Forecast_1

Diomides_Mavroyiannis

25/11/2020

#Loading the data

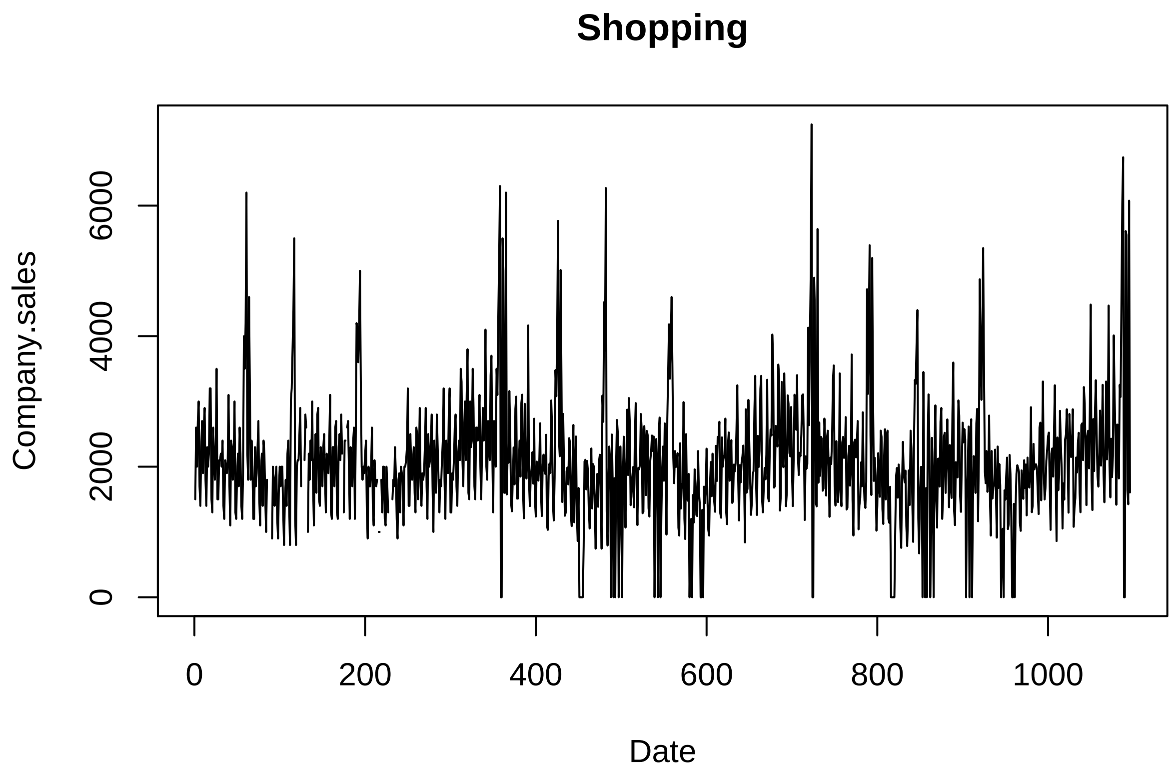
```
data <- read.csv("C:/Users/DavidEttinger02/Desktop/Forecasting/Bi-weekly assignment data.csv")

#Preparing the library
library(forecast)
```

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

```
#Part A
```

Visualize the Company sales column

```
plot(data$Company.Sales , type = "l", main = "Shopping", ylab = "Company.sales", xlab = "Date" )
```



#missing values The approach here, is to check if a value is missing, if it is, then take the mean of the next and the previous period to fill it in. If the next slot is empty we simply take the average of two older slots

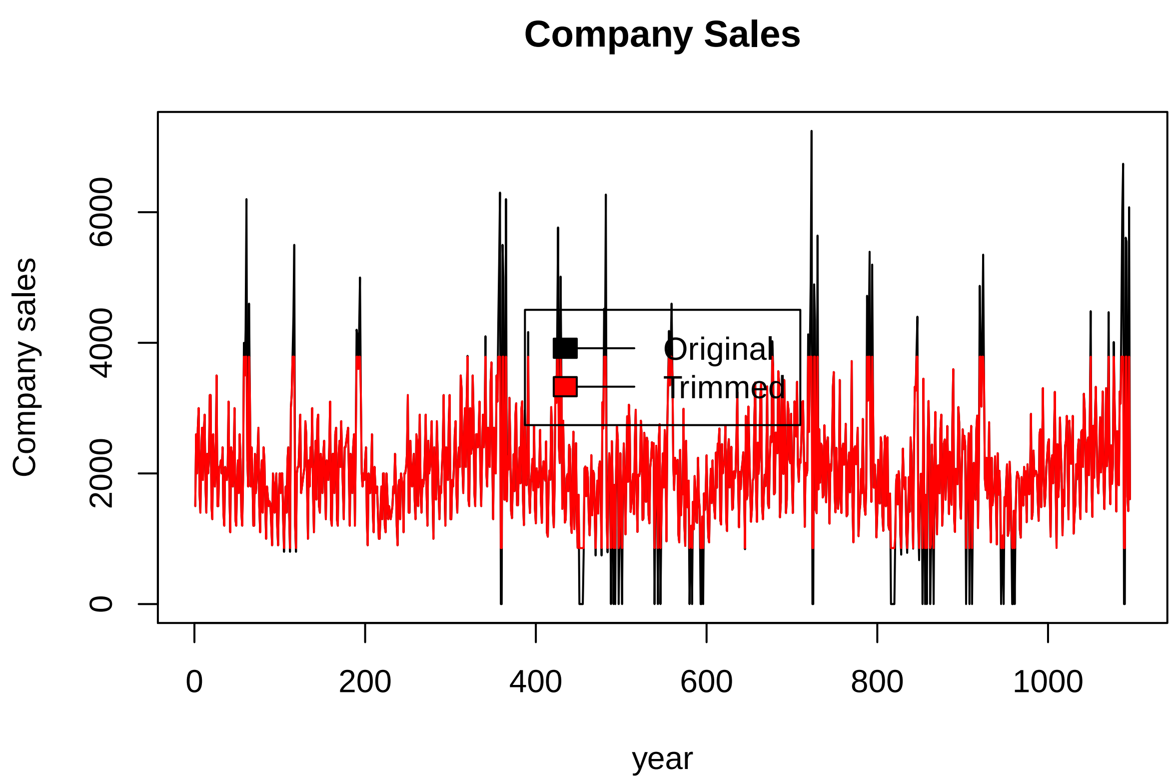
```
for (i in (frequency(data$Company.Sales)+1): (length(data$Company.Sales)-12)){
  if (is.na(data$Company.Sales[i])==TRUE){
    print(i)
    data$Company.Sales[i] <- mean(c(data$Company.Sales[i-frequency(data$Company.Sales)], data$Company.Sa
    if (is.na(data$Company.Sales[i])==TRUE){
      data$Company.Sales[i] <- mean(c(data$Company.Sales[i-frequency(data$Company.Sales) ], data$Company
    }
  }
}
```

```
## [1] 86
## [1] 88
## [1] 89
## [1] 90
## [1] 126
## [1] 128
## [1] 132
## [1] 174
## [1] 178
## [1] 215
## [1] 218
## [1] 228
## [1] 229
## [1] 231
```

#outliers We could reduce the effect of outlier by simply taking the log of the function but here we opt simply to change them to be closer to the other data. #We first compute the values where 95% and 5% of the data is above that

```
lowerlimit <- quantile(data$Company.Sales,0.95)
upperlimit <- quantile(data$Company.Sales,0.05)
#We create a new vector that will hold the series we will create
n_data1 <- data$Company.Sales
#We now replace the values which are above and below the limits calculated earlier limits
n_data1[n_data1>lowerlimit] <- lowerlimit
n_data1[n_data1<upperlimit] <- upperlimit
```

```
plot(data$Company.Sales, type = "l", ylab = "Company sales", xlab = "year", main = "Company Sales")
lines(n_data1, col = "red")
legend("center", legend = c("Original", "Trimmed"),col, c("black","red"),lty=1)
```



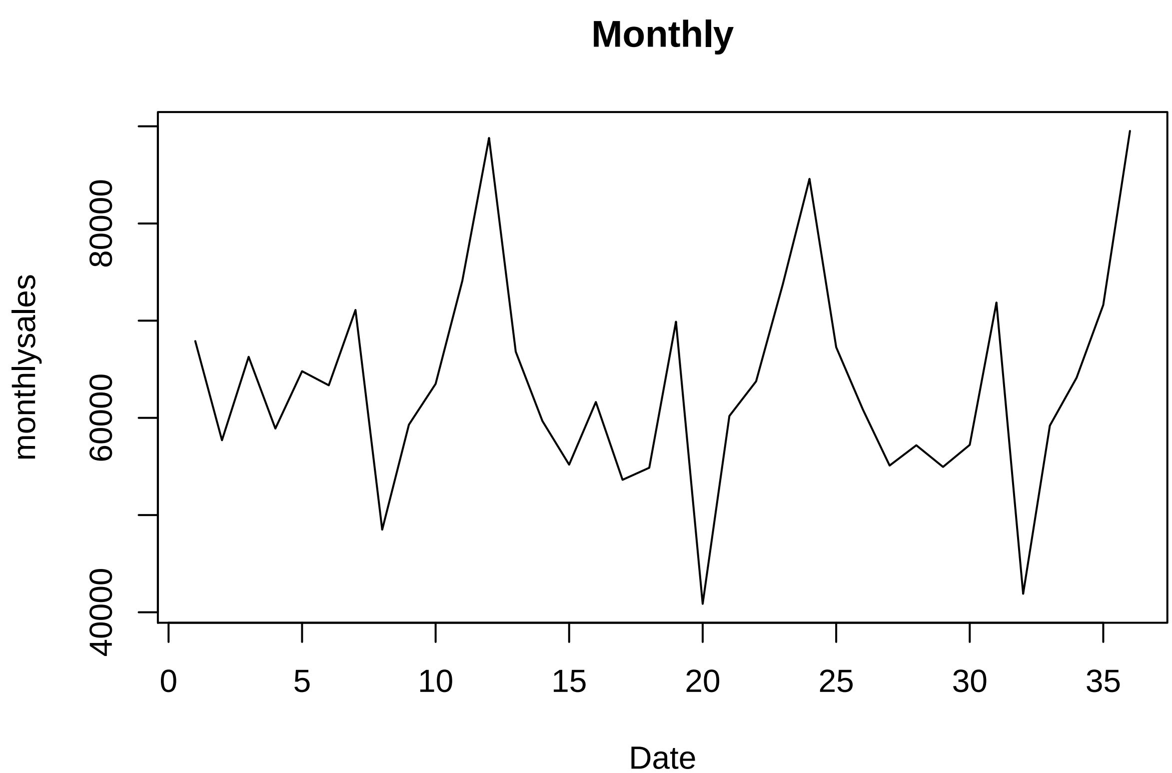
#makeweekly

```
# Coerce to Date class(it says double)
data$Date <- as.Date(data$Date , format = '%d-%m-%Y')
# Extract day of the week (Saturday = 6)
data$Week_Day <- as.numeric(format(data$Date, format='%w'))
# Adjust end-of-week date (first saturday from the original Date)
data$End_of_Week <- data$Date + (6 - data$Week_Day)
# Aggregate over week and climate division
weeklydata <- aggregate(data$Company.Sales~data$End_of_Week, FUN=mean, data=data, na.rm=TRUE)

#makemonthly
```

```
short.date = strftime(data$Date , "%Y/%m")
monthlydata = aggregate(data$Company.Sales ~ short.date, FUN = sum)

#rename
names(weeklydata)[names(weeklydata) == "data$End_of_Week"] <- "Endofweek"
names(weeklydata)[names(weeklydata) == "data$Company.Sales"] <- "weekly"

names(monthlydata)[names(monthlydata) == "short.date"] <- "Month"
names(monthlydata)[names(monthlydata) == "data$Company.Sales"] <- "monthlysales"
```

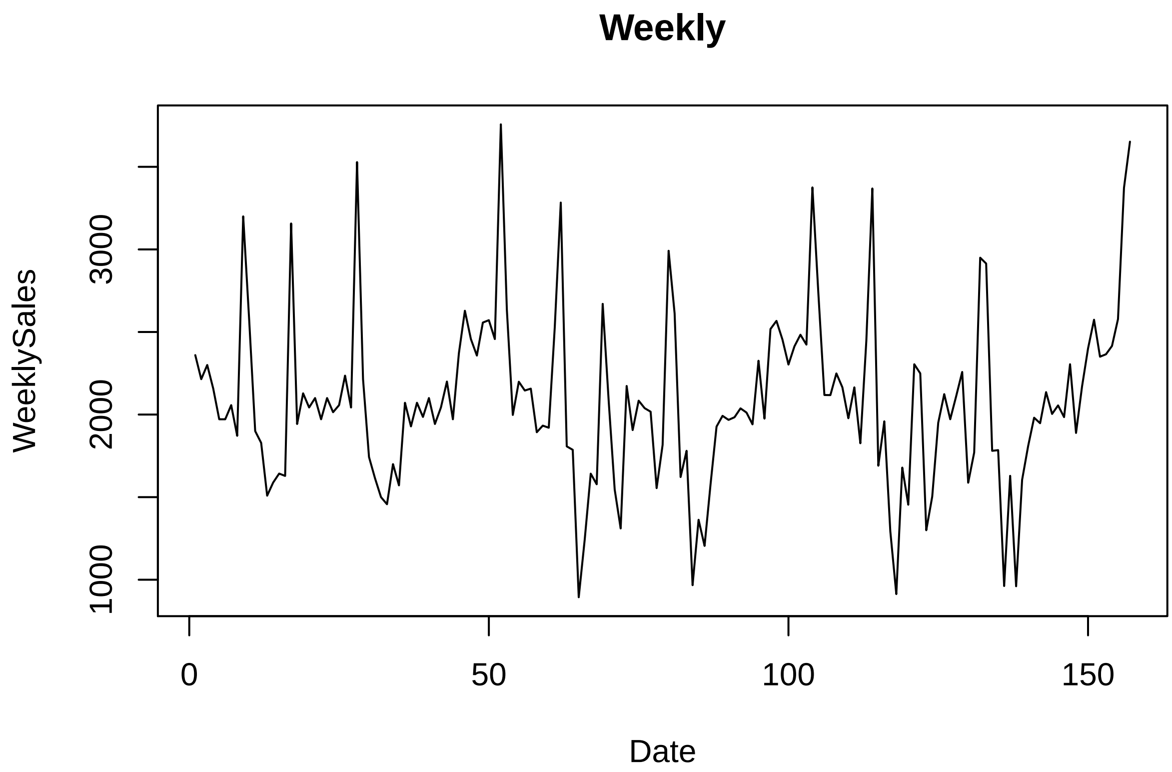
#visualize monthly

```
plot(monthlydata$monthlysales , type = "l", main = "Monthly", ylab = "monthlysales", xlab = "Date" )
```



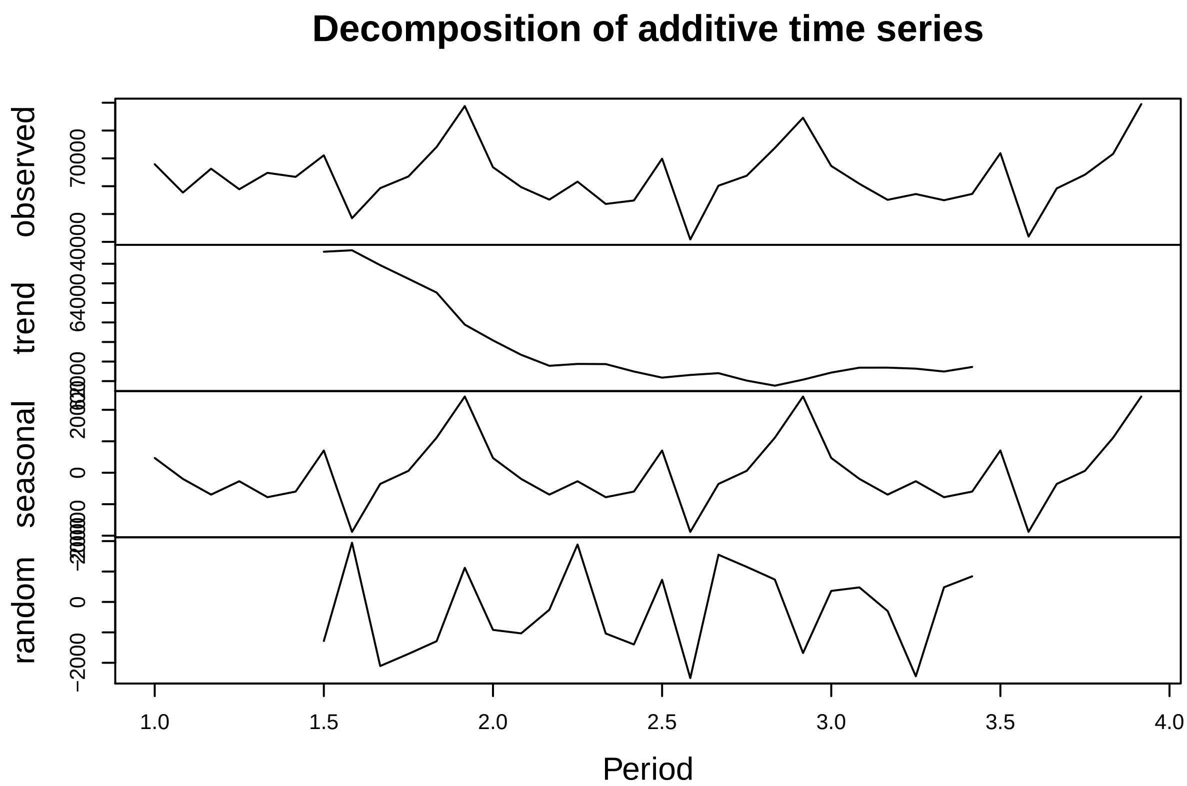
#visualize Weekly

```
plot(weeklydata$weekly , type = "l", main = "Weekly", ylab = "WeeklySales", xlab = "Date" )
```

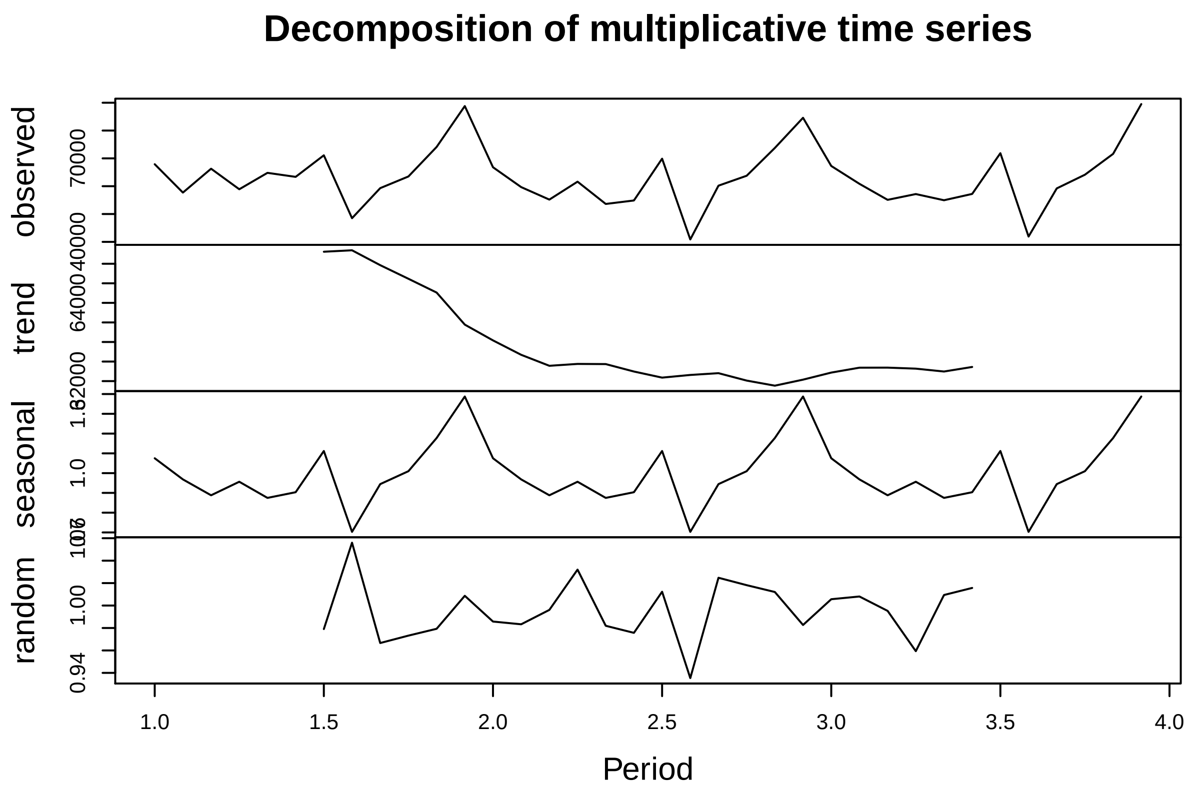



```r
#decompose monthly
TS <- ts(monthlydata$monthlysales, frequency = 12)
dec1 <- decompose(TS , type = "additive")
dec2 <- decompose(TS , type = "multiplicative")
```

```r
plot(dec1, type = "l", ylab = "Index", xlab = "Period")
```




```r
plot(dec2, type = "l", ylab = "Index", xlab = "Period")
```



#As we can see, the main difference between the decomposition is the that the multiplicative decomposition reduces the magnitude of the random shock.

#Average

```r
#The average daily sale is:
mean(data$Company.Sales)
```

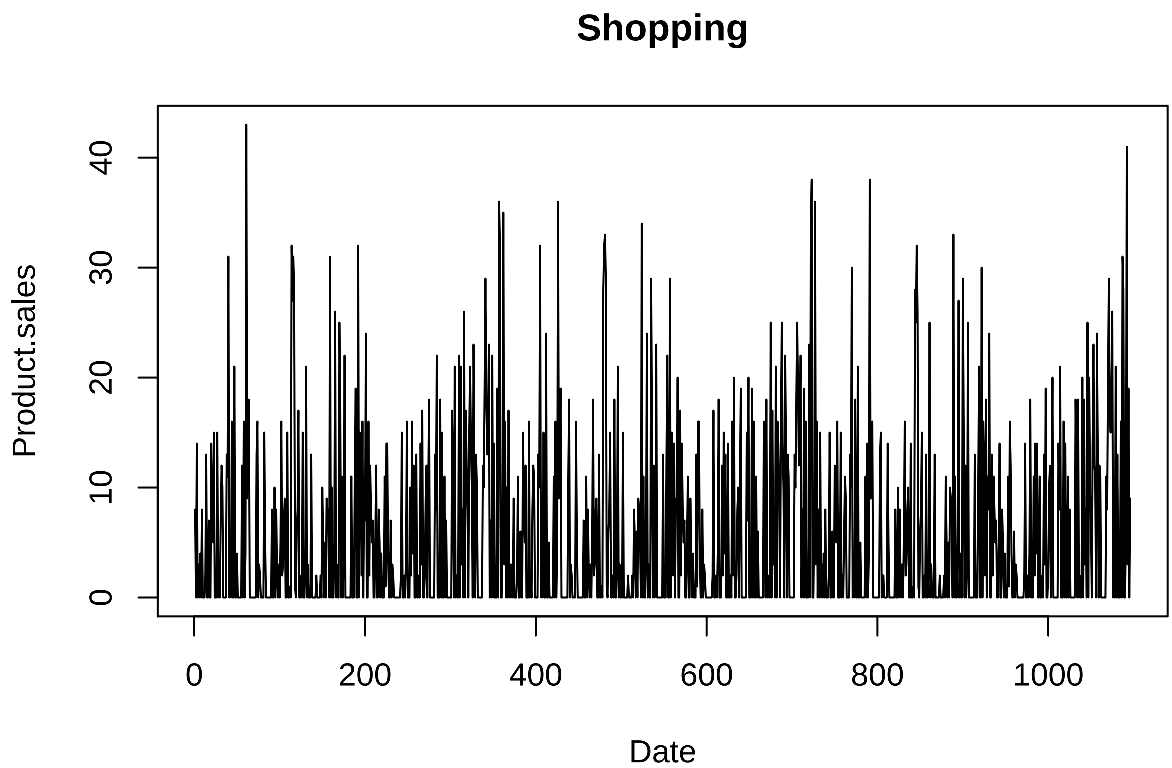
```
## [1] 2080.188
```

```r
#The average monthly sale is:
mean(monthlydata$monthlysales)
```

```
## [1] 63330.17
```

#Part B

```r
#visualize
plot(data$Product.Sales , type = "l", main = "Shopping", ylab = "Product.sales", xlab = "Date" )
```




```
#Average Daily demand
mean(data$Product.Sales)
```

```
## [1] 5.789234
```

```
#Coefficient of Variation
sd(data$Product.Sales[0<data$Product.Sales])*100/mean(data$Product.Sales[0<data$Product.Sales])
```

```
## [1] 69.53238
```

```
#ADI
nonzero <- length(which(data$Product.Sales != 0))
zero <- length(which(data$Product.Sales != 0))
#the rate is:
rate <-zero/(nonzero+zero)
#So we inverse this to get the average expected days
Avgwait <- 1/rate
Avgwait
```

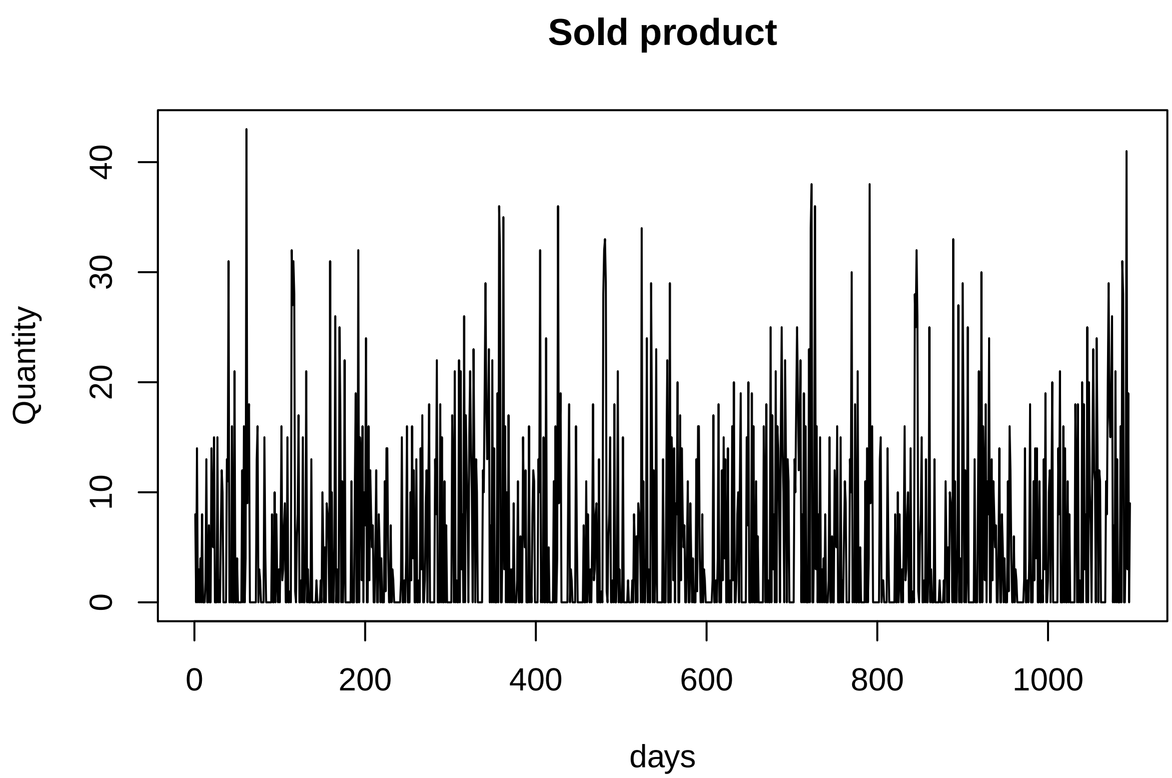
```
## [1] 2
```

```
library(fpp)
```

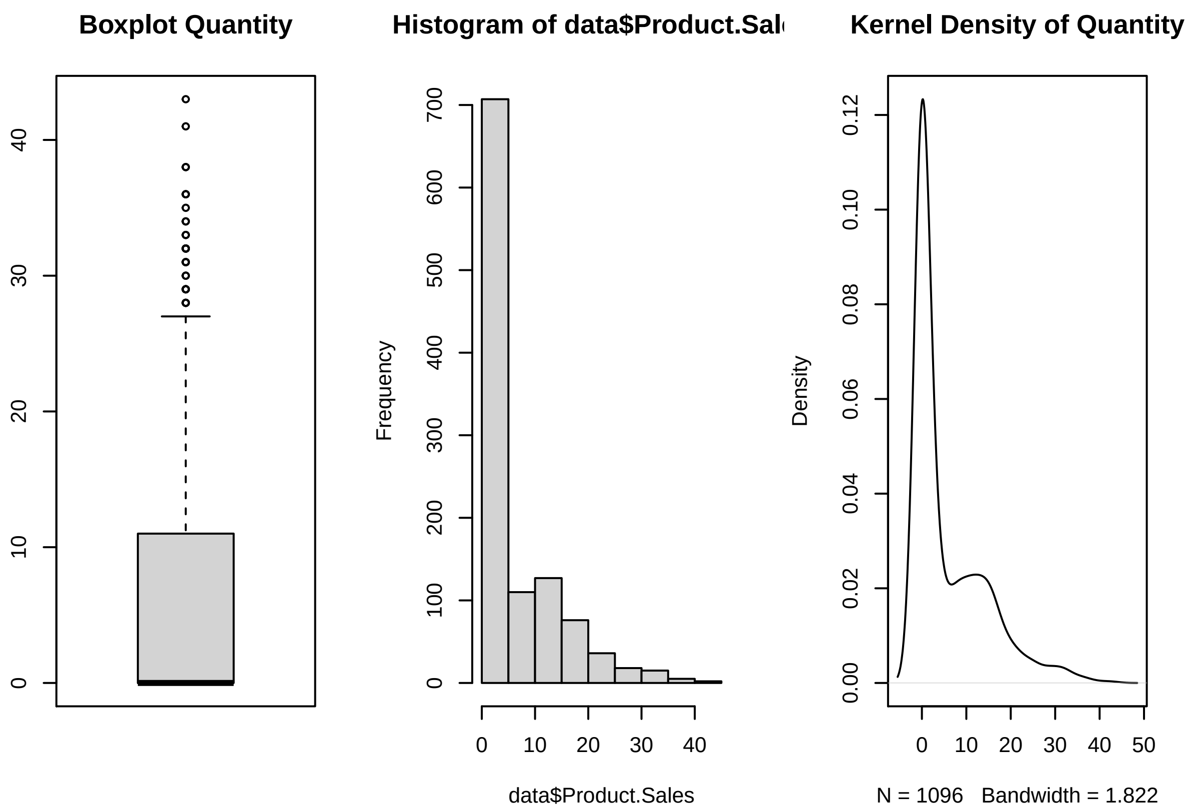
```
## Loading required package: fma

## 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
```

```r
plot(data$Product.Sales , type = "l", ylab = "Quantity", xlab = "days", main = "Sold product")
```




```r
par(mfrow = c(1,3))
boxplot(data$Product.Sales, main = "Boxplot Quantity" )
hist(data$Product.Sales )
plot(density(data$Product.Sales), main = "Kernel Density of Quantity")
```



#We can see the quantiles by looking at the density:

```
density(data$Product.Sales)
```

```
##
## Call:
##  density.default(x = data$Product.Sales)
##
## Data: data$Product.Sales (1096 obs.);    Bandwidth 'bw' = 1.822
##
##        x                 y
##  Min.   :-5.466   Min.   :0.0000023
##  1st Qu.: 8.017   1st Qu.:0.0017089
##  Median :21.500   Median :0.0063186
##  Mean   :21.500   Mean   :0.0185116
##  3rd Qu.:34.983   3rd Qu.:0.0222830
##  Max.   :48.466   Max.   :0.1233287
```

#Or by computing them directly

```
quantile(data$Product.Sales,0.05)
```

```
## 5%
##  0
```

```
quantile(data$Product.Sales,0.5)
```

```
## 50%
##   0
```

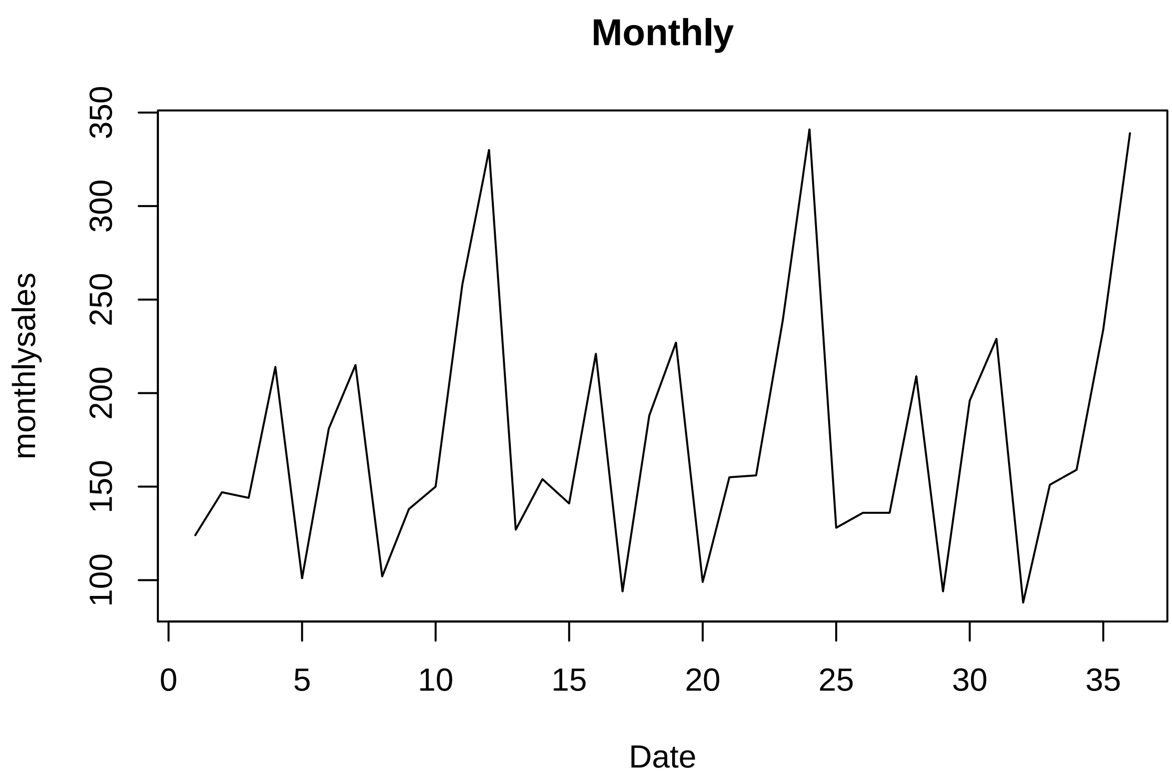
```r
quantile(data$Product.Sales,0.95)
```

```
## 95%
##  23
```

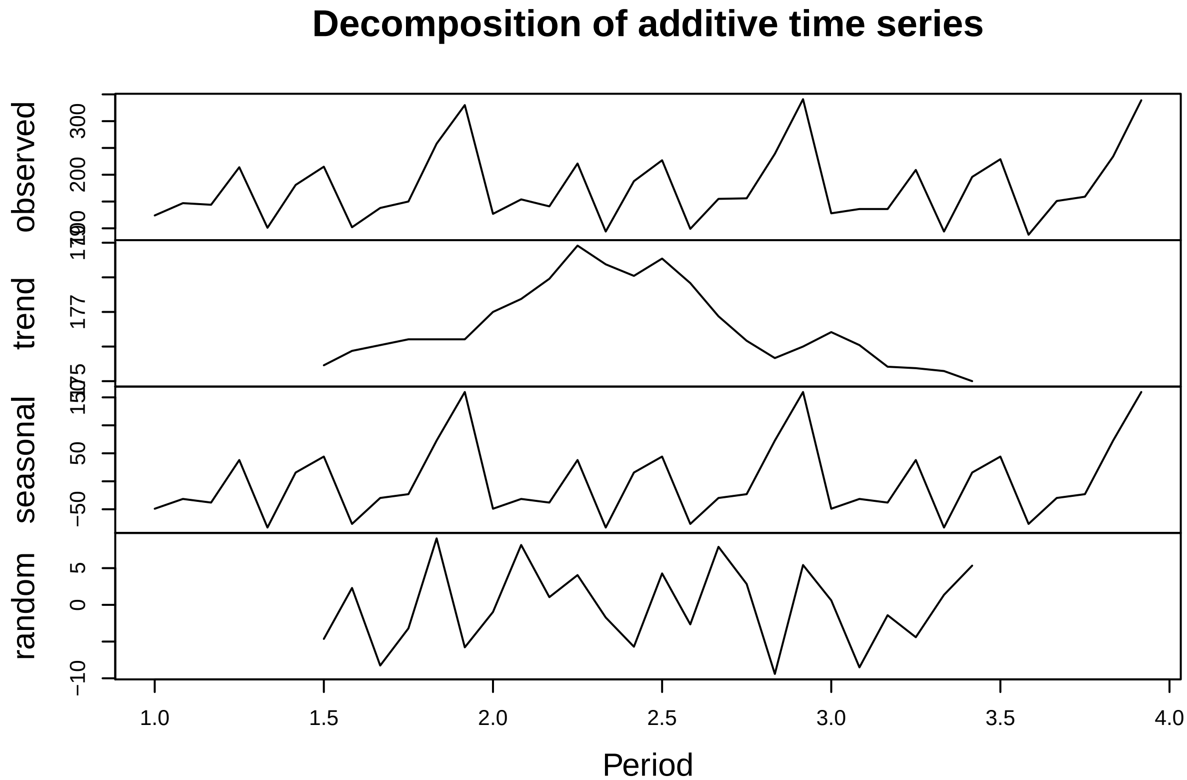
#Which makes sense, most of the data are zeros, so the 5% and the 50% are both zeros.

```r
#makemonthly
qdate = strftime(data$Date , "%Y/%m")
qmonthlydata = aggregate(data$Product.Sales ~ qdate, FUN = sum)

#rename
names(qmonthlydata)[names(qmonthlydata) == "qdate"] <- "Month"
names(qmonthlydata)[names(qmonthlydata) == "data$Product.Sales"] <- "monthlysales"
```

```r
#visualize monthly
plot(qmonthlydata$monthlysales , type = "l", main = "Monthly", ylab = "monthlysales", xlab = "Date" )
```




```r
#decompose monthly
TS <- ts(qmonthlydata$monthlysales, frequency = 12)
dec <- decompose(TS , type = "additive")
#dec <- decompose(TS , type = c("additive", "multiplicative"))
plot(dec, type = "l", ylab = "Index", xlab = "Period")
```



#It appears like the trend is that 2 small spikes and one big one every 12 months. Checking this effect manually confirms the intuition that it must be Christmas