Time Series Analysis and Forecasting - Exercise Set 1

Francesca Sallicati

08 marzo 2018

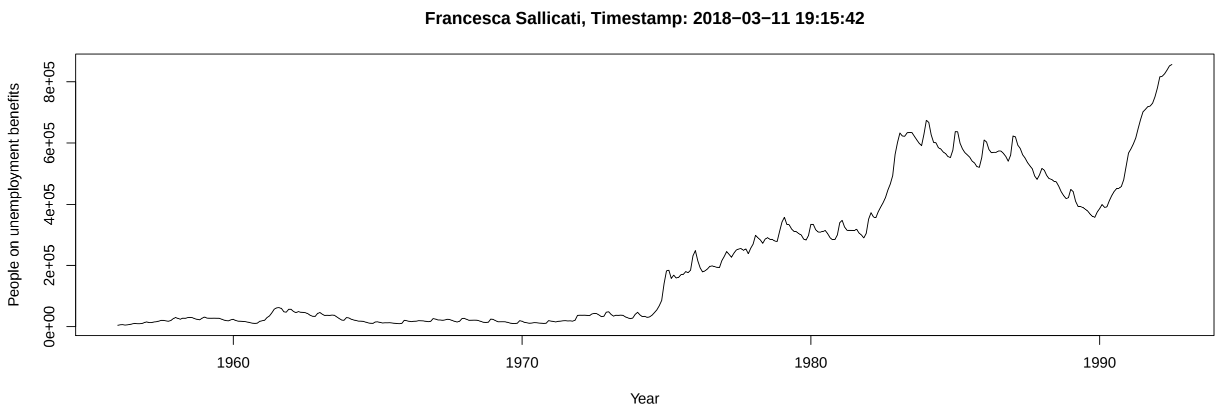
Contents

1 Exercise 1

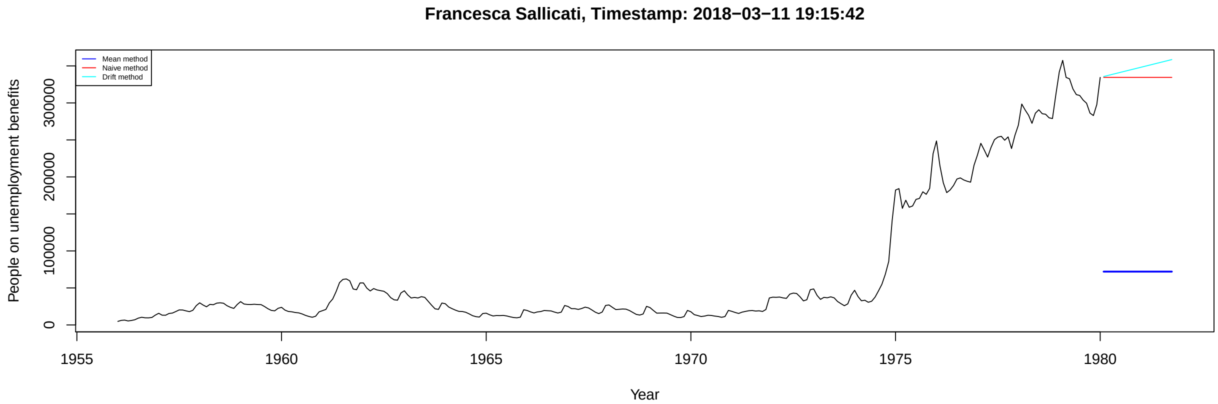
For each of the following series, make a graph of the data with forecasts using the most appropriate of the four benchmark methods: mean, naive, seasonal naive or drift.

a) Monthly total of people on unemployed benefits in Australia (January 1956 - July 1992).Data set dole.

```
data(dole)
dole1 <- window(dole, start = 1956)
par(mfrow = c(1, 1))
plot(dole1, xlab = "Year", ylab = "People on unemployment benefits",main=paste("Francesca Sallicati, Ti
```




```
dole2 <- window(dole,start=1956,end=1980)
dolefit1 <- meanf(dole2, h=21)
dolefit2 <- naive(dole2, h=21)
dolefit4 <-rwf(dole2, h=21,drift=TRUE)
plot(dolefit1, PI=FALSE, ylab="People on unemployment benefits",xlab="Year",
     main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
lines(dolefit2$mean, col=2)
lines(dolefit4$mean, col=5)
legend("topleft",lty=1,cex = 0.5,col=c(4,2,5),
       legend=c("Mean method","Naive method","Drift method"))
```




```
##                             ME      RMSE       MAE         MPE      MAPE
## Training set -5.866280e-13  94382.68  73990.53 -203.55817 233.19607
## Test set      2.399545e+05 240472.91 239954.47   76.85554  76.85554
##                     MASE      ACF1 Theil's U
## Training set  3.785675 0.9813696        NA
## Test set     12.277105 0.6128072  18.21355
```

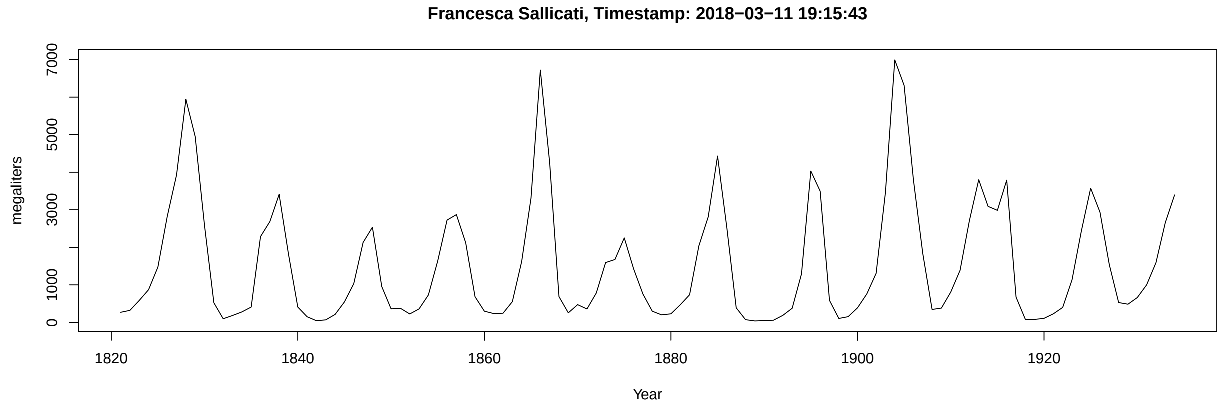
```
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set   1144.976  8840.205  4846.809  0.5157581 9.576022 0.2479836
## Test set     -22518.333 27498.149 24243.095 -7.4905660 7.990278 1.2403813
##                   ACF1 Theil's U
## Training set 0.3651818        NA
## Test set     0.6128072  2.225021
```

```
##                             ME      RMSE       MAE         MPE     MAPE
## Training set  8.038631e-13  8765.743  5089.218  -4.242988 11.17195
## Test set     -3.511307e+04 39087.831 35113.066 -11.536157 11.53616
##                   MASE      ACF1 Theil's U
## Training set 0.2603863 0.3651818        NA
## Test set     1.7965358 0.5790324  3.130936
```

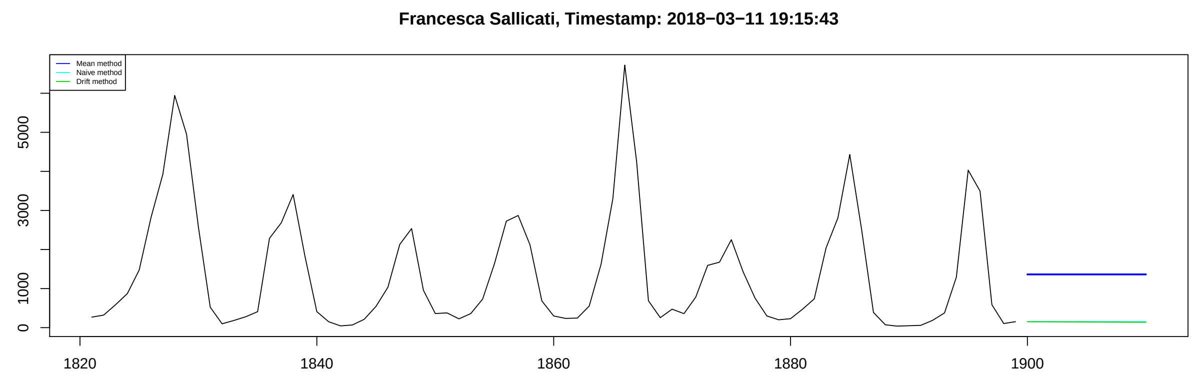
We can see that both from the plot and the accuracy table the drift methods shows the best performances on both training and test set, even though a simple classifier lik this is not able to give good forecast.

b) Annual Canadian lynx trappings (1821-1934). Data set lynx.

```
data("lynx")
lynx1 <- window(lynx, start = 1821)
par(mfrow = c(1, 1))
plot(lynx1, xlab = "Year", ylab = "megaliters",main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
```




```
lynx2 <- window(lynx,start=1821,end=1900-.1)
lynxfit1 <- meanf(lynx2, h=11)
lynxfit2 <- naive(lynx2, h=11)
lynxfit4 <-rwf(lynx2, h=11,drift=TRUE)
plot(lynxfit1, PI=FALSE,
     main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
lines(lynxfit2$mean, col=5)
lines(lynxfit4$mean, col=3)
legend("topleft",lty=1,cex = 0.5,col=c(4,5,3),
       legend=c("Mean method","Naive method","Drift method"))
```




```
##                        ME     RMSE      MAE        MPE     MAPE     MASE
## Training set 9.205295e-14 1483.922 1194.949 -354.85495 389.2782 1.500322
## Test set     1.035677e+03 2529.033 1797.732  -58.86489 115.1207 2.257148
##                   ACF1 Theil's U
## Training set 0.6904448        NA
## Test set     0.6827812  1.579347
```

```
##                       ME     RMSE       MAE       MPE     MAPE     MASE
## Training set   -1.487179 1160.455  796.4615 -50.08001 99.01854 1.000000
## Test set     2245.727273 3219.732 2245.7273  82.16770 82.16770 2.819631
##                   ACF1 Theil's U
## Training set 0.3543092        NA
## Test set     0.6827812  1.617875
```

```
##                         ME     RMSE       MAE       MPE     MAPE     MASE
## Training set 4.336266e-14 1160.454  796.8047 -49.58446 98.99972 1.000431
## Test set     2.254650e+03 3225.599 2254.6503  83.28199 83.28199 2.830834
##                   ACF1 Theil's U
## Training set 0.3543092        NA
## Test set     0.6825763  1.629691
```

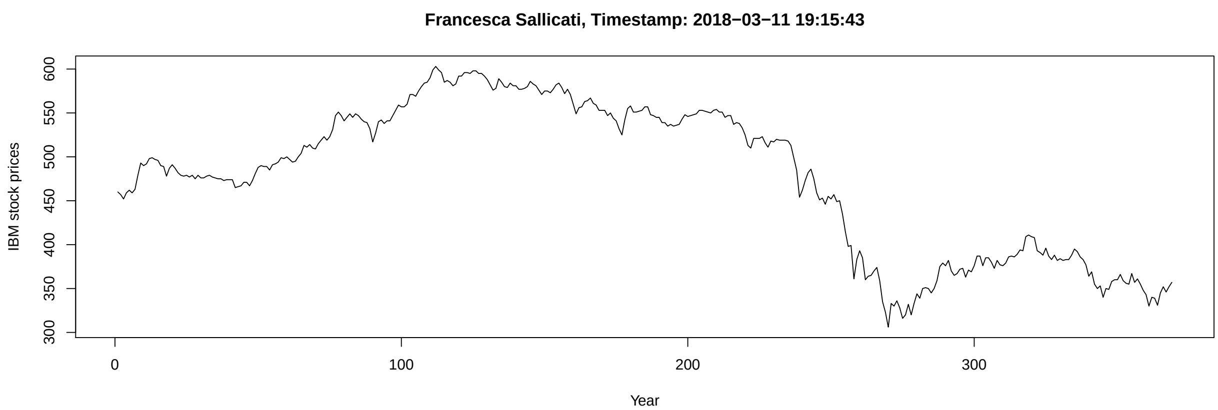
We can see that the forecasting methods are not appropriate for predict the values of this time series (among them the drift is the best). We could try more complex models in order to capture the true trend or use some quarterly data to make the seasonal naive method work fine.

2 Exercise 2:

Consider the daily IBM stock prices (data set ibmclose).

a) Produce some plots of the data in order to become familiar with it.

```r
data("ibmclose")
ibmclose1<-window(ibmclose)
par(mfrow = c(1, 1))
plot(ibmclose1, xlab = "Year", ylab = "IBM stock prices",main=paste("Francesca Sallicati, Timestamp:",Sy
```



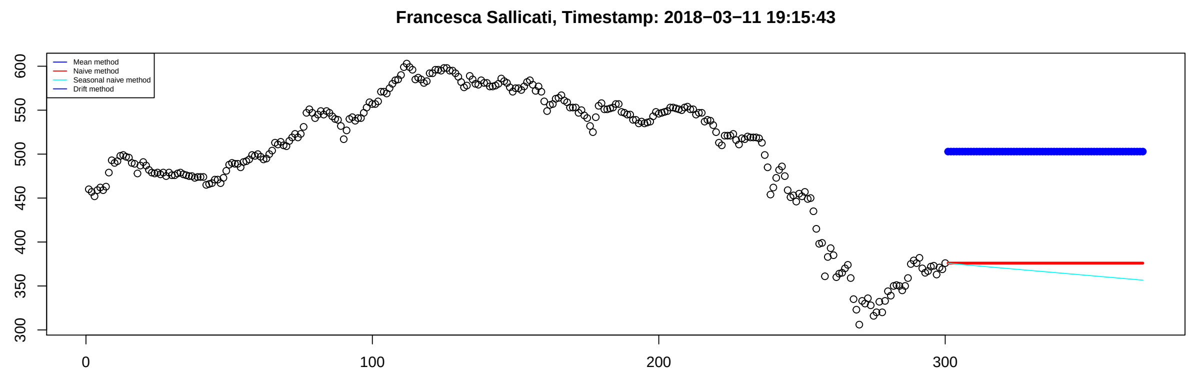
b) Split the data into a training set of 300 observations and a test set of 69 observations.

```r
ibmtrain<-ibmclose[1:300]
ibmtest<-ibmclose[301:369]
```

c) Try various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?

```r
ibmtrain <- window(ibmtrain, start=1)
ibmtest <- window(ibmtest, start=1)
ibmfit1 <- meanf(ibmtrain, h=69)
ibmfit2 <- naive(ibmtrain, h=69)
ibmfit4 <-rwf(ibmtrain, h=69,drift=TRUE)
plot(ibmfit1, PI=FALSE,
     main=paste("Francesca Sallicati, Timestamp:",Sys.time()), lwd=1)
lines(ibmfit2$mean, col=2, lwd=3)
lines(ibmfit4$mean, col=5)
```

```r
legend("topleft",lty=1,cex = 0.5,col=c(4,2,5),
       legend=c("Mean method","Naive method","Seasonal naive method","Drift method"))
```



Among the benchmark classifiers the drift methods seems the best, even though the performances aren't good at all.

```
##                          ME      RMSE       MAE        MPE     MAPE
## Training set  1.660438e-14  73.61532  58.72231  -2.642058 13.03019
## Test set     -1.306180e+02 132.12557 130.61797 -35.478819 35.47882
##                  MASE      ACF1 Theil's U
## Training set 11.52098 0.9895779        NA
## Test set     25.62649 0.9314689  19.05515
```

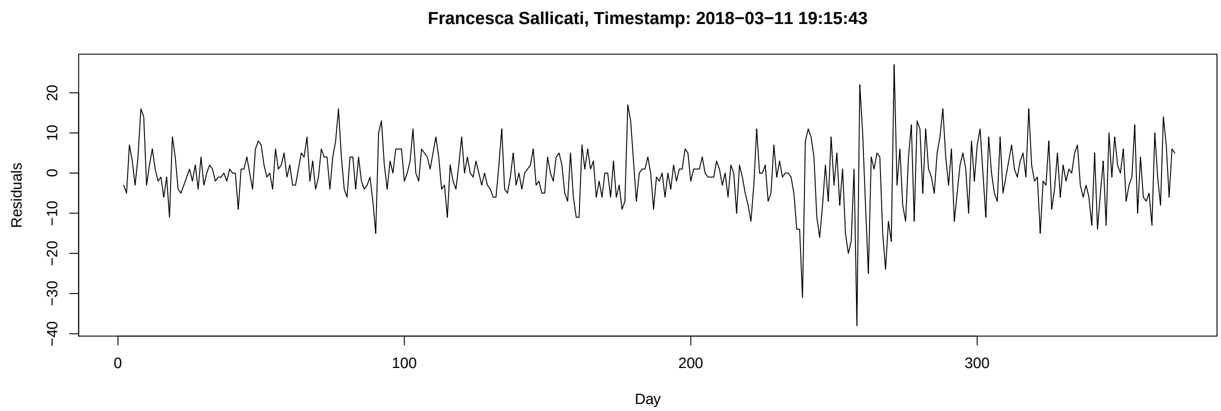
```
##                      ME      RMSE      MAE         MPE     MAPE     MASE
## Training set -0.2809365  7.302815  5.09699 -0.08262872 1.115844 1.000000
## Test set     -3.7246377 20.248099 17.02899 -1.29391743 4.668186 3.340989
##                   ACF1 Theil's U
## Training set 0.1351052        NA
## Test set     0.9314689  2.973486
```

```
##                        ME      RMSE       MAE         MPE     MAPE
## Training set 2.870480e-14  7.297409  5.127996 -0.02530123 1.121650
## Test set     6.108138e+00 17.066963 13.974747  1.41920066 3.707888
##                  MASE      ACF1 Theil's U
## Training set 1.006083 0.1351052        NA
## Test set     2.741765 0.9045875  2.361092
```

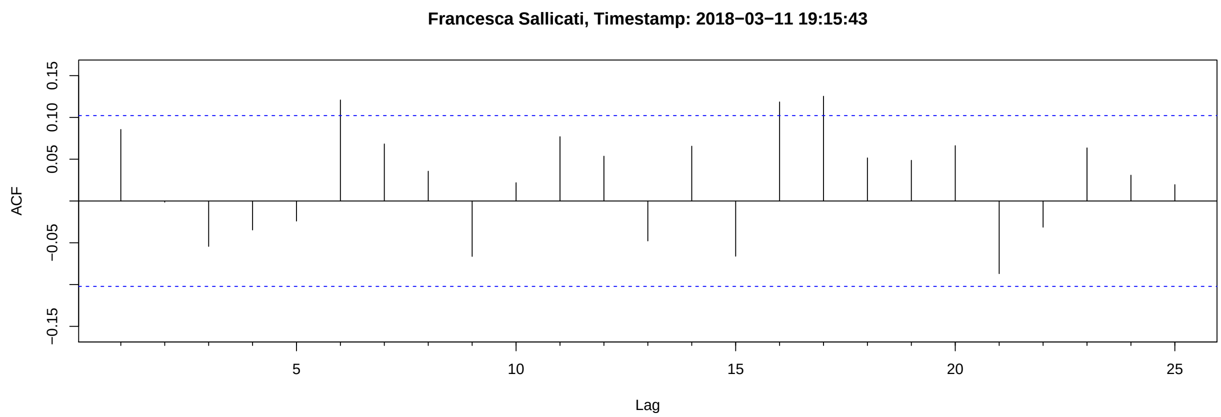
This is supported by the error tables too, even if not all the measures agree.

d) For the best method, compute the residuals and plot them. What do the plots tell you?

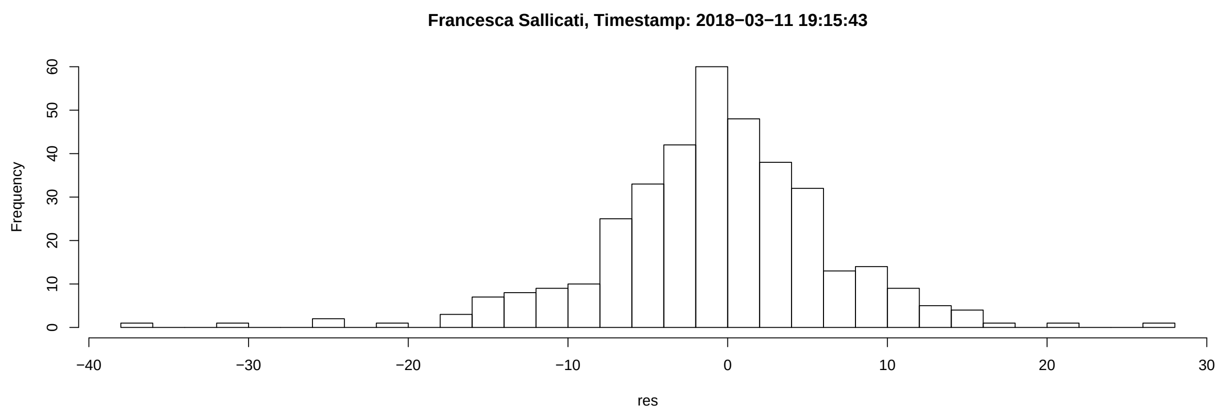
```r
res <- residuals(rwf(ibmclose))
plot(res, main=paste("Francesca Sallicati, Timestamp:",Sys.time()), ylab="Residuals", xlab="Day")
```




```r
Acf(res, main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
```




```r
hist(res, nclass="FD", main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
```




```r
Box.test(res, lag = 10, fitdf = 0)
```

```
##
##  Box-Pierce test
##
```

```
## data:  res
## X-squared = 13.786, df = 10, p-value = 0.183
```

```
Box.test(res, lag = 10, fitdf = 0, type = "Lj")
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 14.064, df = 10, p-value = 0.1701
```

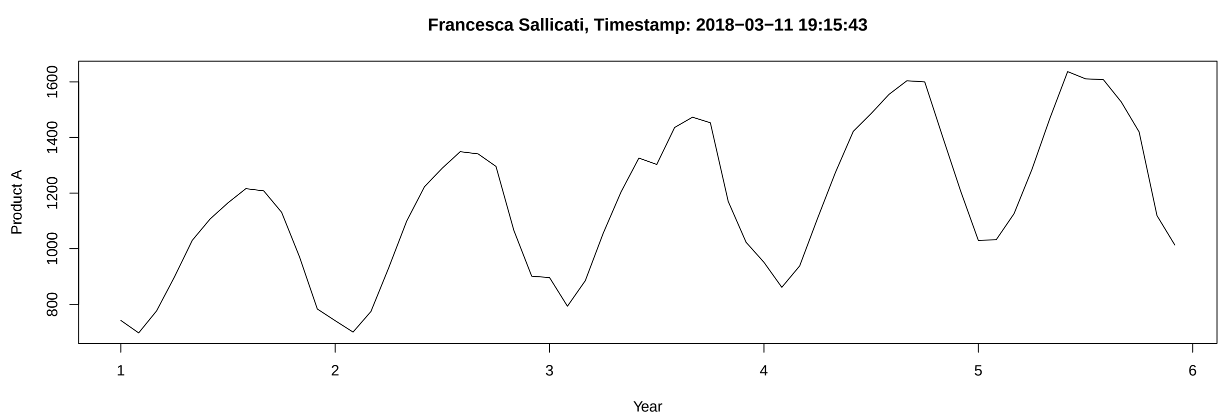
From the residual analysis we can see than no information is left, since the errors follow a random noise as it can be seen in the first plot and they have no autocorrelation within the lags since their values are between the boundaries. Moreover both the Box-Pierce and Ljung-Box test pvalues do not les us reject the null hypothesisi $H0$ of independence, therefore there is no evidence of dependence between them.

3 Exercise 3:

The data represent the monthly sales (in thousands) of product A for a plastics manufacturer for years 1 through 5 (data set plastics).

a) Plot the time series of sales of product A. Can you identify the seasonal fluctuations and/or a trend?

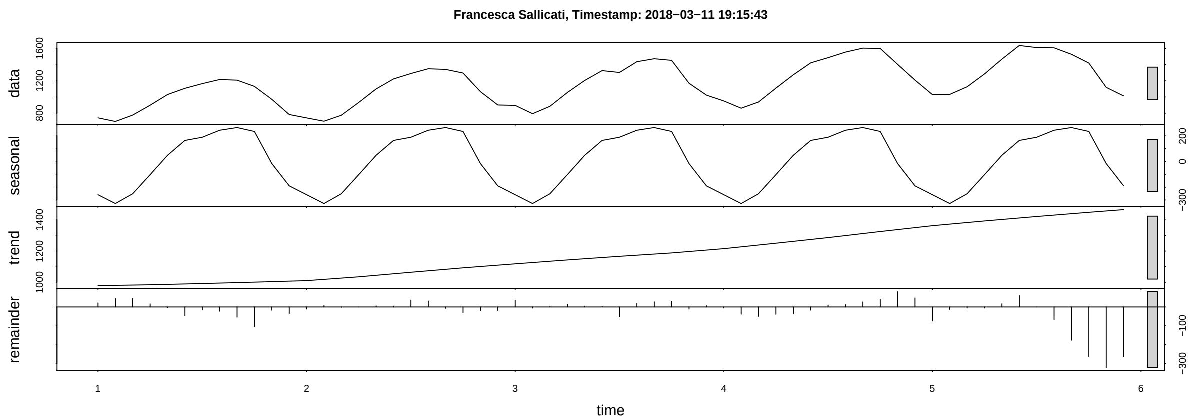
```
data("plastics")
plastics1 <- window(plastics, start = 1)
par(mfrow = c(1, 1))
plot(plastics1, xlab = "Year", ylab = "Product A",main=paste("Francesca Sallicati, Timestamp:",Sys.time
```



Yes, the data shows a seasonality pattern and a potential trend as well, since the fluctuations are constantly repeating themselves the times series gets slightly higher values while increasing the years.

b) Use an STL decomposition to calculate the trend-cycle and seasonal indices. (Experiment with having fixed or changing seasonality).

```
fit <- stl(plastics, t.window=25, s.window="periodic", robust=TRUE)
par(mfrow=c(2,1))
plot(fit,main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
```



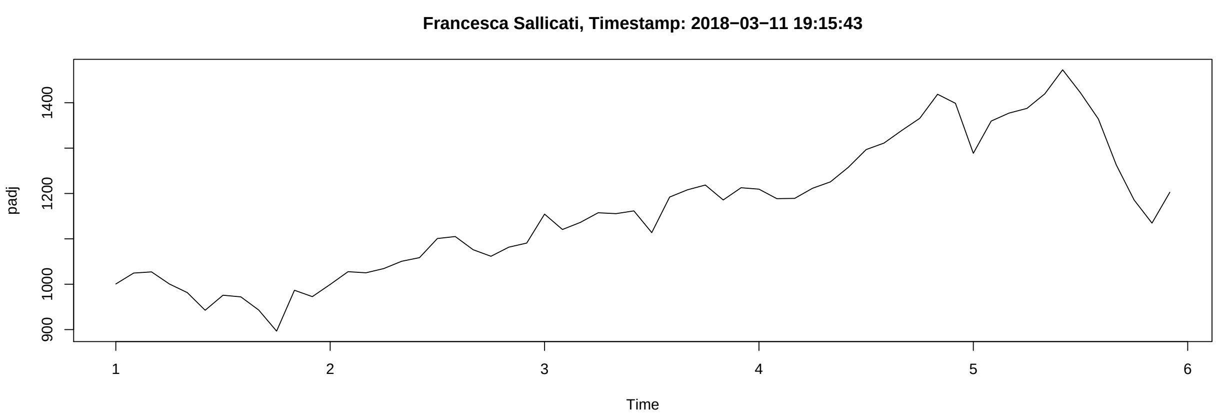
The STL decomposition works fine by imposing the t.window, which represents the lags considered, to a value greater than 25 that is to say by considering at least 2 years. On the contrary changing the s.window does not produce such different decompositions.

c) Do the results support the graphical interpretation form part (a)?

Yes, beacuse the plots of the seasonal part of the STL decomposition clearly captures the seasonal fluctuations, whereas the trend part is showing an increase in the sales of product A over the years, just as one would expect after seeing the plot of the series itself. Moreover there is no big difference among the different plots obtained by changing the seasonality parameter.

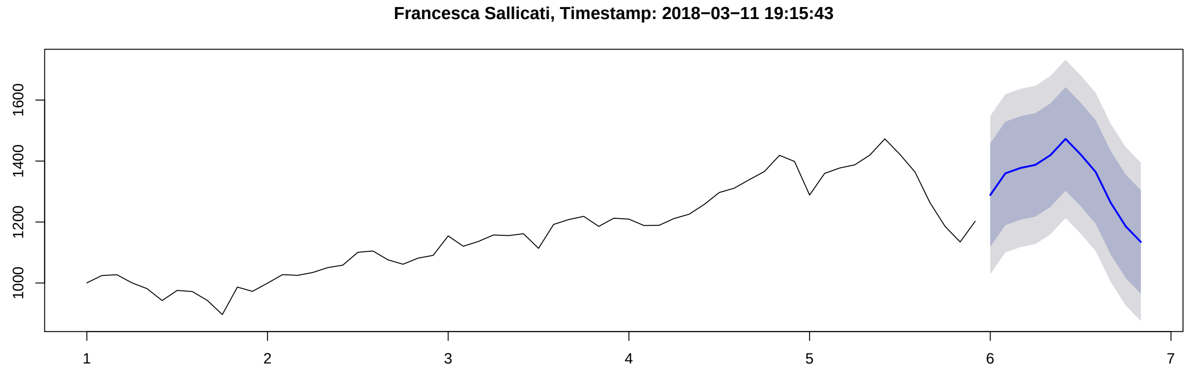
d) (d) Compute and plot the seasonally adjusted data.

```
padj <- seasadj(fit)
plot(padj,main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
```



e) Use a random walk to produce forecasts of the seasonally adjusted data.

```
rw<-snaive(ma(padj,1), h=11)
plot(rw, main=paste("Francesca Sallicati, Timestamp:",Sys.time()))
```



Francesca Sallicati, Timestamp: 2018−03−11 19:15:43
1000
1200
1400
1600
1
2
3
4
5
6
7