Time Series Analysis for Economic Turnover in Switzerland and Simulation

Seoyeon Park

Load the libraries you might need in the following chunck.

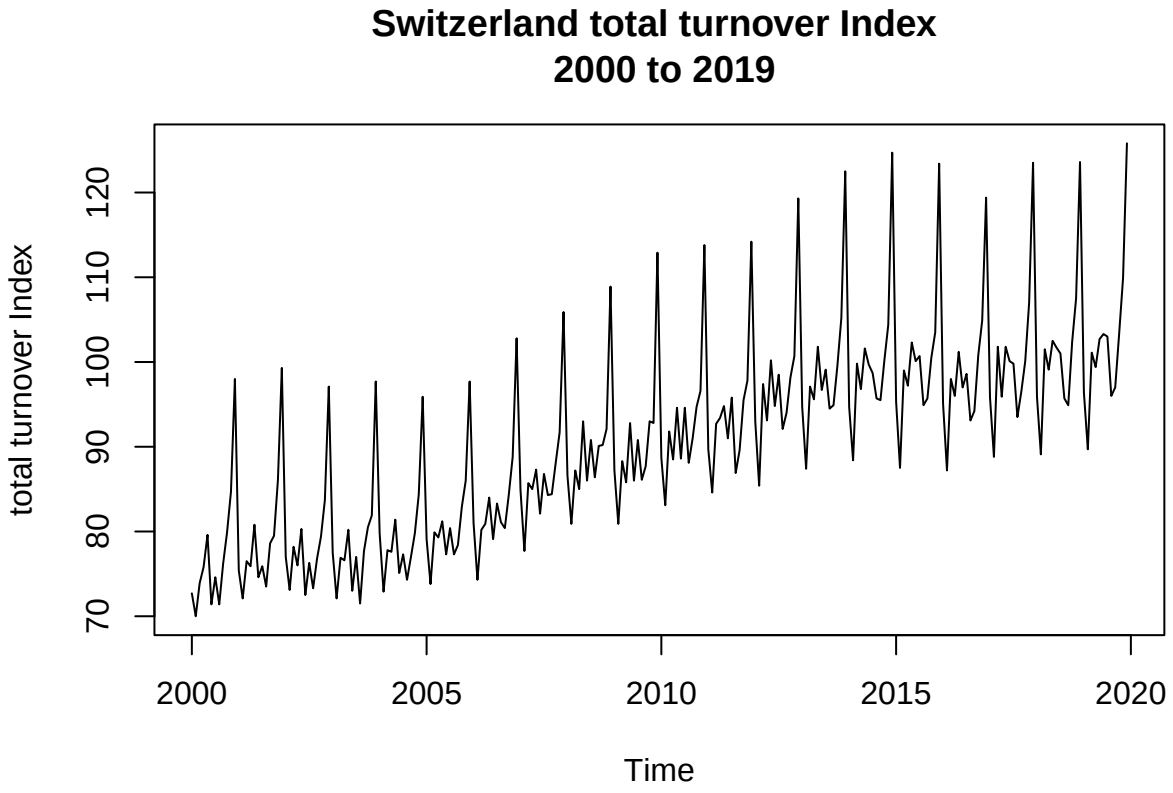
```
library(astsa)
library(forecast)
```

European Countries Turnover Index Analysis:

- **Analyzed** the total turnover index of Switzerland as part of a study on 10 European countries from 2000 to 2019.
- **Utilized time-series analysis** and statistical techniques to identify trends, patterns, and key economic insights in Switzerland's turnover data.
- **Developed visualizations** and reports to effectively communicate findings, providing insights into the turnover index fluctuations in Switzerland over time.

Plot the time series to visualize trends and patterns in the data over time.

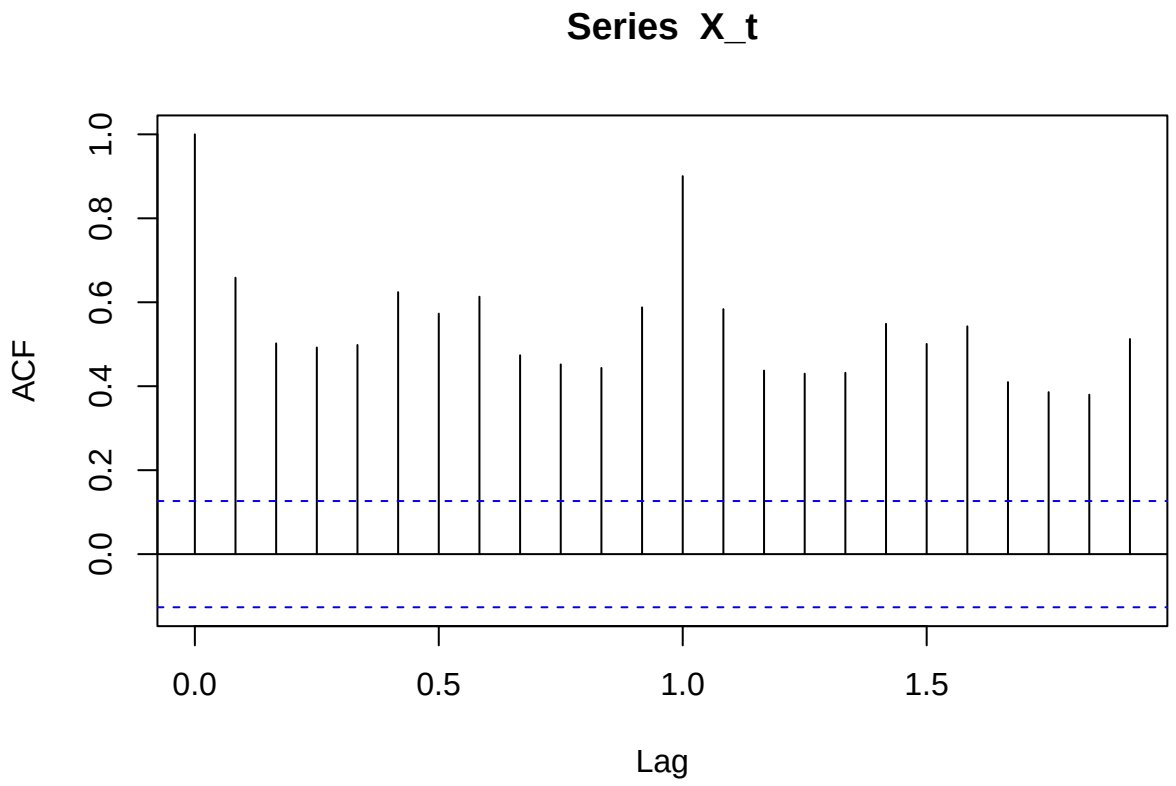
```
data = read.csv( "switzerland.csv")
data$Year <- as.numeric(substr(data$TIME, 1, 4))
data$Month <- as.numeric(substr(data$TIME, 6, 7))
#View(data)
#class(data)

X_t <- ts(data$Switzerland, start=data$Year[1], frequency=12)
#class(X_t)

# Plot the data
ts.plot(X_t, ylab="total turnover Index",
        main="Switzerland total turnover Index \n 2000 to 2019")
```



This plot illustrates the changes in the total turnover index in Switzerland from 2000 to 2019. It examines the presence of trends, seasonal patterns, and variability over time. These features—trend, seasonality, and variability—will be analyzed in the following questions.

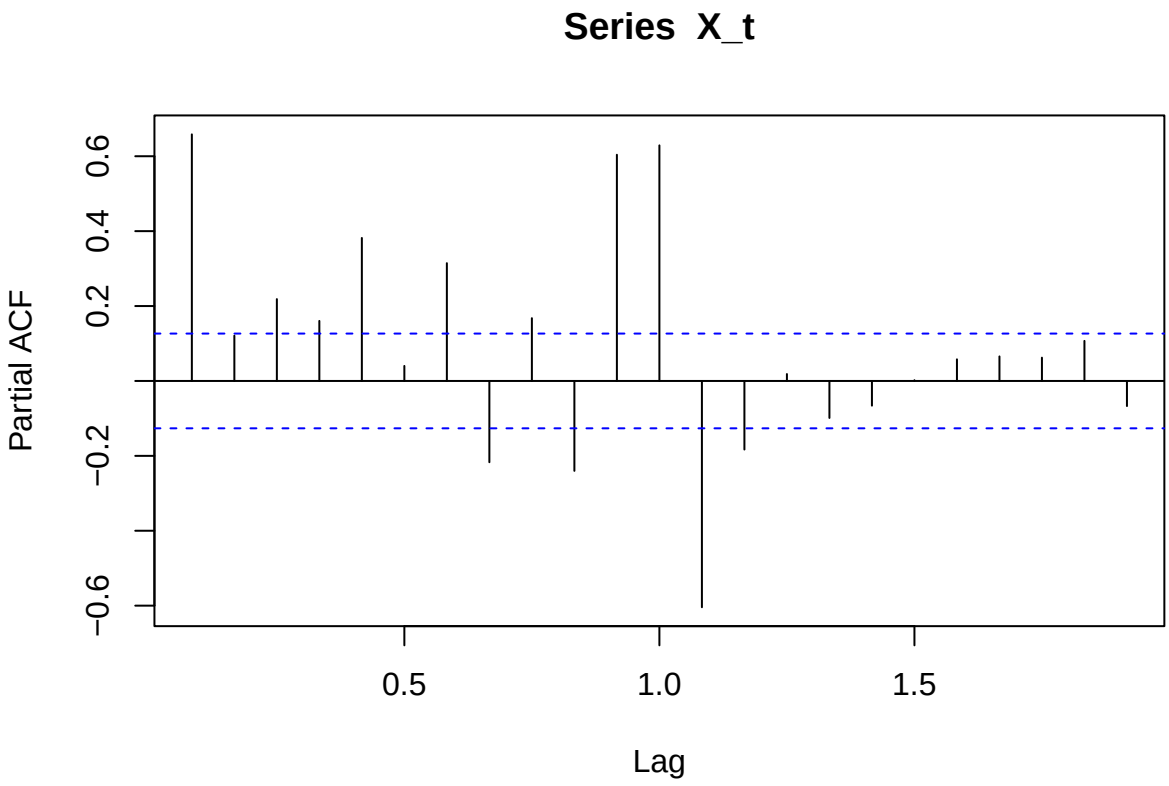
Plotted the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to analyze the time series data, identifying significant spikes and annual seasonality within the dataset.

```
# Plot ACF and PACF
#par(mfrow=c(2,1))
acf(X_t)
```



This plot shows the autocorrelation function (ACF) for the time series. The significant spikes outside of the confidence intervals indicate the presence of autocorrelation in the data. Additionally, the W-shaped pattern suggests annual seasonality in the series.

```
pacf(X_t)
```



This plot displays the partial autocorrelation function (PACF) for the time series.

- The variation in the time series appears to increase over time. This is evident as the fluctuations in the turnover index become larger from 2000 to 2019.
- There is an increasing trend in the total turnover index for Switzerland over the period.
- There is some seasonal downs and ups pattern visible in the plot in i). The regular fluctuations are repeating approximately on annual basis. The significant spikes observed in ACF plot with the seasonal downs and ups pattern additionally confirm the seasonality.

```
library(forecast)
lambda <- BoxCox.lambda(X_t)
lambda
```

```
## [1] 0.1131914
```

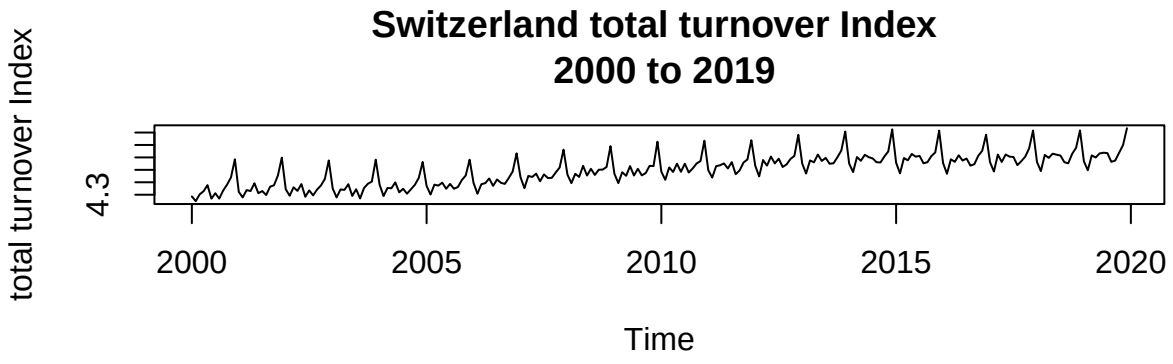
```
X_transform <-log(X_t)
#X_transform
# ts.plot(X_t, ylab="total turnover Index",
#         main="Switzerland total turnover Index \n 2000 to 2019")
#
# ts.plot(X_transform, ylab="log total turnover Index",
#         main="Switzerland log total turnover Index \n 2000 to 2019")
```

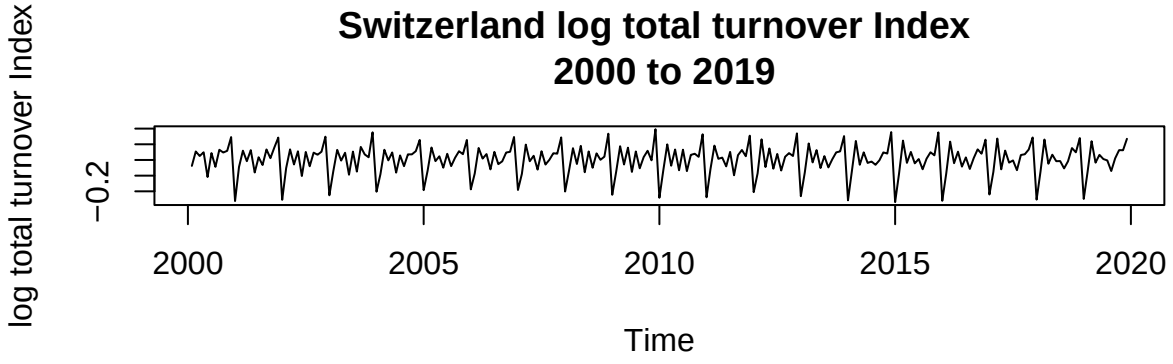
Since the lambda value is 0.113914, which is less than 0.25, the data will be transformed using the log function.

The log transformation ensures that the variance is constant over time, addressing the issue of increasing variation

Detrend the data by applying first-order differencing (lag 1) to remove the trend.

```
# differencing in lag 1
# This step removes the linear trend from the transformed data.
X_diff1<-diff(X_transform, lag=1)

######ADDITIONAL WORKS IGNORE LATER#######
par(mfrow=c(2,1))
ts.plot(X_transform, ylab="total turnover Index",
         main="Switzerland total turnover Index \n 2000 to 2019")

ts.plot(X_diff1, ylab="log total turnover Index",
        main="Switzerland log total turnover Index \n 2000 to 2019")
```



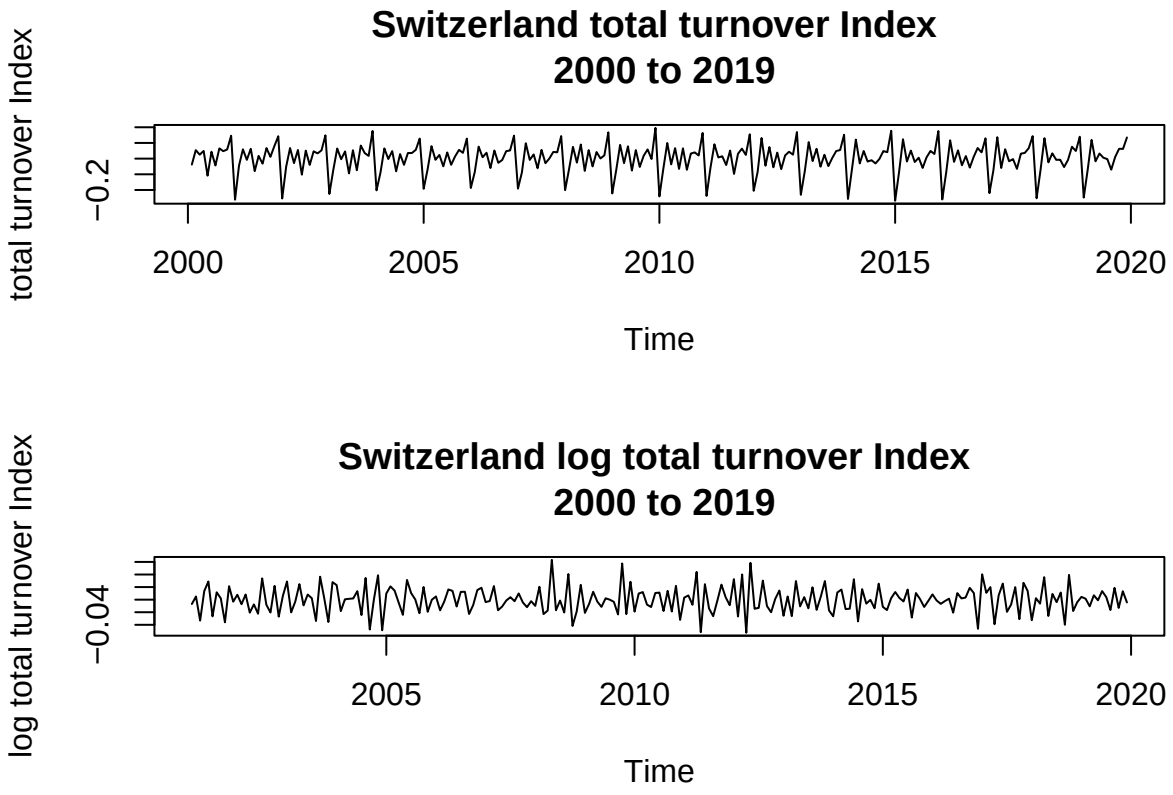


There is an increasing trend in the total turnover index for Switzerland over the period. Differencing with a lag of 1 effectively removes this trend from the data, as demonstrated in the additional plots.

Differencing at the appropriate lag to remove seasonal patterns.

```
#differencing to remove seasonality
#This step removes the seasonality, assuming a yearly seasonal pattern in monthly data.
X_diff1_12 <- diff(X_diff1, lag=12)
```

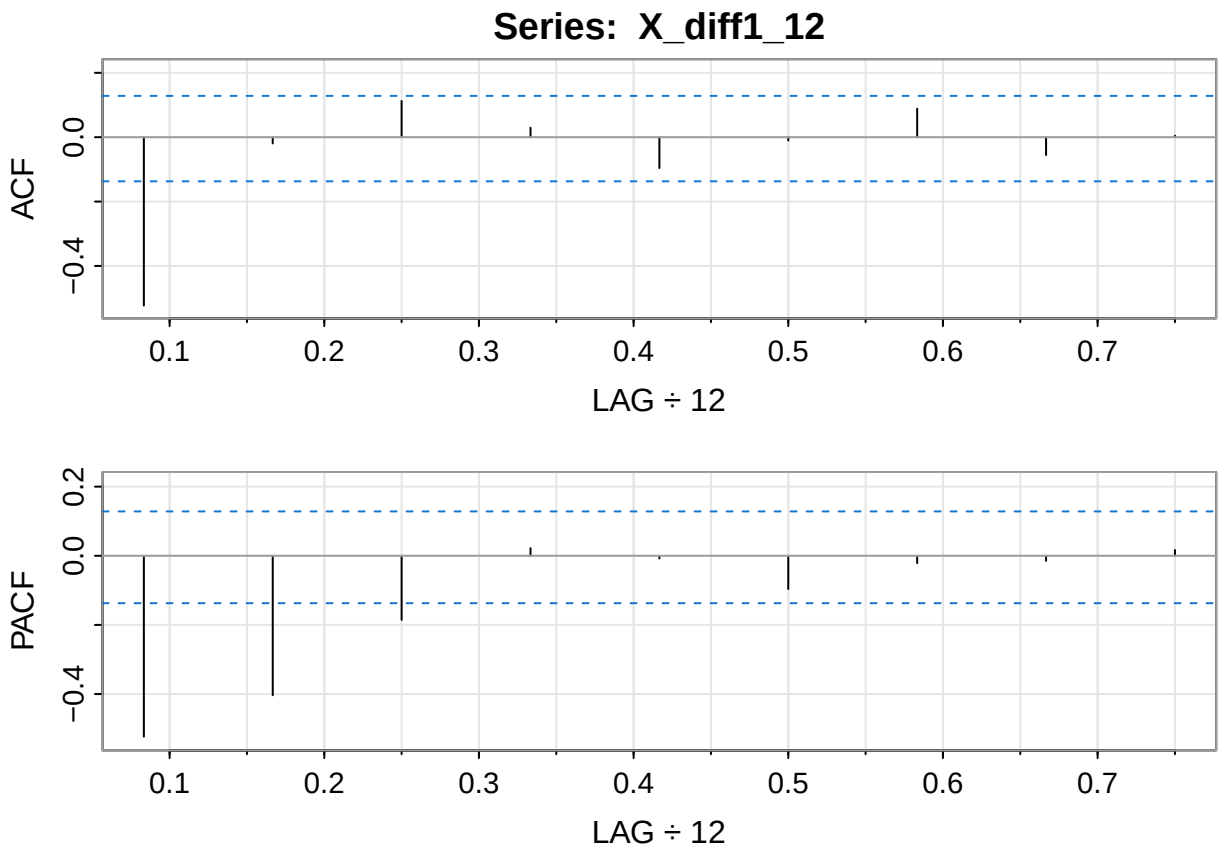
```
######ADDITIONAL WORKS IGNORE LATER#######
par(mfrow=c(2,1))
ts.plot(X_diff1, ylab="total turnover Index",
         main="Switzerland total turnover Index \n 2000 to 2019")

ts.plot(X_diff1_12, ylab="log total turnover Index",
        main="Switzerland log total turnover Index \n 2000 to 2019")
```



There was an annual seasonal trend. Therefore, we differenced the data with a lag of 12 months, which is equivalent to 1 year. After differencing with a lag of 12, the seasonality has disappeared.

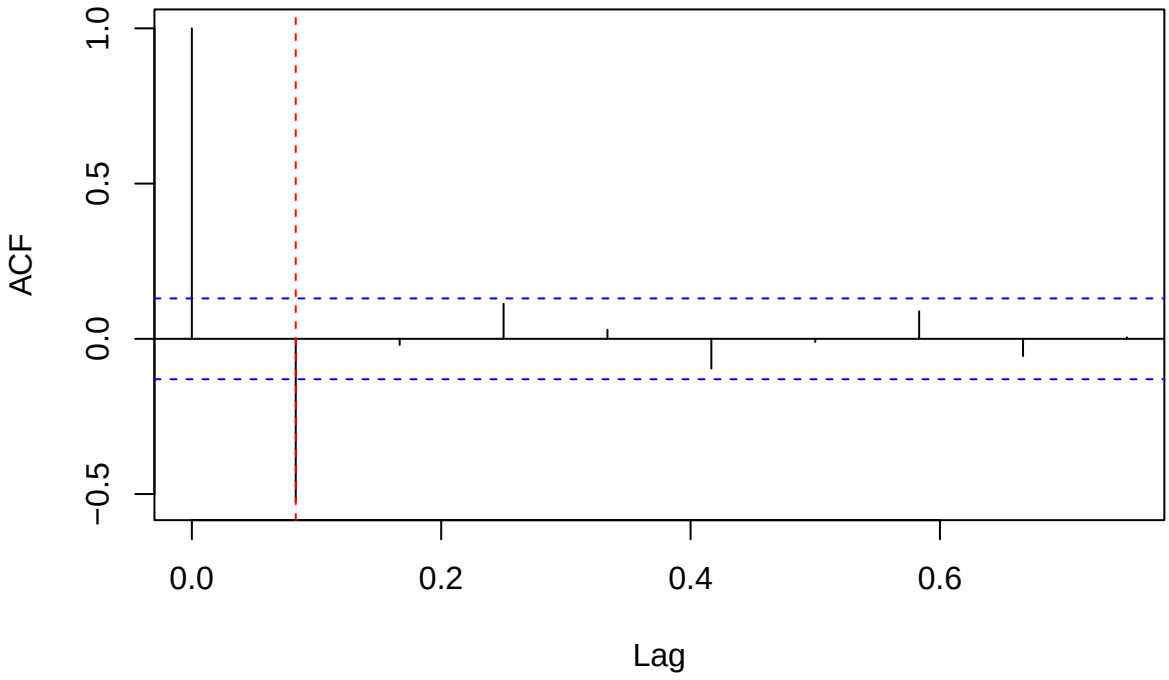
Plot the ACF and PACF of the resulting time series.

```
acf2(X_diff1_12, max.lag=9)
```




```
##       [,1]  [,2]  [,3] [,4]  [,5]  [,6]  [,7]  [,8] [,9]
## ACF  -0.52 -0.02  0.11 0.03 -0.10 -0.01  0.09 -0.06 0.00
## PACF -0.52 -0.40 -0.19 0.02 -0.01 -0.10 -0.02 -0.01 0.02
```

```r
acf_plot <- acf(X_diff1_12, lag.max = 9, plot = FALSE)

plot(acf_plot, main = "ACF of X_diff1_12", xlab = "Lag")
abline(v = 1/12, col = "red", lty = 2)
```



- The above graph was obtained to check if the significant lag in ACF is exactly at lag 1 or not. It shows that a significant spike is indeed at lag 1 and subsequent lags are within the confidence interval. Therefore, the autocorrelations **"cut off"** after the lag 1.
- The PACF plot shows that the first three significant spikes are gradually decreaseed. Therefore, the partial autocorrelations **"decrease/tail off"** gradually over time.
- ACF
  - The ACF plot shows a significant spike at lag 1 (1 month) and subsequent lags are within the confidence interval. Therefore, we conclude that the maximum number of significant lags in the ACF is 1.
- PACF
  - The PACF plots shows that there are spikes outside of the confidence interval until Lag/12=0.25. Therefore, the maximum number of lags in PACF is 12 x 0.25 = 3

Therefore the ARMA model to be fitted to the data should have:

- An AR order of 3 (based on the PACF plot).
- An MA order of 1 (based on the ACF plot).

Thus, the ARMA(3, 1) model would be the appropriate model for the given data based on the identified significant lags in the ACF and PACF plots

```r
ARMA_model <- arima(X_diff1_12, order = c(3, 0, 1),method = "ML")
ARMA_model
```

```
##
## Call:
## arima(x = X_diff1_12, order = c(3, 0, 1), method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3     ma1  intercept
##       -0.8962  -0.6029  -0.2231  0.0914     -1e-04
## s.e.   0.3110   0.2333   0.1338  0.3179      4e-04
##
## sigma^2 estimated as 0.0002575:  log likelihood = 615.54,  aic = -1219.08
```

To determine the significance of the coefficients, we can check if their confidence intervals at a 95% confidence level include 0. If a confidence interval includes 0, the coefficient is not statistically significant at the 5% significance level.

Let us calculate the CI using R codes:

Then,

```r
ar1_lower <- -0.8962 - 1.96 * 0.3110
ar1_upper <- -0.8962 + 1.96 * 0.3110

ar2_lower <- -0.6029 - 1.96 * 0.2333
ar2_upper <- -0.6029 + 1.96 * 0.2333

ar3_lower <- -0.2231 - 1.96 * 0.1338
ar3_upper <- -0.2231 + 1.96 * 0.1338

ma1_lower <- 0.0914 - 1.96 * 0.3179
ma1_upper <- 0.0914 + 1.96 * 0.3179

intercept_lower <- -1e-04 - 1.96 * 4e-04
intercept_upper <- -1e-04 + 1.96 * 4e-04

cat("ar1 CI: [",ar1_lower, ar1_upper, "]", "\n" )
```

```
## ar1 CI: [ -1.50576 -0.28664 ]
```

```r
cat("ar2 CI: [",ar2_lower, ar2_upper, "]", "\n" )
```

```
## ar2 CI: [ -1.060168 -0.145632 ]
```

```r
cat("ar3 CI: [",ar3_lower, ar3_upper, "]", "\n" )
```

```
## ar3 CI: [ -0.485348 0.039148 ]
```

```r
cat("ma1 CI: [",ma1_lower, ma1_upper, "]", "\n" )
```

```
## ma1 CI: [ -0.531684 0.714484 ]
```

```r
cat("intercept CI: [",intercept_lower, intercept_upper, "]", "\n" )
```

```
## intercept CI: [ -0.000884 0.000684 ]
```

- Each confidence interval of coefficients of ar3, ma1, and intercept includes 0 as shown above. Therefore, we conclude that the ar3, ma1, and intercept are not significant.
- On the other hand, the confidence intervals of coefficients of ar1 and ar2 are negative and do not include 0. Therefore, we assume ar1 and ar2 are significant predictors.
- Therefore, our next suggested model would be AR(2).

The codes for our new model is

```r
New_ARMA_model <- arima(X_diff1_12, order = c(2, 0, 0),method = "ML",include.mean=F)
New_ARMA_model
```

```
##
## Call:
## arima(x = X_diff1_12, order = c(2, 0, 0), include.mean = F, method = "ML")
##
## Coefficients:
##           ar1      ar2
##       -0.7314  -0.4005
## s.e.   0.0606   0.0605
##
## sigma^2 estimated as 0.0002671:  log likelihood = 611.44,  aic = -1216.88
```

```r
ar1_lower <- -0.7314 - 1.96 * 0.0606
ar1_upper <- -0.7314 + 1.96 * 0.0606
ar2_lower <- -0.4005 - 1.96 * 0.0605
ar2_upper <- -0.4005 + 1.96 * 0.0605

cat("ar1 CI: [",ar1_lower, ar1_upper, "]", "\n" )
```

```
## ar1 CI: [ -0.850176 -0.612624 ]
```

```r
cat("ar2 CI: [",ar2_lower, ar2_upper, "]", "\n" )
```

```
## ar2 CI: [ -0.51908 -0.28192 ]
```

Since the confidence intervals of the AR(1) and AR(2) coefficients do not include 0, we conclude that all the coefficients: ar1 and ar2 are significant. Therefore, the final suggested model is AR(2).

Based on the significance of the coefficients, the final suggested model is an ARIMA(2,0,0) model (AR(2) model):

$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$

$where \;\; \phi_1 = -0.7314 \;\; and \;\; \phi_2 = -0.4005$

Simulation Study: Yule-Walker Estimator for AR(2) Series

```r
set.seed(8219)
```

In this simulation study, we want to check the asymptotic theory for the Yule–Walker estimator for an AR(2) series with $\phi_1 = 1.5$ and $\phi_2 = -0.75$ by simulating 1000 series of length 100. Let the noise process Z_t be a Gaussian white noise process (with mean zero and variance $\sigma_Z^2 = 4$).

Simulate 1000 time series, each with 100 observations, as mentioned above.

As shown below codes,we simulate 1000 time series each of length 100, using the arima.sim function. Each time series is generated with the specified AR(2) parameters (phi1 = 1.5, phi2 = -0.75) and Gaussian white noise with mean zero and variance 4 (sd = 2).

In each iteration, **compute the Yule-Walker estimator** for the parameters of the model ϕ_1 and ϕ_2.

```r
#initialise vectors to store estimations
a1 = numeric(1000) # to save estimated values for phi_1
a2 = numeric(1000) # to save estimated values for phi_2


armod<-c(1.5, -0.75)
mamod<-NULL # no moving average component.
sdmod<-2
nsamples<-100


#use a loop to simulate 1000 times series and store the estimators
for(i in 1:1000) {
  x<-arima.sim(n = nsamples, list(ar = armod, ma = mamod),
           sd = sdmod)

  # Estimate the AR parameters using the Yule-Walker method
  ar_model <- ar.yw(x, order.max = 2, demean = FALSE)
  a1[i] <- ar_model$ar[1] #store the estimator of a1
  a2[i] <- ar_model$ar[2] #store the estimaior of a2
}
```

- In each iteration, we use the Yule-Walker method (ar.yw function) to estimate the AR parameters from the simulated time series. The estimated values of phi1 and phi2 are stored in vectors a1 and a2 respectively.

Compute the **mean, variance, and covariance** between the vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$.

```r
# Use the vector of estimated values a1 and a2

# Compute mean, variance, and covariance of the estimates
a1_mean <- mean(a1)
a2_mean <- mean(a2)
a1_var <- var(a1)
a2_var <- var(a2)
a1_a2_cov <- cov(a1, a2)

# Display the results
cat("Mean of a1 estimates:", a1_mean, "\n")
```

```
## Mean of a1 estimates: 1.42905
```

```
cat("Mean of a2 estimates:", a2_mean, "\n")
```

```
## Mean of a2 estimates: -0.6873011
```

```
cat("Variance of a1 estimates:", a1_var, "\n")
```

```
## Variance of a1 estimates: 0.007395713
```

```
cat("Variance of a2 estimates:", a2_var, "\n")
```

```
## Variance of a2 estimates: 0.006820539
```

```
cat("Covariance between a1 and a2 estimates:", a1_a2_cov, "\n")
```

```
## Covariance between a1 and a2 estimates: -0.006418188
```

- After the simulation and estimation, we compute the mean, variance, and covariance of the estimated parameters from the vectors a1 and a2. The mean values provide an average estimate of a1 and a2, while the variance and covariance give an indication of the variability and relationship between the estimates.

Compare these values with the theoretical results

```
x_acf = acf(x, plot=FALSE)
attributes(x_acf) #start in lag 0
```

```
## $names
## [1] "acf"    "type"   "n.used" "lag"    "series" "snames"
##
## $class
## [1] "acf"
```

```
acf_matrix=matrix(data=c(1,x_acf$acf[2],x_acf$acf[2],1),nrow=2, byrow=FALSE)
cat("acf matrx:")
```

```
## acf matrx:
```

```
acf_matrix
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.8088297
## [2,] 0.8088297 1.0000000
```

```
rho_hat=matrix(c(x_acf$acf[2],x_acf$acf[3]))
cat("\n","rho_hat:")
```

```
##
##  rho_hat:
```

```
rho_hat
```

```
##           [,1]
## [1,] 0.8088297
## [2,] 0.4442001
```

```
phi_hat=solve(acf_matrix)%*%rho_hat
cat("\n","phi_hat: ")
```

```
##
##  phi_hat:
```

```
phi_hat
```

```
##            [,1]
## [1,]  1.3000422
## [2,] -0.6073127
```

```
cov_matrix_phi_hat =as.vector(1-t(rho_hat)%*%solve(acf_matrix)%*%rho_hat)*solve(acf_matrix) / length(x)
cat("\n","cov_matrix:")
```

```
##
##  cov_matrix:
```

```
cov_matrix_phi_hat
```

```
##              [,1]         [,2]
## [1,]  0.006311713 -0.005105101
## [2,] -0.005105101  0.006311713
```

Using R we get $\hat{\rho}(1) = 0.8528458$, $\hat{\rho}(2) = 0.55898139$. Solving the Yule-Walker equations gives

$$\begin{bmatrix}\hat{\phi}_1\\ \hat{\phi}_2\end{bmatrix} = \hat{R}_p^{-1}\hat{\rho}_p = \begin{bmatrix}1 & \hat{\rho}(1)\\ \hat{\rho}(1) & 1\end{bmatrix}^{-1}\begin{bmatrix}\hat{\rho}(1)\\ \hat{\rho}(2)\end{bmatrix} = \begin{bmatrix}1 & 0.8088297\\ 0.8088297 & 1\end{bmatrix}^{-1}\begin{bmatrix}0.8088297\\ 0.4442001\end{bmatrix} = \begin{bmatrix}1.3000422\\ -0.6073127\end{bmatrix}$$

The asymptotic covariance matrix for the estimates of the autoregressive coefficient is estimated as

$$\Omega(\hat{\phi}) = \left[1 - \hat{\rho}_p^{\top}\hat{R}_p^{-1}\hat{\rho}_p\right]\hat{R}_p^{-1}$$

$$= \left[1 - \begin{bmatrix}0.8088297 & 0.4442001\end{bmatrix}\begin{bmatrix}1.3000422\\ -0.6073127\end{bmatrix}\right]\begin{bmatrix}1 & 0.8088297\\ 0.8088297 & 1\end{bmatrix}^{-1} = \begin{bmatrix}0.006311713 & -0.005105101\\ -0.005105101 & 0.006311713\end{bmatrix}$$

Mean Estimates

From (c), we derive the mean estimations from the simulations:

- Mean of a1 estimates: 1.42905

- Mean of a2 estimates: -0.6873011

Solving the Yule-Walker equations gives: - a1 = 1.3000422 - a2 = -0.6073127 which are reasonably close to the estimates from (c).

Variance and Covariance Estimates

The variances of the estimates of a1 and a2 were 0.007395713 and 0.006820539, respectively, which are close to the theoretical variances of 0.006311713.

The covariance between the estimates a1 and a2 was -0.006418188, which is close to the asymptotic theoretical covariance matrix value of -0.005105101.

In conclusion, the mean estimates of a1 and a2 are reasonably close to the true values. The variance and covariance estimates from the simulation are in good agreement with the theoretical values, validating the asymptotic properties predicted by Lemma 3.1.

Overall, the simulation results support the theoretical findings from Lemma 3.1, demonstrating that the Yule-Walker estimators have the expected asymptotic properties.