Time Series Analysis and Forecasting - Exercise Set 1

Borja Ruiz Amantegui 100357358

10 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.

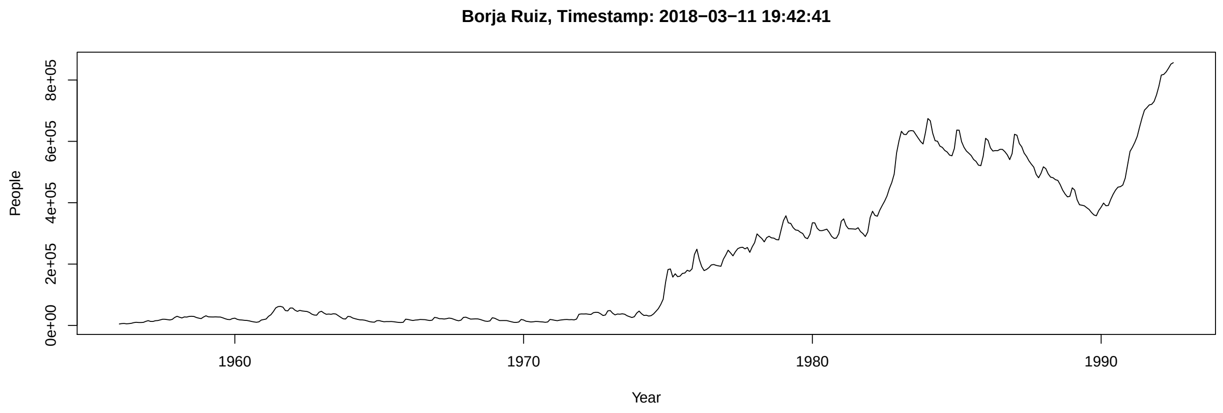
In each case, do you think the forecasts are reasonable? If not, how could they be improved?

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

```
library(fpp)
```

```
## Loading required package: forecast

## Loading required package: fma

## Loading required package: expsmooth

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: tseries
```

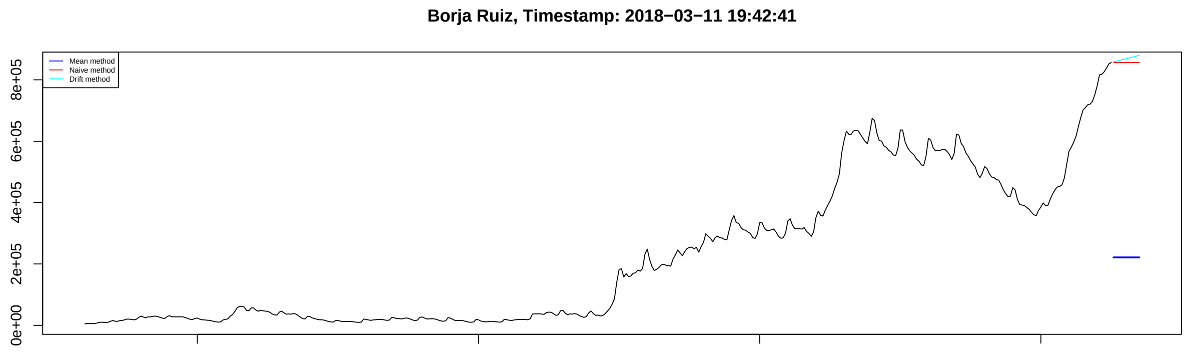
```
library(fma)
data(dole)
dole1 <- window(dole, start = 1975)

par(mfrow = c(1, 1))
plot(dole, xlab = "Year", ylab = "People",main=paste("Borja Ruiz, Timestamp:",Sys.time()))
```



Since we don't see any seasonality we can omit this method. We can also appreciate a general positive trend which doesn't make the mean adequate. And for the same reason, the positive trend, we can decide the best method will be drift.

```
dole2 <- window(dole,start=1956)
dolefit1 <- meanf(dole2, h=12)
dolefit2 <- naive(dole2, h=12)
```

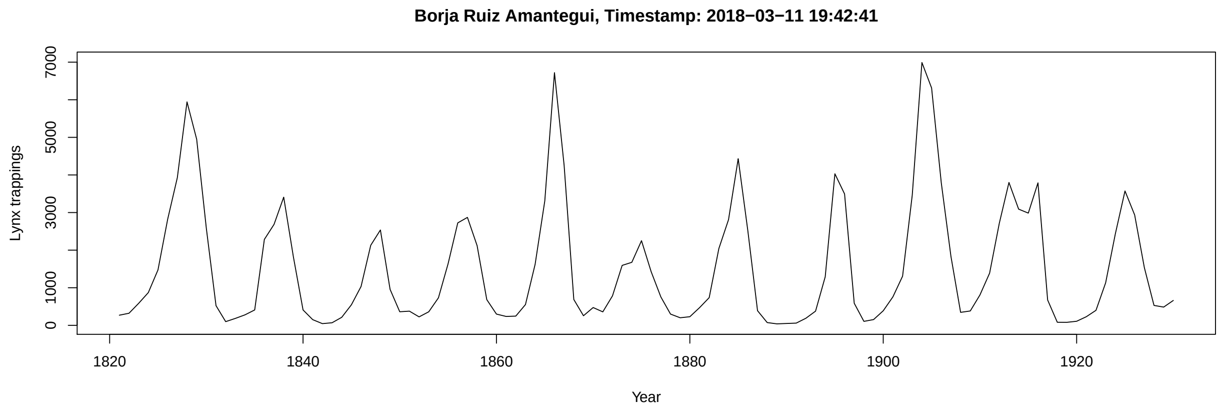
```r
dolefit4 <-rwf(dole2, h=12,drift=TRUE) #drift method (like a reg between fist and last value)
#and ploting
plot(dolefit1, PI=FALSE,
     main=paste("Borja Ruiz, Timestamp:",Sys.time()))
lines(dolefit2$mean, col=2)
lines(dolefit4$mean, col=5)
legend("topleft",lty=1,cex = 0.5,col=c(4,2,5),
       legend=c("Mean method","Naive method","Drift method"))
```



We can corroborate that the drift method is the most appropiate though it still doesn't work satisfactorily.

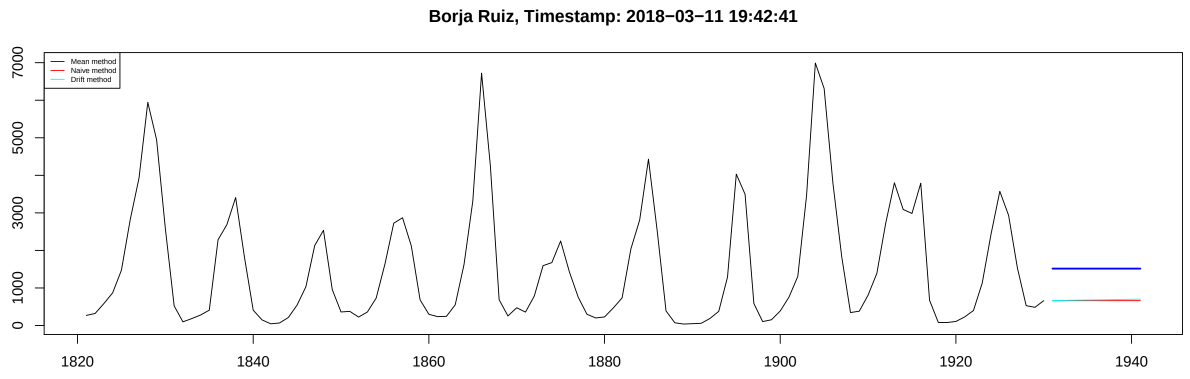
B) Annual Canadian lynx trappings (1821-1934). Data set lynx.

```r
data("lynx")
lynx1 <- window(lynx,start=1821,end=1930)

#Plot the time series
par(mfrow = c(1, 1))
plot(lynx1, xlab = "Year", ylab = "Lynx trappings",main=paste("Borja Ruiz Amantegui, Timestamp:",Sys.ti
```



For this dataset we can omit the seasonal naive method, since the data isn't structured neither in months or quartiles.

```
lynx2 <- window(lynx,start=1821,end=1930)
lynxfit1 <- meanf(lynx2, h=11)
lynxfit2 <- naive(lynx2, h=11)
lynxfit4 <-rwf(lynx2, h=11,drift=TRUE) #drift method (like a reg between fist and last value)
#and ploting
plot(lynxfit1, PI=FALSE,
     main=paste("Borja Ruiz, Timestamp:",Sys.time()))
lines(lynxfit2$mean, col=2)
lines(lynxfit4$mean, col=5)
legend("topleft",lty=1,cex = 0.5,col=c(4,2,5),
       legend=c("Mean method","Naive method","Drift method"))
```

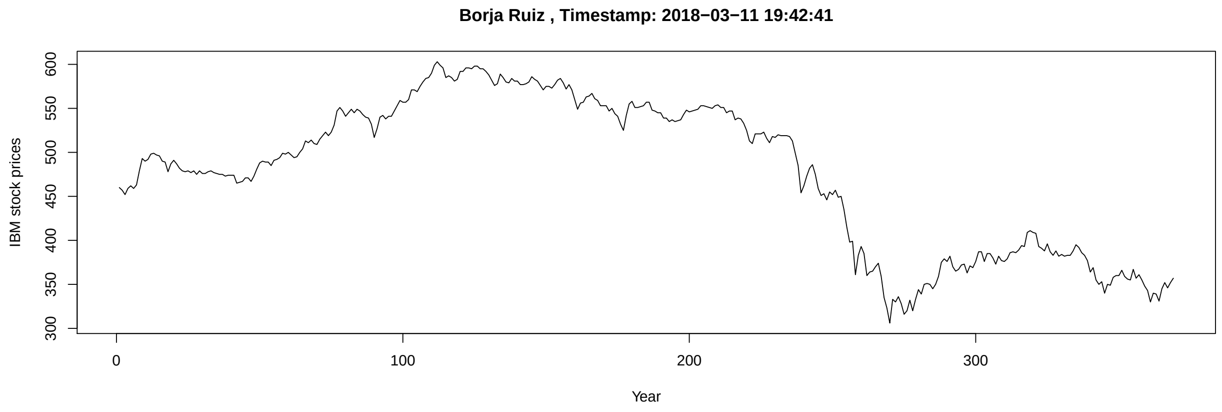


All methods seem incompetent for this dataset. Probably a good forecast will be made if the data was formatted so we could use appropiately seasonal naive method. We could also use a different method which can handle other types of seasonality.

2 Exercise 2

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

```
data("ibmclose")
ibmclose1<-window(ibmclose)

#Plot the time series (i HAVE TO ADD SOME OTHERS)
par(mfrow = c(1, 1))
plot(ibmclose1, xlab = "Year", ylab = "IBM stock prices",main=paste("Borja Ruiz , Timestamp:",Sys.time(
```



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

```
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?

```
ibmtrain <- window(ibmtrain, start=1)
ibmtest <- window(ibmtest, start=1)
ibmfit1 <- meanf(ibmtrain, h=69)
ibmfit2 <- naive(ibmtrain, h=69)
ibmfit3 <- snaive(ibmtrain, h=69)
ibmfit4 <-rwf(ibmtrain, h=69,drift=TRUE)
#and ploting

a1<-accuracy(ibmfit1, ibmtest);a1
```

```
##                          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
```

```
a2<-accuracy(ibmfit2, ibmtest);a2
```

```
##                      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
```

```
a3<-accuracy(ibmfit3, ibmtest);a3
```

```
##                      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
```

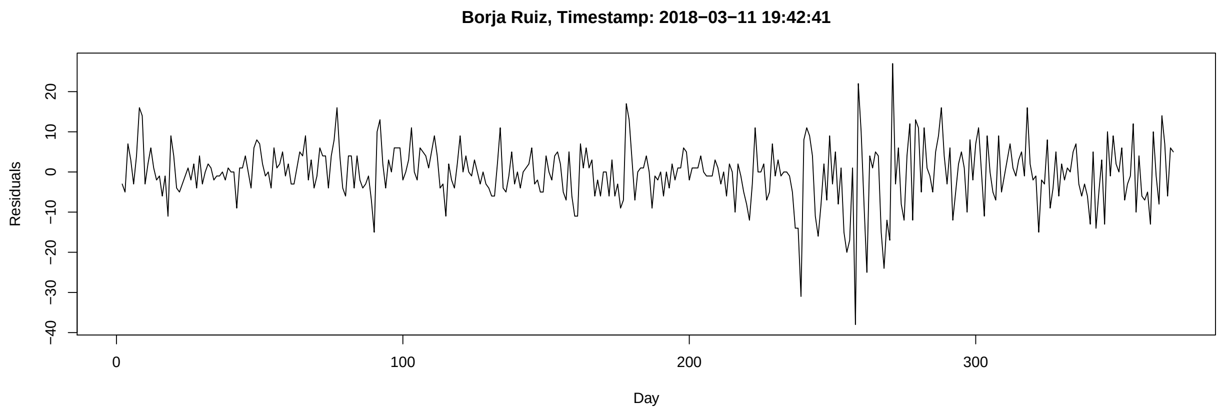
```
a4<-accuracy(ibmfit4, ibmtest);a4
```

```
##                        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
```

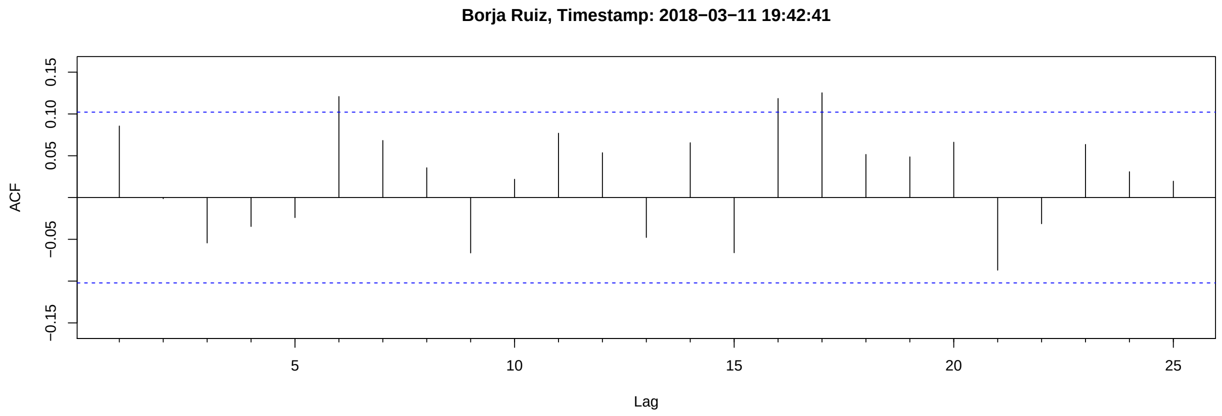
For most of the tests the drift method seems best to predict the test data set.

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

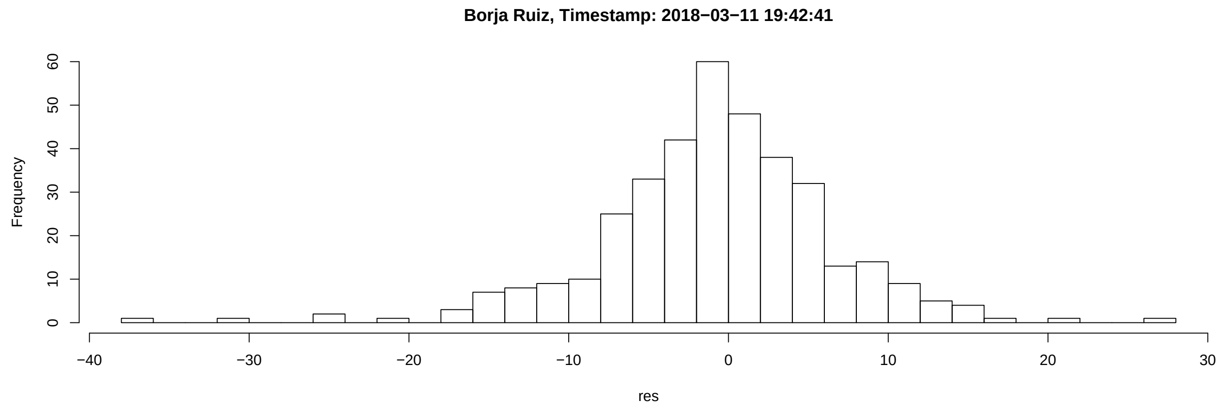
```
res <- residuals(rwf(ibmclose))

plot(res, main=paste("Borja Ruiz, Timestamp:",Sys.time()), ylab="Residuals", xlab="Day")
```



```
Acf(res, main=paste("Borja Ruiz, Timestamp:",Sys.time()))
```



```
hist(res, nclass="FD", main=paste("Borja Ruiz, Timestamp:",Sys.time()))
```



The mean of the residuals is very close to zero and there is no significant correlation in the residuals series. The time plot of the residuals shows that the variation of the residuals stays much the same across the historical data, so the residual variance can be treated as constant.

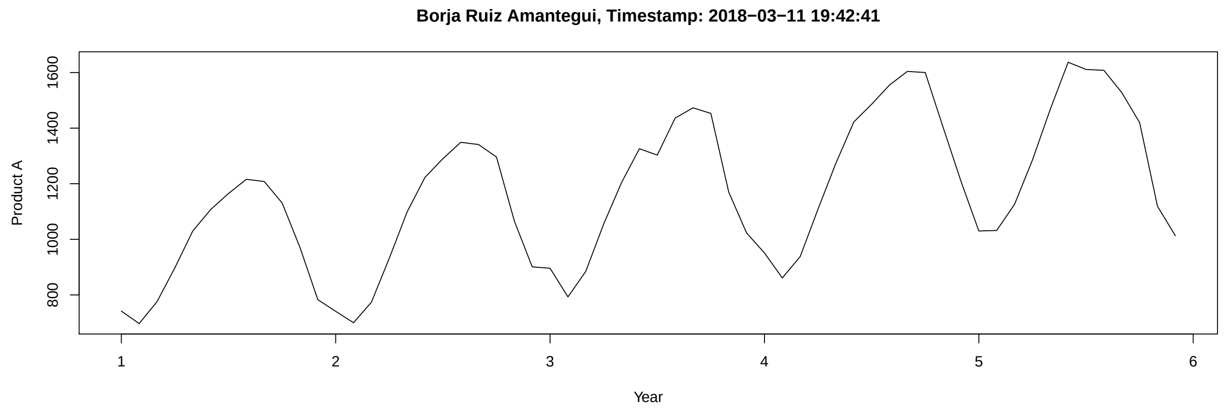
We may also appreciatte a normal distribution in the data.

From the ACF grah we can assume that since we have several values surpassing the limit we probably have autocorrelation problems and missing information in our residuals.

3 Exercise 3

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

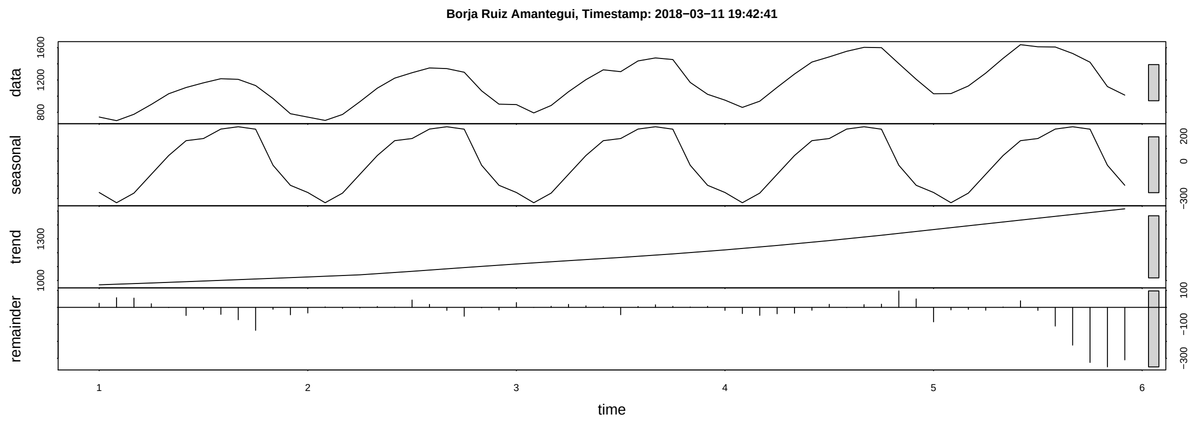
```
data("plastics")
plastics1 <- window(plastics, start = 1)

#Plot the time series
par(mfrow = c(1, 1))
plot(plastics1, xlab = "Year", ylab = "Product A",main=paste("Borja Ruiz Amantegui, Timestamp:",Sys.tim
```



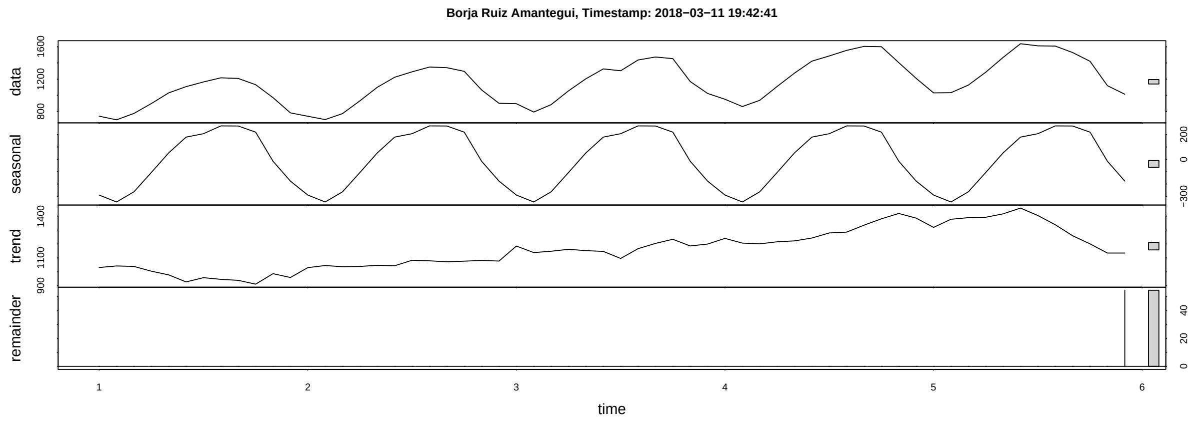
Yes, the data shows a seasonality pattern and a potential trend as well, since the seasonalities maximums are higher every season.

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

```r
fit <- stl(plastics, t.window=30, s.window="periodic", robust=TRUE)
fit2 <- stl(plastics, t.window=3, s.window=13, robust=TRUE)
par(mfrow=c(2,1))
plot(fit, main=paste("Borja Ruiz Amantegui, Timestamp:",Sys.time()))
```



```r
plot(fit2, main=paste("Borja Ruiz Amantegui, Timestamp:",Sys.time()))
```



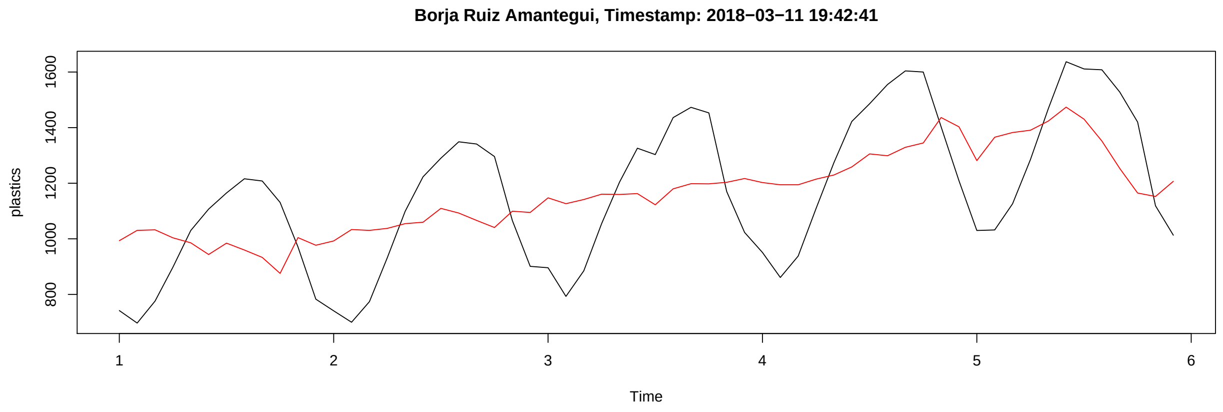
We appreciate that changing de s.window will not change how well the seasonal component adjusts, since this component doesn't change its magnitude over time.

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

Yes, since the decomposition shows how the seasonal fluctuation adjusts perfectly to the data and the trend is positive through the whole graph.

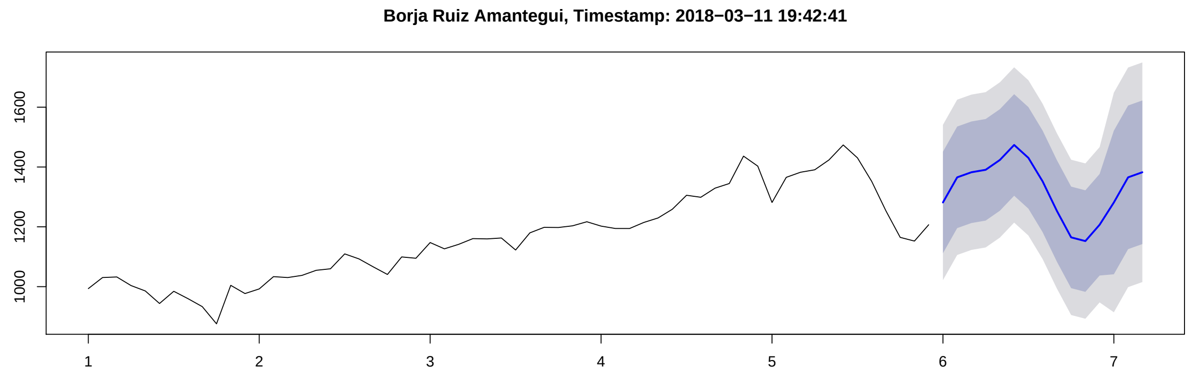
D) Compute and plot the seasonally adjusted data.

```r
padj <- seasadj(fit)
plot(plastics, main=paste("Borja Ruiz Amantegui, Timestamp:",Sys.time()))
lines(padj, col="red")
```



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

```r
naivefit <- snaive(ma(padj, 1), h=15)
plot(naivefit, main=paste("Borja Ruiz Amantegui, Timestamp:",Sys.time()))
```


Borja Ruiz Amantegui, Timestamp: 2018−03−11 19:42:41