Time Series

Midterm project

Sven Bergmann

03/12/24

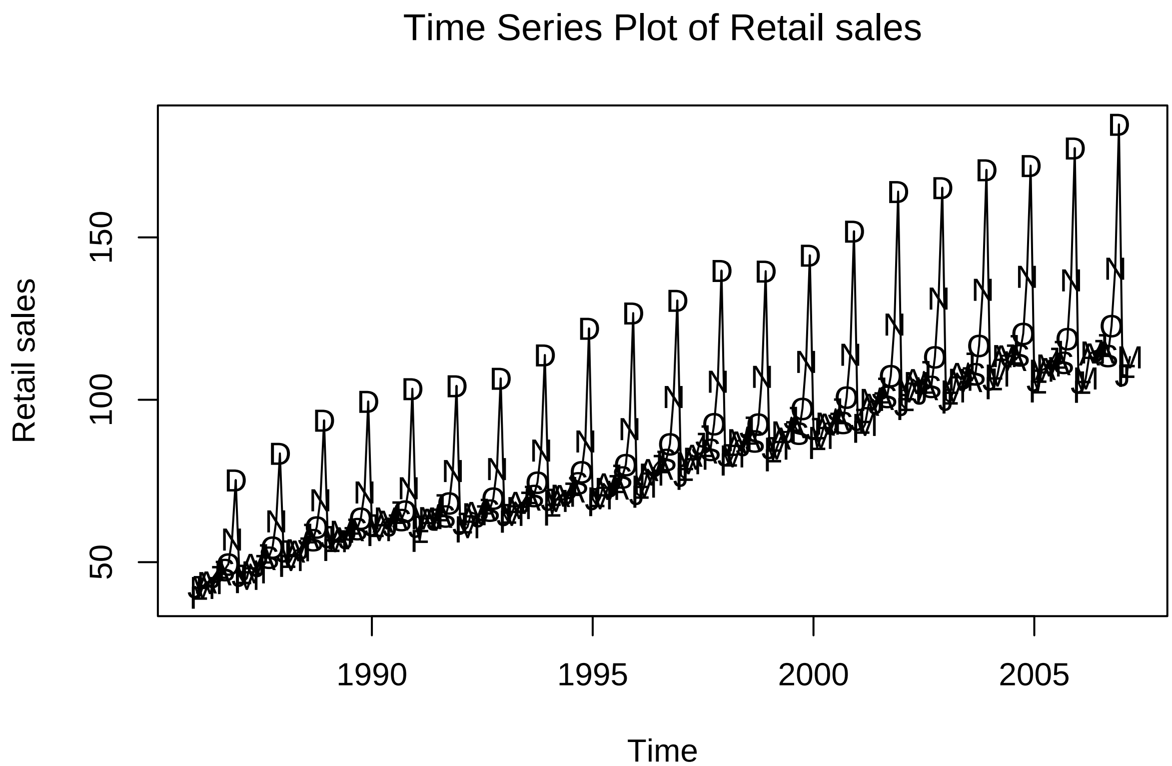
Part 1

1.

I am expecting a somewhat clear upward trend due to the increase in the population. Also, there could be a seasonal trend, such that the retail sales are higher on special holidays. Since the Brexit was later than 2007, I am not expecting a big sink somewhere in the graph.

```
library(TSA)
library(tseries)
```

```
data(retail)
plot(retail, xlab = expression("Time"), ylab = expression("Retail sales"),
    main = expression("Time Series Plot of Retail sales"))
points(retail, pch = as.vector(season(retail)))
```



Seeing the actual graph, I can confirm the upward trend and also a very strong seasonal behaviour. I clearly expected more holidays to be that present as christmas. There surely is a really strong seasonal behaviour due to christmas. This means that the sales are going up starting November each year and reach the peak in December before dropping again.

2.

I would try a seasonal ARIMA model, given the upward trend and the potential seasonality.

```
model <- forecast::auto.arima(retail, seasonal = TRUE)
summary(model)
```

```
## Series: retail
## ARIMA(1,0,4)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1      ma1     ma2     ma3     ma4     sma1   drift
##       0.8469  -0.5939  0.0504  0.0878  0.1742  -0.0966  0.3083
## s.e.  0.0533   0.0788  0.0740  0.0772  0.0756   0.0631  0.0405
##
## sigma^2 = 3.413:  log likelihood = -491.04
## AIC=998.07   AICc=998.69   BIC=1026.02
##
## Training set error measures:
##                          ME     RMSE      MAE         MPE     MAPE      MASE
```

```
## Training set -0.0003999956 1.777353 1.287341 -0.1007814 1.485438 0.3327206
##                    ACF1
## Training set 0.00339952
```

Model type and coefficients

Looking at the summary, we are given an ARIMA model with $p = 1$, $d = 0$, $q = 4$, where q is the autoregressive order with the coefficient $ar_1 = 0.8469$, also known as the lag order. d is the number of differences we take, which is 0, so that indicates that the data is already stationary. And q is the order of the moving average process, where the coefficients are $ma_1 = -0.5939$, $ma_2 = 0.0504$, $ma_3 = 0.0878$, $ma_4 = 0.1742$. Speaking in a sentence, we have a first order degree autoregressive model with zero order of differencing and a fourth order moving average model. That was for the non-seasonal part. For the seasonal part, we have $p = 0$, $d = 1$, $q = 1$, where p, d and q are exactly as above. The number 12 in brackets just states the number of seasons (here: months).

Fit

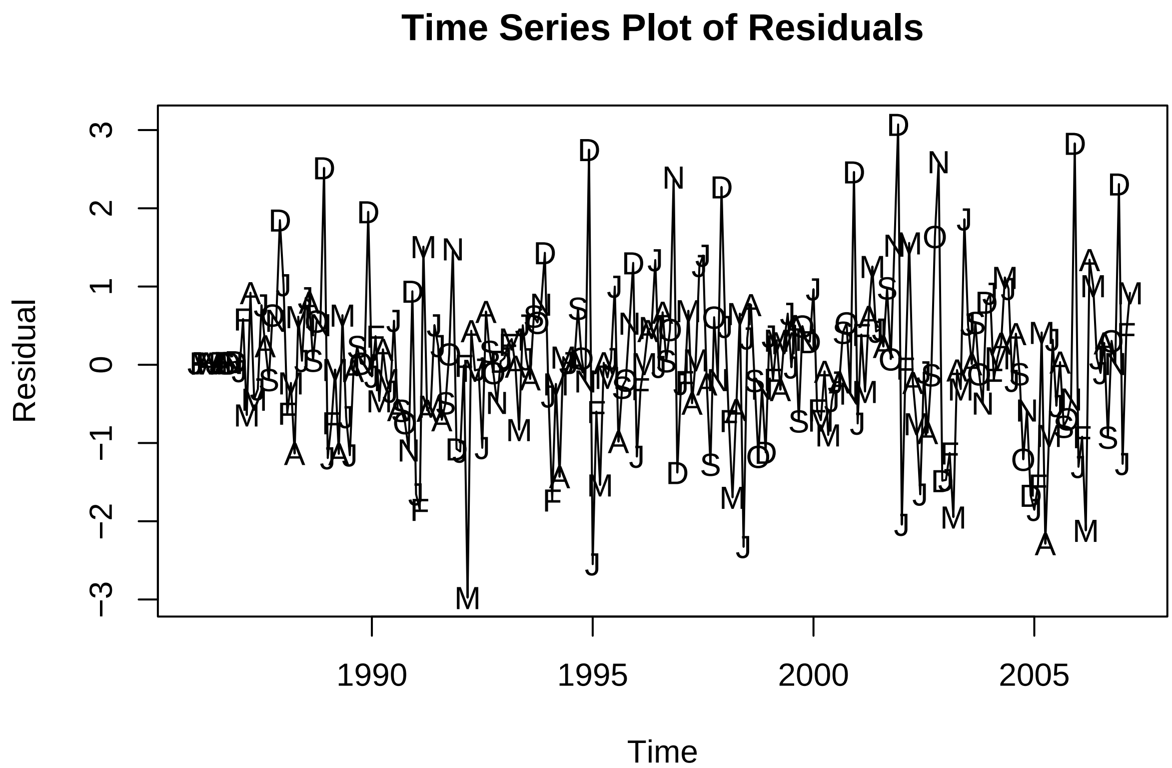
We have some values to rate the model fit, which are

$$\begin{aligned} \sigma^2 &= 3.413, \\ \log\text{likelihood} &= -491.04, \\ AIC &= 998.07, \\ AICc &= 998.69, \\ BIC &= 1026.02. \end{aligned}$$

Here, we can just look at AIC and BIC which suggest a reasonable choice based on model complexity vs fit. The other values are basically for forecasting reasons.

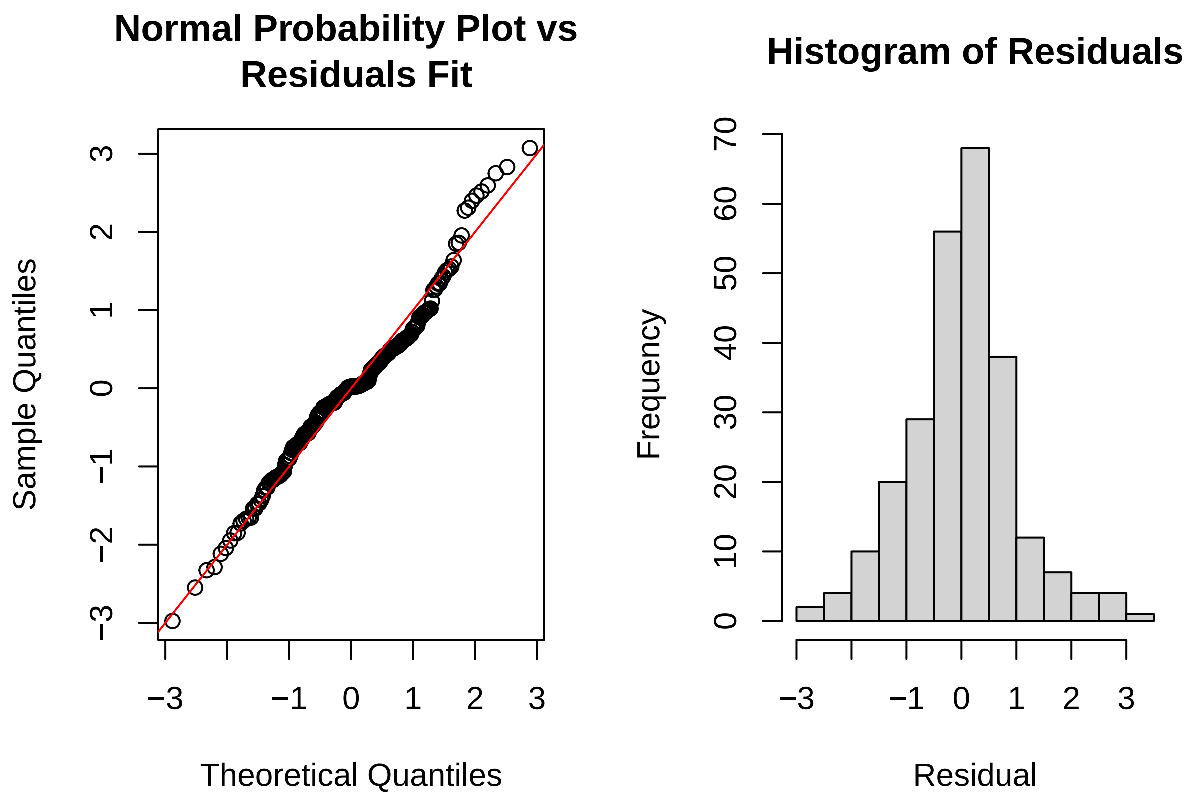
3.

```r
res <- ts(rstandard(model), start = c(1986, 1), end = c(2007,
    3), frequency = 12)
plot(res, xlab = "Time", ylab = "Residual", main = "Time Series Plot of Residuals")
points(y = res, x = time(res), pch = as.vector(season(res)))
```



The plot of the residuals is kind of displaying the seasonal/periodic behaviour.

```
par(mfrow = c(1, 2))
qqnorm(res, main = "Normal Probability Plot vs \n Residuals Fit")
abline(a = 0, b = 1, col = "red")
hist(res, xlab = "Residual", main = "Histogram of Residuals")
```



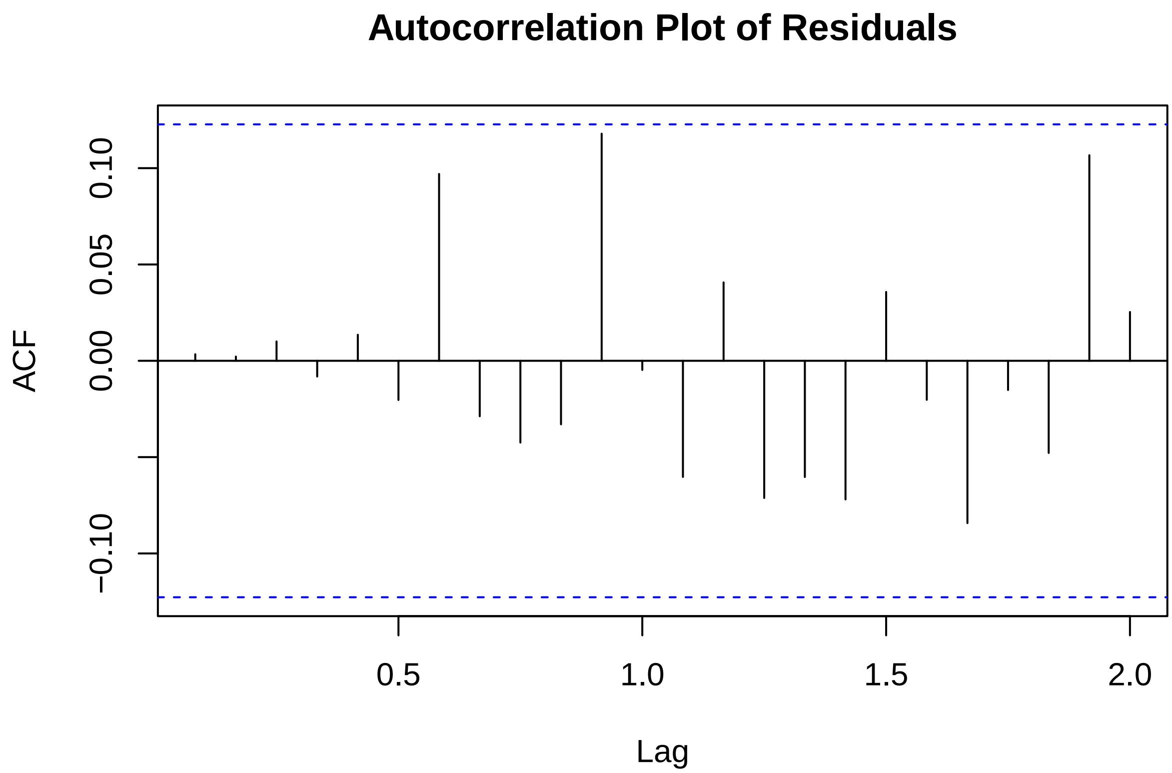
The residuals look like they are normally distributed, indicating a good model fit.

```
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.97597, p-value = 0.0002613
```

Since the p-value of the Shapiro-Wilk test is also really low, we can say that they are normally distributed.

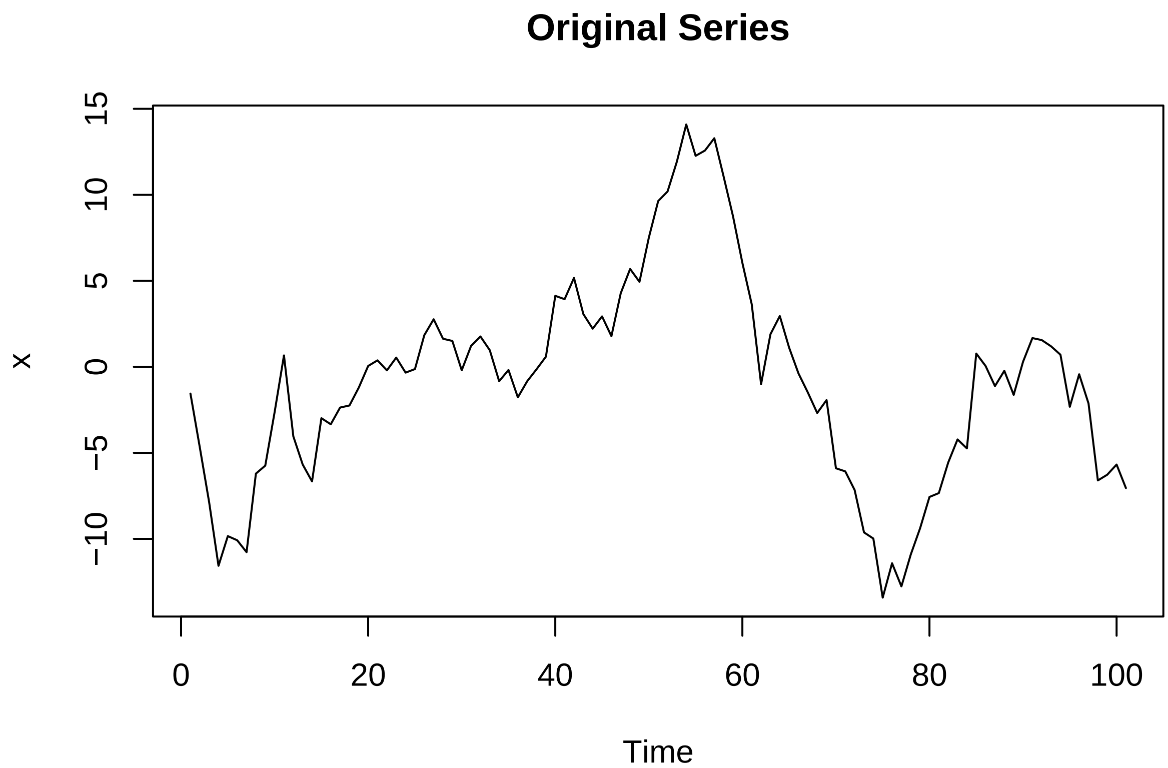
```
acf(res, main = "Autocorrelation Plot of Residuals")
```



Since there are just some "spikes" which are not at all big, the chosen model seems to capture the time-series pretty well. Also, no value is outside of the confidence interval.

Part 2

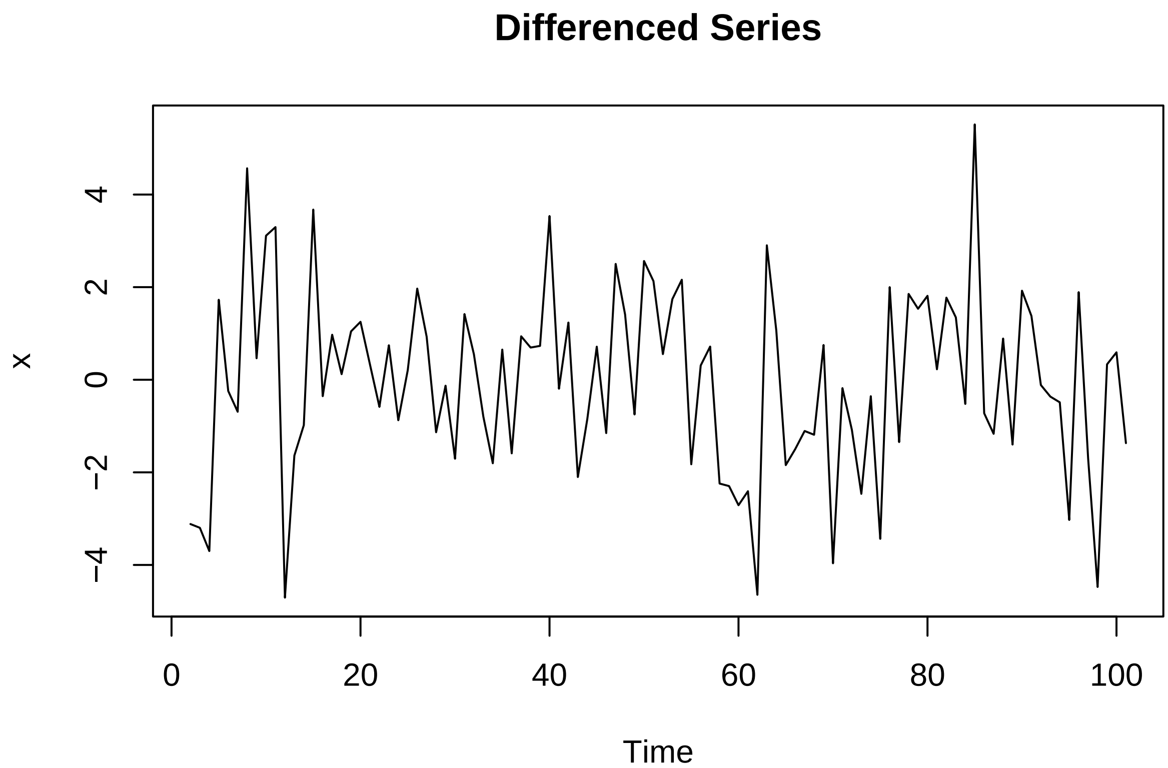
```
data <- as.ts(read.delim(file = "../homework/MidtermPt2.txt",
    header = TRUE, sep = " "))
plot(data, main = "Original Series")
```




```
adf_test <- adf.test(data)
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -2.2725, Lag order = 4, p-value = 0.4637
## alternative hypothesis: stationary
```

```
d_data <- diff(data)
plot(d_data, main = "Differenced Series")
```




```
adf_test <- adf.test(d_data)
```

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

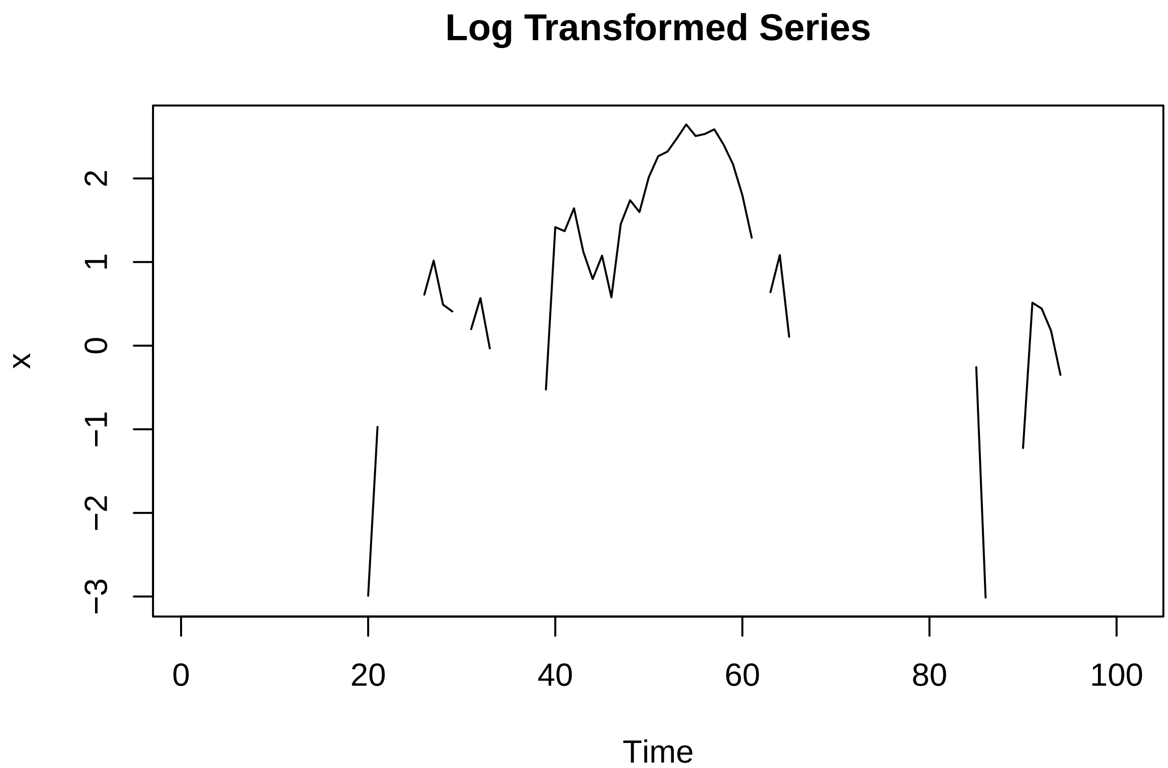
```
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  d_data
## Dickey-Fuller = -4.0875, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
log_data <- log(data)
```

```
## Warning in log(data): NaNs produced
```

```
plot(log_data, main = "Log Transformed Series")
```




```r
# adf_test <- adf.test(log_data) print(adf_test)
```

```r
model <- forecast::auto.arima(d_data)
summary(model)
```

```
## Series: d_data
## ARIMA(0,0,0) with zero mean
##
## sigma^2 = 3.94:  log likelihood = -210.45
## AIC=422.91   AICc=422.95   BIC=425.51
##
## Training set error measures:
##                        ME     RMSE      MAE MPE MAPE      MASE       ACF1
## Training set -0.05495792 1.984958 1.589698 100  100 0.7621968 0.06560549
```

```r
# model <- forecast::auto.arima(log_data) summary(model)
```

```r
model <- forecast::auto.arima(data)
summary(model)
```

```
## Series: data
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
```

```
##          ar1      ar2      ma1
##       1.8705  -0.8893  -0.8441
## s.e.  0.1036   0.0944   0.1535
##
## sigma^2 = 3.72:  log likelihood = -209.37
## AIC=426.75   AICc=427.16   BIC=437.21
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.1121809 1.899798 1.518746 15.07285 141.2535 0.9553675
##                     ACF1
## Training set -0.03684619
```