Exploratory analysis of time series and decomposition

Usue Mori

3 de noviembre de 2022

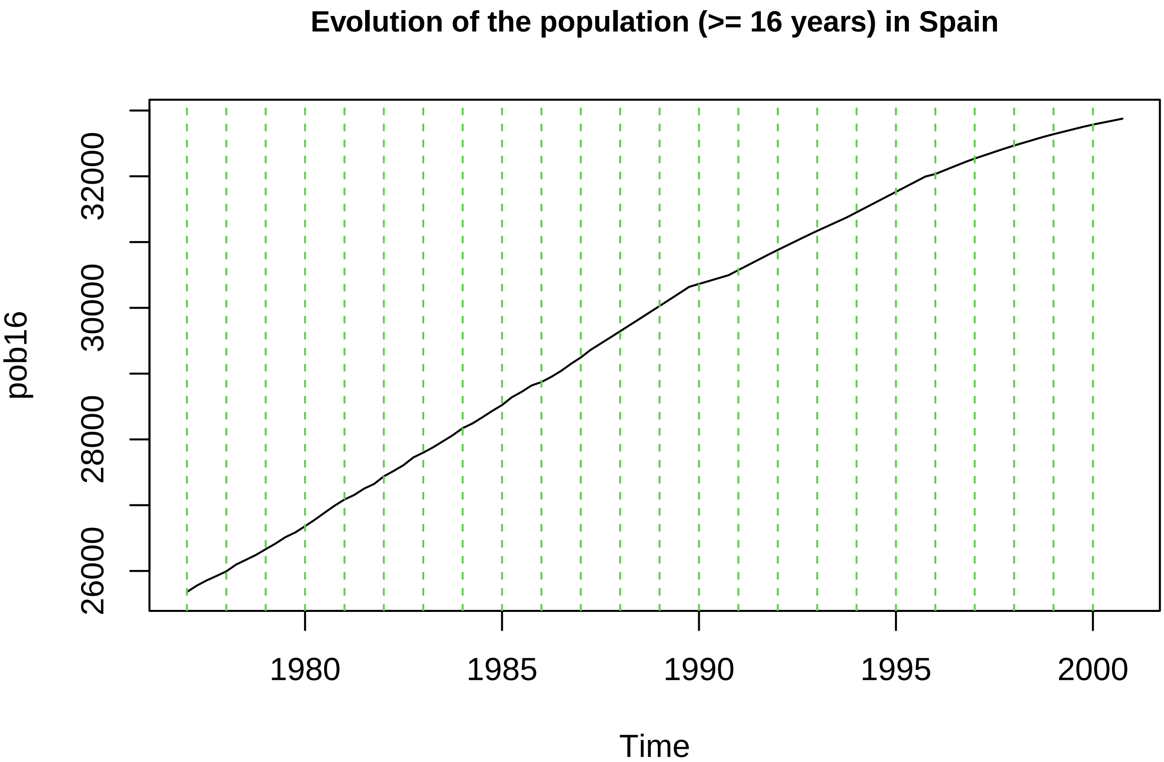
In this tutorial, we will learn how to perform a classical exploratory analysis of a time series by using R. Note that we will perform this exploratory analysis on the raw time series.

1. Definition of a time series object and graphical representation

First, we will load 3 different time series with different types of trend and seasonality characteristics:

- We load and graphically represent the trimestral time series that measures the evolution of the Spanish population (considering only those older that 16) between the years 1977 and 2000.

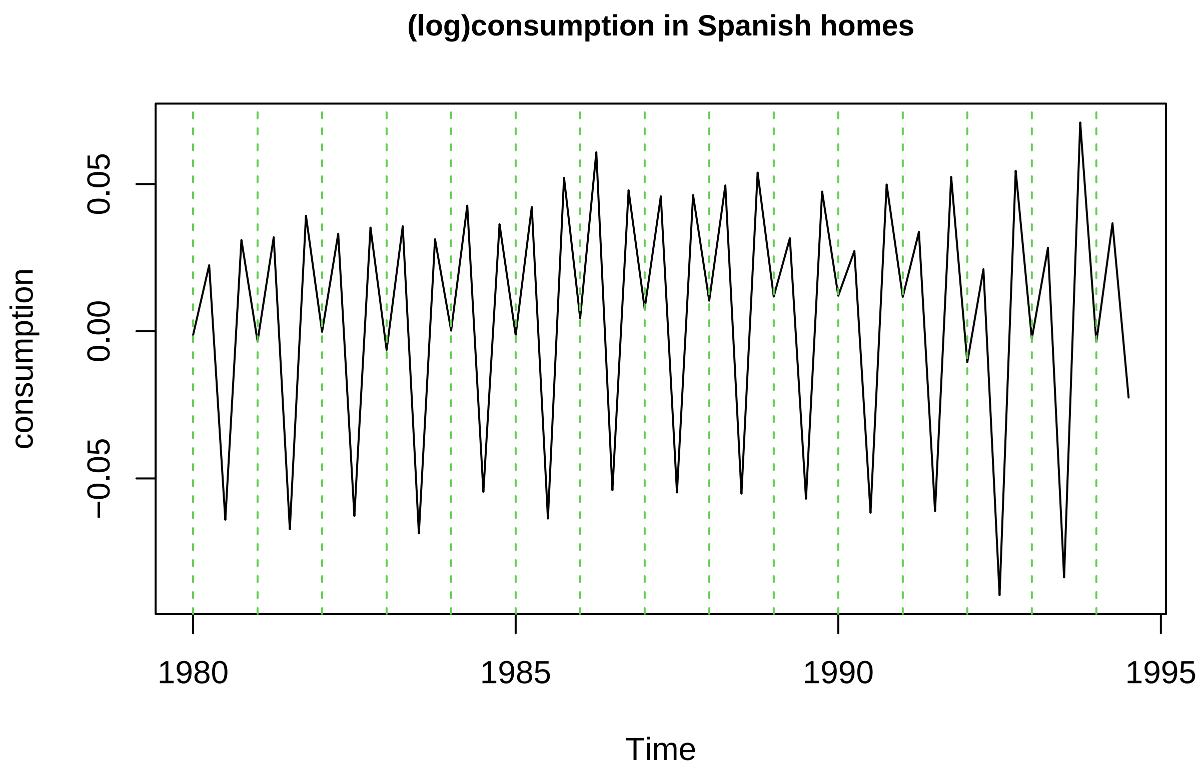
```
pob16 <- read.table(file="./Data/pobmay16.txt")
pob16<-ts(pob16[,2], start=1977, frequency=4)
plot(pob16,main="Evolution of the population (>= 16 years) in Spain", cex.main=0.9)
abline(v=c(1977:2000), lty=2, col=3)
```



Based on this graphical representation, we can conclude that the series appears to have a linear increasing trend.

- We load and represent the trimestral time series that represents the (log)consumption in Spanish homes in the period between 1980 and 1994:

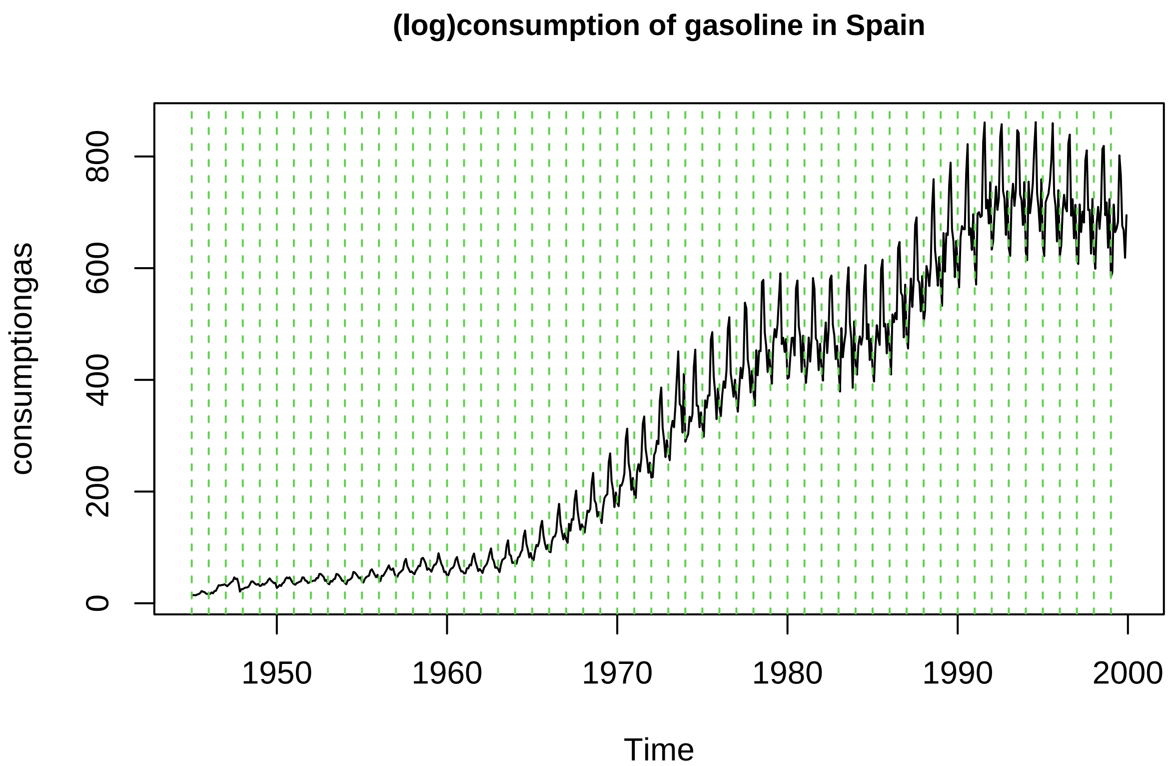
```
consumption <- read.table(file="./Data/consumoE.txt", header=TRUE, sep=" ")
consumption<-ts(consumption[,4], start=1980, frequency=4)
plot(consumption,main="(log)consumption in Spanish homes", cex.main=0.9)
abline(v=c(1980:1994), lty=2, col=3)
```



This graphical representation does not show any clear trend, but the series does appear to have a clear seasonality (trimestral).

- We load the data that measures the monthly (log)consumption of gasoline in Spain in the period between 1945 and 1999:

```
consumptiongas <- read.table(file="./Data/gasolauto.txt")
consumptiongas<-ts(consumptiongas[,2], start=1945, frequency=12)
plot(consumptiongas, main="(log)consumption of gasoline in Spain", cex.main=0.9)
abline(v=c(1945:1999), lty=2, col=3)
```



In this case, the series shows a clear non-linear increasing trend and, additionally, it also appears to have a seasonal component (montly) which shows a higher consumption in the summer months.

2. Estimating the trend component

In this first section, we will focus on estimating the trend of series `pob16`, which we have loaded in the previous section.

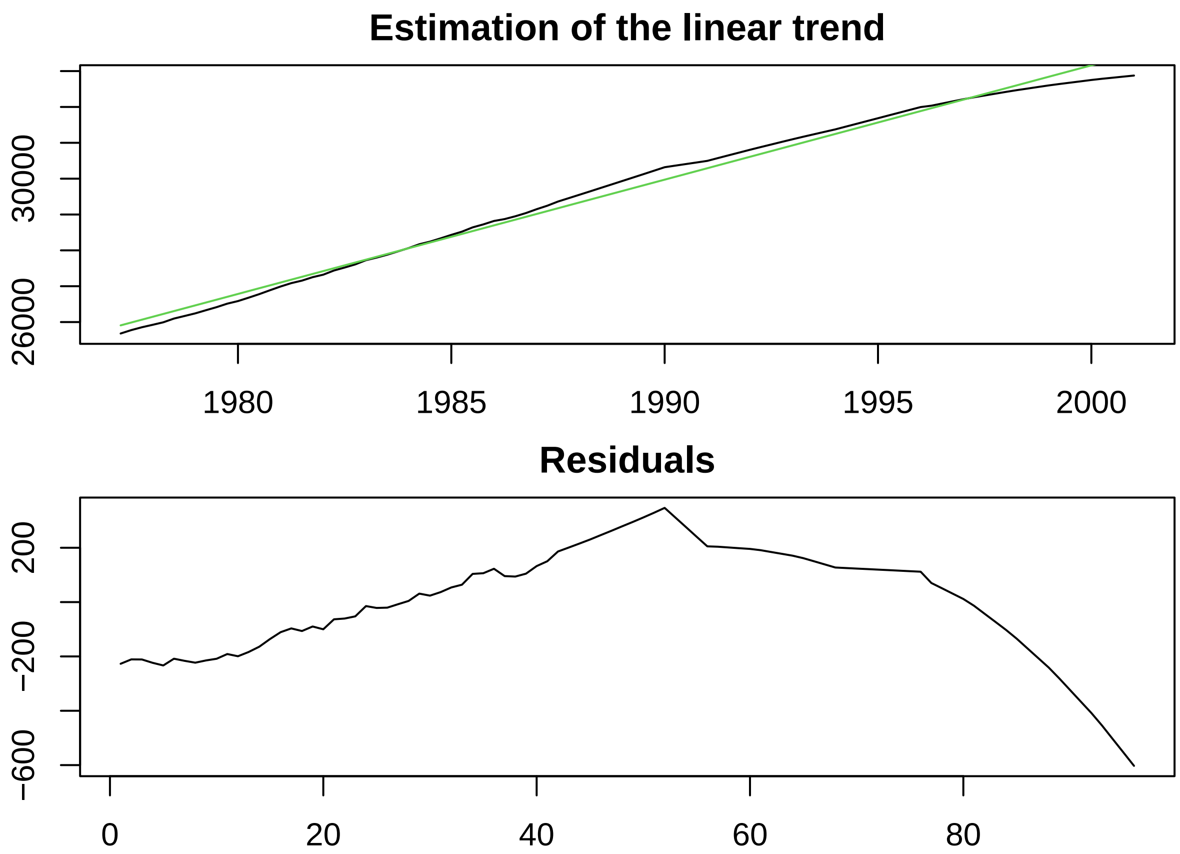
2.1. Estimating deterministic trends

As stated previously, the series `pob16` appears to have a linear increasing trend. In order to find the linear function that best fits the data, we will use the `lm` function, which can be used to fit linear models based on the least squares method:

```
t<-seq(from=1977.25, to=2001, by=0.25)
lmpob16<-lm(pob16~t)
```

After fitting the model, we represent the regression line together with the original series. We also represent the series that is left if we remove this estimated linear trend (the residuals):

```
par(mfrow=c(2,1), mar=c(2,2,2,2))
plot(t, pob16, type="l", ylab="", main="Estimation of the linear trend")
lines(t, lmpob16$fitted,col=3)
plot(as.ts(lmpob16$residuals), main="Residuals")
```



In order to test the stationarity of the residuals, we first have to analyze them graphically. In this case, the linear trend fits quite well around the middle of the series but the error is higher at the beginning and end. This can be seen in the residuals, which do not seem stationary at first sight. As extra support, in order to show that we have estimated the trend correctly, we can use tests such as the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, which is contained in package `tseries`. The null hypothesis of this test is that the series is level stationary (i.e. the trend is a constant):

```
library(tseries)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

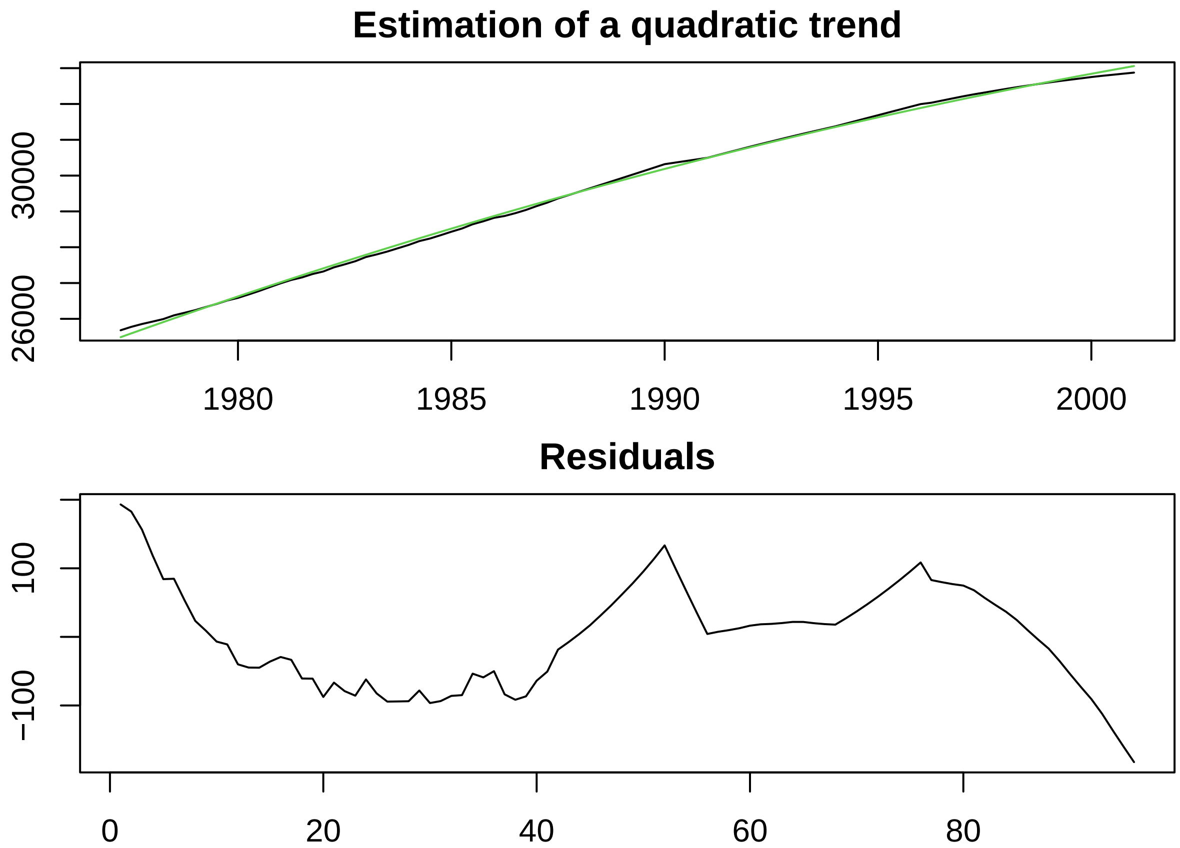
```
kpss.test(lmpob16$residuals)
```

```

    KPSS Test for Level Stationarity

data:  lmpob16$residuals
KPSS Level = 0.57166, Truncation lag parameter = 3, p-value = 0.02553
```

In this case, the p-value is quite small, so we can reject the null hypothesis; it does not seem that the obtained residuals are stationary. As such, in order to try to estimate the trend better, we will fit a polynomial of degree 2:

```
lm2pob16 <-lm(pob16~poly(t,2))
```

```
par(mfrow=c(2,1), mar=c(2,2,2,2))
plot(t, pob16, type="l", ylab="", main="Estimation of a quadratic trend")
lines(t, lm2pob16$fitted,col=3)
plot(as.ts(lm2pob16$residuals), main="Residuals")
```




```
kpss.test(lm2pob16$residuals)
```

```
Warning in kpss.test(lm2pob16$residuals): p-value greater than printed p-value


    KPSS Test for Level Stationarity

data:  lm2pob16$residuals
KPSS Level = 0.23274, Truncation lag parameter = 3, p-value = 0.1
```

In this case, the residuals seem more stationary, when we analyze them graphically, and the obtained p-value is high. So we can not discard stationarity of the residuals.

2.2. Estimating the trend with moving averages

Moving averages are one of the most common methods to estimate the trend of a time series. In this section we will attempt to estimate the trend of the pob16 series by using moving averages of different order. In R, this can be done easily by using the ma function, contained in the package forecast. To perform simple moving averages the only parameter to specify will the the size of the window; the larger the window the smoother the estimated trend.

To begin with we will try with a window size of 5. This means that, to estimate the trend in timestamp t, we will average the values at timestamps $\{t-2, t-1, t, t+1, t+2\}$:

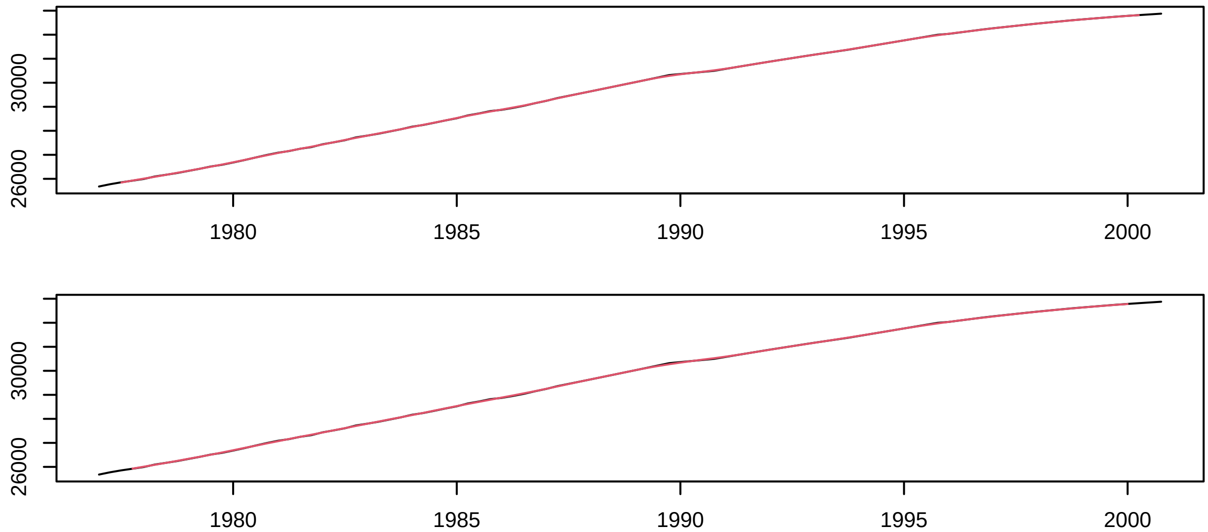
```
library(forecast)
pob16ma2 <- ma(pob16, order=5)
```

Similarly we can try with a window size of 7:

```
pob16ma3 <- ma(pob16, order=7)
```

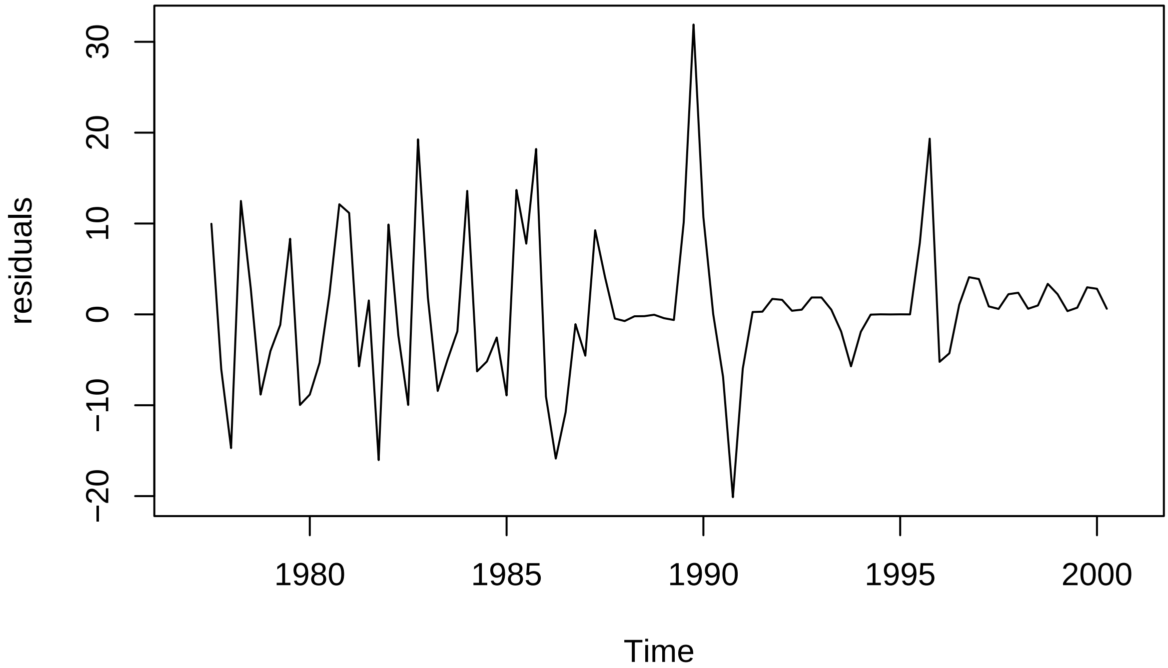
We will now plot the original time series with the estimated trends:

```
par(mfrow=c(3,1), mar=c(2,2,2,2))
plot(pob16, ylab="")
lines(pob16ma2, col=2)
plot(pob16, ylab="")
lines(pob16ma3, col=2)
```



As can be seen, both estimations adjust almost perfectly to the trend of the series. In this situation, since for larger windows we loose more points at the beginning and end of the series, we will keep the estimation obtained with the smallest window (w=5). We will represent the residuals and apply the KPSS test in order to see whether we can assume that they are stationary:

```
residuals <- ts(pob16-pob16ma2, start=1977, frequency=4)
plot(residuals)
```




```
kpss.test(residuals)
```

```
Warning in kpss.test(residuals): p-value greater than printed p-value


    KPSS Test for Level Stationarity

data:  residuals
KPSS Level = 3.4751e-15, Truncation lag parameter = 3, p-value = 0.1
```

The residuals seem quite stationary graphically, conclusion which can not be discarded based on the p-value obtained in the test.

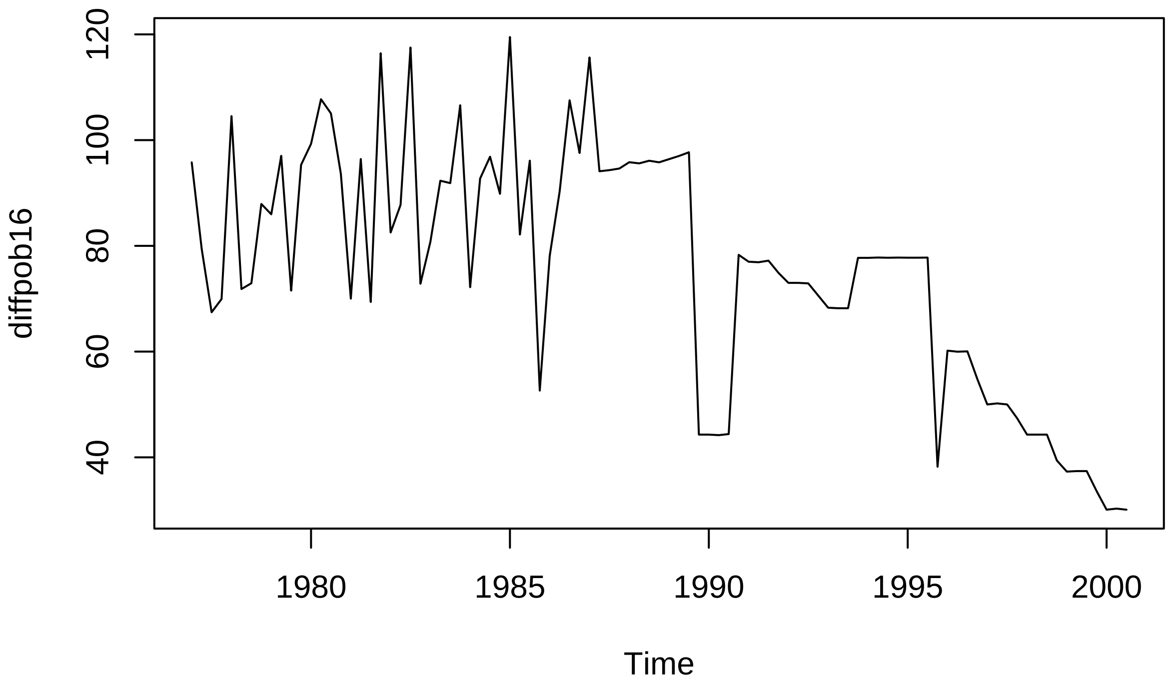
2.3. Elimination of the trend by differentiation

Instead of estimating the trend we can remove it by using differentiation. We can do this by using the function `diff` in R. For example, for series `pob16`:

```
diffpob16 <- diff(pob16)
```

We analyze the remaining series to analyze if it is stationarity:

```
diffpob16 <- ts(diffpob16, start=1977, frequency=4)
plot(diffpob16)
```



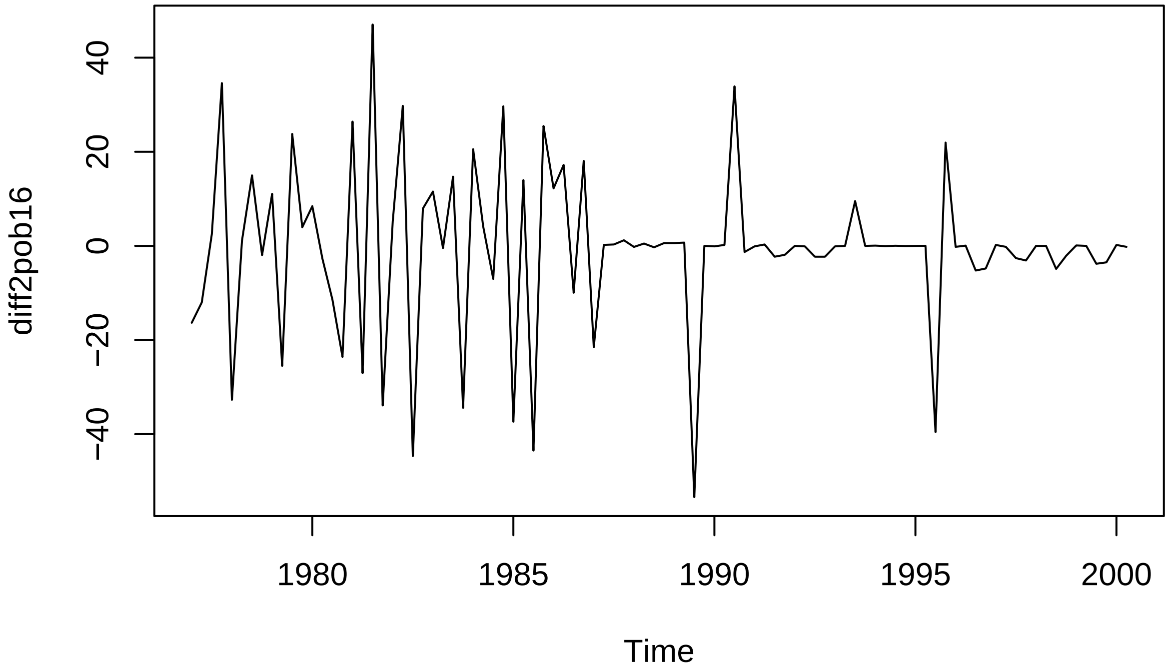

```
kpss.test(diffpob16)
```

```
Warning in kpss.test(diffpob16): p-value smaller than printed p-value
```

```

    KPSS Test for Level Stationarity

data:  diffpob16
KPSS Level = 1.6037, Truncation lag parameter = 3, p-value = 0.01
```

If we analyze the graphical representation of the differenced time series, we can observe that it does not seem stationary. This is confirmed by the p-value obtained in the KPSS test. As such, in order to obtain a stationary time series we will apply differentiation again:

```
diff2pob16 <- ts(diff(diffpob16), start=1977, frequency=4)
plot(diff2pob16)
```




```
kpss.test(diff2pob16)
```

```
Warning in kpss.test(diff2pob16): p-value greater than printed p-value


    KPSS Test for Level Stationarity

data:  diff2pob16
KPSS Level = 0.042915, Truncation lag parameter = 3, p-value = 0.1
```

This time it seems that we have eliminated the trend succesfully.

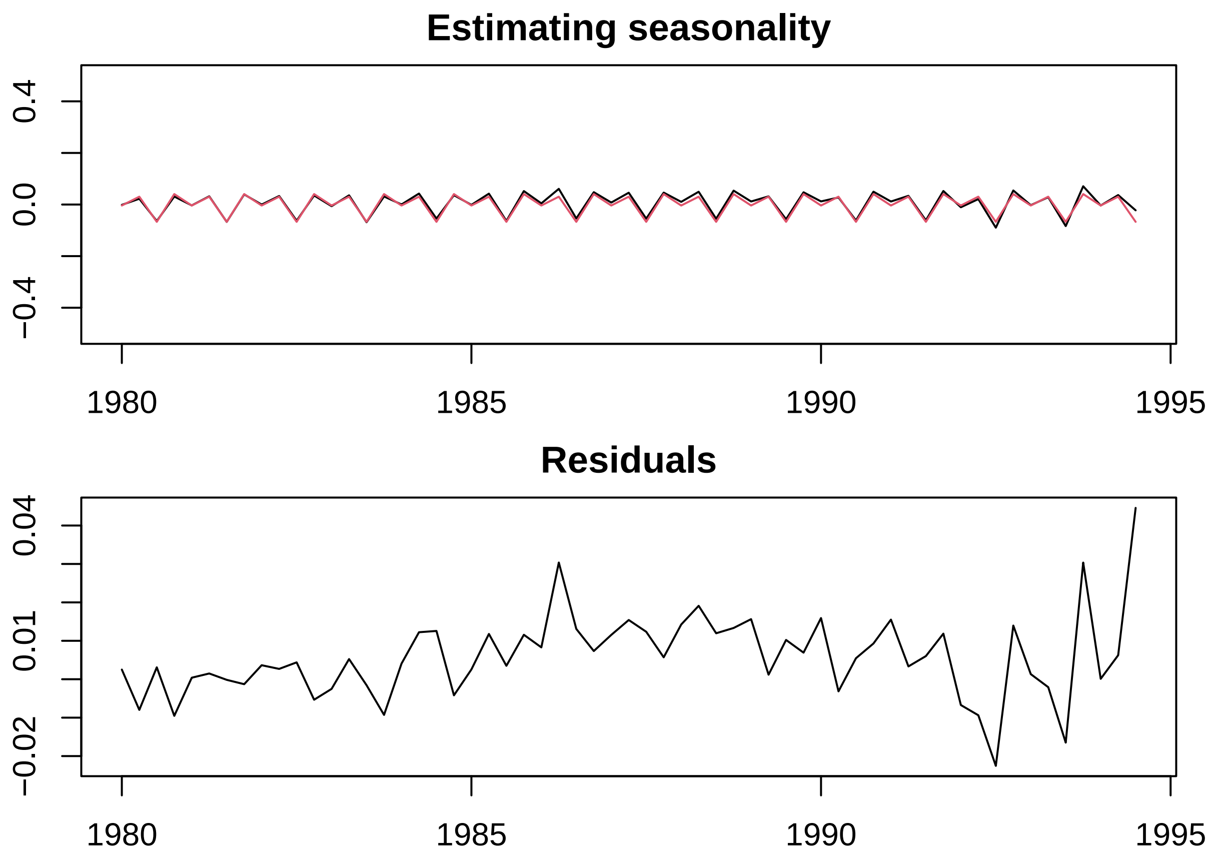
3. Estimating seasonality

In order to estimate the seasonal component of a series we can apply the basic method based on calculating the seasonal means. In this method, we calculate the mean of the observations in the time series with the same seasonal period and standarize them. Let's try, for example with the `consumption` time series:

```
trimestralmeans <- tapply(consumption, cycle(consumption), mean)
standarizedst <- trimestralmeans - mean(trimestralmeans)
```

Finally, we define a time series with the calculated seasonal component and another with the residual.

```
seasonalcomponent <- ts(rep(standarizedst , length.out=length(consumption)), start=1980,
                    frequency=4)
residuals <- consumption - seasonalcomponent
par(mfrow=c(2,1), mar=c(2,2,2,2))
plot(consumption, type="l", ylab="", main="Estimating seasonality", ylim=c(-0.5, 0.5))
lines(seasonalcomponent, col=2)
plot(residuals, main="Residuals")
```

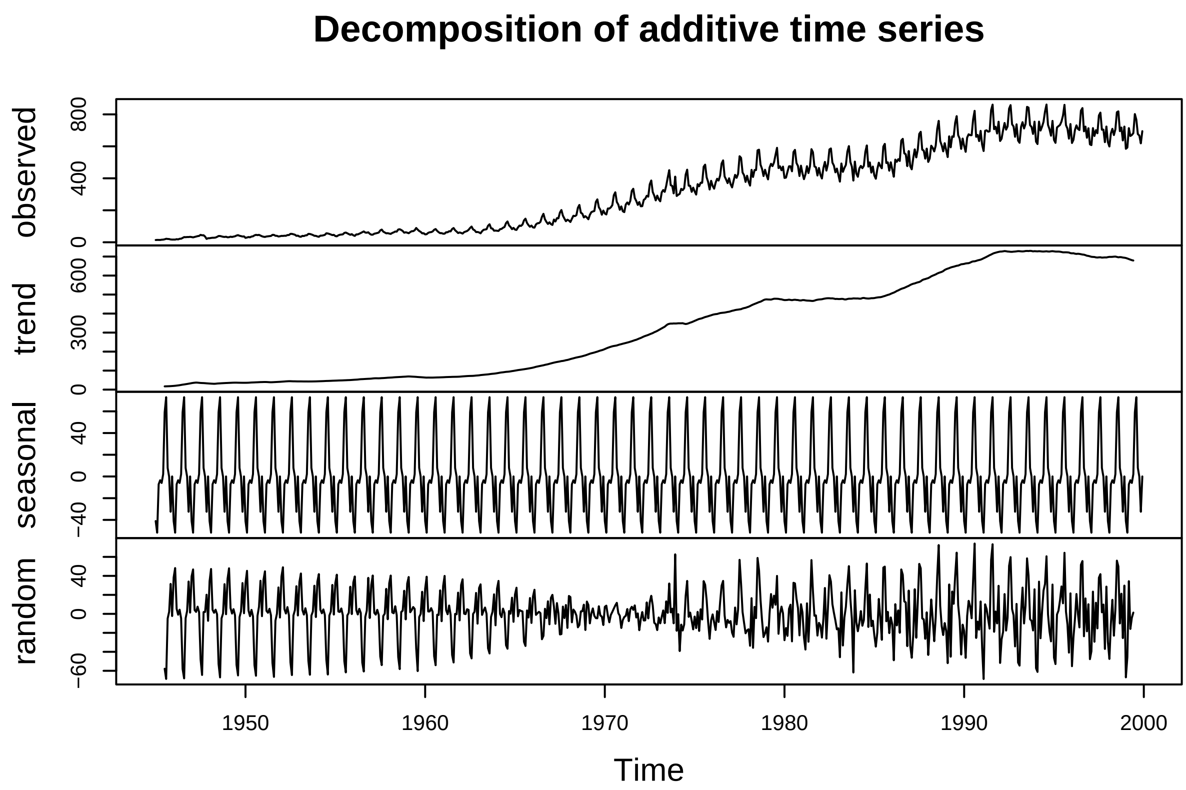


4. Estimating both the trend and seasonality using the classical method

In R, if the series we are analyzing has both a trend component and a seasonal component we can obtain the whole decomposition (additive or multiplicative) using the `decompose` function. This function applies the classical decomposition method, which originated in 1920 and is the precursor to other many more complex decomposition methods. This method first eliminates the trend using a suitable moving average and, then, applies the method of seasonal means to the detrended series to obtain the seasonal component.

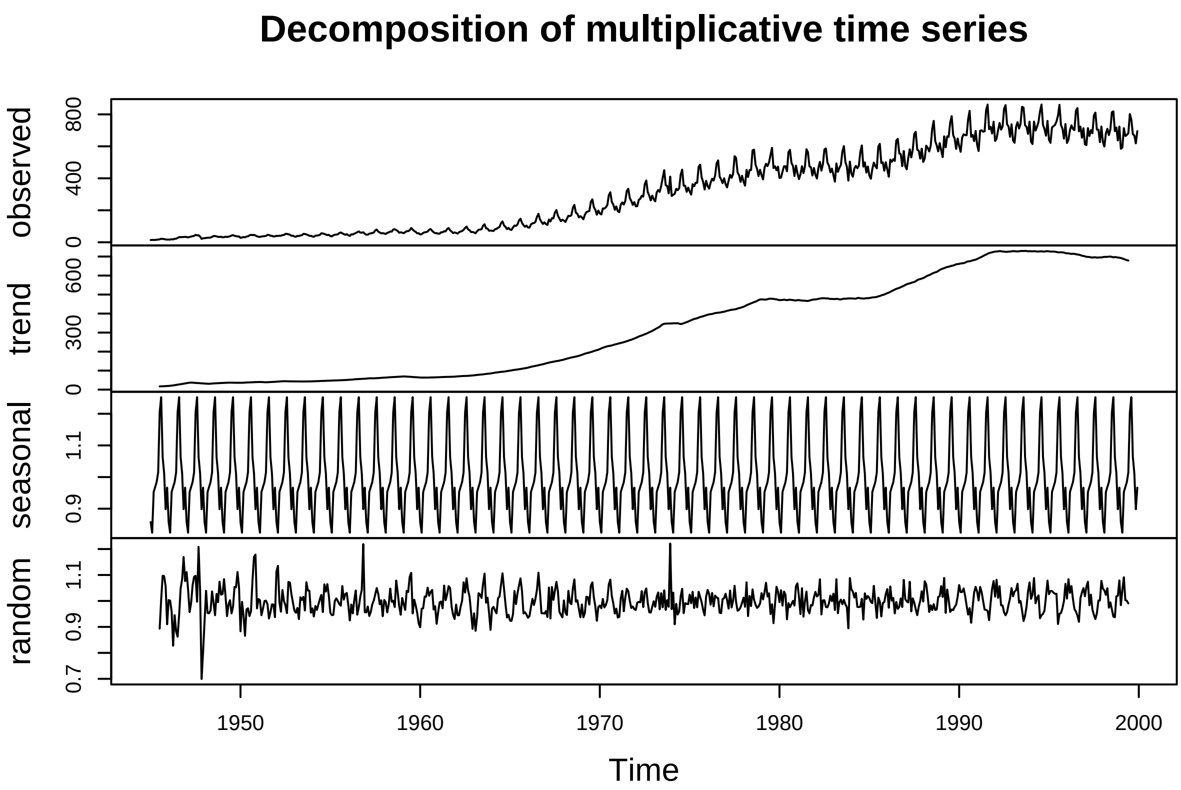
For example, we can obtain the additive decomposition of the `consumptiongas` series as follows:

```
descomposition <- decompose(consumptiongas, type="additive")
plot(descomposition)
```



Similarly we can obtain the multiplicative decomposition:

```
descomposition2 <- decompose(consumptiongas, type="multiplicative")
plot(descomposition2)
```



The different component of the series can be accessed as components of the object created with the `decompose` function and their names are `trend`, `seasonal` and `random` (residuals), as can be seen in the graphics. Based on this, from both decomposition, the multiplicative seems to obtain more stationary residuals. We could analyze the stationarity of the residuals with the KPSS test as follows:

```
kpss.test(na.omit(descomposition$random))
```

```
Warning in kpss.test(na.omit(descomposition$random)): p-value greater than
printed p-value


    KPSS Test for Level Stationarity

data:  na.omit(descomposition$random)
KPSS Level = 0.0055032, Truncation lag parameter = 6, p-value = 0.1
```

```
kpss.test(na.omit(descomposition2$random))
```

```
Warning in kpss.test(na.omit(descomposition2$random)): p-value greater than
printed p-value


    KPSS Test for Level Stationarity

data:  na.omit(descomposition2$random)
KPSS Level = 0.12971, Truncation lag parameter = 6, p-value = 0.1
```

In both cases we have obtained a large p-value so we can not discard stationarity, however, as stated before, the multiplicative decomposition seems more suitable.

5. Exercises

In the file named Data/ParoCCAA.Rdata we have the unenplouyment rates in the period 1977-2009 in the different autonomous communities.

- Load the data and save the column that corresponds to Catalonia as a time series. This is the series that we will analyze.
- Represent the series graphically. What can we say about the trend and seasonality?
- Estimate the trend by using the different methods we have analyzed in class (with different parameters). Which is the most suitable?
- Estimate the seasonal component by using the seasonal means. What can we conclude?
- When we eliminate the trend and seasonality (if it exists), do we obtain a stationary time series?
- Eliminate the trend from the original series by applying differentiation. How many time have you applied the operator?
- Obtain a full decomposition of the series by using the classical decomposition method. Is this decomposition similar to the one obtained in the previous sections? Are the residuals stationary?

6. Extra work

- In all these examples and exercises, when applying moving averages, we have taken an even sized window (symmetric moving average) and performed the most simple moving average type. However, more complex moving averages can also be applied, which can be useful in more complex situations. Go to `https://otexts.com/fpp2/moving-averages.html`, read the information concerned there and think about the following aspects:
  - Why shouldn't we use any type of moving average with seasonal data? What possible solutions do we have for this type of data?
  - What happens when we use even ordered moving averages? How can we solve this issue?
- In these examples we have perfomed the classical decomposition using the `decompose` method. This method is explained in depth in `https://otexts.com/fpp2/classical-decomposition.html`. Read the information contained there carefully and find out what are the disadvantages or limitations of this method. In addition, find out about other decomposition methods and try to apply them to the series of unenplouyment in Catalonia.

7. References

- P. J. Brockwell and R. A. Davis. Introduction to Time Series and Forecasting. Springer Verlag, 1996.
- P. S.P. Cowpertwait and A. V. Metcalfe. Introductory Time Series with R (Use R). Springer, 2009.
- R. H. Shumway and D. S. Stoffer. Time Series Analysis and Its Applications With R Examples. Springer Verlag, 2006.
- Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on 27/10/2022.