How to do a Time-Series Analysis with R

Yonas Khanna

2018-12-11

Introduction

A Time-series `$\{X_T\}$` is a sequence of ordered data points in time, `$T$` is the value of data `$X$` point at time `$T$`. The sequence is indexed over equally spaced time points. Therefore, `$\{X_T\}$` is a sequence of discrete-time data. Time-series are often used to discover for pattern recognition and forecasting. In this technical report, a comprehensive tutorial is given of the analysis of time-series in R. Different types of modelling approaches and explanatory analysis' are discussed.

Time-Series Data

There exist two types time-series (TS), i.e. `$seasonal$` and `$non-seasonal\,series$`. The difference between these two types is that the seasonal time-series includes a returning regular pattern, which repeats over `$S$` time periods. The seasonality of the time-series is simply `$S$`, where `$S$` can be an hourly, daily, weekly, monthly, etc. time-series. To illustrate the difference of seasonal and non-seasonal series, two (fictious) will be simulated and plotted.

```r
#Load Libraries
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

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

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
library(timetk)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(anomalize)
```

Time-Series Simulation

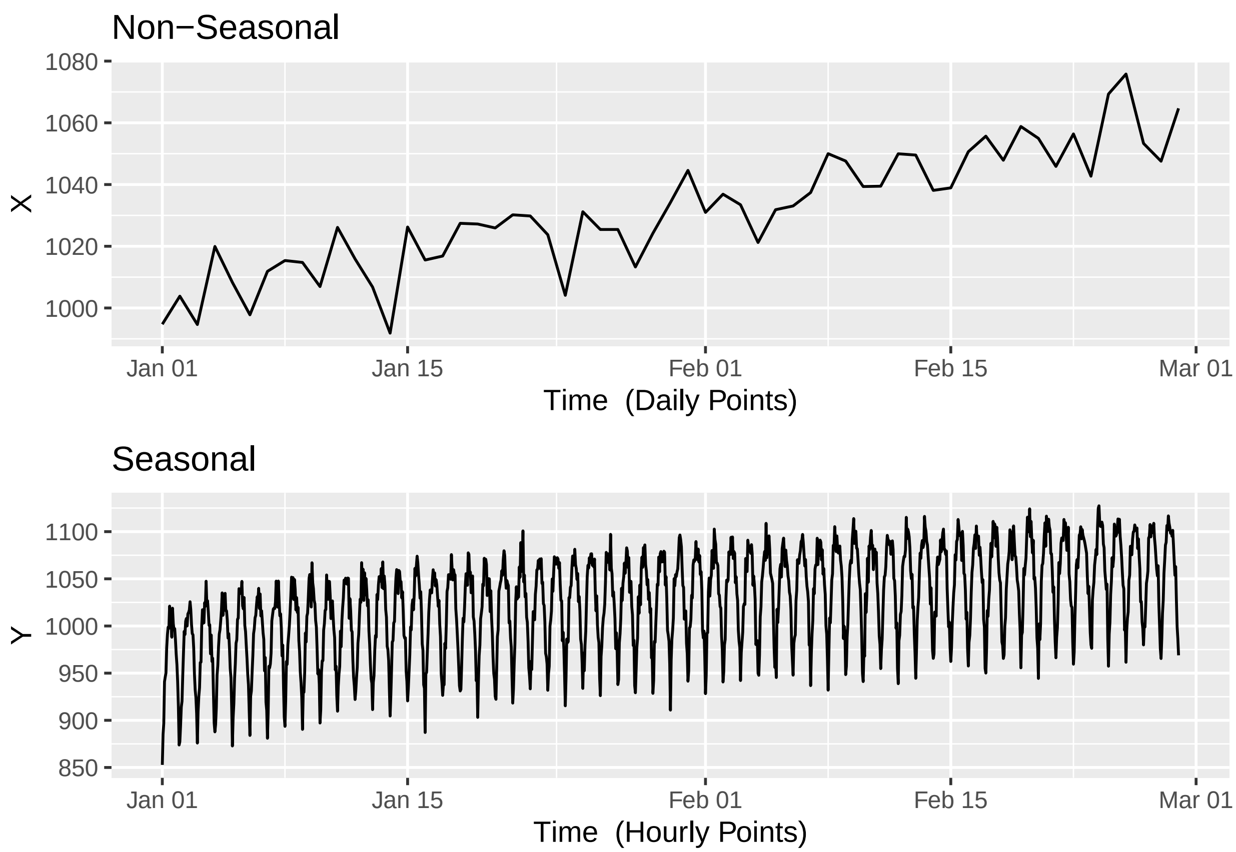
```
#Non-Seasonal Time-Series Generation (X)
set.seed(1)
index <- ISOdatetime(2017,1,1, 0, 0, 0)+0:58*24*60*60
X <- 1000+10*rnorm(length(index)) + 1:length(index)
X <- zooreg(X,  order.by = index)


#Seasonal Time-Series Generation (Y)
set.seed(1)
index <- ISOdatetime(2017,1,1, 0, 0, 0)+0:(58*24)*60*60
month <- month(index)
hour <- hour(index)
set.seed(1)
```

```
Y <- 1000+10*rnorm(length(index))-(hour-12)^2 + sqrt(1:length(index))*3
Y <- zooreg(Y, frequency=24,  order.by = index)

Plot_X <- autoplot(X) + ggtitle("Non-Seasonal") + xlab("Time  (Daily Points)")
Plot_Y <- autoplot(Y) + ggtitle("Seasonal") + xlab("Time  (Hourly Points)")

grid.arrange(Plot_X, Plot_Y)
```



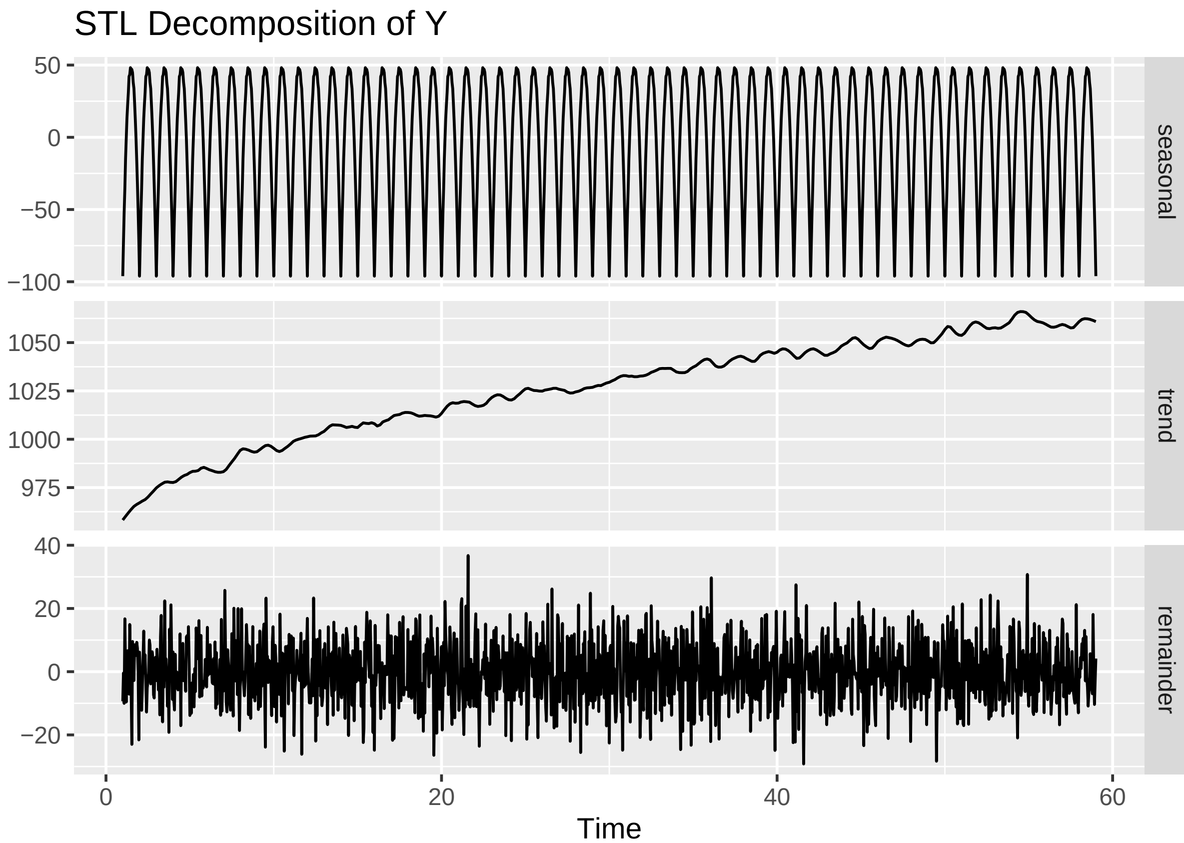
The Figures above show an example of seasonal and non-seasonal TS data. Hence, `$X$` displays random data points over time with no frequency and `$Y$` shows a recurring pattern within each 24 hours (`$S=24$`).

Time-Series Decomposition

TS may not only include the so-called returning pattern `$seasonality$`, but may also include a `$trend$`. The trend is a time-series pattern which shows that the TS moves in a certain direction over time. The TS' above show an `$upward$` trend, i.e. a `$positive\, trend$`. The seasonal and trend patterns of a TS can be discovered using the `$Seasonal\, Trend \, Loess$` (STL) decomposition algorithm.

STL Decomposition

```
Decomposition_Y <- stl(ts(Y, frequency = 24), s.window = 'periodic')
autoplot(Decomposition_Y$time.series, facets = TRUE) + ggtitle("STL Decomposition of Y") + ylab("")
```



The STL decompistion above, visualizes the seasonal and trend compononents of `$\{Y_T\}$` over the span of 58 periods of 24 hours. The `$remainder$` of the TS is component which shows no patterns at all, i.e., `$noise$`.

Time-Series Anomaly Detection

Based on the section above, it follows that a TS can be summarized as the following equation:

$$X_T = S_T + T_T + E_T$$

, where

`$S_T$`: is the seasonal `$S$` component at time `$T$`, \ `$T_T$`: is the trend `$T$` component at time `$T$`, and \ `$E_T = X_T - S_T - T_T$`: is the noise `$E$` component at time `$T$`.

It may occur that TS include outliers. Outliers or anomalies in TS can be detected using