## Information Criterion Statistics:
```

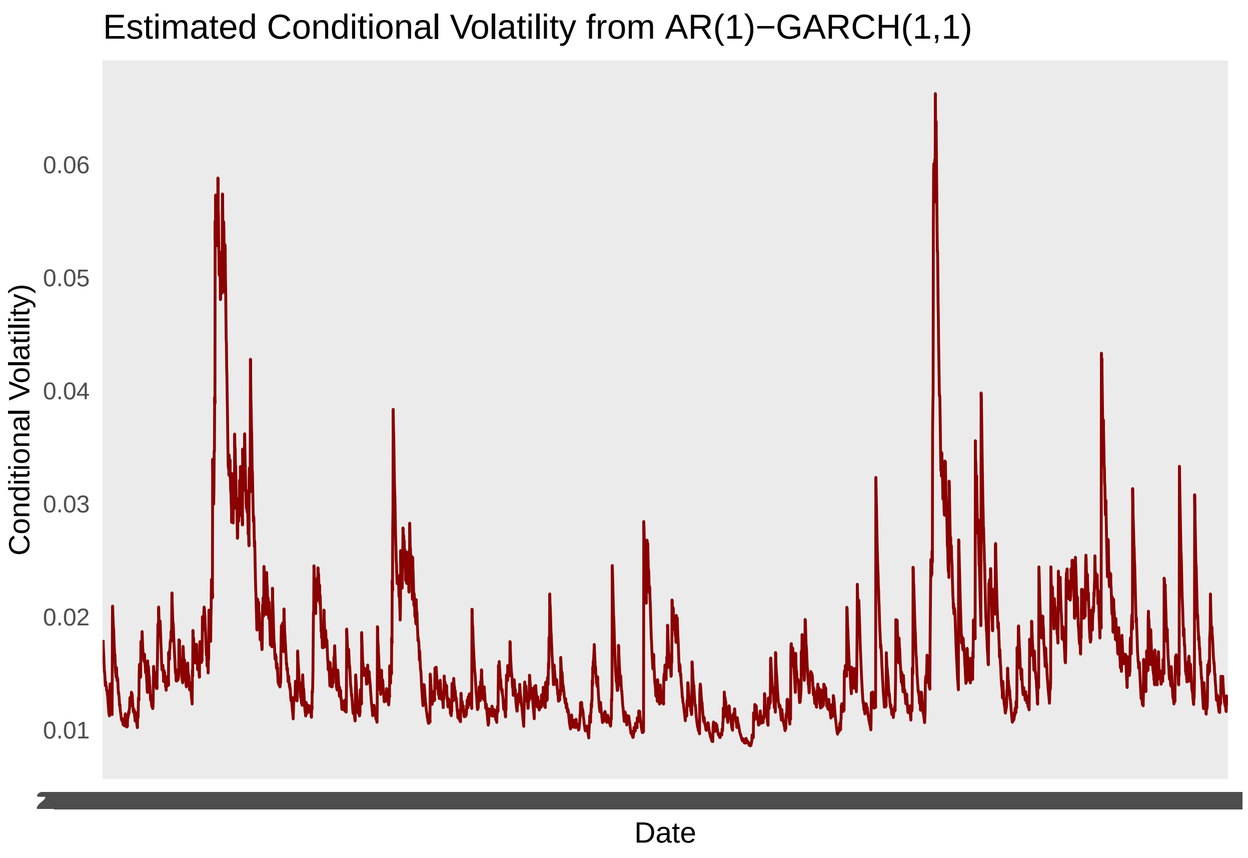
```
##       AIC       BIC       SIC      HQIC 
## -5.306966 -5.301263 -5.306967 -5.304956
```

it looks like the AR(1)-GARCH(1,1) model performed just a bit better, witha BIC of -5.48 compared to AR(1)-ARCH(1)'s BIC of -5.30

GARCH(1,1) did the best but very marginally, but in reality, adding an AR(1) component didn't really chnage much.

(h) Plot the estimated conditional volatility of the best-fitting model among the ones considered in the previous parts.

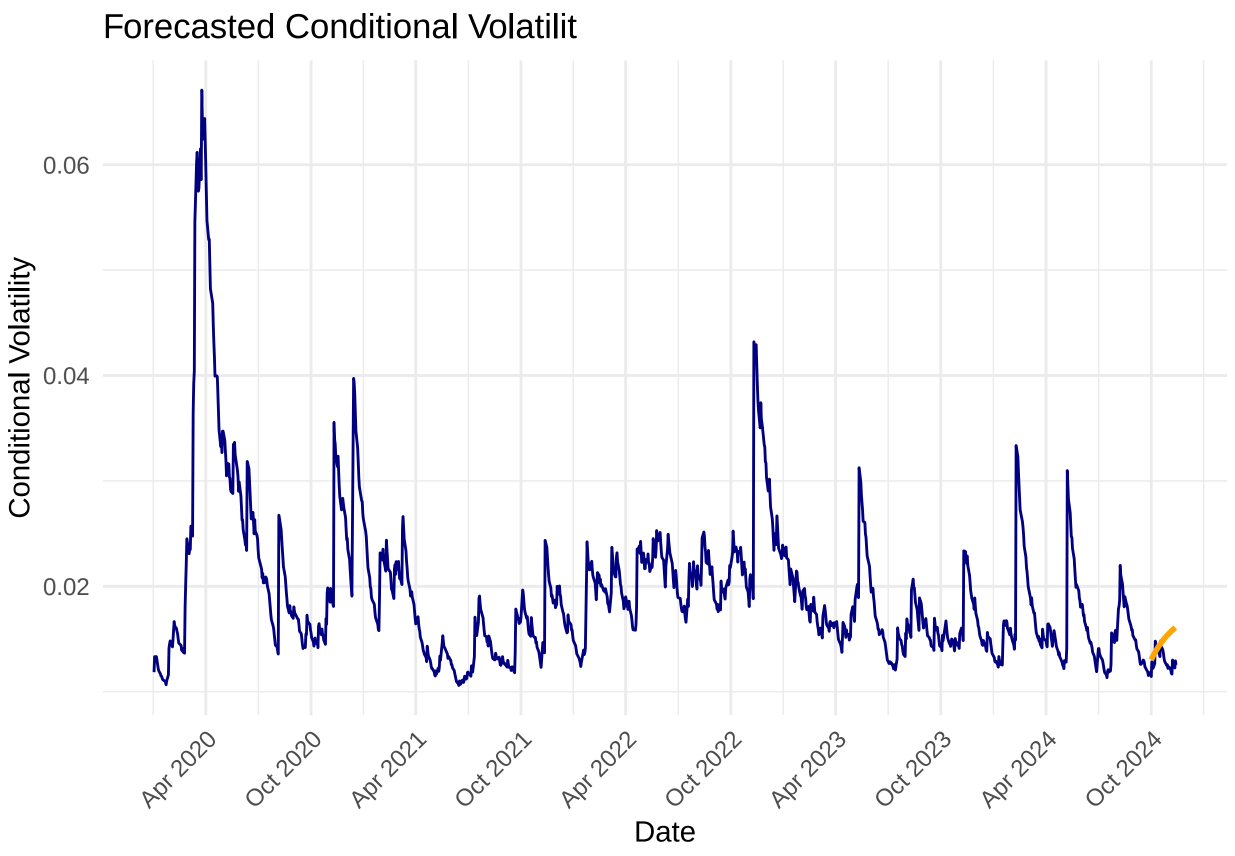
```r
cond_vol <- garch_ar1@sigma.t

vol_df <- data.frame(
  date = disney$date[!is.na(disney$rt)],  # remove NA to align
  volatility = cond_vol
)

library(ggplot2)

ggplot(vol_df, aes(x = date, y = volatility, group = 1)) +
  geom_line(color = "darkred") +
  labs(
    title = "Estimated Conditional Volatility from AR(1)-GARCH(1,1)",
    x = "Date",
    y = "Conditional Volatility)"
  ) +
  theme_minimal()
```



(i) Use the model you chose in the previous part to forecast the conditional volatility of rt for the period of October and November of 2024. Plot your forecasts together with the estimated conditional volatility for the period between 2020 and 2024.

```
vol_forecast <- predict(model_garch, n.ahead = 43)

forecast_dates <- seq(from = as.Date("2024-10-01"), by = "day", length.out = 43)
forecast_df <- data.frame(
  date = forecast_dates,
  forecast_vol = vol_forecast$standardDeviation
)

hist_vol_df <- data.frame(
  date = disney$date[-1],
  vol = model_garch@sigma.t
) %>%
  filter(date >= as.Date("2020-01-01"))

library(ggplot2)
hist_vol_df$date <- as.Date(hist_vol_df$date)
ggplot() +
  geom_line(data = hist_vol_df, aes(x = date, y = vol), color = "navy", linewidth = 0.5) +
  geom_line(data = forecast_df, aes(x = date, y = forecast_vol), color = "orange", linewidth = 1) +
```

```r
  labs(
    title = "Forecasted Conditional Volatilit",
    x = "Date",
    y = "Conditional Volatility"
  ) +
  theme_minimal() +
  scale_x_date(date_breaks = "6 months", date_labels = "%b %Y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Forecasted Conditional Volatilit
Conditional Volatility
Apr 2020
Oct 2020
Apr 2021
Oct 2021
Apr 2022
Oct 2022
Apr 2023
Oct 2023
Apr 2024
Oct 2024
Date