Lab 4

Pstat W 174/274

1. Importing Data into R

a. Once you have downloaded the dataset and naming it `monthly-australian-wine-sales-th.csv`, set the working directory using `setwd`.

b. Read the file into an R object using the following command:

```
wine.csv = read.table("monthly-australian-wine-sales-th.csv",
                      sep=",", header=FALSE, skip=1, nrows=187)
```

The arguments of the generic command `read.table` are explained as follows: `header=FALSE` and `skip=1` tells `R` to ignore the first row of the file, `sep=","` tells `R` that elements to be read are separated by commas (since this is a comma separated file), and `nrows=187` specify to read no more than 187 rows (since there is some text at the end of the file that we don't want to read).

`wine.csv` is now a data frame with two columns; the first giving the year and month and the second giving the sales for that month in thousands of liters. One can use the `head` function to dsplay the first few observations of our `data.frame` object.

```
head(wine.csv)
```

```
##        V1   V2
## 1 1980-01  464
## 2 1980-02  675
## 3 1980-03  703
## 4 1980-04  887
## 5 1980-05 1139
## 6 1980-06 1077
```

c. Create a time series object using the following command.

```
?ts # help file for ts()
wine = ts(wine.csv[,2], start = c(1980,1), frequency = 12)
```

`wine.csv[,2]` accesses the second column of `wine.csv`; `start = c(1980,1)` indicates that the first observation of the time series corresponds the first period of 1980; and `frequency = 12` tells `R` that these are monthly data starting from January of 1980.

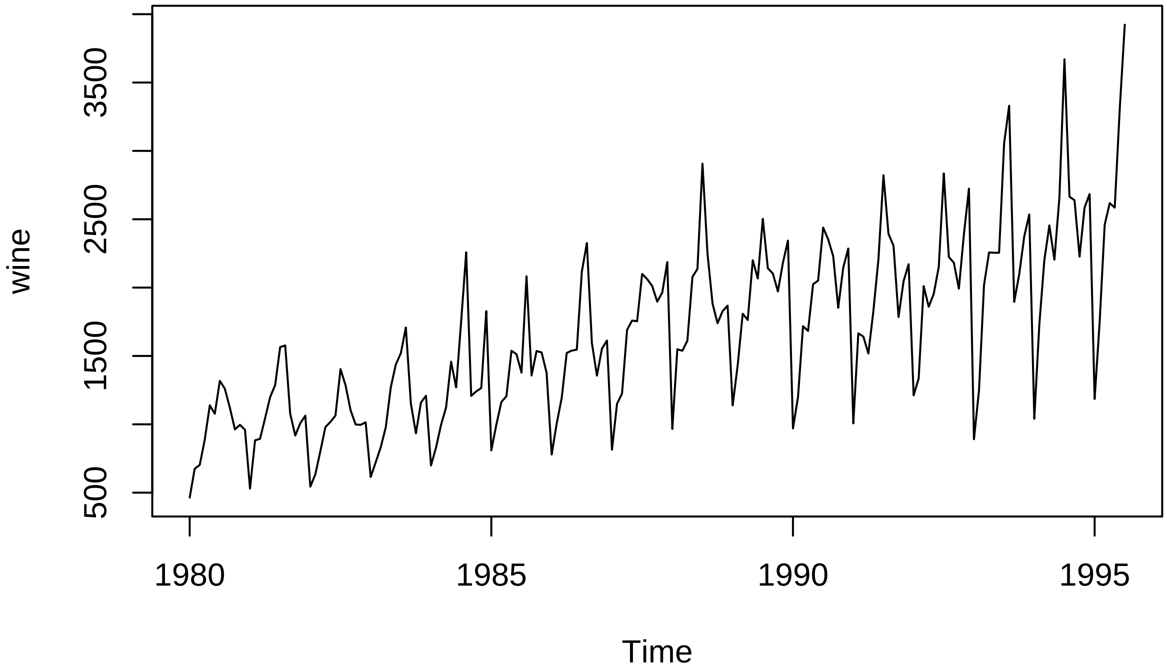
2. Stabilizing the Variance and Removing Trend/Seasonality

a. Plot the time series using `ts.plot(wine)`. What do you notice? Does the variance change over time? Is their a trend and/or seasonal components?

Remarks:

- There is an increasing trend in the data.
- The variability of the data changes according to time (as seen roughly by the changing range of values across different time intervals).
- There is also a strong seasonal component.

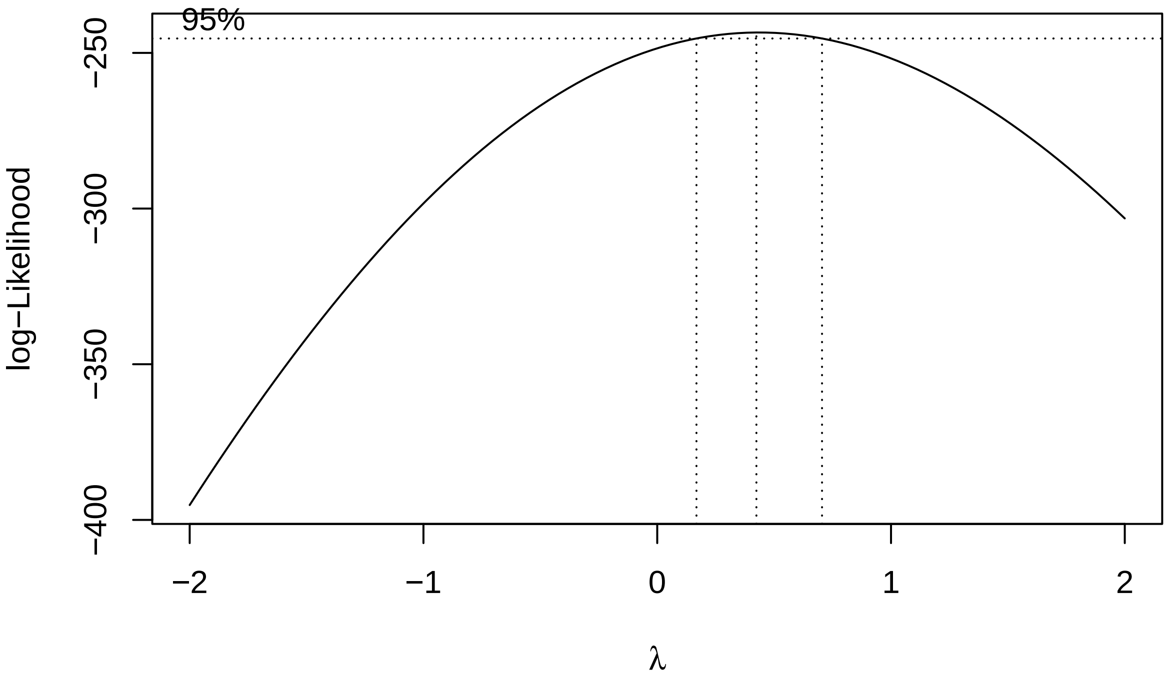
```
ts.plot(wine)
```



b. Apply a Box-Cox transformation to the time series. Use the function `boxcox()` in `R` package `MASS` to find the optimal λ, transform the data, and re-plot the time series. Calculate the sample variance and examine the ACF and PACF. What do you notice? Can you determine the seasonal period from the ACF?

Box-Cox Tranformation

```
library(MASS)
t = 1:length(wine)
fit = lm(wine ~ t)
bcTransform = boxcox(wine ~ t,plotit = TRUE)
```



The dashed vertical lines in the plot above (which is created automatically using the argument plotit = TRUE) correspond to a 95% confidence interval for the true value of λ in the Box-Cox transformation. If the confidence interval includes $\lambda = 0$, then the we can try log transformation (special case of Box-Cox transformation when $\lambda = 0$), given by $Y_t = \log X_t$; otherwise, the Box-Cox transformation for stabilizing the variance is given by:

$$Y_t = \frac{1}{\lambda}(X_t^\lambda - 1);$$

as we implement in the code below:

```
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
wine.bc = (1/lambda)*(wine^lambda-1)
```

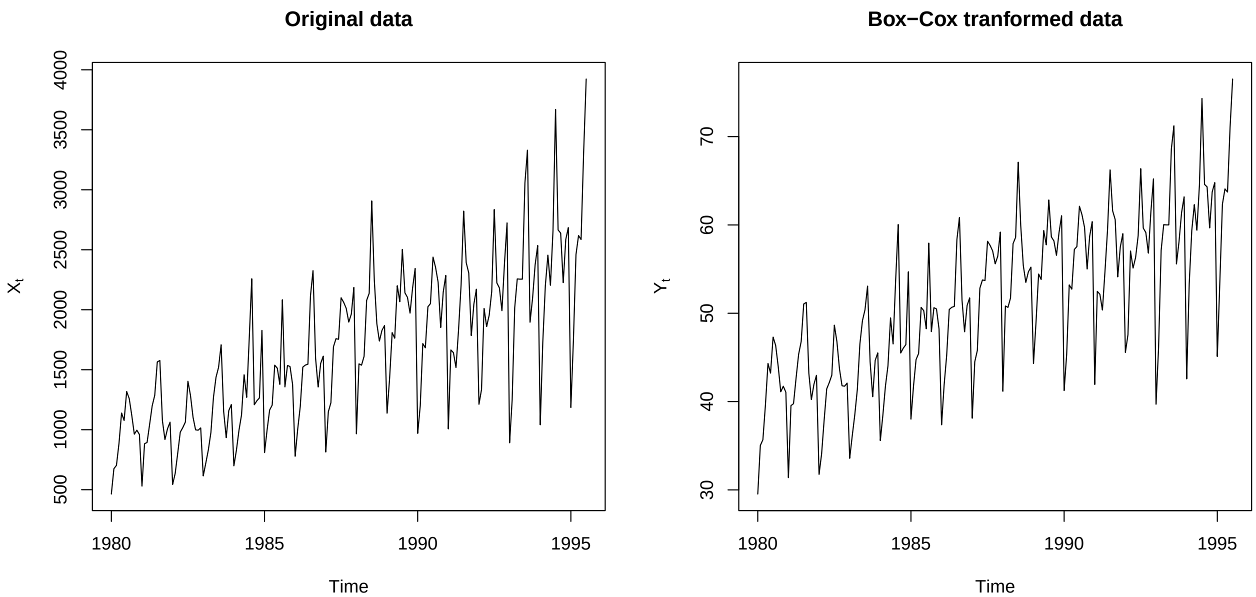
To print out the optimal λ:

```
lambda
```

```
## [1] 0.4242424
```

We now plot the original data vs Box-Cox transformed data:

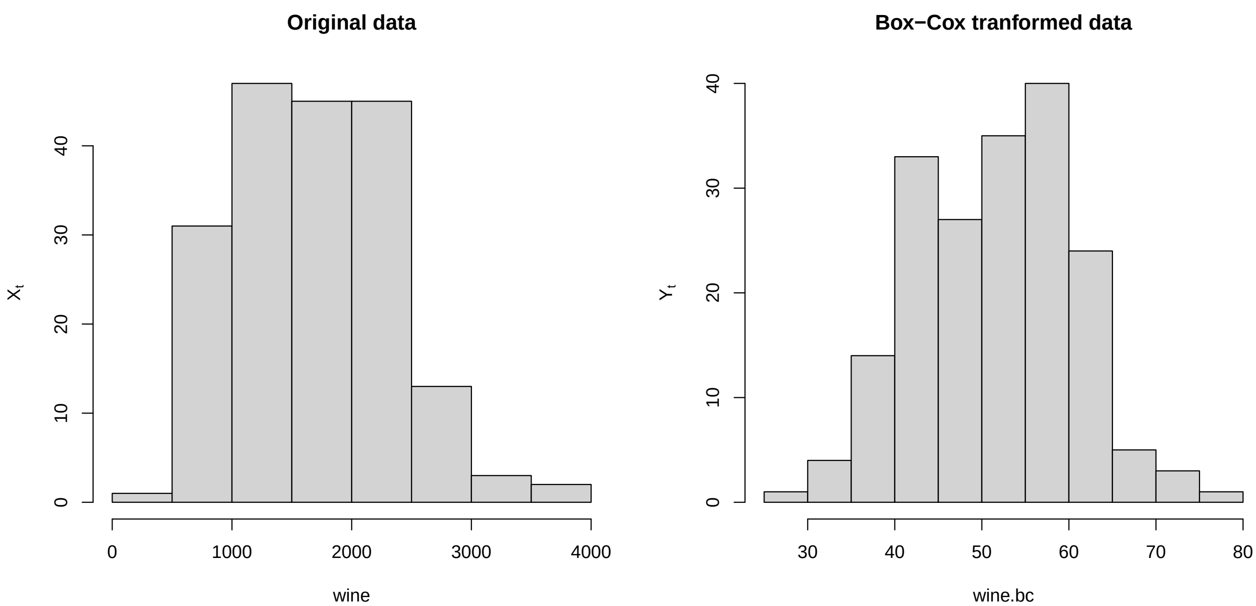
```
op <- par(mfrow = c(1,2))
ts.plot(wine,main = "Original data",ylab = expression(X[t]))
ts.plot(wine.bc,main = "Box-Cox tranformed data", ylab = expression(Y[t]))
```




```
par(op)
```

From the two plots above, the transformed data has a more stable variance acorss time.

```
op <- par(mfrow = c(1,2))
hist(wine,main = "Original data",ylab = expression(X[t]))
hist(wine.bc,main = "Box-Cox tranformed data", ylab = expression(Y[t]))
```




```
par(op)
```

Compared with the histogram of original data, the transformed data is more Gaussian. Therefore, the transformed data is more appropriate than the original.

ACF/PACF of transformed data

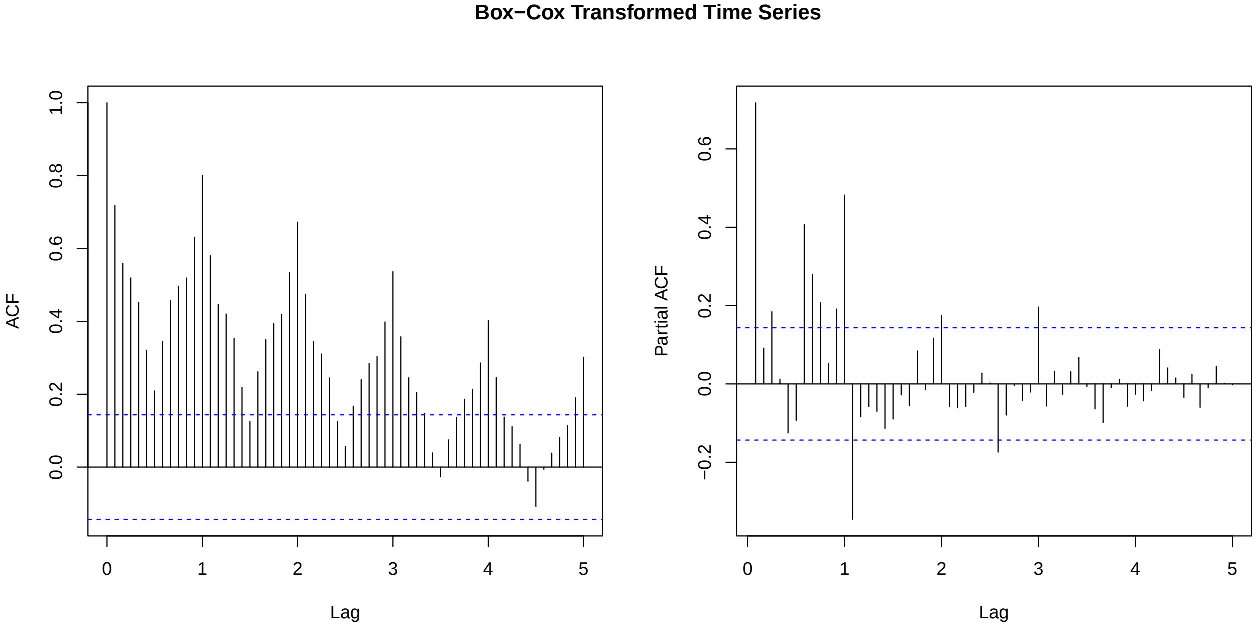
```
# Calculate the sample variance and plot the acf/pacf
var(wine)
```

```
## [1] 421174.6
```

```
var(wine.bc)
```

```
## [1] 83.60385
```

```
op = par(mfrow = c(1,2))
acf(wine.bc,lag.max = 60,main = "")
pacf(wine.bc,lag.max = 60,main = "")
title("Box-Cox Transformed Time Series", line = -1, outer=TRUE)
```

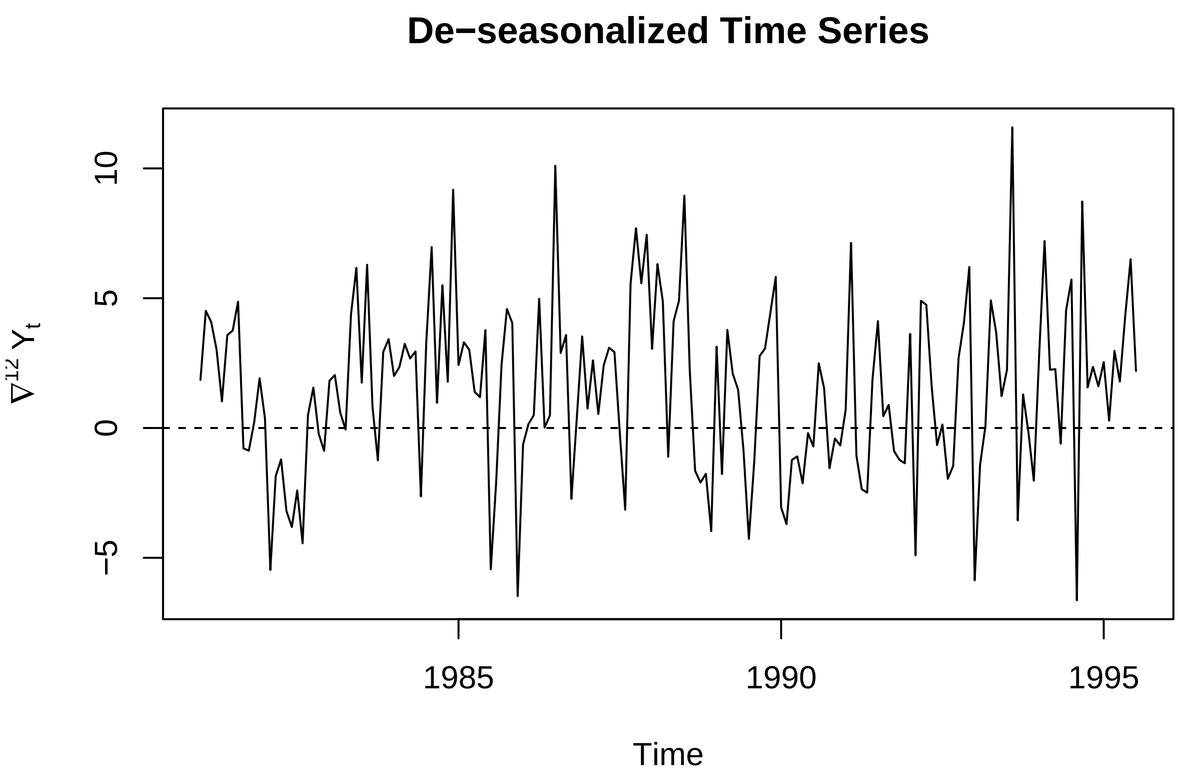



```
par(op)
```

Notice the cyclical behavior in the ACF of the transformed data. Also, notice that there are significant correlations with values moving proportionally every 12 lags. Therefore, we can see that the period of the seasonal component is given by $s = 12$.

c. Remove seasonal components by differencing the transformed time series using the diff() function. Plot the differenced time series. Does it look stationary? Re-calculate the sample variance and examine the ACF and PACF. What do you notice?

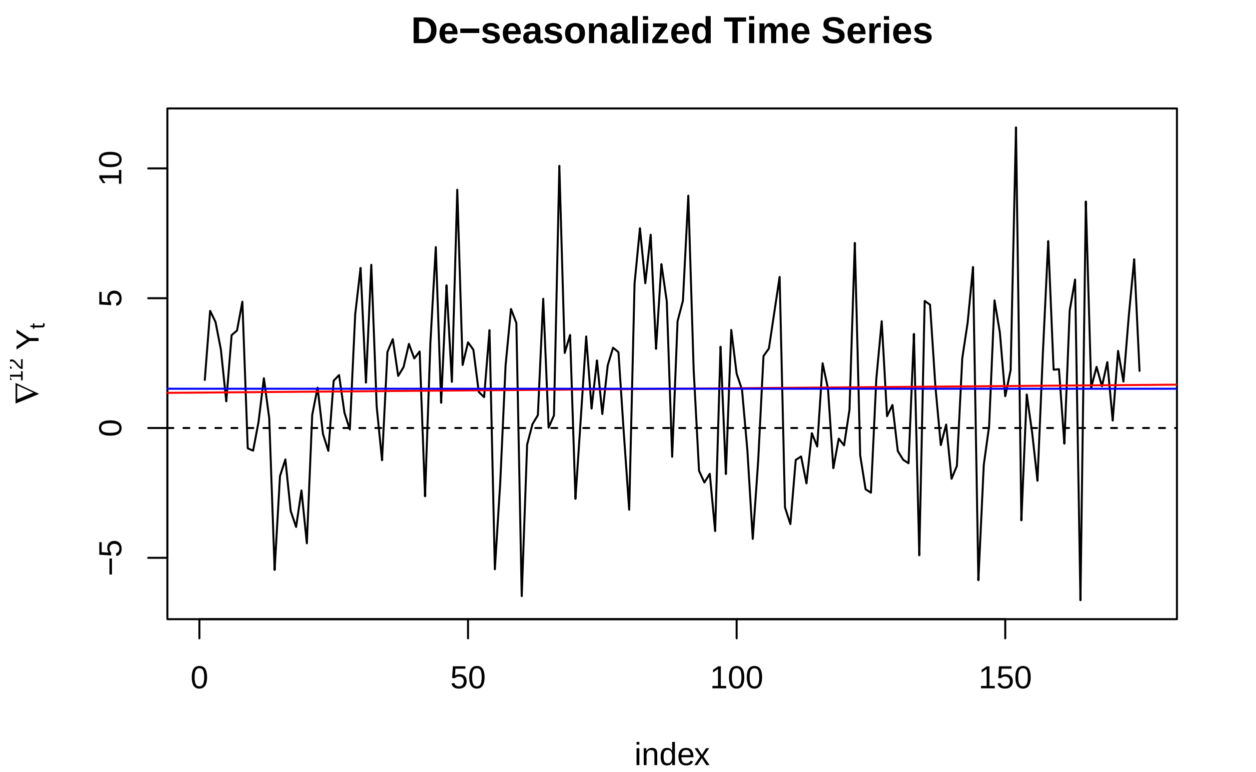
```
# Diference at lag  = 12 to remove seasonal component
y1 = diff(wine.bc, 12)
plot(y1,main = "De-seasonalized Time Series",ylab = expression(nabla^{12}~Y[t]))
abline(h = 0,lty = 2)
```



After differencing to remove seasonal component, we need to assess whether we need to difference to remove trend.

First, we plot the data, add the regression and mean lines:

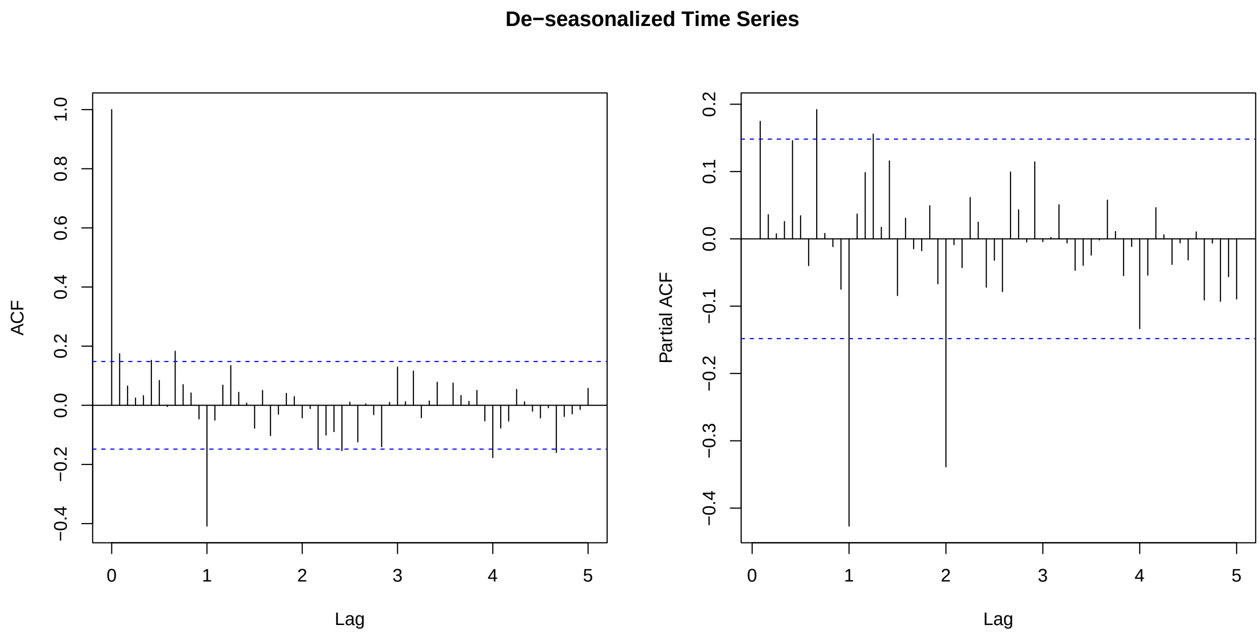
```
plot(1:length(y1),y1, main =
       "De-seasonalized Time Series", type = 'l',xlab='index' ,ylab = expression(nabla^{12}~Y[t]))
index = 1: length(y1)
trend <- lm(y1 ~ index)
abline(trend, col="red")
abline(h=mean(y1) , col='blue')
abline(h = 0,lty = 2)
```



From the two plots above, the regression line is quite horizontal, meaning that we may not have trend now.

Second, we plot acf and see whether it continues to be periodic or large for large lag.

```
# Re-calculate the sample variance and examine the ACF and PACF
op = par(mfrow = c(1,2))
acf(y1,lag.max = 60,main = "")
pacf(y1,lag.max = 60,main = "")
title("De-seasonalized Time Series", line = -1, outer=TRUE)
```



```r
par(op)
```

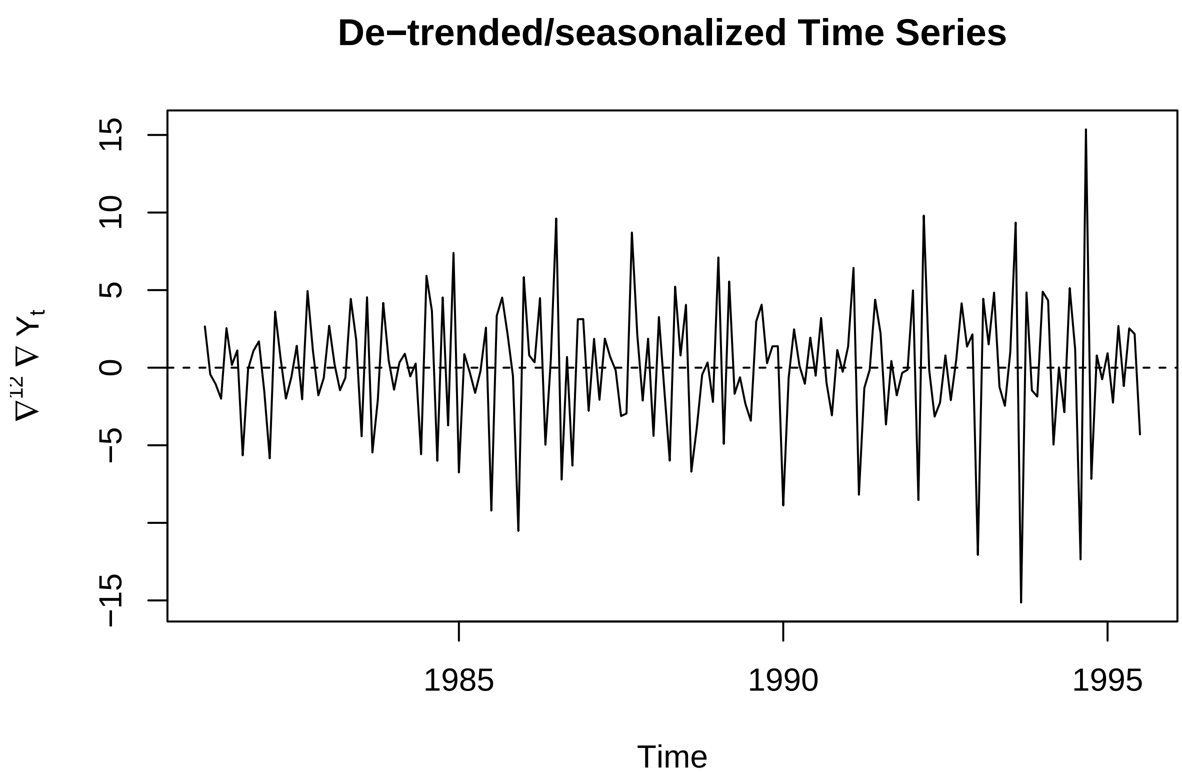
From the two plots, neither ACF nor PACF continues to be periodic or large for large lag. Also, ACF decreases fast, which indicates that we may not have trend here.

From the two plots, we can also identify some significant ACF and PACF values of the de-seasonalized data. First from the sample ACF plot, ACF values at lag 12(1∗s), lag 1 and lag 8 are significant. ACF value at lag 48 (4∗s) is border-line significant. For simplicity, we can choose Q=1 and q=1 for the SARIMA model. Then from the sample PACF, PACF values at lag 12(1∗s), 24(2∗s), 1 and 8 are significant. Therefore, we can choose P=2 and q=1 or 8 for SARIMA model.

d. Remove trend components by using the diff() at lag=1. Plot the differenced time series. Does it look stationary? Re-calculate the sample variance and examine the ACF and PACF.

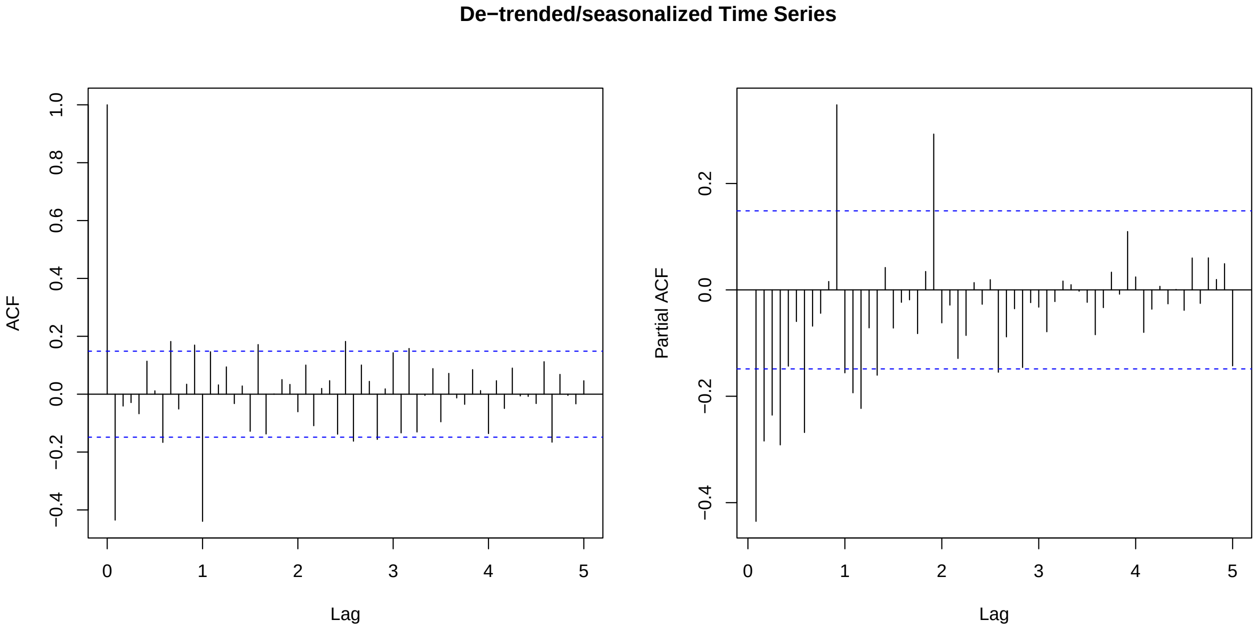
We attempt to remove trend and see what is the result:

```r
# Difference at lag  = 1 to remove trend component
y12 = diff(y1, 1)
ts.plot(y12,main = "De-trended/seasonalized Time Series",ylab = expression(nabla^{12}~nabla~Y[t]))
abline(h = 0,lty = 2)
```



ACF of de-trended/de-seasonalized time series $\nabla^{12}\nabla Y_t$:

```r
# Re-calculate the sample variance and examine the ACF and PACF
op = par(mfrow = c(1,2))
acf(y12,lag.max = 60,main = "")
pacf(y12,lag.max = 60,main = "")
title("De-trended/seasonalized Time Series",line = -1, outer=TRUE)
```




```
par(op)
```

We need to keep track of the variance at each step:

```
#variance of the transformed data
var(wine.bc)
```

```
## [1] 83.60385
```

```
#variance of the de-seasonalized data
var(y1)
```

```
## [1] 10.95448
```

```
#variance of the de-trended/seasonalized data
var(y12)
```

```
## [1] 18.18436
```

From the variance information, variance of the de-seasonalized data achieves the lowest variance. Therefore, we do not need to de-trend the data. De-seasonalized data is enough for further analysis.