Workshop 2: Time Series Decomposition

MSCI 523

January 31, 2019

Contents

1 Introduction

This workshop aims at introducing time series decomposition. Time series decomposition is necessary before performing any forecasting or extrapolative procedure on any time series, since it is crucial to understand the components of the series. In this workshop, you will learn how to:

- Identify visually the Regular and Irregular Components.
- Identify the type of decomposition (Additive, Multiplicative, mixed) needed.
- Filter out the Trend Component.
- Filter out the Seasonal Component.
- Extract the Noise Component.

In order to have a better grasp at the decomposition process, the first part of the workshop will take place in Microsoft Excel, where step-by-step instructions for decomposition will be given. The second part of the workshop will happen in R, where the task will be replicated, and this will also allow you to familiarize yourself more with R and how to use R in a forecasting context.

The series being decomposed in this exercise is assumed to adhere to a Mixed Multiplicative Model, that is:

$$Y_t = T_t \times S_t + E_t \tag{1}$$

where Y_t refers to the data, T_t refers to the Trend Component, S_t refers to the Seasonal Component and E_t refers to the Error Component.

2 Excel

2.1 Time Series Plot

The first step in any decomposition task is to plot the time series and visually inspect it in order to identify possible components. In order to do so, go to the worksheet **Step 0. Visualisation**. In Column B are stored the data points for the series, while the date for each point is stored in Column A.

Since you are dealing with time series, the objective is to observe the evolution of the data points over time. In order to do that, you need to produce a line graph. Select Cells **A1:B61** and go to *Insert* → *Line* and select the *Line* graph. You should get the following:

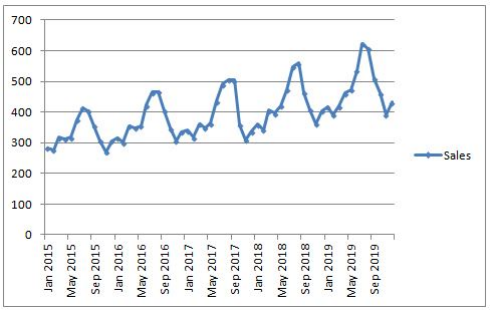

Figure 1: Time Series Plot

Question

What can you infer from this plot? Can you observe any of the Regular or Irregular Components of the time series?

The idea is to check whether any of the Regular Components (Trend, Seasonality) or Irregular Components (Level Shifts, Outliers) appear, and if any appears identify them.

2.2 Trend Identification

By inspecting the plot, you can see that the series is upwards trending over time, and so a trend does exist. Furthermore, the recurrence of the fluctuation pattern in every year points in the direction of seasonality as well. As for the irregular components, there are no indications of either outliers or level shifts at this stage.

The first step consists of handling the trend element. You need to quantify the trend so that you can remove it. In order to do that, you will need to create a Centred Moving Average (CMA). Since this is a trend, the length of the moving average should be equal to the seasonality frequency in the data. Recall that your data consists of monthly sales, and so the seasonality will repeat itself every twelve months. This entails that you need to smooth out the seasonality factor in order to discern the trend, and that can be done with a CMA of length 12.

Go to the worksheet **Step 1. Trend identification**. You will see 4 new columns, from **C** to **F**. The aim is to understand how a Centred Moving Average is calculated. Recall that a Centred Moving Average is just an average positioned in the middle of the observations. This is easy to compute for CMAs of odd length, as the midpoint is surrounded by an even number of observations. The first one to experiment with is the CMA(3).

Go to Cell **C3** and type **=AVERAGE(B2:B4)**. Drag the formula down all the way down to Cell **B60**.

Note

In a CMA, you need to forgo values at the beginning and at the end. This is because at these extremes, there are no points to use to compute the CMA. So in the case of CMA(3), Cell **B2** can not have a value as there is no preceding observation to use in the calculations. The same applies for Cell **B61** .Intuitively, for the same time series, as the length of the CMA increases, fewer points will be included in the calculations.

Task

Calculate the CMA of length 7 in the **D** row, and the CMA of length 15 in the **F** row. Also pay attention to how many values are calculated and how many are not.

Back to the discussion on trend removal, a CMA of length 12 is required. The length is an even number, which requires a little trick. You need to expand the length by a point and treat it as if it is a CMA of of length 13; however the first and last values will be divided by 2.

Go to Cell **E8** and enter the following formula: **=(B2/2+SUM(B3:B13)+B14/2)/12**. Drag this formula down till Cell **E55**. Once this has been achieved, you can see on your right the plot for the CMA(12) along the time series plot in the same graph.

Question

Does the graph with a CMA(12) indicate a trend?

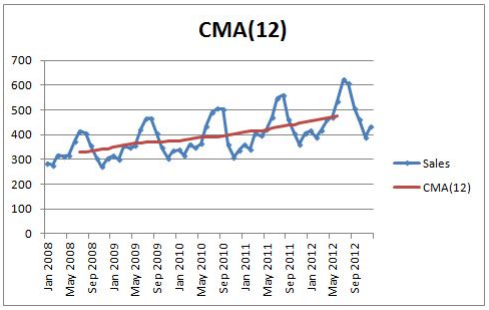

Figure 2: CMA(12)

Question

From inspecting the other graphs, are any of the other CMAs (3,7,15) more suitable for showing the trend? If yes, why? If not, why not?

You can now go to the next worksheet.

2.3 Trend Removal

Now that the trend has been estimated it is time to remove it from the time series. In this exercise we shall assume that the seasonality is multiplicative (see lecture notes for a discussion of additive and multiplicative components) and do not make any assumptions about the trend.

Given that the seasonality is multiplicative, you need to divide the actual series by the trend in order to obtain the detrended sales. Recall that the Centred Moving Average contains the trend as well as the level and the cycle. Once the CMA is removed, you are left with the seasonality component and the noise component.

Go to the worksheet **Step 2. Remove Trend**. Go to Cell **D8** and type: **=B8\C8**. Drag the formula until Cell **D55**. You will notice that for the points where no CMA was calculated, you obviously can't create the detrended sales.

Task

Look at the plot for the detrended sales series on the right. What components are remaining? Do you observe any seasonality or is there just noise?

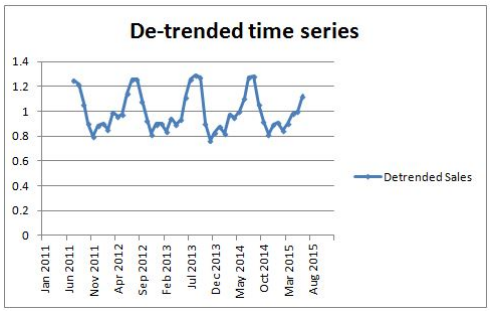

Figure 3: Detrended Sales

Once you have finished this, go to the next worksheet.

2.4 Seasonality Detection

Now that the trend has been removed, the next step consists of dealing with seasonality. In order to detect this component, you should create a seasonal plot. This is done by separating each year and plotting them on top of each other.

Go to the worksheet named **Step 3a. Detect season**. You will see in the range **H1:M13** a table, which has for rows the months and for columns the year. Fill this table by placing the corresponding value of the detrended series in the appropriate cell. So for example, in Cell **I8**, enter **=D8**, in Cell **J2** enter **=D14** etc... You will end up with the following:

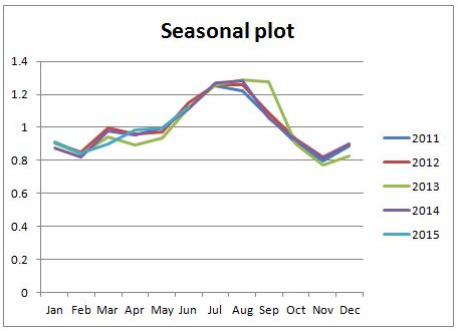

Figure 4: Seasonal Plot Table

The seasonal plot is drawn on the right. You will see it overlays every observation for every month for different years. If similarities emerge, then seasonality, whether weak or strong, exists. If not, then no seasonality is present in the data.

	2011	2012	2013	2014	2015
Jan		0.905	0.906	0.876	0.914
Feb		0.853	0.841	0.823	0.847
Mar		0.995	0.944	0.976	0.901
Apr		0.963	0.896	0.951	0.982
May		0.974	0.933	0.999	0.998
Jun		1.149	1.117	1.109	1.126
Jul	1.253	1.259	1.258	1.272	
Aug	1.220	1.258	1.288	1.285	
Sep	1.061	1.086	1.276	1.058	
Oct	0.907	0.932	0.901	0.923	
Nov	0.796	0.818	0.769	0.812	
Dec	0.889	0.899	0.828	0.899	

Figure 5: Seasonal Plot

Question

Does any seasonality exist in your data? If yes, why? If not, why not?

Now you can proceed to the next worksheet.

Note

In case you haven't managed to fill the seasonality plot table data, these are the necessary cell references.

	2011	2012	2013	2014	2015
Jan		=D14	=D26	=D38	=D50
Feb		=D15	=D27	=D39	=D51
Mar		=D16	=D28	=D40	=D52
Apr		=D17	=D29	=D41	=D53
May		=D18	=D30	=D42	=D54
Jun		=D19	=D31	=D43	=D55
Jul	=D8	=D20	=D32	=D44	
Aug	=D9	=D21	=D33	=D45	
Sep	=D10	=D22	=D34	=D46	
Oct	=D11	=D23	=D35	=D47	
Nov	=D12	=D24	=D36	=D48	
Dec	=D13	=D25	=D37	=D49	

Figure 6: Seasonal Plot Cell References

2.5 Seasonality Removal

So far you have identified the existence of seasonality. Just like trend, this component has to be filtered out. Go to the worksheet named **Step 3b. Remove seasonality**. A new column has been added to the seasonal plot table, which has a header called "Profile". This is the seasonal profile, which is the average of the detrended time series in each month.

Go to Cell **N2** and type **=AVERAGE(I2:M2)**. Drag this for each month. Note that the average is calculated over 5 years, not 4 years, despite having a blank entry in each row. The profile plot is added to the seasonal plot (it is the dark line graph).

Task

Examine the seasonal plot. Since the seasonal profile represents the average, you will notice that the value in each of the rows varies from the profile. This is due to the noise component.

The seasonal profile is the seasonal component that you are trying to extract. Replicate the profile in each year in the **E** column. Bear in mind that seasonality is recurring each year, so the values for example in January 2011 are the same as January 2012.

Go to the next and final Worksheet.

2.6 Decomposition

So far, both trend and seasonality have been estimated and removed, thus leaving you with the noise component. You can now reconstruct the time series with the trend and the seasonal component. The difference between the actual series and the reconstructed series is your noise component.

Go to the worksheet named **Step 4. Decomposition**. Remember that the seasonality is multiplicative, and so reconstructing the series involves multiplying the trend and the season. In Cell **F8**, enter the formula **=C8*E8** and drag until Cell **F55**. This is your reconstructed time series. In the G column, you can estimate the noise by substracting the actual sales from the reconstructed sales. In Cell **G8**, type **=B8-F8**, and again drag until Cell **G55**.

Graphs for the reconstructed vs actual series, trend, seasonality and noise components are plotted on the right (You have to scroll down to see the plots as they are stacked vertically).

Task

Observe the Noise Plot. Can you spot any irregularities such as level shifts or outliers? If so, can you locate it in the original plot of the time series?

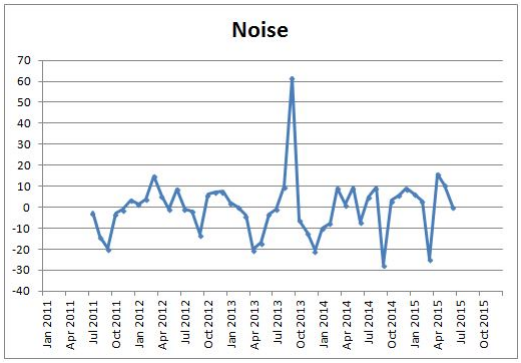

Figure 7: Noise Component

3 R

So far, the decomposition exercise has been performed on Excel. The second task in this workshop is to replicate the exercise in R. Unlike Excel, the functions are already implemented in R, and so with a few lines of code, you will achieve the same outcome as Excel.

Launch R Studio. The first thing you need to do is to load the packages that you will be working with. It is assumed that the list of packages given in the previous workshop have already been installed. For this exercise, you will have to load the packages "tsutils" and "forecast". Go to the *Packages* tab on the right and tick on the checkbox next to "tsutils", and also for the one next to "forecast". Alternatively, you can also type the following statement:

```
library(tsutils)
library(forecast)
```

The package is now loaded, and it is time to load the data. Open the Notepad file called "Sales.txt". This is where the sales data that was used in Excel is stored. The first task is to load the data in R. Copy the file into your R directory. In case you have forgotten where the directory is, just set it all over again by clicking on *Session* → *Set Working Directory* → *Choose Directory*.

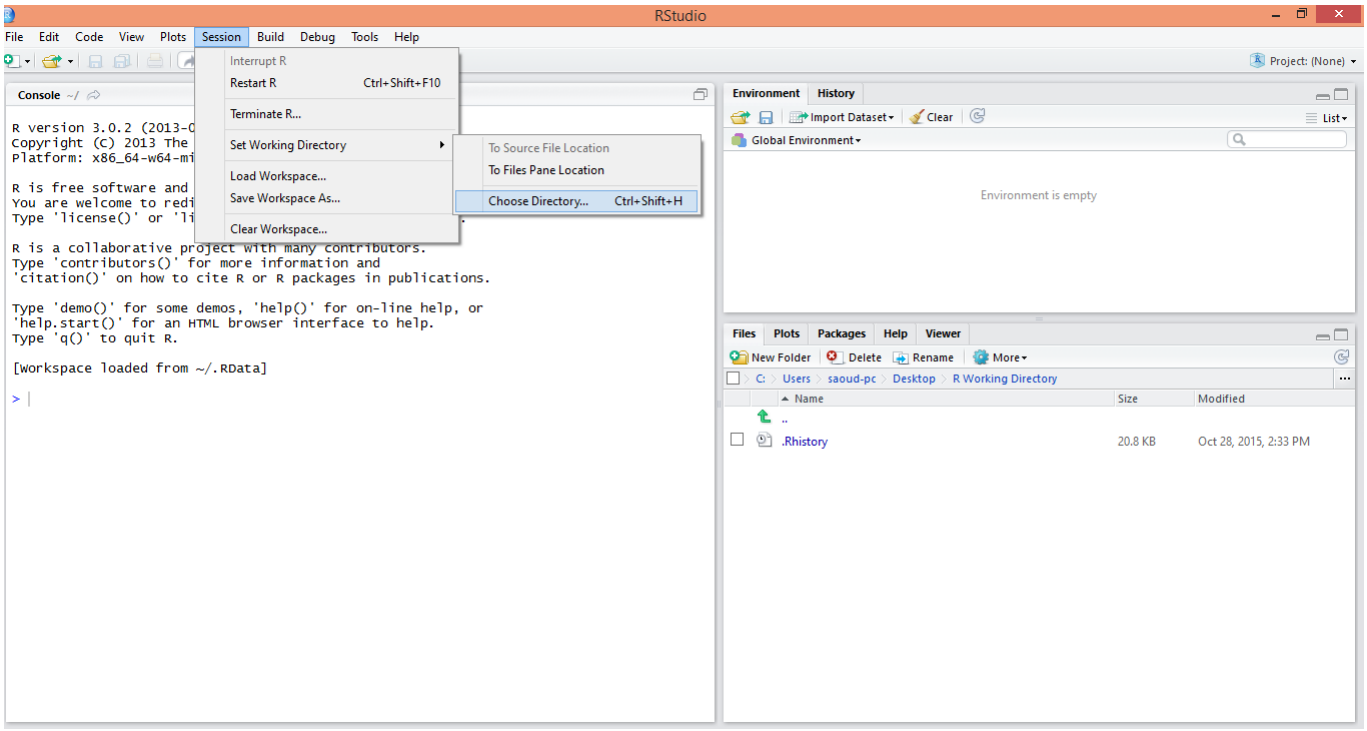

Figure 8: Set the Working Directory

Go to the console and type the following:

```
data <- read.table("Sales.txt")
```

You will notice a variable called "data" has been created in the **Environment** box on the right, which consists of 60 observations. Given that the data is a time series, it would be useful to convert the data into a time series format. A data type which was not explained in the previous workshop is the `ts` format. This converts the data into a time series. A time series format requires a seasonal frequency. In your case, since the data is monthly, the frequency would be 12. In order to do so, enter the statement:

```
data <- ts(data, frequency = 12, start = c(2011,1))
```

The above statement has converted the variable `data` into a time series. The `frequency=12` specifies the seasonal frequency, while the `start=c(2011,1)` specifies that the first entry occurs on January 2011. The start is not a compulsory argument as it helps in organizing the data; however the frequency is!

If you wanted to inspect the seasonal frequency of your series, just type:

```
frequency(data)
```

The result will be 12, as expected. If you are wondering where the result is displayed, look at the **Console** in the bottom.

> **Note**
> The `ts` format only works if the "forecast" package is loaded. Failing to load it will result in R not identifying the statement and returning an error message.

Now that the data has been processed into R in the desired format, you can start the exercise.

3.1 Time Series Plot

The first task consisted of plotting the time series. This is done by using the `plot()` function as follows:

```
plot(data)
```

On the right tab, the time series plot will be displayed as follows:

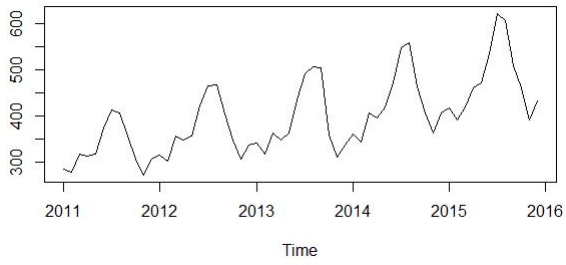

Figure 9: Time Series Plot

3.1.1 Plotting in R

For the `ts` format, when entering the `plot()` function, the output was a line graph. However for other formats this is not the case. In order to see why, follow this example

```
#Create data
data1 <- 1:20

#Plot
plot(data1)
```

As you will see, the output is a scatterplot. In order to specify that you need a line graph, the following should be inserted:

```
plot(data1, type = "l")
```

By adding the `type="l"` argument, the plot will now be a line. Furthermore if you wanted to colour it, then another argument should be added as such:

```
plot(data1, type = "l", col = "blue")
```

This will now return a blue line graph.

3.2 Centred Moving Average

In order to create a CMA, the function `cmav()` from the "tsutils" package is used. To test this, create a variable that will have the CMA values stored in it as such:

```
data_cma <- cmav(data, ma = 12, fill = FALSE)
```

The first argument passed is which series is being used, which in your case is the variable called "data". The `ma()` argument specifies the length of the CMA. If this is omitted, then the default value is the seasonal frequency. So for a CMA(12) in **THIS** example, the `ma()` can be withheld, but for any other type of CMA, the length has to be specified. The `fill=FALSE` argument means that the first and last observations that were expected to be left blank will be left blank.You will see that the first values of the "*data_cma*" variable are **NA**, since they don't exist. If you wanted to impute values on these observations, then use the `fill=TRUE` argument.

Task

Can you plot a CMA(3) for the data?

If you wanted to overlay the graphs, this is how it is done:

```
#Plot the original time series in blue
plot(data, col="blue")

#Plot the CMA(12) in red
lines(data_cma, col = "red")
```

By adding the `lines()` statement, you can overlay a line graph on your original time series. If you were to use the `plot()` statement, then a new graph will be plotted instead.

3.3 Seasonal Plots

In the "tsutils" package, a function has been created that will automatically calculate the seasonal indices and profile, and will display the seasonal plot. Just type:

```
seasplot(data)
```

You will see displayed on the right the seasonal plot as follows:

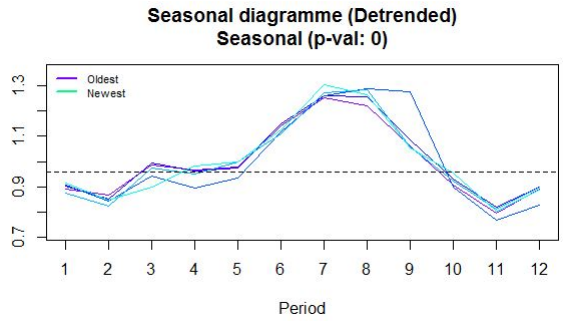

Figure 10: Seasonal Plot

If you wished to stored the seasonal indices in a variable, say "*data_seas_index*", then type:

```
data_seas_index <- seasplot(data)$season
```

3.4 Decomposition

The whole exercise that was done on Excel could have been executed with one line of code in R. This is thanks to the `decomp()` function in the "tsutils" package. Create a variable named "decomposition":

```
decomposition <- decomp(data,decomposition = "multiplicative",outplot = TRUE)
```

The first argument is which series to decompose, which in your case is the "data" series. The "decomposition" argument specifies whether the decomposition should be additive or multiplicative. So if you wanted to try with an additive counterpart then alter the statement to:

```
decomp(data,decomposition = "additive", outplot = TRUE)
```

The `outplot=TRUE` specifies whether the plots should be displayed or not as in the figure below:

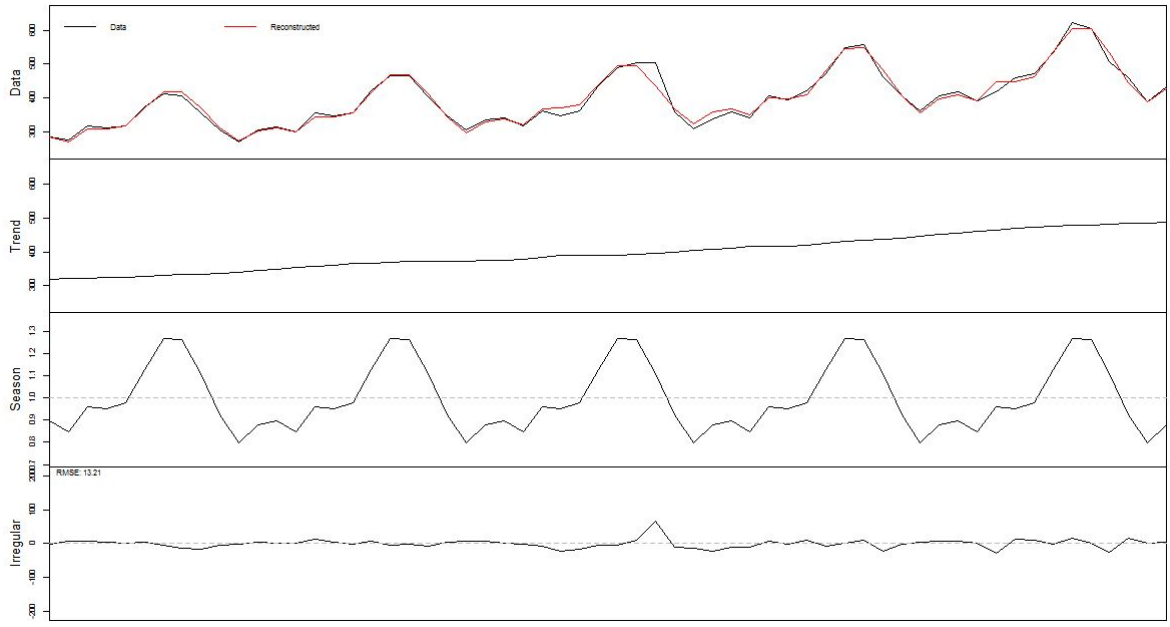

Figure 11: Time Series Decomposition Plot (decomp function)

The other arguments of course can be explored by typing the `decomp()` function in the Help tab on the right or by writing this line in the console:

```
?decomp
```

How can you extract the trend data or the seasonality data from the "decomposition" variable? If you look at the **Environment** tab, you will see that "decomposition" is a 'List of 5". In order to see what it consists of, you need the `str()` function. This function will display the structure of the variable.

```
str(decomposition)
```

You will get the following:

```
> str(decomposition)
List of 5
 $ trend    : Time-Series [1:60] from 2011 to 2016: 319 320 322 323 325 ...
 $ season   : Time-Series [1:60] from 2011 to 2016: 0.898 0.846 0.96 0.952 0.976 ...
  ..- attr(*, "names")= chr [1:60] "p1" "p2" "p3" "p4" ...
 $ irregular: Time-Series [1:60] from 2011 to 2016: -2.226 6.178 8.071 5.023 0.332 ..
.
 $ f.season : NULL
 $ g        : NULL
```

Figure 12: The "str" function

The 5 components are listed as "trend", "season", "irregular", "f.season", "g". The first three components are the ones to look at in this exercise. In order to extract for example the irregular (which is the noise component) component, then run the following command:

```
decomposition$irregular
```

The $ specifies that you are retrieving a component, which is the trend, instead of the whole variable.

Task
Can you extract the seasonality from the decomposition variable and plot it in blue?

3.5 Pure Multiplicative or Mixed Multiplicative?

A question can arise when using the `decomp()` function: Is the Multiplicative decomposition performed by this function a Pure Multiplicative Model or a Mixed Multiplicative Model? You can check it out.

Recall that the Mixed Multiplicative Model that is assumed in the Excel exercise is the following:

$$Data = Trend \times Season + Error \tag{2}$$

This implies that:

$$Error = Data - Trend \times Season \tag{3}$$

In order to confirm this, isolate the Regular Components from the decomposition. So far, all the data is stored in the variable called "decomposition". The first step would be to extract the regular components (Trend and Seasonality) and multiply them.

```
#Extract the Trend
decomposition$trend

#Extract the Seasonality
decomposition$season

#Multiply the Trend and Seasonality
regular_components <- decomposition$trend * decomposition$season
```

This has now created a new variable called "regular_components", which is equal to $Trend \times Seasonality$.

Note

R is case-sensitive. In other words, it distinguishes between variables that have a capital letter and variables that don't.

In order to confirm this, go to the console and type:

```
Regular_components
```

You will see the following message:

```
> Regular_components
Error: object 'Regular_components' not found
```

Figure 13: Object not found

That is because R has stored a variable called "regular_components", and thus does not recognize "Regular_components" since it starts with a capital R.

Getting back to the task, now that the Regular Component has been created, you can calculate the errors for the Mixed Multiplicative Model as:

```
mmm_errors <- data - regular_components
```

The last step is to compare these errors with the errors from the decomposition. Go to the console and type:

```
decomposition$irregular - mmm_errors
```

You will see that the result is a vector of 0. This means that the Multiplicative option in this function is actually a Mixed Multiplicative Model.

Task

Calculate the errors if the assumed model is a Pure Multiplicative Model, instead of the Mixed Multiplicative Model currently used.